INTERNATIONAL
HELLENIC
UNIVERSITY

Data Mining in Sports: Daily NBA Player Performance Prediction

Georgios Papageorgiou

SID: 3308200022

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2022

THESSALONIKI – GREECE

Data Mining in Sports: Daily NBA Player Performance Prediction

Georgios Papageorgiou

SID: 3308200022

Supervisor: Assoc. Prof. Christos Tjortjis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2022

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University.

This dissertation is related to the NBA League and its players; it focuses on Daily performance prediction in terms of Fantasy Points for each player and Lineup Optimization for betting purposes in Fantasy Tournaments. The primary purpose of this dissertation is to explore, develop and evaluate ML predictive models, each one focused separately on each player for Daily Player's Performance Prediction in terms of Fantasy Points. In adittion, tries to develop and evaluate a Lineup Optimizer focused on total Fantasy Points for a range of Dates.

In this project tried to experiment with Pycaret library. Therefore, we develop *four finalized models* for each selected player. We used two primary datasets, with advanced statistics and only basic statistics. Also, the models are developed with historical data from Season 2010-11 to Season 2020-21, and in historical data from last seasons (2018-19, 2019-20, 2020-21) while in cases that the player does not participate in at least 100 games, additional season's data is included. Furthermore, in the next stage of this project, using the predictions, we developed a Lineup Optimizer with restrictions applied, focused on maximizing the sum of NBA Fantasy Points of our selected players. Results show that we can accurately predict the performance of each selected player in terms of Fantasy Points and build a well-performing Lineup for selected game dates.

Acknowledgements

At this point, I would like to truly thank my Supervisor, Professor Christos Tjortjis, for his guidance in every point of this project, providing me with essential assessment and crucial suggestions. In addition, I would like to extend thanks to Ph.D candidate Vangelis Sarlis for his expert assistance at any issue that occurred, providing me immediate feedback anytime I needed it.

Georgios Papageorgiou

7-1-2022

Contents

List of Figures

List of Tables

List of Appendices

1 Introduction

This dissertation consists of 7 chapters: The first chapter is the introduction of this topic. The second chapter is a Historical Background and Literature review for sports analytics domain. In chapter three, general DM and ML methods are used or mentioned in the dissertation. The fourth chapter offers the main problem of this dissertation, with the methodology process that was followed. The procedure of creating and using specific models as target predicting the target values is analyzed in chapter five. The Results of the following methods and the evaluation of these results are located in chapter six. Finally, the conclusion and future work are concluded in chapter 7.

Sports analytics is an emerging field because the domain's value for teams, players, and organizations is enormous. In recent years, it has been discovered that analytics and performance prediction in the sports domain is necessary for the evolution of any sport, team, and even the players. The big organizations-teams have departments focused on their and opponent team analytics, trying to optimize their playstyle and detect problems that staff, players, and coaches cannot see. For this reason, data has incredible value for the teams, and via different methods (cameras, sensors), collect as much data as is possible for evaluation.

This dissertation focuses on Basketball and particularly on the NBA. The scope of this dissertation is to produce predictions for the daily performance of any player participating in the NBA while he has played at least ten matches in the past. Basketball is selected because it offers plenty of statistics for both players and teams. Also, it can be considered as a challenging domain of analysis and prediction making because data should be appropriately selected and used focusing on the target variable.

In Basketball, many metrics and formulas refer to a player's overall performance in a match. These can be; Efficiency (EFF), Player Impact Estimate (PIE), Player Efficiency Rating (PER), and Usage Rate. In the following chapters will be further information about these metrics. However, this dissertation focuses on another metric highly correlated with these metrics above called Fantasy Points (FP).

Moreover, the daily performance of a player can be translated to Fantasy Points and is a metric that betting companies use to rank each player based on his performance. Basketball is a sport that entertains people in many ways, from watching it and supporting their favorite team to betting on it with plenty of choices available (Win, points). However, a new opportunity has come up for people to become coaches and choose their teams in recent years. Their choices are evaluated and rewarded based on the Fantasy Points their players will achieve in each performance.

Trying to predict daily performance is a difficult task to accomplish, while the daily performance of a player can be affected by many factors. However, in this dissertation, regression methods will be implemented to predict NBA players' daily performance using historical advanced player and team data. With careful selection of data and creating efficient features, high accuracy regression models will be compared; as a result, the selection of the best to predict as accurately as possible the Fantasy Points for each NBA player participating in a match.

2 Literature Review

The aim of the section is to provide a comprehensive review of the relevant corpora regarding Sports Analytics and, more specifically, NBA players’ performance prediction, NBA Fantasy, and NBA Fantasy Lineups.

2.1 Historical Background

It is well known that Sports Analytics is an emerging field, and it is used by all big sports organizations, professional teams, helping develop the team, improving the results, noticing problems that are hard to find by people. The technology improvement over the years has created new playstyles and tactics. Also, the evaluation of the results with the help of analytics is a big deal for every kind of sport. Nowadays, the experience of a coach is not enough to be competitive at a professional level.

However, years ago, when computers were not as capable tools as they are nowadays for gathering data and making analyses, collecting data was manual, handwritten, hard to observe, and time-consuming. For this reason, it is considered normal that there are no statistical records for most sports games. While, chronologically, sports analytics appeared in the 19th century, when the idea of analyzing a player's play helps evaluate the player's skills comes up [1].

2.1.1 Baseball

Baseball was the first sport that recording match statistics starts. In early 1837, then baseball did not exist as we know it today. The first Baseball club, the Constitution of the Olympic Ball Club of Philadelphia, played the first version of baseball. They called 'town ball' and kept a scorebook with the runs scored every player did. Subsequent years (1845), the first box scores make their appearance in New York Morning News, with only Batter's columns, include runs and outs [2].

At the start, the sport had not the form it has today, while new regulations and statistics were created in the following years. Until the early years of 1900, baseball was in development, while rules and statistics were continuously expanding. For example, in 1858, nine more columns per player formed a new box score; in 1867, terms like 'base hit' and total 'bases' came up. Moreover, in 1872 the summarized, averaged stats created referring to 'total bases per game' and 'batting average.' Also, often these statistics change form over the years or discard. In 1879 National League set as official statistic the "reached first base". However, remove it one year later, replace it with 'based touched', and discard it later [3].

The need to keep more statistics born after 1900, when in 1905 started count times a player did not complete a match, a new statistic called "times take out". Also, changes needed to be made over the years; in 1912, President John Heydler of the National League replaced the "earned run per game" with a new measure known as "earned run average". While in the same year, measure "Who's Who in Baseball" record active players' batting and fielding averages.

The first attempt for an extensive record book was made in 1914 when Pittsburgh stat freak George Moreland published the "Balldom", which introduces the critical list, "Eight Games in Which First Baseman Made no Putouts". After this, in 1918, brothers Al Munro and Walter Elias started a business known today as Elias Sports Bureau, which began by selling baseball statistics. Finally, the National League hired them to keep the official numbers updated [4][5].

Almost 30 years after, in 1947, Baseball teams started to think that historical data could optimize the results and evaluate the players' performance. For this reason, Brooklyn Dodgers hired Allan Roth as a statistician. His job was to keep all sorts of new statistics to rate players. He used historical data like performance in different ball-strike counts, batting average with runners in scoring position, and more. The same strategy was followed by Branch Rickey, an executive, and manager of ST. Louis Browns, who hired a statistician named Travis Hoke.

The years pass, and people are more interested in baseball statistics and performance. For this reason, "Topps" include on their annual baseball cards complete statistics lines. In 1960, Harvard University professor William Gamson created the "Baseball Seminar", which reminds us of today's baseball fantasy [4].

Following 20 years (from 1960 to 1980) was crucial about the importance of Baseball statistics. In 1969, The first comprehensive historical records book was published, known as "The Baseball Encyclopedia". It concluded over 17 statistics for each player for each year from 1876. In 1970 the Mills brothers released the book "Player Win Averages". In addition, in 1971, Bob Davids established the Society for American Baseball Research (SABR) in Cooperstown, New York. Society for American Baseball Research is still a Non Profit industry, having as purpose to help people do baseball research. In 1979 Houston Astros also hired the first modern stat analyst Steve Mann, while two years later, STATS Inc. developed a computer system known as "Edge 1000" to help clubs keep their advanced statistics. [6]

Still, Sports Analytics had not gained the attention that it should of the fans and teams. This until 1981, when Bill James published "The new Bill James Historical Baseball Abstract" to make popular the sabermetrics to the ordinary people. Soon, Bill's book became an annual bestseller, making him one of the most influential persons in baseball history. From then, people started being interested in SA, more and more clubs started hiring people for today's analyst job and concern about analytics. For example, in 1982, Eric Walker published "The Sinister First Baseman", giving a new philosophy in the sport's strategy, making Sandy Alderson, executive of Oakland, hire him as a consultant.

After the widespread publication of Bill James, it was clear for the clubs that SA makes the difference on and off the court, evaluating players' performance and decision-making for the strategies from clubs staff. For the next 20 years, with the help of the Society for American Baseball Research (SABR), USA Today, and STATS Inc, statistics and sports analytics started to become widely known. New publishes were done, and every professional club used them, making their teams and players more efficient. The famous publicity in 2003 ‘Moneyball: The Art of Winning an Unfair Game’ ensured everyone back then that analytics could make the difference. Lastly, in 2004 the first full-length history of Baseball statistics book was published by Alan Schwarz [7][8].

2.1.2 Football

Football origins are not clear; there are reports that Football first developed in ancient times in Greece in the 7th century. Also, there are reports that in ancient Rome, there was a game with a ball that existed in the military exercises (“Harpastum”), and this Roman culture brings Football to the British Island. Rumors that the first stages of Football started developing as a sport located in England in the 12th century. However, the rules and regulations were much different from the form of Football we know [9].

The first rules and regulations of the game were attempted to be determined at a meeting in Cambridge in 1848. Nevertheless, not a proper formula of rules was decided. The first regulation was formed in England when the first Football Association started and agreed that it was not allowed to carry the ball. In addition, the Association agreed about the size and the weight of the ball. Back then, two playstyles dominate, British and Scottish. British chose to run forward with the ball, and Scottish passed the ball between their teammates.

The 1871-1872 season organized the first Football Association Challenge Cup (FA Cup), participating in twelve British teams. Teams from other countries could not join because the Cup was located in England, and traveling was not easy at those times. The final result for the first final in Football history made Wanderers the first football champions, defeating Royal Engineers 1-0. Worth mentioning that National matchups started organized, with the first national match be played one year later than the FA Cup.

In 1862, in Nottingham, the first professional club was established when there were no other professional teams, while teams were made up of ordinary people and not particularly good fit athletes. In the 1880s, money started to motivate people to play when money was a key factor. Teams started having revenue and paying the players to perform as better as they could. Lastly, in 1885 professional Football was approved, and in 1888 the first Football League was created in England [10].

In 1904 famous Fédération Internationale de Football Association (FIFA) was founded and signed officially by many countries, like Spain, Denmark, France, Belgium, and more, when England joined in 1906. While the first World Cup was organized in 1930, England and other British countries did not participate because two years before left the organization even if they invented the game. Finally, they rejoined in 1946 and participated in the World Cup of 1950 [11].

Football is also a sport that offers plenty of statistics, and most of them have great value for either the club, the opponent, or the crowd. In 1950, a person introduced statistical analysis to Football, Thorold Charles Reep, a war veteran passionate about Football. In 1933, when Charles was located close to London, the captain of Arsenal approached Charles and had a meeting about Arsenal’s playstyle. Charles, fascinated, started observing games for the next seventeen years and used a mix of symbols to keep their notes updated during the match. His primary target, how to maximize goal opportunities. He believed that more goal opportunities could be born by the pair of wingers. His idea produced results for station teams and local amateur teams when they used it.

Finally, in 1950, at the start of the following season, 1950-1951, Charles had the opportunity to advise a Football League team. Charles used his contacts and came in touch with Brentford, a team that had many difficulties ranked in last places of the League. Charles' attacking advises won thirteen of their last fourteen matches, and then his carrier as the first football analyst-consultant started. He consulted many teams in his career, always tried to update and correct his data. After many years of work, he had collected data from 2194. His calculations are based on counting the number of passes and splitting them as sequences, set them into different categories, and finally recond the number of goals scored for each category. While he calculated the average of shots needed to achieve a goal, and the chances score based on the passes are made [12][13].

2.1.3 Backetball (NBA)

In December of 1891, James Naismith, a college teacher at Springfield in Massachusetts, invented Basketball. It all starts when his students run out of choices when they should play a game indoors. His choice then was to force them to play the already invented games, like Football or baseball. However, the circumstances exclude these options. Then he remembered a game called rock-tossing that he was playing as a child, and he proposed a game that players would throw a ball to a target. He used two peach baskets, nailed them on ten feet above the floor at each end of the gym and a football ball. That was the day; Basketball was invented.

Following years, Basketball spread quickly around the world, while Naismith's students helped by introducing the sport to new people. High Schools and colleges started to organize teams, today's basketball ball was invented, and the rules changed, making the sport more entertaining. Soon professional leagues and teams were formed, and the game became very popular around the world. Lastly, in 1936 Basketball became an Olympic sport, and ten years later, on the 6th of June in New York, The National Basketball Association (NBA) was invented [14].

Nowadays, Basketball is a sport that is highly dependent on statistics and analytics. Teams use data for decision-making about players, playstyles, and game strategies. However, Sports Analytics is something new comparing with other sports; it all happened when Lawrence Dean Oliver, an American statistician, published his book, "Basketball on Paper". In 2004, Oliver introduced sports analytics in the Basketball world, while in the same year, and became the first full-time statistical analyst in the NBA. His carrier as a sports analyst starts from the Seattle

SuperSonics team, and since then, he has worked with Denver Nuggets on NBA, with ESPN, Sacramento King, and as an assistant coach to Washington Wizards [15].

A year after the publicity of "Basketball on Paper", the famous SportVU was created. Gal Oz and Miky Tamir from Israel develop a real-time optical tracking system that identifies the movements of every player on the pitch. In the 2010-2011 season, four NBA teams started using SportVU, when in the next season, six more NBA contracted to use it. Lastly, since the 2013-2014 NBA season, all NBA arenas have installed the SportVU camera system, and their teams benefit from advanced statistics. SportVU offers plenty of innovative statistics based on speed, distance, player separation, and ball possession, and teams can benefit from their analysis with ML algorithms [16].

2.2 Related Work

2.2.1 Introduction

In this part of the Dissertation, we will present concisely previous work and relative research. Detailed, the Literature review chapter will conclude three kinds of studies. Firstly, we will cite topics relevant to basketball players' performance prediction, while many formulas can translate the term performance in Basketball. Secondly, we will discuss research having as a topic, players Fantasy points (FP) prediction, a metric high correlated with other performance metrics. Also, we will focus on Daily Basketball Fantasy Line-up prediction, a new topic in recent years.

2.2.2 Basketball Players’ Performance Prediction Overview

Predictions for potential basketball players' performance using DM and ML algorithms is a new research subject. While some years ago, the evaluation methods and predictions for players' performance were only the coach's deal based on his experience. However, in recent years sports analytics and player performance prediction is becoming a significant research subject.

In 2018, Leila Hamdard(B), Karima Benatchba, Fella Beckham, and Nesrine Cherairi [17] using DM methods, tried to predict NBA players' performance working with data from seasons 2005-06 to 2013-14. Firstly, using the K-means clustering algorithm, they cluster players by their historical performance and proven skills. Having as a target to detect any changes to the performance-based clusters predicting their next games performance. Also, the same experiment was transformed and, as a classification task, a Naive Bayes algorithm, using clusters as labels based on their historical performance. While, finally, testing three specific players' performance prediction with both two methods, they compare the results, and two of the three players are classified in the same label-cluster that was assigned in the previous clustering experiment. Lastly, they used a multiple regression model and exponential smoothing algorithm based on athletes' historical statistics to predict their performance. Results show that the exponential smoothing algorithm performed better.

In 2020, Mahboubeh Ahmadalinezhad and Masoud Makrehchi [18] had designed a unique network based on NBA data from all the lineups and matchups of the teams from 2007 to 2019. Using ML and graph theory, they create a metric called Inverse Square Metric and an edge-centric multi-view network with a target to predict the performance of an NBA lineup anytime. Specifically, the edge-centric approach provides a thorough examination of any situation of the teams from 16 perspectives working with data like defense or offensive rebounds and many other features. Results make clear that they constructed a highly accurate system with an edge-centric multi-view method with an 80% average accuracy score, while ISM scored 68%. Compared with the baseline methods, the results are improved by 10%, clarifying how efficient the graph theory is in the lineup performance prediction problem.

Marti Casals and Jose A. Martinez [19], in 2013, tried to predict both points scored and winning scores using mixed models with random effects. Also, they tried to find out which feature-metric was essential to make these predictions. In their study, they considered all the possible variables that may affect player performance. As a result, they created a dataset of 2187 examples, focusing on 27 NBA players in the 2007 regular NBA season. Results clarify that variables like the player, his position, the difference in team quality, if the player started the match, the minutes he played, and his usage rate were crucial to predict the points scored successfully. In addition, the crucial variables to predict the winning score were the player, his age, his position in the field, the difference in team quality, the relationship between his age and his position, the minutes that he played, and the usage percentage. Lastly, they made their predictions using a single model with all the data instead of creating daily models.

In 2020, Vangelis Sarlis and Christos Tjortjis [20] successfully predicted the NBA MVP for the 2017-18, 2018-19, and 2019-20 NBA season. In addition, they predict the best Defender of the year for the following NBA season 2017-18, 2018-19, and 2019-20. These forecasting scenarios are performed based on certificated data from seasons 2017 up to 2020. Every season of the dataset had 82 games and was split into four groups(Q1-Q4), while each group had approximately 20 games of the season, starting from the first Q1 (~20 games) to the last Q4(~20 games). They selected 20 NBA players who participated at least in 30 games per season and at least 15 minutes average participation time in each match. To make their predictions in each category, they created two formulas, Aggregated Performance Indicator (API) and Defensive Performance Indicator (DPI). The first formula, API, is adopted to predict the MVP of the season, and it is a composition of box score statistics and important rating basketball analytics, a synthesis of variables that illustrate the athlete's general performance. The second formula, DPI, is used for forecasting the Best Defender of the year, and it is a combination of advanced analytics variables focused on the player's contribution to the Defensive part of the team. Finally, the predictions were successful; while it predicts the NBA MVP for season 2017 up to 2020, it is worth mentioning that this method is the only one that requires current data to make an accurate prediction of the NBA MVP of the year. In addition, about the Best Defender of the year predictions, the DPI formula successfully verified the Best Defenders of the year. Also, noteworthy that this method is the only one that predicted the right, the best Defender of the Year.

2.2.3 Fantasy Points and Daily Fantasy Lineups

Over the last fifteen years, a new method for fans participating in Basketball became very popular worldwide. Companies offer the chance to users to take the role of the Team Manager or the Coach and create their Fantasy Basketball lineup. Fantasy is a vast sector in the betting industry, with millions of users trying to predict the best basketball lineup daily in terms of performance. While the years pass, more and more companies conclude in their services the

Fantasy sports. Basketball Fantasy is highly competitive, while users compete against the other, and the best lineup predictions are rewarded. While Basketball is a sport filled with analytics, professionals and amateurs try to make predictions using raw statistics or other advanced analytics and ML, building models and making up strategies.

In 2017, Charles South, Ryan Elmore, Andrew Clarage, Rob Sickorez, and Jing Cao [21] introduced a way to predict player's Fantasy Points (FP) and develop a system predicting the best combination of players in the Daily Fantasy Lineups, having as target the best overall score with a sure salary cap. They trained their models with data from Season 2013-14 and used their system on season 2015-16, evaluating their predictions with the actual results. They followed two methods; firstly, they used a Bayesian random-effects model to predict Daily NBA player performance and generate a team baseline based on the game's rules having a specific salary cap and a constraint on the number of players who play in the same position. Secondly, they develop a K-nearest neighbors model using the results from the previous Bayesian model to identify the "successful" lineups. Both methods successfully generate profit in a hypothetical experiment for the season 2015-16, with the KNN approach generating a more significant profit than the Bayesian alone.

In addition, in the Fall of 2020, Connor Young, Andrew Koo, and Saloni Gandhi [22] try to develop a system that predicts the best combination of players daily that their overall fantasy score will be the better as possible with a constraint of overall fantasy cap of 50.000$. In other words, they tried to predict the most efficient lineup in terms of fantasy value per salary unit. They developed two models, Random Forest and Regularized Gradient Boosted Trees (XGBoost), with NBA player and team data from 2014 to 2020. With feature engineering, they experiment with over 70 features and evaluate the importance of these. Finally, their most accurate model was XGBoost, and as they refer performed better by 8.58% than the published projections of the largest betting company in the Fantasy sector (DraftKings).

In 2015, Eric Hermann and Adebia Ntoso [23] attempted to apply ML to Fantasy Basketball to predict daily Basketball players' fantasy scores successfully and generate an eight-person team. They started by framework the problem and scraped box score and team data from season 2014-15 and 2015-16 to train their models. The lineup they tried to predict, was composed of eight players with constraints by the player's position. Also, by selecting a player, they should give him a salary, and they set an overall cap of 50.000$ in total for the eight players. Their study splits into two parts. The first part, to predict players' performance from historical data, which is a regression problem. They used a linear regression model for the first part and achieved an error of 7.5% less than DraftKings (DFS Company). The second part was about choosing a team based on the predicted points of every player that had a game that he participated in the specific night, and to construct their daily baselines, they used a multinomial Naive Bayes Classifier and Beam Search to accelerate the running time.

In the Spring of 2019, James Earl [24] focused his study on a FanDuel (DFS Company) tournament called 50/50s, and his goal was to select the best daily NBA performance lineup. In this tournament, someone had to rank in the top half of the users who compete to have profit. He used a dataset from the 2017-2018 NBA season and included historical data for every player who competed in the League. The researcher implemented ML techniques using the R programming language and specifically the Caret package. Furthermore, he implemented classification methods creating a formula (players' predicted fantasy points divided by his adjusted FanDuel salary produced a value). The classification targeted predicting if a player had a high (a value above 0.5) or low (a value of 0.0 to 0.5) fantasy performance. Continuously, the researcher tried many classification methods with Support Vector Machine (SVM) model performed better than the others succeeding 60% of accuracy. However, because the price of a player is not constant and companies adjust it based on their predictions, the model results were not trustworthy. For this reason, instead of SVM, the linear discriminate analysis was preferred, providing the most useful predictions despite being less accurate than others achieving 59% accuracy.

3 Main Research Topics

In this part of the Dissertation, general terms relative to the topic will be analyzed.

- Data Mining
- Machine Learning
- Sport Analytics
    - The Use of Sport Analytics
    - Basketball Fantasy

3.1 Data Mining

Data Mining is the process that achieves to extract knowledge from a large amount of data using ML, Statistics, and Database Systems. DM is a subfield of Computer Science and evolves statistics trying to discover patterns, trends and find meaning where the human eye cannot [25].

The implementation of DM tasks in databases is a procedure that requires:

- Data Selection

  Data need to be extracted from sources, collected, and stored in Data Warehouses.

- Data Understanding

  Before continuing, data should be understandable from the user

- Pre-processing

  Dataset usually is not in the appropriate form for optimized DM tasks. For this reason, cleaning should be done.

- Transformation

  Datasets that contain "Noise" or missing variables should be processed and transformed in a form capable of extracting knowledge.

- Data Mining

  Common DM tasks are:

    - Anomaly detection; (The detection of an anomaly on data, unusual data records, or data errors).
    - Association rule learning; (Finding relationships between variables).
    - Clustering: (Constructing groups from the data that in some way are related to each other).
    - Classification: (Classify the data to already known structures).

  - Regression: (The attempting to find a function that fits the data with the least possible error for extracting data relations).
  - Summarization; (Data representation, like visualizations or written reports).
- Evaluation

  The results after the DM tasks should be evaluated for their significance in extracting knowledge.

- Visualization

  The visualization of the results from a DM task is essential for a better understanding of them [26].

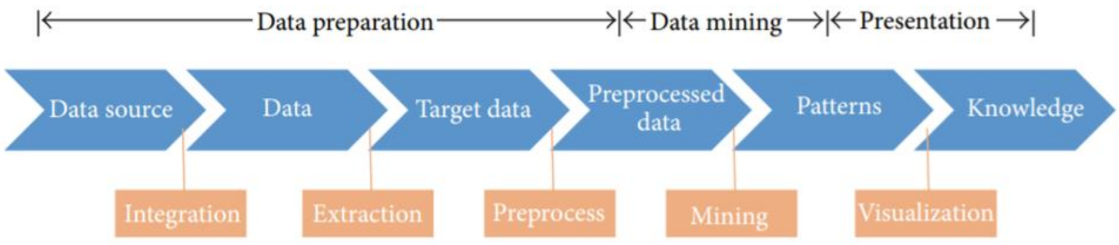


Figure 1: Data Mining Process

3.2 Machine Learning

Machine Learning is a kind of Artificial Intelligence in which applications can generate accurate prediction outcomes without being specially programmed for specific occasions. It is also a subfield of Computer Science, and its uses are to build models based on algorithms, deploy them on data and make predictions or decisions. ML has many applications. Some of them are; text processing, natural language processing, and speech recognition [27].

ML splits into three major types:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

3.2.1 Supervised Learning

Supervised Learning is a type of ML in which the algorithm is trained on accurately labeled data. The ML algorithm works by giving a portion of the data to be trained, processing the problem, the solution, and the kind of data. Train, test, and final datasets are often similar to each other based on their characteristics. After training, the algorithm has found relationships in given data between the input and output. Lastly, the trained model is deployed in the final dataset and attempts to project the class labels for inconspicuous cases correctly. Supervised Learning is separated into two types of problems, Classification, and Regression [28][29].

- Classification methods
    - Linear Classifiers
    - Naive Bayes
    - Support Vector Machines (SVM)
    - Decision Trees
    - K-Nearest Neighbor
    - Random Forest
    - Neural Networks

- Regression methods
    - Linear Regression
    - Logistic Regression
    - Polynomial Regression
    - Neural Networks

3.2.2 Unsupervised Learning

Unsupervised Learning, in contrast with Supervised Learning, can work with unlabeled data as input and output. For this reason, human intervention is not required to make the data understandable to the machine. Because Unsupervised Learning has no labels to process, it creates hidden structures, finding similarities and differences in data [30].

Common Unsupervised Learning approaches:

- Clustering
    - K-Means clustering (Exclusive and Overlapping Clustering)
    - Ward’s linkage (Hierarchical Clustering)
    - Average linkage (Hierarchical Clustering)
    - Complete (or maximum) linkage (Hierarchical Clustering)
    - Single (or minimum) linkage (Hierarchical Clustering)
    - Gaussian Mixture Models (Probabilistic clustering)
- Association Rules
- Apriori Algorithms
- Dimensionality Reduction
- Principal Component Analysis (PCA)
- Singular Value Decomposition
- Autoencoders

3.2.3 Reinforcement Learning

Reinforcement Learning is similar to Supervised Learning; however, the model is not trained using a portion of data but by trial and error—the algorithm works with an interpreter and a reward system. In every loop of the algorithm, the outputs are evaluated by the interpreter as favorable or not. The interpreter reinforces the solution if the outputs are correct; if not, the algorithm is reiterated until a better result emerges [31].

Some applications of Reinforcement Learning can be applied in the following fields:

- Resources management in computer clusters
- Traffic Light Control
- Robotics
- Web System Configuration
- Chemistry
- Personalized Recommendations
- Bidding and Advertising
- Games
- Deep Learning

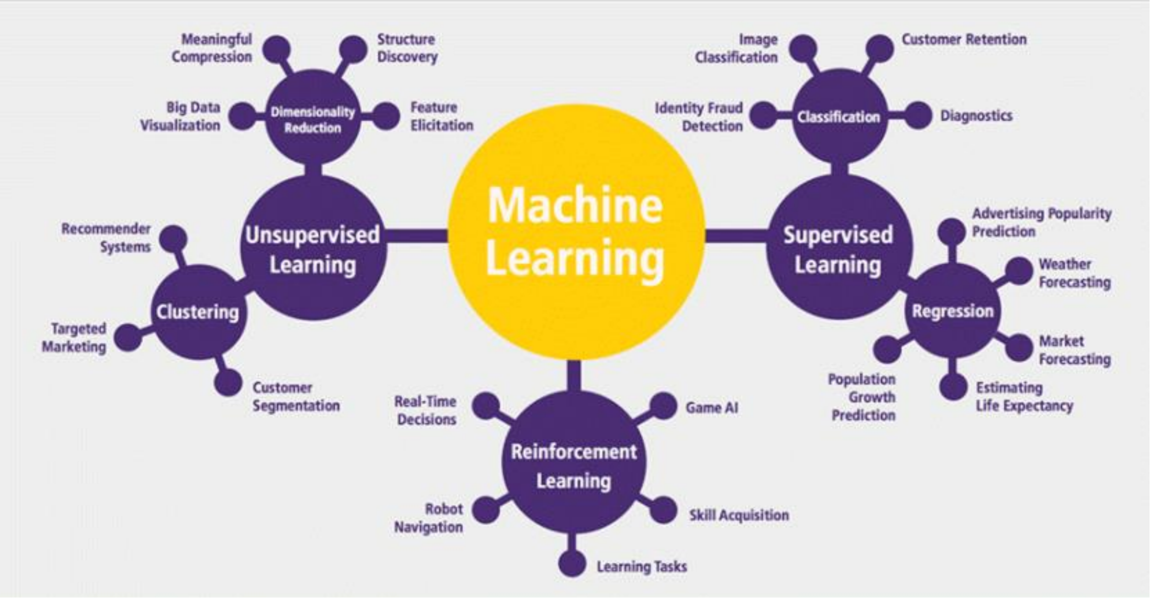


Figure 2: Machine Learning Structure and Applications [27].

3.3 Sports Analytics

Sports Analytics is a rapidly evolving topic because it provides advanced decision-making and improvement information in competitive professional sports. Sports Analytics can offer tremendous competitive advantages to a team using AI technology and statistical analysis on valid data.

3.3.1 The Use of Sports Analytics

Sports Analytics is valuable in improving players and team performance, while it can provide information and find patterns that an experienced coach can not for any individual player performance and team tactics and strategies. One more major category of Sports Analytics is the organization’s business performance, for example, advance analytics for ticket pricing, social media appearance, promotions, and generally the interaction between fans and teams-organizations. In addition, Sports analytics can be used for the analysis of player heal and injuries. Furthermore, with Sports Analytics, it is possible to predict if a player is vulnerable to injuries and must have time off or a therapy to be planned [32].

3.3.2 Basketball (NBA) Fantasy

The subject of Basketball Fantasy is to allow the fans to compete with each other, constructing their team from all players that participate in NBA League. Many companies in the betting industry offer Fantasy games with prizes in the first place. For Ranking the competitors, a metric called "FP" (Fantasy Points) is used, representing each athlete's performance in the actual game.

Fantasy Points Formula [55] :

$$FP = P + 1.2 \times REB + 1.5 \times AST + 3 \times STL + 3 \times BLK - TOV$$

Glossary:

Point = Each point scored

REB = Rebound

AST = Assist

STL = Steal

BLK = Block

TOV = Turnover

However, some constraints on the team build-up exist. In most cases, one of these constraints is the fictional salary cap, while every player has a salary and the budget for building the team is fixed. In addition, one more constraint is the fundamental role of the player, while the user has to pick players with different roles in NBA, mirroring a real team roster. The constructed team must have eight players, one each of the traditional five roles (Point Guard, Shooting Guard, Small Forward, Power Forward, Center), one additional guard, one forward, and one of any of the five positions above. Also, the selected players have to be from at least two NBA teams, and they must be selected from teams that have to play an actual match on the specific day of interest [21][24].

The most often types of games are the "50-50" and the Tournament. The "50-50" is a game in which to make a profit you have to be ranked in the top half places, while on the tournament mode, only the top 20% of the competitors make a profit. The Rank list of the competitors is contacted with each competitor's team Fantasy Points sum, which is an addition of each selected player's Fantasy Points from the match on the specific day of interest. Lastly, the sum of the Fantasy Points of the selected team is computed for each competitor who entered the same game, and the winner is acclaimed [33].

4 Methodology

This Dissertation chapter will analyze the methodology for predicting each player’s Fantasy Points for each game match chosen. Also, the process for Baseline Optimizer Lineup building up using Fantasy Points prediction results will be examined. Starting with Data Engineering, how data are scraped from a valid source, and continue with how data are cleaned and transformed. Finishing with Feature Engineering, where time-lagged features are created.

4.1 Process Description

First of all, the appropriate dataset should be constructed. As valid data needed, the source we used to scrape them was the NBA's official website (nba.com). Basketball is a sport with plenty of statistics. For this reason, several datasets with different types of statistics are needed. The data used is Player's Box Scores statistics and Team's Box Scores statistics.

After data were possessed, the necessary pre-processing was needed. While we had to clean and transform each dataset from unnecessary statistics that gave us no further information about the performance of the player or the team, each type of dataset for players and teams had to be merged to proceed to predictions related to NBA Fantasy Points for each player.

The next step was to pre-process the data, testing for null values, duplicates, and noise. Also, the transformation of each dataset was necessary for merging Players and Teams data into one single dataset. The next most important phase was feature engineering and extraction because each dataset row should have historical data.

In the prediction-making phase, to optimize our results and conduct better predictions, the dataset was split into several smaller datasets per player, and one model for each player was built and selected. The results of the conducted predictions were evaluated in terms of Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

The mean absolute percentage error (MAPE) is a measure of a forecast system's accuracy. It expresses this accuracy as a percentage, calculated as the mean or average of forecast absolute percentage errors. The error is defined as the observed value minus actual values divided by actual values for each time period. The following formula defines MAPE:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

where A_t the actual value and F_t the forecasted value

The MAPE is one of the most common evaluation metrics to determine the quality of a regression model. However, the data should not contain outliers and zeros in order to use MAPE [38].

Mean Absolute Error (MAE) is a model assessment metric that is commonly employed with regression models. A model's mean absolute error with regard to a test set is the average of the absolute values of the individual prediction errors on all instances in the test set. Each prediction error is the difference between the instance's true and predicted values. The following formula defines MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y}_i|$$

where $\tilde{y}_i$ is the expected value of the i-th sample and y_i is the actual value [35].

Finally, we built up a Daily Fantasy Lineup Optimizer with restrictions to select the best eight-player lineup for an NBA Fantasy Tournament for one day. To proceed with the Optimizer, an additional dataset related to Fantasy Salaries and Positions of the players needed and downloaded from the DraftKings website (draftkings.com) offers this kind of Tournaments. The Daily Fantasy Lineup Optimizer results were evaluated by the sum of the actual results of the Fantasy points of each player.

It is worth noting that, because of the large amount of data, the environment used for all the processes was the Google Collaboratory with RAM power(51GB). Also, Python language was utilized for download, pre-processing, feature engineering, building up ML models, predictions, Daily Fantasy Lineup Optimizer phases.

4.1.1 Data Collection

There are plenty of websites that offer plenty of NBA historical data. However, these historical data must be valid. For this reason, the official NBA website “nba.com” is selected to scrape our data. Our research focuses on data from seasons “2011-12” to “2020-21”. Different types of historical data were downloaded using Python and the Request library, specifying the parameters. These parameters were :

- URL (“playergamelogs” / ” teamgamelogs”)
- Season Year (“2011 - 12” to “2020 - 21”)
- Season Type (“RegularSeason” / “Playoffs”)
- Measure Type
    - Player datasets
        - Base (68 features)
        - Advanced (82 features)
        - Misc (46 features)
        - Scoring (56 features)
        - Usage (56 features)

    - Team datasets
        - Base (57 features)
        - Advanced (49 features)
        - Misc (31 features)
        - Scoring (45 features)
        - Four Factors (31 features)
        - Opponent (57 features)

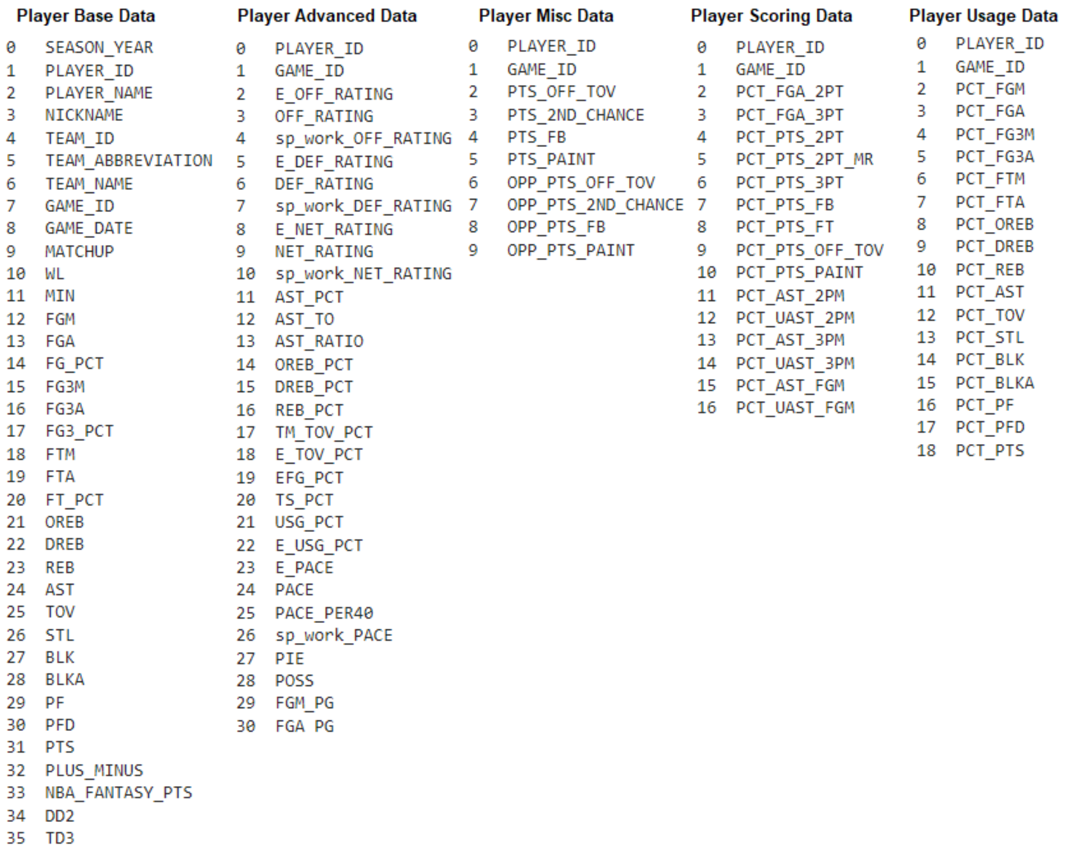
Each Player dataset contained Box Score statistics for each player for each game played for ten Seasons. Also, each Team dataset contained Box Score statistics for each team for each game played for ten Seasons. One hundred Player datasets were scraped for each type of data, ten per type (one for each season), fifty for the Regular Season, and fifty for Playoffs. In addition, One hundred and twenty Team datasets were scraped for each type of data, ten per type (one for each season), sixty for the Regular Season, and sixty for Playoffs. Worth noting that each dataset is stored as an excel file type to keep the appropriate format.

After collecting the data, various actions needed to clean and transform and finally take place to the ML models’ variables and features.

4.1.2 Pre-Processing

The pre-processing phase started with merging datasets and cleaning the data. Firstly, for Player datasets, every year's dataset was merged for each type of dataset (Base, Advanced, Misc, Scoring, Usage) for the Regular Season and Playoffs. To procedure, these datasets contained useless columns and rank statistics in ever. These rank statistics were related to the final ranks for every player or team at the end of the season, and for this reason, they were removed.

In Player Datasets, the columns "SEASON_YEAR", "PLAYER_NAME", "NICKNAME","TEAM_NAME","NICKNAME","TEAM_ID", "TEAM_ABBREVIATION", "GAME_DATE", "MATCHUP" and "WL" dropped in all tables except one. For proceeding with merging, we kept "PLAYER_ID" and "GAME_ID" to merge on.



Player Base Data

```
0   SEASON_YEAR
1   PLAYER_ID
2   PLAYER_NAME
3   NICKNAME
4   TEAM_ID
5   TEAM_ABBREVIATION
6   TEAM_NAME
7   GAME_ID
8   GAME_DATE
9   MATCHUP
10  WL
11  MIN
12  FGM
13  FGA
14  FG_PCT
15  FG3M
16  FG3A
17  FG3_PCT
18  FTM
19  FTA
20  FT_PCT
21  OREB
22  DREB
23  REB
24  AST
25  TOV
26  STL
27  BLK
28  BLKA
29  PF
30  PFD
31  PTS
32  PLUS_MINUS
33  NBA_FANTASY_PTS
34  DD2
35  TD3
```

Player Advanced Data

```
0   PLAYER_ID
1   GAME_ID
2   E_OFF_RATING
3   OFF_RATING
4   sp_work_OFF_RATING
5   E_DEF_RATING
6   DEF_RATING
7   sp_work_DEF_RATING
8   E_NET_RATING
9   NET_RATING
10  sp_work_NET_RATING
11  AST_PCT
12  AST_TO
13  AST_RATIO
14  OREB_PCT
15  DREB_PCT
16  REB_PCT
17  TM_TOV_PCT
18  E_TOV_PCT
19  EFG_PCT
20  TS_PCT
21  USG_PCT
22  E_USG_PCT
23  E_PACE
24  PACE
25  PACE_PER40
26  sp_work_PACE
27  PIE
28  POSS
29  FGM_PG
30  FGA_PG
```

Player Misc Data

```
0   PLAYER_ID
1   GAME_ID
2   PTS_OFF_TOV
3   PTS_2ND_CHANCE
4   PTS_FB
5   PTS_PAINT
6   OPP_PTS_OFF_TOV
7   OPP_PTS_2ND_CHANCE
8   OPP_PTS_FB
9   OPP_PTS_PAINT
```

Player Scoring Data

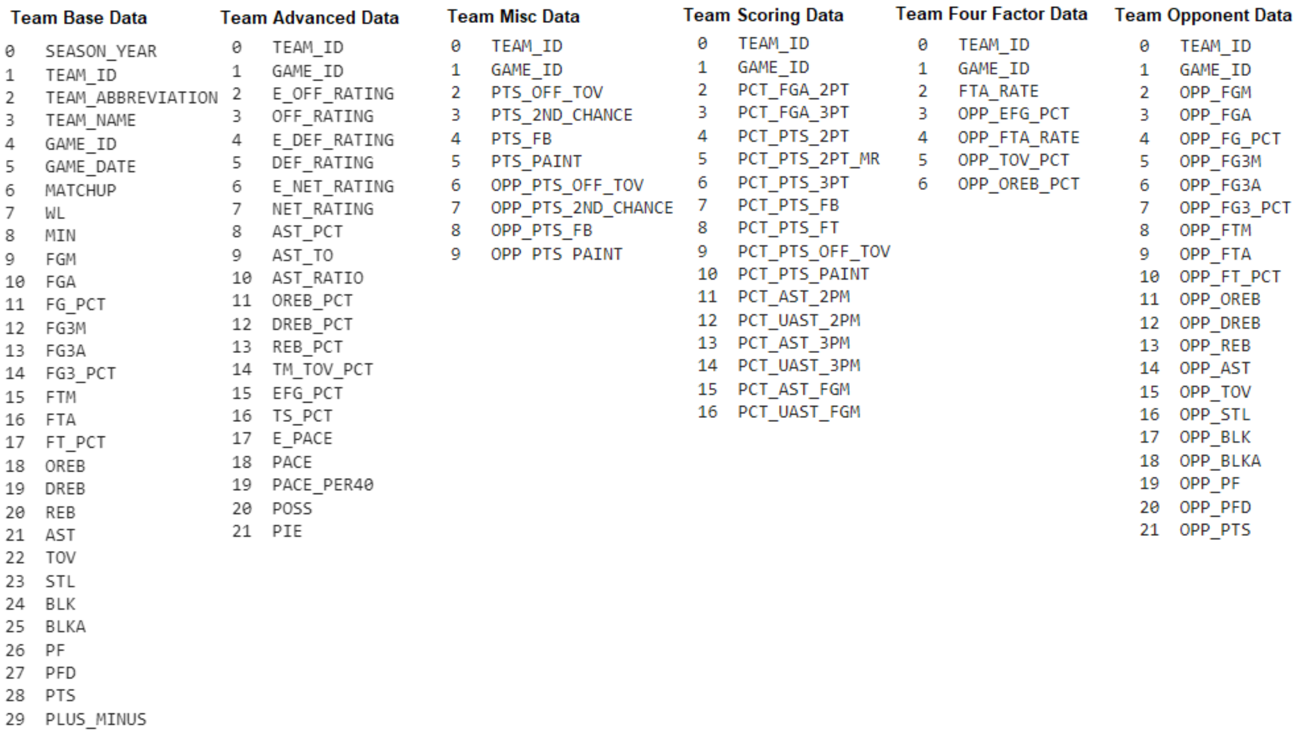
```
0   PLAYER_ID
1   GAME_ID
2   PCT_FGA_2PT
3   PCT_FGA_3PT
4   PCT_PTS_2PT
5   PCT_PTS_2PT_MR
6   PCT_PTS_3PT
7   PCT_PTS_FB
8   PCT_PTS_FT
9   PCT_PTS_OFF_TOV
10  PCT_PTS_PAINT
11  PCT_AST_2PM
12  PCT_UAST_2PM
13  PCT_AST_3PM
14  PCT_UAST_3PM
15  PCT_AST_FGM
16  PCT_UAST_FGM
```

Player Usage Data

```
0   PLAYER_ID
1   GAME_ID
2   PCT_FGM
3   PCT_FGA
4   PCT_FG3M
5   PCT_FG3A
6   PCT_FTM
7   PCT_FTA
8   PCT_OREB
9   PCT_DREB
10  PCT_REB
11  PCT_AST
12  PCT_TOV
13  PCT_STL
14  PCT_BLK
15  PCT_BLKA
16  PCT_PF
17  PCT_PFD
18  PCT_PTS
```

Table 1: Features of Datasets related to Player Box Scores

In Team datasets, the columns “SEASON_YEAR”, “TEAM_ABBREVIATION”, “TEAM_NAME”, “WL”, and “MATCHUP” dropped in all tables except for one. To proceed with merging, we kept “TEAM_ID” and “GAME_ID” to merge on.



Team Base Data	Team Advanced Data	Team Misc Data	Team Scoring Data	Team Four Factor Data	Team Opponent Data
0 SEASON_YEAR	0 TEAM_ID	0 TEAM_ID	0 TEAM_ID	0 TEAM_ID	0 TEAM_ID
1 TEAM_ID	1 GAME_ID	1 GAME_ID	1 GAME_ID	1 GAME_ID	1 GAME_ID
2 TEAM_ABBREVIATION	2 E_OFF_RATING	2 PTS_OFF_TOV	2 PCT_FGA_2PT	2 FTA_RATE	2 OPP_FGM
3 TEAM_NAME	3 OFF_RATING	3 PTS_2ND_CHANCE	3 PCT_FGA_3PT	3 OPP_EFG_PCT	3 OPP_FGA
4 GAME_ID	4 E_DEF_RATING	4 PTS_FB	4 PCT_PTS_2PT	4 OPP_FTA_RATE	4 OPP_FG_PCT
5 GAME_DATE	5 DEF_RATING	5 PTS_PAINT	5 PCT_PTS_2PT_MR	5 OPP_TOV_PCT	5 OPP_FG3M
6 MATCHUP	6 E_NET_RATING	6 OPP_PTS_OFF_TOV	6 PCT_PTS_3PT	6 OPP_OREB_PCT	6 OPP_FG3A
7 WL	7 NET_RATING	7 OPP_PTS_2ND_CHANCE	7 PCT_PTS_FB		7 OPP_FG3_PCT
8 MIN	8 AST_PCT	8 OPP_PTS_FB	8 PCT_PTS_FT		8 OPP_FTM
9 FGM	9 AST_TO	9 OPP_PTS_PAINT	9 PCT_PTS_OFF_TOV		9 OPP_FTA
10 FGA	10 AST_RATIO		10 PCT_PTS_PAINT		10 OPP_FT_PCT
11 FG_PCT	11 OREB_PCT		11 PCT_AST_2PM		11 OPP_OREB
12 FG3M	12 DREB_PCT		12 PCT_UAST_2PM		12 OPP_DREB
13 FG3A	13 REB_PCT		13 PCT_AST_3PM		13 OPP_REB
14 FG3_PCT	14 TM_TOV_PCT		14 PCT_UAST_3PM		14 OPP_AST
15 FTM	15 EFG_PCT		15 PCT_AST_FGM		15 OPP_TOV
16 FTA	16 TS_PCT		16 PCT_UAST_FGM		16 OPP_STL
17 FT_PCT	17 E_PACE				17 OPP_BLK
18 OREB	18 PACE				18 OPP_BLKA
19 DREB	19 PACE_PER40				19 OPP_PF
20 REB	20 POSS				20 OPP_PFD
21 AST	21 PIE				21 OPP_PTS
22 TOV					
23 STL					
24 BLK					
25 BLKA					
26 PF					
27 PFD					
28 PTS					
29 PLUS_MINUS					

Table 2: Features of Datasets related to Team Box Scores.

Continuously, one more column was created, named “PLAYOFFS” in all datasets. In Regular Season datasets, we set the value in all rows as “0”, while in Playoff datasets, we set the value in all rows as “1”.

Regular Season and Playoff datasets are merged on dataset type (Base, Advanced, Misc, Scoring Usage) in the next phase. The precisely same procedure is followed for Teams’ datasets. Finally, five datasets related to player data occurred (Base, Advanced, Misc, Scoring, Usage) and six datasets related to team data occurred (Base, Advanced, Misc, Scoring, Four Factors, Opponent). These datasets contained all Season years for Regular and Playoff Season.

Finally, datasets related to Player statistics (Base, Advanced, Misc, Scoring, Usage) are merged on “PLAYER_ID”, “GAME_ID”. Also, datasets related to Team statistics (Base, Advanced, Misc, Scoring, Four Factor, Opponent) are merged on “TEAM_ID”, “GAME_ID”.

Before merging Player and Team datasets, the final step was to remove players who do not participate in the NBA anymore. For this reason, we list the Active IDs on Season 2020-21, and based on this list; we exclude every player who is not included in this list from datasets.

```
df2021 = data.loc[data['SEASON_YEAR'] == '2020-21']
Active_IDs = df2021['PLAYER_NAME'].copy()
Active_IDs = Active_IDs.drop_duplicates()
Active_IDs = Active_IDs.to_list()
len(Active_IDs)

data = data[~data['PLAYER_NAME'].isin(Active_IDs) == False]
```

Figure 3: Code for clearing non Active NBA Players.

The final two Datasets had the following shape.

Player Data	(145415, 106)
Team Data	(25148, 100)

Table 3: Shape of two Datasets Before Feature Engineering.

4.1.3 Feature Engineering

The tables were already very informative; however, we needed to transform the datasets in a form that its row contained past statistics and add some extra features to optimize our results.

Before continuing in-depth with the Feature extraction and engineering, we will present some functions used to create our past statistics.

These functions are,

```
def get_previous_momentums(dataframe, columns)
```

In which, we calculate the value of the difference of row^n with row^{n-1}.

```
def get_previous_day(dataframe, columns)
```

Generating the value of the row^{n-1}.

```
def get_previous_xdays_avg(dataframe, days, columns)
```

Generating the mean of the number of days we select.

```
def get_previous_xdays_sum(dataframe, days, columns)
```

Generating the sum of the number of days we select.

```
def rest_days(dataframe, date_column)
```

Generating the difference of row^n with row^{n-1} in “GAME_DATE”, calculating how many rest days the player had before his last match.

```
# Previous Days Values
def get_previous_momentums(dataframe, columns):
   for column in columns:
      dataframe[column + '_MOMENTUM']=dataframe[column].diff()

def get_previous_day(dataframe, columns):
   rows = len(dataframe.index)-1
   for column in columns:
      dataframe['LAST_MATCH_' + column]=np.nan
      dataframe['LAST_MATCH_' + column][1:] = dataframe[column][0:rows]

def get_previous_xdays_avg(dataframe, days, columns):
   for column in columns:
      dataframe['LAST_MATCHES_' + str(days) + '_DAYS_' + column +
'_AVG']=dataframe[column].shift(1).rolling(window=days).mean()

def get_previous_xdays_sum(dataframe, days, columns):
   for column in columns:
      dataframe['LAST_MATCHES_' + str(days) + '_DAYS_' + column +
'_SUM']=dataframe[column].shift(1).rolling(window=days).sum()

def get_time_lagged_features(dataframe, columns):
   get_previous_day(dataframe, columns)
   get_previous_momentums(dataframe, dataframe.drop(columns =
['GAME_DATE']).columns.to_list())

   for days in [3,5,7]:
      get_previous_xdays_avg(dataframe, days, columns)
      get_previous_xdays_sum(dataframe, days, columns)
   dataframe.dropna(inplace=True)

#Create Rest_Days
def rest_days(dataframe, date_column):
    dataframe[date_column]= pd.to_datetime(dataframe[date_column])
    dataframe['REST_DAYS'] = dataframe[date_column].diff()
    dataframe.dropna(inplace=True)
    dataframe['REST_DAYS'] = (dataframe['REST_DAYS']).astype(str)
    dataframe['REST_DAYS'] = dataframe['REST_DAYS'].str[:-5]
    dataframe['REST_DAYS'] = (dataframe['REST_DAYS']).astype(int)
    dataframe.loc[dataframe['REST_DAYS'] >= 5, 'REST_DAYS'] = 5
```

Figure 4: Code of functions for Feature Engineering.

First of all, we had to rename the columns of Team dataset to differ with Players' dataset, however we had to keep same name on columns on which we would merge the tables on (for example, "GAME_DATE", "GAME_ID"). In addition, a new column was created "H/A" based on "MATCHUP" column in which value is "0" when Team//Player participates on Home, and "1" on Away. Before merging the datasets, we had to create Opponent last match statistics. For this reason,we created dictionaries with Key: TEAM_NAME and value the table of each team. After sorting these datasets by "GAME_DATE" and using the function `get_previous_day` on Team dataset we created the Opponent statistics and finally added them on Team dataset.

```
#Filter only columns that we are interested in for gain value
df4 = df2.copy()
df4 = df4.filter(items=['GAME_DATE','TEAM_ABBREVIATION',
'TEAM_OFF_RATING','TEAM_DEF_RATING','TEAM_NET_RATING','TEAM_NBA_FANTASY_PTS'])
df4.sort_values(by = 'TEAM_ABBREVIATION', inplace = True)

#Rename them to take Opponent Data on each game
dict2 = {'TEAM_OFF_RATING' : 'OPP_TEAM_OFF_RATING',
  'TEAM_DEF_RATING' : 'OPP_TEAM_DEF_RATING',
  'TEAM_NET_RATING' : 'OPP_TEAM_NET_RATING',
  'TEAM_NBA_FANTASY_PTS' : 'OPP_TEAM_NBA_FANTASY_PTS'}
df4.rename(columns=dict2,
          inplace=True)
tnames=df4['TEAM_ABBREVIATION'].unique().tolist()

#Get Team's Last game Data
def get_previous_day(dataframe, columns):
   rows = len(dataframe.index)-1
   for column in columns:
      dataframe['LAST_MATCH_' + column]=np.nan
      dataframe['LAST_MATCH_' + column][1:] = dataframe[column][0:rows]

#Dictionaries of dataframes for Team to get Lasts match opp points / Split Dataset
to each Team
dfsteam = {}
groups = df4.groupby(df4.TEAM_ABBREVIATION)
columns =
['OPP_TEAM_OFF_RATING','OPP_TEAM_DEF_RATING','OPP_TEAM_NET_RATING','OPP_TEAM_NBA_FAN
TASY_PTS']
dfopp = pd.DataFrame()
for i in tnames:
  dfsteam[i] = groups.get_group(i)
  dfsteam[i].sort_values(by = 'GAME_DATE', inplace = True)
  get_previous_day(dfsteam[i], columns)
  dfsteam[i] = dfsteam[i].filter(items=['GAME_DATE','TEAM_ABBREVIATION',
'LAST_MATCH_OPP_TEAM_OFF_RATING','LAST_MATCH_OPP_TEAM_DEF_RATING',
'LAST_MATCH_OPP_TEAM_NET_RATING','LAST_MATCH_OPP_TEAM_NBA_FANTASY_PTS'])
  dfsteam[i].rename(columns = {'TEAM_ABBREVIATION' : 'OPPONENT'},inplace = True)

#Create a new Dataset with Opponent data
for i in tnames:
  dfopp = dfopp.append(dfsteam[i], ignore_index=True)

#Merge Player Data with Team Data
df_merged = pd.merge(df, df2, on= (df2.columns).intersection(df.columns).to_list(),
how='inner')

#Get column Opponent
df_merged['OPPONENT'] = df_merged['MATCHUP'].str[-3:]
#Drop old, incorrect Opponent Data
df_merged.drop(columns = ['TEAM_OPP_EFG_PCT'  ,'TEAM_OPP_FTA_RATE',
'TEAM_OPP_TOV_PCT', 'TEAM_OPP_OREB_PCT',  'TEAM_OPP_FGM', 'TEAM_OPP_FGA',
               'TEAM_OPP_FG_PCT'  ,'TEAM_OPP_FG3M'  ,'TEAM_OPP_FG3A',
'TEAM_OPP_FG3_PCT', 'TEAM_OPP_FTM',
               'TEAM_OPP_FTA',  'TEAM_OPP_FT_PCT',  'TEAM_OPP_OREB',
'TEAM_OPP_DREB',  'TEAM_OPP_REB', 'TEAM_OPP_AST',
               'TEAM_OPP_TOV',  'TEAM_OPP_STL', 'TEAM_OPP_BLK','TEAM_OPP_BLKA'
,'TEAM_OPP_PF', 'TEAM_OPP_PFD', 'TEAM_OPP_PTS'], inplace=True, axis=1)
```

Figure 5: Code related to Last match Opponent features.

While our goal was to predict each players' performance as best as possible based on his past statistics, our purpose was to create two datasets. The first dataset would contain all available statistics and the second one only with the basic statistics that construct the value of Fantasy Points that we wanted to predict. For this reason, two different procedures were followed. The following procedures are done using dictionaries, as Key: Player name and as value the table of data of each player.

Before proceeding to Feature engineering, we had to exclude rookies and players who underperform in Season 2020-21. For this reason, players that have less than 100 appearances from season 2017-18 to season 2019-20. Also, we exclude the players who had less than 30 appearances in season 2020-21 and had less than 18 minutes mean participation time in season 2020-21. Also, we exclude players who perform poorly on Season 2020-21 in contrast with the rest having FP means less than half of the total FP mean of all players. Furthermore, we encoded all categorical features except of "SEASON_YEAR".

With the Pycaret library, we perform Anomaly Detection on Fantasy points on both of our datasets. We found if there is an Anomaly on Fantasy Points by Standard Deviation (setting the boundary on each players' standard deviation - 2) for each players' dataset for four months. In this way, we created two new features, Smoothed FP, which smooths the value of Fantasy Points. and Anomaly, which took value "1" if Anomaly detected and "0" if there was no Anomaly detected.

After Anomaly detection, we had to create new columns with historical information for each row. For the first dataset with all features, using the functions from above, we created momentum columns for all informative columns, anomaly columns included. However, we could not keep these columns because of the data leak appearance. Motivated by this, we created new columns with last match statistics, last three matches sum statistics, and the last match sums statistics. Furthermore, because our resource focuses on NBA Fantasy Points, Instead of only "LAST_MATCH_NBA_FANTASY_PTS" column, we also created four additional features, containing last three, last five, last seven, and last ten matches averages of Fantasy Points.

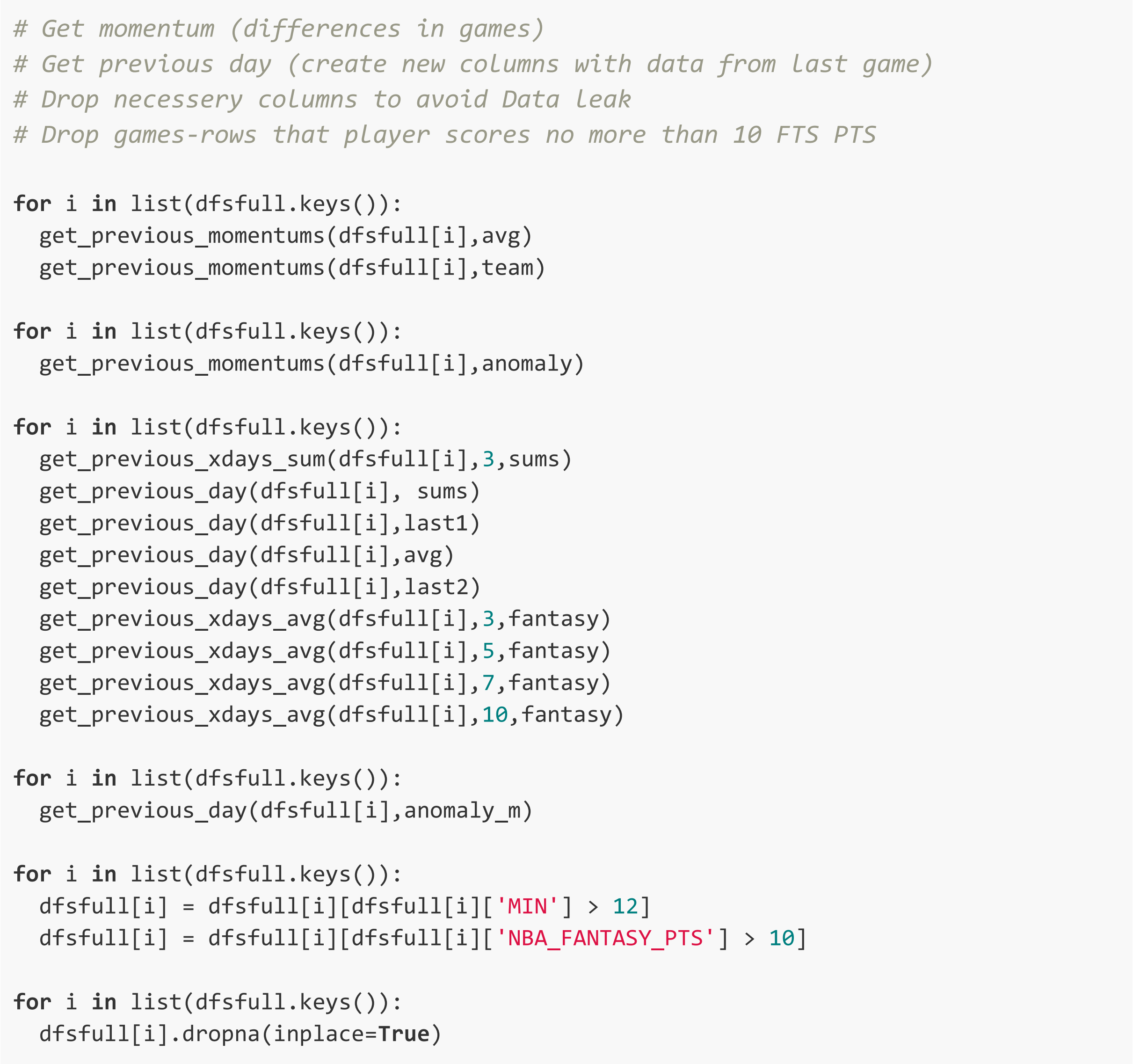
As well, we exclude the appearances of each player that underperformed (FP <= 10) because of injury or played less than one period (MIN <= 12) in each match. The final dataset contained 203 of 540 players. Finally, we dropped all "present" columns related to each game to avoid data leaks and keep only historical data for each match.

The same procedure is followed for the second dataset also. However, we exclude last game sums, and of course, only primary and opponent data were included.

The final output of our two datasets is shown in Table 4,

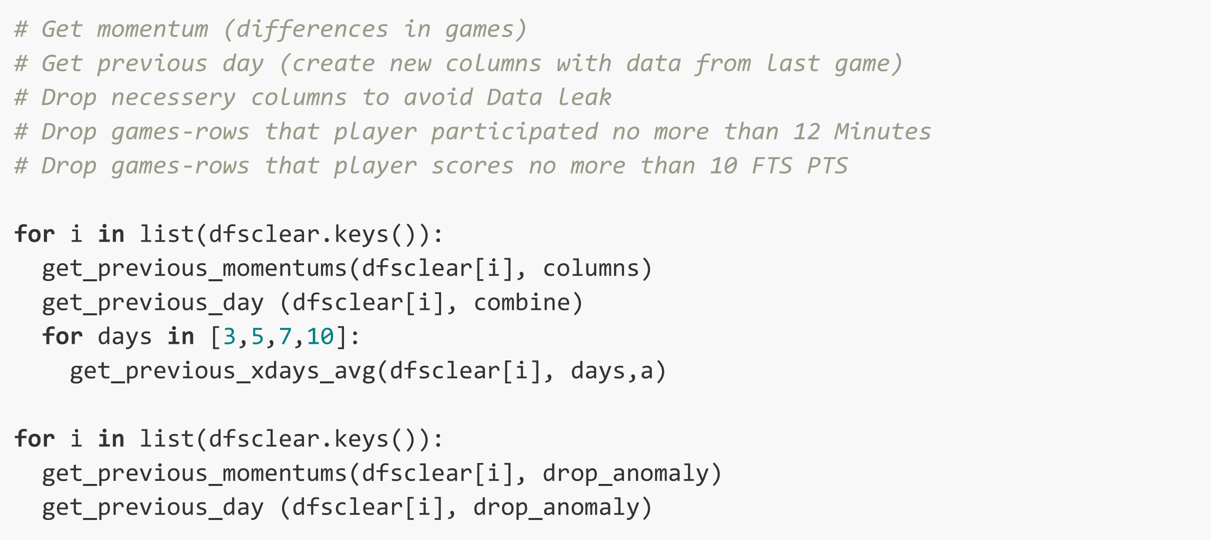
Advanced Features Dataset	(79036, 269)
Basic Features Dataset	(79036, 67)

Table 4: Shape of datasets after Feature Engineering.

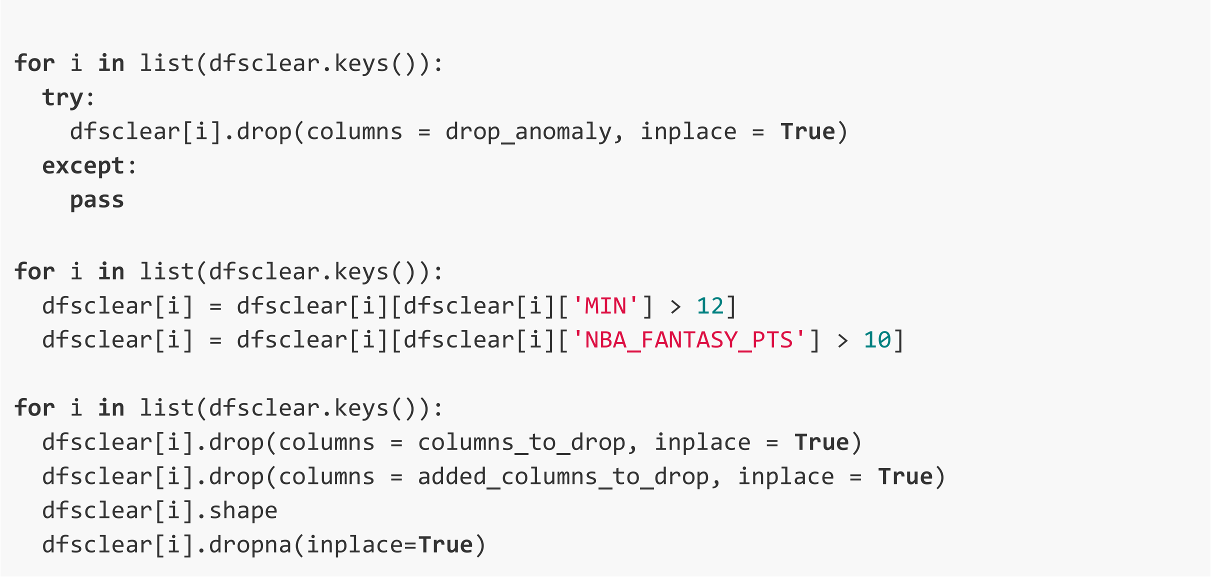


```
# Get momentum (differences in games)
# Get previous day (create new columns with data from last game)
# Drop necessery columns to avoid Data leak
# Drop games-rows that player scores no more than 10 FTS PTS

for i in list(dfsfull.keys()):
  get_previous_momentums(dfsfull[i],avg)
  get_previous_momentums(dfsfull[i],team)

for i in list(dfsfull.keys()):
  get_previous_momentums(dfsfull[i],anomaly)

for i in list(dfsfull.keys()):
  get_previous_xdays_sum(dfsfull[i],3,sums)
  get_previous_day(dfsfull[i], sums)
  get_previous_day(dfsfull[i],last1)
  get_previous_day(dfsfull[i],avg)
  get_previous_day(dfsfull[i],last2)
  get_previous_xdays_avg(dfsfull[i],3,fantasy)
  get_previous_xdays_avg(dfsfull[i],5,fantasy)
  get_previous_xdays_avg(dfsfull[i],7,fantasy)
  get_previous_xdays_avg(dfsfull[i],10,fantasy)

for i in list(dfsfull.keys()):
  get_previous_day(dfsfull[i],anomaly_m)

for i in list(dfsfull.keys()):
  dfsfull[i] = dfsfull[i][dfsfull[i]['MIN'] > 12]
  dfsfull[i] = dfsfull[i][dfsfull[i]['NBA_FANTASY_PTS'] > 10]

for i in list(dfsfull.keys()):
  dfsfull[i].dropna(inplace=True)
```

Figure 6: Code for Feature Engineering implementation in Advanced Features Datasets.



```
# Get momentum (differences in games)
# Get previous day (create new columns with data from last game)
# Drop necessery columns to avoid Data leak
# Drop games-rows that player participated no more than 12 Minutes
# Drop games-rows that player scores no more than 10 FTS PTS

for i in list(dfsclear.keys()):
  get_previous_momentums(dfsclear[i], columns)
  get_previous_day (dfsclear[i], combine)
  for days in [3,5,7,10]:
    get_previous_xdays_avg(dfsclear[i], days,a)

for i in list(dfsclear.keys()):
  get_previous_momentums(dfsclear[i], drop_anomaly)
  get_previous_day (dfsclear[i], drop_anomaly)
```



```
for i in list(dfsclear.keys()):
  try:
    dfsclear[i].drop(columns = drop_anomaly, inplace = True)
  except:
    pass

for i in list(dfsclear.keys()):
  dfsclear[i] = dfsclear[i][dfsclear[i]['MIN'] > 12]
  dfsclear[i] = dfsclear[i][dfsclear[i]['NBA_FANTASY_PTS'] > 10]

for i in list(dfsclear.keys()):
  dfsclear[i].drop(columns = columns_to_drop, inplace = True)
  dfsclear[i].drop(columns = added_columns_to_drop, inplace = True)
  dfsclear[i].shape
  dfsclear[i].dropna(inplace=True)
```

Figure 7: Code for Feature Engineering implementation for Basic Features Datasets.

In conclusion, after the appropriate feature engineering, we appended each value (data frame) of each of the two dictionaries to a new data frame/table to create our two final datasets and stored them to continue with ML Models.

The process followed to create the final datasets that contain each player’s records is shown in Figure 8. The process starts with scraping different type of Player’s and Team’s data and then removing season ranking data from the scraped data. Said data are then merged to create an initial dataset for each player and each team. The process continues by merging each player’s dataset with his respective teams’ dataset to create the base player datasets that will be used as the base for the feature engineer in process.

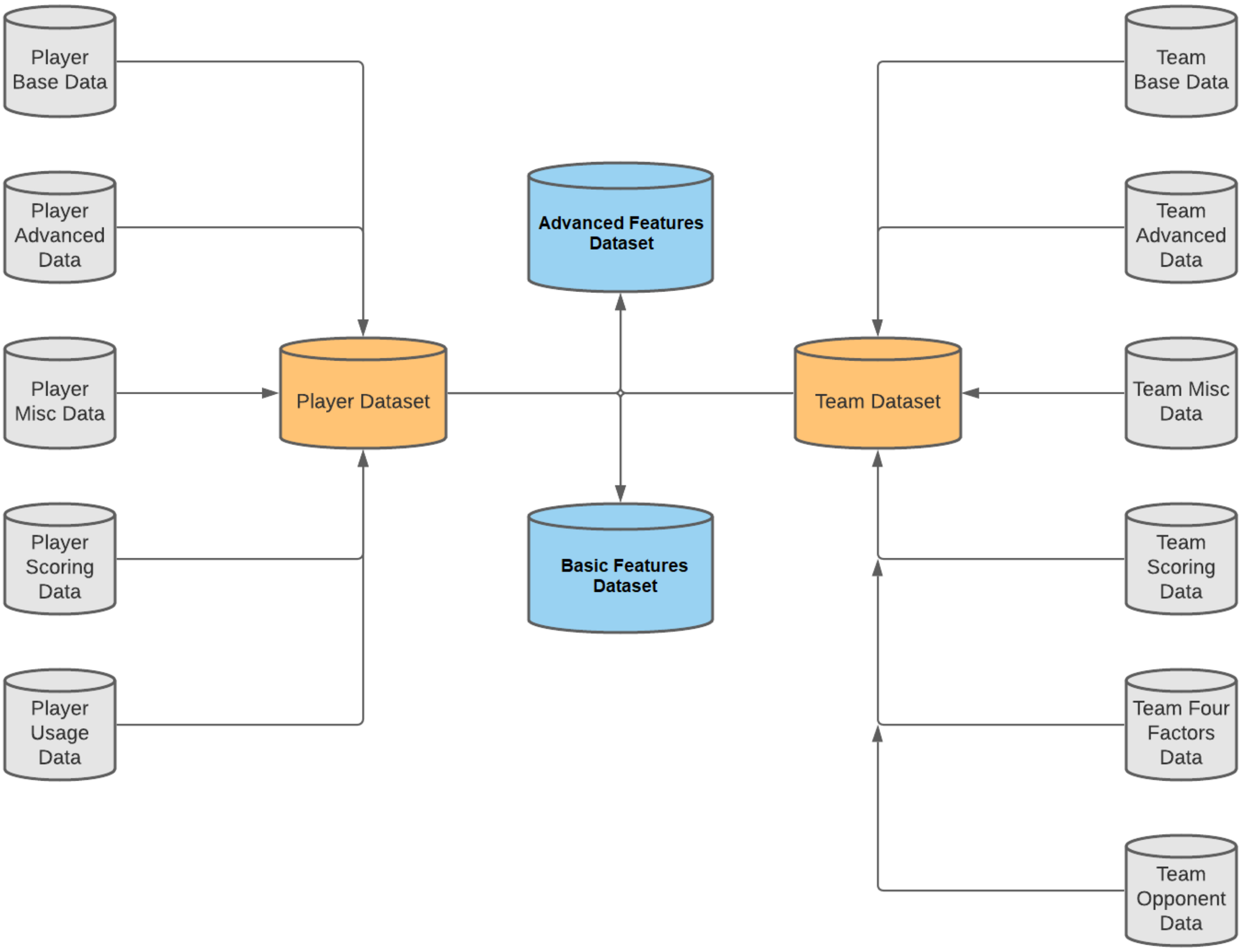


Figure 8: Flowchart Diagram of Data.

5 Modeling

5.1 Pycaret

[54] Pycaret is an open-source ML library that benefits from automating ML workflows. It replaces any handwritten code of ML algorithms in just a few lines. It offers plenty of ML methods in Classification, Regression, Clustering, Anomaly Detection, NLP, and Association Rules. In this particular project was selected because we needed to train each player's data and make predictions separately to optimize the results.

The benefits of working on the Regression problem with Pycaret are plenty; it offers automated data preprocessing services like scale and transformation, feature engineering, and feature selection. Also, automate trains the most of available regression models, tune the selected by the chosen metrics, and pick the best performing model based on test data. In addition, Pycaret finalizes the selected model using testing data to predict unseen data. Furthermore, it offers model ensembling like Bagging, Boosting, Stacking models, Blender models.

Bagging is an ensemble method that aims to decrease the variance. The goal here is to generate various subsets of data from a training sample selected at random via replacement. Each subset of data is utilized to train their regressor. This process results in an ensemble of several models, each generating predictions. The average of these predictions result in a more robust model [36].

Boosting is another ensemble method. The goal here is to train homogeneous regressors, sequentially, as each regressor depends on the predictions of the previous one [36].

Blending and stacking are two assembly approaches, that utilize multimple regressor trained on the same data and use their predictions as trained data for meta-regressor to produce final prediction.[37]

When the target is to train multiple models, it is a great choice that optimizes the results, while when the auto ml function is called, Pycaret picks the best-performing model. In addition, using Pycaret, you can analyze the selected model. Function plot_model Pycaret provides a detailed report and visualization charts to analyze the chosen model [34].

Pycaret can simultaneously train multiple ML regression models and compare them based on results on test data.

A brief description of each Model that Pycaret offers will follow.

- **Huber Regression**
  - Huber Regression technique uses a different loss function instead of the traditional least-squares; it is less sensitive to outliers in data [43].

- **Ridge Regression**
  - Ridge Regression is a specialized technique on data that suffer from multicollinearity. By using Ridge Regression, the parameters are shrunk, preventing multicollinearity, and finally, the complexity of the model is reduced by coefficient shrinkage [44].

- **Linear Regression**
  - Linear Regression is an essential and common technique used for predictive analysis. It refers to a linear approach for modeling between a scalar and the explanatory variables [45].

- **Least Angle Regression**
  - Least Angle Regression is a technique preferred for high dimensional data. Finding the higher correlated features to the target value pushes the regression line in this directive until it contacts another variable with the identical or more increased correlation [46].

- **Bayesian Ridge**
  - Bayesian regression is usually selected for insufficient or inadequately distributed data by formulating linear regression. It works by employing probability distributors instead of point estimates. The response output (y) is assumed to be computed by a probability distribution instead of estimated as a single value [43].

- **Orthogonal Matching Pursuit**
  - The Orthogonal Matching Pursuit approach is used to recover a high-dimensional sparse signal from a small set of noisy linear measurements. It is an iterative greedy method that selects the most correlated feature at each stage [44].

- **Passive Aggressive Regressor**
  - Passive Aggressive Regressor belongs to the category of online learning in ML. This technique works by feeding its instances sequentially, individually, or in groups called mini-batches. It is most commonly used in procedures where data stream in a continuous flow [45][46].

- **Adaboost**
  - Adaboost Regressor is a meta-estimator that works by matching a regressor on the original data, and in the next phase, copies of this Regressor on the same dataset using modified weights of instances based on errors in the first prediction [47].

- **Random Forest Regressor**
  - Random forest works by using ensemble methods (bagging). This technique starts by constructing a large number of decision trees and delivers the mean/mode of prediction of these [48].

- **Gradient Boosting Regressor**
  - Gradient Boosting Regressor technique implements a regression tree by fitting it on the negative gradient of the given loss function. This method enables the optimization of any differentiable loss function [49].

- **Extra Trees Regressor**
  - The Extra Trees Regressor method uses a meta estimator that fits several different decision trees on different sum samples of the dataset and improves accuracy by averaging [50].

- **Lasso Regression**
  - Lasso Regression is a regularization technique and a linear regression method that uses shrinkage. Usually, this method is preferred when a high level of multicollinearity is presented. While, when there is an appearance of a large number of features, it automatically performs feature selection [51].

- **Light Gradient Boosting Machine**
  - Light Gradient Boosting is an extension of the gradient boosting method. It follows an automated feature selection procedure and boosts examples with more considerable gradients [52].
  -

- **Decision Tree Regressor**

- ➢ Decision Tree Regressor is a popular regression that breaks down the dataset into smaller and smaller subsets samples. In this way, a decision tree is incrementally produced. The decision tree is constructed of nodes and leaf nodes in its final form [53].

5.2 Data Modeling

In our case, we use the Pycaret tool and Google Colab with 51GB storage RAM and Python 3 Google Compute Engine backend (GPU). It is worth mentioning that all following procedures are deployed to both of our datasets. First, we load our datasets' Advanced Features and Basic Features and encode the categorical features, 'SEASON_YEAR' and 'TEAM_ABBREVIATION'.

In the next phase, we split our datasets into smaller / per player datasets using dictionaries in which each key is the Player's name and as value the Dataframe with his records. In this way, we trained and evaluated each player model and selected the best one based on the model's MAPE score.

The split that is followed in Modeling is 70% of data is used for training, 20 % of data is used for Testing, and sorted by Date, last 10% of data is considered as Unseen data and used for evaluating our models and their predictions are used for predicting the best possible Lineup for the NBA Fantasy Tournament.

Furthermore, we take advantage of Feature Selection methods to reduce data dimensionality before training our models. The methods that are deployed to our dataset are the following.

At first, transform the target variable that we want to predict (NBA FANTASY PTS) using the Yeo-Johnson method because the distribution of NBA FANTASY PTS is a variable most of the time with non-symmetric distribution. The Yeo-Johnson method can make the distribution more symmetric.

Secondly, because the number of features is grand and our data per Player is contained because each season has maximum 82 regular-season games plus playoffs, which in a few cases a player can participate in all, we had to remove some features that did not add to the explained variance of the model. In this stage, three methods are followed.

We remove multicollinearity between features. In this method, the features that are highly linearly correlated with another feature variable and less correlated with the target variable are dropped in the same dataset. Our dataset contained high correlated features increasing the variance of the coefficients, making them unstable and noisy for the models. For this reason, the multicollinearity threshold is set to 0.50, resulting in the features with inter-correlations higher than 0.50 being dropped.

The second method that we use to constrain the feature space in order to improve efficiency in Modeling is the feature importance method. Using a mix of permutation importance approaches such as Random Forest, Adaboost, and Linear correlation with the target variable. We set the feature selection threshold to 0.90, meaning that the algorithm will keep features that explain at least 90% percent of the dataset's variance.

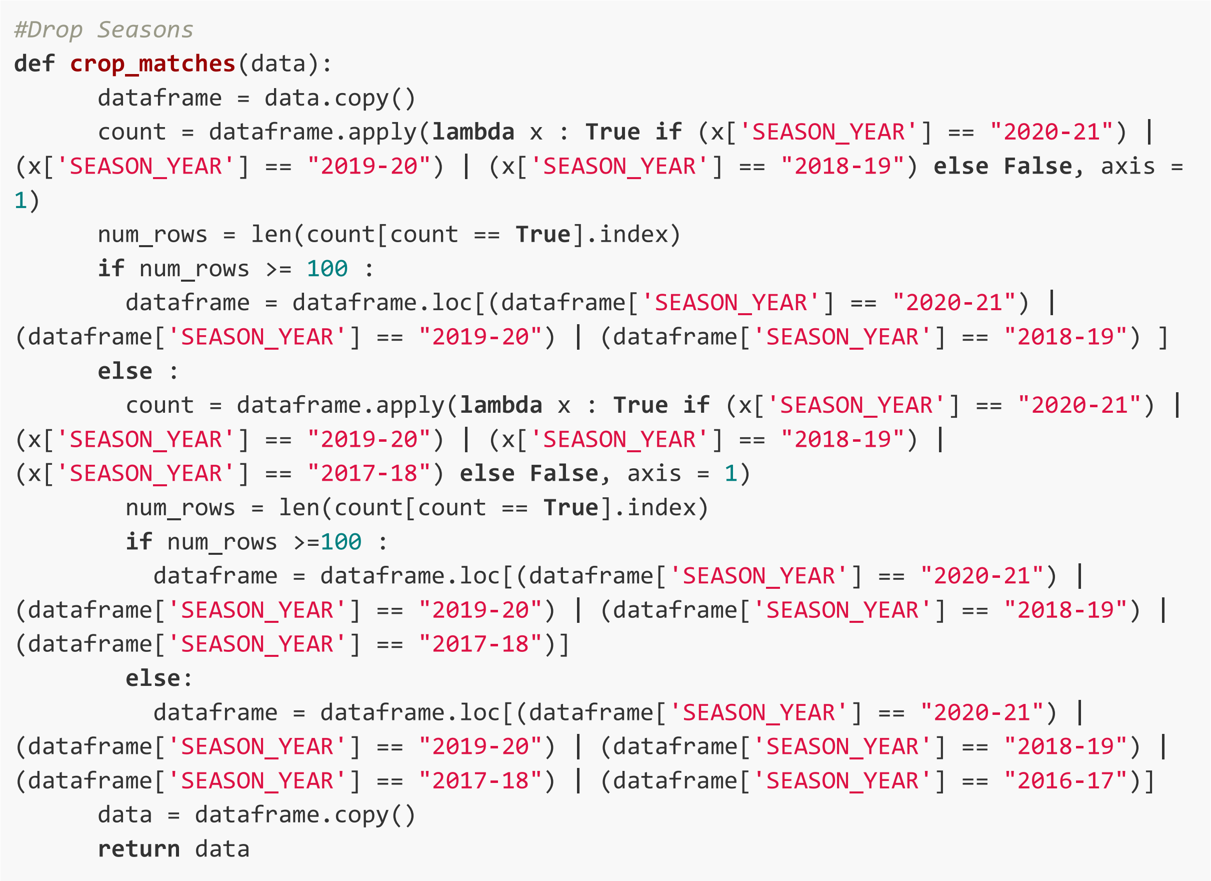
Thirdly, the last method to reduce feature space is related to categorical features ('PLAYOFFS', 'OPPONENT', 'SEASON_YEAR', 'TEAM_ABBREVIATION'). We used ignore low variance method for the categorical data. This method removes features with statistically insignificant variances from the dataset. The variance is computed by dividing the number of samples by the number of unique values and the rate of the most common value by the rate of the second most common value. More detailed, two conditions must be fulfilled to drop a feature by this method :

- Count of unique values in a feature / sample size < 10%
- Count of most common value / Count of second most common value > 20 times.

5.3 Model Selection

After the Feature Engineering, our next goal was to split our primary datasets to continue with training our models. To optimize our results, having as the target to get as better results as possible, we split our two primary datasets to per' player dataset as referred before using dictionaries, resulting in two datasets for each player.

The training, testing, and evaluation stages are performed for all available records for each player's dataset and a filtered version for each player's dataset. In the second version, restrictions on data are applied, while we take the newest records for each player, trying to find the best possible trend in the last three, four, and five seasons, depending on the amount of data of each player. While we tested that to perform our algorithms well, each dataset should contain at least one hundred records.



```
#Drop Seasons
def crop_matches(data):
      dataframe = data.copy()
      count = dataframe.apply(lambda x : True if (x['SEASON_YEAR'] == "2020-21") |
(x['SEASON_YEAR'] == "2019-20") | (x['SEASON_YEAR'] == "2018-19") else False, axis =
1)
      num_rows = len(count[count == True].index)
      if num_rows >= 100 :
        dataframe = dataframe.loc[(dataframe['SEASON_YEAR'] == "2020-21") |
(dataframe['SEASON_YEAR'] == "2019-20") | (dataframe['SEASON_YEAR'] == "2018-19") ]
      else :
        count = dataframe.apply(lambda x : True if (x['SEASON_YEAR'] == "2020-21") |
(x['SEASON_YEAR'] == "2019-20") | (x['SEASON_YEAR'] == "2018-19") |
(x['SEASON_YEAR'] == "2017-18") else False, axis = 1)
        num_rows = len(count[count == True].index)
        if num_rows >=100 :
          dataframe = dataframe.loc[(dataframe['SEASON_YEAR'] == "2020-21") |
(dataframe['SEASON_YEAR'] == "2019-20") | (dataframe['SEASON_YEAR'] == "2018-19") |
(dataframe['SEASON_YEAR'] == "2017-18")]
        else:
          dataframe = dataframe.loc[(dataframe['SEASON_YEAR'] == "2020-21") |
(dataframe['SEASON_YEAR'] == "2019-20") | (dataframe['SEASON_YEAR'] == "2018-19") |
(dataframe['SEASON_YEAR'] == "2017-18") | (dataframe['SEASON_YEAR'] == "2016-17")]
      data = dataframe.copy()
      return data
```

Figure 9: Code for filter only Last Seasons

Each player's training, test, and evaluation processes were made separately, and the total prediction MAPE score is calculated as the mean of all final models. Starting with training all fourteen available algorithms with ten-fold validation, for each player and comparing their results based on each MAPE score, after, we tuned the best three models and used the "blender" function of Pycaret tool, that combine the top three models, one more model "Voting Regressor" is made and compared with others. The next step was to compare all algorithms trained on the player's dataset and pick the best based on MAPE score on Test data. After this selection, the followed procedure was to train this model with all available data and test sets to finalize it. Lastly, we made our predictions on Unseen data using the finalized model and evaluated our results with actual data. The same procedure is followed in all datasets with Full, Basic Features.

5.3.1 Advanced Features Dataset Models

It occurs that every player's model performs differently, while after comparing all trained algorithms for each player dataset, the best model for each player is not the same and performs differently. Using data from the last ten seasons, Random Forest Regressor fits better in 50 datasets/players, Bayesian Ridge Regressor in 42, Voting Regressor in 39.

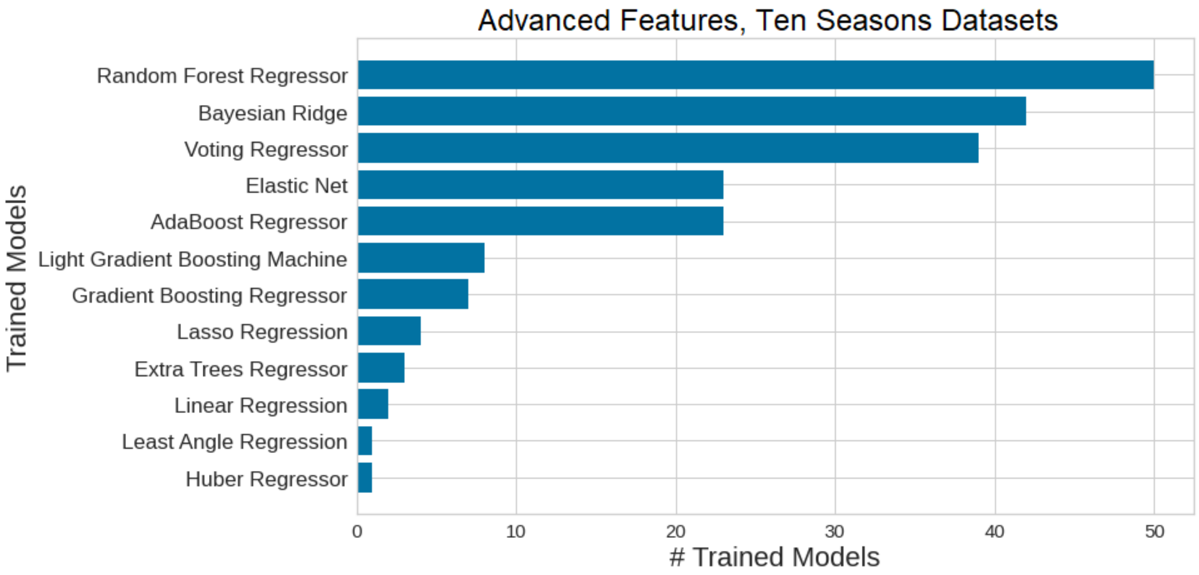


Figure 10: Trained models in Advanced Features Ten Seasons Datasets

While using only last seasons, Voting Regressor fits better in 45 players/ datasets, the second better fitting model in datasets is the Bayesian Ridge in 40 players/ datasets, and the third most accurate model in some of datasets/ players is the Random Forest Regressor in 33 of them.

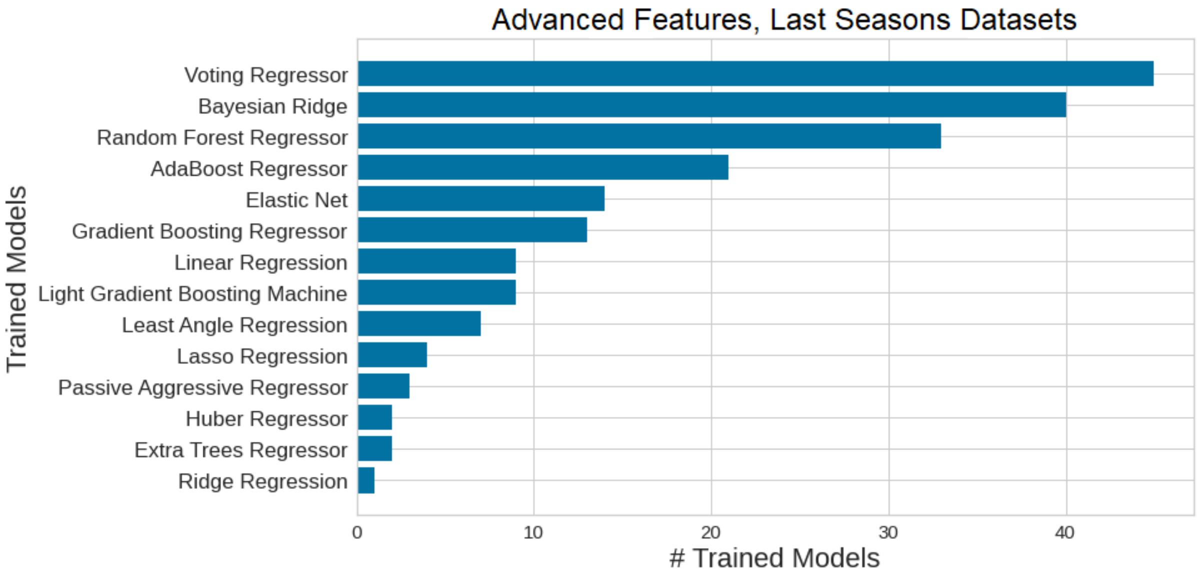


Figure 11: Trained models in Advanced Features, Last Seasons Datasets

It is worth mentioning that Linear, Extra Trees, Huber, Lasso, and Least Angle Regressors were selected as best-fitting models in a few cases. With Random Forest, Bayesian Ridge, and Voting Regressor being the most popular in both cases with ten seasons and last seasons data.

5.3.2 Basic Features Dataset Models

Using Basic Features Datasets, the same results are observed, that every player's model performs differently, while after comparing all trained algorithms for each player dataset, the best model for each player is not the same and performs differently. Train our algorithms in 10 Seasons with Basic Features Datasets, the model that fits better in most players/ datasets is the Voting Regressor while performing better in 61 of 203 of total selected players/ datasets. The second most popular algorithm is AdaBoost Regressor fitting in 43 and the third one Bayesian Ridge fitting in 32 of them.

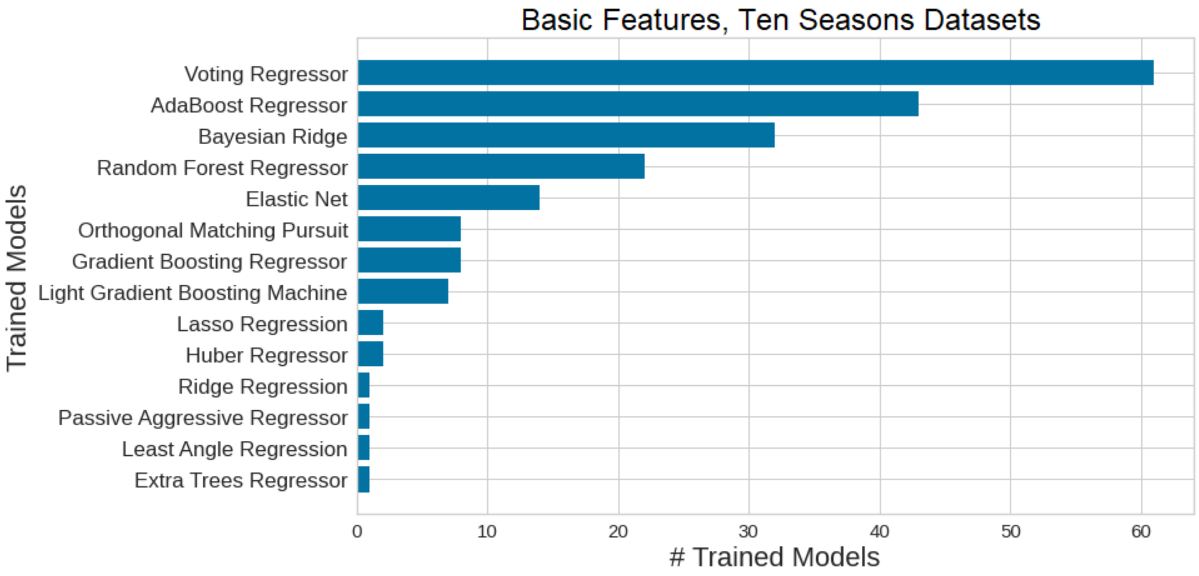


Figure 12: Trained models in Basic Features, Ten Seasons Datasets

Lastly, selecting only Last Seasons for training, testing, and evaluating our datasets, using only the Basic Features, the most popular model that fits better in most players/ datasets is again the Voting Regressor performing best in 42 players' datasets. The second most popular is the AdaBoost fitting better in 36 datasets and third the Random Forest Regressor fitting in 30 of 203 datasets.

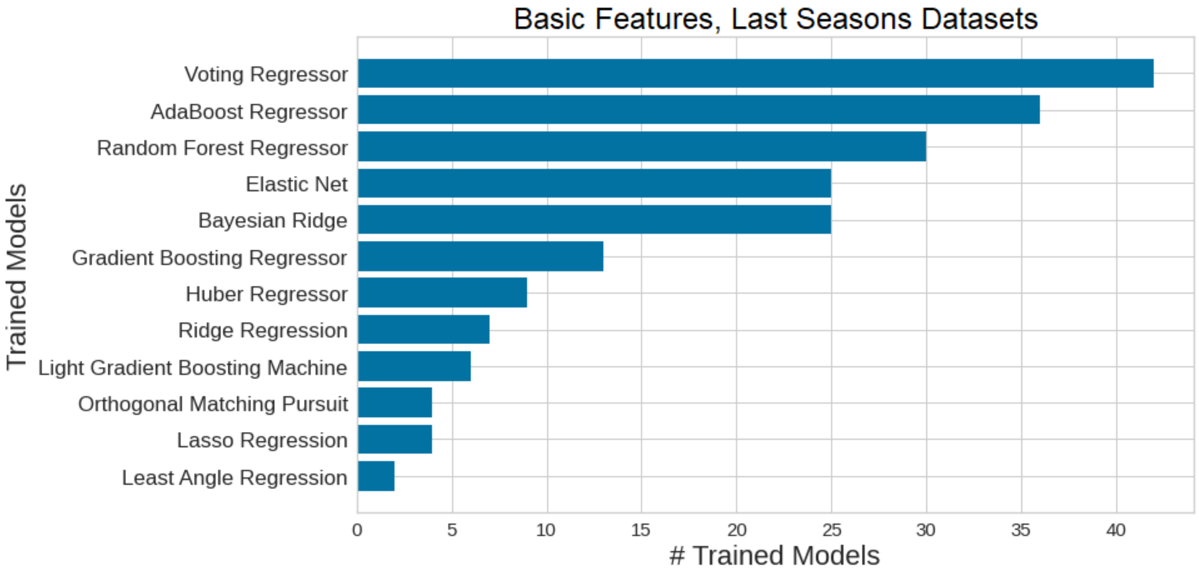


Figure 13: Trained models in Basic Features, Last Seasons Datasets

Finally, using a model selection strategy on the models trained, tested, and evaluated on data that consist of identical records (exact matches), we chose the best-performing model for each player to finalize our results.

6 Results

In this chapter, we present the results of our research, including the results for each dataset. Every player's dataset is trained separately and evaluated separately based on the Test set and Unseen Data. Our split in all datasets was set to 70% of the dataset used for training each model, 20% for testing, and 10% as Unseen data. Unseen data is selected from the tail of each dataset (newest data) to evaluate each player's model in the same period of timing. To evaluate the whole project's results and compete the final MAPE and MAE score, we calculated the average of total models MAPE and MAE score for each experiment, respectively.

Furthermore, except for NBA Fantasy points predictions results that would be evaluated, we present the Daily Lineup Optimizer (DLO) that can be used for NBA Fantasy Tournaments, in which we try to predict the best NBA Lineup with restrictions set by Fantasy Tournaments.

6.1 Advanced Features Datasets

Advanced Features datasets contain 269 features, and each player's model is trained with all available data from Season 2010-11 and separately with data from Last Seasons.

6.1.1 Ten Seasons Datasets

The scores for all 203 models trained with all available records since season 2010-11 are shown in Table 5.

Averages of all 203 selected Models (Advanced Features, Ten Seasons Datasets)	
Test MAE	7.503
Test MAPE	0.306
Unseen MAE	8.032

Unseen MAPE	0.307

Table 5: Results of Advanced Features, Ten Seasons Dataset

We observe that MAPE results on test data are equal with these on Unseen Data, scoring 0.30 (30%), which means that we avoid overfitting, and our models are stable. However, MAE is different on Test and Unseen data because the variance of our target value is more significant on Unseen data.

6.1.2 Last Seasons Datasets

The mean of MAPE and MAE scores for all 203 models trained with Last Seasons records is shown in Table 6.

Averages of all 203 selected Models (Advanced Features, Last Seasons Datasets)	
Test MAE	7.977
Test MAPE	0.316
Unseen MAE	8.150
Unseen MAPE	0.311

Table 6: Results of Advanced Features, Last Seasons Dataset

We also observe that MAPE in Test and Unseen data is similar again, 31%. However, MAE scores are closer to each other than the Ten Seasons models mean. Nevertheless, models trained on Last Seasons performed worse than those trained on the Ten Seasons dataset.

6.2 Basic Features Datasets

Using Last Three Seasons data, we set the restriction that the player must participate in at least 100 games. Otherwise, we added the previous season's games that he participated until we can successfully make our predictions with at least 100 records. We achieve the same length of records for each player with Advanced Features Datasets in Basic Features Datasets. Moreover, the filter for Last Seasons is the same.

6.2.1 Ten Seasons Datasets

The mean of the metrics of all 203 models trained with all available records since season 2010-11 is shown in Table 7.

Averages of all 203 selected Models (Basic Features, Ten Seasons Datasets)	
Test MAE	7.060
Test MAPE	0.289
Unseen MAE	7.544
Unseen MAPE	0.286

Table 7: Results of Basic Features, Ten Seasons Dataset

As we can observe, the results above are the best achieved so far. At the same time, the MAPE average for all players is 0.289 and 0.2866 on Test Data and Unseen Data, respectively. In addition, the MAE score is significantly lower than in other experiments transformed at 7.06 on Test and 7.54 on Unseen data.

6.2.2 Last Seasons Datasets

The mean of the metrics of all 203 models trained with all available records since Last Seasons is shown in Table 8.

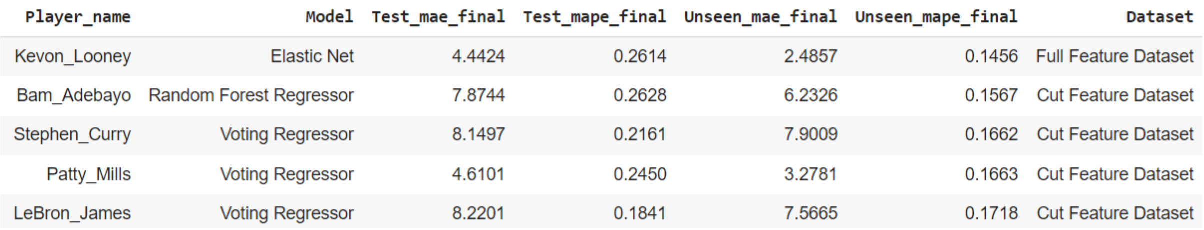
Averages of all 203 selected Models (Basic Features, Last Seasons Datasets)	
Test MAE	7.466
Test MAPE	0.297
Unseen MAE	8.017
Unseen MAPE	0.309

Table 8: Results of Basic Features, Last Seasons Dataset

It was evident that using Last Seasons datasets for predicting the accurate target value of NBA Fantasy Points for each player using the Basic Features; the better strategy is to use all available records offered. While all results are lower in this case, having Test and Unseen MAPE score around 0.30 and MAE score bigger than in any Test and Unseen Data in contrast with Last Season Datasets.

6.3 Optimizing our Results

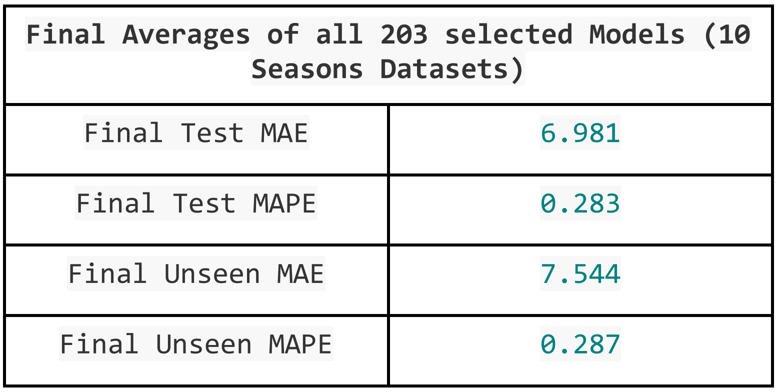
We successfully predicted the target value of NBA Fantasy with an average Test MAPE 0.289 and Unseen MAPE 0.286 using Basic Features Ten Seasons Datasets. In the next phase, we optimize these results by contrasting these results with the results of the Advanced Features Ten Season Datasets. At the same time, both datasets contained the same number of records for each player. We perform a model selection by results of Test Data. In this way, our final results include models trained on both Full and Basic Features Datasets.



Player_name	Model	Test_mae_final	Test_mape_final	Unseen_mae_final	Unseen_mape_final	Dataset
Kevon_Looney	Elastic Net	4.4424	0.2614	2.4857	0.1456	Full Feature Dataset
Bam_Adebayo	Random Forest Regressor	7.8744	0.2628	6.2326	0.1567	Cut Feature Dataset
Stephen_Curry	Voting Regressor	8.1497	0.2161	7.9009	0.1662	Cut Feature Dataset
Patty_Mills	Voting Regressor	4.6101	0.2450	3.2781	0.1663	Cut Feature Dataset
LeBron_James	Voting Regressor	8.2201	0.1841	7.5665	0.1718	Cut Feature Dataset

Table 9: Top 5 most accurate predictable Players on Unseen Data with Ten Seasons Datasets

The mean of the metrics of all 203 models trained with all available records since season 2010-11 is shown in Table 10.



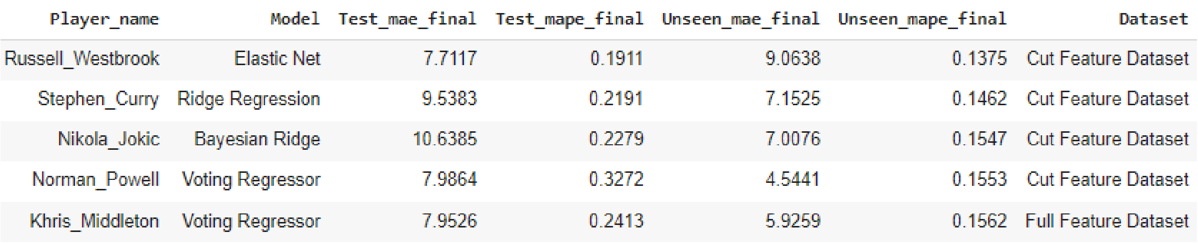
Final Averages of all 203 selected Models (10 Seasons Datasets)	
Final Test MAE	6.981
Final Test MAPE	0.283
Final Unseen MAE	7.544
Final Unseen MAPE	0.287

Table 10: Final Results for Ten Seasons Datasets

We checked that all models, in any case, are stable, and we prevented overfitting; for this reason, we considered making this model selection for our final models, resulting in a more accurate Test MAPE and MAE score for future long-term predictions.

6.3.1 Results Optimization

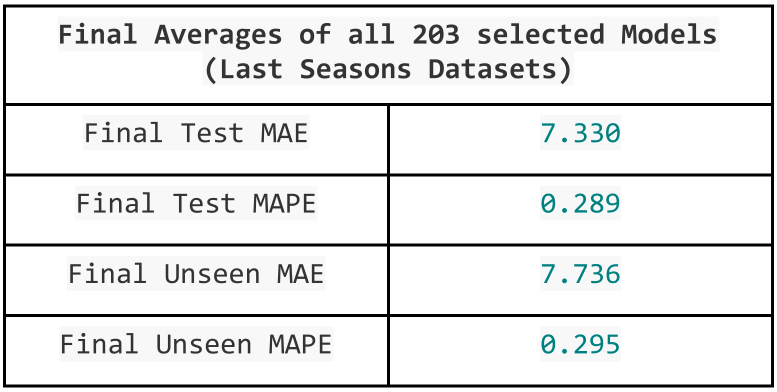
The same procedure is followed for Last Seasons datasets. We produce these results for short terms results, while Last Season models are made to predict around the last 10 Matches (Unseen Data). The following results are related to Last Seasons datasets, merging both Full and Basic Features Datasets.



Player_name	Model	Test_mae_final	Test_mape_final	Unseen_mae_final	Unseen_mape_final	Dataset
Russell_Westbrook	Elastic Net	7.7117	0.1911	9.0638	0.1375	Cut Feature Dataset
Stephen_Curry	Ridge Regression	9.5383	0.2191	7.1525	0.1462	Cut Feature Dataset
Nikola_Jokic	Bayesian Ridge	10.6385	0.2279	7.0076	0.1547	Cut Feature Dataset
Norman_Powell	Voting Regressor	7.9864	0.3272	4.5441	0.1553	Cut Feature Dataset
Khris_Middleton	Voting Regressor	7.9526	0.2413	5.9259	0.1562	Full Feature Dataset

Table 11: Top 5 most accurate predictable Players on Unseen Data with Last Seasons Datasets

The mean of the metrics of all 203 models trained with all available records in Last Seasons is shown in Table 12.



Final Averages of all 203 selected Models (Last Seasons Datasets)	
Final Test MAE	7.330
Final Test MAPE	0.289
Final Unseen MAE	7.736
Final Unseen MAPE	0.295

Table 12: Final Results for Last Seasons Datasets

As we can see, these results are efficient also while the difference in MAPE is only 0.8% on the Unseen set of data predictions.

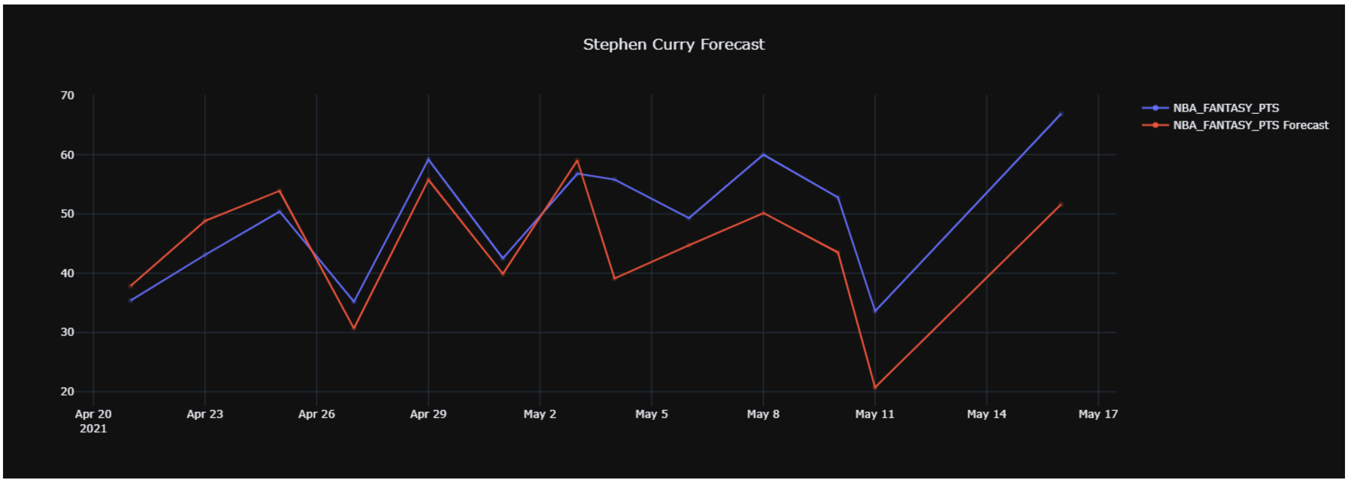


Figure 14: Stephen Curry Forecast Results with Last Seasons Dataset

We needed to save our predictions for the next phase of DLO. We used dictionaries having as key Player Name and value the prediction table. However, each player's prediction table had to be selected carefully based on the final model and the predictions made from this specific model and dataset. We had four predictions tables for each player, and based on the model and dataset used, the correct prediction table is selected for each one.

The code for this process is shown in Figure 15, while `predictions_full` and `predictions_cut` tables had the form of Table 11.

```
#Get the correct prediction table for each Player
#Based on model and Dataset used for training model and prediction are done

names = df['Player_name'].unique().tolist()
dictionary_final = {}
groups_full = predictions_full.groupby(predictions_full.PLAYER_NAME)
groups_CUT = predictions_cut.groupby(predictions_cut.PLAYER_NAME)

for i in names:
  if df.loc[(df['Player_name'] == i) & (df['Dataset'] == 'Full Feature
Dataset')].any().any():
    dictionary_final[i] = groups_full.get_group(i)

  elif df.loc[(df['Player_name'] == i) & (df['Dataset'] == 'Cut Feature
Dataset')].any().any():
    dictionary_final[i] = groups_cut.get_group(i)

  else:
    print('Error')
```

Figure 15: Code for player's prediction tables match by model and dataset.

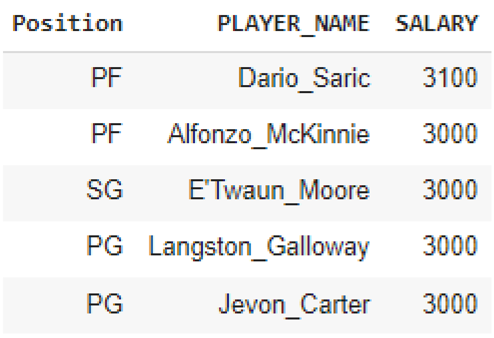
6.4 Daily Lineup Optimizer

This section presents the built-up of an NBA Daily Lineup Optimizer, whose goal is to calculate the best possible combination of the picks that will offer the maximum total NBA Fantasy Points for a Match Day. This optimizer is based on Fantasy Tournaments that several betting companies offer. These Tournaments' goal is to build a team that their players will score the most Fantasy Points. Of course, some restrictions are applied on player selection. These restrictions are the following:

- Buy a player at most once.
- Include players from at least 2 different NBA games
- The 8 roster positions are:
  - One PG (Point Guard)
  - One SG (Shooting Guard)
  - One SF (Small Forward)
  - One PF (Power Forward)
  - One C (Center)
  - One G (PG,SG)
  - One F (SF,PF)
  - One Util (PG, SG,SF,PF,C)
- Spend no more than $60,000.

6.4.1 Lineup Creation Process

In Fantasy Tournaments, there is always a total salary limit, while each player costs a salary to our portfolio. However, our already owned datasets were not related to Fantasy Tournaments, and we missed Salary data and the Position of each player. It is worth noting that the salary variable. To perform our final predictions, we had to acquire the related salary and position data from the Game Date that we wanted to predict the best possible Lineup based on our predicted Fantasy Points records. For this reason, we accessed the DraftKings site and acquired the missing data. This dataset contains every player who could participate in this day's game, his salary value, and his Position in court.



Position	PLAYER_NAME	SALARY
PF	Dario_Saric	3100
PF	Alfonzo_McKinnie	3000
SG	E'Twaun_Moore	3000
PG	Langston_Galloway	3000
PG	Jevon_Carter	3000

Table 13: DraftKings Dataset

The build-up of the best possible Lineup for a specific game is an optimization problem, and to solve it, we used Linear Optimization [56]. For this reason, we used the PuLP library [57]. We used the already scraped dataset from DraftKings and predictions made from Last Seasons datasets to generate our Lineup, even if the scores are lower than 10 Seasons data prediction's results. This is because models that trained using only data from the Last Seasons can better capture the player’s form as a trend. Since the sample size is smaller the date difference from the train set observations and the test set observations is smaller.

To start with the best Lineup prediction, we filtered players, whose performance we predicted, who participated to a specific match day. For our experiment, we selected one of the final games of the Regular Season in 2021 (15 May 2021). 26 different NBA teams participated to 13 events (games) on this Match Day.

After filtering the players that participated in these events, we used a pool of 53 available players to generate our Optimized Lineup. It is worth mentioning that we have a slight pull because, in our research, we contained our predictions to players that are more likely to participate in an event and have a good performance.

6.4.2 Lineup Results

Our aim was to create a lineup based on our predictions that will have the maximum possible sum of NBA Fantasy Points for this matchday, and after evaluating it by the actual sum of NBA Fantasy Points from the selected players.

Our generated Lineup for the 15th of May 2021 Matchday based on Predicted Fantasy Points scored 359 Predicted Fantasy Points, which is the best combination with the specific restrictions that we, set and scored 298 Actual Fantasy Points. These results are considered good, as any lineup that scores around 300 in this kind of Tournaments is considered a good result, specifically on the 15th of May 2021 that our predictions are made, two Fantasy Tournaments took place. At the first Tournament, the average cash line that presents the lower limit that the user’s lineup wins was 243.5 Fantasy Points, and in the second one, it was 294.5 Fantasy Points. [54]

Player Name	Position	Salary
Andre_Iguodala	SF	2400
Bruce_Brown	SF	4300
Caris_LeVert	SG	8600
Devin_Booker	SG	8500
James_Harden	PG	10800
Karl_Anthony_Towns	C	10100
LeBron_James	PG	9600
Thaddeus_Young	PF	5600
Total Predicted Fantasy Points : 358.7		
Total Actual Fantasy Points : 297.5		
Spending $: 59900		
Position Restrictions fulfilled		
PG	2	
SG	2	
SF	2	
PF	1	
C	1	

Table 14: Lineup Prediction for the 15th of May 2021 Matchday

6.5 Evaluation of Results

This section refers to issues that came up during the project and the procedures that we followed to overcome these. Also, the results of this project are evaluated.

The first challenge of this project was to select the appropriate amount and type of data to use for prediction making. Basketball is a game rich off statistics, and almost every statistic can be helpful for research purposes. For this reason, we had to select the type of data and the appropriate number of Seasons correctly. During Preprocessing, some useless attributes were eliminated. Those were different kinds of Ranks that refer to end of season ranks of players or teams for all the kinds of statistics because this kind of data does not give further information for the performance of each player or team in any particular game.

However, extra data could be gathered from wearable devices or betting odds, which is more likely to improve model performance. Nevertheless, it is hard to gather such historical data because they are not accessible online. In addition, financial data about players and teams could be efficient for our models.

Another problem during this research is that because we choose to use one final model for each player if he is a rookie or injured for a long time, we would lack data for splitting sets to train, test, and evaluation. For this reason, we focus our research on players who have participated at least in 100 matches and at least for one period of each match.

The results of this project are promising. Achieved accuracy in the long and short terms can be considered more than satisfactory. While our models were trained with data from Season 2010-11, we can accurately predict their performance in terms of Fantasy Points for over half-season with MAE lower than 29%. The same efficient results occur for short terms (around 15 games) using Last Seasons data.

Building an efficient Lineup with restrictions from scratch is also a challenge. At the same time, our predicted lineup is based on predicted data. We achieved 298.5 Fantasy Points on one Game Date, which is considered a good score, while the average cash line from two tournaments that were held on the 15th of May 2021 were 243.5 and 294.5 [54].

7 Conclusion and Future Work

In this study, we successfully predicted each player's performance-focused on his historical data in terms of Fantasy Points. Additionally, using these predictions, we created a Lineup Optimizer with restrictions which purpose was to maximize the total Fantasy Points of the built Lineup for a specific date.

7.1 Conclusion

Player performance prediction, probably the most fundamental case of sports analytics, was analyzed in this project.

The first stage of this project was to build several successful models to predict each NBA player's performance as well as possible, based on their historical data. For this reason, several models for each player were created and compared to each other to optimize the results. The data were just historical (from Season 2010-11 to Season 2020-21) and contained plenty of different kinds of NBA's Box Scores statistics. Four different experiments were conducted separately for each player with different data periods and different kinds of NBA's Box Score statistics to select the best-performing model. Results showed that we could successfully predict each player's performance in Fantasy Points with MAPE 28,9% and MAE 6.98 on the Test set and 7.54 on Unseen data.

In addition, in the second stage of the project, we successfully built a Daily Lineup Optimizer to maximize the total sum of Fantasy Points. Using our predictions from a specific date, we managed to create an eight-player lineup that scores 298 Fantasy points, which is considered a successful result in contrast with the available Tournament's results for the 15th of May [54].

Concluding, sports analytics is already acknowledged as a hot field that teams, players, and companies are taking into account. Although data are generated rapidly by players and teams during training and matches, the collection and analysis of these data make DM and ML excellent tools for everyone related to Sports. Nowadays, every NBA team has Data Science and Sports Analysis departments, taking this expanding industry into account.

7.2 Future Work

This project shows that it is possible to make short and long-term predictions about player performance based on historical data. For this reason, researchers could work on the same path for further improvement. However, based on this project, there is room for improvement.

Our proposed method involves a detailed prediction-making process based on different metrics related to Box Scores statistics. It contains player's Basic, Advanced, Misc, Scoring, Usage types of metrics and team's Base, Advanced, Misc, Scoring, Four Factors and Opponent types of metrics. While there are not many more statistics related to Box Scores, further improvement in the results might exist, with different analysis.

One idea that could probably improves the results is sentiment analysis on Twitter and other social platforms for every player and team. Furthermore, these results can be used as features at the final forecast. Sentiment Analysis will provide an overview of the public opinion on every player's upcoming performance. However, it should be done precisely to ensure that results are related to future performance and not an evaluation of historic performance [59][60][61][62][63].

Additionally, Association Rules in the analysis of Basketball tactics can optimize results also. In addition, it can provide hidden relations between the players, give us an overview of performance improvement and deterioration for each player depending on the starting lineup. In addition, knowing the starting lineup and subs for any upcoming match, Association Rules results can also be considered a feature in player performance forecast [58].

Finally, a potentially good extension in our data could be the betting odds. Betting odds related to match result, and team points scored, assists, blocks, and other statistics offered for betting reasons. Moreover, odds related to a player's performance can be beneficial, offering the potential probability of each player points score, assist, blocks, turnovers, double-double, and triple-double. However, this type of historical data is hard to find, and players' performance odds are usually offered only for certain star players.

Bibliography

[1]Singh, N., 2020. Sport Analytics: A Review. *The International Technology Management Review*, *9*(1), pp.64-69.

[2]Segal, S., 2012. An Unbreakable Game: Baseball and Its Inability to Bring About Equality during Reconstruction. *The Historian*, *74*(3), pp.467-494.

[3]Chadwick, H., 1860. *Beadle's Dime Base-ball Player: A Compendium of the Game Comprising Elementary Insructions of this American Game of Ball: Together with the Revised Rules and Regulations for 1860, Rules for the Formation of Clubs, Names of the Officers and Delegates to the General Convention, & C*. Irwin P. Beadle.

[4]Rickey, B., 1954. Goodby to some old baseball ideas. *Life*, *2*, pp.78-89.

[5]Neyer, Rob, "Sabermetrics," [Online]. Available: https://www.britannica.com/sports/sabermetrics.

[6]R. Lederer, "Abstracts From The Abstracts," 14 November 2004. [Online]. Available: http://baseballanalysts.com/archives/2004/11/abstracts_from_20.php.

[7]Lewis, M., 2004. *Moneyball: The art of winning an unfair game*. WW Norton & Company.

[8]Steinberg, L., 2015. Changing the game: the rise of sports analytics. *Forbes. Retrieved March*, *14*, p.2017.

[9]Chazan-Pantzalis, V., 2020. Sports Analytics Algorithms for Performance Prediction.

[10]Magoun, F.P., 1938. *History of Football from the Beginnings to 1871* (p. 125). Bochum-Langendreer: H. Pöppinghaus.

[11]Apostolou, K. and Tjortjis, C., 2019, July. Sports Analytics algorithms for performance prediction. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)* (pp. 1-4). IEEE.

[12]Larson, O., 2001. Charles Reep: A major influence on British and Norwegian football. *Soccer & Society*, *2*(3), pp.58-78.

[13]Wilson, J., 2013. *Inverting the pyramid: the history of soccer tactics*. Bold Type Books.

[14]Lyons, K., 1994. Lloyd Lowell Messersmith and the Origins of Notational Analysis. *Centre for Notational Analysis, Cardiff Institute of Higher Education, Cardiff*.

[15]NABC., Timeout Feature: The Early Days Of Basketball Analytics. [online] Available at: <https://www.nabc.com/nabc_releases/timeout_features/2016/timeout-analytics>.

[16] Steinberg, L., 2015. Changing the game: the rise of sports analytics. *Forbes. Retrieved March*, *14*, p.2017.

[17]Hamdad, L., Benatchba, K., Belkham, F. and Cherairi, N., 2018, May. Basketball analytics. Data mining for acquiring performances. In *IFIP International Conference on Computational Intelligence and Its Applications* (pp. 13-24). Springer, Cham.

[18]Ahmadalinezhad, M. and Makrehchi, M., 2020. Basketball lineup performance prediction using edge-centric multi-view network analysis. *Social Network Analysis and Mining*, *10*(1), pp.1-11.

[19]Casals, M. and Martinez, A.J., 2013. Modelling player performance in basketball through mixed models. *International Journal of performance analysis in sport*, *13*(1), pp.64-82.

[20]Sarlis, V. and Tjortjis, C., 2020. Sports analytics—Evaluation of basketball players and team performance. *Information Systems*, *93*, p.101562.

[21]South, C., Elmore, R., Clarage, A., Sickorez, R. and Cao, J., 2019. A Starting Point for Navigating the World of Daily Fantasy Basketball. *The American Statistician*, *73*(2), pp.179-185.

[22]Young, C., Koo, A., Gandhi, S. and Tech, C., 2020. Final Project: NBA Fantasy Score Prediction.

[23]Hermann, E. and Ntoso, A., 2015. Machine Learning Applications in Fantasy Basketball. semantic scholar.

[24]Earl, J., 2019. Optimaztion of Fantasy Basketball Lineups via Machine Learning.

[25]Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G. and Wang, W., 2006. Data mining curriculum: A proposal (Version 1.0). Intensive Working Group of ACM SIGKDD Curriculum Committee, 140, pp.1-10.

[26]Jackson, J., 2002. Data mining; a conceptual overview. Communications of the Association for Information Systems, 8(1), p.19.

[27]Daniel, J., 2021. Machine Learning Tutorial for Beginners: What is, Basics of ML, Available:https://www.guru99.com/machine-learning-tutorial.html

[28]Goodfellow, I., Bengio, Y. and Courville, A., 2016. Machine learning basics. Deep learning, 1(7), pp.98-164.

[29]Saravanan, R. and Sujatha, P., 2018, June. A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 945-949). IEEE.

[30]Wang, D., 2001. Unsupervised learning: foundations of neural computation. Ai Magazine, 22(2), pp.101-101.

[31]Littman, M.L. and Moore, A.W., 1996. Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research 4.

[32]Sarlis, V., Chatziilias, V., Tjortjis, C. and Mandalidis, D., 2021. A data science approach analysing the impact of injuries on basketball player and team performance. Information Systems, p.101750.

[33]Eisenberg, J., 2016. Combating Uncertainty with Context: Optimal Lineup Construction in Daily Fantasy Baseball.

[34] (2000) MEAN ABSOLUTE PERCENTAGE ERROR (MAPE). In: Swamidass P.M. (eds) Encyclopedia of Production and Manufacturing Management. Springer, Boston, MA . https://doi.org/10.1007/1-4020-0612-8_580

[35] (2011) Mean Absolute Error. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_525

[36] Sutton, C.D., 2005. Classification and regression trees, bagging, and boosting. Handbook of statistics, 24, pp.303-329

[37] Ji, A. and Levinson, D., 2020. Injury severity prediction from two-vehicle crash mechanisms with machine learning and ensemble models. IEEE Open Journal of Intelligent Transportation Systems, 1, pp.217-226.

[38] Gain, U. and Hotti, V., 2021, February. Low-code AutoML-augmented Data Pipeline–A Review and Experiments. In Journal of Physics: Conference Series (Vol. 1828, No. 1, p. 012015). IOP Publishing.

[39] Sun, Q., Zhou, W.X. and Fan, J., 2020. Adaptive huber regression. Journal of the American Statistical Association, 115(529), pp.254-265.

[40] Marquardt, D.W. and Snee, R.D., 1975. Ridge regression in practice. The American Statistician, 29(1), pp.3-20.

[41] Maulud, D. and Abdulazeez, A.M., 2020. A Review on Linear Regression Comprehensive in Machine Learning. Journal of Applied Science and Technology Trends, 1(4), pp.140-147.

[42] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R., 2004. Least angle regression. The Annals of statistics, 32(2), pp.407-499.

[43] Massaoudi, M., Refaat, S.S., Abu-Rub, H., Chihi, I. and Wesleti, F.S., 2020, July. A hybrid Bayesian ridge regression-CWT-catboost model for PV power forecasting. In 2020 IEEE Kansas Power and Energy Conference (KPEC) (pp. 1-5). IEEE.

[44] Cai, T.T. and Wang, L., 2011. Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Transactions on Information theory, 57(7), pp.4680-4688.

[45] Shalev-Shwartz, S., Crammer, K., Dekel, O. and Singer, Y., 2003. Online passive-aggressive algorithms. Advances in neural information processing systems, 16, pp.1229-1236.

[46]KHARWAL, A., 2021. Passive Aggressive Regression in Machine Learning, Avaliable:https://thecleverprogrammer.com/2021/07/04/passive-aggressive-regression-in-machine-learning

[47] Solomatine, D.P. and Shrestha, D.L., 2004, July. AdaBoost. RT: a boosting algorithm for regression problems. In 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541) (Vol. 2, pp. 1163-1168). IEEE.

[48] Liu, Y., Wang, Y. and Zhang, J., 2012, September. New machine learning algorithm: Random forest. In International Conference on Information Computing and Applications (pp. 246-252). Springer, Berlin, Heidelberg.

[49] Natekin, A. and Knoll, A., 2013. Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, p.21.

[50] John, V., Liu, Z., Guo, C., Mita, S. and Kidono, K., 2015, November. Real-time lane estimation using deep features and extra trees regression. In Image and Video Technology (pp. 721-733). Springer, Cham.

[51] Roth, V., 2004. The generalized LASSO. IEEE transactions on neural networks, 15(1), pp.16-28.

[52] Chakraborty, D., Elhegazy, H., Elzarka, H. and Gutierrez, L., 2020. A novel construction cost prediction model using hybrid natural and light gradient boosting. Advanced Engineering Informatics, 46, p.101201.

[53] Rathore, S.S. and Kumar, S., 2016. A decision tree regression based approach for the number of software faults prediction. ACM SIGSOFT Software Engineering Notes, 41(1), pp.1-6.

[54] PyCaret.org. PyCaret, April 2020. URL https://pycaret.org/about. PyCaret version 1.0.0.

[55] nba.com, NBA’s Official Site

[56] Bertsimas, D. and Tsitsiklis, J.N., 1997. Introduction to linear optimization (Vol. 6, pp. 479-530). Belmont, MA: Athena Scientific.

[57] Mitchell, S., OSullivan, M. and Dunning, I., 2011. PuLP: a linear programming toolkit for python. The University of Auckland, Auckland, New Zealand, p.65.

[58] Ghafari, S.M.; Tjortjis, C. 'A Survey on Association Rules Mining Using Heuristics', WIREs Data Mining and Knowledge Discovery, Vol. 9, no. 4, July/August 2019, (Wiley)

Yakhchi S., Ghafari S.M., Tjortjis C., Fazeli M., 'ARMICA-Improved: A New Approach for Association Rule Mining', Lecture Notes in Artificial Indigence, vol 10412, pp. 296-306, 2017, Springer-Verlag

[59] P. Koukaras, D. Rousidis and C. Tjortjis, 2021, 'Introducing a novel Bi-functional method for Exploiting Sentiment in Complex Information Networks', Int'l Journal of Metadata, Semantics and Ontologies. Inderscience

[60]C. Nousi and C. Tjortjis, 2021, 'A Methodology for Stock Movement Prediction Using Sentiment Analysis on Twitter and StockTwits Data', Proc. 6th IEEE South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM 21)

[61] E. Tsiara, C. Tjortjis, 2020, 'Using Twitter to Predict Chart Position for Songs', 16th Int'l Conf. on Artificial Intelligence Applications and Innovations (AIAI 20)

[62] Beleveslis D., Tjortjis C., Psaradelis D. and Nikoglou D., 2019, 'A Hybrid Method for Sentiment Analysis of Election Related Tweets', 4th IEEE SE Europe Design Automation, Computer Engineering, Computer Networks, and Social Media Conf. (SEEDA-CECNSM).

[63] L. Oikonomou and C. Tjortjis, 2018, 'A Method for Predicting the Winner of the USA Presidential Elections using Data Extracted from Twitter', 3rd IEEE SE Europe Design Automation, Computer Engineering, Computer Networks, and Social Media Conf. (IEEE SEEDA-CECNSM18)

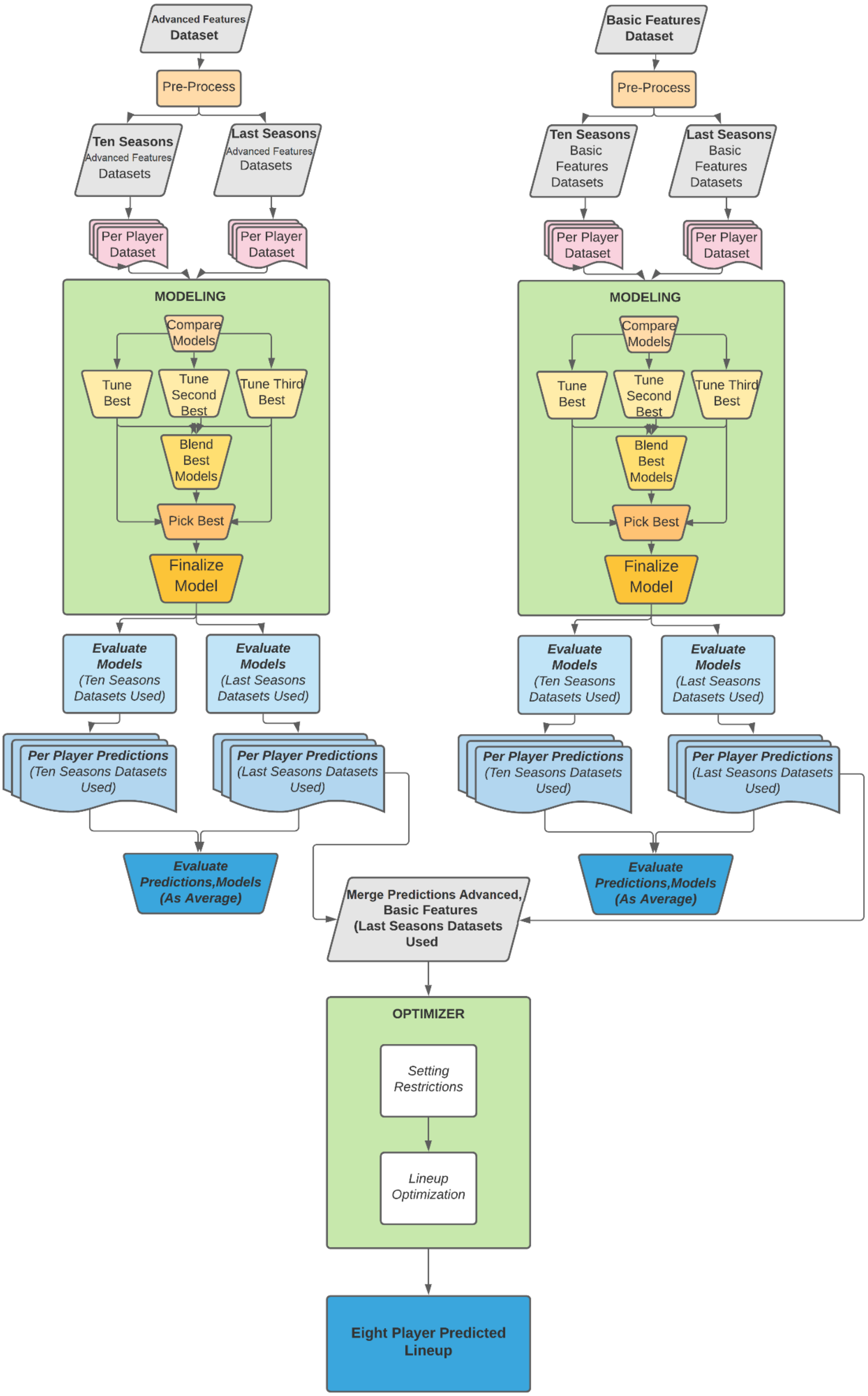
Appendices

We include here our Process Model (Appendix A).

The flow of the process starts with the two types of datasets (Basic and Advanced Features datasets from Section 4.1.1) from which two new datasets are created one contain data from all seasons and one with only the last season. Then multiple ML models are trained on and compared on each dataset using 10-fold cross validation. The 3 best models in terms of MAPE for each dataset are then tuned and then used to create a blended model. For each dataset all the trained models are evaluated on the test set that was held out by Pycaret for the final evaluation. Finally, the best model is used to generate the predictions that will be fed on the lineup optimizer which maximize the sum of the selected lineups FPs based on these predictions, while considering the restrictions imposed by the betting platforms (i.e., budget, player position, number of players etc.).

Additionally, Advanced Features are shown in (Appendix B) and Basic Features are shown in (Appendix C).

Appendix A: Post Data Cleaning Workflow


Advanced Features Dataset
Pre-Process
Ten Seasons Advanced Features Datasets
Last Seasons Advanced Features Datasets
Per Player Dataset
Per Player Dataset
MODELING
Compare Models
Tune Best
Tune Second Best
Tune Third Best
Blend Best Models
Pick Best
Finalize Model
Evaluate Models (Ten Seasons Datasets Used)
Evaluate Models (Last Seasons Datasets Used)
Per Player Predictions (Ten Seasons Datasets Used)
Per Player Predictions (Last Seasons Datasets Used)
Evaluate Predictions,Models (As Average)
Basic Features Dataset
Pre-Process
Ten Seasons Basic Features Datasets
Last Seasons Basic Features Datasets
Per Player Dataset
Per Player Dataset
MODELING
Compare Models
Tune Best
Tune Second Best
Tune Third Best
Blend Best Models
Pick Best
Finalize Model
Evaluate Models (Ten Seasons Datasets Used)
Evaluate Models (Last Seasons Datasets Used)
Per Player Predictions (Ten Seasons Datasets Used)
Per Player Predictions (Last Seasons Datasets Used)
Evaluate Predictions,Models (As Average)
Merge Predictions Advanced, Basic Features (Last Seasons Datasets Used
OPTIMIZER
Setting Restrictions
Lineup Optimization
Eight Player Predicted Lineup

Appendix B: Advanced Features Dataset Glossary

FEATURE	EXPLANATION
SEASON_YEAR	The Season Year
PLAYER_NAME	Player's Name
TEAM_NAME	Team's Name
GAME_DATE	The Date of Match
H/A	Home or Away ('1' for Home, '0' for Away)
NBA_FANTASY_PTS	The Fantasy Points Scored
PLAYOFFS	Playoff Match or not ('1' for Playoff, '0' for Regular Season)
OPPONENT	The Opponent that team/Player faces
LAST_MATCH_OPP_TEAM_OFF_RATING	Opponent's last match Offensive Rating
LAST_MATCH_OPP_TEAM_DEF_RATING	Opponent's last match Defensive Rating
LAST_MATCH_OPP_TEAM_NET_RATING	Opponent's last match Net Rating (Difference of OFF/DEF)
LAST_MATCH_OPP_TEAM_NBA_FANTASY_PTS	Opponent's last match Sum Fantasy Points scored
REST_DAYS	Days brake before last match (over '5' assigned as '5')
LAST_MATCH_OPP_TEAM_OFF_RATING_MOMENTUM	Last Game's Opponent's Offensive Rating difference from Game before last
LAST_MATCH_OPP_TEAM_DEF_RATING_MOMENTUM	Last Game's Opponent's Defensive Rating difference from Game before last
LAST_MATCH_OPP_TEAM_NET_RATING_MOMENTUM	Last Game's Opponent's Net Rating (Difference of OFF/DEF) difference from Game before last
LAST_MATCH_OPP_TEAM_NBA_FANTASY_PTS_MOMENTUM	Last Game's Opponent's Sum Fantasy Points scored difference from Game before last
LAST_MATCHES_3_DAYS_WL_SUM	Last 3 Game's sum of wins
LAST_MATCHES_3_DAYS_DD2_SUM	Last 3 Game's sum of DD2 (Double-Double)
LAST_MATCHES_3_DAYS_TD3_SUM	Last 3 Game's sum of TD3 (Triple-Double)
LAST_MATCHES_3_DAYS_MIN_SUM	Last 3 Game's sum of participation minutes
LAST_MATCH_WL	Last Game's Result (Win '1' or Lose '0')
LAST_MATCH_DD2	Last Game made Double-Double or not
LAST_MATCH_TD3	Last Game made Triple-Double or not
LAST_MATCH_MIN	Last Game's minutes participated

LAST_MATCH_PLAYOFFS	Last Game's type of Game(Playoffs '1' or Regular Season '0')
LAST_MATCH_TEAM_MIN	Last Game's Team's minutes played
LAST_MATCH_TEAM_FGM	Last Game's Team's Field Goals Made
LAST_MATCH_TEAM_FGA	Last Game's Team's Field Goals Attempted
LAST_MATCH_TEAM_FG_PCT	Last Game's Team's Field Goals Percentage
LAST_MATCH_TEAM_FG3M	Last Game's Team's 3-Point Field Goal Made
LAST_MATCH_TEAM_FG3A	Last Game's Team's 3-Point Field Goal Attempted
LAST_MATCH_TEAM_FG3_PCT	Last Game's Team's 3-Point Field Goal Percentage
LAST_MATCH_TEAM_FTM	Last Game's Team's Free Throws Made
LAST_MATCH_TEAM_FTA	Last Game's Team's Free Throws Attempted
LAST_MATCH_TEAM_FT_PCT	Last Game's Team's Free Throws Percentage
LAST_MATCH_TEAM_OREB	Last Game's Team's Offensive Rebound
LAST_MATCH_TEAM_DREB	Last Game's Team's Defensive Rebound
LAST_MATCH_TEAM_REB	Last Game's Team's Rebound
LAST_MATCH_TEAM_AST	Last Game's Team's Assists
LAST_MATCH_TEAM_TOV	Last Game's Team's Turnovers

LAST_MATCH_TEAM_STL	Last Game's Team's Steals
LAST_MATCH_TEAM_BLK	Last Game's Team's Blocks
LAST_MATCH_TEAM_BLKA	Last Game's Team's Blocks Against
LAST_MATCH_TEAM_PF	Last Game's Team's Personal Foul
LAST_MATCH_TEAM_PFD	Last Game's Team's Personal Fouls Drawn
LAST_MATCH_TEAM_PTS	Last Game's Team's Points Scored
LAST_MATCH_TEAM_E_OFF_RATING	Last Game's Team's Estimated Offensive Rating
LAST_MATCH_TEAM_OFF_RATING	Last Game's Team's Offensive Rating
LAST_MATCH_TEAM_E_DEF_RATING	Last Game's Team's Estimated Defensive Rating
LAST_MATCH_TEAM_DEF_RATING	Last Game's Team's Defensive Rating
LAST_MATCH_TEAM_AST_PCT	Last Game's Team's Assist Percentage
LAST_MATCH_TEAM_AST_TO	Last Game's Team's Assist to Turnover
LAST_MATCH_TEAM_AST_RATIO	Last Game's Team's Assist Ratio
LAST_MATCH_TEAM_OREB_PCT	Last Game's Team's Defensive Rebound's Percentage
LAST_MATCH_TEAM_DREB_PCT	Last Game's Team's Offensive Rebound's Percentage
LAST_MATCH_TEAM_REB_PCT	Last Game's Team's Rebound's Percentage

LAST_MATCH_TEAM_TM_TOV_PCT	Last Game's Team's Turnover Percentage
LAST_MATCH_TEAM_EFG_PCT	Last Game's Team's Effective Field Goal Percentage
LAST_MATCH_TEAM_TS_PCT	Last Game's Team's True Shooting Percentage
LAST_MATCH_TEAM_E_PACE	Last Game's Team's Estimated Pace
LAST_MATCH_TEAM_PACE	Last Game's Team's Pace
LAST_MATCH_TEAM_PACE_PER40	Last Game's Team's Pace per 40 Minutes
LAST_MATCH_TEAM_POSS	Last Game's Team's Possessions
LAST_MATCH_TEAM_PIE	Last Game's Team's Impact Estimate
LAST_MATCH_TEAM_PTS_OFF_TOV	Last Game's Team's Points off Turnovers
LAST_MATCH_TEAM_PTS_2ND_CHANCE	Last Game's Team's 2nd Chance Points
LAST_MATCH_TEAM_PTS_FB	Last Game's Team's Fast Break Points
LAST_MATCH_TEAM_PTS_PAINT	Last Game's Team's Paint Touch Points (The number of points scored by a player or team on touches in the paint)
LAST_MATCH_TEAM_PCT_FGA_2PT	Last Game's Team's Percent of Field Goals Attempted (2 Pointers)
LAST_MATCH_TEAM_PCT_FGA_3PT	Last Game's Team's Percent of Field Goals Attempted (3 Pointers)
LAST_MATCH_TEAM_PCT_PTS_2PT	Last Game's Team's Percent of Points Made (2 Pointers)
LAST_MATCH_TEAM_PCT_PTS_2PT_	Last Game's Team's Percent of Points (2-Point Field Goals: Mid Range)
MR	
LAST_MATCH_TEAM_PCT_PTS_3PT	Last Game's Team's Percent of Points (3-Point Field Goals)
LAST_MATCH_TEAM_PCT_PTS_FB	Last Game's Team's Percent of Points (Fast Break Points)
LAST_MATCH_TEAM_PCT_PTS_FT	Last Game's Team's Percent of Points (Free Throws)
LAST_MATCH_TEAM_PCT_PTS_OFF_TOV	Last Game's Team's Percent of Points (Off Turnovers)
LAST_MATCH_TEAM_PCT_PTS_PAINT	Last Game's Team's Percent of Points (Points in the Paint)
LAST_MATCH_TEAM_PCT_AST_2PM	Last Game's Team's Percent of Assists 2 Point Field Goals Made
LAST_MATCH_TEAM_PCT_UAST_2PM	Last Game's Team's Percent of Unassisted 2 Point Field Goals Made
LAST_MATCH_TEAM_PCT_AST_3PM	Last Game's Team's of Assists 3 Point Field Goals Made
LAST_MATCH_TEAM_PCT_UAST_3PM	Last Game's Team's of Unassisted 3 Point Field Goals Made
LAST_MATCH_TEAM_PCT_AST_FGM	Last Game's Team's of Assists Field Goals Made
LAST_MATCH_TEAM_PCT_UAST_FGM	Last Game's Team's of Unassisted Field Goals Made
LAST_MATCH_TEAM_FTA_RATE	Last Game's Team's Free Throw Attempt Rate
LAST_MATCH_OPPONENT	Last Game's Opponent
LAST_MATCH_FGM	Last Game's Field Goals Made
LAST_MATCH_FGA	Last Game's Field Goals Attempted

LAST_MATCH_FG_PCT	Last Game's Percent Field Goals
LAST_MATCH_FG3M	Last Game's Field Goals Made (3 Pointers)
LAST_MATCH_FG3A	Last Game's Field Goals Attempted (3 Pointers)
LAST_MATCH_FG3_PCT	Last Game's Field Goals Percentage
LAST_MATCH_FTM	Last Game's Free Throws Made
LAST_MATCH_FTA	Last Game's Free Throws Attempted
LAST_MATCH_FT_PCT	Last Game's Free Throws Percentage
LAST_MATCH_OREB	Last Game's Offensive Rebounds
LAST_MATCH_DREB	Last Game's Defensive Rebounds
LAST_MATCH_REB	Last Game's Rebounds
LAST_MATCH_AST	Last Game's Assists
LAST_MATCH_TOV	Last Game's Turnovers
LAST_MATCH_STL	Last Game's Steals
LAST_MATCH_BLK	Last Game's Blocks
LAST_MATCH_BLKA	Last Game's Blocks Against
LAST_MATCH_PF	Last Game's Personal Fouls
LAST_MATCH_PFD	Last Game's Personal Fouls Drawn

LAST_MATCH_PTS	Last Game's Points
LAST_MATCH_NBA_FANTASY_PTS	Last Game's Fantasy Points
LAST_MATCH_E_OFF_RATING	Last Game's Estimated Offensive Rating
LAST_MATCH_OFF_RATING	Last Game's Offensive Rating
LAST_MATCH_sp_work_OFF_RATING	Last Game's Sp Work Last Game's Offensive Rating
LAST_MATCH_E_DEF_RATING	Last Game's Estimated Defensive Rating
LAST_MATCH_DEF_RATING	Last Game's Defensive Rating
LAST_MATCH_sp_work_DEF_RATING	Last Game's Sp Work Last Game's Defensive Rating
LAST_MATCH_AST_PCT	Last Game's Assists Percentage
LAST_MATCH_AST_TO	Last Game's Assist to Turnover Ratio
LAST_MATCH_AST_RATIO	Last Game's Assist Ratio
LAST_MATCH_OREB_PCT	Last Game's Offensive Rebound Rating
LAST_MATCH_DREB_PCT	Last Game's Defensive Rebound Rating
LAST_MATCH_REB_PCT	Last Game's Rebound Percentage
LAST_MATCH_TM_TOV_PCT	Last Game's Team's Turnover Percentage
LAST_MATCH_E_TOV_PCT	Last Game's Estimated Turnover Percentage

LAST_MATCH_EFG_PCT	Last Game's Effective Field Goal Percentage
LAST_MATCH_TS_PCT	Last Game's True Shooting Percentage
LAST_MATCH_USG_PCT	Last Game's Usage Percentage (Ratio of plays used to possessions)
LAST_MATCH_E_USG_PCT	Last Game's Estimated Usage Percentage
LAST_MATCH_E_PACE	Last Game's Estimated Pace
LAST_MATCH_PACE	Last Game's Pace
LAST_MATCH_PACE_PER40	Last Game's Pace per 40 Minutes
LAST_MATCH_sp_work_PACE	Last Game's Sp Work Pace
LAST_MATCH_PIE	Last Game's Player Impact Estimate
LAST_MATCH_POSS	Last Game's Possessions
LAST_MATCH_FGM_PG	Last Game's Field Goals Made Per Game
LAST_MATCH_FGA_PG	Last Game's Field Goals Attempted Per Game
LAST_MATCH_PTS_OFF_TOV	Last Game's Points Off Turnovers
LAST_MATCH_PTS_2ND_CHANCE	Last Game's Second Chance Points
LAST_MATCH_PTS_FB	Last Game's Fast Break Points
LAST_MATCH_PTS_PAINT	Last Game's Points in the Paint

LAST_MATCH_OPP_PTS_OFF_TOV	Last Game's Opponent Points Off Turnovers
LAST_MATCH_OPP_PTS_2ND_CHANCE	Last Game's Opponent Second Chance Points
LAST_MATCH_OPP_PTS_FB	Last Game's Opponent Fast Break Points
LAST_MATCH_OPP_PTS_PAINT	Last Game's Opponent Points in the Paint
LAST_MATCH_PCT_FGA_2PT	Last Game's Percentage Of Field Goals Attempted that are two-point field goal attempts
LAST_MATCH_PCT_FGA_3PT	Last Game's Percentage Of Field Goals Attempted that are three-point field goal attempts
LAST_MATCH_PCT_PTS_2PT	Last Game's Percentage Of Points that are from two-point field goals
LAST_MATCH_PCT_PTS_2PT_MR	Last Game's Percentage Of Points that are from two-point field goals from mid-range field goals
LAST_MATCH_PCT_PTS_3PT	Last Game's Percentage Of Points that are from three-point field goals
LAST_MATCH_PCT_PTS_FB	Last Game's Percentage Of Points that are from fast break opportunities
LAST_MATCH_PCT_PTS_FT	Last Game's Percentage Of Points that are from free throws
LAST_MATCH_PCT_PTS_OFF_TOV	Last Game's Percentage Of Points that are off turnovers
LAST_MATCH_PCT_PTS_PAINT	Last Game's Percentage Of Points that are from the paint
LAST_MATCH_PCT_AST_2PM	Last Game's Percentage Of two-point field goals made that are assisted
LAST_MATCH_PCT_UAST_2PM	Last Game's Percentage Of two-point field goals made that are unassisted
LAST_MATCH_PCT_AST_3PM	Last Game's Percentage Of three-point field goals made that are assisted

LAST_MATCH_PCT_UAST_3PM	Last Game's Percentage Of three-point field goals made that are unassisted
LAST_MATCH_PCT_AST_FGM	Last Game's Percentage Of field goals made that are assisted
LAST_MATCH_PCT_UAST_FGM	Last Game's Percentage Of field goals made that are unassisted
LAST_MATCH_PCT_FGM	Last Game's Percentage Of Field Goal Made while on court
LAST_MATCH_PCT_FGA	Last Game's Percentage Of Field Goal Attempts while on court
LAST_MATCH_PCT_FG3M	Last Game's Percentage Of three-point field goal made while on court
LAST_MATCH_PCT_FG3A	Last Game's Percentage Of three-point field goal attempts while on court
LAST_MATCH_PCT_FTM	Last Game's Percentage Of free throws made while on court
LAST_MATCH_PCT_FTA	Last Game's Percentage Of free throw attempts while on court
LAST_MATCH_PCT_OREB	Last Game's Percentage Of Offensive rebounds while on court
LAST_MATCH_PCT_DREB	Last Game's Percentage Of defensive rebounds while on court
LAST_MATCH_PCT_REB	Last Game's Percentage Of Rebounds while on court
LAST_MATCH_PCT_AST	Last Game's Percentage Of Assists while on court
LAST_MATCH_PCT_TOV	Last Game's Percentage Of Turnovers while on court
LAST_MATCH_PCT_STL	Last Game's Percentage Of Steals while on court
LAST_MATCH_PCT_BLK	Last Game's Percentage Of Blocks while on court

LAST_MATCH_PCT_BLKA	Last Game's Percentage Of Blocks Attempted while on court
LAST_MATCH_PCT_PF	Last Game's Percentage Of Personal Fouls while on court
LAST_MATCH_PCT_PFD	Last Game's Percentage Of Personal Fouls Drawn while on court
LAST_MATCH_PCT_PTS	Last Game's Percentage Of Points while on court
LAST_MATCH_MIN_MOMENTUM	Last Game's minutes participated Difference From Game Before Last
LAST_MATCH_FGM_MOMENTUM	Last Game's Field Goals Made Difference From Game Before Last
LAST_MATCH_FGA_MOMENTUM	Last Game's Field Goals Attempted Difference From Game Before Last
LAST_MATCH_FG_PCT_MOMENTUM	Last Game's Percent Field Goals Difference From Game Before Last
LAST_MATCH_FG3M_MOMENTUM	Last Game's Field Goals Made (3 Pointers) Difference From Game Before Last
LAST_MATCH_FG3A_MOMENTUM	Last Game's Field Goals Attempted (3 Pointers) Difference From Game Before Last
LAST_MATCH_FG3_PCT_MOMENTUM	Last Game's Field Goals Percentage Difference From Game Before Last
LAST_MATCH_FTM_MOMENTUM	Last Game's Free Throws Made Difference From Game Before Last
LAST_MATCH_FTA_MOMENTUM	Last Game's Free Throws Attempted Difference From Game Before Last
LAST_MATCH_FT_PCT_MOMENTUM	Last Game's Free Throws Percentage Difference From Game Before Last
LAST_MATCH_OREB_MOMENTUM	Last Game's Offensive Rebounds Difference From Game Before Last
LAST_MATCH_DREB_MOMENTUM	Last Game's Defensive Rebounds Difference From Game Before Last

LAST_MATCH_REB_MOMENTUM	Last Game's Rebounds Difference From Game Before Last
LAST_MATCH_AST_MOMENTUM	Last Game's Assists Difference From Game Before Last
LAST_MATCH_TOV_MOMENTUM	Last Game's Turnovers Difference From Game Before Last
LAST_MATCH_STL_MOMENTUM	Last Game's Steals Difference From Game Before Last
LAST_MATCH_BLK_MOMENTUM	Last Game's Blocks Difference From Game Before Last
LAST_MATCH_BLKA_MOMENTUM	Last Game's Blocks Against Difference From Game Before Last
LAST_MATCH_PF_MOMENTUM	Last Game's Personal Fouls Difference From Game Before Last
LAST_MATCH_PFD_MOMENTUM	Last Game's Personal Fouls Drawn Difference From Game Before Last
LAST_MATCH_PTS_MOMENTUM	Last Game's Points Difference From Game Before Last
LAST_MATCH_NBA_FANTASY_PTS_MOMENTUM	Last Game's Fantasy Points Difference From Game Before Last
LAST_MATCH_E_OFF_RATING_MOMENTUM	Last Game's Estimated Offensive Rating Difference From Game Before Last
LAST_MATCH_OFF_RATING_MOMENTUM	Last Game's Offensive Rating Difference From Game Before Last
LAST_MATCH_sp_work_OFF_RATING_MOMENTUM	Last Game's Sp Work Last Game's Offensive Rating Difference From Game Before Last
LAST_MATCH_E_DEF_RATING_MOMENTUM	Last Game's Estimated Defensive Rating Difference From Game Before Last
LAST_MATCH_DEF_RATING_MOMENTUM	Last Game's Defensive Rating Difference From Game Before Last

LAST_MATCH_sp_work_DEF_RATING_MOMENTUM	Last Game's Sp Work Last Game's Defensive Rating Difference From Game Before Last
LAST_MATCH_AST_PCT_MOMENTUM	Last Game's Assists Percentage Difference From Game Before Last
LAST_MATCH_AST_TO_MOMENTUM	Last Game's Assist to Turnover Ratio Difference From Game Before Last
LAST_MATCH_AST_RATIO_MOMENTUM	Last Game's Assist Ratio Difference From Game Before Last
LAST_MATCH_OREB_PCT_MOMENTUM	Last Game's Offensive Rebound Rating Difference From Game Before Last
LAST_MATCH_DREB_PCT_MOMENTUM	Last Game's Defensive Rebound Rating Difference From Game Before Last
LAST_MATCH_REB_PCT_MOMENTUM	Last Game's Rebound Percentage Difference From Game Before Last
LAST_MATCH_TM_TOV_PCT_MOMENTUM	Last Game's Team's Turnover Percentage Difference From Game Before Last
LAST_MATCH_E_TOV_PCT_MOMENTUM	Last Game's Estimated Turnover Percentage Difference From Game Before Last
LAST_MATCH_EFG_PCT_MOMENTUM	Last Game's Effective Field Goal Percentage Difference From Game Before Last
LAST_MATCH_TS_PCT_MOMENTUM	Last Game's True Shooting Percentage Difference From Game Before Last
LAST_MATCH_USG_PCT_MOMENTUM	Last Game's Usage Percentage (Ratio of plays used to possessions) Difference From Game Before Last
LAST_MATCH_E_USG_PCT_MOMENTUM	Last Game's Estimated Usage Percentage Difference From Game Before Last
LAST_MATCH_E_PACE_MOMENTUM	Last Game's Estimated Pace Difference From Game Before Last

LAST_MATCH_PACE_MOMENTUM	Last Game's Pace Difference From Game Before Last
LAST_MATCH_PACE_PER40_MOMENTUM	Last Game's Pace per 40 Minutes Difference From Game Before Last
LAST_MATCH_sp_work_PACE_MOMENTUM	Last Game's Sp Work Pace Difference From Game Before Last
LAST_MATCH_PIE_MOMENTUM	Last Game's Player Impact Estimate Difference From Game Before Last
LAST_MATCH_POSS_MOMENTUM	Last Game's Possessions Difference From Game Before Last
LAST_MATCH_FGM_PG_MOMENTUM	Last Game's Field Goals Made Per Game Difference From Game Before Last
LAST_MATCH_FGA_PG_MOMENTUM	Last Game's Field Goals Attempted Per Game Difference From Game Before Last
LAST_MATCH_PTS_OFF_TOV_MOMENTUM	Last Game's Points Off Turnovers Difference From Game Before Last
LAST_MATCH_PTS_2ND_CHANCE_MOMENTUM	Last Game's Second Chance Points Difference From Game Before Last
LAST_MATCH_PTS_FB_MOMENTUM	Last Game's Fast Break Points Difference From Game Before Last
LAST_MATCH_PTS_PAINT_MOMENTUM	Last Game's Points in the Paint Difference From Game Before Last
LAST_MATCH_OPP_PTS_OFF_TOV_MOMENTUM	Last Game's Opponent Points Off Turnovers Difference From Game Before Last
LAST_MATCH_OPP_PTS_2ND_CHANCE_MOMENTUM	Last Game's Opponent Second Chance Points Difference From Game Before Last
LAST_MATCH_OPP_PTS_FB_MOMENTUM	Last Game's Opponent Fast Break Points Difference From Game Before Last

LAST_MATCH_OPP_PTS_PAINT_MOMENTUM	Last Game's Opponent Points in the Paint Difference From Game Before Last
LAST_MATCH_PCT_FGA_2PT_MOMENTUM	Last Game's Percentage Of Field Goals Attempted that are two-point field goal attempts Difference From Game Before Last
LAST_MATCH_PCT_FGA_3PT_MOMENTUM	Last Game's Percentage Of Field Goals Attempted that are three-point field goal attempts Difference From Game Before Last
LAST_MATCH_PCT_PTS_2PT_MOMENTUM	Last Game's Percentage Of Points that are from two-point field goals Difference From Game Before Last
LAST_MATCH_PCT_PTS_2PT_MR_MOMENTUM	Last Game's Percentage Of Points that are from two-point field goals from mid-range field goals Difference From Game Before Last
LAST_MATCH_PCT_PTS_3PT_MOMENTUM	Last Game's Percentage Of Points that are from three-point field goals Difference From Game Before Last
LAST_MATCH_PCT_PTS_FB_MOMENTUM	Last Game's Percentage Of Points that are from fast break opportunities Difference From Game Before Last
LAST_MATCH_PCT_PTS_FT_MOMENTUM	Last Game's Percentage Of Points that are from free throws Difference From Game Before Last
LAST_MATCH_PCT_PTS_OFF_TOV_MOMENTUM	Last Game's Percentage Of Points that are off turnovers Difference From Game Before Last
LAST_MATCH_PCT_PTS_PAINT_MOMENTUM	Last Game's Percentage Of Points that are from the paint Difference From Game Before Last
LAST_MATCH_PCT_AST_2PM_MOMENTUM	Last Game's Percentage Of two-point field goals made that are assisted Difference From Game Before Last
LAST_MATCH_PCT_UAST_2PM_MOMENTUM	Last Game's Percentage Of two-point field goals made that are unassisted Difference From Game Before Last
LAST_MATCH_PCT_AST_3PM_MOMENTUM	Last Game's Percentage Of three-point field goals made that are assisted Difference From Game Before Last

LAST_MATCH_PCT_UAST_3PM_MOMENTUM	Last Game's Percentage Of three-point field goals made that are unassisted Difference From Game Before Last
LAST_MATCH_PCT_AST_FGM_MOMENTUM	Last Game's Percentage Of field goals made that are assisted Difference From Game Before Last
LAST_MATCH_PCT_UAST_FGM_MOMENTUM	Last Game's Percentage Of field goals made that are unassisted Difference From Game Before Last
LAST_MATCH_PCT_FGM_MOMENTUM	Last Game's Percentage Of Field Goal Made while on court Difference From Game Before Last
LAST_MATCH_PCT_FGA_MOMENTUM	Last Game's Percentage Of Field Goal Attempts while on court Difference From Game Before Last
LAST_MATCH_PCT_FG3M_MOMENTUM	Last Game's Percentage Of three-point field goal made while on court Difference From Game Before Last
LAST_MATCH_PCT_FG3A_MOMENTUM	Last Game's Percentage Of three-point field goal attempts while on court Difference From Game Before Last
LAST_MATCH_PCT_FTM_MOMENTUM	Last Game's Percentage Of free throws made while on court Difference From Game Before Last
LAST_MATCH_PCT_FTA_MOMENTUM	Last Game's Percentage Of free throw attempts while on court Difference From Game Before Last
LAST_MATCH_PCT_OREB_MOMENTUM	Last Game's Percentage Of Offensive rebounds while on court Difference From Game Before Last
LAST_MATCH_PCT_DREB_MOMENTUM	Last Game's Percentage Of defensive rebounds while on court Difference From Game Before Last
LAST_MATCH_PCT_REB_MOMENTUM	Last Game's Percentage Of Rebounds while on court Difference From Game Before Last
LAST_MATCH_PCT_AST_MOMENTUM	Last Game's Percentage Of Assists while on court Difference From Game Before Last
LAST_MATCH_PCT_TOV_MOMENTUM	Last Game's Percentage Of Turnovers while on court Difference From Game Before Last

LAST_MATCH_PCT_STL_MOMENTUM	Last Game's Percentage Of Steals while on court Difference From Game Before Last
LAST_MATCH_PCT_BLK_MOMENTUM	Last Game's Percentage Of Blocks while on court Difference From Game Before Last
LAST_MATCH_PCT_BLKA_MOMENTUM	Last Game's Percentage Of Blocks Attempted while on court Difference From Game Before Last
LAST_MATCH_PCT_PF_MOMENTUM	Last Game's Percentage Of Personal Fouls while on court Difference From Game Before Last
LAST_MATCH_PCT_PFD_MOMENTUM	Last Game's Percentage Of Personal Fouls Drawn while on court Difference From Game Before Last
LAST_MATCH_PCT_PTS_MOMENTUM	Last Game's Percentage Of Points while on court Difference From Game Before Last
LAST_MATCHES_3_DAYS_NBA_FANTASY_PTS_AVG	Last Three Game's Fantasy Points Average
LAST_MATCHES_5_DAYS_NBA_FANTASY_PTS_AVG	Last Five Game's Fantasy Points Average
LAST_MATCHES_7_DAYS_NBA_FANTASY_PTS_AVG	Last Seven Game's Fantasy Points Average
LAST_MATCHES_10_DAYS_NBA_FANTASY_PTS_AVG	Last Ten Game's Fantasy Points Average
LAST_MATCH_NBA_FANTASY_PTS_Smoothed_MOMENTUM	Last Game's Fantasy Points Smoothed Difference From Game Before Last
LAST_MATCH_Anomaly_MOMENTUM	Last Game's Anomaly Detected Difference From Game Before Last
LAST_MATCH_NBA_FANTASY_PTS_Smoothed	Last Game's Fantasy Points Smoothed
LAST_MATCH_Anomaly	Last Game's Anomaly Detected

Appendix C: Basic Features Dataset Glossary

FEATURE	EXPLANATION
SEASON_YEAR	The Season Year
PLAYER_NAME	Player's Name
GAME_DATE	The Date of Match
H/A	Home or Away ('1' for Home, '0' for Away)
NBA_FANTASY_PTS	The Fantasy Points Scored
TEAM_ABBREVIATION	Team's Name
OPPONENT	The Opponent that team/Player faces
LAST_MATCH_OPP_TEAM_NBA_FANTASY_PTS	Opponent's last match Sum Fantasy Points scored
LAST_MATCH_OPP_TEAM_OFF_RATING	Opponent's last match Offensive Rating
LAST_MATCH_OPP_TEAM_DEF_RATING	Opponent's last match Defensive Rating
LAST_MATCH_OPP_TEAM_NET_RATING	Opponent's last match Net Rating (Difference of OFF/DEF)
PLAYOFFS	Playoff Match or not ('1' for Playoff, '0' for Regular Season)
REST_DAYS	Days brake before last match (over '5' assigned as '5')
LAST_MATCH_OPP_TEAM_NBA_FANTASY_PTS_MOMENTUM	Last Game's Opponent's Sum Fantasy Points scored difference from Game before last
LAST_MATCH_OPP_TEAM_OFF_RATING_MOMENTUM	Last Game's Opponent's Offensive Rating difference from Game before last
LAST_MATCH_OPP_TEAM_DEF_RATING_MOMENTUM	Last Game's Opponent's Defensive Rating difference from Game before last
LAST_MATCH_OPP_TEAM_NET_RATING_MOMENTUM	Last Game's Opponent's Net Rating (Difference of OFF/DEF) difference from Game before last
LAST_MATCH_MIN	Last Game's minutes participated
LAST_MATCH_PTS	Last Game's Points
LAST_MATCH_FG3M	Last Game's Field Goals Made (3 Pointers)
LAST_MATCH_REB	Last Game's Rebounds
LAST_MATCH_AST	Last Game's Assists
LAST_MATCH_STL	Last Game's Steals
LAST_MATCH_BLK	Last Game's Blocks

LAST_MATCH_TOV	Last Game's Turnovers
LAST_MATCH_DD2	Last Game made Double-Double or not
LAST_MATCH_TD3	Last Game made Triple-Double or not
LAST_MATCH_NET_RATING	Last Game's Net Rating (Difference of OFF/DEF)
LAST_MATCH_USG_PCT	Last Game's Usage Percentage (Ratio of plays used to possessions)
LAST_MATCH_PIE	Last Game's Player Impact Estimate
LAST_MATCH_WL	Last Game's Result (Win '1' or Lose '0')
LAST_MATCH_TEAM_OFF_RATING	Last Game's Team's Offensive Rating
LAST_MATCH_TEAM_DEF_RATING	Last Game's Team's Defensive Rating
LAST_MATCH_TEAM_NBA_FANTASY_PTS	Last Game's Team's Fantasy Points
LAST_MATCH_MIN_MOMENTUM	Last Game's minutes participated Difference From Game Before Last
LAST_MATCH_PTS_MOMENTUM	Last Game's Points Difference From Game Before Last
LAST_MATCH_FG3M_MOMENTUM	Last Game's Field Goals Made (3 Pointers) Difference From Game Before Last
LAST_MATCH_REB_MOMENTUM	Last Game's Rebounds Difference From Game Before Last

LAST_MATCH_AST_MOMENTUM	Last Game's Assists Difference From Game Before Last
LAST_MATCH_STL_MOMENTUM	Last Game's Steals Difference From Game Before Last
LAST_MATCH_BLK_MOMENTUM	Last Game's Blocks Difference From Game Before Last
LAST_MATCH_TOV_MOMENTUM	Last Game's Turnovers Difference From Game Before Last
LAST_MATCH_DD2_MOMENTUM	Last Game's Double-Double Difference From Game Before Last
LAST_MATCH_TD3_MOMENTUM	Last Game's Triple-Double Difference From Game Before Last
LAST_MATCH_NET_RATING_MOMENTUM	Last Game's Net Rating (Difference of OFF/DEF) Difference From Game Before Last
LAST_MATCH_USG_PCT_MOMENTUM	Last Game's Usage Percentage (Ratio of plays used to possessions) Difference From Game Before Last
LAST_MATCH_PIE_MOMENTUM	Last Game's Player Impact Estimate Difference From Game Before Last
LAST_MATCH_OPPONENT_MOMENTUM	Last Game's Opponent's Difference From Game Before Last
LAST_MATCH_NBA_FANTASY_PTS_MOMENTUM	Last Game's Fantasy Points Difference From Game Before Last
LAST_MATCH_TEAM_NBA_FANTASY_PTS_MOMENT	Last Game's Team's Fantasy Points Difference From

UM	Game Before Last
LAST_MATCH_WL_MOMENTUM	Last Game's Win or Lose Difference From Game Before Last
LAST_MATCH_REST_DAYS_MOMENTUM	Last Game's Rest Days Difference From Game Before Last
LAST_MATCH_PLAYOFFS_MOMENTUM	Last Game's Playoffs Difference From Game Before Last
LAST_MATCH_H/A_MOMENTUM	Last Game's Home or Away Difference From Game Before Last
LAST_MATCH_TEAM_OFF_RATING_MOMENTUM	Last Game's Team's Offensive Rating Difference From Game Before Last
LAST_MATCH_TEAM_DEF_RATING_MOMENTUM	Last Game's Team's Defensive Rating Difference From Game Before Last
LAST_MATCH_NBA_FANTASY_PTS	Last Game's Fantasy Points
LAST_MATCH_OPPONENT	Last Game's Opponent
LAST_MATCHES_3_DAYS_NBA_FANTASY_PTS_AVG	Last Three Game's Fantasy Points Average
LAST_MATCHES_5_DAYS_NBA_FANTASY_PTS_AVG	Last Five Game's Fantasy Points Average
LAST_MATCHES_7_DAYS_NBA_FANTASY_PTS_AVG	Last Seven Game's Fantasy Points Average
LAST_MATCHES_10_DAYS_NBA_FANTASY_PTS_AVG	Last Ten Game's Fantasy Points Average
LAST_MATCH_NBA_FANTASY_PTS_Smoothed	Last Game's Fantasy Points Smoothed
LAST_MATCH_Anomaly	Last Game's Anomaly Detected
LAST_MATCH_Anomaly_MOMENTUM	Last Game's Anomaly Detected Difference From Game Before Last
LAST_MATCH_NBA_FANTASY_PTS_Smoothed_MOMENTUM	Last Game's Fantasy Points Smoothed Difference From Game Before Last