MSCI 446

Project Report

Bill Sheng - 20720680

Vivian Li - 20713933

December 9, 2020

Table of Contents

Abstract

Every June, the basketball world tunes into the NBA Draft, an annual event hosted by the National Basketball Association (NBA) in which the league's teams draft players from a pool of young and promising talent. Using the NBA Draft to select the "right" players is an essential skill amongst all successful NBA teams, and is deservingly so a very difficult skill to master. It allows successful teams to invest in their future and poor-performing teams to rebuild. It is known that an athlete's background and career prior to playing professionally are the greatest determinants of their future careers in the game. A team's performance every season is heavily dependent on this influx of new talent as well, placing pressure on the league to ensure that the best talent from all corners of the world are being recruited. As such, we present a multiple linear regression model to predict player standings within draft classes. Additionally, we have implemented a clustering algorithm to identify the skills that make up the top players in the league today, as well as identify where efforts should be focused to garner further international interest in the sport.

Introduction

Every year in the month of June, the eyes of the world turn to the National Basketball Association (NBA) and the NBA Draft. Here, upcoming basketball athletes from around the world are given a chance to compete for a spot in one of 60 teams. Athletes are judged by their college performances and stats, but ultimately make the draft and are selected by teams depending on individual requirements and the league's landscape. This introduces a lot of ambiguity; players need to understand what variables can improve their standings, while teams need a recruit who will fit their needs. Furthermore, the NBA needs to ensure the quality and diversity of players in each draft are not compromised, as it ultimately affects seasonal outcome and the game for years to come. To remedy these issues, this study will look to find key skills players should focus on for long term success, help teams identify their ideal candidates in the annual drafts, and assist the NBA in garnering greater international interest and reach. Doing so will set teams and athletes up for long term success, as well as generate interest in other areas of the world to introduce greater talent and continually advance the sport.

When selecting a player from the draft, teams tend to pick those who fit a specific set of skills they are looking for. However, it is not uncommon for a team to have lost their first pick to another as it all boils down to the order they've been given. In these scenarios, teams should be able to identify their next ideal candidates from the remaining pool of athletes. We address this issue through a linear regression model of historical draft picks and college stats to predict an athlete's standings and value in their draft classes. Based on the limited data available to teams during drafts, we hypothesize that college career stats are the main determinant in a player's standing and value in the draft. As these analyses are purely numerical, they do not account for perceived player potential or compatibility with a team. Despite these playing an important role in who teams decide on, they cannot be statistically and accurately predicted and thus have been omitted from the study. However, understanding how a potential pick stacks up against others on the court will help teams to better identify their alternatives and make better decisions for a better season.

On the other hand, players want to understand what skills will keep them competitive within their draft classes and in their future careers. The issue that arises is every year's draft is different; team needs and competition shift, resulting in athletic adaptations to match the ongoing evolution of the sport. Early years of the NBA valued taller athletes as they could travel the court faster and had a significant reach and size advantage over others. This was further enforced as dunking became a sought after skill as well. Then in the early 2000's, point guard Allen Iverson proved that height didn't necessarily affect performance. In a position that required speed and agility, he outperformed his taller peers and showed the advantages in having smaller players on the court. This led to others like Stephen Curry turning games with their three-pointers, which resulted in an increase in demand for better rebounders on the opposing team. As can be seen, the value placed on certain skills shifts over the years depending on the league's current active players and landscape. To help younger athletes identify these, we perform a clustering of players by their shooting percentages to identify trends over the years about each position. We hypothesize that

players with higher shooting percentages have a higher likelihood of being drafted in the modern era. Additionally, we believe that modern day athletes that play the center position will improve in their three point shooting and free throwing shooting percentages over the course of their careers. Helping to identify these trends will help younger athletes in the draft re-prioritize their training and skills, consequently having more successful long-term NBA careers.

The third business problem we wanted to solve was from the league office's perspective. For the NBA, their product is basketball. This means that in order for the league to grow, they must be able to market and sell their product. Measuring the number of international prospects is advantageous for a number of reasons. It displays how attractive the NBA is outside of North America based on where draft prospects originate from. It also gives the league a good idea of where basketball is growing in popularity. This allows them to make data-driven decisions when it comes to marketing and hosting international events. In 2020 the NBA announced the Basketball Africa League, the first NBA pro-league outside of North America. Analyzing trends regarding international interest can motivate initiatives similar to that of the Basketball Africa League, where the goal is to provide support and increase exposure for players outside of North America [5].

To help teams identify top players that fit their needs, we applied a supervised predictive model using multiple linear regression. The methodology we implemented goes through our dataset and collects the entries/ players for each position type, and re-ranks them in terms of their standings against others in their draft class who play the same position. Then using the college stats from these players, we try to predict their rankings within their classes. We chose to split up players by their positions because teams usually recruit players based on positions that need to be filled. This approach allows us to draw a comparison between each player and their draftmates, and applies the logic of how teams would select their players. The variables we used to implement this model were 14 - 36 from Figure 1. Up until a player is drafted, the primary indicators of their skills are through their college careers and performance. As such we believe utilizing these numbers will help us to more accurately predict their draft standings, as well as better understand the significance each variable plays in it. As our predictor and outcome variable were to be strictly numerical, we chose to only implement a regression model. Other models like logistic regression are primarily used to model binary dependent variables, which could not be applied to this use case.

For our unsupervised tasks, we implemented and exercised a clustering algorithm. The approach was to select subsets of feature variables in groups of three to segment our data into groups of players. From there, aggregations and trends can be analyzed to drive business insights. Association rule mining was not used in any of our unsupervised learning models as it did not make sense for our data set as well as the respective problem we wanted to solve. The dataset we created consists of mostly numerical variables. The reason behind this is because one of our main goals is to quantify how the landscape of the NBA has changed over time. For example, analyzing how the three-point shooting percentage of draft prospects has changed over time. By clustering

players, we are able to classify each cluster, aggregate our cluster data, and analyze trends using numerical plots to provide quantifiable business insights. In comparison, association rule mining identifies entities that frequently appear in a group within a dataset. Though this may be insightful, the rules only reveal relationships between attributes. They provide no data on numerical changes over time. Furthermore, the nature of statistical sports data is optimal for clustering as we can easily calculate the mathematical distances between numerical statistics and cluster means.

We chose to test two sets of feature variables that we believed would provide valuable insights with respect to the business problems we wished to solve. For our first unsupervised task, we use variables 12 and variable 13 in Figure 1. These represent the coordinates of the player's birthplace. Clustering players by their birthplace allow us to analyze the international landscape within the league. This analysis will also hint at the overall interest in the sport of basketball, assuming that regions with high NBA draft picks reveal a high popularity in the sport. For our second unsupervised, we use variables 30, 31, and 32 in Figure 1. These represent the player's shooting percentages. Clustering players by shooting percentages allow us to segment players by offensive skill set, revealing trends about how the game has evolved over time for each position.

The challenge of predictions involving sports leagues and athletic performance is no novel one. For basketball in particular, countless studies have been geared towards understanding the underlying factors behind draft picks through quantifying player skills, the evolution of each position, and the changing landscape of the sport itself. However the reality is that underlying athletic potential and human psychology play an equally if not more important role in how the current and future of this sport will play out. These independent and unpredictable variables are what make these problems so complex, but we argue that our findings provide practical insight for the NBA, its athletes, and its teams.

Related Work

Our topic explores NBA prospects by analyzing how collegiate statistics translate to success in the NBA as well as how clusters of players with different skill sets have changed over time. We utilize a number of standard supervised and unsupervised model algorithms to drive business insight for our problem.

For our supervised problem, we used multiple linear regression to gain insight on how a prospective athletes' college career influences their standings in the draft. Prior work on this topic have implemented similar strategies in an attempt to identify specific statistics for the draft and player potential. Some researchers suggest that assists, steals, and blocks are the greatest determinants of a player's success in the draft [4, 6] . Other researchers like Sailofsky believe that these are the low risk selections, and that one of the most common errors made by NBA decision makers is undermining important skills like rebounding and turnover percentage, while overrating others like win shares, points, blocks, height [17]. These studies have generalized players and not considered their individual talents by position. As such our applications will include all the college stats to predict NBA draft standings, split by position type. Additionally, we hope to identify the defining traits and stats exhibited by each player type.

In other publications done, it is suggested that a player's current stats could predict and influence the future of their careers in the NBA [13]. In one such instance, Parker draws comparisons between existing methodologies that provide this type of insight; Player Efficiency Rating and standard statistical categories, Roland Beech's Rating System, and Win Shares [14]. This study focuses more on the post-draft careers of athletes, but gives insight into what stats contribute to better careers. These findings about the significance of certain stats can help us to alternatively identify a subset of the associated college stats that play into how players consequently stack up in the draft. Using stats that are deemed valuable in current NBA all-stars, we can more accurately apply them to predicting an athlete's value in the draft and in their future NBA careers.

For our unsupervised tasks, we have focused on the demands for NBA players at each position with respect to shooting to discover whether or not the league's shooting trends align with the types of prospects drafted [8, 22]. Past studies have clustered the offensive data of players and teams to reveal patterns on offensive tendencies. This insight allows professional players and teams to adjust their play-calling and training tactics to maximize their offensive efficiency. These studies are catered towards professionals. For our study, we have focused on college players. Rather than helping professional entities improve their skills, we focus on analyzing draft tendencies to determine the evolution of the NBA's offensive playstyle. This allows us to inform prospects on the skills necessary for them to be successful at the professional level.

Another common application domain for past works has been clustering NBA players to identify hybrid positions as well as to analyze clusters that include high-performing players [15,

21, 23]. Our study is similar in that we want to analyze the emergence of the “hybrid” position by clustering players by offensive efficiency. Instead of specifically clustering these positions, we have analyzed the draft trends behind traditional positions to reveal how these positions have evolved over time to become “hybrid”. Our analysis reveals not only who belongs in which cluster, but the growth of these clusters over time.

Data

Data Description

Our main source, "https://www.basketball-reference.com/", offered draft data that went as far back as 1947. We chose to start our data collection in 1990 for a number of reasons. Firstly, the draft format was extremely different in the past. In the present era, there are 2 rounds of 30 picks, summing to a total of 60 picks. This style was set in 1989. Before that, the draft consisted of 10 rounds of around 15 picks per round [19]. Since we wanted to analyze the trends of variables between draft classes, we wanted to keep the draft style consistent. It would be difficult to cluster data and aggregate it by decade if one decade had triple the amount of total draft picks. This would result in our conclusions being skewed and inaccurate. Therefore, we chose to only take into account data from the three decades after 1990 that used the modern draft style. We also found that not all college statistics were always tracked. For example, between 1950 and 1965, assists per game were not recorded on the college level. This was a problem as a number of these statistics were vital for our model. This was another reason why we chose to start our data collection in 1990 as the data collection process was standardized at this time and we would be confident that we would have all the statistics that were necessary for our model to perform adequately.

Our data consisted of 31 years of data from 1990 to 2020. Each year consisted of 60 draft selections. This summed to a total of 1748 rows and 36 columns per row with each row representing one drafted player. Figure 1 explains the details of each variable, including their name, data type, and semantic meaning.

Figure 1: Feature Variables

ID	Feature Variable	Datatype	Description	Sample Value
1	Pick	Integer	When player was picked	1
2	Team	String	Team they were selected to	NOP
3	Player	String	Player Name	Zion Williamson
4	College	String	College the player attended	Duke
6	Draft Year	String	Year they were drafted	2020
7	Height	Double	Player height	6.5
8	Weight	String	Player weight	284lb
9	Date_of_birth	String	When player was born	July 6, 2000
10	Place_of_birth	String	Where player was born	North Carolina
11	Position	String	Position the player plays	Power

12	BP_Latitude	String	Latitude of player birth place	35° 40' 22.67004'' N
13	BP_Longitude	String	Longitude of player birth place	79° 2' 21.45084'' W
14	college_G	Integer	Games Played in College	33
15	college_MP	Integer	Minutes Played in College	990
16	college_FG	Integer	Field Goals in College	296
17	college_FGA	Integer	Field Goal Attempts in College	435
18	college_3P	Integer	3 -Point Field Goals in College	24
19	college_3PA	Integer	3 -Point Attempts in College	71
20	college_FT	Integer	Free Throws in College	130
21	college_FTA	Integer	Free Throw Attempts in College	203
22	college_ORB	Integer	Offensive Rebounds in College	116
23	college_TRB	Integer	Total Rebounds in College	293
24	college_AST	Integer	Assists in College	68
25	college_STL	Integer	Steals in College	70
26	college_BLK	Integer	Blocks in College	59
27	college_TOV	Integer	Turnovers in College	78
28	college_PF	Integer	Personal Fouls in College	68
29	college_PTS	Integer	Points Scored in College	746
30	college_FG%	Double	Field Goal Percentage in College	.680
31	college_3P%	Double	3 Point Percentage in College	.338
32	college_FT%	Double	Free Throw Percentage in College	.640
33	college_MP.1	Double	Minutes Played Per Game in College	30.0
34	college_PTS%	Double	Points Per Game in College	22.6
35	college_TRB.1	Double	Rebounds per Game in College	8.9
36	college_AST.1	Double	Assists per Game in College	2.1

Data Collection

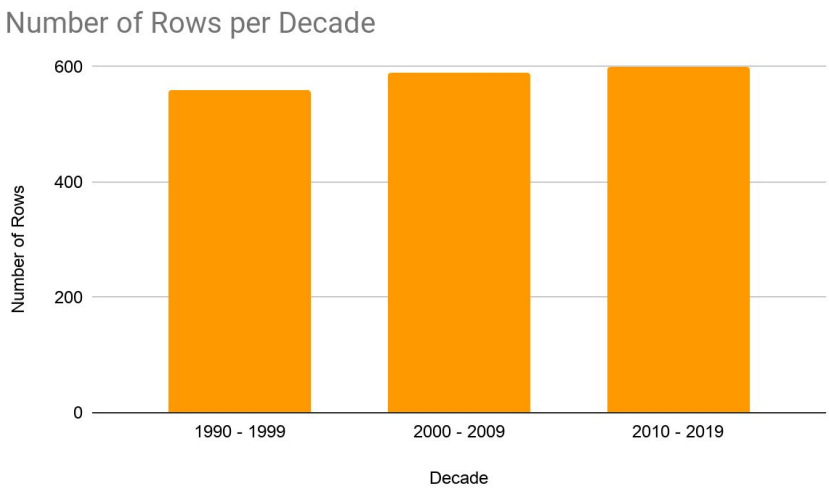
We obtained our data from https://www.basketball-reference.com/, a large North American company that operates a number of sports reference sites that provide a comprehensive database for all things basketball related. The website offers a large breadth of basketball metrics that have been recorded over the course of basketball history. It includes data from a number of leagues, including the NBA and NCAA.

To collect the data, we created a Python script that would loop through all the draft classes of interest. We used the BeautifulSoup Python library to fetch and parse raw HTML dataset tables into a Python-readable format. Specifically, we wanted to fetch data from all the draft years between 1990 and 2020. The NBA Draft dataset (variables 1 to 6 of Figure 1) was stored in separate web pages, following the general URL "https://www.basketball-reference.com/draft/NBA_1990.html" where 1990 can be replaced with the year of interest. For each player drafted, we then fetched the player attributes dataset (variables 7 to 13 of Figure 1) as well as the player college statistics dataset (variables 14 to 36 of Figure 1) from the following URL "https://www.basketball-reference.com/players/c/colemde01.html", where colemde01 was replaced with the ID of the player of interest. This was also done using Python's BeautifulSoup module. One thing to note in the attributes dataset is that the BP_longitude and BP_latitude columns were generated manually by us. To avoid having to parse all birth_place columns in the model itself, we decided to add this process in our data collection process to avoid compute-intensive Dataframe updates in our unsupervised model. For each player, we used the OpenCage API to forward geocode the birth_place column to obtain latitude and longitude coordinates. These coordinate columns were appended to the player attributes dataset. After all of the datasets are fetched and the necessary columns are generated, they are merged into a single row for each player. Once all of the player datasets are merged for each draft year, the data is written to an Excel spreadsheet by grouping all the players by draft year and putting each draft year in a separate sheet.

Missing Data/Data Cleaning

As is often the case with real-world data, we ran into unexpected cases of missing data beyond those discussed above and made decisions to address these roadblocks without affecting the integrity of our model and findings. Figure 2 shows the number of rows available for each decade. We note that the data has become more complete over time. The expected number of rows for each decade is 600. The reason why this number is not always reached for every decade is because of missing profiles. These players could not be included because a subset of the data would be missing and they would be purged from our analysis anyways. We acknowledged that the current data does not significantly deviate from the expected value, therefore it was not a huge problem. To accommodate, we decided that our aggregations should be calculated as percentages of the total players drafted in the decade. This was done for data integrity purposes and prevents missing players from skewing our analysis.

Figure 2: Distribution of Rows per Decade



For the players that had full profiles that could be fetched by our web scraper, some had missing columns. These players were also removed from our analysis. For example in our first unsupervised task, not all players had a birthplace datapoint. Cleaning our data to remove these players yielded a similar distribution to Figure 2. Again, we acknowledged that this would not have a large impact on results if we took a percentage approach when performing aggregations during our unsupervised tasks. Furthermore, it would have minimal effect on the performance of our supervised model as the majority of the data is still present.

There were also more specific scenarios where we needed to clean our data. For example in the second experiment of unsupervised task 2, we wished to cluster NBA centers by their shooting percentages. There were cases of outliers which we had to remove. For example, players who were perfect from the three point line. These players likely made shots by fluke. For example, shooting the ball as time is running out in a quarter. These players would be classified as high percentage three point shooters which is inaccurate as in reality, they are poor three point shooters. These players had to be removed from our data as the data does not accurately reflect their true abilities. This could potentially skew our results and analysis.

Results

Supervised Learning

Purpose

One of the goals of this project is to help teams through the selection process during the drafts. Oftentimes when a team is called to pick their player, their first choice for the position they need has already been selected. To assist these teams in picking the next best athlete suited to their needs, we want to predict player rankings split by cohort type; Bigs, Wings, and Point Guards. The rankings within these groups will be predicted using a multiple linear regression model using college performance and statistics as predictor variables. We present 6 different multiple linear regression models, each a subset of the original dataset we scraped. The first three of these subsets use complete college stats to predict draft rank for each position type. The purpose for this is to compare its performance against the last 3 models where cohort-specific stats are used instead. These statistics are said to be the most influential and telling statistics about a player's performance in each role.

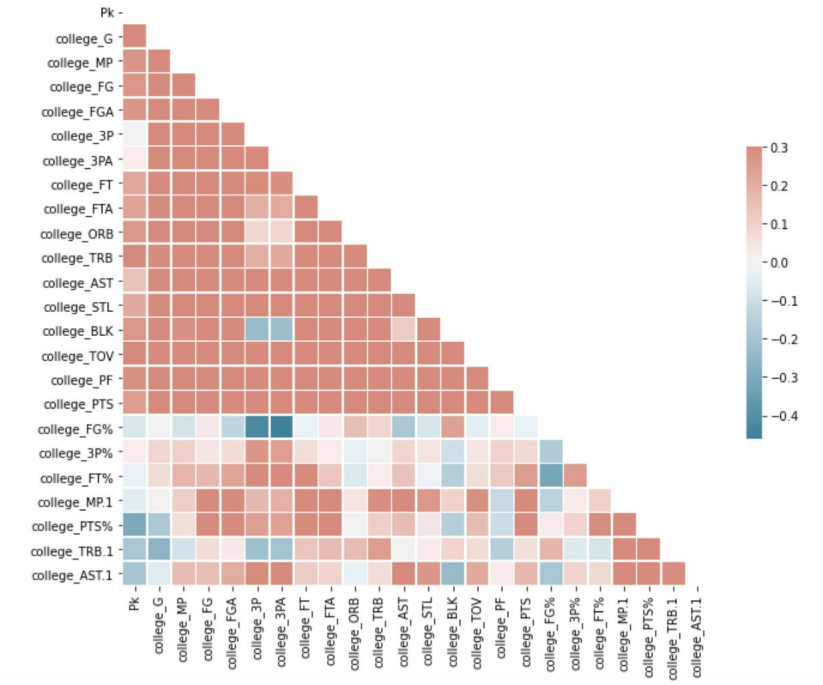
Hypothesis

Bigs tend to be the tallest player on the team, so we predict that their points scored, rebounds, and blocks will be telling variables in their standings [1]. Wing players generally have the best shooting skills on the team; for this reason we believe that variables like points scored per game will greatly improve their draft positions [1]. Finally, Point Guards lead the team's offensive plays and are known for their versatility on the court [1]. We hypothesize that each of the aforementioned college statistics for respective cohort types will be better indicators for rankings in the draft than holistic college performance.

Findings

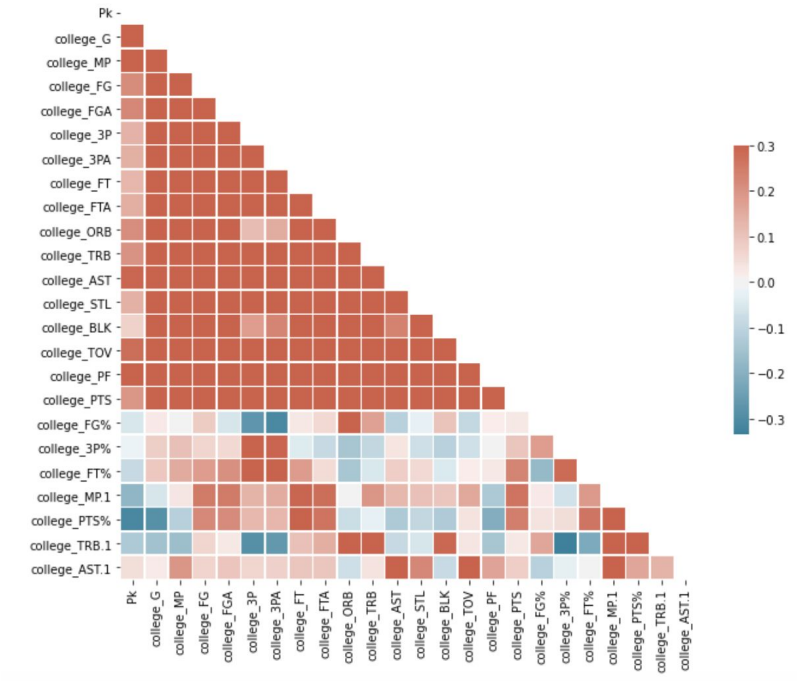
Before performing any further analyses, we first need to re-rank players after they have been split into their cohorts. This is to decrease the range of values that the model will need to predict from. For example, a player who plays as a point guard and might be ranked 10th out of 60 candidates, but after being grouped by his position he may be the top pick out of 12. In this case we want the model to predict a ranking of 1, not 10. Additionally, we want to look at the historical correlation of each college statistic with draft picks. This will help us to decide which ones to include in cohort-specific statistical analyses. To do this we construct correlation matrices shown below.

Figure 3: Correlation Matrix - Bigs



For Bigs, it is observed that the most telling statistics of their draft rankings and value are number of games (college_G), total rebounds (college_TRB), and turnovers (college_TOV).

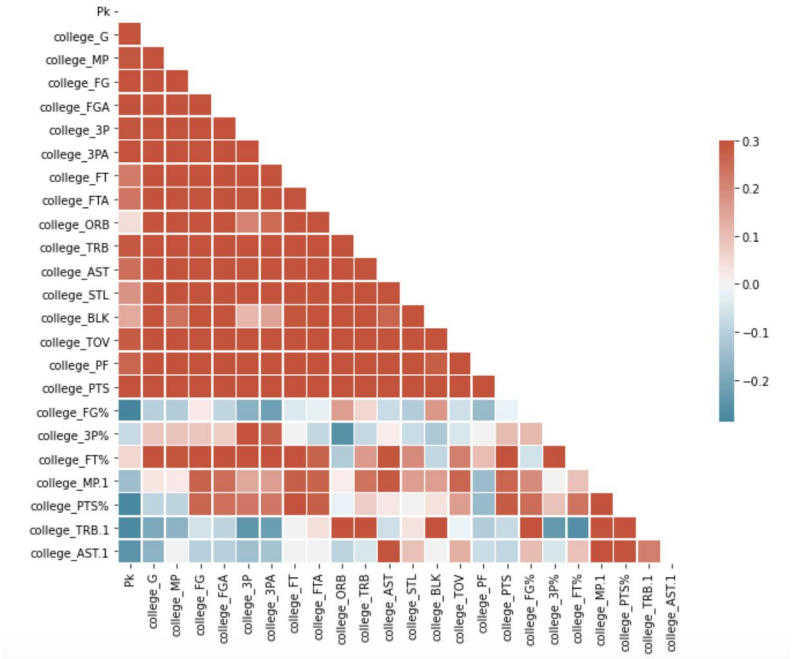
Figure 4: Correlation Matrix - Wings



In players who play Wing positions, it is observed that the most telling statistics of their

draft rankings and value are number of personal fouls (college_PF), assists (college_AST), points percentage (college_PTS%), and minutes played (college_MP).

Figure 5: Correlation Matrix - Point Guards



Point Guards are observed to be more well rounded than other cohorts, as their picks are generally affected by more of their statistics. The most telling statistics of their draft rankings and value are number of games (college_PTS), total rebounds (college_TRB), college_G, college_FG%, college_3P, college_3PA, college_FG, college_FGA, and college_MP.

We implement K-Fold Cross Validation to test model performance and accuracy. Figure 6, Figure 7, and Figure 8 show the impact each k value has on model prediction quality for each position. For model comparisons below we chose a k-value of 10 to ensure consistency.

Figure 6: Model Quality - Bigs

Bigs							
All College Stats (177 Data Points)				Cohort-Specific College Stats (585 Data Points)			
k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions	k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions
10	5.9951942	0.01171	0	10	5.7976763	0.0919686	0
9	5.9678704	0.01876	0	9	5.8060704	0.0717865	0
8	6.1035084	0.03781	0	8	5.7983009	0.0950674	0

Figure 7: Model Quality - Wings

Wings							
All College Stats (158 Data Points)				Cohort-Specific College Stats (389 Data Points)			
k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions	k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions
10	5.5346066	-0.00764	0	10	4.7166014	0.0681262	0
9	5.4435699	0.02316	0	9	4.7292096	0.0557137	0
8	5.4910796	0.0107	0	8	4.7472945	0.0999161	0

Figure 8: Model Quality - Point Guards

Point Guards							
All College Stats (100 Data Points)				Cohort-Specific College Stats (301 Data Points)			
k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions	k-Value	Avg. RMSE	Avg. R2	Avg. Fraction of Correct Predictions
10	3.3071488	-0.00244	0	10	2.9799242	0.1140673	0.1
9	3.358067	-0.05911	0.111111111	9	2.9577376	0.1292312	0.1
8	3.4120158	-0.101	0	8	2.9901465	0.1178891	0

From our results we can see that in all cases, the regression models' R-squared scores are close to zero. This means that our regression doesn't fit the data particularly well, but is not a true indicator of the regression's quality. When looking at R-squared values, it is important to note that they do not indicate whether or not the model is adequate; good models can have low R-squared values, just as poor models can have high R-squared. Furthermore scenarios that include human psychology and behavior almost always have lower R-squared values. Simply put, unquantifiable and unpredictable variables will always be harder to predict than numerical ones.

A better indication of model quality in this case is the Average Root Mean Square Error, which measures the differences between values predicted and values observed. This value indicates the number of rankings that the predictions may be off by, and so the significance of each value is heavily dependent on how many data points are in each group. As the k-value and number of splits increases, the number of training and testing data points decreases. In every cohort type, we can see that the average RMSE score of predicting using full college stats versus cohort-specific college stats is similar. However, the number of datapoints that each model was predicting from differed greatly.

In Bigs, the regression model built for all college stats had an RMSE score of about 5.995 with 177/10 ~ 17 data points in each group of predictions (Figure 6). On the other hand, the model that used only cohort-specific stats scored 5.545 with 585/10 ~ 58 data points per group. The difference data point count is due to the number of missing or null values in certain stat columns, which resulted in us omitting them. Nonetheless it is evident that using the full dataset and college stats are less reliable in predicting standings than if we were to focus on cohort-specific college stats. In a group with significantly more values to predict, the ratio of the RMSE value to data size in cohort-specific stats was relatively lower than using all college stats. This confirms that for Bigs,

the number of assists (college_AST), blocks (college_BLK), assists per game (college_AST.1), points scored (college_PTS), and points scored percentage (college_PTS%) more accurately predict a Big's value and standing in the draft than looking at their college profile holistically [17].

The regression model built for Wings using all college stats had an RMSE score of about 5.535 with 158/10 ~ 15 data points in each group of predictions (Figure 7). Alternatively, the model that used only cohort-specific stats scored 4.717 with 389/10 ~ 38 data points per group. As mentioned above for Bigs, the drop in data points when looking at full college stats is due to missing data. Again however, we can see that the more reliable predictions came from the position specific stats model. The cohort-specific stats group had more data points, so the ratio of the RMSE value to data size in cohort-specific stats is justified and is actually a very low margin of error. This confirms that for Wings, personal fouls (college_PF), assists (college_AST), points percentage (college_PTS%), and minutes played (college_MP) provide a clearer picture of where these type of players stand among their peers [17].

The last regression performed was for Point Guards. Without specifying what college stats to use, the RMSE score was 3.307 with 100/10 ~ 10 data points in each group of predictions (Figure 8). When considering only the college stats that we believed were important for this position, the RMSE value was 2.980 with 301/10 ~ 30 data points per group. Like above, the reliable predictions came from the position specific stats model, and the relative RMSE scores were more proportionate to the size of the data set. This confirms that for Point Guards, the number of games (college_PTS), total rebounds (college_TRB), games played (college_G), field goal percentage (college_FG%), three point field goals (college_3P), three point field goal attempts (college_3PA), field goals (college_FG), field goal attempts (college_FGA), and minutes played (college_MP) are the most telling and accurate predictors for point guards [17].

Unsupervised Learning

We developed three comprehensive clustering models using Python modules such as Numpy, Pandas, and MatPlotLib. Each model was designed so that input parameters could be easily changed by configuring values within a Jupyter Notebook cell. Subsequently running the model would display a number of visualizations and aggregations that reflect the input parameters specified. By creating our model in a dynamic fashion, we were able to seamlessly run our clustering algorithm and tweak our inputs in between runs to maximize our understanding of the clustering outputs while minimizing the time spent adjusting business logic for every specific case we wished to examine.

Figure 9: Model Input Parameters

Parameter	Explanation
num_clusters	The value of K in our K-means clustering algorithm. For example, num_clusters = 3 means that 3 centroids/clusters will be generated.
max_draft_selection	The max draft selection we wish to include in the analysis. For example, max_draft_selection = 30 means that only the top 30 draft picks will be considered for each draft class included.
positions	The player positions we wish to include. Type is a list of strings where each string is a position to be included in the analysis. Setting this as an empty array includes all players in the analysis. For example, positions = [“Point”, “Center”] means that only players of position “Point” and “Center“ will be included in the analysis.
start_year (inclusive) end_year (exclusive)	Defines the range of draft classes we wish to include in the analysis. For example, start_year = 1990, end_year = 2010 means that all draft classes between the year of 1990 and 2010 will be included in the analysis.

Task 1: Birth Place

Purpose

The purpose of this task is to cluster players based on where they are from using the coordinates of their birthplace. We wish to aggregate the data by decade to reveal trends on the number of international draftees in the NBA. This also hints at the overall interest of the sport of basketball in areas with high numbers of draft prospects. From a business perspective, we want to inform the league about the reach of their sport as well as how its popularity is growing in different parts of the world. This will allow league officials to make data-driven decisions regarding where to increase international marketing and hold international events.

Hypothesis

We predicted that the number of international draft picks will significantly increase over time and we will see a slight drop in prospects from North America. This is due to how much the sport of basketball has grown in popularity and competition throughout the years in places like Europe, Africa, and Asia.

Findings

Firstly, we will execute our model on all players drafted between 1990 and 2019 with K = 5. The elbow method tells us that the optimal number of clusters is around 4 or 5. We can first analyze the 2-dimensional centroid scatter plots for bp_longitude vs bp_latitude developed by our model (Figure 10). We can even plot the clusters on a geographical map to fully get a sense of where things are located (Figure 11). We can label our clusters based on their location on the map. Finally, for each decade, we group the draftees by cluster to calculate the percentage of total players that belong in each cluster (Figure 12).

Figure 10: Clustering Player Birthplaces: Centroids and Clusters

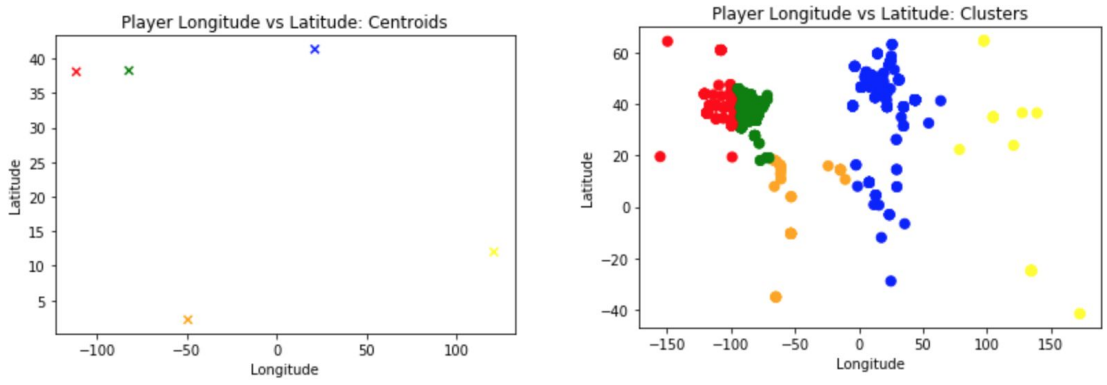


Figure 11: Clustering Player Birthplaces: Geographical Map of Clusters

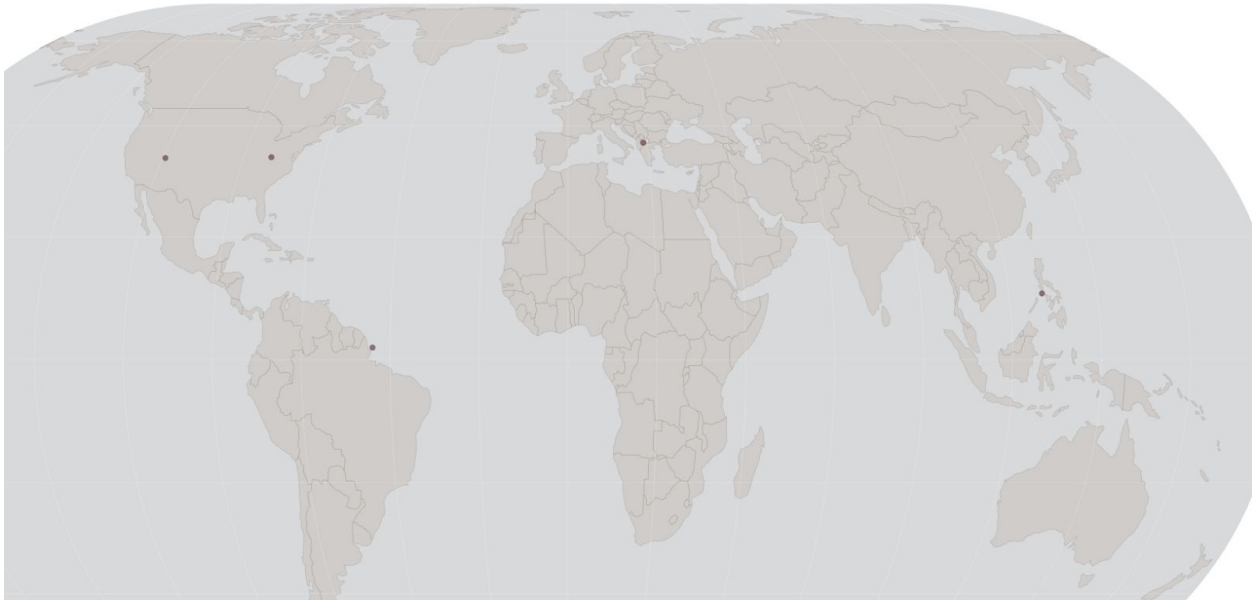


Figure 12: Clustering Player Birthplaces: Aggregating Clusters by Decade

Cluster ID/Color	Cluster Label	1990s Draft %	2000s Draftee %	2010s Draftee %
0/Red	East Coast USA	20.4	19.8	22.3
1/Blue	Europe/Africa	10.8	24.5	22.8
2/Green	West Coast USA	66.5	49.0	50.0
3/Yellow	Asia/Australia	1.0	2.8	1.8
4/Orange	South America/West Africa	1.0	3.7	2.8

Results

We hypothesized that the amount of international talent outside of North America. This is exactly what was reflected in our data. Almost a quarter of the draft picks in the 2010's were from the "Europe/Africa" cluster and the percentage of draft picks in that cluster has increased by 12% over time. The" South America/West Africa" cluster also saw a slight increase of 1.8% in percentage of draft selections. As expected, we saw a 9.6% decrease in draft picks from North American clusters though they still hold the majority of prospects. The number of prospects from Asia remained relatively the same. This was surprising considering the vast population of Asia. The reason could be that Asian players are likely to experience a huge culture shock by transitioning to the NBA and most choose not to go through this [20]. They may also make more money playing in the Asian leagues as the competitive nature of the NBA results in a decreased pay cut.

Task 2: Shooting Percentages

Purpose

The purpose of this task is to cluster players by their shooting positions, aggregate the data by decade and analyze how the demand for shooting has changed over time. We are making the assumption that the skill set of NBA draftees is reflective of that of professional players in high demand. Furthermore, we wish to cluster subsets of players to analyze how the requirements for their position have changed over time. From a business perspective, we reveal to players how their position has evolved and which types of shots to invest in to improve their draft stock. We also wish to analyze how the play style of the sport has evolved over time so that NBA teams can determine the kinds of players to draft in order to stay competitive with the rest of the league.

Hypothesis

We predicted that clusters with a high shooting percentages will have a higher percentage of players drafted in the modern era. This includes free throw shooting and three point shooting. These predictions are based on the NBA's high demand for three-point shooting in the modern era and how teams are more likely to draft players with the potential to become great shooters [12]. Furthermore, we believe that the improved training available in the modern era paired with the demand for scoring has increased the value of a high free throw percentage for NBA prospects [16]. We also predict that clusters with low shooting percentages will have a field goal shooting percentage of >50%. In order for these players to compete in today's league, they must make up for their lack of shooting with accurate play closer to the basket.

We also predicted that clustering NBA draftees that play the center position will reveal a dramatic increase in three point shooting percentage and free throw percentage for players drafted in the modern era. This is due to the rise of "positionless" basketball, the idea that scoring should not be confined to specific positions on the court [10]. This increases the opportunities for players of size to perform the roles of smaller players. In our case, we believe it should be reflected in the shooting mechanics of draftees that play center. We do not expect a large change in field goal percentage because those shots include shots close to the basket, where players of size tend to excel in.

Findings

Firstly, we will execute our clustering model and include all players drafted between 1990 and 2019 in the analysis. We must first clean our data to remove outliers. For example, a player that has shot 100% from three point range with 1 attempt should not be included in our analysis. By performing the elbow method, we note that the optimal value of "K" lies around 4. After running our model, we can first analyze the 3-dimensional scatter plots (Figure 13) and label each cluster by their ability to shoot the basketball. For each decade, we group the draftees by cluster and calculate the percentage of total players in each cluster (Figure 14).

Figure 13: Clustering All Player Shooting Percentages: Centroids and Clusters

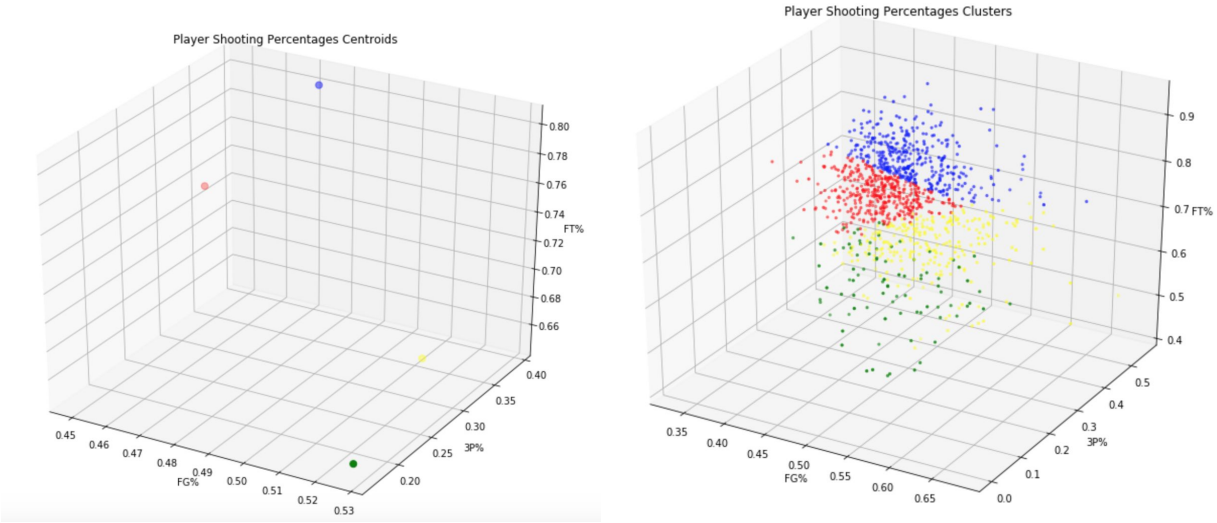


Figure 14: Clustering All Player Shooting Percentages: Aggregating Clusters by Decade

Cluster ID/Color	Shooting Label	Attributes of Cluster (From Centroid Data)	1990s Draft %	2000s Draftee %	2010s Draftee %
0/Red	Average	~73.5% FT ~44.9% FG ~34.2% 3P	31.0	37.3	38.4
1/Blue	Experts	~80.1% FT ~47.3% FG ~39.2% 3P	34.1	32.2	34.4
2/Green	Poor	~64.8% FT ~52.7% FG ~16.6% 3P	8.6	8.1	6.8
3/Yellow	Novice	~65.2% FT ~51.5% FG% ~33.3 3P%	26.1	22.2	20.2

For our next experiment, we configure the model parameters so that only draftees of the center position are included in the analysis. By revisiting the elbow method for the new data, we set K=3. Again, we can analyze the 2-dimensional centroid scatter plots for FT% vs 3P% (Figure 15) and label each cluster on their ability to shoot. We also perform the same aggregation as the previous experiment to calculate how the percentage of total players in each cluster have changed through each decade (Figure 16).

Figure 15: Clustering Center Shooting Percentages: Centroids and Clusters

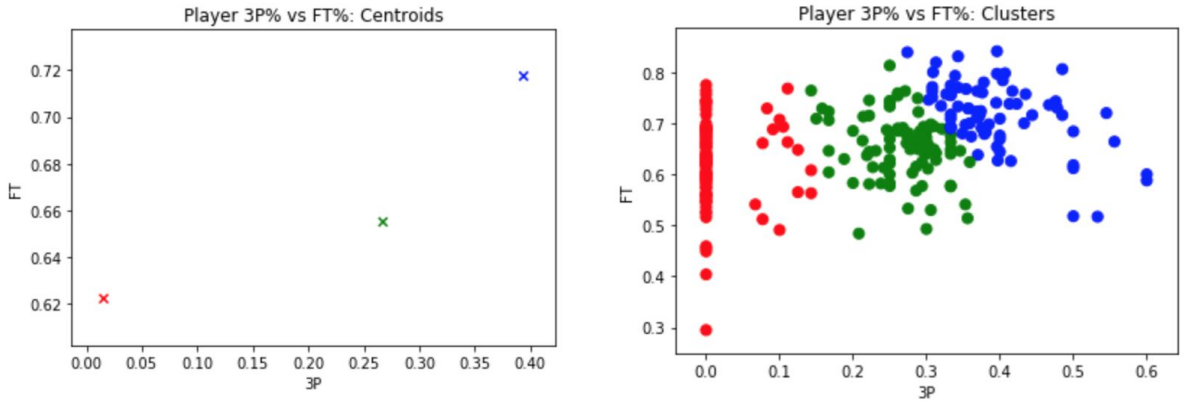


Figure 16: Clustering Center Shooting Percentages: Aggregating Clusters by Decade

Cluster ID/Color	Shooting Label	Attributes of Cluster (From Centroid Data)	1990s Draft %	2000s Draftee %	2010s Draftee %
0/Red	Poor	Low FT% Low 3P%	49.0	32.0	23.3
1/Blue	Expert	High FT% High 3P%	20.5	32.0	38.9
2/Green	Average	Average FT% Average 3P%	30.3	35.8	37.6

Results

For the first experiment, we predicted that clusters with a high shooting percentages will have a higher percentage of players drafted in the modern era. The cluster of "Average" shooters saw a large jump over time and the cluster of "Poor" shooters saw a decrease over time. This makes sense based on our estimation. NBA teams became increasingly interested in players who can shoot to adapt to the changing landscape of the NBA and its reliance on three point shooting [12]. Our findings also show that free throw shooting has become a larger factor when drafting prospects. Teams are more likely to draft players who can make free throws. This could be because good free throw shooters usually have the mechanics to become great three point shooters. Surprisingly, the cluster of "Expert" shooters experienced a small jump. This could be because teams draft based on potential. They are more concerned that a player has the potential to become a great shooter than whether or not the player is already an "Expert" shooter. That would justify why teams have not selected an abundance of "Expert" shooters, as teams are satisfied with "Average" shooting players with other skills that "Expert" shooters do not have to offer. We also made the prediction that clusters with low shooting percentages will have a field goal shooting percentage of > 50% We can see that the "Poor" and "Novice" clusters who are not recognized for their ability to shoot have the highest FG%. This is expected as poor shooting clusters must prove that they can score efficiently closer to the basket.

Our second experiment revealed a 18.4% increase in center prospects who are expert shooters as well as a 25.7% decrease in "Poor" shooting center prospects. The "Average" cluster also saw an increase of 7.3%. This makes sense and further supports the notion of "positionless" basketball [10]. The findings show that teams are more likely to select center prospects with guard-like skills like shooting. They are also less satisfied with center prospects who cannot shoot. It seems like shooting has become an expected skill for NBA centers. We can see that in the 1990's and 2000's, shooting was not required for centers as 49% of centers drafted were poor shooters. In the modern era, it is a liability if the center on an NBA team cannot shoot and is uncommon for a center who cannot shoot to be selected. That is why only a quarter of center prospects in the 2010s are part of the "Poor" cluster.

Conclusions

In our study, we applied several data mining techniques to drive business insight within an application domain centered around the NBA Draft. We extracted, cleaned, transformed a comprehensive dataset containing 1500+ players that have been drafted over 30 years. We leveraged our dataset to predict draft order and clustered players using two sets of feature variables to uncover draft trends that reflect the changing landscape of the NBA.

Findings from our supervised learning models confirm that college stats are still the greatest predictor of a player's standings in the NBA draft. Though the accuracy of the predictions made by the model were quite low, this is ultimately what we expected as our analyses completely omit other equally important but immeasurable variables like perceived potential. These non-quantifiable traits introduced outliers in the data, which ultimately resulted in an average Random Mean Square Error of 3.2526 across the position types. A closer look at the actual predictions and true values showed that most of the results were not too far off, and that it was in fact these outliers that contributed to a poor RMSE score. However when it comes to deciding who their top choices should be, teams can still apply these models to help them make better data-driven decisions. Where it falls short in accuracy, it makes up for it by providing a stronger baseline for coaches to identify the leading players in each position. In future analyses we could look deeper in the data to find and accurately omit these outliers, or look into quantifying them. Another thing to note is that a player's performance at one college cannot be equally compared to someone at another. Just as undergraduate admissions have adjustment factors for high school students, the same should be done for athletes and their respective colleges. To further improve this study in the future, we would further explore this theory and consider normalizing the data and college stats.

Our findings from K-means clustering tasks revealed valuable insight for every aspect of our business problem. For the technique itself, we found that the elbow method was an accurate way to calculate the optimal K value. By setting K larger than the output of the elbow method, our model became less centralized and we would often experience clusters that were sparse. Likewise, setting K smaller than the output of the elbow method revealed clusters that were difficult to classify as they represented a broad range of data points.

From the standpoint of teams around the league, we learned that teams are adapting the concept of "positionless" basketball more than ever [10]. By clustering prospects of the center position, we realized that the demand for centers with a high three point percentage and free throw percentage has increased by 25.7% and that the demand for non-shooting centers. Over time, teams have learned how to use "hybrid" positions to their advantage through experience and analytics. This study benefits teams that continue to build around traditional positions as it proves that the league is changing at a rapid pace and teams should investigate how to adapt "positionless" players in order to stay competitive with other teams that have embraced this concept.

From a player's perspective, we were able to cluster players by shooting percentages to analyze the demand for shooting in the NBA. We learned that the demand for shooters with a high three point percentage and free throw percentage has increased by 7.7% as teams are more reliant on three point shooters than ever [18]. We also learned that teams are more interested in players that possess the potential to shoot while offering other skills, in contrast to just excellent shooters. This study also revealed that it is essential for players who cannot shoot three pointers and free throws to shoot accurately from the field in order to be a serious consideration in the draft. Our study shows that their field goal percentage should be ideally 50%+. This information is beneficial to players as it informs them of the skills they should invest in to stay competitive in the draft.

Lastly, our study analyzed the number of international draftees by clustering the birthplace of NBA draftees. This was also a way to analyze the hotspots of basketball popularity outside of North America. We learned the number of international players drafted has skyrocketed since the 1990s. Specifically, players drafted from Africa and Europe have seen a 12% increase since the 1990s. Therefore, the international popularity of the sport of basketball is growing. Though this is the case, the number of players from Asia has not seen a massive increase despite its dense population. This is likely due to the avoidance of culture shock and the pay gap between the NBA and other Asian leagues [20]. This data is beneficial for the NBA itself as it informs them of areas that would welcome NBA-issued events and programs. Measuring the international outreach of the league also tells league officials where to increase marketing in order to continue to grow the product of basketball.

References

1. Basketball Positions and Roles. (n.d.). Retrieved November 02, 2020, from https://www.myactivesg.com/Sports/Basketball/How-To-Play/Basketball-Rules/Basketball-Positions-and-Roles
2. Basketball Statistics and History. (n.d.). Retrieved October 17, 2020, from https://www.basketball-reference.com/
3. Beckham, J. (2017, June 03). Analytics Reveal 13 New Basketball Positions. Retrieved November 08, 2020, from https://www.wired.com/2012/04/analytics-basketball/
4. Coates, Dennis & Oguntimein, Babatunde. (2008). The Length and Success of NBA Careers: Does College Production Predict Professional Outcomes?. International Journal of Sport Finance. 5.
5. Conway, T. (2019, February 16). NBA Announces Basketball Africa League to Start in 2020; Barack Obama Involved. Retrieved November 08, 2020, from https://bleacherreport.com/articles/2821107-nba-announces-basketball-africa-league-to-start-in-2020-barack-obama-involved
6. Evans, Brent. (2017). The Determinants of Draft Position for NBA Prospects. 10.13140/RG.2.2.28360.11528.
7. Favale, A. (2020, December 04). Bleacher Report's Top 100 Player Rankings from the 2019-20 NBA Season. Retrieved November 11, 2020, from https://bleacherreport.com/articles/2889335-bleacher-reports-top-100-player-rankings-from-the-2019-20-nba-season
8. Gandhi, A., Tiwari, S., & Nelson, C. (2017). Identifying high frequency shooting zones for NBA teams using clustering. IIE Annual Conference.Proceedings, , 1811-1816. Retrieved from http://search.proquest.com.proxy.lib.uwaterloo.ca/scholarly-journals/identifying-high-frequency-shooting-zones-nba/docview/1951123198/se-2?accountid=14906
9. Green, J. (2019, March). How does the NBA draft work? Retrieved October 17, 2020, from https://blog.betway.com/basketball/how-does-the-nba-draft-work-nba-draft-explained/
10. Gross, G. (2020, July 17). Positionless basketball: The future, or a waystation? Retrieved November 08, 2020, from https://www.denverstiffs.com/2020/7/17/21328531/nba-positionless-basketball-the-future-or-a-waystation-denver-nuggets-jokic-bol-bol
11. Haefner, J. (2015). 9 Stats That Every Serious Basketball Coach Should Track. Retrieved November 11, 2020, from https://www.breakthroughbasketball.com/stats/9_stats_basketball_coach_should_track.html
12. Kram, Z. (2019, February 27). The 3-Point Boom Is Far From Over. Retrieved November 13, 2020, from https://www.theringer.com/nba/2019/2/27/18240583/3-point-boom-nba-daryl-morey

13. Lorenzo, J., Lorenzo, A., Conte, D., & Giménez, M. (2019). Long-Term Analysis of Elite Basketball Players' Game-Related Statistics Throughout Their Careers. Frontiers in psychology, 10, 421. https://doi.org/10.3389/fpsyg.2019.00421
14. Parker, C. (2018). NBA Draft Pick Valuation (Working paper). ResearchGate. doi:https://www.researchgate.net/publication/328381652_NBA_Draft_Pick_Valuation
15. Patel, R. (2017). Clustering Professional Basketball Players by Performance. UCLA. ProQuest ID: Patel_ucla_0031N_16330. Merritt ID: ark:/13030/m54j59wd. Retrieved from https://escholarship.org/uc/item/917739k8
16. Reynolds, T., & Kurz, H., Jr. (2019, March 21). How cutting-edge technology helps basketball players shoot. Retrieved November 09, 2020, from https://www.usatoday.com/story/sports/ncaab/2019/03/21/how-cutting-edge-technology-helps-basketball-players-shoot/39233299/
17. Sailofsky, D. (2018, April 8). Drafting Errors and Decision Making Theory in the NBA Draft [Scholarly project]. In Brock University Library. Retrieved October 15, 2020, from https://dr.library.brocku.ca/bitstream/handle/10464/13452/Brock_Sailofsky_Daniel_2018.pdf?sequence=1&isAllowed=y
18. Shea, S. (n.d.). The 3-Point Revolution. Retrieved October 27, 2020, from https://shottracker.com/articles/the-3-point-revolution
19. The Long Weird History of the NBA Draft. (n.d.). Retrieved November 12, 2020, from https://nbahoopsonline.com/Articles/History/Drafthistory.html
20. Wong, A. (2017). Why hasn't China produced more NBA talent? Retrieved November 16, 2020, from https://www.thescore.com/nba/news/1362618
21. Xu, M., & Gao, J. (2017). Analysis of NBA Team Strength Using Players' Race Data Based on Clustering Method (12th ed., Vol. 7, Working paper). International Journal of Social Science and Humanity. doi:http://www.ijssh.org/vol7/919-HS006.pdf
22. Yin, F., Hu, G., & Shen, W. (2020). Analysis of professional basketball field goal attempts via a Bayesian matrix clustering approach (Publication). Cornell University. Retrieved from https://arxiv.org/abs/2010.08495
23. Zhang, L., Lu, F., Liu, A., Guo, P., & Liu, C. (2016). Application of K-Means Clustering Algorithm for Classification of NBA Guards. International Journal of Science and Engineering Applications, 5, 001-006.

Appendix

Data Pipeline/Web Scraper Code

basketball-reference-scraper.ipynb

Basketball Reference Scraper

In [1]:
```
from urllib.request import urlopen
from bs4 import BeautifulSoup, Comment
import pandas as pd
import requests
import numpy as np
```

In [2]:
```
# Function that appends player college stats and other player properties such as height, weight, position.
# This function is abstracted because this data had to be fetched from a the player URL
def append_college_stats(rows, year):
    player_stats = []
    for i in range(len(rows)):
        player = [td.getText() for td in rows[i].findAll('td')]
        if player == []:
            continue
        td_list = rows[i].findAll('td')
    for i in range(len(rows)):
        player = [td.getText() for td in rows[i].findAll('td')]
        if player == []:
            continue
        td_list = rows[i].findAll('td')
        for td in td_list:
            if td['data-stat'] == "player":
                button = td.findAll('a')

                if button != []:
                    link = button[0]['href']
                    player_url = "https://www.basketball-reference.com{}".format(link)
                    player_html = urlopen(player_url)
                    player_soup = BeautifulSoup(player_html)
                    college_stats = player_soup.findAll(id="all_all_college_stats")

                    player_height_temp = None
                    if player_soup.findAll("span",{'itemprop': 'height'})[0]:
                        player_height_temp = player_soup.findAll("span",{'itemprop': 'height'})[0].text

                    player_height = None
                    if (int(player_height_temp.split('-')[0])+(int(player_height_temp.split('-')[1]))):
                        player_height = int(player_height_temp.split('-')[0])+(int(player_height_temp.split('-')[1])/12

                    player_weight = None
                    if(player_soup.findAll("span",{'itemprop':'weight'})[0]):
                        player_weight = player_soup.findAll("span",{'itemprop':'weight'})[0].text
```

```
                    birth_date = None
                    if(player_soup.findAll("div",{"itemtype":"https://schema.org/Person"})[0].findAll("span",{"id":"nec
                        birth_date = player_soup.findAll("div",{"itemtype":"https://schema.org/Person"})[0].findAll("sp

                    birth_place = None
                    if(player_soup.findAll("div",{"itemtype":"https://schema.org/Person"})[0].find("span",{"itemprop":'
                        birth_place = player_soup.findAll("div",{"itemtype":"https://schema.org/Person"})[0].find("span

                    for each in player_soup.findAll('p'):
                        if 'Position:' in each.text:
                            player_position = each.text.replace('\n','').strip().split(' ')[4]

                    if college_stats != []:
                        for comments in college_stats[0].findAll(text=lambda text:isinstance(text, Comment)):
                            x = comments.extract()
                            comment_soup = BeautifulSoup(x , 'lxml')
                            tfoot = comment_soup.findAll('tfoot')
                            td = tfoot[0].findAll('td')
                            for i in range(2, len(td)):
                                player.append(td[i].getText())
                    else:
                        noneArr = [None]*23
                        player = player + noneArr

                    player.append(player_height)
                    player.append(player_weight)
                    player.append(birth_date)
                    player.append(birth_place)
                    player.append(player_position)
                    player.append(year)
                    r =requests.get('https://api.opencagedata.com/geocode/v1/json?q={}&key=14a3bf18acca4556a59095760016
                    results = r.json()['results']
                    if results:
                        coord = r.json()['results'][0]['annotations']['DMS']
                        player.append(coord['lat'])
                        player.append(coord['lng'])
                    else:
                        player.append(0)
                        player.append(0)

        player_stats.append(player)

        # Print string as a status update to user on what player is being processed at the moment
        print('PROCESSING PLAYER {}'.format(player_url))
    return player_stats
```

In [3]:
```
# Defining range of draft classes
start_year = 1990
end_year = 2020
college_headers = ['college_G',
                   'college_MP',
                   'college_FG',
                    'college_FGA',
                    'college_3P',
                    'college_3PA',
                    'college_FT',
                    'college_FTA',
                    'college_ORB',
                    'college_TRB',
                    'college_AST',
                    'college_STL',
                    'college_BLK',
                    'college_TOV',
                    'college_PF',
                    'college_PTS',
                    'college_FG%',
                    'college_3P%',
                    'college_FT%',
                    'college_MP',
                    'college_PTS%',
                    'college_TRB',
                    'college_AST',
                   'height',
                   'weight',
                   'date_of_birth',
                   'place_of_birth',
                   'position',
                   'year',
                   'bp_latitude',
                   'bp_longitude'
                  ]

# Initialize .csv writer
writer = pd.ExcelWriter('nba_draft.xlsx', engine='xlsxwriter')
```

```
# Push data to .csv file (1 sheet per draft class)
for i in range(start_year, end_year + 1):
    url = "https://www.basketball-reference.com/draft/NBA_{}.html".format(i)
    html = urlopen(url)
    soup = BeautifulSoup(html)
    rows = soup.findAll('tr', limit=2)
    headers = [th.getText() for th in rows[1].findAll('th')]
    headers = headers[1:]
    headers = headers + college_headers
    rows = soup.findAll('tr')[2:]
    player_stats = append_college_stats(rows, i)
    stats = pd.DataFrame(player_stats, columns = headers)
    data = stats.head(60)
    stats.to_excel(writer, sheet_name='draft_data_{}'.format(i), index = False)

# Save .csv data
print('Scraping Completed!')
writer.save()
```

Supervised Model Code

correlation_matrices.ipynb

Correlation Matrices

Import Modules

In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline
```

Import Excel Data

In [2]:
```
# import scraped NBA stats from Excel file

xls = pd.ExcelFile('nba_draft.xlsx')

map = {}
for sheet_name in xls.sheet_names:
    map[sheet_name] = xls.parse(sheet_name)

del map['draft_data_2020']
```

Correlation Matrices

Bigs

In the cell below, we look at the correlation between college stats and draft pick for Bigs (Center and Power Forward positions)

In [3]:
```
new_frame = pd.DataFrame(columns=map['draft_data_2019'].columns)
# looping through each dataframe item in map, and only acquiring the rows for players who play Small Forward
for key, value in map.items():
    # collect statistics for cohort types
    test = value.loc[value['position'].str.contains('Center') | value['position'].str.contains('Power')]
    # reset indices and renumber
    test.reset_index(drop=True,inplace=True)
    test.index = test.index + 1
    test.Pk = test.index
    # drop any rows that are missing data
    test = test.dropna()
    if test.empty is False:
        new_frame = new_frame.append(test)

new_frame = new_frame[['Pk', 'college_G', 'college_MP', 'college_FG', 'college_FGA',
       'college_3P', 'college_3PA', 'college_FT', 'college_FTA', 'college_ORB',
       'college_TRB', 'college_AST', 'college_STL', 'college_BLK',
       'college_TOV', 'college_PF', 'college_PTS', 'college_FG%',
       'college_3P%', 'college_FT%', 'college_MP.1', 'college_PTS%',
       'college_TRB.1', 'college_AST.1']]
for column in new_frame:
    new_frame[column] = pd.to_numeric(new_frame[column])

# compute the correlation matrix
corr = new_frame.corr()

# generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# format diverging colour palette for correlation matrix
cmap = sns.diverging_palette(230, 20, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

Wings

In the cell below, we look at the correlation between college stats and draft pick for Wing positions (Shooting Guard and Small Forward positions)

In [4]:
```
new_frame = pd.DataFrame(columns=map['draft_data_2019'].columns)

# looping through each dataframe item in map, and only acquiring the rows for players who play Small Forward
for key, value in map.items():
    # collect statistics for cohort types
    test = value.loc[value['position'].str.contains('Shooting') | value['position'].str.contains('Small')]
    # reset indices and renumber
    test.reset_index(drop=True,inplace=True)
    test.index = test.index + 1
    test.Pk = test.index
    # drop any rows that are missing data
    test = test.dropna()
    if test.empty is False:
        new_frame = new_frame.append(test)

new_frame = new_frame[['Pk', 'college_G', 'college_MP', 'college_FG', 'college_FGA',
       'college_3P', 'college_3PA', 'college_FT', 'college_FTA', 'college_ORB',
       'college_TRB', 'college_AST', 'college_STL', 'college_BLK',
       'college_TOV', 'college_PF', 'college_PTS', 'college_FG%',
       'college_3P%', 'college_FT%', 'college_MP.1', 'college_PTS%',
       'college_TRB.1', 'college_AST.1']]
for column in new_frame:
    new_frame[column] = pd.to_numeric(new_frame[column])

# compute the correlation matrix
corr = new_frame.corr()

# generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# format diverging colour palette for correlation matrix
cmap = sns.diverging_palette(230, 20, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

Point Guards

In the cell below, we look at the correlation between college stats and draft pick for Point Guards.

In [5]:
```
new_frame = pd.DataFrame(columns=map['draft_data_2019'].columns)

# looping through each dataframe item in map, and only acquiring the rows for players who play Small Forward
for key, value in map.items():

    # collect statistics for cohort types
    test = value.loc[value['position'].str.contains('Point')]
    # reset indices and renumber
    test.reset_index(drop=True,inplace=True)
    test.index = test.index + 1
    test.Pk = test.index
    # drop any rows that are missing data
    test = test.dropna()
    if test.empty is False:
        new_frame = new_frame.append(test)

new_frame = new_frame[['Pk', 'college_G', 'college_MP', 'college_FG', 'college_FGA',
       'college_3P', 'college_3PA', 'college_FT', 'college_FTA', 'college_ORB',
       'college_TRB', 'college_AST', 'college_STL', 'college_BLK',
       'college_TOV', 'college_PF', 'college_PTS', 'college_FG%',
       'college_3P%', 'college_FT%', 'college_MP.1', 'college_PTS%',
       'college_TRB.1', 'college_AST.1']]
for column in new_frame:
    new_frame[column] = pd.to_numeric(new_frame[column])

# compute the correlation matrix
corr = new_frame.corr()

# generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# format diverging colour palette for correlation matrix
cmap = sns.diverging_palette(230, 20, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

linear-regression.ipynb

Multiple Linear Regression

Import Modules

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

import sklearn
from sklearn import linear_model
from sklearn.model_selection import KFold
from sklearn.metrics import mean_squared_error, r2_score
```

Import Excel Data

In [2]:
```
# import scraped NBA stats from Excel file

xls = pd.ExcelFile('nba_draft.xlsx')

map = {}
for sheet_name in xls.sheet_names:
    map[sheet_name] = xls.parse(sheet_name)

del map['draft_data_2020']
```

Initialize Parameters

In [3]:
```
new_frame = pd.DataFrame(columns=map['draft_data_2019'].columns)

# create an instance of the model
lin_reg_mod = linear_model.LinearRegression()
test_set_rmse = 0
test_set_r2 = 0

# number of splits/ folds
n = 10

# initialize variable for sum of correct prediction fractions
sum_of_c_fraction = 0
```

Clean, Partition, Transform Data

In [4]:
```

# looping through each dataframe item in map, and only acquiring the rows for players who play Small Forward

for key, value in map.items():
    # collect statistics for cohort types - this is for Bigs
    test = value.loc[value['position'].str.contains('Center') | value['position'].str.contains('Power')]
    test = test[['Pk', 'college_G', 'college_TRB', 'college_TOV' ]]
    # reset indices and renumber
    test.reset_index(drop=True,inplace=True)
    test.index = test.index + 1
    test.Pk = test.index
    # drop any rows that are missing data
    test = test.dropna()
    if test.empty is False:
        new_frame = new_frame.append(test)

new_frame = new_frame[['Pk', 'college_G', 'college_TRB', 'college_TOV']]

for column in new_frame:
    new_frame[column] = pd.to_numeric(new_frame[column])


X = new_frame.loc[:,new_frame.columns.str.startswith('college')]
y = new_frame['Pk'].astype(int)
```

Linear Regression with K-Fold Cross Validation

In [5]:
```
kf = KFold(n_splits=n, shuffle=True, random_state=1)


for train_index, test_index in kf.split(X):
    # initialize variable for counting number of correct predictions
    c = 0

    lin_reg_mod.fit(X.iloc[train_index],y.iloc[train_index])

    y_pred = lin_reg_mod.predict(X.iloc[test_index])

    # compare final prediction values against true values
    final_predictions = pd.DataFrame(columns = ['True Ranking','Predicted Ranking'])
    y_test = list(y.iloc[test_index])
    y_pred = list(y_pred)
    for i in range(0,len(y_test)):
        new_row = {'True Ranking': y_test[i],'Predicted Ranking': int(y_pred[i])}
        if y_test[i] == int(y_pred[i]):

            # increment c value if prediction is correct
            c = c + 1
        final_predictions = final_predictions.append(new_row, ignore_index=True)

    print(final_predictions)
    print(str(c/n))
    print(r2_score(y_test,y_pred))
    sum_of_c_fraction = sum_of_c_fraction + (c/len(y_test))
    # check the predictions against the actual values by using the root mean square deviation
    # and coefficient of determination metrics
    test_set_rmse = test_set_rmse + (np.sqrt(mean_squared_error(y_test, y_pred)))
    test_set_r2 = test_set_r2 + r2_score(y_test, y_pred)

print('Average RMSE: ' + str(test_set_rmse/n))
print('Average R2: ' + str(test_set_r2/n))
print('Accuracy: ' + str((1/n)*int(sum_of_c_fraction)))
```

Coefficients Obtained From Linear Regression

In [6]:
```
# dataframe showing all the features and their estimated coefficients obtained from the linear regression
coeff_df = pd.DataFrame(X.columns)
coeff_df.columns = ['Features']
coeff_df['Coefficient Estimate'] = pd.Series(lin_reg_mod.coef_)
coeff_df
```

Out[6]:

	Features	Coefficient Estimate
0	college_G	0.115267
1	college_TRB	-0.008381
2	college_TOV	-0.006985

Unsupervised Model Code

clustering-location.ipynb

Clustering Analysis (Birth Place Location)

Import Modules

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import sklearn
from sklearn.cluster import KMeans
import sklearn.metrics as sm
from sklearn import datasets
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import scale
from lat_lon_parser import parse
from scipy.cluster.hierarchy import dendrogram, linkage
%matplotlib inline
```

Import Excel Data

In [2]:
```
map = {}

# Loading Excel Data and adding each sheet into a map with its respective data
xls = pd.ExcelFile('nba_draft.xlsx')
for sheet_name in xls.sheet_names:
    map[sheet_name] = xls.parse(sheet_name)
```

Initialize Parameters

In [3]:
```
# Define number of clusters (K)
num_clusters = 5

# Define the maximum draft pick number to be included in the analysis
max_draft_selection = 60

# Define the positions of interest
positions = []

# Define start year of draft class range
start_year = 1990

# Define end year of draft class range
end_year = 2020

# Define color theme for scatter plots
color_theme = np.array(['red','blue','green', 'yellow', 'orange', 'brown', 'purple', 'teal', 'dark blue', 'dark green'
```

Clean, Partition, Transform Data

In [10]:
```
df = []

# Concatenate all sheets in draft class range
for i in range(start_year, end_year):
    df.append(map['draft_data_{}'.format(i)])
location_df = pd.concat(df)

# Filter draft picks
location_df = location_df[location_df['Pk']  <= max_draft_selection]

# Filter positions of interest
if len(positions) != 0:
    location_df = location_df[location_df['position'].isin(positions)]

# Get columns of interest
loc_df = location_df.iloc[:, -2:]

# Drop null rows
loc_df = loc_df.dropna()

# Update longitude and latitude to decimal degree format
loc_df['bp_latitude'] = loc_df.bp_latitude.apply(lambda x: parse(x))
loc_df['bp_longitude'] = loc_df.bp_longitude.apply(lambda x: parse(x))

# Create a copy of the above dataframe with the year column (for aggregation)
loc_df_with_years = location_df.iloc[:, list(range(50,52)) + [-3]]
loc_df_with_years = loc_df_with_years.dropna()

# Perform K-means clustering and define feature variables
data = loc_df.to_numpy()
clustering  = KMeans(n_clusters=num_clusters,random_state=5)
clustering.fit(data)
loc_df.columns = ['Latitude','Longitude']
```

Elbow Method

In [5]:
```
# Execute elbow method to determine optimal K value
distortations = {}
for k in range(1,30):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(data)
  distortations[k] = kmeans.inertia_

plt.plot(list(distortations.keys()),list(distortations.values()))
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('Within-cluster SSE')
plt.show()
```

Clustering Output (2D Plots)

In [11]:
```
centroids = clustering.cluster_centers_
print('Clusters: {}'.format(centroids))
print('\n')
for i in range(len(centroids)):
    print('{}, {}'.format(centroids[i][1], centroids[i][0]))
print('\n')
t = np.arange(num_clusters)
for i in range(len(centroids)):
    print('{}: Cluster {}'.format(color_theme[i], i))

# Show centroid data in scatter plot
plt.scatter(centroids[:,1], centroids[:,0], marker="x", c=color_theme[t])
plt.title('Player Longitude vs Latitude: Centroids')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()

# Show clustering results in scatter plot
plt.scatter(x=loc_df.Longitude,y=loc_df.Latitude,c=color_theme[clustering.labels_],s=50)
plt.title('Player Longitude vs Latitude: Clusters')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

Clustering Output Aggregations

In [7]:
```
# Aggregate clustering result, group by cluster and count rows
cluster_counts = {}
for i in range(len(clustering.labels_)):
    if clustering.labels_[i] not in cluster_counts:
        cluster_counts[clustering.labels_[i]] = 1
    else:
        cluster_counts[clustering.labels_[i]] += 1
print('Totals:')
for i in cluster_counts:
    print('{} players were drafted in the top {} picks of their respective draft that fall in cluster {}'.format(cluste

print('Sum: {}'.format(sum(cluster_counts.values())))
print('\n')

# Further group cluster by decade
decade_counts = {1990: {}, 2000: {}, 2010: {}}
for i in range(num_clusters):
    decade_counts[1990.0][i] = 0
    decade_counts[2000.0][i] = 0
    decade_counts[2010.0][i] = 0

for i in range(len(clustering.labels_)):
    if (1990.0 <= loc_df_with_years.iloc[i][2] < 2000.0):
        decade_counts[1990][clustering.labels_[i]] += 1
    elif (2000.0 <= loc_df_with_years.iloc[i][2] < 2010.0):
        decade_counts[2000][clustering.labels_[i]] += 1
    elif (2010.0 <= loc_df_with_years.iloc[i][2] < 2020.0):
        decade_counts[2010][clustering.labels_[i]] += 1

# Output aggregations calculated for each year by cluster
# Including percentage of each cluster count for given decade and total sum of prospects in each decade
for i in decade_counts:
    print('In the {}\'s'.format(i))
    for j in decade_counts[i]:
        percentage = str(decade_counts[i][j]/sum(decade_counts[i].values()))[:5]
        print('    ({}) {} players were drafted in the top {} picks of their respective draft that fall in cluster {}'.
    print('Sum: {}'.format(sum(decade_counts[i].values())))
    print('\n')
```

Cluster Dendogram

In [8]:
```
# Execute cluster dendogram
Z = linkage(data,'ward')

plt.figure(figsize=(25,10))
plt.title('Clusters')
plt.ylabel('distance')
dendrogram(Z)

plt.axhline(y=15)
plt.show()
```

clustering-percentages.ipynb

Clustering Analysis (Shooting Percentages)

Import Modules

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import sklearn
from sklearn.cluster import KMeans
import sklearn.metrics as sm
from sklearn import datasets
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import scale
from scipy.cluster.hierarchy import dendrogram, linkage
%matplotlib inline
```

Import Excel Data

In [2]:
```
map = {}

# Loading Excel Data and adding each sheet into a map with its respective data
xls = pd.ExcelFile('nba_draft.xlsx')
for sheet_name in xls.sheet_names:
    map[sheet_name] = xls.parse(sheet_name)
```

Initialize Parameters

In [3]:
```
# Define number of clusters (K)
num_clusters = 3

# Define the maximum draft pick number to be included in the analysis
max_draft_selection = 60

# Define the positions of interest
positions = ['Center']

# Define start year of draft class range
start_year = 1990

# Define end year of draft class range
end_year = 2020

# Define color theme for scatter plots
color_theme = np.array(['red','blue','green', 'yellow', 'orange'])
```

Clean, Partition, Transform Data

In [4]:
```
df = []

# Concatenate all sheets in draft class range
for i in range(start_year, end_year):
    df.append(map['draft_data_{}'.format(i)])
percentages_df = pd.concat(df)

# Filter draft picks
percentages_df = percentages_df[percentages_df['Pk']  <= max_draft_selection]


# Filter positions of interest
if len(positions) != 0:
    percentages_df = percentages_df[percentages_df['position'].isin(positions)]

# Filter out outliers
percentages_df = percentages_df[percentages_df['college_3P%'] < 1]

# Get columns of interest
sg_df = percentages_df.iloc[:, 37:-12]
sg_df_with_years = percentages_df.iloc[:, list(range(37,40)) + [-3]]

# Drop null rows
sg_df = sg_df.dropna()
sg_df_with_years = sg_df_with_years.dropna()

# Perform K-means clustering and define feature variables
data = sg_df.to_numpy()
clustering  = KMeans(n_clusters=num_clusters,random_state=5)
clustering.fit(data)
sg_df.columns = ['FG','threePointPercentage','FT']
```

Elbow Method

In [5]:
```
# Execute elbow method to determine optimal K value
distortations = {}
for k in range(1,15):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(sg_df)
  distortations[k] = kmeans.inertia_

plt.plot(list(distortations.keys()),list(distortations.values()))
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Within-cluster SSE')
plt.show()
```

Clustering Output (2D Plots)

In [6]:
```
centroids = clustering.cluster_centers_
print('Clusters: {}'.format(centroids))
print('\n')
t = np.arange(num_clusters)
for i in range(len(centroids)):
    print('{}: Cluster {}'.format(color_theme[i], i))

# Show centroid data in scatter plot for FG% vs FT%
plt.scatter(centroids[:,0], centroids[:,2], marker="x", c=color_theme[t])
plt.title('Player FG% vs FT%: Centroids')
plt.xlabel('FG')
plt.ylabel('FT')
plt.show()

# Show clustering results in scatter plot for FG% vs FT%
plt.scatter(x=sg_df.FG,y=sg_df.FT,c=color_theme[clustering.labels_],s=50)
plt.title('Player FG% vs FT%: Clusters')
plt.xlabel('FG')
plt.ylabel('FT')
plt.show()

# Show centroid data in scatter plot for 3P% vs FT%
plt.scatter(centroids[:,1], centroids[:,2], marker="x", c=color_theme[t])
plt.title('Player 3P% vs FT%: Centroids')
plt.xlabel('3P')
plt.ylabel('FT')
plt.show()

# Show clustering results in scatter plot for 3P% vs FT%
plt.scatter(x=sg_df.threePointPercentage,y=sg_df.FT,c=color_theme[clustering.labels_],s=50)
plt.title('Player 3P% vs FT%: Clusters')
plt.xlabel('3P')
plt.ylabel('FT')
plt.show()

# Show centroid data in scatter plot for FG% vs 3P%
plt.scatter(centroids[:,0], centroids[:,1], marker="x", c=color_theme[t])
plt.title('Player FG% vs 3P%: Centroids')
plt.xlabel('FG')
plt.ylabel('3P')
plt.show()

# Show clustering results in scatter plot for FG% vs 3P%
plt.scatter(x=sg_df.FG,y=sg_df.threePointPercentage,c=color_theme[clustering.labels_],s=50)
plt.title('Player FG% vs 3P%: Clusters')
plt.xlabel('FG')
plt.ylabel('3P')
plt.show()
```

Clustering Output (3D Plots)

In [7]:
```
centroids = clustering.cluster_centers_
print('Clusters: {}'.format(centroids))
print('\n')
t = np.arange(num_clusters)
for i in range(len(centroids)):
    print('{}: Cluster {}'.format(color_theme[i], i))

# Show centroid data in 3D scatter plot
fig = plt.figure(figsize=(12,10))
centroid_3d = fig.add_subplot(111, projection='3d')
centroid_3d.scatter(xs=centroids[:,0],ys=centroids[:,1], zs = centroids[:,2], s=50, label=sg_df.columns, c=color_theme[
centroid_3d.set_title('Player Shooting Percentages: Centroids')
centroid_3d.set_xlabel('FG%')
centroid_3d.set_ylabel('3P%')
centroid_3d.set_zlabel('FT%')

# Show clustering results in 3D scatter plot
fig2 = plt.figure(figsize=(12,10))
ax = fig2.add_subplot(111, projection='3d')
ax.scatter(xs=sg_df.FG, ys=sg_df.threePointPercentage, zs = sg_df.FT, label=sg_df.columns, c=color_theme[clustering.lab
ax.set_title('Player Shooting Percentages: Clusters')
ax.set_xlabel('FG%')
ax.set_ylabel('3P%')
ax.set_zlabel('FT%')
```

Clustering Output Aggregations

In [8]:
```
# Aggregate clustering result, group by cluster and count rows
cluster_counts = {}
for i in range(len(clustering.labels_)):
    if clustering.labels_[i] not in cluster_counts:
        cluster_counts[clustering.labels_[i]] = 1
    else:
        cluster_counts[clustering.labels_[i]] += 1
print('In Total:')
for i in cluster_counts:
    print('{} players were drafted in the top {} picks of their respective draft that fall in cluster {}'.format(cluste
print('Sum: {}'.format(sum(cluster_counts.values())))
print('\n')

# Further group cluster by decade
decade_counts = {1990: {}, 2000: {}, 2010: {}}
for i in range(num_clusters):
    decade_counts[1990.0][i] = 0
    decade_counts[2000.0][i] = 0
    decade_counts[2010.0][i] = 0

for i in range(len(clustering.labels_)):
    if (1990.0 <= sg_df_with_years.iloc[i][3] < 2000.0):
        decade_counts[1990][clustering.labels_[i]] += 1
    elif (2000.0 <= sg_df_with_years.iloc[i][3] < 2010.0):
        decade_counts[2000][clustering.labels_[i]] += 1
    elif (2010.0 <= sg_df_with_years.iloc[i][3] < 2020.0):
        decade_counts[2010][clustering.labels_[i]] += 1

# Output aggregations calculated for each year by cluster
# Including percentage of each cluster count for given decade and total sum of prospects in each decade
for i in decade_counts:
    print('In the {}\'s'.format(i))
    for j in decade_counts[i]:
        percentage = str(decade_counts[i][j]/sum(decade_counts[i].values()))[:5]
        print('    ({}) {} players were drafted in the top {} picks of their respective draft that fall in cluster {}'.
    print('Sum: {}'.format(sum(decade_counts[i].values())))
    print('\n')
```

Cluster Dendogram

In [9]:
```
# Execute cluster dendogram
Z = linkage(sg_df,'ward')

plt.figure(figsize=(25,10))
plt.title('Clusters')
plt.ylabel('distance')
dendrogram(Z)

plt.axhline(y=15)
plt.show()
```