Allegheny College
Department of Computer Science

Senior Thesis

NBA PyAnalysis: Using machine learning to predict how much NBA players are worth in free agency

by

ALLEGHENY COLLEGE
Caden Hinckley

COMPUTER SCIENCE

Project Supervisor: **Professor Kapfhammer**
Co-Supervisor: **Professor Nonnenmacher**

22 July 2022

Abstract

There is a big knowledge gap in the NBA with trying to figure out how much a player is worth. Front office personal have been trying to figure out that gap for many years. Using past research and computer science I will try and fill that knowledge gap by implementing a machine learning program that will group free agents with players in the league with similar VORP and age in order to figure out how much that how much that free agent is worth. Overpaying players either its in free agency or re-signing a player hurts the teams chancing of winning the championship, that is why it is so important to make sure the contract is the right amount for a player. I used a tool in the Python programming language called pandas in order to create my table of data in order for the machine learning algorithm to produce the desired output. This will take out the guess work that happens during free agency and limit the amount of contracts that turn out to be a poor waste of resources, which is very limited in the NBA due to the salary cap that is implemented by the CBA.

Table of Contents

List of Figures

Introduction

Motivation

I have always been very interested in understanding how the front offices for NBA teams knew how much money players should be given in free agency. Free agency has changed the landscape of the NBA and has influenced many aspects of basketball. However, no one is able to know how much a player is truly worth. This is a problem in the NBA because they have a salary cap, which means that they only have a limited amount of money to give to players, and it is very hard to win basketball games when you have allocated resources to a player that isn't playing up to the size of their contract. The salary cap for the 2022-23 season is $123.655 million. There are ways to be able to go over the salary cap, if a team does that, it's called going into the luxury tax. "These teams pay a penalty for each dollar their team salary (with a few exceptions) exceeds the tax level." [4] These rules disincentive teams from going into the luxury tax because it is very costly to do so. Only teams with a good chance at winning the finals think about going into the luxury tax. If a team didn't think they had a good chance at the finals, the owners wouldn't want to give away that much money for an average to below-average team.

My original plan was to use SQLite to create a database where I could run queries to find out how much a player should be worth in free agency according to their VORP(Value over Replacement Player). I chose SQLite in the beginning because that is a database system that I have had the most experience with and felt most comfortable using However, after discussing my project with my first reader, we decided to use a Python data analysis library called pandas and Scikit-learn. Pandas is a super useful library when trying to filter through lots of data. Pandas can make the data very flexible and able to manipulate the data in the way that is needed. For example, I

was able to combine two CSV files that contained player statistics and player salaries in order to get one table as output. This was a very important step in my project because it saved me from having to manually enter each player's salary. Being able to use Python is extremely helpful because of all the data libraries like pandas, that I have been able to use in order to implement my program. I also wanted to have a visual to show that I was able to combine both CSV files that had player statistics and players' salaries; in order to do that, I used Matplotlib in order to graph the player's VORP and salaries. This was a very interesting visual because I was able to see that there were a lot of players with low VORP getting paid more than players that have a higher VORP. This was able to show to me that there are players who a getting overpaid for their production and that teams had made bad investments in those players, and they are now paying an average to below-average player the same as some of the best players in the NBA.

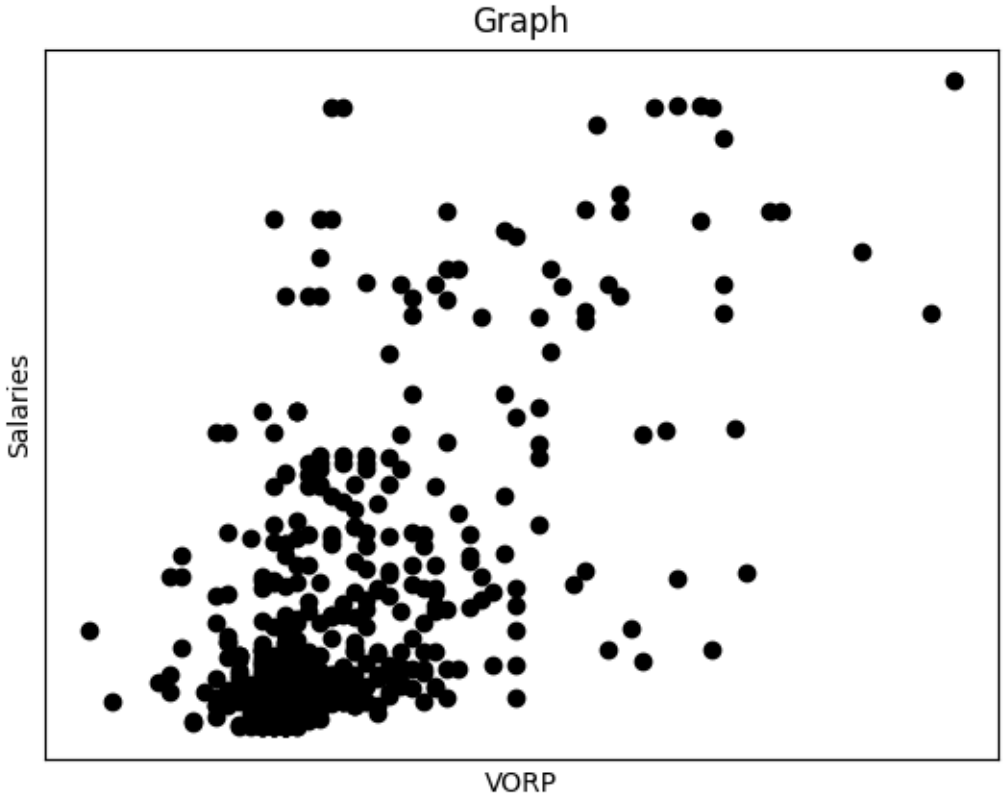

Figure 1: Players_VORP_And_Salaries

The book and movie Moneyball looked into how the Oakland Athletics tried to create the best possible team with their limited resources. The Oakland A's are a poor team compared to other teams in the MLB. So the General Manager, Billy Beane, tried to find ways to gain an advantage on teams like the Yankees and the Red Sox that had the ability to pay players a lot more than the A's could. They were able to find ways to compete with these bigger teams by finding out what is important in terms of winning baseball games. This ushered in a new way of baseball statistics that changed how every baseball team looks at players. NBA front offices have started to try and implement more statistics in order to find an advantage and win more

games and have a better chance of winning the NBA finals. After reading the book I have wanted to try and see if I could find a way that would help teams find a way to get players that outperformed their contract or be able to find a way for teams to not pay players more than they are actually worth. Billy Beane and the Oakland A's were able to improve their team through free agency by finding valuable players that didn't fit the mold of the MLB prototypical player. This allowed the A's to gain a competitive advantage over the rest of the league, while paying players cheaply, and caused a massive wave in the MLB of teams looking at the data instead of just looking at the players.

After reading Moneyball, I started to think about how other professional leagues are going to find ways to gain a competitive advantage like MLB teams. The NBA has started to implement analytics in its decision-making but still hasn't been as impactful as it has been in the MLB. Since the MLB is at a much slower pace and every play has a clear start and finish it is easier to parse through the data and clearly see what is happening in the MLB. But since the NBA has a quicker game state, it is a lot harder to implement the same analytics that they do in the MLB, so different and new analytics have been starting to be used by the front office in the NBA. VORP, which stands for Value Over Replacement Player is one of the Advanced Statistics that have been implemented to try and get a better understanding of the game and how players affect the game both positively and negatively. VORP is very similar to WAR, which is a baseball statistic that stands for Wins Above Replacement. There are several other advanced statics in the NBA that help front offices better understand the impact players have such as TS%(True shooting percentage) and USG%(Usage percentage). True shooting percentage is a measure of shooting efficiency that takes into account 2-point field goals, 3-point field goals, and free throws. Usage percentage is an estimate of the percentage of team plays used by a player while they were on the floor.

I chose to focus on VORP because it shows how players affected the game on both the defensive side and the offensive side. It also incorporates Box Plus Minus, which is a stat that shows how well the team did when you were on the court to see how you affected the game. I found all the player statistics on a website called Basketball-Reference, this helped me proceed with my project because the website has all the statistics that I will need to use in order to complete this project. Basketball-Reference has every single player who has ever played in the NBA and their stats in an easy-to-export

format which allowed me to convert it into a CSV format. The CSV format allows me to write code that is able to parse through the data.

There are three ways NBA teams can improve their roster: trades, the draft, and free agency. Trades are the most difficult to make happen because two different teams have to mutually agree to the trade. The NBA Draft is only two rounds, so it is very hard to make a lot of changes to your roster through the draft with only two rounds in the draft. Free agency is how teams can add players to their roster with NBA experience. Since NBA teams have a limit on how much they can spend on the total roster teams, have to be smart with there money. Teams cannot go and sign every free agent they want, they have to identify the weaknesses in their team and try and find players that will be able to help solve some if not all of those weaknesses. That is why it is very important to have players on your team that aren't affecting your ability to upgrade the team because they are on a contract that is more than they are worth. The best teams in the NBA has players on their team that are overproducing their contracts, which is why it's so important to make sure the worth of the free agent.

While free agency is one of the most important aspects of team building, it is also very hard not to make mistakes in signing players. There isn't a way to make sure that signing a player to a contract will turn out to be the right decision. Sometimes a player doesn't fit into the system the team is trying to run, so they become less effective on their new team than on their old team. A player that you thought would just be a role player might turn into one of the best players on your team and is helping the team win a lot of games. This is why it is very important to do everything possible in order to make the right decision because a decision could either help or hurt your team. That is why teams are using data to drive their decision-making; it is less prone to errors.

There are a lot of aspects that go into an NBA team deciding to sign a player in free agency other than just the basketball aspect, such as: work ethic, fit with the rest of the team, character, and reputation. So the front office of NBA teams can't just look at a stat sheet and decide if a player would be a good fit for the team. That is why the best teams don't just have one simple approach to finding players that they will sign in free agency. The best teams, like the Warriors and the Celtics, use advanced statistics along with the more conventional ways in order to decide if that player would help the team.

Free agency has become a very important way for teams to improve their

roster year after year. But for a while, the NBA didn't allow free agency, so players would have to play for the team that drafted them, which obviously didn't allow for a player movement like the kind we see today. It wasn't until the summer of 1988 that free agency was allowed due to the signing of the new Collective Bargaining Agreement(CBA) that was agreed upon by the NBA owners and the NBPA(National Basketball Players Association). Before this collective bargaining agreement was implemented, players struggled to move between teams. Oscar Robertson, an all-time great basketball player that played from 1960-1974 and was a 16-time All-Star, 11-time All-NBA, and 1-time NBA champion who paved the way for free agency when he sued the NBA. "Robertson v. National Basketball Association was a class-action lawsuit filed in 1970. Robertson, at the time, was president of the National Basketball Players Association" [11]. Oscar Robertson's goal in the lawsuit was to get players playing conditions improved and for players to be able to decide where they want to play. Before this lawsuit players would be stuck on the team that drafted them until they got traded on the team didn't want them anymore. Between 1972 and 1988, players still didn't switch teams in free agency because of their old team's "right of first refusal" to match any offer they might receive.

The CBA is a very important part of the NBA because without the CBA being agreed upon, there wouldn't be any NBA season. "The Collective Bargaining Agreement between the NBPA and the NBA sets out the terms and conditions of employment for all professional basketball players playing in the National Basketball Association, as well as the respective rights and obligations of the NBA Clubs, the NBA, and the NBPA. The current Agreement was ratified by NBPA membership in December 2016, took effect on July 1, 2017, and runs through the 2023-24 NBA season" [2]. If the three parties aren't able to come to an agreement there isn't a season until they come to an agreement; the last time that happened was 2011, which was only the fourth time in NBA history. There were several reasons for the strike in 2011: salary cap, luxury cap, and the distribution of revenue.

The History of the CBA

The history of the CBA is so important because it gives a great background on how the NBA and its players have been having a constant fight about pay and rights. Without knowing about the CBA it would be very hard to

understand how the salary cap came to be and how much the players had to fight in order to get the rights they currently have today.

The CBA was created in 1954 when Bob Cousy started to organize the NBPA. He wrote letters to players on each team "seeking their input and support for a formal union to represent players' interests" [2]. When Bob Cousy became the president of the NBPA, he went to the NBA President Maurice Podoloff with demands that he and the players had. The NBA refused to recognize the union and refused all the demands the players brought forward, it wasn't until Bob Cousy met with the AFL-CIO(American Federation of Labor and Congress of Industrial Organizations) that the NBA started to talk with the NBPA. In April 1957, the NBA Board of Governors formally acknowledged the NBPA and agreed to the requests: "An abolition of the whispering fine, A $7 per diem and reasonable traveling expenses, An increase in the 1957-58 playoff pool, Reasonable moving expenses for players traded during the off-season, Referral of player-owner disputes to the NBA League President or a committee of three NBA Governors chosen by the players; Elimination of exhibition games within three days of the season opener, and Regular players not required to report to training camp earlier than four weeks prior to the season" [2].

After battling for a pension for retired players, in 1967, Oscar Robertson, newly elected NBPA president, was able to come to an agreement that "A $600 a month pension plan for all players with ten years of service and over age 65" [2], along with several other important terms: "New medical and insurance benefits, Negotiations for exhibition game pay, An 82-game limit on the regular season, The elimination of games played immediately prior to the All-Star Game, A new committee to review the standard player contract prior to the 1967-68 season". [2]).

In 1967 the NBA started competing with a new professional basketball league called the ABA. This increased the amount of money that the players were getting paid. The NBA wanted to merge with the ABA to decrease competition and lower the players' salaries. "The players filed the "Oscar Robertson Suit" under the antitrust laws in 1970" [2]. They hoped that this would block the merger and end the players being stuck with being unable to leave the team that drafted them. The NBA decided to negotiate with the NBPA in order for the merger to go through. The owners pay the 500 players a $4.3 million settlement and the union $1 million for legal fees. This was dependent on the "Oscar Robertson Suit" being dropped. The NBA and ABA finally merged, but the minimum salary was raised from $200,000 to

$300,000. They also agreed to better dental/medical coverage along with a better pension.

Another landmark agreement was agreed to in 1983, a four-year agreement in which the league and players agreed to share revenue and a salary cap. This helped raise salaries for everyone, teams couldn't just pay the top players a lot of money and keep paying role players the low salaries they were receiving. Due to this change, the minimum salary was raised to $40,000.

Jump forward to the conclusion of the four-year agreement, going into discussion with the NBA and their owners about reaching a new agreement. The players felt that they deserved a bigger portion of the revenue that the league was getting. The president of the NBPA at the time was Junior Bridgeman, who became president in 1985 and filed a lawsuit against the NBA(the Bridgeman antitrust suit). After a favorable ruling, the NBA decided to negotiate with the NBPA. They agreed to a 6-year agreement that included the following demands: "The elimination of the right of first refusal after a player completes his second contract, with unrestricted free agency for veteran players, the inclusion of five-year veterans who finished their careers prior to 1965 in the pension plan, and a reduction of the college draft to three rounds in 1988 and two rounds in 1989" [2]. In 1991, the NBPA thought that the NBA and the owners were hiding the true amount of income that was coming into the NBA in order to lower the salary cap artificially and players' wages. The NBPA thought that the NBA wasn't counting the international TV contracts towards revenue, arena signage, and luxury suites. The NBA reached an agreement that saw them pay $62 million for the players. This did affect the trust and relationship between the NBA and the NBPA.

After the 1995 NBA finals, the owner did something unprecedented and imposed a lockout, meaning there were no basketball activities nor pay. The union was fighting two battles, one against the NBA and the owners and the other within the NBPA itself. The owners wanted to make changes to the salary system that was in place and weren't going to back down. Within the union, it had two very different viewpoints on how to handle the lockout best. One group thought that the best way to handle the pressure the owners were putting on them was to decertify the union, take the owners to court, and leave it up to the court's view of the antitrust law. The other side of the NBPA didn't think decertifying was a good idea. The NBA and the owners didn't want to have to face litigation if the NBPA decided to decertify and go to the courts; eventually, the owners caved and modified their demands. There were several key agreements that are still influencing the game today.

First, they agreed to a preset wage-scale, a rookie pay scale. The most important part of the agreement is known as the "Larry Bird exception" or "Bird rights," which allows teams to go over the salary cap to re-sign players heading towards free agency. This helps incentives teams to draft well because they will be able to exceed that salary cap in order to keep their players. Thirdly, both parties agreed to reduce the amount a multi-year contract went up from 30% to 20%. This means that if a player was under contract for four years and $100 million dollars after that contract was over, that player could resign for another four years but could only increase their salary by 20%. Lastly, they also agreed to a new revenue-sharing formula.

It didn't last long until the owners decided to exercise their option to terminate the CBA in March 1998. By then, around 400 NBA players were collectively making over a billion dollars. The owners wanted to make the contracts for players not guaranteed anymore and create a hard salary cap. If the players agreed with that, the players that make had contracts that made them the "middle class" of the league would be wiped out and have to sign for a lower salary that's not guaranteed. This holdout threatened the NBA from having a season, but both sides were able to come up with an agreement on the day before the deadline to have a season. This move by the owner was meant to try and lower the salaries of NBA players; however, the move didn't work the way the owners envisioned. By 2004, the last year of the signed CBA, the average player's salary was over $4.5 million dollars. And the players as a whole, so an "80% increase in salaries and benefits"[2].

The most recent lockout happened in 2011 when NBA commissioner David Stern notified the NBPA that there would be no extension of the current CBA. The owners were trying to make a series of drastic changes to players' salaries due to the harsh economic decline due to the 2008 recession. The owners wanted to have "rollbacks including 40% reductions in the value of all existing and future contracts" [2]. In order for the players to not have to fold to the owner's demands, the NBPA decided to "relinquishing its status as the players' exclusive collective bargaining representative" [2]. After disbanding the NBPA, players in Michigan and California brought antitrust lawsuits against the NBA. The owners once again decided to settle the cases instead of leaving it up to the courts. Once those lawsuits were closed, the players voted to re-create the NBPA and signed the CBA on December 8th, 2011, 161, after the lockout took effect.

The Collective Bargaining Agreement is very important for the owners and players to agree upon terms for the NBA season to be played. Since

the NBPA has been able to stand as one in order to fight for the rights and rules that have made the NBA different from other leagues by having their contracts fully guaranteed. This differs from the NFL, where players can sign big-money contracts but aren't guaranteed to get all the money.

Current State of the Art

Most NBA teams have their own data analytics department that tries to find competitive advantages to improve their team. The Phoenix Suns, under head coach Mike D'Antoni, in the mid-2000s, tried to create a competitive advantage by coming up with a system called "7 seconds or less," which was an offensive strategy to focus on getting as many shots in a basketball game as possible. They wanted to shoot the ball within seven seconds of getting it, which would create more possessions and, therefore more opportunities to score. That style of basketball didn't become mainstream in the NBA, so when Mike D'Antoni became the head coach of the Houston Rockets, he tried a different offensive strategy. With General Manager Daryl Morey and coach D'Antoni, they came up with a system that tried to only shoot 3-pointers and lay-ups. They found that shooting 2-pointers that weren't layups was very inefficient, low percentage shot, and they would score more if they only shot 3-pointers and layups In order for this system to be successful, Daryl Morey had to go out and find a lot of shooters that they could sign or trade for. This system has spread league-wide, and very few players shoot 2-pointers. Teams will continually try to gain a competitive advantage, whether through an offensive or defensive system or through acquiring players.

Goals of the Project

The goal of this project is to find out how much a player in the NBA is worth. If this program were to be implemented by an NBA front office, it would allow them to see how much a player should get paid. This would limit the number of contracts that are considered bad. In the NBA, there are times were teams pay a player a lot of money because they think it will help the team win more, but sometimes the players don't live up to the contract they signed and are hurting the team because they are talking about a big portion of the salary cap. With my program, it would make overpaying players less

likely to happen, which would help NBA teams not have players on contracts that they aren't living up to.

Ethical Implications

My project doesn't have many ethical implications due to the fact that players' salaries and statistics are freely available to anyone who wants to see them. However, there are still a couple of ethical implications to be aware of. One is Algorithmic Bias and the other is Data Collection Issues. Algorithmic Bias is a concern for this project because I will be using a machine learning algorithm to see how much an NBA player is worth, and it is possible that algorithm might create biased outcomes that would affect the results and cause them to be incorrect. The other ethical concerns about this project are Data Collection Issues, this is because of all the data that is needed for this project, there could be issues with the data that could skew the results.

Related Work

Teams in the NBA have been trying to figure out how much a player is worth. This question has been eluding people since the start of the NBA; people have tried to figure out how to find ways to get as close as possible in order to make good decisions and find ways not to overpay players. Since the NBA has a salary cap, it is really important not to overpay players because it will hurt your team and make it hard to make changes to the roster to improve the team. One reason it's so hard to figure out how much a player is worth is that players don't always play the same season after season. Players' progressions are not always linear; they tend to go up and down, which makes it hard to know how much a player is worth. Advanced statistics have become more used in the NBA, and those stats are able to tell a more descriptive story about the level certain players are at. I am going to implement a Python programming language machine learning program that is going to take in all the stats from the NBA using basketball-reference an online database and figure out how much a player is worth based on previous seasons and looking at VORP. The advanced metric called VORP, which stands for Value over replacement player, is a very helpful stat that uses Box Plus-Minus(BPM) and minutes played in order to find out how a player compares to a "replacement level player". The replacement level player is given a VORP of -2.0, so if you have a VORP greater than -2.0, you are more valuable than a replacement player.

One problem many teams have is ensuring that players are properly motivated and incentivized to play hard and perform at the top of their potential. In a paper written by Berri et al.[3]. The authors wanted to see if players have a tendency not to play as hard as they can if they have a contract that still has multiple years left on it. This is a huge problem for NBA teams because they are paying players to play their best, but since NBA contracts are almost always guaranteed, it turns out that 58% of the players in the NBA

see a decline in their stats after signing a new multi-year contract. This is very concerning for NBA front offices because how do they ensure that the players they sign will continue to play at a high level? NBA teams should look at players' stats over a long period of time in order to see if there are any trends that they can see. This is very important because of what Stiroh, K. J. discovered in his article [12]. Stiroh found that players tend to try much harder when they don't have a multi-year contract. Players with only one year left on their contract tend to do better than they do when signing a new multiyear contract. This makes decision-making very hard for NBA front office personnel because how do they know which version of the player they will get after signing a contract? This is why it's extremely hard to predict if a player is
worth a big contract if they will just shrink after signing a new deal. This is why some teams use incentives to try to get the best out of their players, even if they have multiple years left on their contracts.

Since players tend to shirk when they have multi-year contracts, teams have to predict which version of the player will get. One option would be to have a contract with many incentives and a low base salary; however, players don't like those kinds of contracts. The Brooklyn Nets tried to do this for Kyrie Irving in the 2022-23 off-season after a season that saw Kyrie not sign a long-term contract that would have been very heavy with incentives because Kyrie has a reputation for not playing that many games for multiple reasons: Vaccine mandate, suspensions, and injuries. If he had signed the contract, it would have been equivalent to a max contract if he had met the incentives. However, he decided not to sign that contract and his player option, which was one year, $36,934,550. This was a bet on himself that he would get a long-term contract next off-season that would have either all guaranteed money or major of the contract will be guaranteed. So how can teams ensure that they pay the right amount for players if they can't use incentives; one option would be to find a way to predict salaries or find cheap players that will be able to help the team.

Wu, W. et al.[13] tried to find if it was possible to find cheap players that would help teams win. Sports Analytics Group at Berkeley, The author realized the importance of filling out your roster behind the best players. Most teams in the NBA have 1 or 2 max players. Max players take up a huge chunk of the salary cap, making filling out the rest of the team very important to create a competitive team that can compete for the NBA title. The authors looked at a group of statistics that included advanced statistics.

The authors showed lots of examples of teams trying to add talent to their team, but it clearly didn't work out and hurt that team's ability to add talent because they used their salary cap to try and pay players lots of money that weren't worth the money that they were getting paid. The authors tried to develop a model that could predict a player's worth and help identify players who could help the team without taking up a big part of the salary cap. Wu, W. et al. model didn't work in the way that they wanted. Their model was more skewed to volume-scorers over role players. One possibility was that they used stats that overlapped, they used points, rebounds, and VORP, but VORP, in certain ways, incorporated points and rebounds. Since VORP has BPM (box-plus minus), which will be affected if you score and get a lot of rebounds, the box plus-minus will be higher, which would cause the VORP to be higher. This might be one reason the results didn't match the desired output the authors wanted. For my senior project, I don't want to look at too many stats in order to avoid any correlation between the different stats I would look at. Since advanced stats tend to incorporate a lot of different stats together, I don't want to use conventional stats along with advanced stats. Along with the players, on-court performance, the Collective Bargaining Agreement(CBA) also factors heavily in how much players get paid.

The CBA plays a large part in how salaries are distributed across the players in the NBA. The NBPA and its owners decide the structure of contracts, and that is exactly what Hill et al. looked at[6]. Salary distribution and collective bargaining agreements: A case study of the NBA.
This paper wanted to understand the salary structure due to the collective bargaining agreement, and they had some very interesting results. They found that NBA had a very interesting contract structure that was eye-opening. Hill et al.[6] found that "approximately 25 percent of the players in the league, but their compensation accounted for only 12 percent of the total compensation in the league. On the other hand, the top 5 percent of the players in the league accounted for approximately 18 percent of the total compensation" (Hill et al.[6]). This salary separation is quite staggering since a large portion of the league gets paid so little, and such a small percentage of the league makes up much of the salary cap. There is some simple reason behind those numbers; one might be that players with more household names create more revenue for the league and make a big portion of the salary cap.

Another reason is that players in the league have more say in how the salary cap is distributed for several years because they have been part of the

NBPA for a while and have more say than the rookies. This was an important article because it showed that rookies might be considered undervalued because the rookie pay scale is very low, and if a player is a very good player as a rookie and has a high VORP, that will show that he is outperforming his contract by a lot. This helped illustrate that I need to be aware of all the contract rules that are in the NBA's collective bargaining agreements because certain players might be outperforming their contract but can't be paid any higher. For example, Kevin Durant was paid $42,018,900 last year by the Brooklyn Nets and was underpaid for the money and attention he brought to the Brooklyn Nets. But he can't get paid any higher because of how contracts are constructed. The way the contracts are set up, if players are able to win certain types of awards after the season, such as All-NBA, they can increase the potential value of their next contract. The CBA is how the salary structure of the NBA is decided, and Hastings[5] wanted to see the distribution of rents in the NBA since the CBA agreement in 1999.

In Hastings et al.[5] paper, they talked about how the agreement between the NBPA and the NBA owners limited the amount of money that max players could sign for. Before this agreement, players could sign for any amount of money if the team was still under the salary cap. But after this agreement, there is now a limit on how much a specific player's salary can take up the salary cap, with max-level players making less money, which allowed other players on the team to make more money. In the paper, they found that the money was given to the team's second and third-best players. One way that players can increase their salary is by making an All-NBA team, So making those teams are very important for players to be able to increase their next contract, so João Vítor Rocha da Silva, & Paulo Canas Rodrigues[7] wanted to see how players were chosen for those teams.

In João Vítor Rocha da Silva et al.[7] found some very interesting results about how NBA awards are given out. The people that vote for the awards are journalists, but they questioned how accurate the journalists were when it came to voting for awards that could affect a player's salary. Did the journalists get caught up with storylines about players deserving the awards, or did the journalist pick the right people? The authors used the LASSO regression model in order to predict the players for the 3 All-NBA teams. A LASSO regression model is short for Lasso Least Absolute Shrinkage and Selection Operator. The results of the article showed that based on the outcome of the LASSO regression, the journalists didn't correctly vote for the players that deserved to be on one of the three Al-NBA teams, which

means that players that deserved to be part of that team didn't because the journalists didn't pick the right players. This shows that people who follow the league every day and have seen almost every player still don't know who are the best players in the NBA. This shows that the 'eye test' is not very accurate, and using all the resources available in order to make a decision would lead to greater accuracy of correct decisions.

If a model can be accurate at predicting who should win awards, then there should be models that can estimate a player's salary based on on-court performances, which is exactly what Papadaki and Tsagris[10] did in their 2020 paper. In order to estimate the player's salary, they needed to decide on what stats they would want to be in their machine learning approach. They landed on three statistical categories that they thought would be the best to have in their model, and they were: points per game, rebounds per game, and assists per game. Solely based on those three statistical categories, when they ran their non-linear models, they were able to predict players' salaries accurately. They stated that they were looking for a .6 or .7 coefficient of determination in order to classify as predicting the salaries as 'correct'. The coefficient of determination is a method of test the accuracy of the regression is; a high score would reflect high correlation and a low score reflects a low correlation. They were able to get a reasonably high coefficient of determination just by looking at the offensive side of the game, but since the game is played both offensively and defensively, it is essential to look at the defensive side of the game as well.

Assani et al[1] looked at both the offensive side and defensive side. These papers were very interesting because of how different it was from the other papers I have read because of the author's approach. Assani, Asghar, and Yang used DEA, which is a mathematical approach based on linear programming used to evaluate the relative efficiency of a set of homogeneous DMUs. In this case, the NBA players were the DMUs. Using this approach, they found some very interesting results in their study, they found that overall efficiency is low when offensive efficiency is high. This potentially shows why certain players in the NBA have been as successful at winning as people think they should. For example, when James Harden was a member of the Houston Rockets, he was the best offensive player in the NBA for several years; he averaged 30 points per game for three straight years but was known for not playing good defense. The Houston Rockets were amongst the top teams in the NBA for those years, but they could never find a way into the NBA finals always fall short in the playoffs year after year. People questioned

why the Rockets couldn't find a way to when, but this paper showed players with high offensive efficiency have a lower overall efficiency. Based on this paper, teams should try and find players with high overall efficiency, even if it means their offensive efficiency is high. Another thing to remember is how front-office personnel and coaches behave when a player they just signed in free agency is playing poorly.

Quinn Keefer[9] wanted to figure out if coaches and front office personnel are able to find a way to fix a mistake by playing other players if a newly signed player is playing poorly or if they are stubborn and continue to play the player that is struggling. In his paper titled Sunk costs in the NBA: the salary cap and free agents he looked at the 2015-16 season and players who signed in free agency for the 2016-17 season; because of the new TV deal, salaries went up in the 2016-17 season. Compensation due to the new TV deals rose by 81.7%. Quinn Keefer used a difference-in-difference method in order to find the results, and he found that an 87.1% increase in salary leads to an additional 1.93 minutes per game. While that doesn't seem like much but if you play all 82 games in the season, that player will play an additional 158.26 minutes for that team. Even if the player isn't playing that well, teams don't tend to stop playing those players. With those extra minutes over the course of the season, the player might be able to pad their stats, so their stats don't look at bad as they indeed are.

I wanted to look at every possible reason why there is any sort of wage discrimination in the NBA. At first, I thought there wouldn't be any wage discrimination in the NBA since 71.8% of players are African-Americans. However, I read a paper by Johnson and Minuci[8] and the authors found some interesting results. Using a weighted linear wage model controlled for several aspects, such as player performance and a team, they found that black NBA athletes are, on average, underpaid by 13.1% compared to their counterparts. When I read this article, I was shocked at the findings because I assumed there wouldn't be any wage discrimination because teams want to have the best chance at winning and wouldn't care about race. This is very important for me to keep in mind when I run my machine learning algorithm in order to see how much a free agent is worth because I am comparing free agents to players with contracts, so a player that is black could be predicted to have a high salary but might not get it since there does appear to be wage discrimination in the NBA.

These articles have given me a clear insight into some possible areas to focus on, like incorporating defensive metrics in my machine learning in order

to get the most accurate results. Without these papers, it would have been very hard to understand the research that has already been done trying to predict salaries, and they give me a good clear picture of what can be added in this field of research. I will go more in-depth about my machine-learning algorithms in the next section to better understand how I conducted my research and got my results.

Method of Approach

In order to be able to test one of my core questions, does VORP have an effect on player salaries, I had to build a program that would take in a large dataset that I compiled from 2 different online databases. The first database I used was called basketball-reference; this was a very important part of this project because it contains all the stats from the NBA throughout its entire history of the NBA. It allowed me to go to the past season and find all the necessary information regarding players' stats in order to give my machine learning algorithm data points to train on. It was also super helpful because they had the functionality to convert the dataset into a CSV file, allowing me to transfer the datasets into a folder I used to train the model on. The other database that I used was SPOTRAC which is also an online database that keeps track of every player's salary in the NBA. This was a great resource because it gives the salary for every player that signed a new contract that year, so all I had to do was create a spreadsheet that had the players for the year they signed the contract and the amount of salary per year. I wanted to be able to connect the basketball-reference spreadsheet and the SPOTRAC spreadsheet and create one table that had the players' information, including:

- Salary
- Year of the signed contract
- VORP
- Age

A	B	C	D	E	F	G	H
Player	Pos	Age	Type	FROM	TO	Years	Value
Alec Burks	SG	29.3>	UFA	PHI	NYK	1	6000000
Alex Len	C	27.4>	UFA	SAC	TOR	1	2258000
Anthony Davis	C	27.7>	UFA	LAL	LAL	5	189903600
Aron Baynes	C	33.9>	UFA	PHX	TOR	2	14350000
Austin Rivers	SG	28.2>	UFA	HOU	NYK	3	9975000
Avery Bradley	PG	29.9>	UFA	LAL	MIA	1	5635000
Bismack Biyombo	C	28.2>	UFA	CHA	CHA	1	3500000
Bobby Portis	PF	25.8>	UFA	NYK	MIL	2	7427150
Bogdan Bogdanovic	SG	28.2>	RFA	SAC	ATL	4	72000000
Brandon Ingram	SF	23.2>	RFA	NOP	NOP	5	158253000
Bryn Forbes	SG	27.3>	UFA	SAS	MIL	2	4791147
Carmelo Anthony	PF	36.4>	UFA	POR	POR	1	2564753
Chimezie Metu	C	23.7>	UFA	SAS	SAC	3	5294220
Chris Boucher	PF	27.8>	RFA	TOR	TOR	2	13520000
Christian Wood	PF	25.1>	UFA	DET	HOU	3	41000000
D.J. Augustin	PG	33.0>	UFA	ORL	MIL	3	21000000

Figure 2: Table used by the machine learning algorithm

After building that spreadsheet, I needed a way to structure my data so that I could write a Python program to train on a large subsection of the table and predict the 2022 NBA free agent contract, in order to complete this goal, I needed to use Pandas. Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool built on top of the Python programming language. I wanted to sort my data set by the year they signed the contract, and using pandas, allowed me to do just that. Once I was able to sort the table, I needed to figure out which regression model I wanted to use. There were two regression models that I thought would be good to implement and see which one was better:

- Lasso regression
- Linear regression

Before I can test which regression model is the best to use, I have to import several libraries and tools to be able to use tools like Pandas and Matplotlib. With a couple of import statements, I was able to get the following:

- Pandas
- Matplotlib

- Scikit learn
- Linear regression
- Lasso regression

In order to test which regression model would be better to use, I had to implement both and see which regression had a better Coefficient of determination and Mean squared error. In order to implement the regression models, I had to use a tool from Scikit-learn, in which I could run both regression models and see the results. Scikit-learn is a machine learning, predictive data analysis tool that I leveraged in order to predict the 2022 free agent salaries. The Scikit Learn package was the most important tool I utilized to complete this project.

Before this project, I had never worked with any machine learning tool, so I had to learn how to work with Scikit learn. Scikit Learn has great documentation showing how to use the linear and lasso regression models. With this documentation, I learned how to implement those regression models. A lasso model is a linear model that estimates sparse coefficients. When you run the lasso regression, you need to import the lasso functionality `from sklearn import linear_model`, which allows you to start implementing the lasso regression. In order to get the Lasso regression model, you have to have another import statement `from sklearn.linear_model import Lasso`, which allows you to run the Lasso regression, and then you have to train the Lasso regression with a portion of the dataset. In order to complete this, you need to tell lasso what you want to train: `lasso.fit(X_train, y_train)`. After you train the model, it is ready to predict the salaries that the players should get based on their VORP.

After doing some research about the best machine learning techniques on the Scikit learn website, I came across the Lasso regression. I wanted to test multiple different regressions using the Scikit learn package in order to see if there was a specific regression model that would provide a more accurate prediction of player salaries. Lasso stands for Least Absolute Shrinkage and Selection Operator. The benefit of using a Lasso regression is that it can stabilize the linear regression by making it more robust against outliers and overfitting. In order to calculate a Lasso regression, you take `the sum of the squared residuals + lambda * |the slope|`. Lambda can be any value from 0 to positive infinity and is determined using Cross Validation. Cross-Validation is a technique in which one trains their model using the subset of the dataset and then evaluates using the complementary subset

of the dataset. Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. L1 regularization addresses the issue of over-fitting; overfitting is the production of an analysis that corresponds too closely to a particular set of data and may therefore fail to fit additional data or predict future observations reliably. This was one reason I wanted to run a Lasso regression and compare it to a simple linear regression like OLS and see if there was any difference in predicting player salaries.

Code setup

$$\min_{w} \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha ||w||_1$$

Figure 3: Lasso minimize function

In order to implement the Lasso regression into my code after I imported the package to call the Lasso function:

```
lasso.fit(X_train, y_train)
```

```
▾ Lasso
Lasso()
```

Figure 4: Lasso call

The next step was to find out the Coefficient of Determination(R^2 or r-squared), which is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable or how well the data fits the model. In order to complete this process, I had to leverage the Scikit learn to package and find the coefficient of determination.

```
"Coefficient of determination: %.2f" % r2_score(y_test, y_pred))
```

Figure 5: R Squared lasso

My next step was to use a simple linear regression and see how the data would impact the results. I decided to test OLS(Ordinary Least Squares). Leveraging Scikit learns package, I could see how the machine learning algorithm uses linear regression and predicts players' salaries. The steps I took in order to predict players' salaries based on the Ordinary Least Squares was very similar to how I ran the Lasso regression. I had to have a training group and a testing group. The training group was players that signed their contracts before the period I wanted to test, so from 2017-2021, players' VORP and Salaries were put into the training group. The players for 2022 were put into the testing group in order to see how much my machine-learning algorithm predicted that the players were worth.

```
# Split the data into training/testing sets
X_train = np.array(X[-259:],A[-259:])
X_test = np.array(X[:-259], A[:-259])

# X_1_train = np.array(X_1[:-25])
# X_1_test = np.array(X_1[-25:])
# Split the targets into training/testing sets
y_train = np.array(Y[-259:])
y_test = np.array(Y[:-259])
```

Figure 6: Splitting training and test groups

Due to splitting the dataset into a training and testing group, I was able to run my code and plot the players into a chart that had VORP on the X-axis and salaries and the Y-axis. Thexf chart showed every player in the dataset and allowed me to visualize any trends that my dataset had without running any tests. In order to display and plot the graph, I had to use another Python library called Matplotlib, which has the power to create many types

of visualizations in Python. I choose to use a static graph to show players' salaries and VORP. After importing the library, I could call assign the X and Y axis by leveraging the panda's data frame that I constructed earlier.

```
Y = df1['Ave Salary']
X = df1['VORP_0']
```

Figure 7: Assigning X and Y

After assigning X and Y with specific columns in the panda's data frame, I was curious to see any trends I could see just visually without testing the data. In order to see that, I had to display the graph and use Matplotlib in order to get a clear graph of the players.

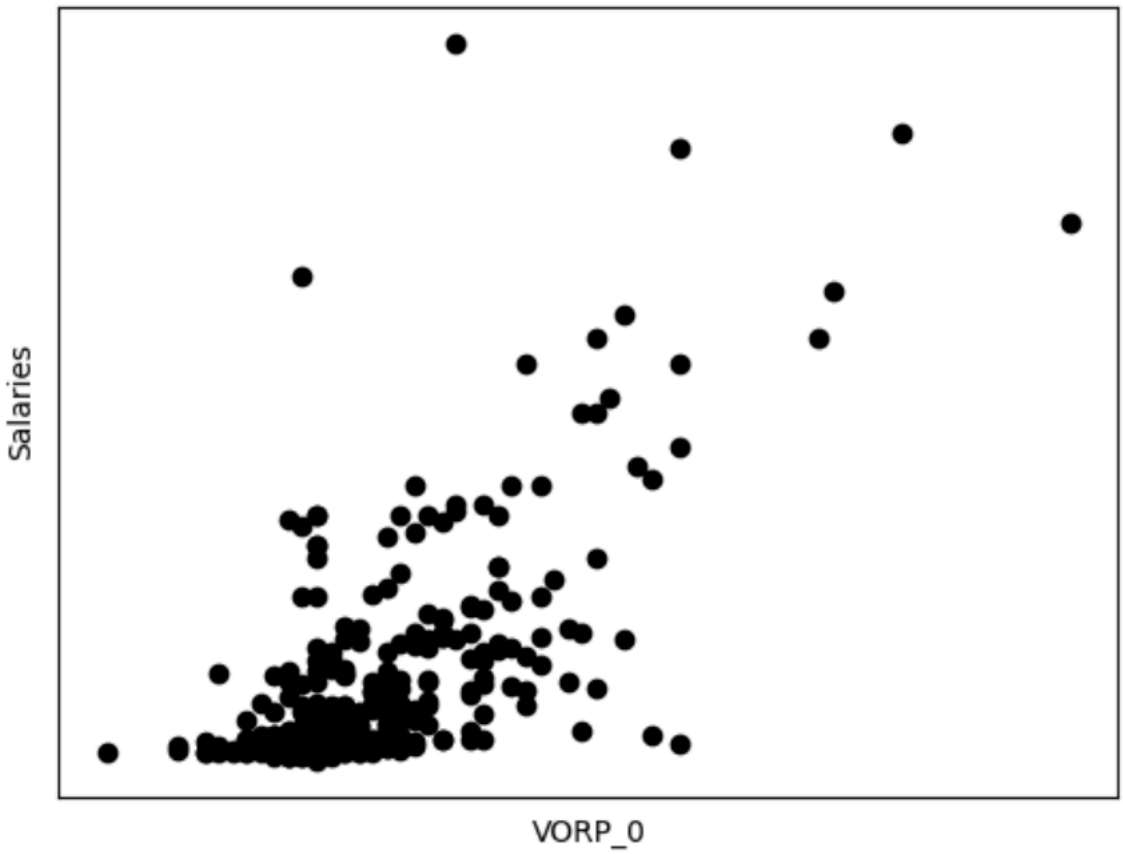

Figure 8: Graph using Matplotlib

This graph allowed me to see the relationship between players' salaries and VORP visually. After seeing the graph I wanted to run the OLS(Ordinary Least Squares) linear regression and predict players' salaries. Implementing this linear regression was very similar to the Lasso regression. I first utilized the Scikit learn package again and called the linear regression.

```
regr = linear_model.LinearRegression()
# Train the model using the training sets
regr.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Figure 9: Linear regression

This predicted the player's salaries that were divided up into the test group. After predicting the player's salaries, I wanted to see how the regression behaved with the data that I had. I wanted to see a line of best fit to ensure that using this method was possible and provided good predictions. In order to complete this, I had to use Matplotlib again in order to print out the line of best fit. Since Matplotlib is very user-friendly and has great documentation, I quickly figured out how to display the line of best fit. Then I also wanted the graph to display three important statistical measurements that would provide further insight into how well the OLS worked with the data I had provided.

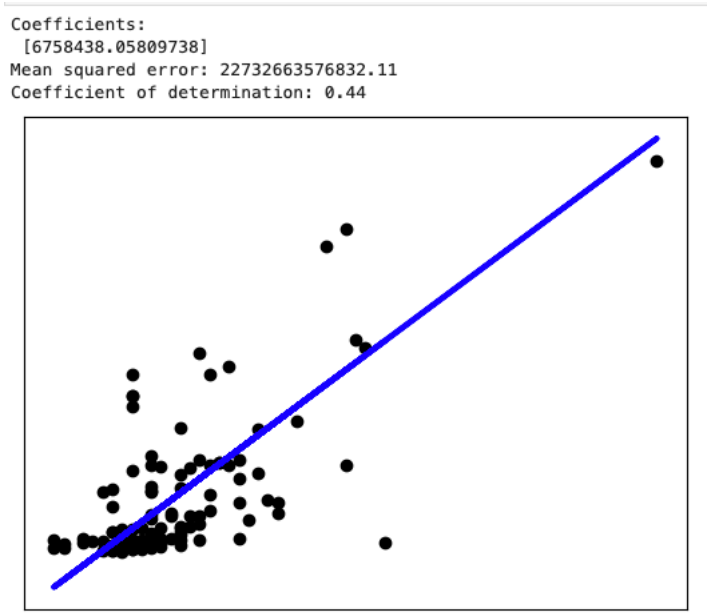

Figure 10: Graph with statistical measurements

With the libraries that I used, I was able to leverage Matplotlib and Scikit learn and predict players' salaries using machine learning, along with having three statistical measurements that allowed was able to show how well the linear regressions worked.

This does raise a concerning question; Is there enough data for the machine learning algorithm to correctly predict a player's salary? This question is very concerning when I will look at the results of the program. In many machine learning programs, there are hundreds of thousands of data points, if not millions, however given that the salary cap in the NBA changes every year and the CBA has getting re-negotiated every time the CBA is over, the rules of players' contracts adjust, and therefore it's hard to create the vast amount of data points that would help improve my machine learning algorithm.

I choose the time period 2017-2022 to look at players and their salaries for a couple of reasons. One reason is that the last Collective Bargaining Agreement was implemented at the start of the NBA calendar in 2017. I didn't use the 2016 data because that year, the salary cap jumped and became a lot higher than it was in 2015, so a lot of players got contracts that wouldn't

normally happen, but due to the steep increase in the salary cap, teams were handing out big contracts to players that didn't deserve them. Since the time period I'm looking at is relatively small, it does raise concerns about the validity of the machine learning algorithm. However, the complexity makes the dataset smaller than it would be in an ideal scenario. Even with low amount of data points that the machine learning algorithm is given, it is able to predict players' salaries. Given that the dataset is small, the validity of the prediction should be questioned. Machine learning works best when the dataset is very big because it allows the model to train on more data which correlates with a better prediction, but since I'm using a time series analysis since the data is constant over time, it's much harder to have a large dataset. With the NBA, there are a lot of changes year after year, the salary cap gets adjusted based on the revenue the league, and teams are able to bring in. With that, salaries are always changing along with teams trying to find an advantage that other teams don't have, so they try new things, and that could lead to them giving players more money even if they don't deserve it; they could just fit the system the team is trying to implement and need to ensure that the player chooses that team in free agency. This makes implementing machine learning difficult because I need to limit the outside factors from skewing my data in a way that could change the results of the prediction. That is why I'm starting at 2017 and using that data because it reduces the chances of the prediction being affected by outside forces.

Experiments

Experimental Design

Especially as it pertains to responsible computing if conducting experiments or evaluations that involve particular ethical considerations detail those issues here.

The first experiment that I conducted was to try a K-fold cross-validation test. A K-fold cross-validation test randomly selects portions of the dataset to train and test the machine learning algorithm. I wanted to see what the R-squared(Coefficient of determination) value was for my data. I used the `train_test_split` tool in the Sci-kit learn to package, which performs a version of a K-fold cross-validation test because if you don't set the random-state parameter, the `train_test_split` tool will randomly select the train and test sizes of the dataset. Since the `train_test_split` tool can run a K-fold cross-validation test, I wanted to see the average R-squared number that I would get if I ran the `train_test_split` function hundreds of times. In order to do so, I had to create a for loop that would recursively run my regression and get the R-squared values for every time my for loop runs. I wanted to see the difference in the mean of the R-squared when the loop is run ten times, then 100 times, and so on. I also had to declare what variables in the data set take the x and y in the linear regression; since I'm trying to predict salary, I set the salary to y and VORP and Age as X.

```
# Split the data into features and target variables
X = df[['VORP_0', 'Age']]
y = df['Ave Salary']
```

Figure 11: Setting X and Y

After I declare what my x and y are I'm able to use the for loop to find the average R-squared values.

```
for i in range(1000):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
    # fit a linear regression model to the training data
    model = LinearRegression().fit(X_train, y_train)

    # calculate the R-squared value on the testing data and append it to the lis
    y_pred = model.predict(X_test)
    r_squared_values.append(r2_score(y_test, y_pred))

# compute the average R-squared value
avg_r_squared = np.mean(r_squared_values)

print("Average R-squared value:", avg_r_squared)
```

Figure 12: For loop recursively ran for 1000X

This for loop is a very simple way to run the `train_test_split()` for however many times that we want to and get the average of R-squared. I wanted to have a better understanding of the relationship between VORP and salary. If VORP and salary didn't have any correlation, then I wouldn't be able to predict an NBA player's salary.

```
Average R-squared value: 0.42538667092169036
Mean Squared Error: 26012724214220.406
R^2: 0.5435776933497738
```

Figure 13: Average R-squared value

When I ran the k-fold cross-validation test one thousand times I and took the average of those results, I got an R-squared value of 0.42 or 42%. If I ran the for loop for 100 times the average R-squared value is 0.43 or 43%, and if I ran it 10,000 times, the average R-squared value is 0.42 or 42%.

```
Average R-squared value: 0.4225454793297771
Mean Squared Error: 38026085985218.17
R^2: 0.3948905045309443
```

Figure 14: Average R-squared value when ran 10000X

I also wanted to look at the MSE or Mean Squared Error. This is a common metric when using machine learning, the MSE stats the average squared difference between the predicted versus the actual values of the dataset; the lower the MSE is, the more accurate the predictions are. In the picture above, the MSE is a super large number, showing that the predicted values do not vary close to the actual values.

After I found the average R-squared value through various amounts of simulation and the MSE, I wanted to split data in the train data set and the test data set. The training dataset is the data portion on which the `train_test_split` function will train its predictions. Through the machine learning algorithm, the program will analyze the data based on the parameters I set, VORP, and Age. I used such a small dataset, relatively for a machine learning program, due to the change in players' salaries going up and the rules being changed due to the CBA. I wanted to use the test data set for the players that were free agents in 2022, so I could see the accuracy of the predicts. If I looked at the upcoming free agents, players who would be free agents in 2023, but I couldn't see how accurate the program was.

```
# Train the model on the training data
reg = LinearRegression().fit(X_train, y_train)

# Make predictions on the test data
predictions = reg.predict(X_test)
```

Figure 15: Code to predict salary

Since I used pandas to create the data frame, I used the iloc[] function to split the dataset into the train and the test data set built-in with importing pandas. After I split my dataset into two separate sections, I was to predict the salaries using a function that is built into the Sci kit learn machine a learning program that it allows the user to predict their desired variable.

```
# Define the first 259 rows as the test set
n = 122
test_df = df.iloc[:n]

# Define the remaining rows as the training set
train_df = df.iloc[n:]
```

Figure 16: Code that split the dataset

The (n:) inside the [] means take everything after n while (:n) takes everything up to n, which is this case is 122. In other words, I'm setting the train data frame to get all the stats past row 122, and I'm testing on rows up to 122. I split up my dataset at 122 because that is where the last player that was a free agent in 2022.

After splitting the dataset using the iloc[] property in the Pandas, I was able to train the machine learning algorithm with the specific rows in the dataset and test on the 2022 free agent class. I then used the `train_test_split` model selection that is in the Sci-kit learn package in order to allow the machine learning algorithm to predict the predicted salaries for the 2022 free agent class.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 122)
# fit a linear regression model to the training data
model = LinearRegression().fit(X_train, y_train)
```

Figure 17: The train_test_split function

The next step was to fit the linear regression with my data set. In order to predict the 2022 NBA free agent class, the model must be based on the training portion of the dataset, which is shown above. This allows the machine learning to analyze the training portion and how those players have gotten paid relative to their VORP and age, then the machine learning algorithm is able to predict the test portion of the dataset.

```
# Make predictions on the test data
predictions = reg.predict(y_test)
print(predictions)
```

Figure 18: Predicting the salaries

Below it a picture of a technical diagram that illustrates how the code works. Down the spine which is with the arrows that have the white end is the order that the code runs in. The arrows that have a black end represents what the code is running. For example after clicking play in `githubtocolab` the cell will import the required packages, which are `pandas` and `Scikit-learn`. Then the code splits the pandas dataframe into two, one where year is equal to 2022 and the other one is when year doesn't equal 2022. After the code does that it runs the regression, either lasso or OLS, depending on what cell you decide to run. This is followed by the output which will create an Excel spreadsheet and the Coefficients and the regression that was run. Both of those are automated, the one thing that is need is to put the players names back into the spreadsheet in order to visually see the players name, predicted salary, and actual salary.

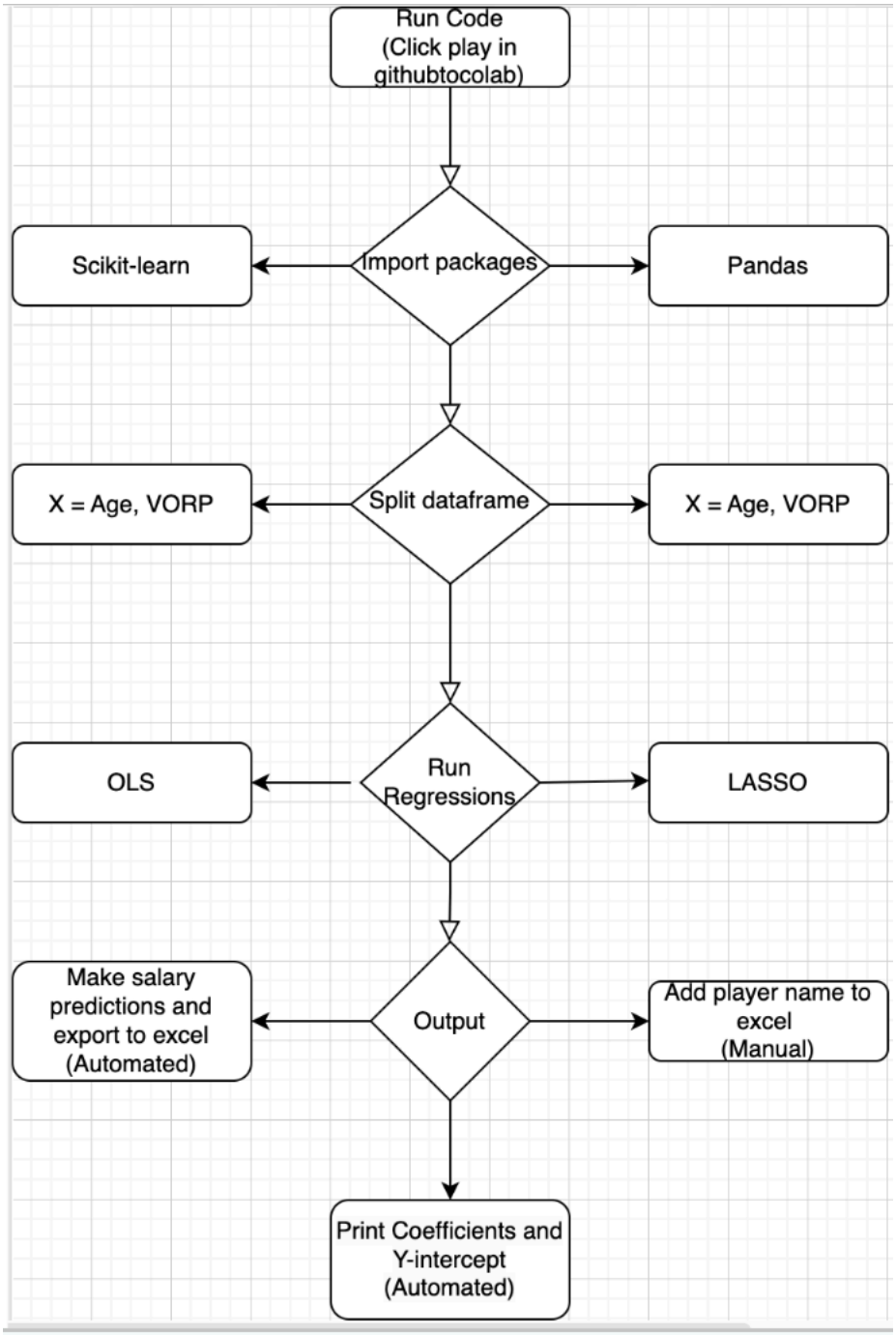

Figure 19: Technical diagram

Evaluation

In order to fully test my tool, I wanted to see how machine learning would predict NBA players' salaries. I choose to predict the last class of NBA free agents that signed in the summer 2022. I created a variable called train_mask which looked at the pandas data frame and found the column titled Year and found all the rows that had the value 2022 in it. Once that variable was created, I could train on all the rows that didn't have 2022 in them and test on the rows that didn't have that value in them.

```
# Split the data into training and testing sets
train_mask = df['Year'] != 2022
X_train, y_train = X[train_mask], y[train_mask]
X_test, y_test = X[~train_mask], y[~train_mask]

# Fit a linear regression model to the training data
reg = LinearRegression().fit(X_train, y_train)

# Make predictions on the test data
predictions = reg.predict(X_test)
```

Figure 20: Splitting the dataframe

Once I could predict the contracts, I needed to display the results somehow. After doing some research, I found that since I was using pandas to build my data frame in pandas, I could easily convert it into an Excel spreadsheet. I created another pandas data frame called test_df which copied over two columns from my main data frame(Player number and Ave Salary) then I appended the results of the machine learning algorithm.

```python
# Create a DataFrame with the actual and predicted values
test_df = pd.DataFrame({'Actual Ave Salary': y_test, 'Predicted Ave Salary': predictions})

# Load the player_number column from the CSV file
player_numbers = pd.read_csv('data/VORP_Salary.csv', usecols=['Player number'])

# Add the player_number column to the test_df DataFrame
test_df['Player number'] = player_numbers.iloc[X_test.index]['Player number'].tolist()

# Export the DataFrame to an Excel file
excel_file = 'Pred_actual.xlsx'
test_df.to_excel(excel_file, index=False)

print(f'The DataFrame has been exported to {excel_file}.')
```

Figure 21: Exporting results to Excel

After the Excel export, I could clearly see how the machine learning algorithm could predict salaries against how NBA General Managers decided to pay players. With a brief overview, I noticed that there was a big difference between the predicted salaries vs. actual salaries when the actual salaries were large. For example, James Harden got paid $34,320,000 million dollars but was predicted to make $2,714,694 million dollars, so why the massive gap in salaries? James Harden had a VORP of -0.1 going into his contract year, which translates to being a slightly above-average player but getting paid like a top player. There are several possible reasons that he got paid so much: He is a household name, former MVP(Most Valuable Player), scoring Champ, and one of the best scorers in NBA history. Looking into great detail, there are several factors that could have contributed to him getting such a large contract.

Below is a picture of the first ten players on the Excel spreadsheet.

Actual Ave Salary	Predicted Ave Salary	Player number	Player
50203930	9905736.544	1	Bradley Beal
43031940	20196048.82	2	Zach LaVine
34320000	2714694.444	3	James Harden
33232282	27360848.12	4	Deandre Ayton
26000000	17066162.05	5	Jalen Brunson
25000000	15912180.45	6	Anfernee Simons
17737500	2483233.806	7	Collin Sexton
17500000	9319998.145	8	Jusuf Nurkic
16500000	6968296.953	9	Luguentz Dort
15000000	16497918.85	10	Mitchell Robinson

Figure 22: First 10 players

After I looked at the results, I wanted to find the MAE(Mean absolute value), which will indicate how accurate my predictions were; in order to find the MAE, I had to take the absolute difference between the actual and the predicted, then find the mean of those absolute difference values. As seen below, the MAE ended up being 3512566, which means that my prediction was around $3.5 million off of the actual salaries.

Actual Ave Salary	Predicted Ave Salary	Player number	Player	Absolute Value	MAE
50203930	9905736.544	1	Bradley Beal	40298193.46	3512566
43031940	20196048.82	2	Zach LaVine	22835891.18	
34320000	2714694.444	3	James Harden	31605305.56	
33232282	27360848.12	4	Deandre Ayton	5871433.881	
26000000	17066162.05	5	Jalen Brunson	8933837.953	
25000000	15912180.45	6	Anfernee Simons	9087819.552	
17737500	2483233.806	7	Collin Sexton	15254266.19	
17500000	9319998.145	8	Jusuf Nurkic	8180001.855	
16500000	6968296.953	9	Luguentz Dort	9531703.047	
15000000	16497918.85	10	Mitchell Robinson	1497918.846	

Figure 23: First 10 players with MAE

After using OLS to predict the free agents of 2022, I wanted to test the machine learning algorithm using Lasso in order to see which prediction produces more accurate results. Setting up the test was very simple; I used the same outline that I used for OLS the only difference between the two was that I used Lasso in order to produce the linear model which allowed the machine learning algorithm to predict.

```python
import pandas as pd
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split

df = pd.read_csv('https://raw.githubusercontent.com/R
df.dropna(inplace=True)

# Split the data into features and target variables
X = df[['VORP_0', 'Age']]
y = df['Ave Salary']

# Split the data into training and testing sets
train_mask = df['Year'] != 2022
X_train, y_train = X[train_mask], y[train_mask]
X_test, y_test = X[~train_mask], y[~train_mask]

# Fit a linear regression model to the training data
reg = Lasso().fit(X_train, y_train)

# Make predictions on the test data
predictions = reg.predict(X_test)
```

Figure 24: Code for the Lasso

After creating the prediction, I wanted to test the R-squared using the same experiment I used when using OLS, I found that the R-squared to very similar. For the Lasso regression, the average R-squared was 0.4245, and the R-squared for the OLS was 0.4242, which shows that the that Lasso regression explained the data better by .03% better, which doesn't influence the prediction in any significant way.

After getting the prediction, I wanted to see if the MAE would differ or not; the MAE when using OLS is 3512565, and the MAE when using Lasso is the exact same, which shows that the Lasso regression doesn't improve the overall prediction of the 2022 free agents.

Then I wanted to look at the coefficients and y-intercept that both my regression to see if there is any difference between the regressions I ran. For both regressions, the results were identical.

```
X-intercept:  5024814.055584514
Coefficents:  [6580245.19430351  -51623.5520544 ]
```

```
X-intercept:  5024814.055584514
Coefficents:  [6580245.19430351  -51623.5520544 ]
```

Threats to Validity

The biggest threat to the validity of my results is the size of my dataset. In order for machine learning to work the best and produce the most accurate predictions the size of the dataset should be as large as possible sometimes, the dataset has hundreds of thousands of rows, if not millions. Since the NBA salary cap changes year to year depending on the revenue the league creates, I didn't want to go back too far due to the fact that the salaries aren't consistent, and since the salaries could go up, I didn't want to have the machine learning algorithm produce numbers that are influenced by lower salaries from the past seasons; that is why my dataset is small compared to other datasets that are used in machine learning.

Another threat to the validity of the predictions is how teams value players. Some teams that don't have a huge market, like Oklahoma City, Portland, Utah, and Minnesota, tend to pay players more than they are worth based on play in order to keep the player in the smaller market. In the CBA rule, teams are able to pay their own players more than other teams would be able to pay them, which was put into the CBA in order to try and keep players and not have the best teams be in the biggest markets and leaving the smaller market teams with worse players. Since this is the case, that explains why the R-squared value is only at .43; If a team in a smaller market wants to keep their best player from leaving in free agency, the team might decide to overpay in order to stay competitive. For example, Bradley Beal, who plays for the Washington Wizards, recently signed a 5-year 251 million contract with a no-trade clause, meaning that Bradley Beal has to agree to be traded. This raised many eyebrows when he signed the contract because that doesn't happen very often there are around 20 players that currently have a no-trade clause. This was the only way that the Washington Wizards were going to be able to keep Bradley Beal. This is because the Washington Wizards aren't very good, but they don't want to go into a full rebuild, so they were willing to pay the max in order to keep him; this could prove to be a contributing factor as to why the predicted salaries for players are different from the actual salary.

The third threat to the validity of the predictions is how the CBA allows teams to structure contracts. Players on their first NBA contract, young players 0-6 years of experience, have a lower max salary at 25% of the salary cap. While players with seven to nine years of experience can make up to 30% of the salary cap, and players with ten-plus years of experience can make up to 35% of the cap. Since certain players can only make a certain max contract, my predictions could be off for several reasons. For one, a player might be one of the league's best players while still on their rookie contract. Ja Morant and Luka Doncic would be making the same money as Bradley Beal and other highly paid players, but since the CBA restricts how much those players can make, it could affect the accuracy of my predictions.

The fourth possible threat to validity is the parameters I used for machine learning. I didn't test for parameters that might be useful due to the time constraint, which might cause the predictions not to be the same as the actual salary. With more time, I would have run more tests that allowed me to see the most influential stat that could help predict future NBA free agents' salaries. I used VORP because it covered several key stats along with taking minutes played into the stat. Every General Manager in the NBA talks about how important availability is, which would naturally help your VORP if you are healthy and playing more minutes.

The fifth possible threat to validity is whether you can market this player and bring attention to the team. Some NBA teams are trying to win championships year after year, while some teams and owners are trying to make a profit. Those teams might resign a player that is marketable and could bring attention to the team, which could increase profit through ticket sales and jerseys.

The last threat to the possible validity is that the free agent class for a year is considered weak. There are not many top players in free agency. What might happen is the best player in free agency will get overpaid because more teams are bidding against each other, and that might raise the salary for the player. Another possibility is that a team thinks that sign a very specific player will make them title contenders. Teams might be inclined to overpay to ensure that they get that player in order to improve their chances of winning a title. If a player gets overpaid, it might affect the machine learning algorithm and predict salaries that are more than the player's worth. Teams that think they are one piece away from a championship will try anything they think will help them achieve their goal of winning the NBA finals because of how hard it is to win a title.

With more time, I would have added more to my data set in order to help the machine learning algorithm be able to make better predictions, I would need to adjust players' salaries in the past due to the salaries and salary cap increasing. I would also have implemented many more experiments with what statistics are most important in NBA salaries. I also need to figure out how to designate players' salaries due to how the CBA affects how certain players are able to get paid.

Conclusion

Summary of Results

My NBA salary prediction tool which used machine learning was able to predict free agent players salaries. I wanted to calculate the MAE in order to find how accurate my predictions were. As seen below, the MAE ended up being 3512566, which means that my prediction was around $3.5 million off of the actual salaries. I also had an R-squared value of .62 when I split my data up by 2022. When I randomized how my data was split up and ran it 10,000 times, I got an R-squared value of .42 or 42% which shows that VORP and salary are correlated. When looking through the results, you can see that there are some predicted salaries that are significantly lower than what the player signed for, which raises a couple of questions: Why did the player make so much more money than they predicted? Did they have a bad year leading up to a free agency? The answer to the ladder is yes, which affects the machine learning algorithm. Players with the biggest difference in predicted salary versus actual salary were players that have been very good for many years and just had a down season heading into free agency for various reasons, be it injury, being on a bad team, or personal issues. James Harden had one of the biggest differences between actual salary and predicted salary, and there are reasons for this. His VORP was -0.1, meaning he was an average to below-average player, but why did the 76ers pay him so much? He was injured for a long time and got traded halfway into the season, which affects your play when you go into a new team. Since the 76ers just traded for him, they couldn't let him leave in free agency, so they had to pay him to keep him. These are some reasons why the predicted salary is so much lower than he actually got.

Future Work

This project can be used as a building block for future computer science majors to enhance and improve. I only had two independent variables that I chose to look at as predictors of salary, one possible way of building on this work would be to introduce more independent variables that could be important for salaries. Introducing more variables will most likely be able to make the predicted salaries closer to the actual salaries the players get from NBA teams. Plenty of stats could be added to the machine learning algorithm to improve the prediction; however, it's important not to pick stats overlapping. For example, VORP takes many stats and combines them in order to get the final result, so it would be very redundant if someone picked a stat to put along with VORP that VORP uses; this might affect the results of the machine learning prediction.

Another possible improvement would be to have a larger database which would help improve the machine learning tool since the tool looks at data, it is important that there is a large database in order to give the algorithm the best chance to train the machine learning predictions that would improve the prediction results. This comes with obvious challenges; since NBA salaries tend to increase year after year, it would be very important to create a database that would make the salaries amount for past players reflect what they would get paid in the current market or find the percentage of the salary cap that the players were getting; this could lead to many challenges.

Another angle that future researchers could take is to have VORP for several years leading up to when they hit free agency. I only looked at one season's VORP for players but past seasons' VORP could be included it might make the prediction more accurate because it balances a player's productivity better than just looking at one season. The more seasons that can be added to the machine learning algorithm, the better the predictions will be.

One threat to validity that would be hard to account for is teams in small markets over paying for a player in order to keep him on the team, one possible way to find that is to eliminate players that are in a small market and test the machine learning program and see if it becomes more accurate.

Future Ethical Implications and Recommendations

One future ethical implication is stat collection and stat padding. If stats are inaccurate, then the machine learning program will not be able to produce accurate results. It is very important to make sure you use data from a reputable source and double-check that your data hasn't been affected by anything throughout the process. Another concern is stat padding, this is where players will purposefully increase their stats in order to look better. For example, a player will steal rebounds in order to inflate their rebounds per game stat. This could make players look better than they actually are and affect the prediction that is made by the machine learning algorithm. Another ethical concern is that players might find ways to make their stats favorable to the machine learning algorithm that way a player would get more money if a team uses these machine learning artifact.

Conclusions

This tool will help close the knowledge gap that exists in the NBA today, where there still is uncertainty about how much players are worth. With the CBA and salary cap, some players will never truly get paid their worth, but that is such a small percentage of the NBA that this tool will help with predicting the vast majority of NBA players contracts. For this tool to be as accurate as possible, there still needs to be many added parameters to get the most accurate predictions possible.

References

[1] Mansoor Assani S. 2020. Efficiency, RTS, and marginal returns from salary on the performance of the nba players: A parallel dea network with shared inputs. *Journal of Industrial and Management Optimization* (2020).

[2] No Author. History of the NBPA.

[3] Krautmann Berri D. J. 2006. Shirking on the court: Testing for the incentive effects of guaranteed pay. *Economic inquiry* 44 (2006).

[4] Larry Coon. 2017. 2017 CBA. (2017).

[5] Stephenson Hastings K. M. 2015. The NBA's maximum player salary and the distribution of player rents. *International Journal of Financial Studies* (2015).

[6] JOLLY HILL J. R. 2012. Salary distribution and collective bargaining agreements: A case study of the NBA. *Industrial Relations (Berkeley)* (2012).

[7] Paulo Canas Rodrigues. João Vítor Rocha da Silva. 2022. All-NBA teams' selection based on unsupervised learning. *Stats* (2022).

[8] Minuci Johnson C. 2020. Wage discrimination in the NBA: Evidence using free agent signings. *Southern Economic Journal* (2020).

[9] Q. Keefer. 2021. Sunk costs in the NBA: The salary cap and free agents. (2021).

[10] Tsagris Papadaki I. 2020. Estimating NBA players salary share according to their performance on court: A machine learning approach. (2020).

[11] William C. Rhoden. 2017. Locker room talk: First day of NBA free agency should be called big o day. (2017).

[12] K. J. Stiroh. 2007. Playing for keeps: Pay and performance in the NBA. *Economic Inquiry* (2007).

[13] Feng Wu W. 2018. Classification of NBA salaries through player statistics. (2018).