Final Project Submission - Analysis of the NBA's MVPs

Emmet Young and Jeremiah Burden
DS 352 Machine Learning
Dr. Rhodes
December 14, 2022

Project Description

We title our project "Analysis of the NBA's MVPs" because we intend to evaluate the distinguishing factors between elite basketball players. The dataset that we have found provides statistical information on every player in the National Basketball Association since 1980. Each row in the set contains a player's statistical averages in attributes like points scored per game and shooting percentage– as well as the amount of Most Valuable Player votes received– over the course of an individual season. In order to analyze what contributes most to the evaluation of the league's most valuable player, we will look at which facets of the game are assessed most highly for the voters of the award. We expect to find a relationship between the amount of points scored in conjunction with other facets of the game to most directly influence how many MVP votes a player receives.

The audience for this project is for all sports fans because they can see what statistics contribute to players winning the MVP. However, NBA fans in particular are most likely to be interested in these results. Basketball players would be intrigued by the results because they can see which statistics are more important in MVP voting. Basketball analysts can predict future MVPs during the season based off of the results found in the study. Also, sports bettors can use the results to predict the MVP during and after the season. We both enjoy watching basketball, play sports at Juniata, and our POE is data science, so inspecting sports statistics is a potential future job for both of us. Emmet is also a Philadelphia 76ers fan, who insists Joel Embiid was robbed of the MVP last year, so he wants confirmation that Nikola Jokic actually deserved the MVP based on the statistics of previous winners. Jeremiah is a Chicago Bulls fan, so he wants to see if Michael Jordan's MVP seasons are more dominant than LeBron James's MVP seasons.

Data Sources

Here is the url of our dataset: https://www.kaggle.com/datasets/javigallego/stats-nba.

This table summarizes the attributes in our dataset.

Summary of Data Variables

Name (variable name)	Description	Data Type	Values Range	Other Notes
Player's Name (Player)	This column identifies the full name of the player whose statistics are held in that row.	Arbitrary Nominal	Any string of letters	
Player's Height (Ht)	This column provides the height in the form 'foot-inch'.	Discrete Ordinal	Any height, but values between 5-10 and 7-7 (unbounded)	
Player's Weight (Wt)	This column tells the weight of the player in pounds.	Continuous Ordinal	Any weight, but values between 133.0 and 303.25 lbs. (unbounded)	The values are given to the nearest hundredth of a pound.
Player's Position (Pos)	This column gives his roster-listed position.	Nominal Categorical	PG, SG, SF, PF, C	
Games Started (GS)	This column lists the number of games started in that season	Discrete Ordinal	Between 0 and 82 games (bounded)	
Minutes Played (MP)	This column lists the average number of minutes played per game.	Continuous Ordinal	Between 0 and 48 minutes (bounded)	
Made Field Goal Shots (FG)*[3P, 2P, FT]	The number of field goal shots made per game.	Continuous Ordinal	Values between 0 and 13.5 shots (unbounded)	The values are given to the nearest tenth.

Field Goal Shots Attempted (FGA)*[3PA, 2PA, FTA]	The number of field goal shots attempted per game.	Continuous Ordinal	Values between 0 and 27 shots (unbounded)	The values are given to the nearest tenth.
Field Goal Percentage (FG%)*[3P%, 2P%, FT%]	The percentage of field goal shots made per game.	Continuous Ordinal	Percentages bound between 0 and 1	The decimal values are given to the nearest thousandth.
Effective Field Goal Percentage (eFG%)	The percentage of shots made given the amount of points they are worth.	Continuous Ordinal	Percentages bound between 0 and 1	The statistic is calculated as follows: (FG + 0.5 * 3P) / FGA
Offensive rebounds (ORB)	The average number of offensive rebounds per game.	Continuous Ordinal	Values between 0 and 7 rebounds (though unbounded)	The decimals are given to the nearest tenth.
Defensive Rebounds (DRB)	The average number of defensive rebounds per game.	Continuous Ordinal	Values between 0 and 12 rebounds (though unbounded)	The decimals are given to the nearest tenth.
Total Rebounds (TRB)	The average number of total rebounds per game.	Continuous Ordinal	Values between 0 and 18 rebounds (though unbounded)	The decimals are given to the nearest tenth.
Blocks per game (BLK)	The average number of blocks per game.	Continuous Ordinal	Values between 0 and 3 blocks (though unbounded)	The decimals are given to the nearest tenth.
Turnovers per game (TOV)	The average number of turnovers per game.	Continuous Ordinal	Values between 0 and 5 turnovers (though unbounded)	The decimals are given to the nearest tenth.
Personal Fouls per game (PF)	The average number of	Continuous Ordinal	Values between 0 and 6	The decimals are given to the

	personal fouls per game.		(bounded)	nearest tenth.
Points per game (PTS)	The average number of points scored per game.	Continuous Ordinal	Values between 0 and 35 (unbounded)	The decimals are given to the nearest tenth.
Season (Year)	The year that the player ended the season for the related statistics.	Discrete Ordinal	Values between 1991 and 2021 (bounded)	
MVP points won (PTS Won)	The number of MVP points won for that particular season.	Discrete Ordinal	Values between 0 and 121	The NBA MVP voting system works like this: 100 members rank MVP candidates first through fifth. For each place, players earn points as follows: 1st - 10, 2nd - 7, 3rd - 5, 4th - 3, 5th - 1.
Maximum MVP points possible (PTS Max)	The maximum number of MVP votes in a season	Discrete Ordinal	Values between 0 and 121 (bounded)	The number of MVP voters has changed over time resulting in a different maximum number of votes received.
Share Obtained (Share)	The proportion of MVP shares earned out of the maximum possible	Discrete Ordinal	Values between 0 and 1 (bounded)	
Team's Wins (W)	The number of wins for the team that the player competed for that season.	Discrete Ordinal	Values between 0 and 82 (bounded)	

Team's Losses (L)	The number of losses for the team that the player competed for that season.	Discrete Ordinal	Values between 0 and 82 (bounded)	
Percentage of wins vs losses (W/L%)	The percentage of wins versus the number of losses for a team in that season.	Continuous ordinal	Percentages bound between 0 and 1	The decimal values are given to the nearest thousandth.
Games Back (GB)	The number of games between the team with the best record and the team that the player plays for.	Discrete Ordinal	Values between 0 and 82 (bounded)	This is calculated by subtracting the difference between the leading team and the team of interest's wins and loses by their own and then finding the difference between the two ranges and then dividing by two.
Points Per Game (PS/G)	The average number of points scored per game for that team.	Continuous Ordinal	Values between 0 and 130 (unbounded)	The decimal values are given to the nearest tenth.
Opponent Points Per Game (PA/G)	The average number of points given up to the opponent per game for that team.	Continuous Ordinal	Values between 0 and 130 (unbounded)	The decimal values are given to the nearest tenth.
Simple Rating System (SRS)	A rating system that tracks the average number of a player's own scoring versus their counterpart	Continuous Ordinal	Values between -15 and 15	The decimal values are given to the nearest hundredth.

	player on the other team while the player of interest is in the game.			

*Each of the Field Goal columns are repeated for more specific instances, three and two point shots and free throws for nine additional columns. The bracketed abbreviations correspond to the variable names of the columns with the same statistic under different specifications.

Some variables that we have eliminated from our project are not included in the table like an id column and the player's college. Of course, we intend to change some of the names of the columns and potentially remove other unneeded attributes. Below is a sample snapshot of our data.

cleaned_data

Player	Ht	Wt	Pos	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year	Pts Won	Pts Max	W	L	W/L%	GB	PS/G	PA/G	SRS
Alaa Abdelnaby	6-10	240.0	PF	0	6.7	1.3	2.7	0.474	0.0	0.0	0.0	1.3	2.7	0.474	0.474	0.6	1.0	0.568	0.6	1.4	2.1	0.3	0.1	0.3	0.5	0.9	3.1	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Danny Ainge	6-4	175.0	SG	0	21.4	4.2	8.9	0.472	1.3	3.1	0.406	2.9	5.8	0.508	0.543	1.4	1.7	0.826	0.6	2.0	2.6	3.6	0.8	0.2	1.3	2.4	11.1	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Mark Bryant	6-9	245.0	PF	0	14.7	1.9	3.8	0.488	0.0	0.0	0.0	1.9	3.8	0.49	0.488	1.4	1.9	0.733	1.2	2.4	3.6	0.5	0.3	0.2	0.6	2.3	5.1	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Wayne Cooper	6-10	220.0	C	1	11.1	0.9	2.2	0.393	0.0	0.0	0.0	0.9	2.1	0.396	0.393	0.5	0.6	0.786	0.8	2.0	2.8	0.3	0.1	0.9	0.3	1.8	2.2	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Walter Davis	6-6	193.0	SG	14	20.9	5.7	12.1	0.468	0.2	0.5	0.306	5.5	11.6	0.475	0.474	1.5	1.6	0.915	1.0	1.5	2.5	1.8	1.1	0.0	1.2	2.1	13.0	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Clyde Drexler	6-7	210.0	SG	82	34.8	7.9	16.3	0.482	0.7	2.3	0.319	7.1	14.0	0.509	0.505	5.1	6.4	0.794	2.6	4.1	6.7	6.0	1.8	0.7	2.8	2.8	21.5	1991.0	75.0	960.0	63	19	0.768	0.0	114.7	106.0	8.47
Kevin Duckworth	7-0	275.0	C	81	31.0	6.4	13.4	0.481	0.0	0.0	0.0	6.4	13.4	0.482	0.481	3.0	3.8	0.772	2.2	4.4	6.6	1.1	0.4	0.4	2.3	3.1	15.8	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Jerome Kersey	6-7	215.0	SF	72	32.3	5.8	12.2	0.478	0.1	0.2	0.308	5.8	12.0	0.481	0.48	3.2	4.5	0.709	2.3	4.3	6.6	3.1	1.4	1.0	2.0	3.4	14.8	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Terry Porter	6-3	195.0	PG	81	32.9	6.0	11.7	0.515	1.6	3.9	0.415	4.4	7.8	0.564	0.584	3.4	4.2	0.823	0.6	2.8	3.5	8.0	2.0	0.1	2.3	1.9	17.0	1991.0	25.0	960.0	63	19	0.768	0.0	114.7	106.0	8.47
Clifford Robinson	6-10	225.0	PF	11	23.7	4.5	9.8	0.463	0.1	0.2	0.316	4.5	9.6	0.466	0.467	2.5	3.8	0.653	1.5	2.8	4.3	1.8	1.0	0.9	1.6	3.2	11.7	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Buck Williams	6-8	215.0	PF	80	32.3	4.5	7.4	0.602	0.0	0.0	0.0	4.5	7.4	0.602	0.602	2.7	3.9	0.705	2.8	6.6	9.4	1.2	0.6	0.6	1.7	3.1	11.7	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Danny Young	6-3	175.0	PG	1	12.0	1.4	3.6	0.38	0.5	1.4	0.346	0.9	2.2	0.401	0.446	0.5	0.6	0.911	0.3	0.7	1.0	1.9	0.7	0.1	0.7	0.7	3.8	1991.0	0.0	0.0	63	19	0.768	0.0	114.7	106.0	8.47
Alaa Abdelnaby	6-10	240.0	PF	1	13.2	2.5	5.1	0.493	0.0	0.0	0.0	2.5	5.1	0.493	0.493	1.1	1.4	0.752	1.1	2.5	3.7	0.4	0.4	0.2	0.9	1.9	6.1	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Danny Ainge	6-4	175.0	SG	6	19.7	3.7	8.3	0.442	1.0	2.8	0.339	2.7	5.5	0.496	0.5	1.3	1.6	0.824	0.5	1.3	1.8	2.5	0.9	0.2	0.9	1.8	9.7	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Mark Bryant	6-9	245.0	PF	0	14.3	1.7	3.5	0.48	0.0	0.1	0.0	1.7	3.5	0.487	0.48	0.7	1.1	0.667	1.6	2.0	3.6	0.7	0.5	0.1	0.5	1.9	4.1	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Wayne Cooper	6-10	220.0	C	0	9.8	1.0	2.3	0.427	0.0	0.0	0.0	1.0	2.3	0.427	0.427	0.2	0.3	0.636	1.1	1.8	2.9	0.6	0.1	0.8	0.4	1.6	2.2	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Clyde Drexler	6-7	210.0	SG	76	36.2	9.1	19.4	0.47	1.5	4.4	0.337	7.6	15.0	0.51	0.509	5.3	6.6	0.794	2.2	4.4	6.6	6.7	1.8	0.9	3.2	3.0	25.0	1992.0	561.0	960.0	57	25	0.695	0.0	111.4	104.1	6.94
Kevin Duckworth	7-0	275.0	C	82	27.1	4.4	9.6	0.461	0.0	0.0	0.0	4.4	9.5	0.462	0.461	1.9	2.8	0.69	1.8	4.2	6.1	1.2	0.5	0.5	1.7	3.2	10.7	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Jerome Kersey	6-7	215.0	SF	76	33.2	5.2	11.1	0.467	0.0	0.1	0.125	5.2	11.0	0.47	0.468	2.3	3.4	0.664	3.1	5.1	8.2	3.2	1.5	0.9	2.0	3.3	12.6	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Robert Pack	6-2	180.0	PG	0	12.4	1.6	3.8	0.423	0.0	0.1	0.0	1.6	3.6	0.439	0.423	1.4	1.8	0.803	0.4	0.9	1.3	1.9	0.6	0.1	1.3	1.4	4.6	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Terry Porter	6-3	195.0	PG	82	34.0	6.4	13.8	0.461	1.6	4.0	0.395	4.8	9.8	0.488	0.518	3.8	4.5	0.856	0.6	2.5	3.1	5.8	1.5	0.1	2.3	1.9	18.1	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Clifford Robinson	6-10	225.0	SF	7	25.9	4.9	10.4	0.466	0.0	0.1	0.091	4.8	10.3	0.471	0.467	2.7	4.0	0.664	1.7	3.4	5.1	1.7	1.0	1.3	1.9	3.3	12.4	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Lamont Strothers	6-4	190.0	SG	0	4.3	1.0	3.0	0.333	0.0	0.5	0.0	1.0	2.5	0.4	0.333	0.5	1.0	0.5	0.3	0.0	0.3	0.3	0.3	0.3	0.5	0.5	2.5	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Ennis Whatley	6-3	177.0	PG	0	9.1	0.9	2.2	0.412	0.0	0.2	0.0	0.9	2.0	0.447	0.412	1.2	1.3	0.871	0.3	0.7	0.9	1.5	0.6	0.1	0.6	0.5	3.0	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Buck Williams	6-8	215.0	PF	80	31.5	4.3	7.0	0.604	0.0	0.0	0.0	4.3	7.0	0.605	0.604	2.8	3.7	0.754	3.3	5.6	8.8	1.4	0.8	0.5	1.6	3.1	11.3	1992.0	0.0	0.0	57	25	0.695	0.0	111.4	104.1	6.94
Alaa Abdelnaby	6-10	240.0	PF	52	17.5	3.3	6.3	0.518	0.0	0.0	0.0	3.3	6.3	0.519	0.518	1.2	1.5	0.759	1.7	2.8	4.5	0.4	0.3	0.3	1.3	2.5	7.7	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
John Bagley	6-0	185.0	PG	0	9.7	0.9	2.5	0.36	0.0	0.1	0.0	0.9	2.4	0.375	0.36	0.5	0.6	0.833	0.1	0.6	0.7	2.0	0.2	0.0	1.7	1.1	2.3	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Kenny Battle	6-6	210.0	SG	1	9.7	2.0	4.3	0.462	0.0	0.3	0.0	2.0	4.0	0.5	0.462	0.7	0.7	1.0	2.3	1.3	3.7	0.7	0.3	0.0	0.7	0.7	4.7	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Dee Brown	6-1	160.0	PG	48	28.2	4.1	8.8	0.468	0.3	1.0	0.317	3.8	7.7	0.488	0.486	2.4	3.0	0.793	0.6	2.5	3.1	5.8	1.7	0.4	1.7	2.5	10.9	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Dee Brown	6-0	185.0	PG	48	28.2	4.1	8.8	0.468	0.3	1.0	0.317	3.8	7.7	0.488	0.486	2.4	3.0	0.793	0.6	2.5	3.1	5.8	1.7	0.4	1.7	2.5	10.9	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Sherman Douglas	6-0	180.0	PG	36	24.5	3.3	6.7	0.498	0.1	0.4	0.207	3.3	6.3	0.515	0.504	1.1	1.9	0.56	0.8	1.2	2.1	6.4	0.6	0.1	2.0	2.1	7.8	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Rick Fox	6-7	230.0	SF	14	15.2	2.6	5.4	0.484	0.1	0.3	0.174	2.5	5.0	0.504	0.489	1.1	1.4	0.802	0.8	1.5	2.2	1.6	0.9	0.3	1.1	1.9	6.4	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Kevin Gamble	6-5	210.0	SF	58	31.0	5.6	11.0	0.507	0.6	1.7	0.374	5.0	9.4	0.531	0.535	1.5	1.8	0.826	0.6	2.4	3.0	2.8	1.0	0.5	1.0	2.3	13.3	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Joe Kleine	6-11	255.0	C	3	14.5	1.4	3.4	0.404	0.0	0.1	0.0	1.4	3.3	0.414	0.404	0.5	0.7	0.707	1.4	3.0	4.4	0.5	0.2	0.2	0.5	1.6	3.3	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Bart Kofoed	6-4	210.0	SG	0	5.9	0.4	1.9	0.231	0.0	0.1	0.0	0.4	1.7	0.25	0.231	1.6	2.0	0.786	0.0	0.1	0.1	1.4	0.3	0.1	0.4	0.1	2.4	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93
Reggie Lewis	6-7	195.0	SG	80	39.3	8.3	17.6	0.47	0.2	0.8	0.233	8.1	16.9	0.481	0.475	4.1	4.7	0.867	1.1	3.2	4.3	3.7	1.5	1.0	1.7	3.1	20.8	1993.0	0.0	0.0	48	34	0.585	12.0	103.7	102.8	0.93

Preprocessing

In our preprocessing, we primarily used python because we both have experience with it, and exporting and importing a CSV in Python is simple. We did not need to merge datasets

because our data frame is already extremely robust and contains the variables that we think will be most important in identifying the relationship between MVPs. We also had minimal cleaning to do because the dataset was already processed for easy use. We decided to keep rows of two athletes who have the same name because we decided not to merge players by name.

To limit the sheer amount of observations, we cut out those who played less than half the games in the NBA season (which has a total of 82 games). Therefore, any player who did not start in at least 43 games was removed from our analysis. These measures would be to focus our attention on the players that were actually gaining votes, or those who were closest so that we can better understand why they did not receive any shares of the votes.

Initially, we chose to eliminate 13 attributes that were not indicative of anything that we wanted to analyze. Four of them were personal characteristics of the players including their height, weight, age during that season, and their college attended. We did not want to look into any of these variables and felt that any information that they did hold would be duplicated in other variables, like position would generally indicate height and weight. Some other duplicate variables that we dropped were the player's team, total games played (we kept the games started), average minutes played, wins, losses, and games back of the leading team. These columns also had information in other variables that we kept like the win% instead of keeping wins and loses, for example. Finally, we dropped Pts Won and Pts Max that were redundant variables about the MVP votes received.

Later in our analysis, we limited our dataset by focusing on the attributes that were closest to the Shares attribute based on Euclidean distance in order to examine if a more focused, targeted dataset on the commonly discussed attributes related to MVP votes produced more accurate models.

[31]:
```
data2 = data.loc[(data["GS"] > 43)]
data2
```

[31]:

	Unnamed: 0	Player	Ht	Wt	Colleges	Pos	Age	Tm	G	GS	...	Pts Max	Share	Team	W	L	W/L%	GB	PS/G	PA/G	SRS
5	5	Clyde Drexler	6-7	210.0	Houston	SG	28	POR	82	82	...	960.0	0.078	Portland Trail Blazers	63	19	0.768	0.0	114.7	106.0	8.47
6	6	Kevin Duckworth	7-0	275.0	Eastern Illinois University	C	26	POR	81	81	...	0.0	0.000	Portland Trail Blazers	63	19	0.768	0.0	114.7	106.0	8.47
7	7	Jerome Kersey	6-7	215.0	Longwood University	SF	28	POR	73	72	...	0.0	0.000	Portland Trail Blazers	63	19	0.768	0.0	114.7	106.0	8.47
8	8	Terry Porter	6-3	195.0	University of Wisconsin-Stevens Point	PG	27	POR	81	81	...	960.0	0.026	Portland Trail Blazers	63	19	0.768	0.0	114.7	106.0	8.47
10	10	Buck Williams	6-8	215.0	Maryland	PF	30	POR	80	80	...	0.0	0.000	Portland Trail Blazers	63	19	0.768	0.0	114.7	106.0	8.47
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17959	17959	Mo Williams	6-1	198.0	Alabama	PG	27	CLE	69	68	...	0.0	0.000	Cleveland Cavaliers	61	21	0.744	0.0	102.1	95.6	6.17
17961	17961	Al Horford	6-9	240.0	Florida	C	26	ATL	74	74	...	0.0	0.000	Atlanta Hawks	44	38	0.537	22.0	98.0	97.5	-0.08
17965	17965	Kyle Korver	6-7	212.0	Creighton	SF	31	ATL	74	60	...	0.0	0.000	Atlanta Hawks	44	38	0.537	22.0	98.0	97.5	-0.08
17970	17970	Josh Smith	6-9	225.0	Not American	PF	27	ATL	76	76	...	0.0	0.000	Atlanta Hawks	44	38	0.537	22.0	98.0	97.5	-0.08
17972	17972	Jeff Teague	6-3	195.0	Wake Forest	PG	24	ATL	80	78	...	0.0	0.000	Atlanta Hawks	44	38	0.537	22.0	98.0	97.5	-0.08

5549 rows × 45 columns

[83]:
```
%matplotlib inline

data3 = data2.drop(['Ht', 'Wt', 'Colleges', 'Age', 'Tm', 'G', 'MP', 'W', 'L', 'Pts Won', 'Pts Max', 'GB', 'Unnamed: 0'],axis=1)
data3
```

[83]:

	Player	Pos	GS	FG	FGA	FG%	3P	3PA	3P%	2P	...	TOV	PF	PTS	Year	Share	Team	W/L%	PS/G	PA/G	SRS
5	Clyde Drexler	SG	82	7.9	16.3	0.482	0.7	2.3	0.319	7.1	...	2.8	2.8	21.5	1991.0	0.078	Portland Trail Blazers	0.768	114.7	106.0	8.47
6	Kevin Duckworth	C	81	6.4	13.4	0.481	0.0	0.0	0.000	6.4	...	2.3	3.1	15.8	1991.0	0.000	Portland Trail Blazers	0.768	114.7	106.0	8.47
7	Jerome Kersey	SF	72	5.8	12.2	0.478	0.1	0.2	0.308	5.8	...	2.0	3.4	14.8	1991.0	0.000	Portland Trail Blazers	0.768	114.7	106.0	8.47
8	Terry Porter	PG	81	6.0	11.7	0.515	1.6	3.9	0.415	4.4	...	2.3	1.9	17.0	1991.0	0.026	Portland Trail Blazers	0.768	114.7	106.0	8.47
10	Buck Williams	PF	80	4.5	7.4	0.602	0.0	0.0	0.000	4.5	...	1.7	3.1	11.7	1991.0	0.000	Portland Trail Blazers	0.768	114.7	106.0	8.47
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17959	Mo Williams	PG	68	5.5	12.4	0.442	2.3	5.4	0.429	3.2	...	2.5	2.5	15.8	2010.0	0.000	Cleveland Cavaliers	0.744	102.1	95.6	6.17
17961	Al Horford	C	74	7.8	14.3	0.543	0.0	0.1	0.500	7.7	...	2.0	2.2	17.4	2013.0	0.000	Atlanta Hawks	0.537	98.0	97.5	-0.08
17965	Kyle Korver	SF	60	3.7	8.1	0.461	2.6	5.6	0.457	1.2	...	0.9	2.3	10.9	2013.0	0.000	Atlanta Hawks	0.537	98.0	97.5	-0.08
17970	Josh Smith	PF	76	7.2	15.6	0.465	0.8	2.6	0.303	6.4	...	3.0	2.3	17.5	2013.0	0.000	Atlanta Hawks	0.537	98.0	97.5	-0.08
17972	Jeff Teague	PG	78	5.5	12.2	0.451	1.1	3.1	0.359	4.4	...	2.9	2.3	14.6	2013.0	0.000	Atlanta Hawks	0.537	98.0	97.5	-0.08

5549 rows × 32 columns

[75]:
```
data4 = data3.loc[(data["Share"] > 0.0)]
data4
```

[75]:

	Unnamed: 0	Player	Pos	GS	FG	FGA	FG%	3P	3PA	3P%	...	TOV	PF	PTS	Year	Share	Team	W/L%	PS/G	PA/G	SRS
5	5	Clyde Drexler	SG	82	7.9	16.3	0.482	0.7	2.3	0.319	...	2.8	2.8	21.5	1991.0	0.078	Portland Trail Blazers	0.768	114.7	106.0	8.47
8	8	Terry Porter	PG	81	6.0	11.7	0.515	1.6	3.9	0.415	...	2.3	1.9	17.0	1991.0	0.026	Portland Trail Blazers	0.768	114.7	106.0	8.47
16	16	Clyde Drexler	SG	76	9.1	19.4	0.470	1.5	4.4	0.337	...	3.2	3.0	25.0	1992.0	0.584	Portland Trail Blazers	0.695	111.4	104.1	6.94
75	75	Kareem Abdul-Jabbar	C	82	10.2	16.9	0.604	0.0	0.0	0.000	...	3.6	2.6	24.8	1980.0	0.665	Los Angeles Lakers	0.732	115.1	109.2	5.40
86	86	Kareem Abdul-Jabbar	C	80	10.5	18.2	0.574	0.0	0.0	0.000	...	3.1	3.1	26.2	1981.0	0.414	Los Angeles Lakers	0.659	111.2	107.3	3.27
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17847	17847	Karl Malone	PF	82	9.2	18.0	0.509	0.0	0.1	0.250	...	2.8	2.8	25.5	2000.0	0.258	Utah Jazz	0.671	96.5	92.0	4.52
17909	17909	Michael Redd	SG	82	7.7	17.5	0.440	1.5	4.4	0.350	...	1.4	1.9	21.7	2004.0	0.001	Milwaukee Bucks	0.500	98.0	97.0	0.42
17915	17915	Kevin Garnett	PF	82	9.1	18.1	0.502	0.2	0.9	0.282	...	2.8	2.4	23.0	2003.0	0.732	Minnesota Timberwolves	0.622	98.1	96.0	2.46
17933	17933	LeBron James	SF	81	9.7	19.9	0.489	1.6	4.7	0.344	...	3.0	1.7	28.4	2009.0	0.969	Cleveland Cavaliers	0.805	100.3	91.4	8.68
17949	17949	LeBron James	SF	76	10.1	20.1	0.503	1.7	5.1	0.333	...	3.4	1.6	29.7	2010.0	0.980	Cleveland Cavaliers	0.744	102.1	95.6	6.17

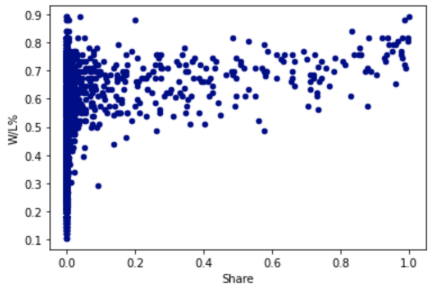
675 rows × 33 columns

Initial Data Visualization

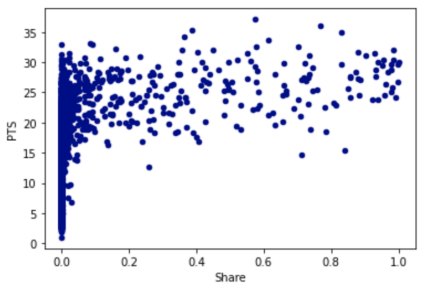
Scatterplots

Since, we are analyzing what contributes most to the evaluation of the league's most valuable player, we looked at which facets of the game are assessed most highly for the voters of the award. Therefore, the primary variable we are targeting is the Share variable, which is the percentage of the MVP votes that each player received each year. By using scatterplots, we are testing to see if any other attribute correlates to higher values of the share variable. The only variable that showed some correlation was the W/L%, and even that had very little correlation. The rest of the scatterplots showed a range, but did not show any correlation between the variables. For example, for PTS, you need to score at least fifteen points to even be considered a viable MVP option. Even by scoring fifteen points, very few players below twenty points per game received a substantial amount of shares.

```
WL = data3.plot.scatter(x='Share',
                        y='W/L%',
                        c='DarkBlue')
```




```
PTS = data3.plot.scatter(x='Share',
                         y='PTS',
                         c='DarkBlue')
```

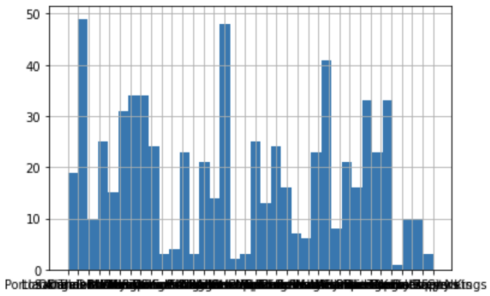


Histograms

After seeing a small correlation between W/L% and share, we created a histogram of only players who received votes and which team they played for. This histogram showed that the Los Angeles Lakers, San Antonio Spurs, Utah Jazz, Boston Celtics, and Phoenix Suns had the most players receive at least one MVP vote. These teams also happen to be the top five

winningest teams since the 1980s, which correlates with our initial observation that higher W/L% helps a player receive MVP votes.

```
[96]: data4['Team'].hist(bins=36)
      data4['Team'].value_counts(sort=False)

[96]: Portland Trail Blazers    19
      Los Angeles Lakers        49
      Sacramento Kings          10
      Dallas Mavericks          25
      Oklahoma City Thunder     15
      Detroit Pistons           31
      Boston Celtics            34
      Phoenix Suns              34
      Milwaukee Bucks           24
      Memphis Grizzlies          3
      Washington Bullets         4
      Denver Nuggets            23
      Charlotte Hornets          3
      Miami Heat                21
      Los Angeles Clippers      14
      San Antonio Spurs         48
      Charlotte Bobcats          2
      New Orleans Pelicans       3
      New York Knicks           25
      Minnesota Timberwolves    13
      Golden State Warriors     24
      Orlando Magic             16
      Washington Wizards         7
      New Orleans Hornets        6
      Cleveland Cavaliers       23
      Utah Jazz                 41
      Indiana Pacers             8
      Seattle SuperSonics       21
      Atlanta Hawks             16
      Houston Rockets           33
      Chicago Bulls             23
      Philadelphia 76ers        33
      San Diego Clippers         1
      Toronto Raptors           10
      New Jersey Nets           10
      Kansas City Kings          3
      Name: Team, dtype: int64
```



Similarity

We also ran similarity measures in order to better understand how our variables relate. For this time, we took the Euclidean distance between each attribute and the number of shares of the MVP votes received because this is our variable of most interest. The attributes that are most related to the Shares attribute are shooting percentages (like FG%, eFG%, 2P%, 3P%) and the win percentage of the team that season (W/L%). This is not necessarily surprising, but it was useful because it provided a basis to eliminate more variables in our dataset when investigating various data mining algorithms. This also is mainly because the percentages are decimals, so they are more likely to be closer when using Euclidean distance than PA/G, PS/G, and PTS. However, it is interesting that PTS are very dissimilar when it comes to players who have received votes.

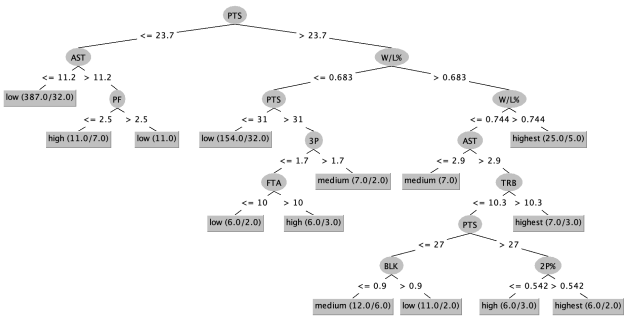
```
print('Euclidean Distances from Shares:' )
for i in range(24):
    print(data4.columns[i+3], ':', euclidean(data4["Share"],data4.iloc[:,i+3]))

for j in range(4):
    print(data4.columns[j+28],':',euclidean(data4["Share"],data4.iloc[:,j+28]))
```

```
Euclidean Distances from Shares:
FG : 214.35190955995702
FGA : 439.93510490980367
FG% : 10.932041758061482
3P : 29.240706763004205
3PA : 81.76713479143072
3P% : 8.069539887750727
2P : 196.23606073298558
2PA : 385.832090853003
2P% : 11.285706180828916
eFG% : 11.35555163785538
FT : 133.9289652465067
FTA : 173.4092821391058
FT% : 17.557171298361247
ORB : 56.47812436687324
DRB : 152.93011453601937
TRB : 209.32416088927718
AST : 145.94263096162135
STL : 36.2089786102839 35
BLK : 32.99359228698809
TOV : 72.96633834858372
PF : 67.17125673976928
PTS : 588.4926658268563
Year : 51922.00751172216
Share : 0.0
W/L% : 13.769770695258508
PS/G : 2734.7636336860996
PA/G : 2635.6091946136476
SRS : 123.2449887500502
```

Data Mining Algorithms

Decision Tree

We created a decision tree by discretizing our ‘shares’ classifier attribute. It was originally an ordinal variable but discretized it by creating four classes of how many MVP shares a player got in a season (and ignoring those that got no votes). What we find is that the biggest determinants of what classifies high to low MVP vote earners is points and team win percentage. This is not necessarily a surprising result– players who scored the most for the best teams typically win MVPs from our knowledge of the domain. The discretized attribute used natural splits in the data, leaving the majority below 0.15 shares (those who got very few votes), two middle groups split at 0.25 and then greater than or equal to 0.5 because if the player earned that many votes, they automatically earned the MVP.




```
=== Summary ===

Correctly Classified Instances         166               74.4395 %
Incorrectly Classified Instances        57               25.5605 %
Kappa statistic                          0.3398
Mean absolute error                      0.1492
Root mean squared error                  0.3118
Relative absolute error                 74.368  %
Root relative squared error             92.499  %
Total Number of Instances              223

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.932    0.541    0.821      0.932    0.873      0.459    0.764     0.852     low
               0.273    0.075    0.158      0.273    0.200      0.153    0.610     0.092     high
               0.033    0.026    0.167      0.033    0.056      0.016    0.643     0.254     medium
               0.550    0.015    0.786      0.550    0.647      0.630    0.814     0.595     highest
Weighted Avg.  0.744    0.402    0.697      0.744    0.709      0.400    0.745     0.711

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 151   7   2   2 |   a = low
   5   3   2   1 |   b = high
  24   5   1   0 |   c = medium
   4   4   1  11 |   d = highest
```

KNN

Next, we used the same discretized ‘shares’ attribute in order to run kNN on our data. For our original KNN, we used K=5. This model actually performed better than the decision tree, but still not extremely accurate. This seems to be because of the high number of misclassifications into the low group, even though the group's distributions are more dispersed. In our best KNN model, we used K=15. This helped the accuracy drastically, 83.84%. The classifications of the low were better, but the main difference was the prediction of the medium and highest categories of shares. We attempted using higher K’s, but found that no accuracy beat the 83.84% when using KNN.

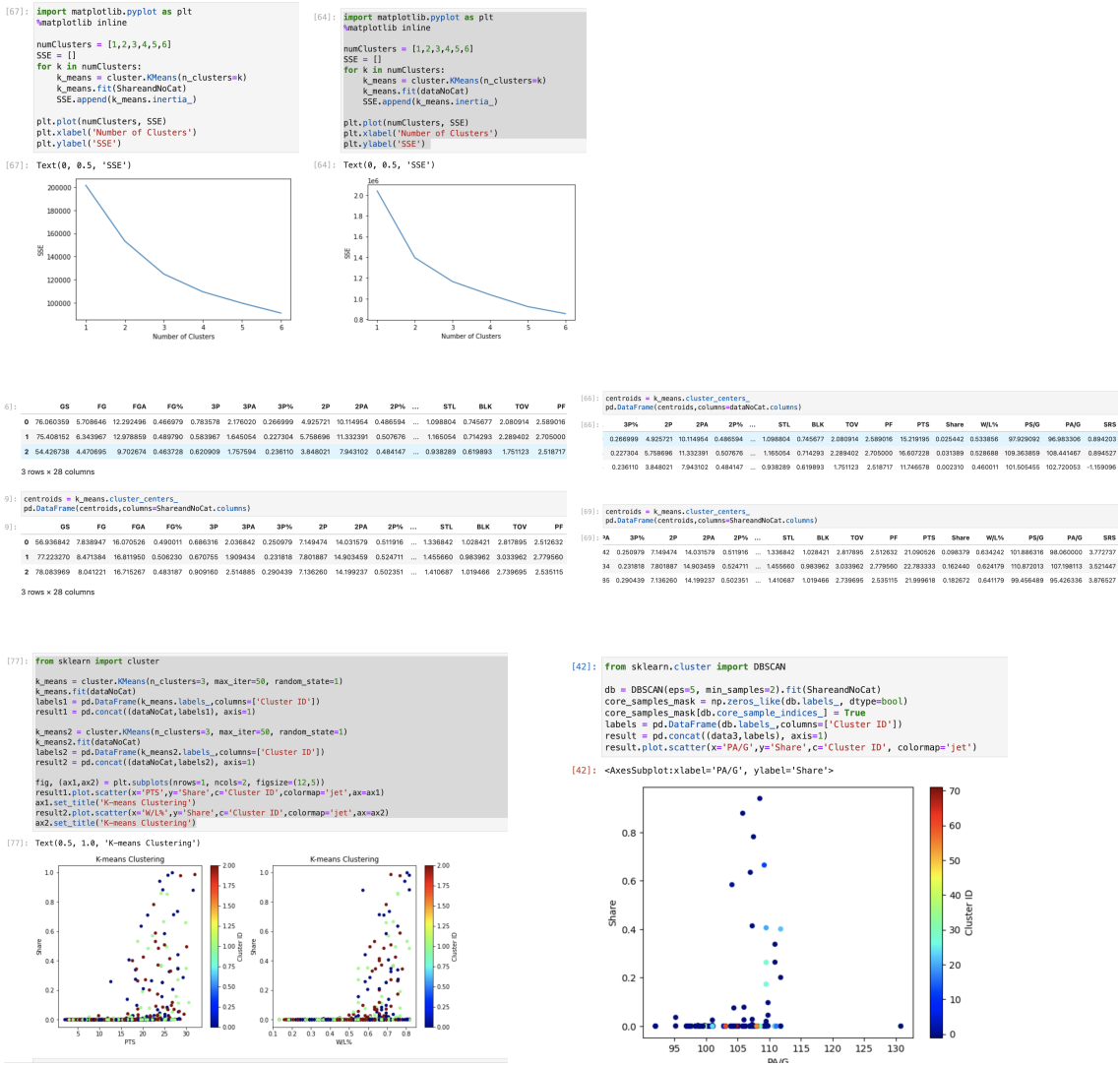
```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         512               78.0488 %
Incorrectly Classified Instances       144               21.9512 %
Kappa statistic                          0.2418
Mean absolute error                      0.1607
Root mean squared error                  0.3017
Relative absolute error                 84.4279 %
Root relative squared error             98.1066 %
Total Number of Instances              656

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.965    0.692    0.830      0.965    0.892      0.385    0.785     0.900     low
               0.116    0.026    0.238      0.116    0.156      0.127    0.641     0.110     high
               0.154    0.034    0.333      0.154    0.211      0.172    0.598     0.173     medium
               0.132    0.011    0.417      0.132    0.200      0.210    0.709     0.214     highest
Weighted Avg.  0.780    0.544    0.718      0.780    0.736      0.337    0.753     0.736

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 492   5  11   2 |   a = low
  29   5   5   4 |   b = high
  50   4  10   1 |   c = medium
  22   7   4   5 |   d = highest
```

```
=== Summary ===

Correctly Classified Instances         550               83.8415 %
Incorrectly Classified Instances       106               16.1585 %
Kappa statistic                          0.6267
Mean absolute error                      0.0935
Root mean squared error                  0.2152
Relative absolute error                 37.1697 %
Root relative squared error             60.7778 %
Total Number of Instances              656

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.368    0.853      1.000    0.921      0.734    0.958     0.966     low
               0.728    0.035    0.747      0.728    0.737      0.701    0.979     0.807     highest
               0.431    0.015    0.757      0.431    0.549      0.538    0.962     0.642     high
               0.254    0.000    1.000      0.254    0.405      0.485    0.961     0.630     medium
Weighted Avg.  0.838    0.257    0.845      0.838    0.812      0.687    0.961     0.882

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 447   0   0   0 |   a = low
  22  59   0   0 |   b = highest
  25  12  28   0 |   c = high
  30   8   9  16 |   d = medium
```

Clustering

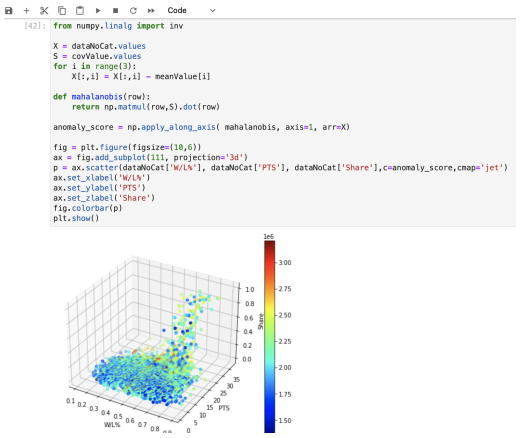
We found that in both of our datasets, the one that included every starter and the one that only contained players who received shares, that k=3 would be the best for k-means clustering.

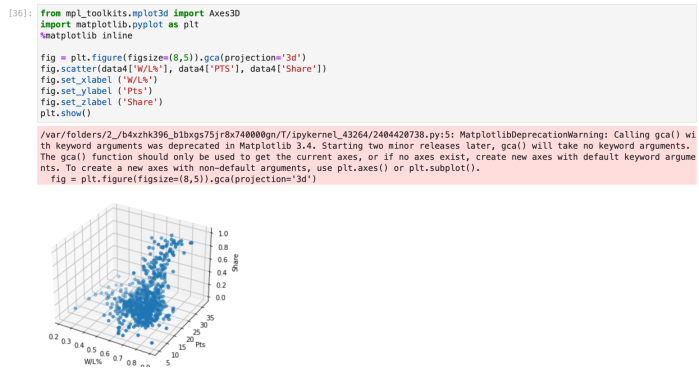
However, to do this method, we needed to delete all categorical attributes: this was only the team and position. After comparing the three clusters in the dataset that contained all of the starters, we realized that the clusters would be difficult to differentiate from each other. From cluster 0 and cluster 1, there was a 0.01 difference in share, 0.01 in W/L%, and a difference of 1.4 in PTS. Cluster 2 was the main difference because it was mostly players who did not receive votes. This did have by far the least points and W/L%, which we expected from our previous findings. In the cluster diagrams, we could clearly see that there are handfuls of variables that go into votes received.



When analyzing the data from the dataset that only used players who received shares, we saw that the highest share group did not have the highest points. Every other statistic was extremely similar between the highest share group and the highest point group, except two. These were PS/G and PA/G. We attempted to use k-means clustering to help us get a better understanding of these centroids. However, our data was extremely scattered and did not form clusters in the plots. We tried many different variables that resulted in similar issues. There are most likely too many variables that affect a players chance of receiving votes that it is too difficult to cluster these players using k-means clustering. We also attempted to use DBSCAN clustering. Again, the data points were too far apart to get decent clustering. The only true cluster was the players who received 0 votes, so when we attempted to cluster our dataset with only players receiving votes, the EPS had to be way too high for successful clusters. Even with PA/G, PTS, and PS/G we could not get decent clusters when using the share values.

Anomaly





Player	Pos	GS	FG	FGA	FG%	3P	3PA	3P%	...	PF	PTS	Year	Share	Team	W/L%	PS/G	PA/G	SRS	Anomaly score
Alex English	SF	82	10.4	18.9	0.551	0.0	0.1	0.000	...	3.2	25.4	1982.0	0.017	Denver Nuggets	0.561	126.5	126.0	0.13	3.441066e+06
Dan Issel	C	81	8.0	15.3	0.527	0.0	0.1	0.667	...	3.0	22.9	1982.0	0.004	Denver Nuggets	0.561	126.5	126.0	0.13	3.390458e+06
Alex English	SF	82	11.7	22.6	0.516	0.0	0.1	0.167	...	2.9	28.4	1983.0	0.056	Denver Nuggets	0.549	123.2	122.6	0.27	3.317854e+06
Kiki Vandeweghe	PF	79	10.3	18.7	0.547	0.2	0.6	0.294	...	2.4	26.7	1983.0	0.001	Denver Nuggets	0.549	123.2	122.6	0.27	3.254421e+06
George Gervin	SG	78	13.1	24.9	0.528	0.4	1.3	0.314	...	2.7	33.1	1980.0	0.086	San Antonio Spurs	0.500	119.4	119.7	-0.24	3.174719e+06
Alex English	SF	81	11.6	22.4	0.518	0.0	0.1	0.200	...	3.2	27.9	1985.0	0.015	Denver Nuggets	0.634	120.0	117.6	2.05	3.100984e+06
Alex English	SF	82	11.8	23.4	0.503	0.0	0.2	0.267	...	2.6	28.6	1987.0	0.003	Denver Nuggets	0.451	116.7	117.6	-1.14	3.042608e+06
Calvin Natt	PF	76	8.8	16.1	0.546	0.0	0.0	0.000	...	2.3	23.3	1985.0	0.028	Denver Nuggets	0.634	120.0	117.6	2.05	2.995476e+06
Chris Mullin	SF	82	10.1	19.9	0.509	0.3	1.2	0.230	...	2.2	26.5	1989.0	0.004	Golden State Warriors	0.524	116.6	116.9	-0.59	2.978683e+06
Chris Mullin	SF	81	10.2	19.6	0.524	0.8	2.2	0.366	...	2.1	25.6	1992.0	0.084	Golden State Warriors	0.671	118.7	114.8	3.77	2.953764e+06

```
anom = pd.DataFrame(anomaly_score, index=data3.index, columns=['Anomaly score'])
result = pd.concat((data3,anom), axis=1)
result.nlargest(10, 'Anomaly score')
```

	Unnamed: 0	Player	Pos	GS	FG	FGA	FG%	3P	3PA	3P%	...	PF	PTS	Year	Share	Team	W/L%	PS/G	PA/G	SRS	Anomaly score
160	16160	Alex English	SF	82	10.4	18.9	0.551	0.0	0.1	0.000	...	3.2	25.4	1982.0	0.017	Denver Nuggets	0.561	126.5	126.0	0.13	3.208512e+06
164	16164	Dan Issel	C	81	8.0	15.3	0.527	0.0	0.1	0.667	...	3.0	22.9	1982.0	0.004	Denver Nuggets	0.561	126.5	126.0	0.13	3.152275e+06
170	16170	Kiki Vandeweghe	PF	78	8.6	15.4	0.560	0.0	0.2	0.077	...	2.6	21.5	1982.0	0.000	Denver Nuggets	0.561	126.5	126.0	0.13	3.119320e+06
021	5021	Alex English	SF	82	11.7	22.6	0.516	0.0	0.1	0.167	...	2.9	28.4	1983.0	0.056	Denver Nuggets	0.549	123.2	122.6	0.27	3.108527e+06
684	5684	Alex English	SF	77	11.1	20.9	0.529	0.0	0.1	0.143	...	3.1	26.4	1984.0	0.000	Denver Nuggets	0.463	123.7	124.8	-1.27	3.099729e+06
690	5690	Kiki Vandeweghe	PF	71	11.5	20.6	0.558	0.1	0.4	0.367	...	2.4	29.4	1984.0	0.000	Denver Nuggets	0.463	123.7	124.8	-1.27	3.082590e+06
211	211	Michael Adams	PG	66	8.5	21.5	0.394	2.5	8.5	0.296	...	2.5	26.5	1991.0	0.000	Denver Nuggets	0.244	119.9	130.8	-10.31	3.069661e+06
228	228	Orlando Woolridge	SF	50	9.2	18.5	0.498	0.0	0.1	0.000	...	2.7	25.1	1991.0	0.000	Denver Nuggets	0.244	119.9	130.8	-10.31	3.059159e+06
030	5030	Kiki Vandeweghe	PF	79	10.3	18.7	0.547	0.2	0.6	0.294	...	2.4	26.7	1983.0	0.001	Denver Nuggets	0.549	123.2	122.6	0.27	3.030854e+06
159	16159	T.R. Dunn	SG	80	3.1	6.1	0.512	0.0	0.0	0.000	...	2.6	8.2	1982.0	0.000	Denver Nuggets	0.561	126.5	126.0	0.13	3.006031e+06

rows × 34 columns

For anomaly testing we decided to base our visualization from the tutorial 9 homework assignment. For some of the visualizations to work, we had to take out categorical variables, so we simply used data.drop and dropped all of the categorical columns. We created a 3d visualization for both our data with players who have received votes and one with all of the players. The three variables we used for this are share, W/L%, and PTS. We then calculated the anomaly score and used mahalanabis to help us find the score. We then found the top ten biggest anomalies in our two main datasets that have been used throughout this project. These anomalies were some of the most interesting and useful for our findings, which we will describe in the conclusion section.

Association Rules

Using the limited attribute data set, we examined what association rules were produced. There were many rules that related to the middle groups of players. The rules had high confidence and lift for rules that utilized a ‘middle’ FG% and eFG% to the middle 2P%, which is not surprising that these are strongly related. These attributes include the majority of the data so they don’t really distinguish between the top vote getters, so this algorithm was not extremely useful for us.

```
Best rules found:

 1. FG=medium eFG=medium WL=high 186 ==> twoP=medium 176    <conf:(0.95)> lift:(1.57) lev:(0.1) [63] conv:(6.7)
 2. FG=medium eFG=medium Share=low 202 ==> twoP=medium 191    <conf:(0.95)> lift:(1.57) lev:(0.11) [69] conv:(6.67)
 3. FG=medium eFG=medium 303 ==> twoP=medium 285    <conf:(0.94)> lift:(1.56) lev:(0.16) [102] conv:(6.32)
 4. FG=medium eFG=medium FT=high 166 ==> twoP=medium 156    <conf:(0.94)> lift:(1.56) lev:(0.09) [55] conv:(5.98)
 5. FG=medium eFG=medium WL=high SRS=high 144 ==> twoP=medium 135    <conf:(0.94)> lift:(1.55) lev:(0.07) [48] conv:(5.71)
 6. FG=medium eFG=medium SRS=high 169 ==> twoP=medium 157    <conf:(0.93)> lift:(1.54) lev:(0.08) [54] conv:(5.15)
 7. FG=medium eFG=medium PTS=high 183 ==> twoP=medium 170    <conf:(0.93)> lift:(1.54) lev:(0.09) [59] conv:(5.18)
 8. FG=medium Share=low SRS=high 155 ==> WL=high 142    <conf:(0.92)> lift:(1.54) lev:(0.08) [49] conv:(4.47)
 9. FG=medium threeP=medium eFG=medium 198 ==> twoP=medium 180    <conf:(0.91)> lift:(1.51) lev:(0.09) [60] conv:(4.13)
10. Share=low SRS=high 223 ==> WL=high 201    <conf:(0.9)> lift:(1.51) lev:(0.1) [68] conv:(3.92)
```

Neural Networks

Using artificial neural networks, we expected to increase the accuracy of our models. This however was not necessarily the case. When we ran our dataset with the default ANN settings, it did not have a higher accuracy than many of our other data mining algorithms (78.2%). When we limited our dataset to only include the most similar attributes, though, the accuracy of the model improved significantly (peaking at 86.4%). It seemed that by simplifying the construction of the network, the model was able to more aptly predict the result class. We do note, though, that there was large variability between trials.

Initial default model results:

```
=== Summary ===

Correctly Classified Instances         513               78.2012 %
Incorrectly Classified Instances       143               21.7988 %
Kappa statistic                          0.4656
Mean absolute error                      0.1105
Root mean squared error                  0.3189
Relative absolute error                 43.9387 %
Root relative squared error             90.0815 %
Total Number of Instances              656

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.991    0.522    0.803      0.991    0.887      0.599    0.887     0.920     low
                 0.704    0.045    0.687      0.704    0.695      0.652    0.944     0.738     highest
                 0.031    0.002    0.667      0.031    0.059      0.129    0.803     0.341     high
                 0.175    0.012    0.611      0.175    0.272      0.294    0.726     0.359     medium
Weighted Avg.    0.782    0.362    0.756      0.782    0.722      0.530    0.870     0.786

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 443   4   0   0 |   a = low
  22  57   0   2 |   b = highest
  41  17   2   5 |   c = high
  46   5   1  11 |   d = medium
```

Updated limited dataset results:

```
=== Summary ===

Correctly Classified Instances         567               86.4329 %
Incorrectly Classified Instances        89               13.5671 %
Kappa statistic                          0.7079
Mean absolute error                      0.0889
Root mean squared error                  0.2166
Relative absolute error                 35.3478 %
Root relative squared error             61.1848 %
Total Number of Instances              656

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.964    0.278    0.881      0.964    0.921      0.734    0.952     0.966     low
                 0.741    0.012    0.896      0.741    0.811      0.791    0.968     0.886     highest
                 0.677    0.030    0.710      0.677    0.693      0.660    0.940     0.802     high
                 0.508    0.010    0.842      0.508    0.634      0.628    0.895     0.724     medium
Weighted Avg.    0.864    0.195    0.862      0.864    0.857      0.724    0.947     0.917

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 431   3   8   5 |   a = low
  17  60   3   1 |   b = highest
  18   3  44   0 |   c = high
  23   1   7  32 |   d = medium
```

Ensemble Method

Because the accuracy and interpretability of the decision tree model was high, the ensemble method that we spent the majority of our time investigating was the random forest. This proved to be beneficial because when we used the dataset with the limited attribute set as well as constructing the model with the training set, the model's accuracy was our highest: 87.5%. The model output is below.

```
=== Summary ===

Correctly Classified Instances         574               87.5    %
Incorrectly Classified Instances        82               12.5    %
Kappa statistic                          0.7379
Mean absolute error                      0.1129
Root mean squared error                  0.2101
Relative absolute error                 44.8804 %
Root relative squared error             59.3523 %
Total Number of Instances              656

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.962    0.215    0.905      0.962    0.933      0.778    0.972     0.987     low
                 0.778    0.014    0.887      0.778    0.829      0.809    0.986     0.926     highest
                 0.677    0.030    0.710      0.677    0.693      0.660    0.980     0.862     high
                 0.587    0.019    0.771      0.587    0.667      0.643    0.977     0.837     medium
Weighted Avg.    0.875    0.153    0.871      0.875    0.871      0.757    0.975     0.952

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 430   3   8   6 |   a = low
  12  63   4   2 |   b = highest
  14   4  44   3 |   c = high
  19   1   6  37 |   d = medium
```

Conclusion

Algorithm	Best Accuracy
Decision Tree	74.44%
K Nearest Neighbor	83.84%
Artificial Neural Network	86.43%
Random Forest	87.5%

Since the beginning, we assumed we would find correlation between the attributes PTS, W/L%, and Share. The visualizations and data mining techniques we recently learned have only furthered our belief that PTS and W/L% are good predictors of the share a player receives during the MVP voting. However, we found several other important variables that elevate an MVP candidate's shares. These include assists, PS/G, and PA/G. Our decision tree was decently accurate at a successful prediction rate of 74.44%. In the decision tree, we can clearly see that PTS and high W/L% are major factors of receiving high MVP votes, as they are the two largest splits in the tree. The only way that the player can receive a high amount of votes with lower points than 23.7 PTS per game is if they have greater than 11.2 assists per game. This shows that the player needs to be very well-rounded as well as score a decent amount of points.

While creating the centroids for K-means clustering, we found that the clusters show that the high W/L%, PTS, low PS/G and low PA/G have the highest share. This cluster had an 18% share while the other two clusters had 16% and 9%. We found that the cluster with the highest points did not have the highest share, so we found the major differences which were PS/G and PA/G. The centroid with the highest share had the lowest PA/G and PS/G. This cluster’s PA/G and PS/G were lower by around twelve points each. This claim was emphasized by the top

anomalies. Almost every anomaly was from a Denver Nuggets player from 1982-1985. These players scored tons of points and had around a 60% win percentage. However, there were three players averaging over 20 points a game, and their defense gave up 120 points a game. To win these games, they needed a handful of valuable offensive players. However, a handful of great scorers on a team does not help a player be the MOST valuable player. A winning team who does not score many points usually means that only one player scores a lot of points, rather than three. This one player who scores high amounts of points is more likely to be seen as a more valuable player than the one who has a handful of scorers on. We think this is what happened in the 2021-2022 MVP race when Nikola Jokic beat Joel Embiid. Although Joel Embiid had more PTS and a higher W/L%, he finished with less votes than Nikola Jokic. Nikola Jokic was slightly more well-rounded by having higher assists by three, but the major attribute that affected the voting was that Nikola Jokic's highest teammates points per game was fifteen. Joel Embiid had four other teammates average at least fifteen points per game.

In total, the process of running the algorithms to discover these relationships proved to be difficult. Ensuring that the data types were proper for the algorithms took time because we wanted to ensure we discretized using appropriate split points and there were many attributes to consider. Some of the models were not as useful, like the association rules and clustering, since their outputs were not exactly contributing to our goal. Others were useful because they produced high accuracies, but weren't very interpretable, like the neural networks and ensemble methods. Some were slightly less accurate, but we could gather information from them, like the decision trees and k-Nearest Neighbors. All of the data preparation and manipulating of the models was a tall task but is worthwhile given the interesting conclusions we are able to draw from them.