1. Business Problem

From sport fanatics to bettors, predicting the outcome of sports matches has become as data driven as ever. Huge amounts of data are accumulated in individual games allowing for bookmakers/sportsbooks to integrate betting algorithms in order to turn a profit on bettors. Utilizing common and advanced sports statistics, these algorithms become more and more intricate and modernized as machine learning techniques are improved. Due to the vast array of data provided by sports leagues, sports betting becomes a great topic in machine learning models. This brings the question; can we utilize machine learning to predict the next MVP for 2022 seasons? The NBA MVP is utilized as an award for the most valuable player in the league, usually exemplifying a player who is crucial for team success. It was first given in the 1955-56 season where each member of the voting panel (a group of sports journalist and broadcasters) casts a vote from 1st to 5th place in deciding on the most valuable player. Coming from a business standpoint predicting the MVP would be beneficial for forecasting supply of merchandise for NBA teams. Having an MVP candidate leads to more sales of that perspective players merchandise leading to more income for the NBA as well as the perspective NBA team.

1. Data Explanation

Data was retrieved via the official NBA API where data was pulled from 1989 to 2022. Player statistics were pulled from 3 endpoints: Player Career Stats, Team Year Stats and Player Awards. The player career stats endpoint pulls commonly recorded statistics such as Points Scored in a season, Rebounds in a Season, Games Played in a Season, Minutes Played in a season etc. etc. The Team Year Stats records the teams wins, losses and rankings for a given season. The Awards endpoint allows us to retrieve whether a player won a specific award or not and, in this case, we were able to extract out MVP given a specific player ID. After data was retrieved, we stored it into pickle so that data is stored much more efficiently and is quite easy to read with Pandas. Data was saved in a local environment as the NBA API limits the number of requests you can send, this means when requesting large swaths of data from the API there will be some time for the request to complete.

Once data was retrieved it was time to merge and normalize the dataset for our model. Some non-important features were dropped such as league\_id, team\_id, team\_abbreviation as well as other playoff statistics which are not utilized for regular season predictions. Data was reshaped and normalized using SKLearns minmaxscaler to a size of (-1,1) for training using ML models.

A picture containing icon

Description automatically generated

Figure 1: Training Data

The final dataset consists of 15570 non MVP values since the season and 32 MVP values as seen as in  *figure 1*. This will be the training set which will be used to identify MVP type players.

From the 32 MVP values we extracted some means to find some useful information on what could possibly influence MVP player picks from the media. We see average shooting splits for an NBA caliber player to be that of 50% from the floor, 29% from 3 and 81% from the free throw line. We also see that MVP players averaged a 74%-win percentage equating to a 60-win season for that team. MVP players usually play for a team that averages a 1.49 Division ranking meaning that in the eastern or western conference we expect our player to be in one of the top two teams in their conference or top 4 teams in the league. Some more interesting statistics show that the MVP will normally play around 35 minutes per game and will receive the player of the week award 3.17 times in a season as well as 1.7 player of the month’s awards in a season.

Graphical user interface, application

Description automatically generated

Figure 2: Correlation Matrix of Features

*Figure 2* shows the correlation matrix of features in our dataset. We can see that so far not many of the values are very strongly correlated with MVP players, seeing that the highest correlated value player of the month carriers an R of .41 with player of the week holding an R of .35. While these values can be influential, we don’t see any strong correlations between the features. To get more of an insight of what features could play a big role in predicting MVP we can run a RandomForestClassifier to essentially extract which features mathematically play a bigger role in a decision trees ability to accurately predict an MVP.

A picture containing chart

Description automatically generated

Figure 3: Random Forest Feature Importance

*Figure 3* shows the weight of different features when using a Random Forest algorithm to classify MVP’s. Of the past 32 MVPs, we see that the highest impact features are a Players Win Percentage, Field Goals Made and Pts Scored in a season. This means that MVPs tend to play on teams with high win percentages, make a lot of field goals, and have high point totals by the end of a season. Some features which were highly correlated in *Figure 2* fall in feature importance such as player of the month and player of the week although they still carry a sense of importance in final predictions for an MVP player.

1. Methods

After exploring our data, we were able to see what features could play a role in classifying MVP players, now we plan on selecting a few models to explore how classification can be conducted using select models. In this case we will use a Logistic Classification, Adaboost, and Random Forest Model to conduct MVP classification.

The first model will be the Logistic Classification, a relatively simple classification model where linear combinations of multiple variables are able to output a prediction in the form of a probability. For the logistic model we will import the model from sklearn’s linear model modules, apply a balanced class weight and use a newton-cg solver parameter. We use Newton’s method for regression as the quadratic approximation is used. The drawbacks of Newton’s method are that it is computationally expensive and may be attracted to saddle points within the data. For simplicity a balanced class is applied, therefore we do not have to manually apply weights to each class, but the drawback is that as seen in Figure 3, we will not have one feature be more important that the other and will be treating every feature as equivalent predictor.

Our next model will utilize the Adaboost module from SKLearns library. Adaboost, which is short for adaptive boosting, is a classification algorithm, which adapts with each classification. The algorithm works by using multiple weak classifiers into a single strong classifier. The Adaboost classification algorithm uses decision trees with 1 split and puts more weight on situations where instances are difficult to classify. Adaboost defaults N\_estimators to 50, but can go up to 5000 for modeling, in this case we went with 12 estimators with a learning rate of 2. Due to the low number of MVPs in the large dataset, higher estimators led to less accuracy in the model, and since we were low on estimators, I increased the learning rate so that each estimator had a higher impact on the final ensemble.

The final model that we will use will be a Random Forest Classifier which is also from SKlearns library. The random forest classifier works by utilizing multiple decision trees and combining these individual classifiers into one classification model. The model uses various subsamples of decision trees and uses the averaging of these subsamples to improve predictive accuracy as well as control for over-fitting of data. Overall, random forests are a very good model for a lot of use cases with how adaptable they are. We are using the criterion of entropy for information gain with decision trees with a depth of 10. We are using 100 n\_estimators for this model.

1. Analysis

Chart

Description automatically generated

Figure 4: Logistic Regression Confusion Matrix

Training our Logistic Model resulted in the classifier (*figure 4)* classifying 103 players of 15667 as MVP over the last 32 years. Our actual MVP value was 31, meaning that the logistic classification model over predicts the amount of MVP caliber players. This means that by using the logistic regression model we have a 99.4 percent accuracy with 72 wrongfully predicted players.

Chart, bar chart

Description automatically generated

Figure 5: Logistic Regression MVP Prediction

Looking at the players the logit model predicts, we see in *figure 5* that Stephen Curry of the Golden State Warriors is the overall favorite at this point in the season for the MVP pick followed by James Harden and Kevin Durant of the Brooklyn Nets.

Chart

Description automatically generated

Figure 6: Adaboost Confusion Matrix

Utilizing adaboost our model proved a little more accurate with the train dataset, but miscategorized one of the validations sets MVP’s.

Chart, bar chart

Description automatically generated

Figure 7: Adaboost Predictions

Using Adaboost, our model predicted that Kevin Durant would be the MVP for the current season, with Steph Curry falling to 4th place in MVP rankings by the model. Devin Booker ended up being a new entrance in the top 10 for the model with a very high MVP consideration in the model.

Chart

Description automatically generated

Figure 8: Random Forest Confusion Matrix

The Random Forest Classifier was more selective with its selection of MVP players only selecting 20 of the 31 MVP candidates in the training dataset, although it was able to correctly classify the validation set.

Chart, bar chart

Description automatically generated

Figure 9: Random Forest Results

*Figure 9* shows the result of the Random Forest classification, selecting Steph Curry as the top MVP candidate for the 2022 season. The results were interesting, and we see 2 of the 3 models selecting Stephen Curry as the overall MVP with the Adaboost model selecting a different finalist as the overall MVP.

1. Conclusion

To conclude we find that it is entirely possible to use machine learning algorithms as solutions to decide on professional league sports awards. The application of machine learning algorithms for this purpose was very interesting, as the eye test for the season has Stephen Curry as one of the players in the top ranks of the KIA MVP ladder. Steph’s high ranking is attributed to the team win percentage as his in-game statistics are not as impressive as some of the other candidates such as Giannis Antetokounmpo who reaches the no. 1 rank of the MVP ladder and Kevin Durant who is no. 2. One of the problems, with the model is Giannis and Nikola Jokic of the Denver Nuggets, are often very low on the list or sometimes not even in the top 10 of these classification models. This is most definitely attributed to some challenges in regard to the application of machine learning models for such a task, where we are at the season, and data limitations.

1. Limitations, and Challenges

The project did face quite a few limitations, as MVP votes are conducted by Sports Journalists, while statistics play an important role in the decision-making process of who will become MVP, it is not necessarily able to take other things into account such as a narrative being built around a player, the eye test, and voter fatigue. Our machine learning models don’t watch NBA games, so they are not able to make inferences that someone who has been making these decisions for years.

One limitation about the dataset, is that there have only been 67 MVP awards given out since the league started in a dataset of over 25,000 players giving us a small sample of MVP caliber players to train our model on. We also did not use any advanced statistics for basketball such as VORP (value over replacement) eFG% (efficient field goals) or PER (Player Efficiency Rating). The use of advanced statistics is often argued whether they carry any importance in deciding on the ability of a player, but they still carry signs of advanced play. Another limitation and challenge are that our testing dataset only carries data for the current data of January 9th, 2022, which is 3 months into the 8-month season. Using a complete season for end of season predictions would yield better results as MVP awards are decided at the end of season, essentially are dataset is too small for a completely confident prediction.

1. Future Uses/Recommendations

There are many ways to improve the ability to predict the MVP as part of a machine learning assignment. One of the easiest solutions is to incorporate some advanced statistics not found in the NBA API as well as to use a fuller season to make the predictions. Obviously, at this point in time we cannot use that, but if we were to rerun this model at the end of season in May we would get better results. Another interesting solution to the narrative problem could be to use sentiment analysis. For example, reading through new sights, or a social media like twitter, we could see the total number of positive and negative tweets or articles for a player which could give us a sense of when and how to interpret the best possible way for a model to figure out the narrative of an MVP race.

1. Ethical Assessment

The first ethical consideration is using a model for generating money. In sports betting it is profitable to bet on teams with lower odds as they often generate greater profits. This model will not be meant for any real-life sports betting uses. It is for educational purposes only, and to learn from the process of creating a model. Another project specific ethical issue is the ability to grab and data which produced a lot of issues with this project. Retrieving the data, I needed resulted in a 3 hour wait time for the API to send all the data due to the NBA limiting requests. The NBA limits API requests in order to prevent any attacks. If an API is completely open, and doesn’t feature limitations for requests and other security protocol, they could be completely open to outages and attacks.

1. References

<https://www.nba.com/news/kia-mvp-ladder-jan-7-2022-edition>

<https://towardsdatascience.com/predicting-2020-21-nbas-most-valuable-player-using-machine-learning-24aaa869a740>

<https://scikit-learn.org/stable/>