Final Report

DATA 1030

Kaiwen Yang

Brown DSI

GitHub Repository: https://github.com/Melo96/data1030-project.git

Introduction

The goal of this project is to use machine learning methods to explore the relationship between players' usage rate and his stats. Usage rate is an estimate of the percentage of team plays that a player involved while he was on the floor. In one offense play, a player is considered involved if the play is ended by either this player took a shot, turned the ball over, or drew a foul (Basketball Reference.com). Thus, essentially players who tend to handle the ball a lot would have high usage rate, while players who pass a lot or only focus on defense would have low usage rate. The result of this project is expected to help NBA teams make decision about how much should players on the floor involve in game plays given their expected performance.

In "nba_shot_types.csv", there are 3007 data points and 23 features. Here are column descriptions:

"YEAR": NBA season. Category.

"PLAYER": Player name. Category.

"TEAM": Team of the player. Category.

"AGE": Age of the player. Integer.

"GP": Number of games the player played. Integer.

"W": Team wins. Integer.

"L": Team losses. Integer.

The following data type are all float:

“MIN”: Average minutes per game.

"PCT_FGA_2PT": Percentage of Field Goal Attempts That Were 2 PT Shots

"PCT_FGA_3PT": Percentage of Field Goal Attempts That Were 2 PT Shots

"PCT_PTS_2PT": Percentage of Points That Came From 2 PT Field Goals Made

"PCT_PTS_MR": Percentage of Points That Came From Midrange

"PCT_PTS_3PT": Percentage of Points That Came From 3 PT Field Goals Made

"PCT_PTS_FSTBRK": Percentage of Points That Came on Fast Breaks

"PCT_PTS_FT": Percentage of Points That Came From Free Throws

"PCT_PTS_OFF_TOS": Percentage of Points That Came Off Turnovers

"PCT_PTS_INTHEPT": Percentage of Points That Came In the Paint

This dataset is from Kaggle. Two projects have used this dataset. The first one is called “PCA and Clustering Analysis of NBA Shot Selections”. This project used this dataset to find the offensive types of players that exist in the NBA. The second one is called “Starter: NBA Shot Types 2013-2019 f3f7fff6-4”. This is a auto-generated kernel by Kaggle. It provides starter code to read in data.

In "Seasons_stats_complete.csv", there are 26.1k of data points and each has 50 features. Only data between 2013-2019 will be used. Here are column descriptions for columns that will be using:

"Year": NBA season. Category.

"Player": Player name. Category.

"Pos": Position. Category.

“Age”: Age of the player. Integer.

“Tm”: Team of the player. Category.

“G”: Number of games the player played. Integer.

“MP”: Average minutes played of the player. Float.

“PER”: player efficiency rating. Float.

“TS%”: True shooting percentage. Float.

“eFG%”: effective field goal percentage. Float.

“OBPM”: Offensive box plus/minus. Float

“DBPM”: Defensive box plus/minus. Float

“BPM”: Box plus/minus. Float

“VORP”: Value over Replacement Player. Float.

“FG”: Field goal made. Integer

“FGA”: Field goal attempt. Integer

"OWS": offensive win share. Continuous numerical.

“DWS”: Defensive win share. Continuous numerical.

“WS”: Total win share. Continuous numerical.

“2P”: 2-point made. Integer.

"2PA": 2-point attempt. Integer.

“3P”: 3-point made. Integer.

"3PA": 3-point attempt. Integer.

“FT”: Free throw made. Integer

"FTA": Free throw attempt. Integer

“ORB”: Offensive rebound. Integer

“DRB”: Defensive rebound. Integer

“TRB”: Total rebound. Integer

“AST”: Assistance. Integer

“STL”: Steal. Integer

“BLK”: Block shots. Integer

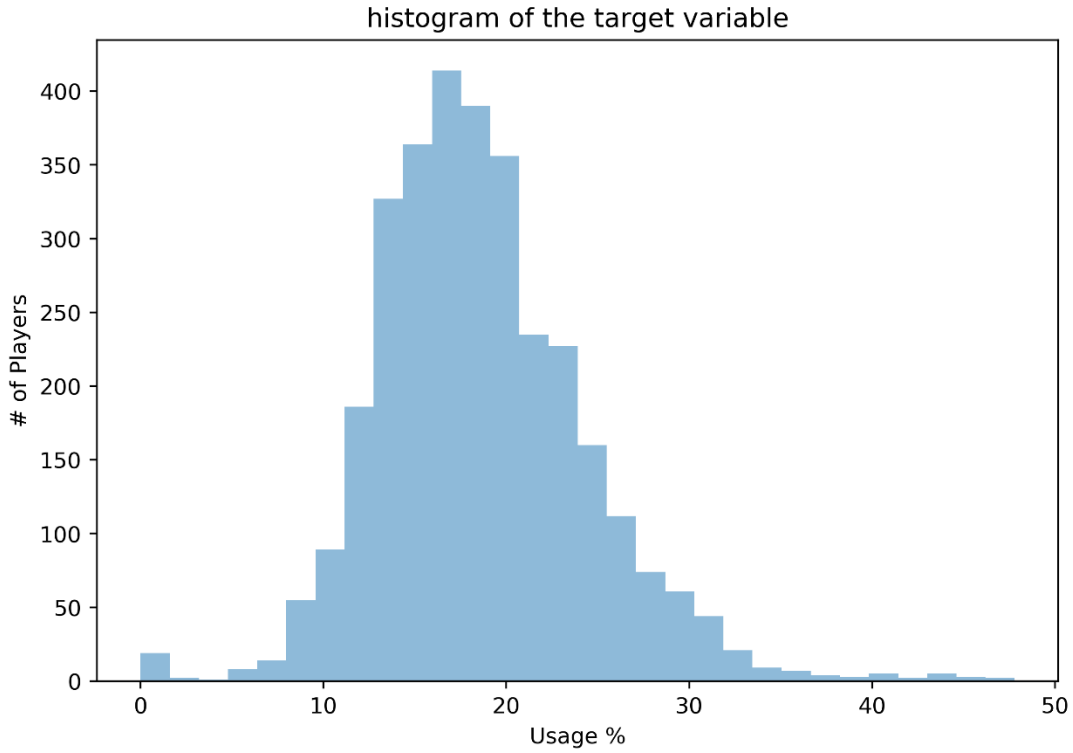
“TOV”: Turnovers. Integer

“PF”: Personal fouls. Integer

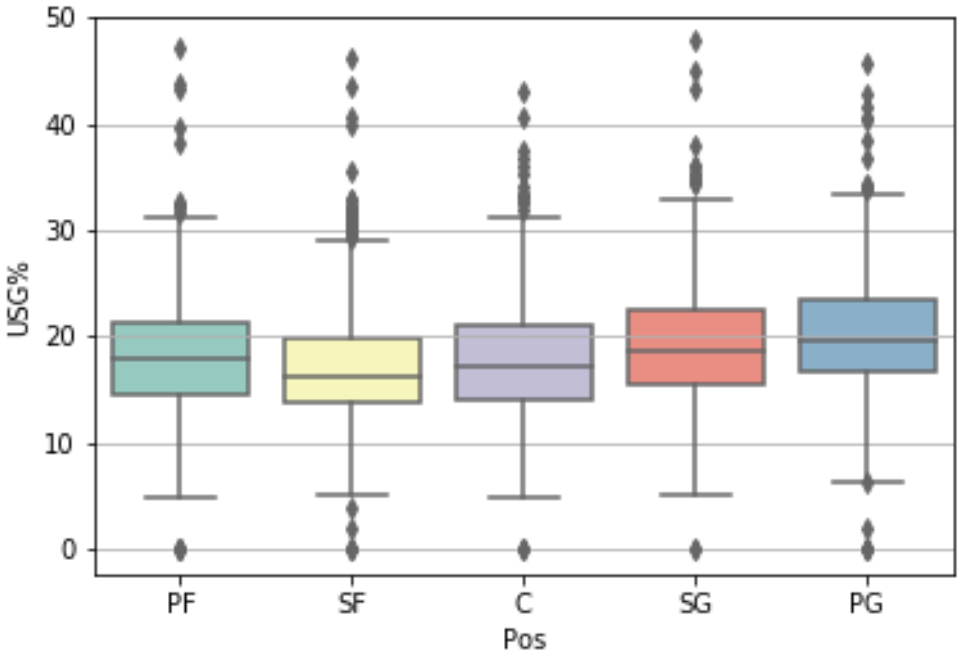
“PTS”: Points. Integer

This dataset is also from Kaggle. Three projects have used this dataset. The first one is called “How 3p shooting is changing NBA”. This project aims to explore how 3-point shot changes NBA. The second one is called “90s vs 00s”. This project was trying to compare the stats between the top ten players from 90s and 00s. The third one is called “Starter: Seasons_stats_50_19 8faa34c4-1”. This is also a auto-generated kernel by Kaggle providing starter code.

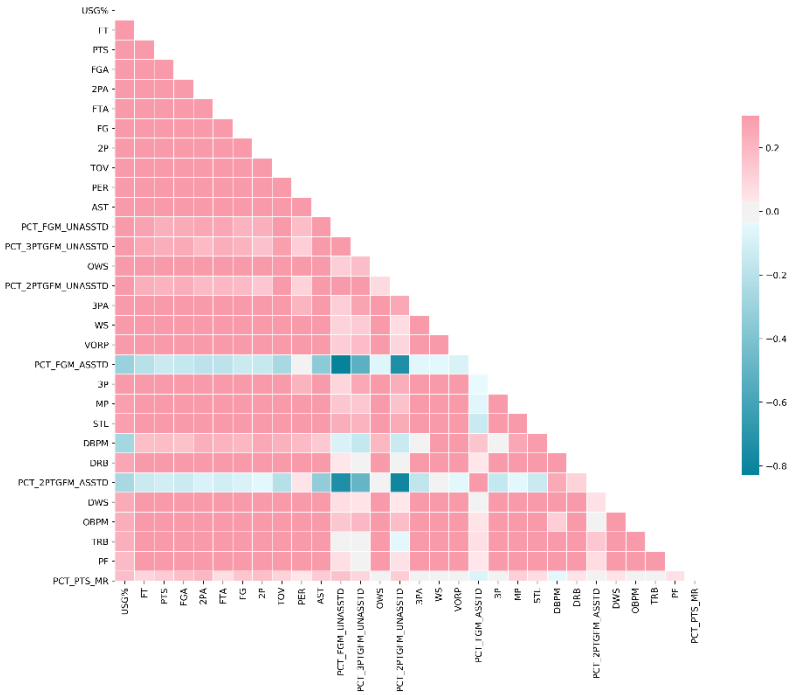
EDA



This is a histogram of the distribution of target variable. The mean of this distribution is 18.62, with standard deviation equals to 5.74. The distribution is a little bit right skewed because some super-star level players tend to have much higher usage rate than average.



This figure shows the distribution of usage rate among different positions. As we can see, PG (point guard) has the highest mean usage rate and SF (small forward) has the lowest. In modern basketball, the role of point guard is to lead the team offense. It means that they need to handle and pass the ball a lot. As a result, it is reasonable that their average usage rate is higher than the other positions. The role of small forward, on the other hand, is changing to mainly “3D” player, which means the main mission is to defense and shot open 3-pointers in offense. Because of that, small forwards are expected to be less involved in team offense, which matches the distribution above.



This graph is the correlation matrix of the dataset. The target variable is positively correlated to most of the main stats such as points and shot attempts. This is because players with higher stats are usually good players, and good players always play an important role in both offense and defense.

The three negatively correlated features among the top 30 absolute correlation are overall and 2-point field goal assisted rate and defense box plus/minus. This implication matches exactly about an increasingly crucial role in modern basketball: 3D player. This type of players has excellent defense but are less involved in offense, so they have high OBPM but low usage rate. Additionally, usually they only take open shots, and nearly all open shots are assisted. Thus, there is also a negatively correlation between filed goal assisted rate and usage rate.

Methods

The dataset that was used in this project is merged by two different datasets: “nba_shot_type.csv” and “Seasons_stats_complete.csv”. There are a lot of missing values in the early seasons because some advanced stats were not calculated in those early years. But in this project only data after 2013-2014 season were used, so there are no such missing values. The only type of missing values occurs when some players do not have any stats count, so features that represent the percentage sometimes appear as 0. Because we have all the original stats, columns that represent percentage were dropped to avoid missing values problem. This is viable because percentage data can be calculated directly by the original stats whenever we need them. There were also some duplicate rows, where the value of “Team” feature is “TOT”, that show the player’s total stats in a given season. This situation occurs when a player was traded to a different team during the season. To avoid duplication, rows that the value of “Team” is “TOT” were dropped.

In preprocessing, standard scaler was largely used because most of the features are numerical and they have tailed distributions. One-hot encoder was applied to “YEAR”, “Pos” and “TEAM” because they are categorical and not binary. After preprocessing, the data size is: 3199 rows, 85 columns, and 271,295 data points.

In this project, three ML pipeline were used: linear regression, random forest and XGBoost. The metric I used in all three models are R-squared score. One main reason is that R-squared score is easy to interpret because the baseline is given, which is 0, and the model performs better as the score approaches 1. Additionally, R-squared score is

easy to calculate. Many models have built-in methods for R-squared score. Thus, R-squared score can be calculated by simply calling that method.

In linear regression, I tried both Ridge and Lasso regularization. The R-squared score of both Lasso and Ridge are 0.79. The parameter I tuned is alpha. The alpha values I tried are logspace(-5, 2, num=29). The best alpha is usually between 0.003 and 0.01.

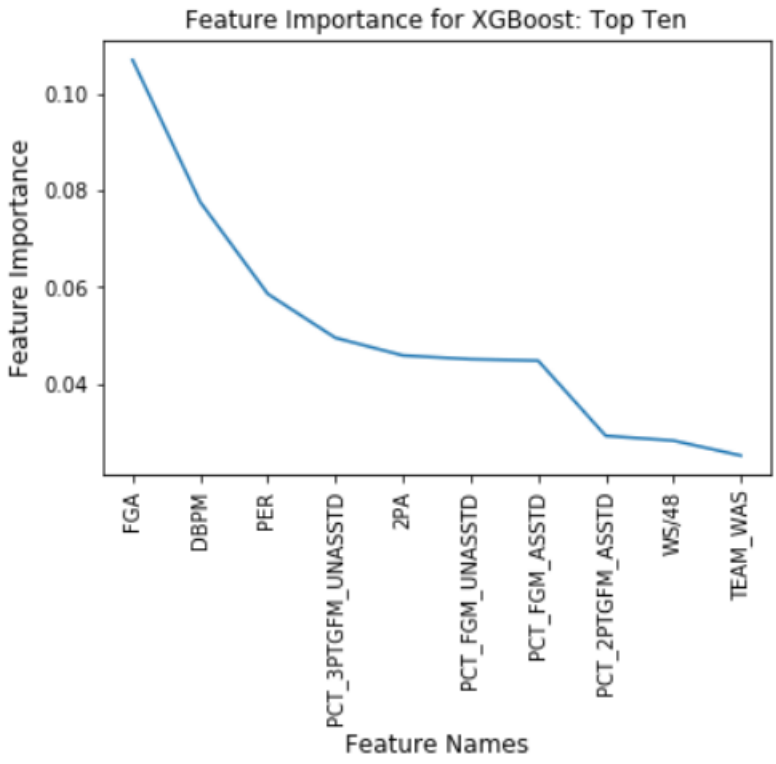
In random forest, I tuned max depth and max features. Both max depth and max features I tried were between 1 and 30. The best max depth is usually around 20 and the best max features is usually around 27. After the best parameters found, I fixed n estimators to 30 because it gives the best performance. Repeat the training 5 times, the best R-squared score is around 0.74.

In the last model, XGBoost, many parameters can be tuned. I choose to tune max depth, sub sample and alpha because these three parameters make the most influence on the performance of the model, results derived from this article: https://towardsdatascience.com/fine-tuning-xgboost-in-python-like-a-boss-b4543ed8b1e. For max depth, I tried 1, 3, 10, 30, and 50. For sub sample, I tried 0.3, 0.5, 0.66 and 0.8. For gamma, I tried 0, 1, and 5. The training process was repeated 5 times, and the best parameters varies. The resulting best R-squared score on average is 0.83.

To reduce the uncertainties due to splitting data, I used different random state and repeat the training process several times, put the resulting score in a list and finally calculated the average.

Result

Overall, all three models have good predictive power. For R-squared score, the baseline is 0. All three score are above 0.7, which means they all do a better job than a baseline model. Among the three models, XGBoost has the best performance as it has the highest R-squared score 0.82. To see what features are the most important, I plotted a line graph to show the feature importance of the top 10 most important features. The graph is shown below:



Several conclusions can be made from this graph. “FGA” (field goal attempt) has the highest feature importance. It means that compare to other stats, if this feature was shuffled, the accuracy of the model decreases the most. The second most important feature is “DPBM”, which indicates how good a player is in defense. As mentioned

before, good defense players tend to participate less in offence, so “OBPM” negatively influence the usage rate the most. An interesting finding is that “PCT_3PTGFM_UNASSTD” (3-pointers made unassisted) and “PCT_FGM_UNASSTD” (field goal mode unassisted) are also important features. Nowadays in NBA, players who made a lot of shots that are unassisted are James Harden, Steph Curry, Kevin Durant and other all-star level players. This type of players is extremely crucial to their team offense, so they have extremely high usage rate, and they tend to play more isolation offense during the game so many of their shots are unassisted. Therefore, the number of unassisted shots made by a player has a strong relation to his usage rate.

The findings above give many implications about how NBA teams can arrange offense plays to effectively balance usage rate of different players on a team. For example, if a team want to sign some players and they already have a few players that have a high usage rate, then they can sign players that are good at taking open shots (assisted shots) to balance the usage rate. Another important application would be that NBA coaches can use the implications of this model to adjust the offense habit of players to balance their usage rate. For example, Carmelo Anthony is an NBA player who is famous for his excellent isolation offense. Player who makes a lot of isolation offense have high usage rate. When signed by Portland Trail Blazers this year, the team wants to reduce his usage rate, so he was asked to shoot more open 3-pointers and participate more in defense. This is exactly what this model suggested.

Outlook

In this project, only limited amounts of models are implemented, so I might want to try this out on other models to see if there is a better prediction. Another weak spot is that, there are many parameters in XGBoost that can be tuned to potentially improve the performance of the model. In this project I only tuned 3 parameters due to the limitation of time and computer performance. Last but not least, this project is essentially to help NBA teams build up their team chemistry, but only usage rate was taken into account. This is just an intuitive idea. In the future, I will try to come up with more sophisticated methods to figure out the magic of team chemistry.

Reference

1. "Glossary." *Basketball Reference*, https://www.basketball-reference.com/about/glossary.html .
2. Revert, Felix. "Fine-Tuning XGBoost in Python like a Boss." *Medium*, *Towards Data Science*, 20 Sept. 2019, https://towardsdatascience.com/fine-tuning-xgboost-in-python-like-a-boss-b4543ed8b1e.
3. Lancharro, David. "NBA stats until 2018-2019". *Kaggle.* https://www.kaggle.com/lancharro5/seasons-stats-50-19
4. Wasserman, Josh. "NBA Shot Type 2013-2019". *Kaggle*. https://www.kaggle.com/joshuawasserman/nba-shot-types-20132019