NBA Game Winner Classifier and Player Stats Predictor

Prabjyot Obhi, Janaarthana Harri Palanisamy, Philip Salire

Spring 2021 CMPE 255 Project Report, SJSU

https://github.com/psalire/cmpe255-term-project

(Dataset also available in the repository)

1. Introduction

Sports generate an abundance of data points for each game played. Specifically, game and player statistics provide interesting datasets that people have always been analyzing and using as a basis of prediction such as for game winners, points margins, etc. Game and player statistics provide high dimensionality datasets that are great candidates for data mining and building models for prediction. This field of applying data mining to sports data is known as "sports analytics," which serves to aid sports teams in their strategies and coaching practices. It is a very fast growing field and most professional sports teams now employ data scientists to analyze sports data [3]. In addition, predicting sports statistics is a large market with regards to sports betting; worldwide, the market size for sports betting has been estimated to be as high as 608 Billion USD [1]. Therefore, there is high demand for an accurate predictor for sports data.

In this project, we apply data mining techniques to NBA (National Basketball Association) game and player statistics. These statistics can be leveraged to build a plethora of different types of predictors as the data ranges from high-level data to low-level, granular data. For example, within NBA data we can analyze high-level data such as a team's overall PPG*, RPG*, APG*, etc on the season, or we can analyze low-level, granular data such as a player's statistics only when playing against a certain other player, the FGP* of players when shooting on the left side of the three-point line, the statistics of a team when playing on a Sunday, etc. Within this project however, we focus only on analyzing high-level team and player statistics to build game winners and statistics classifiers.

For our models, we look at cumulative statistics of a team or player on a given day in a season. For example, when predicting the game winner of the 42nd game in a season, we use the aggregated statistics of the 1st to 41st game for our classifiers and predictors. This aggregation only accounts for the games within the same season, and each team plays 82 games each season with a possible additional maximum of 32 games if a team plays in the post-season playoffs.

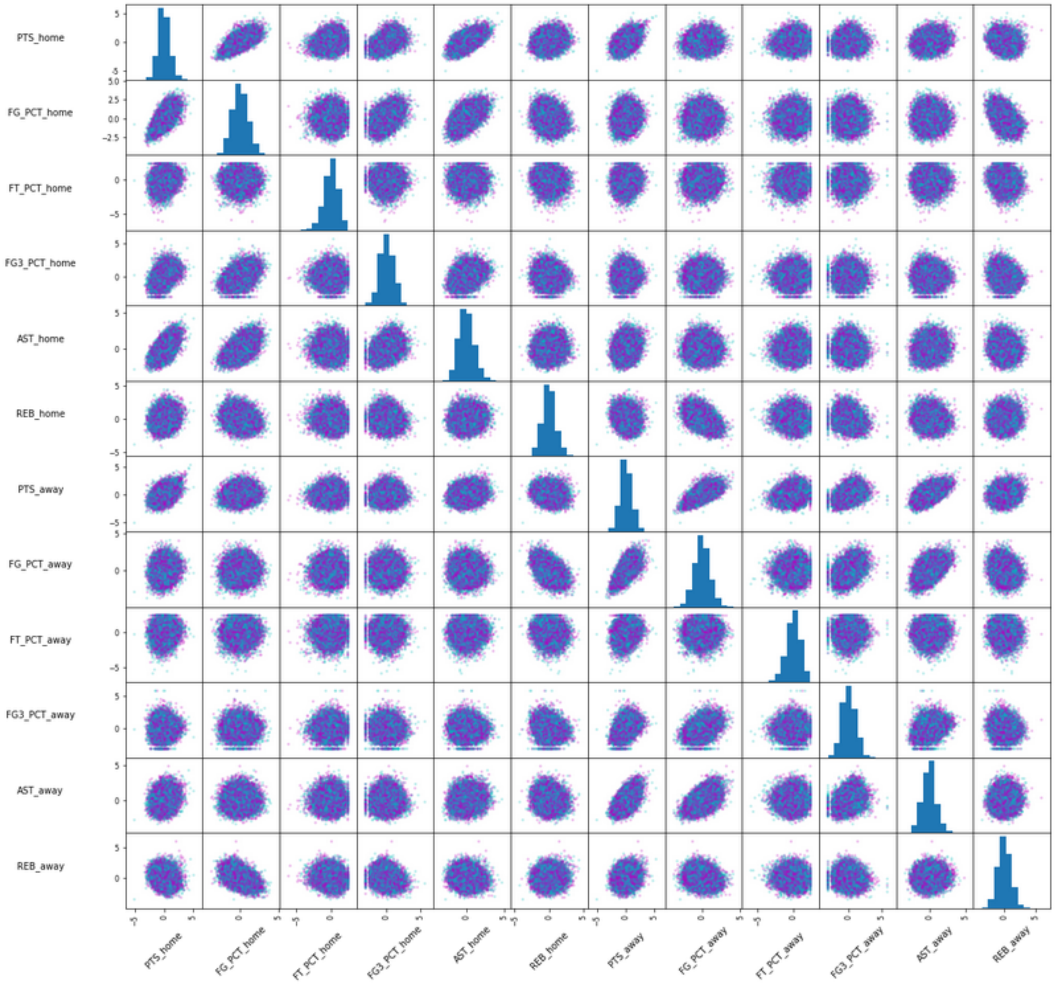
* PPG, RPG, APG, FGP respectively mean Points-Per-Game, Rebounds-Per-Game, Assists-Per-Game, and Field-Goals-Per-Game

2. System Design & Implementation

The main application for the model trained on this data would be for predicting the results of a game, e.g. the winner and the statistics, given inputs of the previous cumulative statistics of two teams. Therefore generally, an actual system to implement this must continuously keep track of statistics and accumulate them into a single database, then use this database to train the model with each update. Retrieving the data is simple to implement as it would involve a simple scraper or API but possibly time consuming, so the main focus on this project is on implementing an accurate model.

Designing our model application and deciding how to approach the data first required preparing the dataset. The dataset that we obtained had an issue that made it unusable for training predictors: the dataset provides only the stats of a game *after* the game's completion. Therefore, the data can serve as target prediction data, but not as training data. To solve this, this dataset was used to build a new dataset that includes all cumulative stats of the playing teams *prior* to a given date. This new dataset that was built served as the training data, and the original dataset was used to generate the target data.

The resulting training features include stats for both home and away teams, which are displayed in the following figure in a scatter matrix:

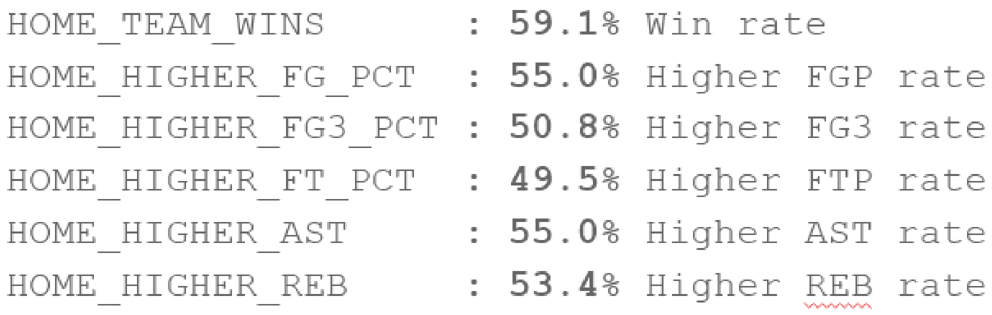


There exist no apparent clusters and the majority of the data has no patterns or relationships e.g. linear relationships. Therefore, methods such as clustering or regression are respectively not readily feasible and not applicable. Regression may be useful to reduce the dimensionality by applying to features that show linear correlations, but as of now, the total features is not overly excessive to necessitate this in initial attempts. As a result, as the intuitive approach we apply supervised learning models with the sklearn library, which was used since sklearn is the most convenient to use, including decision trees (`DecisionTreeClassifier`), random forests (`RandomForestClassifier`), and XGBoost (`XGBoostClassifier`). In addition, class balancing is also applied for which library `imblearn` is used.

3. Proof of Concept Evaluation

With a now usable training set, it was necessary to determine what target data to use. As determined in the previous section, we decided to use classifiers. As such, we generate binary target data for each stat in the training set: whether the home team wins, whether the home team has a higher FGP, whether the home team has higher AST, etc. These are labeled as `HOME_TEAM_WINS`, `HOME_HIGHER_FG_PCT`, `HOME_HIGHER_FG3_PCT`, `HOME_HIGHER_FT_PCT`, `HOME_HIGHER_AST`, `HOME_HIGHER_REB`.

Before training, it would also be useful to determine what evaluation metrics to use and what scores would be considered successful. Initially, it may seem that given that our target data is all binary, achieving higher than 50% accuracy can be considered a success. However, taking the mean of the binary target data reveals the true target accuracy to beat. For example, taking the mean of `HOME_TEAM_WINS` outputs 0.59. That means if one were to always guess that a home team wins in any game, he or she would be correct 59% of the time. Therefore, whether the trained model has a greater accuracy than the mean of the target data reflects whether the model is good, and as such we use it as a benchmark accuracy for our models.



```
HOME_TEAM_WINS       : 59.1% Win rate
HOME_HIGHER_FG_PCT   : 55.0% Higher FGP rate
HOME_HIGHER_FG3_PCT  : 50.8% Higher FG3 rate
HOME_HIGHER_FT_PCT   : 49.5% Higher FTP rate
HOME_HIGHER_AST      : 55.0% Higher AST rate
HOME_HIGHER_REB      : 53.4% Higher REB rate
```

These means, which show the proportion of the two target classes, also shows that class imbalance exists i.e. HOME_TEAM_WINS has 0.59 proportion of 1s to 0s. Therefore, it may be beneficial to the model accuracy to apply class balancing to the data.

The approach to training and evaluating the models is as follows:

1. Use 5-fold cross-validation to effectively generate the training and test sets
2. Pre-process dataset with class balancing
3. Train with the best pruned model, which is found by conducting a hyperparameter search optimizing for accuracy via `RandomizedSearchCV`.
4. Consider the result successful if the macro average accuracy of the mean accuracy of the 5-fold cross-validation is greater than the macro average accuracy of the target data mean. I.e., `macro_avg_prediction - macro_avg_target>0`

The following tables are results using the steps listed above with decision trees, random forests, and XG boost. Different class balancing methods were applied and only the successful methods are shown.

<u>Decision tree results</u>

- `RandomOverSampler` - ΔMacro Avg Accuracy: +2.8%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
`HOME_TEAM_WINS`	59%	none	64%	Y
`HOME_HIGHER_FG_PCT`	55%	max depth	58%	Y
`HOME_HIGHER_FG3_PCT`	51%	max features	52%	Y
`HOME_HIGHER_FT_PCT`	50%	none	52%	Y
`HOME_HIGHER_AST`	55%	max depth	58%	Y
`HOME_HIGHER_REB`	53%	max features	56%	Y

- `SMOTEENN` - ΔMacro Avg Accuracy: +7.8%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
`HOME_TEAM_WINS`	59%	none	66%	Y
`HOME_HIGHER_FG_PCT`	55%	max leaf nodes	63%	Y
`HOME_HIGHER_FG3_PCT`	51%	max depth	60%	Y
`HOME_HIGHER_FT_PCT`	50%	none	59%	Y
`HOME_HIGHER_AST`	55%	max depth	62%	Y
`HOME_HIGHER_REB`	53%	max depth	60%	Y

Random forest results

- `RandomOverSampler` - ΔMacro Avg Accuracy: +2.7%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
`HOME_TEAM_WINS`	59%	max depth	66%	Y
`HOME_HIGHER_FG_PCT`	55%	max depth	58%	Y

HOME_HIGHER_FG3_PCT	51%	max leaf nodes	51%	N
HOME_HIGHER_FT_PCT	50%	max depth	52%	Y
HOME_HIGHER_AST	55%	max depth	59%	Y
HOME_HIGHER_REB	53%	max depth	55%	Y

- SMOTEENN - ΔMacro Avg Accuracy: +13.3%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
HOME_TEAM_WINS	59%	max depth	71%	Y
HOME_HIGHER_FG_PCT	55%	max depth	69%	Y
HOME_HIGHER_FG3_PCT	51%	max leaf nodes	64%	Y
HOME_HIGHER_FT_PCT	50%	max depth	65%	Y
HOME_HIGHER_AST	55%	max depth	68%	Y
HOME_HIGHER_REB	53%	max depth	66%	Y

XGBoost results

- RandomOverSampler - ΔMacro Avg Accuracy: +1.8%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
HOME_TEAM_WINS	59%	gamma	64%	Y
HOME_HIGHER_FG_PCT	55%	gamma	57%	Y
HOME_HIGHER_FG3_PCT	51%	gamma	51%	N
HOME_HIGHER_FT_PCT	50%	max depth	51%	Y
HOME_HIGHER_AST	55%	gamma	58%	Y

HOME_HIGHER_REB	53%	gamma	53%	N

- SMOTEENN - ΔMacro Avg Accuracy: +14.1%

Target	Target Accuracy	Pruning Method	Prediction Accuracy	Beat Target?
HOME_TEAM_WINS	59%	max depth	74%	Y
HOME_HIGHER_FG_PCT	55%	none	68%	Y
HOME_HIGHER_FG3_PCT	51%	max depth	65%	Y
HOME_HIGHER_FT_PCT	50%	max depth	64%	Y
HOME_HIGHER_AST	55%	max depth	70%	Y
HOME_HIGHER_REB	53%	max depth	67%	Y

In conclusion, the best model macro average accuracy gains are as follows:

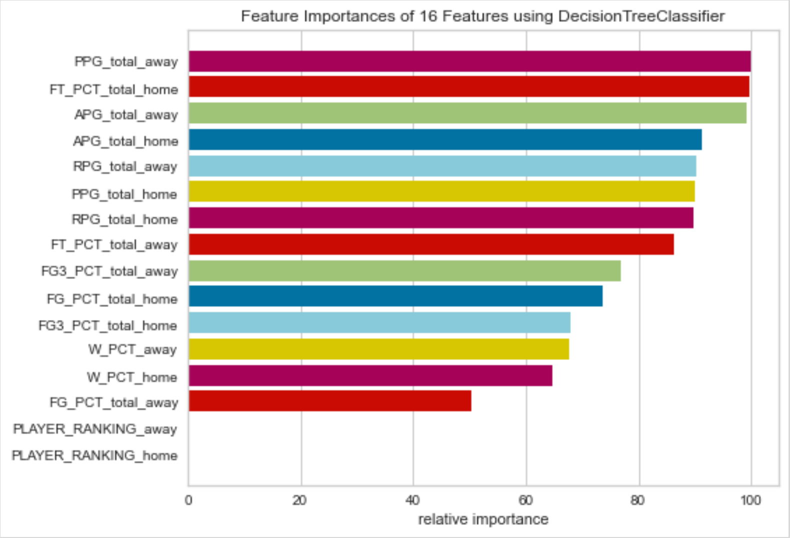
Classifier	Macro Accuracy Gain
Decision Tree	7.8%
Random Forest	13.3%
XGBoost	14.1%

4. Conclusions

The approach and application of this project was mostly an exercise on supervised learning methods for classification. Unsupervised learning methods and regression models were also attempted, but seemed either unfeasible for our dataset or out of our scope of knowledge.

We accomplished training classifiers on binary target data for predicting game winners and which team will finish a game with higher stats with regards to PTS, FGP, FG3, AST, and REB. All models scored good improvements in accuracy from pure guessing, with decision trees having a 7.8% macro average gain, random forests having a 13.3% macro average gain, and XG boost having a 14.1% macro average gain. All classifiers benefited the most with SMOTEENN class balancing, which is a hybrid of over and under sampling. Without any class balancing, prediction accuracies were very close to 50% across the board, which is worse than the target accuracies. RandomOverSampler was used which provided good accuracy gains but not as much as SMOTEENN. Furthermore, RandomUnderSampler was also used which provided accuracies that were nearly the same as without class balancing i.e. 50% across the board. Future work for this project includes further improving the accuracy and implementing regression models.

Our implementation of the supervised models relies on analyzing past statistics. We look only at high level statistics on the team level, and even then achieve good accuracies. Therefore, with this initial success in using only high level data, it becomes compelling to analyze more data at more granular levels to further improve prediction accuracies. For example, we did attempt to incorporate player stats in team level predictions by performing a dimensionality reduction on player stats and including the reduced data into the training set, but were unsuccessful in this as attempts resulted in the models outright ignoring player stats as seen in the feature importances plot below with PLAYER_RANKING having 0 importance. Every single feature importance plot for every fold showed the same 0 importance result. However, players do undeniably play a factor in the result of games, so successfully incorporating this into the models should be of benefit. Furthermore, models can also be trained for predicting player data, and the outputs of which can even be used to further train models for team data. In addition, industry (i.e. sports analytics) focus much on granular data such as the performance of one player when playing against another specific player, the performance of a team when playing against another team with certain playstyles, etc. This is a much larger task, but surely including the results of these types of models would also be a strong factor in predicting game winners and statistics.



Our models predict the binary outcomes of games and statistics. However, it may be of more use to predict the actual numerical values of these stats. This would require training regressors, which we could not successfully do with the used dataset due to lack of apparent relationships in data that can be leveraged with regressors, i.e. by interpreting the scatter matrix of training data. Sports betting often involves predictions that regressors output (e.g. point margins, points scored), so this is an important area of future work.

Lastly, the only scoring metric that we focused on was overall accuracy. We selected accuracy because it was important to account for every decision made regardless if it was a true/false positive or true/false negative. More analysis will involve evaluating metrics at more granular levels such as respective accuracies for predicting 0s and 1s e.g. it is possible that our models may be more accurate at predicting home losses than home wins. This is important to note especially if more work is done to apply the models to sports betting.

5. Task Distribution

Each of us worked on different classifiers mostly independently, then came together afterwards with our findings. The idea was to then take the best findings from each person's implementation to further improve each respective model.

Philip was assigned the decision tree classification, Janaarthana was assigned the XG boost classification, and Prabjyot was assigned the random forest classification. Our collaborative work resulted in us applying class balancing to yield the best accuracies.

References

[1]: "Sports Betting Market." *Transparency Market Research*, www.transparencymarketresearch.com/sports-betting-market.html.

[2] "Sports Analytics Market by Sports Type." *Research and Markets*, www.researchandmarkets.com/reports/4904383/sports-analytics-market-by-sports-type

[3]: Apostolou, Konstantinos, and Christos Tjortjis. "Sports Analytics algorithms for performance prediction." *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2019.