Prediction of NBA Games Using Machine Learning

Rohan Agarwal
BITS Pilani K. K. Birla Goa Campus
Goa, India 403726
f20180123@goa.bits-pilani.ac.in

Susmit Wani
BITS Pilani K. K. Birla Goa Campus
Goa, India 403726
f20180116@goa.bits-pilani.ac.in

Swapnil Ahlawat
BITS Pilani K. K. Birla Goa Campus
Goa, India 403726
f20180178@goa.bits-pilani.ac.in

Wandan Tibrewal
BITS Pilani K. K. Birla Goa Campus
Goa, India 403726
f20180094@goa.bits-pilani.ac.in

Abstract

National Basketball Association (NBA) is the men's professional basketball league in North America. A tremendous amount of research is going on in predicting the result of NBA matches as the NBA has managed to record very detailed statistics of each match. This project is a comparative study of various models for prediction of Win/Loss of a basketball game based on the team's as well as players' past statistics. We have used 3 types of datasets:- only team statistics, only players statistics and combined team and player statistics to compare the amount of impact they have on the match outcome. Apart from trying different models for classification, the project is also focused on the web scraping techniques to scrap raw datasets from the nba/stats website and feature engineering on the collected datasets to best suit the classification problem. Along with the standard features like win average, points difference, etc., a special emphasis has been given to attributes like playing as a home team or as a visitor team and statistics for the last 8 matches to depict the current form of the team and players. The best accuracy was seen for the Multi-Layer Perceptron Classifier which came out to be around 66.8% when a combination of both player and team features are used for classification.

1. Introduction

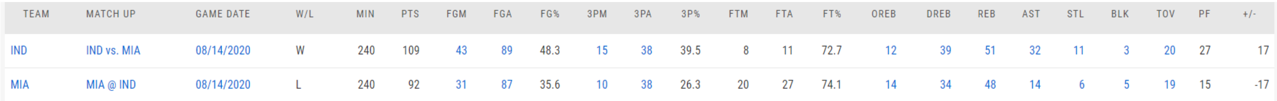
National Basketball Association (NBA) is the men's professional basketball league in North America. The game has fans all around the world and because of its popularity and valuation, a lot of studies are going on to predict match results, player comparisons, etc. NBA has managed to record some of the finest data in the sports industry, covering almost all the attributes affecting a basketball game and hence, making it a hotspot among machine learning enthusiasts to get hands-on experience. Though the data is very detailed, it is still a very complex task to analyze the data and predict games on the basis of it.

The objective of this project is to collect raw data from the official NBA statistics website [1] using some web scraping techniques, engineer and form relevant features which may play an important role in predicting a basketball match. Then, these features are used to train different machine learning models with the aim of maximizing the accuracy score of prediction of win/loss result of a match. This project also pays special attention to the formation of features based on the parameters of playing as home team or as a visitor team and separate features dealing with the recent history concerned with the last 8 matches and all-time history of the players and teams. Furthermore, the project can be seen as a comparative study among three different models trained on a separate set of domain-specific features focusing on 1) team performance, 2) individual player performance and 3) combination of the above two.

2. Related Works

There has been a lot of research for prediction of NBA matches using machine learning models. The major inspiration for our work was the methodology proposed by Renato Amorim Torres as a part of their research. Our work tried to explore more of the features without restricting us only to very intuitive features like the win percentage of a particular team. However, the features related to the last 8 matches were inspired by Renato's work. We chose the number 8 and not anything else, since they tried all numbers between 1-10 and 8 gave their model the best results.

Another Linear Regression algorithm to predict the winner was used in a paper by M.Beckler, H.Wang, 33 and M.Papamichael NBA Oracle (Beckler, Wang, Papamichael,



TEAM	MATCH UP	GAME DATE	W/L	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TOV	PF	+/-
IND	IND vs. MIA	08/14/2020	W	240	109	43	89	48.3	15	38	39.5	8	11	72.7	12	39	51	32	11	3	20	27	17
MIA	MIA @ IND	08/14/2020	L	240	92	31	87	35.6	10	38	26.3	20	27	74.1	14	34	48	14	6	5	19	15	-17

Figure 1. Example of a Team-wise statistics for a Single Match

2008). Using previous games, they achieved a result of 73% accuracy, which was the best result found.

3. Dataset

All types of data ranging from the basic game box score to a detailed game analysis including each player's contribution are available online. The authenticity of the website, quality of data, the ease of extracting data, and the number of seasons of data available were the major factors considered while choosing the website to extract data from.

The website NBA Official [1] was chosen as the source of our raw data since it is the official NBA website and presented all the relevant features. The dataset for a single match covering the statistics of the two participating teams is shown in Figure 1.

3.1. Web Scraping

After a detailed analysis of the data available on the website, we finalised to scrape the data consisting of team's match wise statistics and player's match wise statistics starting from the NBA season of 2008-09 to 2019-20. The website's developer kit showed that all the required tables and stats were dynamically loaded using javascript and hence, could not be scraped using simple static web scraping scripts.

To extract the data, a python script was written using selenium and the tables were scraped directly into different .csv files. The website [2] provided official box scores of each team in the team's match wise statistics. The complete 12 year data is stored into a single .csv file. For the player data, the dataset was huge and couldn't be accommodated in a single file. So, box scores of players from players' match wise statistics were stored separately for each season.

3.2. Feature Engineering

It was really important to spend some time in deciding the relevant input vector to the model. The data which we extracted only had match wise results and we needed to find a way to relate and make features which captured team performance over a time period. Hence, the data was first sorted based on the match date from the earliest and data was converted to cumulative figures to support the performance over a time period and then averaged out later on by the number of matches played by that time till that time.

The NBA website described each match in form of two rows displaying stats for the two teams with one as the home team and the other as the visitor team. An example is shown below:-

Along with the present features, we also generated additional features which took into consideration only the last 8 games in order to depict the current forms of the teams and players. Some features seemed to be relevant based on the all-time history while others on the basis of current status which may be selected in the later stages before feeding them to the classifiers.

The features for both the teams were concatenated in a single row using the tags 'H' for 'Home' and 'V' for 'Visitor' which can further be manipulated easily to generate a prediction for that match. The concatenated features represented the feature vector of a particular match.

We also scraped the performance of the players for each match and the dataset consisted of statistics of various players associated with a single team for a particular match. This data was added along with team features by converting all the players' performance into a single score, based on the weighted average with weights taken as the minutes played by the player in the match. Similar to the team data, the features were generated as Home and Visitor to support the two teams in a single row. Also, a similar technique of Last 8 was used with the player data to cover the recent forms of the player in a team.

Along with these, 4 special features were used. Home Team's Win Average when playing at Home Ground and Visitor Team's Win Average when playing at Away Ground were added as they capture the home ground advantage of the game. Last 8 of both these were also added to capture recent performance.

Finally, this gave us the main .csv file with 101 features and 14,305 matches (training examples) from the period of about 12 years. Later the features were dropped according to the domain of the model and informativeness of the features. All the features are briefly described in Table 1.

3.3. Data Cleaning

We encountered various instances of missing values in our dataset for some features. Considering the size of the dataset we had, we decided to replace the missing values with the average value for that feature for that position. This was done to minimize the deviation and reduce the number of unwanted outliers in the dataset.

The features for a particular match is based on the cumulation of features of the matches which have been played

Words in Features	Meaning
H	Refers to features related to home team
V	Refers to features related to visting team
Player	Weighted average of features related to players of a particular team
Last 8	Refers to weighted average of a particular feature for last 8 matches only
Match Up	Refers to the names of playing teams
W/L	Home team win status (1 meaning the home team won)
Win Avg	Ratio of the games won by the team vs the total number of games played by the team
FG%	Ratio of the summation of FG (Field Goal)% over all games played by the team vs the total number of games played by the team
3P%	Ratio of the summation of 3P(3 Point Field Goals)% over all games played by the team vs the total number of games played by the team
FT%	Ratio of the summation of FT(Free Throw)% over all games played by the team vs the total number of games played by the team
OREB	Average of OREB(Offensive rebound) over all games played by the team vs the total number of games played by the team
DREB	Average of DREB(Offensive rebound) over all games played by the team vs the total number of games played by the team
AST	Average of AST(Assists) over all games played by the team vs the total number of games played by the team
STL	Average of STL(Steals) over all games played by the team vs the total number of games played by the team
BLK	Average of BLK(Blocks) over all games played by the team vs the total number of games played by the team
TOV	Average of TOV(Turnovers) over all games played by the team vs the total number of games played by the team
PF	Average of PF(Personal Fouls) over all games played by the team vs the total number of games played by the team
Pts Diff Avg	Difference of points scored by a team with the points conceded by a team
H Win Avg At Home	Ratio of the games won by home team while playing at their home ground with the total number of games played at the home ground
V Win Avg On Visit	Ratio of the games won by visiting team while playing at away from their home ground with the total number of games played away from the home ground

Table 1. Explanation of Words Used to Represent Features

before for the two participating teams. For example, the result for a match between IND and MIA is to be predicted and it is the 10th match for IND while 11th for MIA, then the features for IND will deal with the first 9 matches while those for MIA will deal with the first 10 matches. Hence, the first match for any team is not predicted and not used as a training example. For the Last 8 features, if a team or player has played less than 8 games, then all the played games cumulative scores are used as features.

It was important to select only relevant features as some of the features were highly correlated and some didn't contribute positively to the prediction results. We started off with preparing a correlation matrix which looked like:

4. Methodology

After collecting the data, our next job was to design robust and useful machine learning or deep learning models to derive information from the data. The main aim of the project was to draw comparisons between three different models taking into account 1) only team statistics, 2) only player statistics and 3) combination of both team and player statistics. The methodology followed for all these three models is essentially the same. It involved creating a basic model to check the baseline metric score, feature selection, hyperparameter tuning and classifier selection.

4.1. Feature Selection

After the feature engineering part, we were left with 101 features consisting of many correlated ones which proved to be very less informative for training the model and hence, leading to poor performance. Also, some of the engineered features lacked relevance in determining the result of a basketball match and were needed to be dropped before feeding the data to the classifiers.

First step required for selecting features was to construct a correlation matrix(as shown in Figure 2) and analyse the same for highly correlation values. Also, the correlation values of all the features with the target variable i.e. ‘W/L’ is taken into account to measure the relative importance of features before dropping them.

Most of the features created came out to be redundant for the players statistics and team statistics which were required to be dropped. The attribute(player or team) better describing the given feature was taken and given more importance while considering for correlation. For example, the rebounds count seemed to be more relevant for players and hence, the features concerned with the team ‘DREB’ and ‘OREB’ were dropped. Another basis for feature selection included dropping the redundant features based on relevance with respect to the last 8 or all time history. Features like personal fouls or turnovers seemed to be more relevant based on the last few matches and hence, the other counterparts were dropped.

Lastly, some of the features which showed high correlation were kept due to their high significance and importance for determining the results of a match. Example can be seen in keeping both features of win avg and win avg at home. However, some of the less correlated features were dropped based on personal bias and intuitions which seemed to be of very less importance in relation with the classification problem.

All the final features selected in all the models have both Home and Visitor counterparts in order to enable the comparisons of the values by the classifiers used.

Features giving best performance for the player statistics only model are:- ‘Win Avg’, ‘Win Avg Last 8’, ‘Player Pts Diff Avg’, ‘Player FG%’, ‘Player FT%’,’Player OREB’, ’Player DREB’, ’Player AST’, ’Player STL’, ’Player BLK’, ’Player Pts Diff Avg Last 8’, ’Player FG% Last 8’, ’Player

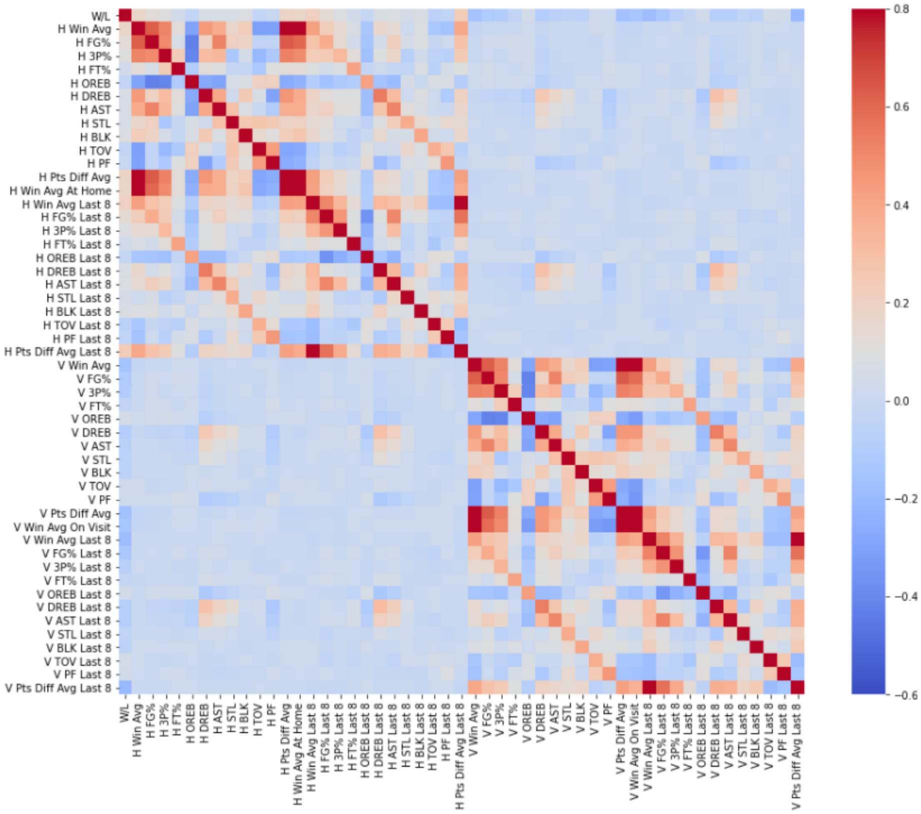


Figure 2. Correlation Matrix of Whole Dataset

3P% Last 8', 'Player FT% Last 8', 'Player OREB Last 8', 'Player DREB Last 8', 'Player BLK Last 8', 'Player TOV Last 8', 'Player PF Last 8'.

Features giving best performance for the team statistics only model are:- 'FG%', '3P%', 'FT%', 'OREB', 'DREB', 'AST', 'STL', 'BLK', 'Pts Diff Avg', 'Win Avg At Home', 'FG% Last 8', '3P% Last 8', 'DREB Last 8', 'AST Last 8', 'STL Last 8', 'BLK Last 8', 'TOV Last 8', 'PF Last 8', 'Pts Diff Avg Last 8'.

Features giving best performance for the combined statistics model are:- 'H Win Avg', 'H FG%', 'H FT%', 'H AST', 'H STL', 'H BLK', 'Pts Diff Avg', 'Win Avg At Home', 'Win Avg Last 8', 'FG% Last 8', '3P% Last 8', 'FT% Last 8', 'BLK Last 8', 'TOV Last 8', 'Pts Diff Avg Last 8', 'Player OREB', 'Player OREB Last 8', 'Player DREB Last 8', 'Player PF Last 8'.

4.2. Classifier Selection

Model Selection dealt with trying on various classifiers and testing them on the basis of the accuracy score keeping the features to be constant. The classifiers which outshined among the others were:-
Neural Networks, Random Forest Classifiers, Support Vector Classifiers(Linear Kernel)

The next step was to fine tune all the models to get the best results for our features based on the chosen metrics. However, the library defined hyperparameter search algorithms for these models were either not found or seemed to be a bit too complicated and time consuming and hence, we decided to devise our own algorithms for the same.

For Neural networks, we started with a bias of using a 3-layered model as optimal for our dataset and ran nested loops for determining the layer sizes. To further accelerate the search, we used just 4,8 or 16 to be the size of the first layer and only even numbers in the range (2, 32) for the subsequent layers. SGD optimiser performed better than adam optimiser for all 3 data types. The best neural network layer sizes for different feature types came out as below:-

1. Only team stats - (8, 12, 10)
2. Only player stats - (16, 24, 4)
3. Combined stats - (16, 10, 30)

For Random Forest Classifier, the hyperparameter tuning was done for a number of estimators and max depth with similar idea of running nested for loops as a method of searching. The range for the number of estimators was taken from 100 to 600 with a jump of 100 and the range for max depth to be from 5 to 30. The best hyperparameters for different feature type came out as below:-

Data Type	Neural Network	Random Forest	SVM
Only Player Stats	66.27	66.38	65.8
Only Team Stats	66.52	66.66	66.27
Combined Stats	**66.83**	66.13	66.55

a. Accuracy Comparison (in %)

Data Type	Neural Network	Random Forest	SVM
Only Player Stats	11.65	11.61	11.79
Only Team Stats	11.57	11.52	11.65
Combined Stats	**11.46**	11.70	11.55

b. Binary Cross-Entropy Comparison

Table 2. Metric Comparison of Various Model on Different Data Type

1. Only team stats - Estimators: 400, Max depth=12
2. Only player stats - Estimators: 200, Max depth=13
3. Combined stats - Estimators: 400, Max depth=6

For Support Vector Classifier(Linear Kernel), the hyperparameter tuning was required for just determining the value of 'C' (regularization parameter). After running the for loop, not much change was noticed in the accuracy scores of the 3 models and the same value of 0.01 was taken for all the models.

5. Results

5.1. Choice of Metric

The dataset we used showed a balance of target labels with around an equal percentage of the wins and losses of the team. This eliminated the use of the metrics like precision, recall and F1-score. Precision score is used when more surety is required of a positive result while recall is used to capture as many positive results as possible. As we don't have any inclination for the target, these scores were not used.

The metrics used in the projects are accuracy score and binary cross-entropy which suited best for our classification problem. Accuracy scores proved to be relevant due to almost equal shares of target labels in the dataset and hence, increased prediction of one value will affect the accuracy in a negative way. Our project focused on classifying an instance in two categories namely "Win" or "Loss" and hence, binary cross-entropy suited as a metric. It takes into account the uncertainty of our prediction based on how much it varies from the actual label.

5.2. Metric Comparison

Following results were obtained on feeding different data types:-

1. Only Player Features: Features only related to player performances based on weighted averages were considered for this. The accuracy score for this model was 66.38% and the binary cross-entropy was 11.61. This was achieved using the Random Forest Classifier.

2. Only Team Features: Features only related to team stats were considered for this. The accuracy score for this model was 66.66% and the binary cross-entropy was 11.52. This was achieved using the Random Forest Classifier.

3. Combined Features: All the features, consisting of team data as well as player data were considered for this. The accuracy score for this model was 66.83% and the binary cross-entropy was 11.46. This was achieved using Neural Networks.

The best results are obtained on feeding both player and team features to Neural Networks. On feeding player and team features separately, team features gave better results.

Detailed Comparison of metrics can be found in Table 2.

6. Conclusion

The results showed better accuracy for the neural-network model using both the team and player statistics as features as compared to accuracy of only team statistics and only player statistics. This shows that the combined data of teams and players is better than the individual data as it gives more in-depth information about the game that will be played. In the later two, team features are able to give better results compared to player features. The reason for this can be attributed to the fact that the result of a game is more related to team statistics than individual player statistics.

Metric comparison on binary-cross entropy showed that even with a notable difference in accuracy score for the three models, the binary-cross entropy did not change much. This meant that all the three forms of features captured almost similar amounts of information with combined features providing some finer level of granularity for classification.

All the classifiers seemed to show comparable results for the 3 types of datasets fed. Neural Network Classifier and Random Forest Classifier outperformed the other classifiers like Logistic Regression and Support Vector Classifier.

The NBA game predictor developed in this report can be used to simulate season matches and predict the most probable winning team. Other than this, we can use the relation that we got from feature engineering between the winning probability of a team and player statistics to gain insights and use this information to generate customised training for different players.

Further improvements in this predictor can be to collect more data so as to improve the performance. Also, instead of using average historical data of a team, we can use some kind of weighted average technique that gives more weightage to recent matches and less to very old matches.

7. References

[1] NBA Official Website
[2] Player Box Scores, nba.com
[3] Team Box Scores, nba.com
[4] Prediction of NBA games based on Machine Learning Methods, University of Wisconsin Madison, 2013
[5] Various Machine Learning Approaches to Predicting NBA Score Margins, Stanford, 2016
[6] MLP Classifier, sklearn
[7] Metrics for Binary Classification, Jakub Czakon, Neptune.ai
[8] Metrics And Scoring, Scikit-learn.org, 2020
[9] Code available at github.com/swapnil-ahlawat/NBA_Game_Predictor