Predicting the Outcome of Professional Basketball Game

Duo Jiang, Kuanlin Liu, Guojin Tang, Zichang Ye

December 2019

1 Introduction

The sports of basketball fascinates us for its unpredictability, but whenever we look retrospectively, the result almost always makes sense. We can explain a victory by many factors: better star players, better team chemistry, better coaching, better momentum, and the list goes on and on. With the belief that the result of a basketball game is not entirely random, we landed this project to predict the result of the game based on the previous records of the participating opponents.

2 Business Understanding

Two businesses will be benefited from a reliable prediction of a given game: the betting business and the merchandising and licensing business. In the betting business, the company sets the odds based on their belief and calculation about the odds of the occurrence of certain events. Conversely, a bettor can tailor his/her betting strategy to beat the betting company using such a model's prediction, as illustrated by a similar study in tennis (Sikpo, 2016). Secondly, since the sales of the licensed products are likely to be correlated with the performance of the team, merchandising and licensing businesses need an estimation of the team's success to decide the resources and budget required for production. A reliable prediction can help them control their budgets.

This prediction task can be thought as a classification problem. The target variable is whether the home team wins this game, and the features are previous information about the competing two teams, such as their up-to-date ability, their previous records of competing, as well as their level of fatigue and momentum. The detailed descriptions of the features can be found in the section of data preparation.

3 Related Work

In the process of looking for inspirations, we found three papers that are doing similar things and are achieving excellent results. The first paper [9] and the second paper [10] apply models like Linear Regression, Logistic

Regression, Naive Bayes, SVM, Decision Tree, and Artificial Neural Networks. They are in the traditional models domain and the second paper can achieve an accuracy of 0.73, which is better than 0.68 in the first paper. The third paper[11] creates a new model called NBAME based on the maximum entropy principle. This model achieves an accuracy of 0.744 percent, which is the best we have found online.

4 Data Understanding

4.1 Data Source

There was no comprehensive dataset that includes all the information we need for this project, so we made the best efforts to scrape data from the internet and then merged these datasets to create our dataset. The three data sources are basketball-reference, stats.nba.com and ESPN's NBA website.

From basketball-reference, we got the schedule for all games from 2008 to 2019, and we call it Total-schedule. Total-schedule, including includes games from regular seasons and playoff seasons, contains 15625 games in total. The features in Total-schedule are the two opponents(home team and away team), game date, and final points for the two teams. Also, from basketball-reference, a player-level dataset, player-details, was obtained. Player-details tab has 34 statistics for each player in each game, such as minutes played, points, and rebounds.

We believe team chemistry is a key factor in determining the result of a game, and lineup-level statistics may capture it. Therefore, we scraped stats.nba.com to get lineup-level statistics. For each game and each lineup, 15 different statistics were collected. These statistics include basic features like minutes played as well as more advanced features like PIE(which is a weighted sum of some other basic features).

The last data source is ESPN's NBA website. From ESPN, we obtained a statistics called Real Plus and Minus(RPM) which can also be divided into Offensive Real Plus and Minus(ORPM) and Defensive Real Plus and Minus(DRPM). As a player-level statistics, ORPM and DPRM can quantify a player's contribution on the offensive end and defensive end, respectively.

Note that the home team wins about 59.1% of all games in this dataset, which serves a proper baseline for model evaluation.

4.2 Selection Bias

NBA rules have changed significantly since the start of league, leading to a great change in playing styles in the NBA. Since the data we collected are from 2008 to 2019, our model may achieve much worse results when trying to predict the outcomes of games a long time ago, like in the 20th century.

4.3 Data Leakage

Attention is paid to avoid data leakage in our models since most datasets describing games, whether lineup-level or player-level, contain the statistics and results for the same game in the same row. Obviously, to predict game results, statistics in the same game cannot be used as predictors. Therefore, we carefully calculated predictors based on only statistics in the past. Another similar problem concerns the split of training data set and testing data set. To mimic the real environment of deployment, we split the dataset so that all the games in the training dataset happened before the games in the testing dataset. To be more specific, we used the games in the previous three seasons as a training dataset to predict the results in the next season. For example, we used data from 08-09, 09-10, and 10-11 seasons to predict the results in the 11-12 season. To continue this logic in cross-validation, we employed a strategy called Blocking Time Series Split[8] which will be discussed in detail in the evaluation section.

5 Data Preparation

5.1 Data Preprocessing

The goal is to create a merged table in which each data instance is game, and the target variable is a binary variable: 1 if the home team wins and 0 otherwise. For each data instance, predictors can be divided into predictors for the home team and predictors for an away team. This means that all predictors should be team-level, and we needed to convert our player-level and lineup-level predictors to team-level predictors.

In basketball domain knowledge, there are three factors that cannot be ignored in determining the result of an NBA game: 1. Absolute abilities of both teams. 2. Relative advantages of both sides, i.e., whether team A is particularly good at beating team B compared with other teams. 3. The recent momentum or the energy level of both teams. Our strategy is to design team-level proxies for these three factors using the dataset we collected and then combine the three factors.

For the absolute abilities of both teams, we tried to design two proxies: one using RPM mentioned above from ESPN and the other using the winning percentage before that game for both teams. To construct a predictor from RPM, we first found the player list for each team that season and then summed up all players' RPM, weighted by average minutes each player played last season. In this way, we succeeded in constructing team-level RPM from player-level RPM. Supposing there are n players in the team, the formula is presented below:

$$w_i = \frac{t_i}{\sum_{i=1}^{n} t_i}$$

$$\text{Team}RPM = \sum_{i=1}^{n} w_i RPM_i$$

where t_i is the average time player i played per game in the last season and RPM_i is RPM for i-th player last season. Winning percentage can be calculated using Total-schedule which contains all games from 2008 to 2019.

Relative advantages of both teams are constructed from lineup-level statistics collected from stats.nba.com. For each game and its corresponding two opponents, we found all the games in which the two teams played against each other last season and this season so far and calculated 13 predictors in total for each team. Since the 13 predictors are computed using the same logic, only one of them, OFFRTG(a lineup's points gained scored per 100 possessions), would be demonstrated here. The team-level OFFRTG is computed by summing up OFFRTG of all lineups, weighted by the time each lineup played:

$$w_i = \frac{t_i}{\sum_{i=1}^{n} t_i}$$
$$\text{Team}OFFRTG = \sum_{i=1}^{n} w_i OFFRTG_i$$

where t_i is the time lineup i played in the previous games, and OFFRTG_i is OFFRTG for the ith lineup.

To represent recent momentum of both teams, we constructed 35 proxies for each team: the first 34 predictors evaluating last five games' average performance using player-level statistics data from basketball-reference and the last one indicating whether the two teams had another game the day before, namely a back-to-back game. The first 34 predictors were constructed using the same logic, so only one of them, points obtained, would be mentioned here as an example to demonstrate the process. For the home team in the game, we present the team-level feature by computing the weighted average of player's data. We use the reciprocal of playing time as the weight so that the function helps weaken an inefficient player's performance even if he plays a number of minutes. However, ignoring the ability of the opponent seems irrational when we are estimating the team's recent performance. Thus, we compute the difference of the weighted average feature between the matchup. To illustrate, suppose there are n players in the home team, with i-th player had p_i points in a_i minutes, and m players in the opponent team,with i-th player had q_i points in b_i minutes in the game before:

$$\text{Home Team pts} = \frac{\sum_{i=1}^{n} \frac{p_i}{a_i}}{\sum_{i=1}^{n} \frac{1}{a_i}} - \frac{\sum_{i=1}^{m} \frac{q_i}{b_i}}{\sum_{i=1}^{n} \frac{1}{b_i}}$$

All first 34 predictors were constructed the same way. The last predictor indicating whether it is a back-to-back game was calculated using Total-schedule. Therefore, to represent energy level, we have 35 predictors for home team so 70 in total for both home and away teams.

5.2 Feature Selection

Because we used the previous three seasons to predict the following season, we applied feature selection separately on each training dataset, namely every three seasons. In total, we have eight train-test splits, which means that we need at least eight different sets of features. Also, since feature selection is crucial in determining the result, we chose to design two methods to conduct feature selection: one is first to drop highly correlated features and then use the random forest to select the most 15 important features; the other is to use mutual information. By making feature selection in such two ways, we ended up in 16 sets of features for eight train-test splits. Now, we will call the first method using correlation and random forest as the feature selection strategy one and mutual information as the feature selection strategy two.

The correlations between a set of features is in figure 1 and the result of analyzing feature importance using strategy two is in figure 2.

Figure 1: Feature correlations

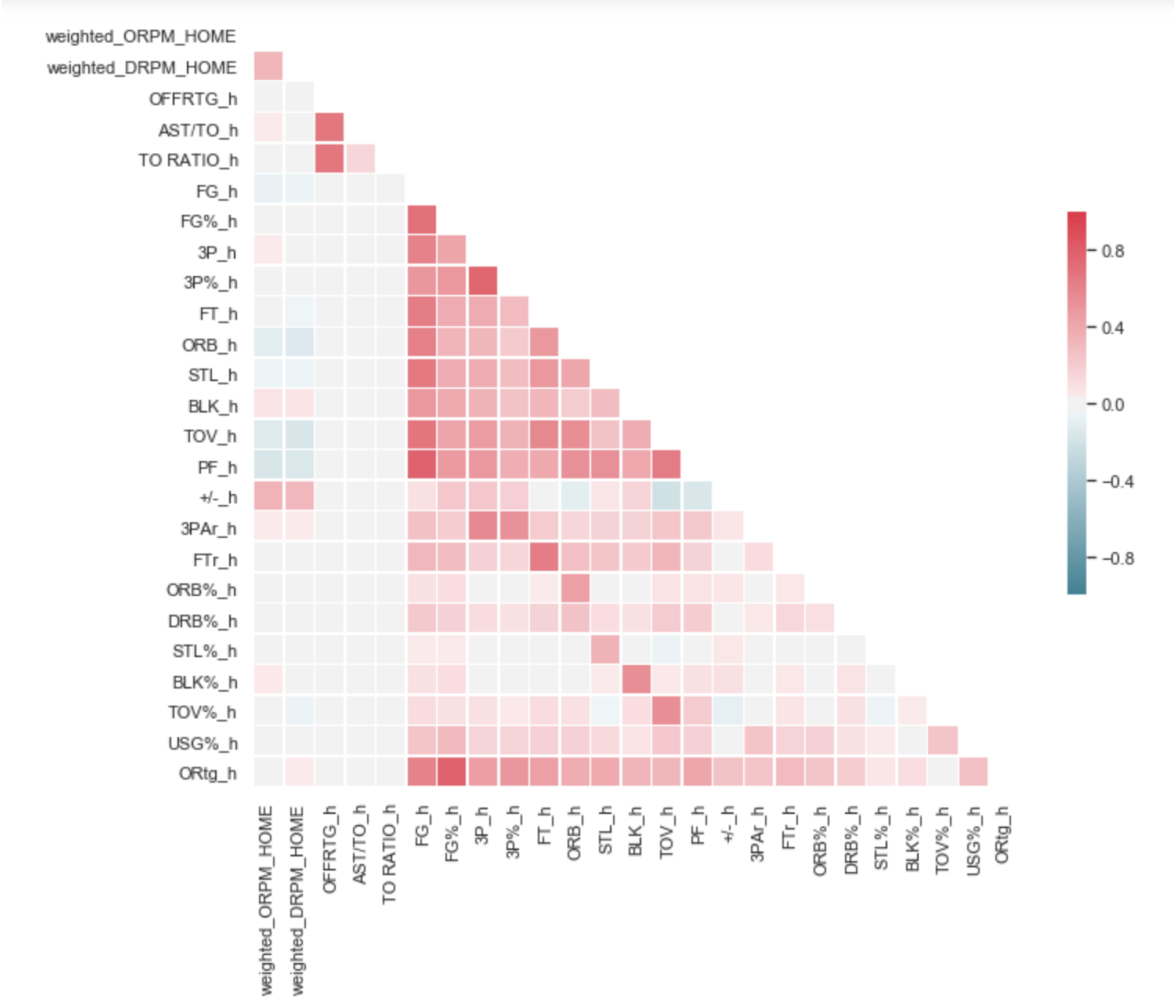


Figure 2: Feature Importance with Mutual Information

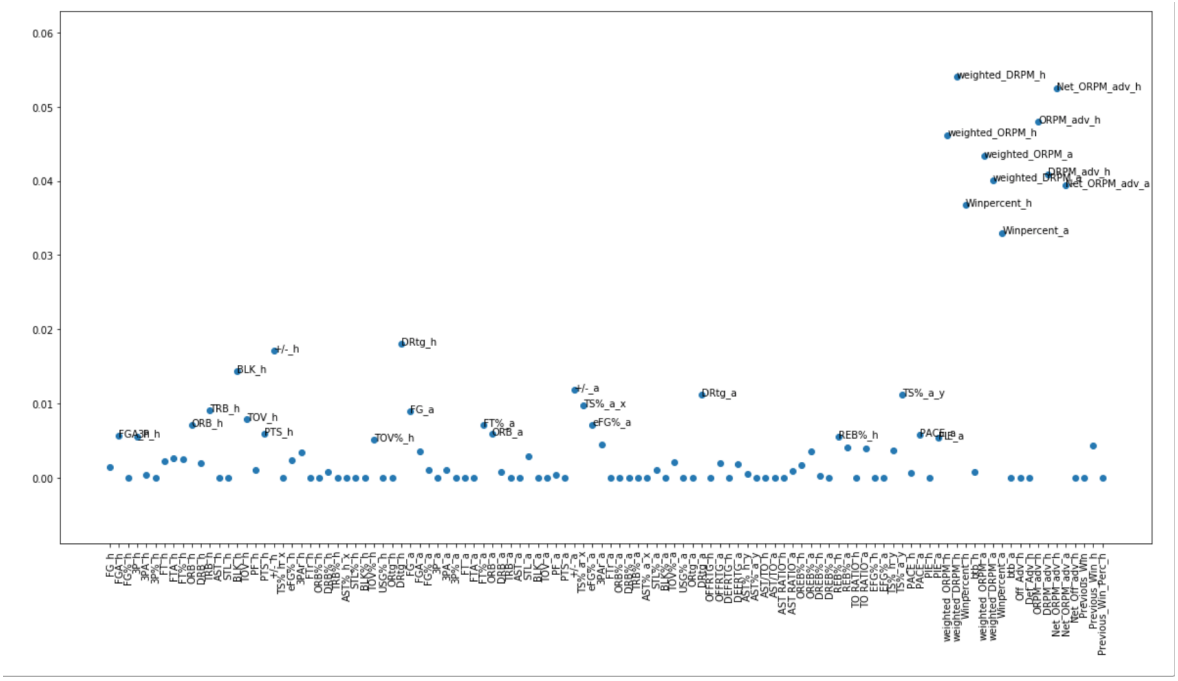
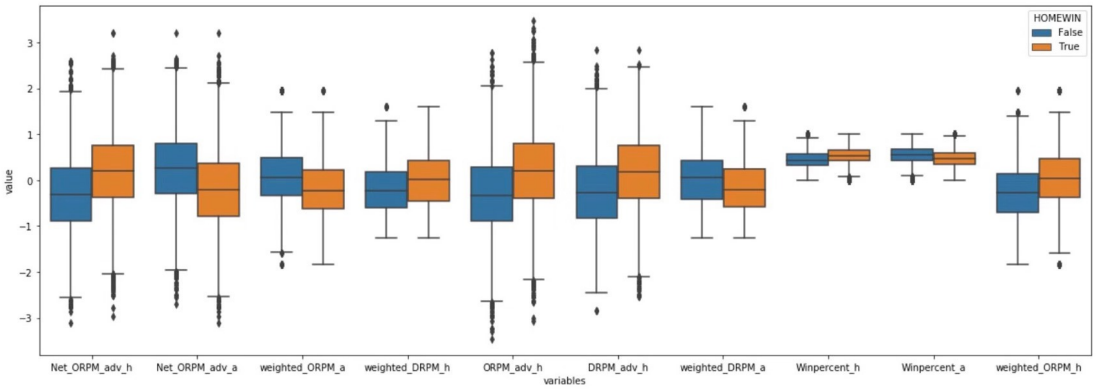


Figure 3: Boxplot for important features selected by Mutual Information



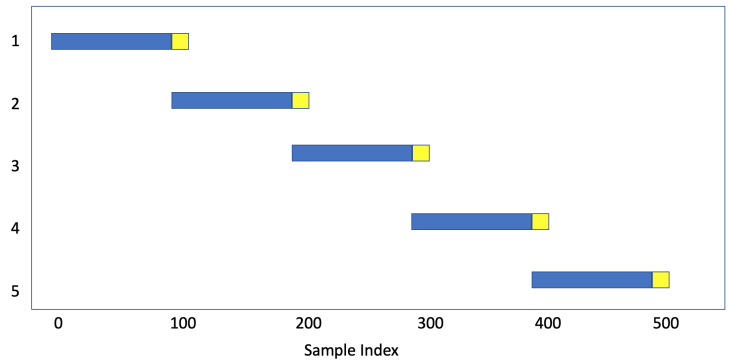
The family of RPM metrics, a measure of the *Abusolute Ability* of the home team, stands out, followed by a series series of features measuring their relative advantage over their opponent. To contrast, energy level doesn't seem to have much relevancy with the winning of the home team.

6 Modeling and Evaluation

6.1 Blocking Time Series Split

To avoid potential problems of data leakage, traditional Cross Validation does not apply here due to its randomness. Instead, we used Blocking Time Series Split[8] written by staff in Packt as a strategy to tune the hyperparameters. Different from k-fold Cross Validation, Blocking Time Series Split uses a smaller set of data as test data and a fixed size of data before testing data as training data. The strategy can be seen by looking at the visualization Figure 1. The folds used for training are in blue and folds used for validation are in yellow.

Figure 4: Blocking Time Series Split



We used logistic regression without any hyperparameter-tunning as the baseline model, and used Random Forest, K-Nearest Neighbors, Logistic Regression, and Support Vector Machine as our models.

6.2 Decision Trees

We attempted different depths of the trees, the minimum number required to split a node, and the minimum number of samples in a leaf. Decision trees turn out to be very poor in this problem, acquiring a mean AUC of 0.611.

6.3 Random Forest

With Random Forest, the hyperparameters we tuned were the number of trees, number of features in each tree, max depth of the tree, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node and whether bootstrap is used. With feature selection strategy one, Random Forest was able to achieve an average AUC 0.692 for all seasons, and the highest is 0.718 in 12-13 season. We did not run Random Forest with feature selection strategy two, because there are 32 (4 times 8) different dataset to train on, let alone the potential combination of the hyper-parameters.

6.4 K-Nearest Neighbors

There were not many hyperparameters in determining the result of K-Nearest Neighbors and the only one we tuned was the number of neighbors used. With feature selection strategy one, the average AUC achieved was 0.676, and the highest AUC was 0.707 when predicting 16-17 season.

6.5 Logistic Regression

For Logistic Regression, we attempted to tune the penalty, l1-ratio, as well as the weights of penalty. With feature selection strategy one, Logistic Regression can achieve an AUC of 0.708 on average, and the highest AUC was 0.733 when predicting the 15-16 season. Using feature selection strategy two, the best model uses the first 20 most relevant features, in the sense of mutual information, yielding an average AUC of 0.701 across different years.

6.6 Support Vector Machine

We tuned the weights on penalties and kernel in Support Vector Machine. Using feature selection strategy one, the model can achieve an AUC of 0.708, and the highest AUC was 0.733 when predicting the 15-16 season. Using feature selection strategy two, the best model uses the first ten most relevant features, yielding an average AUC of 0.701 across different years.

6.7 Evaluation Metrics

Accuracy and AUC scores are used as evaluation metrics. The accuracy and AUC of the best-performing models on multiple years of testing sets are shown in the following table.

Model	Feature Selection	2011	2012	2013	2014	2015	2016	2017	2018	Mean
Logistic Regression	Strategy One	0.699	0.731	0.702	0.728	0.733	0.681	0.693	0.698	**0.708**
SVM	Strategy One	0.698	0.731	0.700	0.729	0.733	0.680	0.694	0.696	**0.708**
KNN	Strategy One	0.678	0.668	0.682	0.702	0.706	0.677	0.670	0.622	0.676
Random Forest	Strategy One	0.697	0.718	0.684	0.697	0.703	0.684	0.685	0.670	0.692
Logistic Regression	Strategy Two	0.702	0.729	0.698	0.725	0.728	0.678	0.693	0.699	**0.706**
SVM	Strategy Two	0.702	0.730	0.681	0.698	0.728	0.674	0.694	0.698	0.701
KNN	Strategy Two	0.679	0.691	0.660	0.678	0.683	0.669	0.670	0.673	0.675
Decision Tree	Strategy Two	0.625	0.583	0.587	0.635	0.608	0.617	0.617	0.617	0.611
Logistic Regression	None	0.705	0.709	0.699	0.722	0.730	0.676	0.697	0.699]	**0.705**
SVM	None	0.703	0.726	0.683	0.716	0.727	0.682	0.697	0.682	0.702
KNN	None	0.682	0.671	0.603	0.632	0.686	0.664	0.670	0.668	0.659
Random Forest	None	0.669	0.693	0.658	0.692	0.702	0.675	0.654	0.645	0.674
Decision Tree	None	0.527	0.461	0.565	0.550	0.561	0.57	0.482	0.539	0.532

Table 1: AUC on Different Test Year

6.8 Results

According to the Table 1, AUC scores have been computed from the machine learning models within two feature selection methods. By doing so, not only can we compare the performance of models but also we are able to choose a better feature selection technique. Since we are dealing with a time series problem, the original dataset has been separated by the blocking split technique mentioned above. Closely looking at the Fig. 5, we can see that there are two lines, SVM and Logistic Regression, having a similar trend and even overlapping to each other. Although the results of the models we have tried do not surpass the performance of our baseline Logistic Regression, there are still some patterns for the further work. All of the models obtain lower AUC scores in 2013 and 2016. The patterns could be caused by that we have different feature sets for the blocking-split subsets, and those features in 2013 and 2016 may not be as useful as the feature sets in the other years after implementing our feature selection methods.

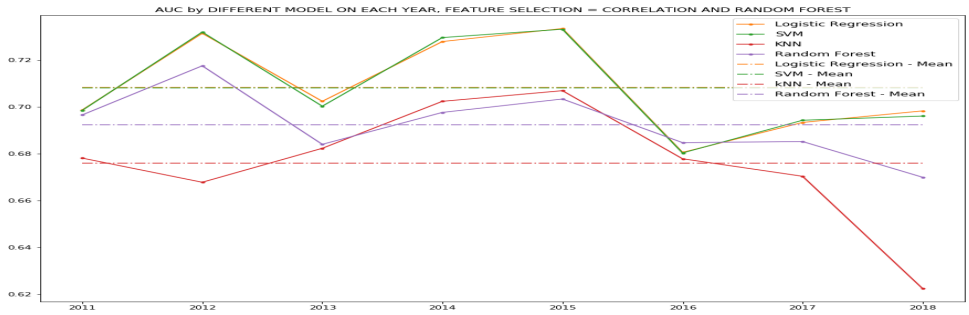


Figure 5: AUC of Models, Feautre Selection = 1

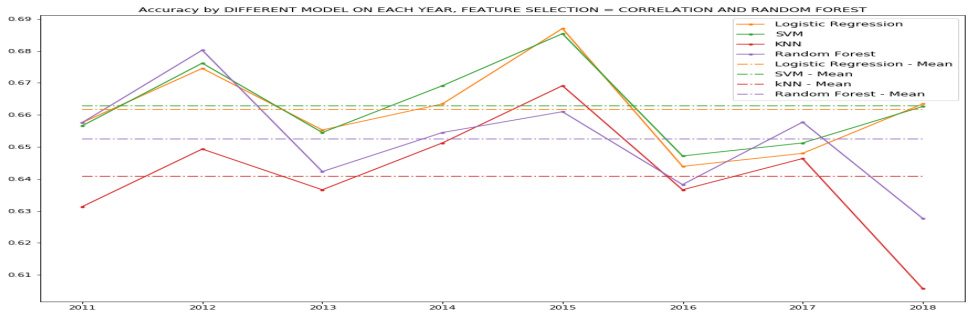


Figure 6: Accuracies of Models, Feautre Selection = 1

7 Deployment

The model can be deployed in the betting industry and the merchandising and licensing industry.

In the betting industry, both the hard prediction of the outcome and the soft prediction of the winning probability of one time can be used to calculate the betting odds. The profit can evaluate the model, comparing to the company's previous strategy to predict the outcome of the game. In specific, the companies can predict different games using this model and competing strategies, and compare the revenues they generate. The deployment of the model can be continuously monitored as the season go on, and the betting companies can check whether such a model can stably generate revenues.

The soft and hard predictions of the models can similarly be used in the merchandising industry to decide the production of the licensed product. There are two evaluation strategies. First is the changes in production costs, excess stocks, and profits after such a model is deployed. Because it takes time for the company to set its production goal, execute it, observed the reaction of the market, and readjust its production goal, the monitoring of the model will happen between the production cycles. The risk is the slow feedback that may cause the company too much money before the model is tuned and adjusted appropriately. As a result, the accuracy of the model should be

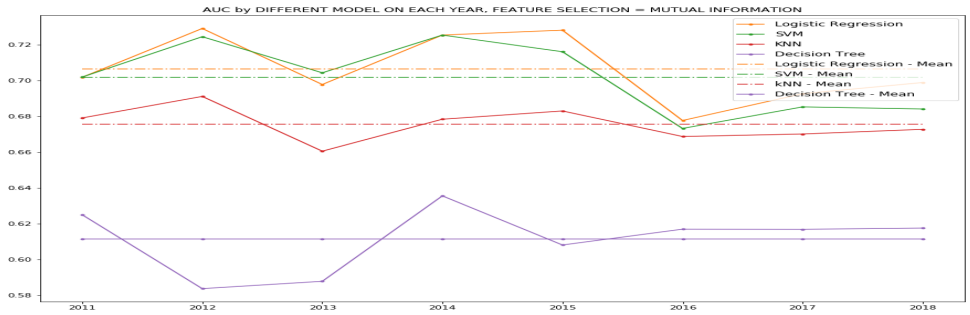


Figure 7: AUC of Models, Feautre Selection = 2

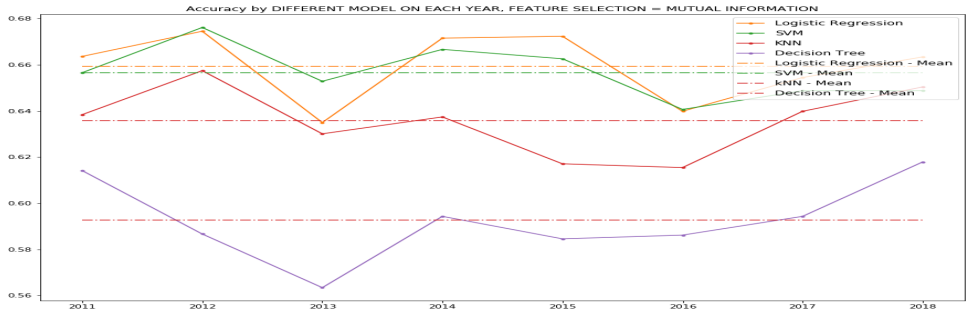


Figure 8: Accuracies of Models, Feautre Selection = 2

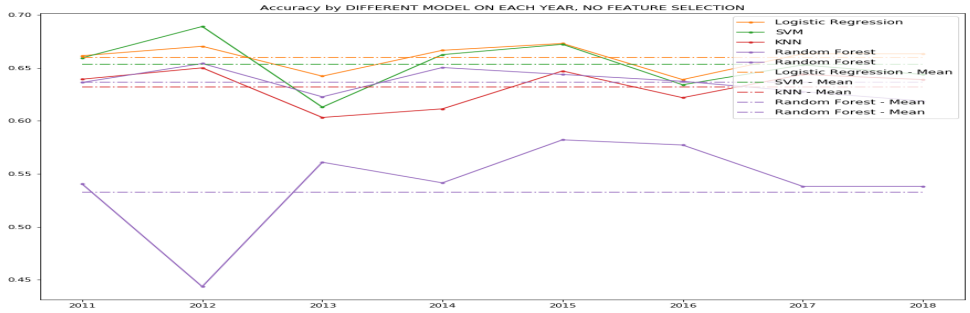


Figure 9: Accuracies of Models, No Feature Selection

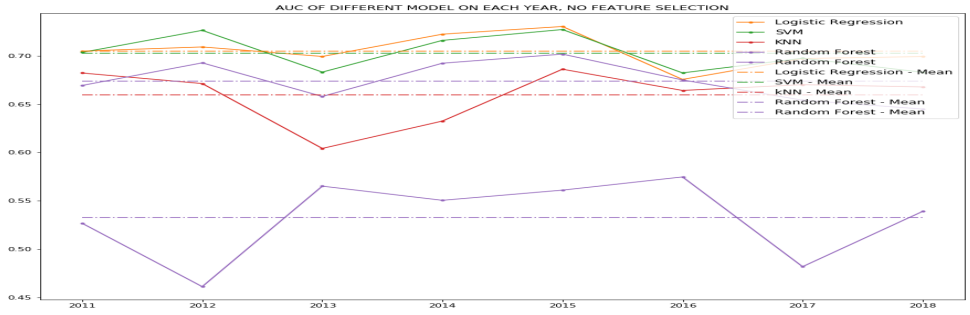


Figure 10: AUC of Models, No Feautre Selection

introduced as a fast evaluation of how reliable the model is as a factor in determining the production goals.

The business should be aware that the models rely entirely on previous records, and is not expected to perform well in a dramatically different environment due to reasons such as significant change of players in a team, unexpected injuries, and change of rules by the league.

8 Conclusion and Future Works

Our findings strengthen our initial claim: the game of basketball is not entirely random. Given the base rate of the home team winning a game is 0.60, our work joins other studies and claims that we can improve our prediction of game with machine learning techniques.

However, current work is still not satisfying enough to deploy in a real business setting: a production plan may not afford to rely on a prediction with 68% of accuracy. We hereby propose one strategy, or hypotheses, as a start of the future work.

- Given that the decision tree performs poorly, does that mean that the features are *too continuous* so that the decision tree can not separate them effectively? Binning on features can be conducted to validate this hypothesis.

9 Appendix

9.1 Contribution

- Duo Jiang: Data Preparation, Data Understanding, Dataset Normalization, Random Forests, kNN, Related Works, Feature Selection with Correlation and Random Forest, Blocking Time Series Split
- Kuan-lin Liu: Web Scraping, Data Collection from Basketball-Reference and NBA Stats, Machine Learning theories Introduction, Data Cleaning, Exploratory Data Analysis, Feature Selection, Scaling
- Guojin Tang: Feature Selection with Ridge and Lasso, Feature Visualization, Proofreading, Feature Selection with Mutual Information, Related works
- Zichang Ye: Introduction, Business Understanding, Deployment, Data Collection for Advantage Family,Web Scrapping, Feature Selection with Mutual Information, Logistic Regression and Support Vector Machine

9.2 Feature Description

FG	Field Goal
FGA	Field Goal Attempted
FG%	Field Goal Percentage
3P	3 Point
3PA	3 Point Field Goals Attempted
3P%	3 Point Field Goal Percentage
FT	Free Throws
FTA	Free Throws Attempted
FT%	Free Throw Percentage
ORB	Offensive Rebounds
DRB	Defensive Rebounds
TRB	Total Rebounds
AST	Assists
STL	Steals
BLK	Blocks
TOV	Turnovers
PF	Personal Fouls
PTS	Points
+/-	The point differential when a player or team is on the floor
TS%	True Shooting Percentage
eFG%	Effective Field Goal Percentage
FTr	Free Throws rate
ORB%	Offensive Rebounding Percentage (of the team, hereby the same)
DRB%	Defensive Rebounding Percentage
TRB%	Total Rebounding Percentage
AST%	Assists Percentage
STL%	Steals Percentage
BLK%	Blocks Percentage
TOV%	Turnover Percentage
USG%	Usage
OFFRtg	Offensive Rating
DFFRtg	Defensive Rating
AST/TO	Assists/Turnover ratio
TO RATIO	The number of turnovers a player or team averages per 100 possessions used
PACE	The number of possessions per 48 minutes for a team or player.
PIE	Player Impact Estimate

Table 2: Feature Description

References

[1] Jeff Racine,Consistent cross-validatory model-selection for dependent data: hv-block cross-validation, Journal of Econometrics, Volume 99, Issue 1, 2000, Pages 39-61, ISSN 0304-4076

[2] R.P. Bunker, F. Thabtah A Machine Learning Framework for Sport Result Prediction Appl. Comput, Informatics., 15 (2019), pp. 27-33

[3] Zimmermann, Albrecht Moorthy, Sruthi Shi, Zifan. (2013). Predicting college basketball match outcomes using machine learning techniques: some results and lessons learned.

[4] Loeffelholz, B., Bednar, E. Bauer, K. (2009). Predicting NBA Games Using Neural Networks. Journal of Quantitative Analysis in Sports, 5(1), pp. -. Retrieved 9 Dec. 2019, from doi:10.2202/1559-0410.1156

[5] D. Miljković, L. Gajić, A. Kovačević and Z. Konjović, "The use of data mining for basketball matches outcomes prediction," IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, 2010, pp. 309-312. doi: 10.1109/SISY.2010.5647440

[6] Kain, K. J., Logan, T. D. (2014). Are Sports Betting Markets Prediction Markets?: Evidence From a New Test. Journal of Sports Economics, 15(1), 45–63

[7] Hu, Feifang; Zidek, James V. Forecasting NBA basketball playoff outcomes using the weighted likelihood. A Festschrift for Herman Rubin, 385–395, Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2004. doi:10.1214/lnms/1196285406. https://projecteuclid.org/euclid.lnms/1196285406

[8] https://hub.packtpub.com/cross-validation-strategies-for-time-series-forecasting-tutorial/, 2019.

[9] Lee Richardson, Daren Wang, Chi Zhang, Xiaofeng Yu *NBA Predictions*.

[10] MatthewBeckler, HongfeiWang, MichaelPapamichael. *NBA Oracle*

[11] Ge Cheng, Zhenyu Zhang, Moses Ntanda Kyebambe, Nasser Kimbugwe. *Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle*. MDPI