CS6140 Machine Learning Final Project: Predicting NBA Games

Team Bento
Di Yi, 001826332
Yichen Wang, 001854571
Yuan Gao, 001202419

1 Introduction

NBA is one of the most popular sports game all over the world. Not only because it is highly competitive, but also due to its uncertainty. We often see the gambling company forecast the game odds before the game, that's why we choose the topic — we want to find a way to predict game result. Currently, ESPN has an average of 68.3% accuracy prediction, Bob Voulgaris, who is a professional gambler, has an average of around 70% accuracy. The goal of this project was to see if it was possible to predict game result based on just previous game results.

The problem is hard due to the uncertainty of the game. There are many factors affect one game. Before we work on this project, we think over a question, what will human consider when predict a game? We may consider team standing, like if Celtics rank 6th in Eastern league and Cavaliers rank 13th in Eastern league, we will choose Celtics to win. The other factors may include home/away status, straight wins in last several games and so on. So choosing features wisely is very important.

In this project, each team member is responsible for different tasks.

- Di retrieve raw data from stat.nba.com using web crawler and a python API nba-py. Processed raw data to clean data with 67 fields each game and build our data warehouse. Build baseline model with linear regression and SVM using 7 seasons data as train data and another one season data as test data.
- Yichen is responsible for the neural network configuration design, implementation and improvement.
- Yuan extracted main feature of our model. He also applied Random Forrest Classifier, SVM and Logistic Regression model.

2 Related work

There has been some project done on this problem, the goal of them is to predict the game winner. One of the paper is using Linear Regression algorithm to predict the winner was used in a paper by M.Beckler, H.Wang, 33 and M.Papamichael NBA Orcale (Beckler, Wang, Papamichael, 2008). The accuracy result is 70%, which is the best result ever. We also see a stanford student Jaak Uudmae juudmae working on this project, in his project, he used SVM Classifier, Neural Network, and Linear Regression to train the model, which received around 62% to 65% accuracy. The other paper Various Machine Learning Approaches To Predicting NBA Score Margins, which is done by G.Avalon, B.Balci, J.Guzman, Papamichael using Gaussian Discriminant, Random Forrest and Linear Regression, achieved arount 64% acuracy.

Most of the previous works consider the previous game winning rate, in our work, we add some other features including previous N game winning rate, home/away team winning rate at home/away.

3 Method/Model

3.1 Overview

For our baseline model, we simply use team vector and win-loss percentage as our feature, and the model we use is SVM and Linear Regression. After we extract more features, the four models we applied is Random Forrest Classifier, Logistic Regression, SVM and Neural Networks. The features we feed in to our model includes win-loss percentage of home/away teams, average point difference of home/away teams, home/away team win-loss percentage at home/away and also home/away teams win-loss percentage of previous 10 games.

3.2 Formal Definition

3.2.1 Linear Regression

Linear Regression fits a linear model with coefficients $w = (w_1, ..., w_n)$ to our observed data set of values. Its goal is to minimize the residual sum of squares.

$$argmin_w ||y - Xw||_F^2$$

3.2.2 Random Forrest Classifier

Random Forrest Classifier first select random samples from a given dataset, and construct a decision tree for each sample and get a prediction result from each decision tree. Then it perform a vote for each predicted result. Finally it select the prediction result with the most votes as the final prediction.

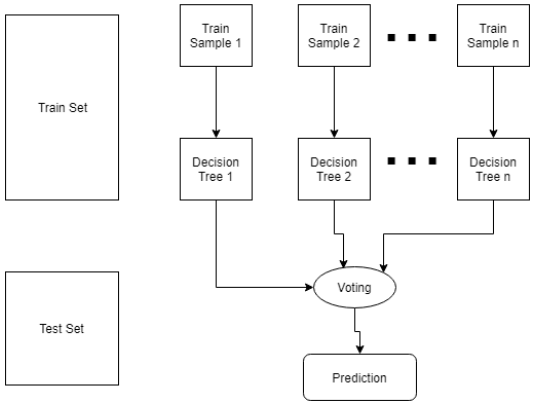

Figure 1: Random Forrest Classifier

3.2.3 Logistic Regression

Logistic Regression is to model a binary dependent variable. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Model Class:

$$f(\vec{x}|\vec{w}, b) = \sigma(\vec{w}^\top - b)$$

$$\sigma(a) = \frac{e^a}{1+e^a} = \frac{1}{1+e^{-a}}$$

The loss function of logistic regression is log loss. And the learning objective of it is

$$argmin(w, b) \sum_{i=1}^{N} L(y_i, f((x_i|w, b))$$

3.2.4 SVM

SVM is trying to find an hyperplane dividing the data into two categories. Using the full game data we can classify each game sample as a binary variable 1 (win) or 0 (loss).

$$argmin_{w,b,\xi} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

For $i : y_i(w^T x_i - b) \geq 1 - \xi$ and $\xi_i \geq 0$

Kernel method is applied here,

For linear kernel

$$K(x, x') = x^T x'$$

For sigmoid kernel

$$K(x, y) = tanh(\alpha x^T y = c)$$

For Gaussian kernel

$$K(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}, \sigma > 0$$

3.2.5 Neural Networks

A Sequential Neural Network model is a linear stack of layers. In Keras, this is done by passing a list of layer instance to the constructor. Stochastic Gradient Descent (SGD) is an approximation of gradient descent optimization, with the difference that samples in training set is selected randomly instead of as a single group as in standard gradient descent.(https://en.wikipedia.org/wiki/Stochastic_gradient_descent) As a result, SGD responds to effects from every sample.

$$W = W - \alpha \nabla L(W, b, x^{(i)}, y^{(i)})$$

where W is the weight of the model, α is the learning rate and x and y represents a single training record.

3.3 Features

3.3.1 Baseline Model

For our baseline model SVM and Linear Regression, we use team vectors and win-loss percentage as our feature.

1. First 30 features represent by 01 vectors are the 30 teams in the league. The only one 1 represent the home team.

2. Second 30 features represent by 01 vectors are the 30 teams in the league. The only one 1 represent the away team.

3. Home team win percentage

4. Away team win percentage

3.3.2 Improved Model

To find the best features, we consider different factors as following categories:

- A. Win-loss percentage of both teams in all previous games during the season
- B. Win-loss percentage of both teams in all previous N games during the season.
- C. Average point difference of both teams in all previous games of the season.
- D. Previous game result between two teams.
- E. Home team win-loss percentage at home, away team win-loss percentage as visitor.

Simply predicting by these factors, we can get a prediction result. For example, if before the game, Celtics has win-loss percentage higher then Spurs, we predict Celtics to win.

	A	B	C	D	E
2010	*0.65243902*	*0.61463415*	*0.65813008*	*0.58123249*	*0.65853659*
2011	*0.63080808*	*0.58484848*	*0.62626263*	*0.60554371*	*0.64343434*
2012	*0.64593496*	*0.61544715*	*0.65894309*	*0.59123055*	*0.64390244*
2013	*0.65243902*	*0.61585366*	*0.64227642*	*0.56795422*	*0.64186992*
2014	*0.66788618*	*0.61300813*	*0.67113821*	*0.58773181*	*0.64878049*
2015	*0.65813008*	*0.61300813*	*0.65691057*	*0.61951909*	*0.66178862*
2016	*0.60650407*	*0.59186992*	*0.62195122*	*0.55827338*	*0.61869919*
2017	*0.63495935*	*0.61219512*	*0.62845528*	*0.58981612*	*0.63943089*
Mean	***0.6436376***	***0.60760809***	***0.64550844***	***0.58766267***	***0.64455531***

Figure 2: Predicting accuracy by different factors

The factors except D has average accuracy of 63%, while the result of D is relatively low, so we make our final decision on 8 features(All before current game).

- Win-loss percentage of Home team.
- Win-loss percentage of Away team.
- Win-loss percentage of previous 10 games of Home team.
- Win-loss percentage of previous 10 games of away team.
- Home team win-loss percentage at home.
- Away team win-loss percentage as visitor.
- Average point difference of home team.
- Average point difference of away team.

After the preprocessing, we have the final valid features file like below. (Column 1-4 are team information, Column 5-12 are our features, Column 13 is result)

	1	2	3	4	5	6	7	8	9	10	11	12	13
487	18	103	3	88	0.4286	0.1667	0.2424	0.2727	-6.5758	-5.6667	0.2500	0.3750	1
488	27	101	21	74	0.9000	0.6667	0.8750	0.6765	8.2813	2.6176	0.8750	0.6250	1
489	17	99	7	87	0.5000	0.9167	0.4000	0.7742	-2.3000	5.1935	0.3750	0.6250	1
490	8	104	26	86	0.8125	0.1818	0.5806	0.2069	2.0323	-6.6897	0.5000	0.1250	1
491	29	98	15	92	0.6471	0.2941	0.6667	0.4375	2.8182	-0.1563	0.6250	0.5000	1
492	20	98	12	92	0.5625	0.3571	0.5758	0.4516	0.9091	-0.5484	0.3750	0.3750	1
493	13	98	1	107	0.3684	0.5000	0.3030	0.6000	-3.7273	1.8286	0.6250	0.6250	-1
494	28	79	2	93	0.4375	0.6875	0.3438	0.7742	-3.8750	7.8387	0.2500	0.6250	-1
495	6	95	7	104	0.3333	0.8462	0.2424	0.7500	-9.0606	4.6563	0.1250	0.6250	-1
496	25	100	11	85	0.7857	0.3529	0.5152	0.5000	0	1.5625	0.6250	0.7500	1
497	26	94	24	89	0.2222	0.3750	0.2000	0.4516	-7.0667	-1.9032	0.1250	0.3750	1
498	14	85	15	104	0.7333	0.2778	0.6970	0.4242	6.3030	-0.3333	0.6250	0.3750	-1
499	4	82	16	96	0.5000	0.7059	0.3548	0.7429	-4.8065	9.4286	0.3750	0.8750	-1
500	22	110	10	90	0.7059	0.2632	0.6176	0.3939	4.2353	-4.4242	0.6250	0.5000	1
501	2	96	18	93	0.8667	0.1053	0.7813	0.2647	8.0313	-5.9412	0.6250	0.3750	1
502	19	84	23	77	0.7647	0.2632	0.5882	0.3939	1.7941	-0.8182	0.5000	0.3750	1
503	8	113	11	106	0.8235	0.3333	0.5938	0.4848	2.5313	1.0606	0.5000	0.7500	1
504	29	102	9	97	0.6667	0.1765	0.6765	0.3333	2.9118	-4.4848	0.6250	0.5000	1
505	16	101	17	89	0.7778	0.3125	0.7500	0.4194	9.5556	-1.8387	0.8750	0.3750	1
506	20	128	27	115	0.5882	0.8333	0.5882	0.8788	1.0588	8.8485	0.3750	0.8750	1
507	5	111	28	91	0.8235	0.2500	0.6875	0.3333	5.3438	-4.1818	0.7500	0.2500	1
508	15	110	21	105	0.6000	0.6250	0.4412	0.6571	0.2353	1.7714	0.3750	0.5000	1

Figure 3: Final cleaned data and feature

Here, for the feature Win-loss percentage of previous 10 games of team, actually it is not independent with Win-loss percentage. However, after some test and discussion, we think that last 10 games is an important factor that affect a team. For example, if a team fails many game in the first half of the season, while it wins the last 9 games of 10, it may beat the team who ranks higher than it while lose more than it in the last 10 games.

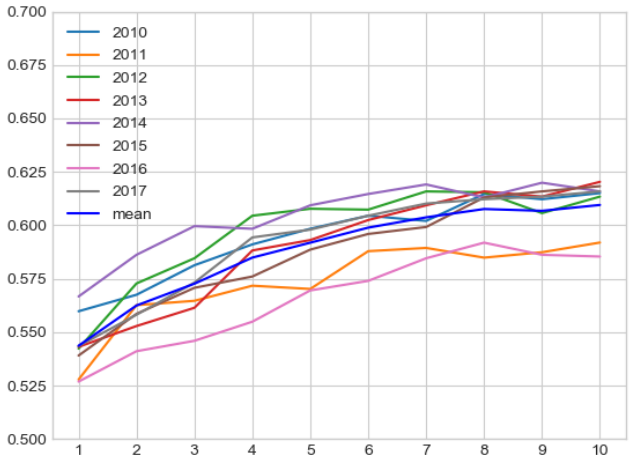

Figure 4: Prediction accuracy based on last N games

4 Experiments

4.1 Dataset

The data we used is from stats.nba.com. To scrape the data, we used a python library called nba_py(https://github.com/seemethere/nba_py).

We have 30 csv files of 30 teams, which include game stats of each team in that season. The data fields are as follows.

- TEAM_ID:(String) - An unique identifier for each team.
- GAME_ID:(String) - An unique identifier for each game.
- GAME_DATE:(String) - Represented by Month Day, Year.
- MATCHUP:(String) - A@B: A away, B home. AvsB: A home, B away.
- WL:(char) - Result of the game corresponding to the given team_id.
- W, L:(char) - Winning/Losing games after this game.
- W_PCT:(double) - Winning rate after this game.
- MIN:(int) - Game duration. Typically 240 min, may have extra time.
- FGM, FG3M, FTM:(int) - Field goal / Field goal 3 point / Free throw made.
- FGA, FG3A, FTA:(int) - Field goal / Field goal 3 point / Free throw attempt.
- FG_PCT(double) - Field goal / Field goal 3 point / Free throw made percentage.
- OREB/DREB:(int) - Offense/Defense rebounds.
- REB, AST, STL, BLK, TOV, PF:(int) - Rebound/Assistant/Steal/Block/Turnover/Personal Foul.
- PTS:(int) - Final points of given team_id.

The training data we use were the games of 30 teams in 6 seasons(2010-2017) which is about 10000 games. The test data was the game of 30 teams in 2017-2018 season which is 1230 games.

4.2 Baselines

In our baseline model, SVM is designed to predict win(1) or loss(0). For Linear Regression, although it is not a classifier model, in order to compare the performance of LR with other classifier model,

we first use LR to predict the net point of each game, the input features are the same with SVM, but the output will be the net point of each game which is the difference of home teams score and away teams score. Then based on the net point of each game, we could get the result of each game. In this way, we can compare the result of LR with other classifier model.

Method	Training Set	Test Set	Accuracy on Test Set
Linear Regression	2010 - 2016	2017	60.45%
SVM			61.39%

Figure 5: Prediction accuracy of baseline model

4.3 Method

4.3.1 Linear Regression

For our Linear Regression we use the sklearn library. Our baseline model uses linear regression, the accuracy is about 60%. Applying linear regression on our extracted features, it doesn't give us very good result, around 55%, so we tried to use Least Mean Square to get the weights, it turns our that it will not converge. Then we apply PCA on our features and reduce it to 3-dimensional, the weights converges. After that, we define our linear regression model as: if $Y > 0$ then home team will win, otherwise the away team will win. However, this method is not correct to some sense, even if the accuracy is not bad, around 64%, it's better to change the predicting result point difference and then predict again.

4.3.2 Logistic Regression

For logistic regression we also use the sklearn library, the final prediction accuracy is 66.64%.

4.3.3 SVM

Using SVM as a binary classification, we assume that 1 represents home team win, and 0 represents away team win. In SVM we tried three different kernel trick, Linear, Gaussian and Sigmoid. Among all this kernel, linear gives us the best result.

Kernel	Training Set	Test Set	Accuracy on Training Set	Accuracy on Test Set
Linear	2010 - 2016	2017	68.13%	67.91%
Gaussian			67.42%	66.08%
Sigmoid			57.52%	56.01%

Figure 6: Prediction accuracy of SVM using different kernel trick

4.3.4 Random Forrest Classification

We tried different estimator number and max depth of random forrest, the result is shown in figure7.

Md\Est	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
2	66.54% 61.94%	66.46% 61.56%	66.49 61.19%	66.49% 61.19%	66.52% 61.19%	66.52% 61.19%	66.50% 61.19%	66.50% 61.19%	66.49% 61.19%	66.48% 61.19%
3	67.11% 61.94%	67.11% 61.56%	67.05% 61.94%	67.11% 61.56%	67.11% 61.56%	67.14% 61.56%	67.10% 61.56%	67.14% 61.56%	67.15% 61.56%	67.10% 61.56%
4	67.93% 62.31%	**68.05%** **62.31%**	67.99% 62.31%	67.97% 62.31%	67.97% 62.31%	67.97% 62.31%	67.93% 62.31%	67.94% 62.31%	67.94% 62.31%	67.91% 62.31%
5	68.78% 60.82%	68.81% 60.44%	68.86% 60.82%	68.70% 61.19%	68.67% 61.19%	68.68% 60.82%	68.70% 61.19%	68.68% 61.19%	68.67% 61.19%	68.68% 61.19%

Figure 7: Prediction accuracy on train/test set of rfc(md: max depth, est: #estimator)

4.3.5 Neural Network

For the last part, we applied neural network to this problem as a comparison to the previous methods.

To implement neural networks, the framework we use is keras, which is a relatively popular toolkit for machine learning developers with broad range of machine learning algorithms integrated within.

Due to the complexity and limitation on the size of data, we choose to use sequential structure of network, which is relatively simple.

For basic parameters, we use relu as the activation function for each neuron within to introduce non-linearity. And for the update of model weights, we choose stochastic gradient descent algorithm as the optimizer. Finally at the output layer, we apply sigmoid function to return a 0/1 value to indicate the prediction result of the game. (Win or Lose)

We started with the simplest model sturcture with only one hidden layer. (Input layer: 16 units, Hidden layer: 8 units) And we reach 63% - 65% of accuracy, which is close to the results returned by most of the other relative work we found online. To improve it, we firstly tried increasing the complexity of the model by adding more layers to it, which indeed returns better result as more than 67% of accuracy on training set when 1 more layers added in. (Input layer: 16 units, Hidden layer-1: 12 Units, Hidden layer-2: 8 Units) But we soon realize the problem of overfitting due to the visible drop of accuracy on test set. To avoid overfitting, we apply Dropout(0.1) to each layer of the model, which selects out 10% of neurons to participate in the training for each data pass. This finally gives us a 66.91% of accuracy.

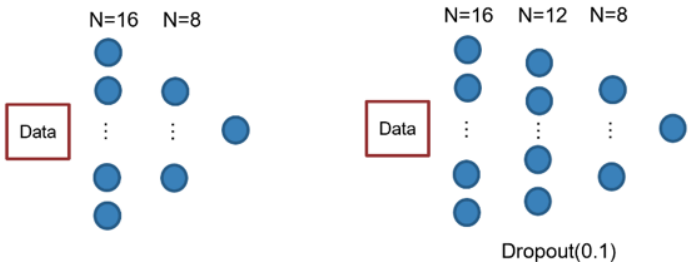

Figure 8: Neural Network Structure

Model Structure Adjustments (N1 as input layer)	Acc.
N1=16, N2=8	64.90%
N1=16, N2=12, N3=8	64.36%
N1=16, N2=12, N3=8, with dropout = 0.1	**67.73%**
N1=16, N2=12, N3=8, N4=4, with dropout = 0.1	67.00%
N1=16, N2=12, N3=8, N4=4, N5=4, with dropout = 0.1	58.34%

Figure 9: Neural Network Performance with different configuration

4.4 Results

The best result is given by SVM(linear), which is 67.91%, and Neural network ranks the second, which is 67.73%.

Method	Training Set	Test Set	Accuracy on Training Set	Accuracy on Test Set
Random Forrest Classifier	2010-2016	2017	67.93%	62.81%
Logistic Regression			67.11%	66.64%
SVM(linear)			68.13%	67.91%
Neural Network			69.03%	67.73%

Figure 10: Final prediction result

5 Conclusion

Our SVM model achieves 67.91% accuracy, which is the best model in our project. While neural network ranks the second, 67.73%, but we think with more features, it has the potential to beat SVM. Compared with the baseline we set previously, the improvement we made, which includes data selection and model design, has gained us a 8% of leap of accuracy. However, there are still many factors affecting a game maybe considered by expert but we have no data source, like team chemistry, player health condition, and emergency. All in all, our model is very close to expert and professional gambler's 70% accuracy.

References

[1] Jaak Uudmae juudmae. Predicting NBA Game Outcomes. Accessed from http://cs229.stanford.edu/proj2017/final-reports/5231214.pdf

[2] Beckler, M and others (2008). NBA Oracle. Accessed from https://www.mbeckler.org/coursework/ 150 2008-2009/10701_report.

[3] Colet, E. and Parker, J. Advanced Scout: Data mining and knowledge discovery in NBA data. Data Mining and Knowledge Discovery, Vol. 1, Num. 1, 1997, pp 121 125.