Analysis and Prediction Models For NBA Games

Danying Xu; Weidi Luo; Yuxin Sun
Department of Artificial Intelligence; Department of Information and Computing Science;
Department of Computer Science
Southeast University; Xi'an Jiaotong liverpool University;
Beijing Institute of Technology
Nanjing, 211189; Suzhou, 215028; Beijing, 100081
danyingxu@126.com; WEIDI.LUO20@student.xjtlu.edu.cn; bitsunyuxin@163.com;

1 Background & Introduction

NBA's annual MVP and team rankings always arouse the curiosity of thousands of fans. Before the release of the all-star list every year, thousands of fans bet, but they didn't expect their favorite player to lose the election. The real gambler naturally can't rely on his own feelings alone. A single player data support is not enough. As an ordinary person, facing all kinds of information from hundreds of players, it's not easy to choose the final all-star list. At this time, we can use machine learning method to predict MVP list and team performance.

Based on the data from 1946-2004 provided by basketball-reference, our project models NBA-related data from two aspects: (1) outstanding players predication (2) game result predication. We use and improve methods based on multiple machine learning and data mining, and integrate the advantages of each model through ensemble learning. The contributions of our project are as follows:

1. We perform two different modeling approaches for the outstanding players prediction problem.
2. For each modeling method for each problem, we use a variety of models for training, and achieve optimal results through multi-ensemble learning.

2 Method

Our task from Jan 25^{th}-Jan 31^{st} is to finish data processing and outlier detection. We have finished the datasets for tasks: classifying outstanding players and predicting game results.

Our task from Feb 1^{st}-Jan 8^{th} is to finish the question about finding outstanding players. We solve it in two ways: multi-class classification and binary-class classification. For each of them, we use three methods, which are Bayesian Classifier, Logistic Regression and deep learning.

2.1 Dataset for Finding Outstanding Players

2.1.1 Feature-processing

We use the professional formulas,Player Efficiency (also known as unit efficiency criterion). The basic idea of this efficiency index is to convert a player's on-court performance into a comparable number, so that players at different positions can be compared at the same starting line after conversion. Player Efficiency ignores the performance of the Defensive efficiency, so we use the OffenseREB%(offense rebound efficiency) and Defence REB%(defence efficiency) to represent the Defensive and Offense efficiency of the players. finally we use the shooting efficiency to evaluate the Hit Rate of the players. which are given as below to integrate the features including oreb, dreb,

Preprint. Under review.

reb, fta, ftm, pts, asts, stl, blk, fga, fgm, turnover, gp, tpm, tpa:

$$Playerefficiency = \frac{pts + reb + asts + stl + blk - ((fga - fgm) + (fta - ftm) + turnover)}{gp} \quad (1)$$

$$Shootingefficiency = \frac{fgm + 0.5\frac{tpm}{tpa}}{fga} \quad (2)$$

$$OffenseREB\% = \frac{oreb}{oreb + dreb} \quad \&\& \quad DefenseREB\% = \frac{dreb}{oreb + dre} \quad (3)$$

2.1.2 Outlier detection

We use the API from sklearn to get the outliers and integrating the results of different parameters which are used in LOF[1] and Iforest[2].

Lof(Local Outlier Factor) is a typical high precision outlier detection method based on density. In the LOF method, each data point is assigned an lof outlier factor which depends on the neighborhood density to judge whether the data point is an outlier, if the factor > 1, the data point is an outlier.if lof is close to 1, the data point is a normal point.

IForest (Isolation Forest)[5] is a fast anomaly detection method based on ensemble with linear time complexity and high accuracy. IForest is suitable for continuous numerical data, which defines anomaly as "more likely to be separated". It uses binary trees to segment the data, and the depth of the data point in the binary tree reflects the degree of "alienation" of the piece of data.

2.1.3 Data-processing

We use pandas to delete the outliers of playoffs, playoffs_ career, player_regular_season, player_regular_season_career, And then integrate them into two tables, meanwhile, we count the times of the players who were elected in the All star every year to be the labels of multi-train, and set '1' for the elected players, '0' for the players who is not be elected in the years of all star, which correspond to the players in playoff and regular seasons for binary-train.

2.2 Dataset for Predicting Game Results

2.2.1 Feature selection

We use two methods to select features for subsequent data processing: PCA(principal component analysis) and SelectFromModel based on LinearSVC.

PCA (Principal Component Analysis) is a method of feature dimensionality reduction. It obtains more valuable low-dimensional data by retaining some of the most important features in high-dimensional data and removing noise and unimportant features. And we aim to obtain the importance score of each raw feature by using PCA. By using some of the main features that account for more than 80% after PCA projection and the proportion of the original features in each projected feature, we can get the importance score of each original feature so that we can perform subsequent dataset processing.

SelectFromModel[3] is a meta-transformer for selecting features based on importance weights. The method we use is based on Linear Support Vector Classification.

2.2.2 Imputation of missing values

We delete data with too many missing values, and use KNNImputer models each feature with missing values as a function of label(we set $win/lost$ in dataset $team\_season.txt$ as label), and uses that estimate for imputation.

KNNImputer is that each samples missing values are imputed using the mean value from n_neighbors nearest neighbors found in the training set. Two samples are close if the features that neither is missing are close.[4]

2.3 Finding Outstanding Players

2.3.1 Problem Setup

We use two kinds of classification ways to train models and aim to find a better way for this problem.

1. Binary-class Classification The label has two values under this circumstances: 0 and 1. We use samples from 1979 to 2003 for each player and have five features in total. And the model chooses around 23 outstanding players.

2. Multi-class Classification The sample in this problem is the career data for each player. The label is the total numbers rated as outstanding player for each player. And the model chooses around 23 outstanding players.

2.3.2 Methods

There are three methods we use.

1. Bayesian Classifier

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \tag{4}$$

2. Logistic Regression Logistic regression is a model that uses a logistic function to solve classification problems. Its model is expressed as

$$P(y|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}} \tag{5}$$

The loss function is

$$L(\mathbf{w}) = \sum_{i=1}^{n} [y_i(\mathbf{w}^\top \mathbf{x_i}) - log(1 + e^{\mathbf{w}^\top \mathbf{x}_i})] \tag{6}$$

And after the model is trained, we can get the decision boundary:

$$\mathbf{w}^\top \mathbf{x} = 0 \tag{7}$$

3. Deep Learning

We have trained the two deep learning model to predict the MVP.
•Multilayer Perceptron: Mutilayer Rerception is a model Based on artificial neural network, is also a typical model to solve the binary classification.

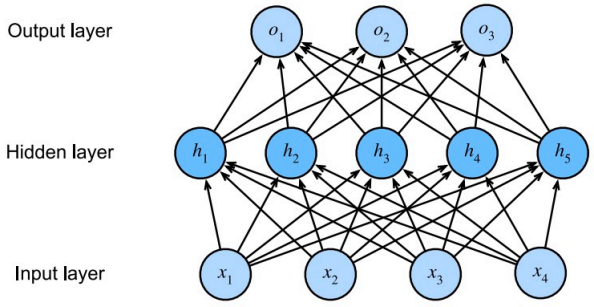

Figure 1: Multilayer Perceptron

•Long Short-Term Memory: we assume that the MVPs in every year hide a time series, LSTM is a NLP model based on RNN, which we think can be used to deal the binary classification with time series. we make use of the cell renewal and forgetting principle to remove some of the effects of retired or new players.

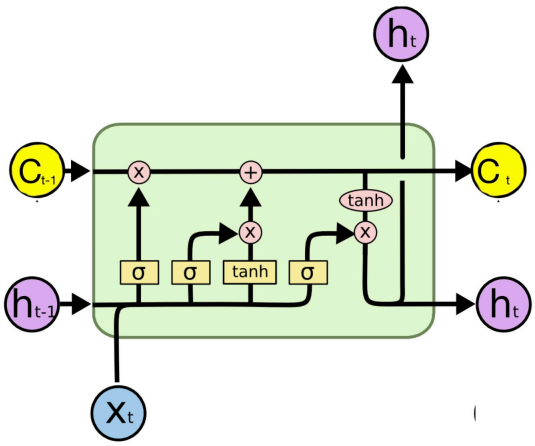

Figure 2: Long Short-Term Memory

3 Experiment Setup

3.1 Dataset for Finding Outstanding Players

1. LOF: There are mainly two parameters in LOF API, they are "k_neighbors" and "contamination". Since we have two independent training sets for finding outstanding players, the parameters are different which is show in table 1.

Parameter	Regular Season		Playoff Season	
	multi-label	binary label	multi-label	binary label
k_neighbors	20	20	20	20
contamination	$\mathbf{x}^a$	$\mathbf{x}^a$	$\mathbf{x}^a$	$\mathbf{x}^a$

Table 1: Parameters Setup in Lof

a. x represents three numbers 0.025, 0.05, 0.075 which represents the outlier ratio defined in all data, and the value range is [0, 0.5].

2. IForest: There are three parameters in IForest algorithm which are "n_estimators", "contamination" and "max_samples". Since we have two independent training sets for finding outstanding players, the parameters are different which is show in table 2.

Parameter	Regular Season		Playoff Season	
	multi-label	binary label	multi-label	binary label
n_estimators	100	100	200	100
contamination	$\mathbf{x}^a$	$\mathbf{x}^a$	$\mathbf{x}^a$	$\mathbf{x}^a$
max_samples	256	256	256	256

Table 2: Parameters Setup in IForest

a. x represents three numbers 0.025, 0.05, 0.075 which represents the outlier ratio defined in all data, and the value range is [0, 0.5].

3.2 Dataset for Predicting Game Results

3.2.1 Feature Selection

We need to use relatively complete data for feature selection, so we select the data from 1999 to 2003 as the training set for feature selection. For PCA, we can directly carry out dimension reduction training for all features. For the selectfrommodel algorithm, we specify the $win/lost$ ratio as the objective function to train it.

3.2.2 Imputation of missing values

Through data $team\_season.csv$, we can find that the missing value feature includes o_oreb,o_dreb,o_reb,o_stl,o_to,o_blk,o_3pm,o_3pa,d_fgm,d_fga,d_oreb,d_dreb,d_reb,d_asts,d_pf, d_stl,d_to,d_blk,d_3pm,d_3pa, pace, up to 21 kind of features. We simply delete the data of 1946-1972 and 1976-1978 with more missing data.

Then we use KNNImputer class to fullfill the rest missing data.

3.3 Finding Outstanding Players

3.3.1 Problem Setup

1. Binary-class Classification We use data from 1979 to 2002 as training data and 2003 as testing data. There are 8890 samples in training set and 435 samples in test set. (We tried to divide the data set into training set: test set = 3:1, but the results showed that too little training data resulted in worse model performance, so the above split method was adopted.)

2. Multi-class Classification The value of label is 0-20 (without 15, 17, 18) . The total numbers are counted from 1946 to 2002, and we still use the test set in binary-class classification as the test set here (the multi labels are changed to 1 and 0 before testing). There are 3718 samples in training set and 435 samples in test set. (We tried to divide the data set into training set: test set = 3:1, but the results showed that too little training data resulted in worse model performance, so the above split method was adopted.)

3.3.2 Methods

1. Bayesian Classifier

2. Logistic Regression For binary-class classification, penalty is L2 norm and the solver for loss function uses "lbfgs". We get the best model with class 0 and 1 both weights 0.5.

For multi-class classification, penalty is L2 norm and the solver for loss function uses "lbfgs". We get the best model with class 0 weights 0.6, class 1 weights 0.86 and other labels weight 1.

3. Deep Learning For MLP of binary-class classification, we add the batch normal layer and drop out layer as a training skill, then we have set 3 layers,with ReLU as the Activation Function(hidden layer) and choose the cross entropy function for binary classification. For the enormous dataset, We choose Adam as optimizer to make the cost function convergent quickly. we get the best model with with 0 and 1 both weight 0.5.

For LSTM of binary-class classification, We also use the LSTM to do the pre-training, then use a fully connected layer to output two classes. Cross entropy function is used for binary classification,and we use the Adam as optimizer to do the backward.we get the best model with with 0 and 1 both weight 0.5.

4 Results

4.1 Dataset for Finding Outstanding Players

1. Outlier Detection

For LOF and IForest algorithms, our detailed outliers numbers are listed in table 3 and table 4.

Outlier Ratio	Regular Season Unioned Number	Playoff Season Unioned Number
0.025	38	0
0.05	72	2
0.075	115	2

Table 3: Multi-labels Dataset Outliers

Note that both outlier algorithms have several defined outlier ratio, so we unioned the outliers of the two algorithms to remove outliers that were completely unacceptable.

Outlier Ratio	Regular Season Unioned Number	Playoff Season Unioned Number
0.025	72	19
0.05	166	70
0.075	275	144

Table 4: Binary-labels Dataset Outliers

4.2 Dataset for Predicting Game Results

1. "team_season.txt" Feature Selection

For PCA feature selection, we have selected first 7 dimension that has 80% explain variance in Figure .

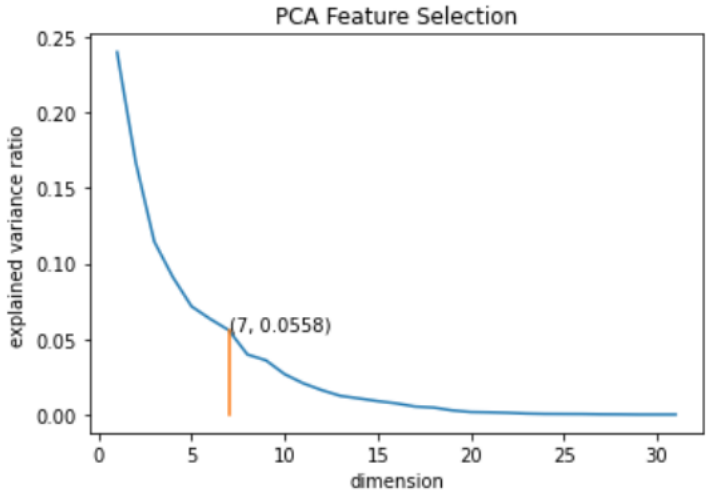

Figure 3: PCA Feature Selection

The selected features indexes (31 total features) in each dimension is shown in table .

dim 1	31	2	17	16
dim 2	9	18	11	19
dim 3	12	7		
dim 4	15	3		
dim 5	28	29	26	21
dim 6	8	4		
dim 7	10	16		

Table 5: Selected Feature index

Since they are not similar, we can't eliminate features from the original datasets.

For embedded feature selection, we have 30 selected features out of 31 features, and the selector deleted the feature $pace$.

4.3 Finding Outstanding Players

1. Bayesian Classifier

2. Logistic Regression

The evaluation for the model are shown in Figure 4 and 5.

	precision	recall	f1-score	support
0	0.98	0.98	0.98	413
1	0.58	0.64	0.61	22
accuracy			0.96	435
macro avg	0.78	0.81	0.79	435
weighted avg	0.96	0.96	0.96	435

Figure 4: binary-class classification model

	precision	recall	f1-score	support
0	0.97	0.99	0.98	413
1	0.73	0.50	0.59	22
accuracy			0.97	435
macro avg	0.85	0.75	0.79	435
weighted avg	0.96	0.97	0.96	435

Figure 5: multi-class classification model

According to the results, the class 0 can be modeled well yet class 1 is not well modeled. We think the reason is that the number of samples labeled "1" are too low to train a good model. And changing the weights of classes and samples have no use.

3. Deep Learning

The evaluation for the model are shown in Figure 6 and 7.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	414
1	0.71	0.55	0.62	22
accuracy			0.97	436
macro avg	0.84	0.77	0.80	436
weighted avg	0.96	0.97	0.96	436

Test loss: 0.1164049357175827, Acc: 0.9655963302752294

Figure 6: Long Short-Term Memory model

	precision	recall	f1-score	support
0	0.97	0.99	0.98	414
1	0.69	0.41	0.51	22
accuracy			0.96	436
macro avg	0.83	0.70	0.75	436
weighted avg	0.96	0.96	0.96	436

Test loss: 0.11209103465080261, Acc: 0.9610091743119266

Figure 7: Multlayer Perception model

According to the model, the LSTM model seems better than MLP, but the effect is also not very good. The reason is that Basketball is a memorized process, and there are mvps in each era, but the excessive number of zeros in training sets causes the deviation of the model effect.

5 Conclusion

During Jan 25^{th}-Jan 31^{st}, we have obtained two datasets for finding outstanding players and one dataset for predicting game results.

To get the dataset for finding outstanding players, we first finish the outlier detection using mainly LOF and IForest algorithms. In order to eliminate unacceptable samples, we union results in two algorithms. For multi-label classification dataset, we delete 38 samples in regular season and 2 samples in playoff season. For binary-label classification dataset, we delete 166 samples in regular seasons and 70 samples in playoff season. Then we combine regular and playoff together with weight equals to 0.7 and 0.3 respectively and obtain 2 classification datasets. The datasets include training set and test set, where the size of multi-label dataset is 3718×7 (label included), and the size of binary-label dataset is 9788×8 (label included).

To get the dataset for predicting game results, we first use two feature selection algorithms (PCA and embedding) to evaluate feature importance. Considering some samples lost too much features, we only use samples in 1973-1975 and 1979-2004 (with test samples included). We use regression to fill in 2 feature values in vacant year which is trained by 1999-2004 existing data. The final dataset includes training set and test set, where the size is 766×35(label included).

Xu&Luo's poster draft is like Figure 8.

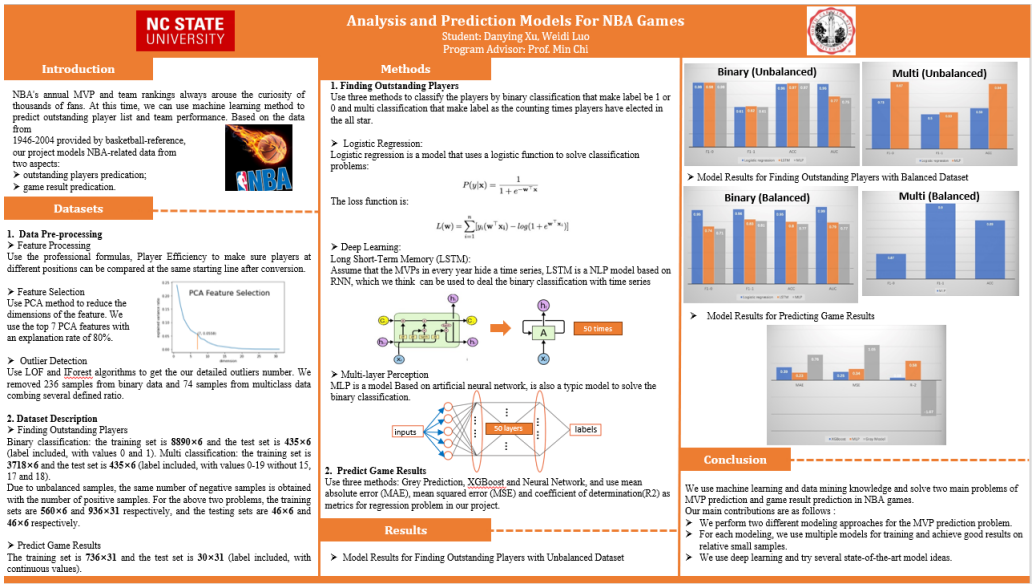

Figure 8: Poster Draft

Sun's poster draft is like Figure 9.

References

[1] Sci-kit Learn.$LocalOutlierFactor$.Sci-kit Learn.
URL: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

[2] Sci-kit Learn.$IsolationForest$.Sci-kit Learn.
URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

[3] Sci-kit Learn.$SelectFromModel$.Sci-kit Learn.
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html

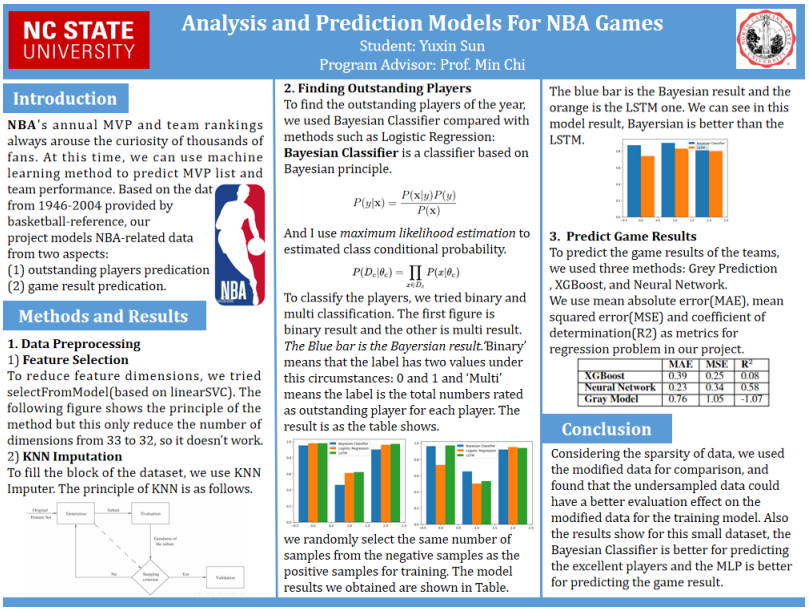

Figure 9: Poster Draft of Sun

[4] Sci-kit Learn.$KNNImputer$.Sci-kit Learn. https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

[5] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." 2008 eighth ieee international conference on data mining. IEEE, 2008.