Players' Value Prediction Based on Machine Learning Method

Daokang Zhang*, Caixin Kang
School of Computer Science, Sichuan University, Chengdu, Sichuan, 610207

*Corresponding author: daokangzhang@scu.edu.cn

Abstract. It is important to predict football players’ value, especially during transfer period. This paper uses the player information and value data of the game FIFA 18 as data source. It is able to realize the prediction of its players’ best positions and values. After reducing the dimensionality of the value prediction model, a cluster analysis on the player’s position is introduced, and then grid search method is adopted to adjust the Xgboost parameters. Finally, Xgboost method is used to predict the player's worth. The experimental results show that certain accuracy is achieved, but there is still room for improvement in the accuracy of prediction. Discussions based on experiment results are made.

Keywords: Machine learning, value prediction, decision tree.

1. Introduction
At present, football is one of the most popular sports. Predicting the value of players during transfer period based on various data of active players can effectively assess the value of players and assist major clubs in formulating transferring strategies with a limited budget to achieve a cost-effective lineup adjustment. Therefore, predicting the value of each player is of great importance.

There are many research methods to predict player value based on various data, such as artificial neural network model, regression analysis, K-Nearest Neighbor (KNN) regression and support vector machine model. These methods have different advantages and disadvantages, and have different performances in different data sets. However, artificial neural network method suffers from over-fitting, and may deviate greatly from the actual situation, as a result of falling into the local optimal solution. The support vector machine method will weaken the computer's solving ability when the amount of data is too large.

This paper takes the FIFA 18 game generated data set as an example. Through the preprocessing and analysis of the original data, all players are divided into two categories: goalkeepers and non-goalkeepers. Firstly, based on the K-MEANs clustering algorithm, non-goalkeeper players are clustered into three groups. The analysis, on one hand, provides a reference for the selection of the player's position. On the other hand, it removes some features that have little relationship with the player's value to improve the accuracy of prediction. Subsequently, based on the Xgboost model, the player's ability values are mined to predict their value.
2. **K-means clustering algorithm**

The K-means clustering algorithm is an unsupervised real-time clustering algorithm proposed by Mac Queen [1], which divides the data into a predetermined number of clusters on the basis of minimizing the error function.

### 2.1. The basic idea of K-means algorithm [2]

The basic idea of partition is that given a data set containing \( n \) samples, the partition method divides the data into \( k \) partitions. Each partition represents a cluster, at the same time satisfying:

1. Each cluster contains at least one sample;
2. Each sample must belong to one and only one cluster.

The basic idea of the K-means algorithm is to select \( k \) samples as the initial clustering center, and divide the data objects into different clusters through iteration, so that the similarity between the objects in the cluster is very large, and the similarity of the objects between the clusters is small. The steps of the algorithm are:

1. Randomly select \( k \) objects from \( n \) samples as initial cluster centers.
2. Calculate the distance between each object and these central objects, and re-divide the corresponding objects according to the minimum distance, and assign each object to the closest class.
3. Recalculate the mean of each cluster.
4. Repeat (2) and (3) until each cluster no longer changes.

The K-means algorithm tries to find the \( k \) clusters that minimize the square error function value, which is defined as follows:

\[
E = \sum_{i=1}^{k} \sum_{p \in c_i} | p - m_i |^2
\]  

In the formula, \( E \) is the sum of squared errors of all objects. \( p \) is a point in space, representing a given data object. \( M_i \) is the average value of the cluster. This criterion makes the resulting clusters as compact and independent as possible.

3. **Xgboost algorithm**

The full name of Xgboost is Extreme Gradient Boosting, which is an extreme gradient boosting tree. This algorithm is an extension of the gradient boosting machine algorithm.[3]

The Boosting classifier comes from an ensemble learning model. Its basic idea is to combine multiple tree models with lower accuracy into a model to improve accuracy. The model is iterated continuously, and each iteration generates a new tree. When the gradient boosting algorithm generates each tree, the idea of gradient descent is used. Based on all the trees generated in the previous step, it moves in the direction of minimizing the given objective function. But when the data set is large and complex, the calculation amount of the gradient boosting algorithm is huge.

Xgboost algorithm is the implementation and improvement of gradient boosting machine algorithm. Xgboost's base learner has the characteristics of high accuracy, not easy to overfit, and scalability. It can process high-dimensional sparse features in a distributed manner. Under the same circumstances, the XGBoost algorithm is more efficient than similar algorithms.

### 3.1. Decision tree ensemble classifier [4]

The decision tree representation method has been widely used in classification algorithms due to its intuitive form and reliable algorithm basis. However, the prediction performance of a single decision tree is relatively limited, so decision tree integration is generally used to improve performance.

Given a data set \( D = \{ (x_i, y_i) \} \) (\( |D| = n, x_i \in R^m, y_i \in R \)), the integration model of the tree is as follows:
\[ \hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \]  

(2)

In the formula, \( F = \{ f(x) = w_{q(x)} \}(q : R^n \rightarrow T, w \in R^T) \) is the collection space of the regression tree. \( x_i \) represents the feature vector of the \( i \)th data point, \( q \) indicates that the structure of each tree is mapped to the leaf index corresponding to the sample, and \( T \) indicates the number of leaf nodes of the tree. Each tree \( f_k \) corresponds to an independent tree structure \( q \) and leaf weight \( w \).

The objective function consists of two parts:

\[ \text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \]  

(3)

The first part is the training error between the predicted value and the target true value, and the second part is the sum of the complexity of each tree.

3.2. Xgboost basic principles [5]

Because the objective function in the integrated decision tree model of the above formula cannot be optimized by traditional methods such as Euclidean distance, the Xgboost algorithm adds a new function to the model every time on the basis of retaining the original model.

\[ \hat{y}_i^{(0)} = 0 \]
\[ \hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \]
\[ \hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \]

(4)

...\n
\[ \hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \]

\( \hat{y}_i^{(t)} \) is the model prediction value of the \( i \)th sample in the \( t \)th iteration, after \( \hat{y}_i^{(t)} \) retaining the predicted value \( \hat{y}_i^{(t-1)} \), a new function \( f_t(x_i) \) is added.

Subsequently, rewrite the objective function of Xgboost as:

\[ \text{Obj}^{(t)} = \sum_{i=1}^{n} l \left( y_i, \hat{y}_i^{(t-1)} + f_t(x_i) \right) + \Omega(f_t) + \text{cons} \tan t \]  

(5)

We use \( f_t \) to optimize this objective function. When the error \( l \) is the square error, the objective function is:

\[ L^{(t)} = \sum_{i=1}^{n} [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + \text{cons} \tan t \]  

(6)

After removing the constant, a relatively uniform objective function is obtained:
\[
\hat{L}^{(s)} = \sum_{i=1}^{n} [g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_i)
\] (7)

3.3. Xgboost model parameters

Xgboost mainly contains three types of parameters: general parameters, booster parameters and task parameters.

Among them, the general parameters are related to the lifter we use for lifting, usually a tree model or a linear model. The booster parameters depend on the lifter selected by the algorithm here. The parameters of the learning task determine the learning scenario, including learning tasks and corresponding learning objectives.

In the Xgboost model, the common parameters are described as follows:

1. Booster: to set the ascent model to be used. You can choose tree model or linear model.
2. Learning rate: to determine whether the objective function can converge to the local minimum and when to converge to the minimum.
3. Max depth: the maximum depth of the decision tree, generally between 3-10. The deeper the tree, the easier it is to overfit.
4. N_estimators: the maximum number of iterations. The more iterations, the more accurate the result is, but it will also increase the time complexity.
5. Min_child_weight: the minimum leaf node weight sum. If in a split, the weight sum of all samples on the leaf node is less than min_child_weight, the splitting will stop, which can effectively prevent overfitting and learning of special samples.
6. Objective: to define learning goals.

4. Data preprocessing

The data preprocessing stage is an indispensable part of machine learning, as it will make the data more effectively recognized by the model or evaluator.

4.1. Data sources

All data in this paper comes from the data set of player data indicators in the game FIFA 18. FIFA 18 is a football sports game produced and distributed by Electronic Arts. The game simulates various skill indicators of players in reality, and sets their market value according to player data. The worth of the game excludes some uncontrollable factors, including media hype, commercial competition, etc. The prediction of player value based on this data set pays more attention to the ability of the players themselves, which can provide a reference for the formulation of the transfer policy during the transfer period.

4.2. Differentiated processing of data characteristics

Each player in the data set has 53 common characteristics. Players are divided into goalkeepers and non-goalkeepers. Non-goalkeepers have 10 characteristics that describe their own position-related ability values. Goalkeepers have the ability to describe themselves in the goalkeeper position gk.

Since the gk ability value for non-goalkeeper players does not exist, instead of treating it as a missing value, this paper prefers to divide the entire data set into a goalkeeper data set and a non-goalkeeper data set, with 54 and 64 features respectively.

Even so, the number of characteristics of each player, especially non-goalkeepers, is still large. Too many characteristics may lead to a decrease in the prediction effect of the model. Therefore, this paper proposes to cluster each player's position based on K-Means algorithm, using the location-related features of the player. This processing method can not only reduce the number of features in the player's value model, but also provide a reference for the player's best position. Since the selected features are all related to the determination of the player's position, the clustering result has a certain degree of credibility.
4.3. Data cleaning
In order to obtain valid data, we perform data cleaning on the original data. The specific process is as follows:

1. The data set contains goalkeeper and non-goalkeeper data. Among the goalkeeper positions, only gk (goal keeper) is non-empty, while other position data is empty. Non-goalkeepers are just the opposite. Therefore, the data set is divided into two categories, representing goalkeepers and non-goalkeepers.
2. Fill in the birth_date data which means year of each player, for example, add 09/10/89 to the complete 09/10/1989 in order to use the date and time library to calculate the age.
3. Convert weight and height into BMI (Body Mass Index) index when using K-means clustering algorithm.
4. For quantitative features, such as work_rate_att and work_rate_def represent the player’s offensive tendency and the player’s defensive tendency, respectively. They are represented by Low, Medium, and High, which are converted to 0, 1, and 2.

4.4. Feature processing
(1) Since there are many initial features of each sample, if all are used for value prediction, the prediction error may increase. Therefore, some location-related features are selected for player location clustering before the value prediction is performed.
(2) Delete the meaningless features such as id and league when predicting the value. These features will bring noise to the model.
(3) The position-related features are not considered in the value prediction to reduce the forecast error.

5. Experimental results and analysis
5.1. K-means clustering algorithm
Since the K-means algorithm needs to set the value of k by itself, and the elbow diagram is an important indicator for selecting the value of k, the elbow diagram of this dataset is as follows:

![Fig.1. The elbow diagram of the dataset](image)

The results found that the elbow diagram has the greatest downward trend when k=3, which is also coincident with the position distribution of the forward, midfield and back on the football field. Therefore, k=3 is used for K-means cluster analysis.

The clustering scatter plot is as follows:
5.2. **Xgboost algorithm**

5.2.1. **Experimental conditions.**

(1) Software environment:

(a) Operating system: Windows 10 x64
(b) Development platform: Python3.8
(c) Third-party library: Python: numpy + pandas + matplot + Xgboost + graphviz

(2) Hardware environment:

Processor: Intel(R) Core™ i5-8250U CPU @ 1.60GHz
RAM: 8GB

5.3. **Parameter**

After obtaining the required features, grid search method is used to adjust the parameters.

The grid search method is a parameter adjustment method. It uses the idea of exhaustive search. During the selection of all candidate parameters, every possible combination is tried and finally the best one is left.

The final set Xgboost parameters of non-goalkeeper data set are shown in the table:

| parameter       | value |
|-----------------|-------|
| learning_rate   | 0.07  |
| n_estimators    | 1000  |
| max_depth       | 4     |
| min_child_weight| 5     |
| objective       | reg:gamma |

The following figure shows the importance scores of the features in the Xgboost model. It can be seen from the figure that the feature scores of potentials, birth_date, and reaction are very high, which means that these features have a very large impact on the model. This is also coincident with real life.[6]
Fig. 3. The importance scores of the features

At the same time, decision tree of this model is shown. It can be seen that some important features are the key nodes of the decision tree.

Fig. 4. The decision tree of model

Finally, three types of indicators, including MSE (Mean Squared Error), R2-score, and average error are used. It is found that most of the value prediction fluctuates within a relatively normal error range. The values of each indicator are as follows:

Table 2. The values of each indicator

| index        | value               |
|--------------|---------------------|
| MSE          | 5709.696328350747   |
| R2-score     | 0.971717378775396   |
| average error| 0.14238160388609106 |

As shown in the figure below, most of the players' prediction errors in the test set are within the normal range, and a few of them have large errors.
The experimental results show that the Xgboost method has certain accuracy under each evaluation system, but there is still room for improvement in the accuracy of prediction. The reason is that the data set itself comes from a game. In fact, the value of each player should be generated by a certain algorithm. Although the Xgboost method can predict the value of a player with a certain degree of accuracy, it does not fully explore the actual value of each player.

Fig.5. Comparisons and Visualization of Absolute Errors in Output of Predicted Values of Different Models

6. Conclusions
This paper is based on the Xgboost method to predict players' value. The paper uses the player information and ability data in the game FIFA 18 as data source. Xgboost method is adopted to predict the player value.

This paper firstly performs a cluster analysis of player's position, then uses the grid search method to adjust the parameters of Xgboost, and finally uses the Xgboost method to predict the player's value.

This Xgboost-based method can predict players' value according to the player's ability value to a certain extent. It can also provide new ideas for prediction in other sports fields. The innovative application of machine learning algorithms in the football field has certain practical significance.

References
[1] MingXiu Duan. Research and Application of Hierarchical Clustering Algorithm [M]. Central South University, 2009.
[2] Chunfang Luo, Guohua Zhang, Dehua Liu, Dinghuan Zhu. Research on K-means clustering based XGBoost ensemble algorithm College of Science [M]. Hunan University of Technology, 2020.
[3] ZHAO Tianao, ZHENG Shanhong, LI Wanlong, LIU Kai. A Study of the Credit Risk Analysis Based on XGBoost [J]. School of Computer Science & Engineering, Changchun University of Technology, Jilin 130012, China, 2018.
[4] QianYi Ye. Commercial sales forecast based on Xgboost [J]. Nanchang University, 2017.
[5] SONG Guo-qin, LIU Bin. The Establishment and Application of Drop-Out-Index of MOOCs Based on XGBoost Feature Selection [J]. China West Normal University, University of Electronic Science and Technology of China, 2018.
[6] Jinwen JIANG, Weiguang LIU, Application of XGBoost algorithm in manufacturing quality prediction [J]. Shenzhen City University, 2017.