International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 3, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v3n2.13
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Multiple Regression Models Were Used to Predict Match Outcome Through Football Team Match Data

Chenrui Wang

Zhejiang University of Technology, Hangzhou, China
202103340327@zjut.edu.cn

ABSTRACT

With the increasing prosperity of football, the result prediction of football matches has become a hot spot in the commercial operation of sports, and also an important issue studied by the academic circle. Research about the results of football prediction, most research scholars from the factors of the results, such as the team strength, the weather, the team ranking, team status, coach, team home and away combat ability, but a large number of historical game data collection is more difficult, and part of the political factors cannot be quantified. A former study found that gambling companies mainly analyze the data of football games, while the team mainly focuses on the presence of the players, so as to analyze the situation before and after the game. The analysis of the influencing factors of competition results is mostly the way to calculate the complete influencing factors by traditional methods to realize the purpose of competition analysis and prediction. This paper takes the game data in the football website of the scout network as the data source, and captures the historical data of all the recent two seasons of the English Premier League through the web crawler technology. The collected data were cleaned in detail, and the football history data were analyzed by multiple factors. Multiple regression models were used to predict the outcome of football matches, and the influencing factors that were valuable for the analysis and prediction of football matches were found.

KEYWORDS

Football game; Crawler; Regression model; Outcome prediction

1. INTRODUCTION

As a popular project in the field of sports, football has attracted wide attention from fans at home and abroad. Meanwhile, data analysis and data mining in the field of football are also developing rapidly. As a favorite of the gambling industry and sports participants, the game prediction has become one of the main focus of football business activities, and also the focus of the industry. Due to the many factors affecting the outcome of football matches, it is quite difficult to predict the outcome of a football match in reality. For this reason, both business circles, academia and sports circles are more willing to study the correlation analysis of football matches. As a scientific evaluation method for multi-factor quantitative analysis, the paste gray association analysis method has a good effect on the complex problems of the influencing factors, so it is very suitable for the association analysis of football matches. Through the correlation analysis method of this paper, we can explore the influencing factors after feature engineering more practically, explore valuable information, and provide effective analysis and prediction methods for football participants. With a deeper understanding of Python app development, I want to predict the results of the team's recent games to see if the so-called "surprises" can trace the data characteristics. In order to unify the data, the author

chose all the games in the last two seasons in the Premier League, hoping to roughly predict the outcome through machine learning.

2. DATA COLLECTION-CRAWLER

The author first checked the robots of Tencent sports network. The txt file, found that the game data is allowed.

Figure 1. The robots of Tencent Sports Network.txt document

When the author used the Beautifulsoup library to crawl the data of the website, I found that no data could be accessed. After querying the data, I learned that the Beautifulsoup library only supports the standard HTML resolution library of Python and some third-party resolution libraries, but for. The shhtml website, the common urllib method is not to grasp the data, so the author subsequently use the Selenium open source tool to simulate the human opening the website to obtain the data.

The suffix of Premier League matches is a string of regular numbers, and most of them use the same css style, so you can automatically climb the data by loop, and capture some websites that cannot be crawled by setting a hidden waiting time.

Figure 2. Example of data crawl code demonstration results

```
for i in range(begin, end + 1):

    if (i in cantCrawl):
        continue

    try:

        options = EdgeOptions()
        options.use_chromium = True
        options.add_argument('headless')
        browser = Edge(options=options)

        net = "https://kbs.sports.qq.com/kbsweb/game.htm?mid=8:"+str(i)
        browser.get(net)
        error = ""

        try:
            element = WebDriverWait(browser, 15).until(
                EC.presence_of_element_located((By.XPATH, '//*[@id=\"data-box\"]/div[1]/

        except:
            error = error +
            raise Exception()


        error = error + "win"
```

Figure 3. Handling exceptions through implicit waiting

The resulting data are:

```
In [2]: data = pd.read_csv('./data.csv')
        data
```

Out[2]:

	index	胜平负	主队	主进球	主控球率	主助攻	主射正	主射门	主传球	主抢断	...	客射正	客射门	客传球	客抢断	客角球	客任意球	客越位	客犯规	客黄牌	客红牌
0	2210271	0	布伦特福德	2	35.3%	1	3	8	22	20	...	4	22	22	20	2	12	1	8	0	0
1	2210272	1	伯恩利	1	35.9%	1	3	14	14	16	...	8	14	14	16	7	10	0	7	1	0
2	2210273	0	切尔西	3	61.7%	1	6	13	4	21	...	1	4	4	21	5	14	1	11	0	0
3	2210274	0	埃弗顿	3	48.1%	3	6	14	6	23	...	3	6	6	23	6	11	0	15	0	0
4	2210275	0	莱斯特城	1	56.3%	1	5	9	17	15	...	3	17	17	15	5	6	4	10	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
654	2293152	0	富勒姆	5	57.1%	4	7	17	18	22	...	9	18	18	22	2	11	4	15	3	0
655	2293153	0	利物浦	1	53.9%	1	5	15	5	9	...	1	5	5	9	7	19	2	10	2	0
656	2293154	0	曼城	2	81.3%	2	6	18	4	7	...	2	4	4	7	10	3	3	16	3	0
657	2293155	1	纽卡斯尔联	0	54.2%	0	5	12	10	16	...	6	10	10	16	9	15	1	12	0	0
658	2293156	0	诺丁汉森林	4	36.1%	3	4	9	19	19	...	5	19	19	19	2	16	1	10	0	0

659 rows × 30 columns

Figure 4. Data obtained by the crawler crawling

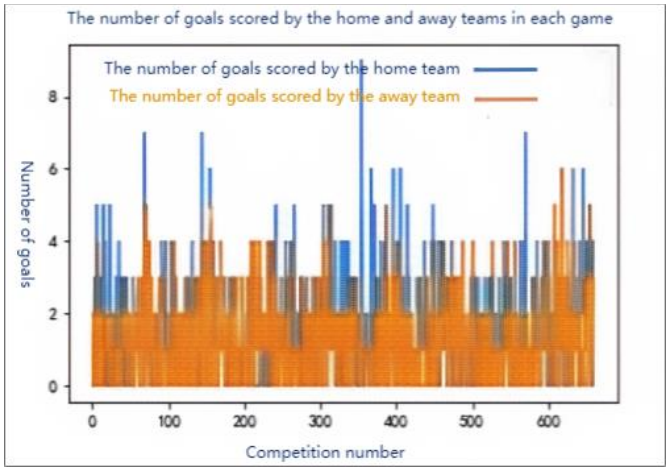

Figure 5. The main ball possession rate and win and draw the negative relationship chart

3. VISUAL PRESENTATION OF THE DATASET

3.1. Main Ball Possession Rate and Winning and Drawing Negative Data Display

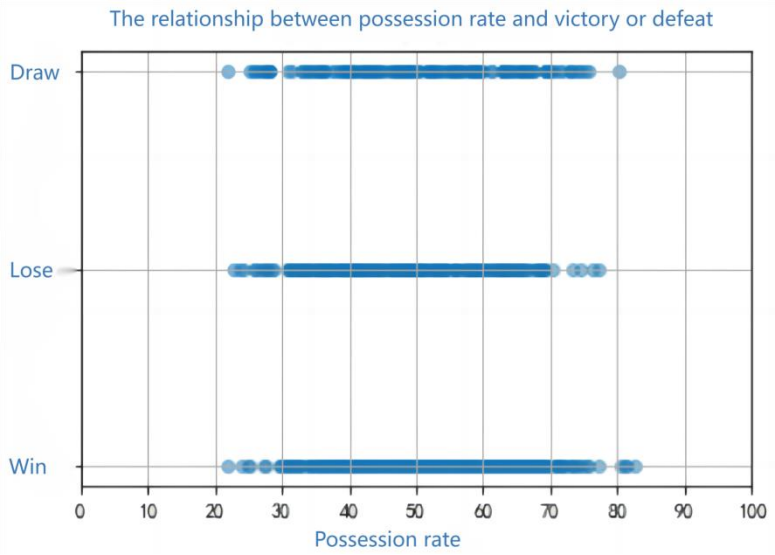

Figure 6. Line chart of main goal and guest goal

It can be seen from this figure that the ball control rate affects the relationship between victory and defeat to a certain extent. The higher the ball control rate, the tighter the data points in the "win"

column, the greater the probability of winning, the lower the ball control rate, the tighter the data points in the "negative" column, the smaller the probability of winning, but this influence is relative and not decisive. The possibility of a draw also needs to be considered.

3.2. Display of Main Goal and Guest Goal Data

As can be seen from this chart, the goals of the home team are generally more than the visiting team. It can reflect the influence of home advantage on the game. For further analysis, home advantage includes temperature, humidity, court size, the familiarity of the court, the number of home fans, etc. Such advantages often lead to the home team and scoring more goals than the visiting team.

3.3. Display of Main Possession Rate and Main Shot Times

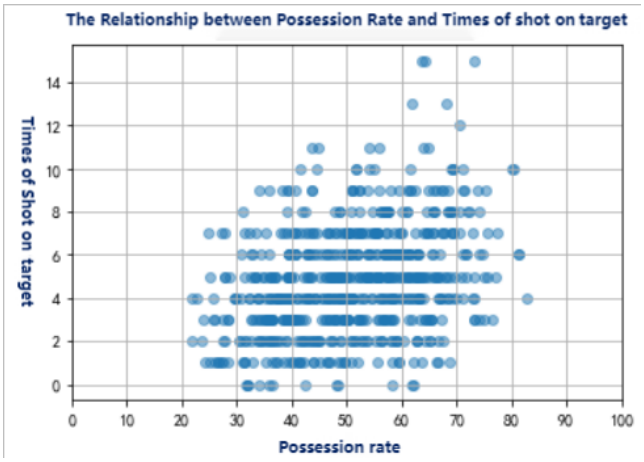

Figure 7. The Relationship between possession Rate and Times of Shot on target

The relationship between the rate of ball control and the number of positive shots can be seen from this scatter plot. The higher the ball, the more shots on goal. From the figure, the relationship between the number of shots is more obvious, so the rate of possession greatly affects the number of shots. Further analysis, the higher the ball control, the more times the team gets the ball, and the more times the two teams get, the more the ball, the more chances the team has to advance to the other half, the more chances to shoot, and the more shots to correct.

4. DATA PREPROCESSING

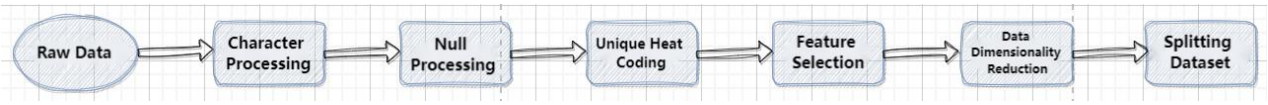

Figure 8. Data preprocessing process

Data and features determine the upper limit of machine learning. The data obtained by the author may contain a large number of missing values, or a lot of noise, or may have abnormal points due to manual input errors, which is very detrimental to the training of the algorithm model. The result of data cleaning is to process all kinds of dirty data, get standard, clean and continuous data, and provide data statistics, data mining, etc.

(1) Character processing

There is a percentage in the crawled data, and python will be treated as a string, which is not conducive to the subsequent model establishment. So change this data to floating point.

(2) Null value processing

Tuples with null values in the data were removed.

(3) Independent heat encoding treatment

There are category data (team name) in the data, which needs to use single-heat coding to transform meaningless discrete features into meaningful single-heat coding.

(4) feature selection

There are 70 data attributes in the data, considering the attribute of variance 0 in the data (excluding the encoding).

(5) Data dimension reduction

In this experiment, the PCA algorithm was used for data dimension reduction, and the parameters were adjusted for each model.

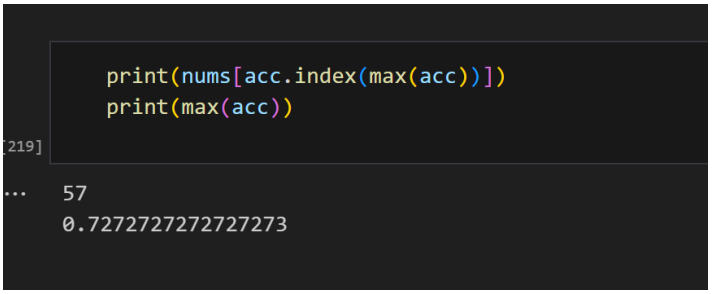

Figure 9. shows the random forest model where n_components take 57

(6) Data Set Division

Figure 10. divides the data set according to the ratio of 8:2

5. MACHINE LEARNING

5.1. Logical Regression Model

The main goal of this code is to classify the data after PCA dimension reduction using logistic regression models and find the number of features with the highest accuracy.

Code analysis:

(1) Import the required libraries:

- `from sklearn.decomposition import PCA`: For performing PCA dimension reduction.

- `from sklearn.linear _ model import LogisticRegression`: Logical regression model.

- `from sklearn.metrics import accuracy_score `: It is used to calculate the accuracy rate.

- `from sklearn. Model _ selection import train _ test _ split `: used for dividing the training set and the test set.

- `import pandas as pd`: For data processing.

(2) Initialize the empty list ` numbers ` and ` acc ` are used to store feature number and accuracy.

(3) Number of loop iteration features ` i ` from 2 to 69:

-Extract features from the raw data and remove the target variable column (win, draw, main goal, guest goal) by ` drop ` method.

-Convert feature column names to string type to ensure consistent feature name format required by the PCA.

-Create the PCA instance ` pca ` and set the number of principal components to ` i `.

-PCA dimensionality reduction for the training data ` X ` using the ` fit _ transform` method.

-PCA dimensionality reduction for the test data ` tX ` using the ` fit _ transform` method.

-The dimension reduction dataset was divided into training and test sets, using the ` train_test_split ` method.

-Create a logistic regression model instance ` model `, increasing the maximum number of iterations.

-Training the model, fitting the training set data using the ` fit ` method.

-Prediction on the test set, using the `predict` method to predict the labels of the test data ` tX `.

-To calculate the prediction accuracy, and the accuracy of the actual label ` ty ` and the predicted label ` ty _ pred ` was calculated using the `accuracy_score ` method.

-The feature number ` i ` and accuracy `accuracy` are added to the list ` numbers ` and ` acc `.

(4) Output the number of features with the highest accuracy, using ` numbers [acc. Index (max (acc))] ` and ` max (acc) `.

Explain the logistic regression model:

Logic regression is a statistical model used for classification problems. In dichotomous problems, the logistic regression predicts the probability of a sample belonging to a certain category by multiplying the features with the weights and transforming them through a logistic function (e. g., the sigmoid function). If the prediction probability is greater than a predefined threshold, the sample is classified as a positive class; otherwise, it is classified as a negative class.

The logistic regression model has the following characteristics in the classification task:

-Simple and effective: Logistic regression is a linear model that is easy to understand and implement.

-Predicted probability: Logistic regression outputs the probability that a sample belongs to a category, not just the result of binary classification.

-Interpretability: The logistic regression model can explain the influence of each feature on the classification results, and the importance of the features can be judged by the coefficient.

In this code, the logistic regression model is used to classify the data after PCA dimension reduction and find the number of features with the highest accuracy on the test set by adjusting the number of features.

5.2. Random Forest Model

The main goal of this code is to classify the data after PCA dimension reduction using a random forest model and to find the number of features with the highest accuracy.

Code analysis:

(1) Import the required libraries:

- `from sklearn.decomposition import PCA`: For performing PCA dimension reduction.

- `import pandas as pd`: For data processing.

- `from sklearn. Model _ selection import train _ test _ split `: used for dividing the training set and the test set.

- `from sklearn.ensemble import RandomForestClassifier`: Random forest classifier model.

- `from sklearn.metrics import accuracy_score `: It is used to calculate the accuracy rate.

(2) Initialize the empty list ` numbers ` and ` acc ` are used to store feature number and accuracy.

(3) Number of loop iteration features ` i ` from 2 to 69:

-Extract features from the raw data and remove the target variable column (win, draw, main goal, guest goal) by ` drop ` method.

-Convert feature column names to string type to ensure consistent feature name format required by the PCA.

-Create the PCA instance ` pca ` and set the number of principal components to ` i `.

-PCA dimensionality reduction for the training data ` X ` using the ` fit _ transform` method.

-PCA dimensionality reduction on the test data ` tX ` using the `transform` method.

-The dimension reduction dataset was divided into training and test sets, using the ` train_test_split ` method.

-Create a random forest classifier model instance ` model `.

-Training the model, fitting the training set data using the ` fit ` method.

-Prediction on the test set, using the `predict` method to predict the labels of the test data ` tX _ transformed`.

-To calculate the prediction accuracy, and the accuracy of the actual label ` ty ` and the predicted label ` ty _ pred ` was calculated using the `accuracy_score ` method.

-The feature number ` i ` and accuracy `accuracy` are added to the list ` numbers ` and ` acc `.

(4) Outputs the number of features with the highest accuracy and the highest accuracy, respectively using ` numbers [acc. Index (max (acc))] ` and ` max (acc) `.

5.3. Support Vector Machine Model

The goal of this code is to use the support vector machine (SVM) model to classify the data after the PCA dimension reduction, and to find the number of features with the highest accuracy.

Code analysis:

(1) Import the required libraries:

- `import pandas as pd`: For data processing.

- `from sklearn. Model _ selection import train _ test _ split `: used for dividing the training set and the test set.

- `from sklearn.svm import SVC`: Support vector machine model.

- `from sklearn.metrics import accuracy_score `: It is used to calculate the accuracy rate.

(2) Initialize the empty list ` numbers ` and ` acc ` are used to store feature number and accuracy.

(3) Number of loop iteration features ` i ` from 2 to 69:

-Extract features from the raw data and remove the target variable column (win, draw, main goal, guest goal) by ` drop ` method.

-Convert feature column names to string type to ensure consistent feature name format required by the PCA.

-Create the PCA instance ` pca ` and set the number of principal components to ` i `.

-PCA dimensionality reduction for the training data ` X ` using the ` fit _ transform` method.

-PCA dimensionality reduction for the test data ` tX ` using the ` fit _ transform` method.

-The dimension reduction dataset was divided into training and test sets, using the ` train_test_split ` method.

-Create the SVM model instance ` model `.

-Training the model, fitting the training set data using the ` fit ` method.

-Prediction on the test set, using the `predict` method to predict the labels of the test data ` tX `.

-To calculate the prediction accuracy, and the accuracy of the actual label ` ty ` and the predicted label ` ty _ pred ` was calculated using the `accuracy_score ` method.

-The feature number ` i ` and accuracy `accuracy` are added to the list ` numbers ` and ` acc `.

(4) Output the number of features with the highest accuracy, using ` numbers [acc. Index (max (acc))] ` and ` max (acc) `.

Explain the SVM model:

Support vector machine (Support Vector Machine, SVM) is a supervised learning model used for classification and regression. The goal of the SVM is to find an optimal hyperplane that is able to separate the different classes of samples and maximize the spacing on both sides. The SVM determines the classification boundary by finding the support vector (the closest point to the sample in the hyperplane).

5.4. KNN Model

```
In [132]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
import numpy as np

nums = []
acc = []

for i in range(2, 70):
    X = data.drop(labels=['胜平负', '主进球', '客进球'], axis=1)  # 提取特征
    tX = test_data.drop(labels=['胜平负', '主进球', '客进球'], axis=1)  # 提取特征

    # 将特征列名转换为字符串
    X.columns = X.columns.astype(str)
    tX.columns = tX.columns.astype(str)


    pca = PCA(n_components=i)
    X = pca.fit_transform(X)
    tX = pca.fit_transform(tX)

    # Split the data into training and test sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # 创建KNN模型
    model = KNeighborsClassifier()

    # 训练模型
    model.fit(X_train, y_train)

    # 在测试集上进行预测
    ty_pred = model.predict(tX)

    # 计算准确率
    accuracy = accuracy_score(ty, ty_pred)
    acc.append(accuracy)
    nums.append(i)

D:\Anaconda3-2022.10-Windows-x86_64\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction fu
nctions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this beha
vior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, an
d the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
D:\Anaconda3-2022.10-Windows-x86_64\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction fu
nctions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this beha
vior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, an
d the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
D:\Anaconda3-2022.10-Windows-x86_64\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction fu
nctions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this beha
vior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, an
d the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
D:\Anaconda3-2022.10-Windows-x86_64\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction fu
nctions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this beha
vior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, an
d the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.

In [133]: print(nums[acc.index(max(acc))])
print(max(acc))

2
0.451505016722408
```

Figure 11. KNN model

The goal of the code is to use the K nearest neighbor (K-Nearest Neighbors, KNN) model to classify the data after PCA dimension reduction and find the number of features with the highest accuracy.

Code analysis:

1) Import the required libraries:

- `import pandas as pd`: For data processing.

- `from sklearn. Model _ selection import train _ test _ split `: used for dividing the training set and the test set.

- `from sklearn.neighbors import KNeighborsClassifier`: K nearest neighbor model.

- `from sklearn.metrics import accuracy_score `: It is used to calculate the accuracy rate.

- `from sklearn.decomposition import PCA`: Dimreduction for principal component analysis.

- `import numpy as np`: To process the numerical data.

2) Initialize the empty list ` numbers ` and ` acc ` are used to store feature number and accuracy.

3) Number of loop iteration features ` i ` from 2 to 69:

-Extract features from the raw data and remove the target variable column (win, draw, main goal, guest goal) by ` drop ` method.

-Convert feature column names to string type to ensure consistent feature name format required by the PCA.

-Create the PCA instance ` pca ` and set the number of principal components to ` i `.

-PCA dimensionality reduction for the training data ` X ` using the ` fit _ transform` method.

-PCA dimensionality reduction for the test data ` tX ` using the ` fit _ transform` method.

-The dimension reduction dataset was divided into training and test sets, using the ` train_test_split ` method.

-Create the K nearest neighbor model instance ` model `.

-Training the model, fitting the training set data using the ` fit ` method.

-Prediction on the test set, using the `predict` method to predict the labels of the test data ` tX `.

-To calculate the prediction accuracy, and the accuracy of the actual label ` ty ` and the predicted label ` ty _ pred ` was calculated using the `accuracy_score ` method.

-The feature number ` i ` and accuracy `accuracy` are added to the list ` numbers ` and ` acc `.

Explain the K-nearest neighbor model:

K nearest neighbor (K-Nearest Neighbors, KNN) is an instance-based supervised learning algorithm used for classification and regression problems. In KNN, the classification of a sample is determined by calculating the labels of the nearest K neighbors to that sample.

In the given code, the K nearest neighbor model is used to train and predict the data after PCA dimension reduction, and to calculate the prediction accuracy. The final output is the number of features with the highest accuracy.

5.5. Gradient Lifting Model

The goal of the code is to use a gradient lift (Gradient Boosting) classifier to classify the data after PCA dimension reduction and to find the number of features with the highest accuracy.

Code analysis:

1) Import the required libraries:

- `import pandas as pd`: For data processing.

- `from sklearn. Model _ selection import train _ test _ split `: used for dividing the training set and the test set.

- `from sklearn.ensemble import GradientBoostingClassifier`: Gradient lift classifier model.

- `from sklearn.metrics import accuracy_score `: It is used to calculate the accuracy rate.

2) Initialize the empty list ` numbers ` and ` acc ` are used to store feature number and accuracy.

3) Number of loop iteration features ` i ` from 2 to 69:

-Extract features from the raw data and remove the target variable column (win, draw, main goal, guest goal) by ` drop ` method.

-Convert feature column names to string type to ensure consistent feature name format required by the PCA.

-Create the PCA instance ` pca ` and set the number of principal components to ` i `.

-PCA dimensionality reduction for the training data ` X ` using the ` fit _ transform` method.

-PCA dimensionality reduction for the test data ` tX ` using the ` fit _ transform` method.

-The dimension reduction dataset was divided into training and test sets, using the ` train_test_split ` method.

-Create a gradient lift classifier model instance ` model `.

-Training the model, fitting the training set data using the ` fit ` method.

-Prediction on the test set, using the `predict` method to predict the labels of the test data ` tX `.

-To calculate the prediction accuracy, and the accuracy of the actual label ` ty ` and the predicted label ` ty _ pred ` was calculated using the `accuracy_score ` method.

-The feature number ` i ` and accuracy `accuracy` are added to the list ` numbers ` and ` acc `.

In the given code, the gradient lift classifier is used to train and predict the data after PCA dimension reduction, and to calculate the prediction accuracy. The final output is the number of features with the highest accuracy.

5.6. Feedforward Neural Network

```
10/10 [==============================] - 0s 970us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 889us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 942us/step
10/10 [==============================] - 0s 902us/step
10/10 [==============================] - 0s 920us/step
10/10 [==============================] - 0s 2ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 913us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 910us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 914us/step
10/10 [==============================] - 0s 980us/step
10/10 [==============================] - 0s 991us/step
10/10 [==============================] - 0s 934us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 965us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 953us/step
10/10 [==============================] - 0s 971us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 892us/step
10/10 [==============================] - 0s 953us/step
10/10 [==============================] - 0s 988us/step
10/10 [==============================] - 0s 862us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 937us/step
10/10 [==============================] - 0s 785us/step
10/10 [==============================] - 0s 866us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 996us/step
10/10 [==============================] - 0s 983us/step
10/10 [==============================] - 0s 957us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 895us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 2ms/step
10/10 [==============================] - 0s 848us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 988us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 973us/step
10/10 [==============================] - 0s 908us/step
10/10 [==============================] - 0s 953us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 861us/step
10/10 [==============================] - 0s 890us/step
10/10 [==============================] - 0s 849us/step
10/10 [==============================] - 0s 914us/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
57
0.47157190635451507
```

Figure 12. Feature selection and classification tasks are performed using feedforward neural networks

The main function of this code is to use a feedforward neural network for feature selection and classification tasks, and to output the number of features with the highest accuracy and the highest accuracy.

Code analysis:

1) convert the data into NumPy array, and convert the feature matrix ` X_ ` and the target variable ` y ` to NumPy array for subsequent processing.

2) Initialize the empty list ` numbers ` and ` acc ` for storing feature number and accuracy.

3) for iteration from the feature number range from 2 to 69.

4) Features were extracted from the raw data and assigned to the variables ` X ` and ` tX `.

5) Convert the feature column names to a string type.

6) Use PCA dimension reduction technology to reduce the training data and test data, and reduce the dimension to the current number of features ` i `.

7) Divided the data set into the training and test sets, using the ` train_test_split ` function.

8) Create a feedforward neural network model, using ` keras. Sequential` Build the model, including two hidden layers and one output layer, with the activation functions of ReLU and softmax, respectively.

9) Compile the model, specify the optimizer as Adam, the loss function as sparse classification cross-entropy, and the evaluation index as accuracy.

10) The model was trained using the training set, assigning the number of rounds, batch size and detailed patterns.

11) With the predictions on the test set, the prediction results were obtained using the `predict` method of the model, and the ` np. The argmax` function takes the category label corresponding to the maximum probability.

12) The prediction accuracy was calculated, using the `accuracy_score ` function.

13) The feature number ` i ` and accuracy `accuracy` were added to the ` numbers ` and ` acc ` lists, respectively.

14) Output the number of features with the highest accuracy rate and the highest accuracy rate.

Feedforward neural network is a basic artificial neural network structure, information propagation unidirectional in the network, from the input layer through a series of hidden layers to the output layer. It is a fully connected neural network, and the connections between the various neurons do not form a loop. Feedforward neural networks are widely used in classification, regression, and other machine learning tasks. Its hidden layer processes the input data through a nonlinear activation function, enabling the network to learn the nonlinear patterns and complex feature representations. In this code, a feedforward neural network model is used for feature selection and classification tasks, by training the model to learn the feature representation and classification rules of the data, and to predict and calculate the accuracy on the test set

6. SUMMARY

This study first by Python tool to climb the English football league part of the odds data, then the data pretreatment, related analysis and principal component analysis, and according to the data characteristics, choose the logistics regression model modeling, get football results prediction model, finally verify the model prediction accuracy, method can be used in the prediction of football results.

Through the more easily available disk data, the result of the football match is predicted. According to the data characteristics, the Logistics regression model is selected for modeling, and finally the conclusion of the model is validated, which has good practical significance. The match data of the League 1 football match is selected, and the amount of data is relatively small. In the practical application of the model established, the prediction accuracy may fluctuate due to the problems of the match type or region. How to improve the stability of the model will be further investigated. Compared with the traditional football match analysis and prediction research, the use of football history data correlation analysis and prediction method, not only to batch data collection and processing, and can construct a reasonable characteristics of football game engineering model, to provide support for the results analysis and prediction.

REFERENCE

[1] Jiang Ting, Li Qing. Study on the influencing factors of men's singles matches in professional tennis tournament — Based on the four grand slam tournaments from 2014 to 2018 [J]. China Sports Science and Technology, 2021, 57 (7): 62-68. DOI:10.16470/j.csst.2019115.

[2] Su Yangjie. Influenced the first goal in the Chinese Super League index study [J]. Exercise and Health, 2023,2 (5): 81-83.

[3] Wu Jinlong. Prediction of football match results based on genetic algorithm and BP neural network [D]. Guangdong: Guangdong University of Technology, 2016. DOI:10.7666/d.D01243566.

[4] Huang Yi. Prediction of football match results using neural networks [J]. MicroPC, 2021,37 (11): 137-140. DOI:10.3969/j.issn.1007-757X.2021.11.040.

[5] Li Qiang. Prediction of Basketball Game Results Based on Conditional Random Field [D]. Hunan Province: Xiangtan University, 2016.

[6] Zhu Wenfu. A tentative study of the outcome prediction model of sports competition [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2011,28 (3): 318-321. DOI:10.3969/j.issn.1672-058X.2011.03.027.

[7] Shantou University. A method for basketball results based on SVM: CN201910170959.2 [P].2019-07-16.

[8] Bian Jie. Structure-adaptive neural network prediction of NBA competition outcome [J]. Wireless Internet Technology, 2018,15 (20): 107-109. DOI:10.3969/j.issn.1672-6944.2018.20.048.

[9] Zou Xin. Logistic Construction and analysis of the winning factor model for women's tennis —— Take the Australian Open as an example [J]. Sports Science and Technology Literature Bulletin, 2022,30 (3): 65-67. DOI:10.19379/j.cnki.issn.1005-0256.2022.03.017.

[10] Li Chenghui, Zhang Yushan. The long-and short-term memory network was used to predict the outcome of NBA matches [J]. Computer Applications, 2021,41 (z2): 98-102. DOI:10.11772/j.issn.1001-9081.2021010160.

[11] Wu Xingqun. Football lottery prediction method based on the Logistic regression analysis model [J]. Technology Information, 2013 (19): 198-201. DOI:10.3969/j.issn.1672-3791.2013.19.142.

[12] Zou Xin. Comprehensive strength evaluation and regression model prediction analysis of world women's hard tennis singles [J]. Sports Science and Technology Literature Bulletin, 2022,30 (1): 67-70. DOI:10.19379/j.cnki.issn.1005-0256.2022.01.021.

[13] Feng Pan. Comparative study of technical winning factors among the top 6 teams in the 2019 Women's Volleyball World Cup —— based on logistic regression model [D]. Jiangsu: Nanjing Normal University, 2020.