Data Analysis and Game Prediction Based on NBA 2020-2021 Season

Yichen Ding, Xingzhi Mei, Yunxiang Zhang, Yifeng Lu

***Abstract*—Data analysis has become a hot topic in today's society. In the aspect of data prediction, the time series method proves its effectiveness and reliability in the classification model. NBA regular season prediction is considered as a representative item of data mining and analysis. This paper mainly describes the data preprocessing, classification model selection and model optimization for the game data of the specified NBA season. At the same time, compare and evaluate the number of different windows based on the historical data obtained by time series. And select the best classification model reporting the accuracy .**

Keywords- Data Analysis, Big Data, Data Prediction, Optimization

I. Introduction to PRoblem

In general, how to efficiently predict the victory or defeat of the team is a problem we need to solve. Because the team personnel situation will be adjusted every season, it is inaccurate to predict the outcome of another season according to the data of the previous season. Therefore, this paper will roughly describe the application of data preprocessing, training and testing, visual analysis and Optimization in predicting the victory and defeat of the team. First, for the game data, we need to find a suitable website, collect and process the data of the whole 2020-2021 regular season. Through the classification model to complete the training and evaluation of data. Of course, it is also an important step to analyze the data by adjusting different parameters. Finally, the prediction results of the two teams are visualized and displayed through the interface.

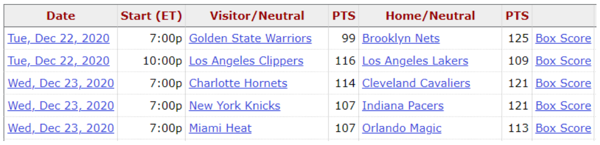
II. Proposed Solution

For this project, we need to know that the most basic goal is how to find the appropriate eigenvector. As a feature of the classification model. This problem may become a risk we need to deal with. First of all, we want to input the ranking of each team in the regular season as a feature. However, for a whole regular season, the team's ranking is constantly changing. And the final ranking can only be obtained at the end of the regular season. However, for a season's game prediction, it is unreasonable to use the final results to judge the victory or defeat of the team. Therefore, we decided to train the classifier and analyze its performance from two aspects. The first goal is to find the team's winning rate or number of wins in the past several games. The second goal is to find the average of the score gap of the team's past wins. This can help us analyze the effectiveness of the data.

III. Experiments

A. Data Loading:

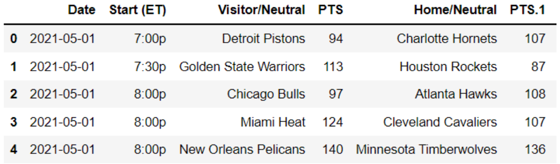
The website called ‘Basketball reference’ provides strong data support for the regular season results of NBA 2020-2021 season. The website provides the scores and results of each game in December, January, February, March, April and May of the regular season of 2020-2021. There is a separate txt file every month. Figure 1 shows the sample of each game summary.There are 30 teams in total, and each team needs 82 games in the regular season



Date	Start (ET)	Visitor/Neutral	PTS	Home/Neutral	PTS	
Tue, Dec 22, 2020	7:00p	Golden State Warriors	99	Brooklyn Nets	125	Box Score
Tue, Dec 22, 2020	10:00p	Los Angeles Clippers	116	Los Angeles Lakers	109	Box Score
Wed, Dec 23, 2020	7:00p	Charlotte Hornets	114	Cleveland Cavaliers	121	Box Score
Wed, Dec 23, 2020	7:00p	New York Knicks	107	Indiana Pacers	121	Box Score
Wed, Dec 23, 2020	7:00p	Miami Heat	107	Orlando Magic	113	Box Score

Figure 1 The example data from website [1]

By collecting and loading the five .txt files for each month using module pandas as DataFrame (using concat() in pandas to combine all files to one data frame), the total size of Dataframe for games in the regular season are (1036 rows x 10 columns). Figure 2 shows the Data Frame used in jupyter notebook.



	Date	Start (ET)	Visitor/Neutral	PTS	Home/Neutral	PTS.1
0	2021-05-01	7:00p	Detroit Pistons	94	Charlotte Hornets	107
1	2021-05-01	7:30p	Golden State Warriors	113	Houston Rockets	87
2	2021-05-01	8:00p	Chicago Bulls	97	Atlanta Hawks	108
3	2021-05-01	8:00p	Miami Heat	124	Cleveland Cavaliers	107
4	2021-05-01	8:00p	New Orleans Pelicans	140	Minnesota Timberwolves	136

Figure 2 The example data from website

B. Data Pre-processing

1) Drop The empty columns

By Observing the raw data frame, There are several columns which need to be removed from vectors. Therefore, classes named ‘Unname:6’ including the useless ‘box score’ and the class ‘Unnamed: 7’ including the useless ‘OT’ values.

2) Change Time and Date type

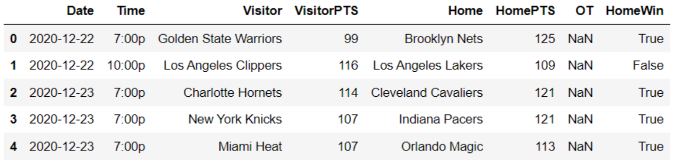
Secondly, The raw data contains the string type of games time and date. Those kinds of things need to be converted to 'date_time' type. We use the 'pandas .to_datetime()' to convert them to the correct columns.

3) Remove NaN value

As we can see from the dataset, There are OT columns which contain how many overtime games each game has. Moreover, There will be the number of watchers attending each game. However, By observing the dataset, those numbers are not relatively representative because of the covid-19 restriction. No fans are allowed to watch games. Therefore, Those NaN values can be directly removed.

4) Setting the Home Winner columns

Importantly, for the classification, the labels (called Y) are required for every eigenvector. By comparing the score of each team in each game, we can judge whether the home team wins or not. A new class' homewin 'will be added to the column .Figure 3 shows the result after adding the HomeWin column. This column is of boolean type. If the home team scores more than the away team, it returns true; otherwise, it returns false.



	Date	Time	Visitor	VisitorPTS	Home	HomePTS	OT	HomeWin
0	2020-12-22	7:00p	Golden State Warriors	99	Brooklyn Nets	125	NaN	True
1	2020-12-22	10:00p	Los Angeles Clippers	116	Los Angeles Lakers	109	NaN	False
2	2020-12-23	7:00p	Charlotte Hornets	114	Cleveland Cavaliers	121	NaN	True
3	2020-12-23	7:00p	New York Knicks	107	Indiana Pacers	121	NaN	True
4	2020-12-23	7:00p	Miami Heat	107	Orlando Magic	113	NaN	True

Figure 3 Adding the HomeWin results

C. Adding Features

For the data prediction of the winner between the two NBA teams, the primary analysis method is to use 'Time Series' to record the relationship between the past and the present. Time series data represents data in a series of specific time intervals. If we want to build sequence prediction in machine learning, we must deal with sequential data and time. Series data is a summary of sequential data. Data sorting is an important feature of sequential data [2]. Figure 4 shows the basic concept of Time Series. We iterate through the data and save a "queue" type for each game. The data structure of the priority queue is used to track the "timeline" corresponding to each team. The maximum size of the priority queue is variable. This size is an important parameter for later comparative analysis results. The python program showed in 'preprocessing.py'

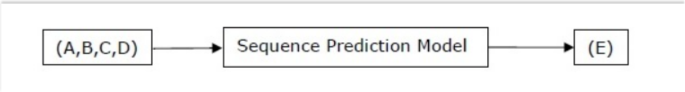
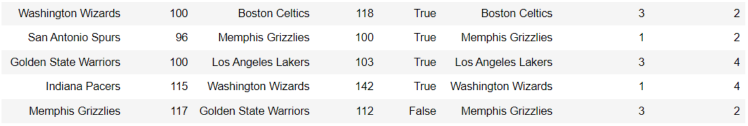


Figure 4 Basic Idea of Creating feature vectors

We choose the default past five games of each team as the eigenvalue to compare. It determines whether the home team will win by comparing the number of wins of the two teams. Then, we will visualize the result in the classification step. Figure 5 shows the numbers of the last 5 winning games for every team in each game.



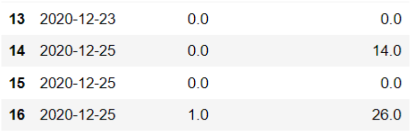
Washington Wizards	100	Boston Celtics	118	True	Boston Celtics	3	2
San Antonio Spurs	96	Memphis Grizzlies	100	True	Memphis Grizzlies	1	2
Golden State Warriors	100	Los Angeles Lakers	103	True	Los Angeles Lakers	3	4
Indiana Pacers	115	Washington Wizards	142	True	Washington Wizards	1	4
Memphis Grizzlies	117	Golden State Warriors	112	False	Memphis Grizzlies	3	2

Figure 5 numbers of Last home and visitors win

In addition, the feature which contains the average lead score for the past N games will also be added. If the game wins the three in N games. The average leading score can be written as this formula:

$$AveLead = \frac{\sum_{1}^{N}(winnerPTS - LoserPTS)}{N}$$

Thus, The two calculations for average leading score will also be counted for training. Figure 6 is the sample of saverage leading score for each team in the past 5 games.



13	2020-12-23	0.0	0.0
14	2020-12-25	0.0	14.0
15	2020-12-25	0.0	0.0
16	2020-12-25	1.0	26.0

Figure 6 Sample of average leading score

D. Data Classification

After getting the characteristics of the number of wins of the team in the past, we try to use different classification models for data classification. There are several suitable classification models: logistic regression, SVC, decision tree, Random Forest, Perceptron and K Nearest Neighbor. Visualization is carried out by analyzing different performances under different classification models. The main Python package used is scikit-Learn. The training vector that I choose contains two values: the last 5 games for home and the last 5 games for visitors.

The training set is set 0.7 and the rest of 0.3 is the validation set. By comparing each classifier with default value, we can choose 4 proper models to modify the parameters so that they can perform well. As figure 7 shows, SVC gives better performance in this problem.

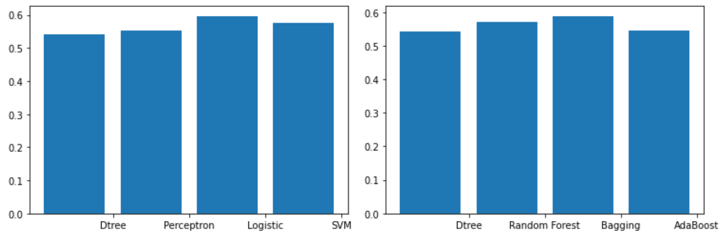


Figure 7 Classification accuracy in different classifiers

By observing the plot from each classifier. The performances of Logistic Regression, SVM and Random Forest are impressive. Here, we compare the performance of three typical classifiers under different parameters.The first classifier which is used is Decision Tree. Thus, For different max depth, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50 are tried for the best performance with highest accuracy. Figure 8 shows the result of training and testing in different max depths. We can intuitively see that when the depth is larger and larger, the training accuracy of the model is approaching 1. However, the test set has not kept up with good progress and is getting lower and lower. Their differences are becoming larger and larger, which shows that the model has been over-fitted.

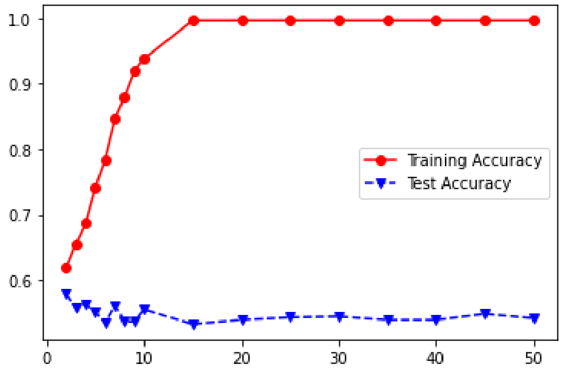


Figure 8 the decision tree used in different depths

The Decisive part for the decision tree is the effect by gini and entropy. We used both to classify the data with the constant depth 4 (best performance). Thus, the difference in accuracy is shown in here:

1. **Gini**: 0.557161629434954
2. **Entropy**: 0.5781865965834428

Moreover, The KNN and Logistic Regression are tested. For KNN, The numbers of neighbors are 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 38, 40, 45 ,50, 60, 70. Therefore, the result for the accuracy plot is shown in figure 9. The 45 are set as the best neighbors which is 0.595.

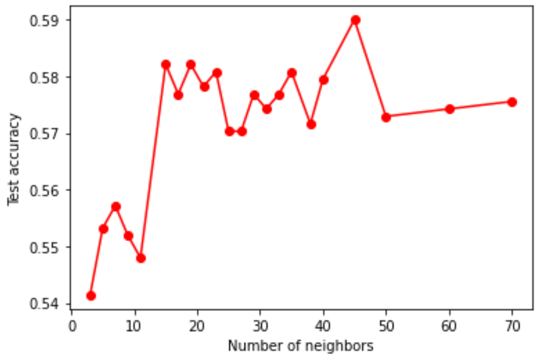


Figure 9 the accuracy in different neighbors

E. Modifying Windows

According to the requested data, the performance of different classifiers in Windows = 5 is analyzed. We also try to adjust the number of windows to observe the results verified by the classifier.

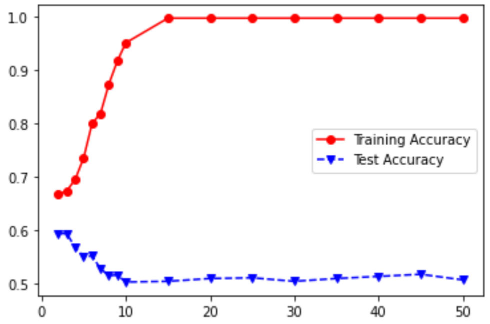


Figure 10 decision tree classifier in windows = 10

Take the example of Decision tree, we set the window as '10', the result will be different for the performance. Figure10 shows the Decision Tree classification result when the window is 10. When the max depth is the same value 4. The accuracy has been increase to:

1. **Gini**: 0.5939553219448095
2. **Entropy**: 0.5913272010512484

We used a Decision Tree with Gini input as our prediction model because it shows the best performance in the prediction for the rest of games. (The model has been saved into the 'dt.pkl' file to used in the future)

IV. Conclusion

By designing this project, our group learned the solid concept of Time Series in data processing. Also, by doing the several analysis operations.

V. Future Work

Because it is hard to catch the data from the website that contains detailed data in offensive and defensive and the details of score, rebound, and assistants for each player. Our group is willing to improve by adding more representative features into the feature vector. Also, we hope the prediction model not only just predicts the games in the regular season, but also predicts the playoffs games, even the final that gets the answer of which team will get the championship. We hope we can do it better by learning deeply in the field of big data in the future.

References

[1] Basketball-Reference: https://www.basketball-reference.com/leagues/NBA_2021_games.html

[2] Analyzing Time Series Data, Wiki Tutorial, https://iowiki.com/artificial_intelligence_with_python/artificial_intelligence_with_python_analyzing_time_series_data.html