Fan-Map*†

Group coursework for COMP6235

Junming Zhang (Leader)
University of Southampton
jzlg17@ecs.soton.ac.uk

Zezhen Zeng
University of Southampton
zz8n17@ecs.soton.ac.uk

Yinghan Peng
University of Southampton
yp4a17@ecs.soton.ac.uk

Ying Luo
University of Southampton
yl2lu17@ecs.soton.ac.uk

Yixuan Xu
University of Southampton
yx5u17@ecs.soton.ac.uk

ABSTRACT

This project is trying find the valuable NBA basketball players and draw a map to show the location of their fans. It can be used in the advertisement presenting. The company can use the result to find the appropriate spokesman to get in their advertisement when they want to enhance their popularity in some specific area.

1 INTRODUCTION

The problem this project first tries to solve is to predict the 2017-2018 NBA all-star player based on the players game data. And then the problem was changed to predict the more things in order to add more commercial value. The rookie players who have played less than three years(include three) have the potential to be the future all-star player, the rookie players who are going to be selected in the first and second rookie team in the 2017-2018 season and the players who are going to be selected in the 2017-2018 NBA all-star team. These are the three problems and these problems are chosen because the team members are all fans of basketball. So this project is kind of a combination of the commercial consideration and the interest of team members. The model is built based on neural network. The neural network has a high accuracy of classification and strong parallel distribution processing ability. And it has a strong robustness and fault tolerance to noise nerves. The neural network can handle the nonlinear relationship well. After the prediction finished, it is going to be a web application to show the result. The data visualization contains the fan map of the players the model predicts, five-star figure about the game data of those players. The reason that this type was chosen is that map is an appropriate way to compare the number of fans in different areas, and it will contain the American map and the World map. And the five-star figure is a straight way to show the strength and the weakness of the player. And it will help the audience to know why this player is valuable.

2 IMPLEMTATION

2.1 Data collection

2.1.1 Statistics of NBA players. Two datasets are collected in NBA official website, first is the statistics result of rookies from 2004 to 2016. It contains 25 different technical statistics of these rookies which are shown in Fig 1. The explanation of different technical statistics in the first dataset Second is the full technical statistics of all players from 2000 to 2017, it contains 173 different technical statistics. It will be used to predict the all-star players in 2018.

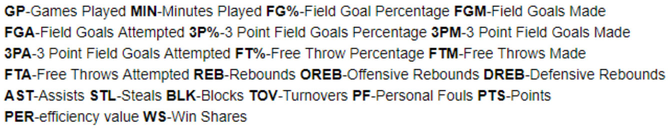

GP-Games Played **MIN**-Minutes Played **FG%**-Field Goal Percentage **FGM**-Field Goals Made
FGA-Field Goals Attempted **3P%**-3 Point Field Goals Percentage **3PM**-3 Point Field Goals Made
3PA-3 Point Field Goals Attempted **FT%**-Free Throw Percentage **FTM**-Free Throws Made
FTA-Free Throws Attempted **REB**-Rebounds **OREB**-Offensive Rebounds **DREB**-Defensive Rebounds
AST-Assists **STL**-Steals **BLK**-Blocks **TOV**-Turnovers **PF**-Personal Fouls **PTS**-Points
PER-efficiency value **WS**-Win Shares

Figure 1: The explanation of different technical statistics in the first dataset

2.1.2 Location information of twitter followers. To plot the fan map, the location information is needed, the API of Twitter, Facebook and Instagram are researched in this project. Facebook and Instagram banned their API to prevent third part obtain information of users. Therefore, Twitter is the only source used in this project to plot the fan map. twitteR package is used to grab the location information of followers, it provides access to the Twitter API by R [4]. The relevant code is shown in folder R in GitHub, it requires the developer account of twitter and the twitter account of target players. At the beginning of this project, the fans locations information of all rookies from 2015 and 2016 are obtained.

2.2 Prediction

2.2.1 PCA(Principal components analysis). The propose of PCA is transferred high dimensional data to independent low dimensional data with a high eigenvalue of data. The process of it is applying mean centering in original data in each attribute and using the eigenvalue decomposition of data covariance of the data matrix [1, 7]. It could help neural network to obtain the relationship between data with a new character. The algorithm of PCA:

(1) Mean centering: It uses the element of each column to subtract the mean value of each column.
(2) Calculating the covariance matrix $C = \frac{1}{m} X X^T$, which m is the dimensional of original data.
(3) Calculating the eigenvalue of the covariance matrix and corresponding eigenvector.

[*] GitHub Link: https://github.com/promer94/Fan-Map
[†] Website link:https://d2v4olxsjbfep7.cloudfront.net/

(4) Sorting these eigenvalues and choose reasonable number K of eigenvalue to construct new matrix P. K can be calculated by $\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{j=1}^{n}\lambda_j}$.

(5) Finding the dimensionality reduction result $A^{'} = PX$

The relevant code is shown in folder PCA of neural network in GitHub. For the statistics result of rookies from 2004 to 2016, the new matrix become $A^{'}_{364*6} = P_{364*25} * X_{25*6}$. For the full technical statistics of all players from 2000 to 2017, the new matrix becomes $A^{'}_{6850*48} = P_{6850*173} * X_{173*48}$. The visual comparison of normal players and good players can be observed in Fig 2 and 3 by desires the original data to 2 dimensional and those players who became an all-star or first rookies are labeled.

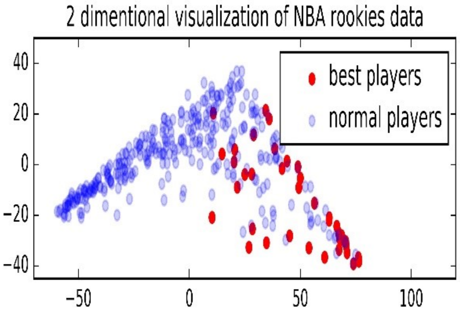


Figure 2: PCA visualization of dataset 1

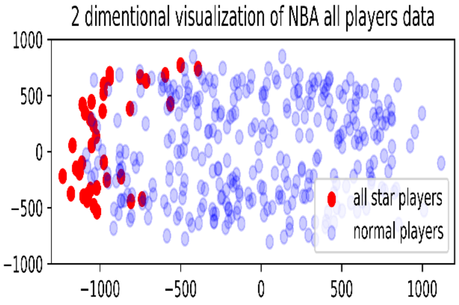


Figure 3: PCA visualization of dataset 2

It could find both all-star player and first rookie player are not linearly separable, therefore, neural network is needed.

2.2.2 Neural network training. Keras is a high-level neural networks API which using TensorFlow as a backend, it provides a convenient frame to build and run neural networks [3]. The basic structure of the Keras network is shown in Fig 4. It could find it consist of the input layer, dense layer, dropout layer and relevant configuration.

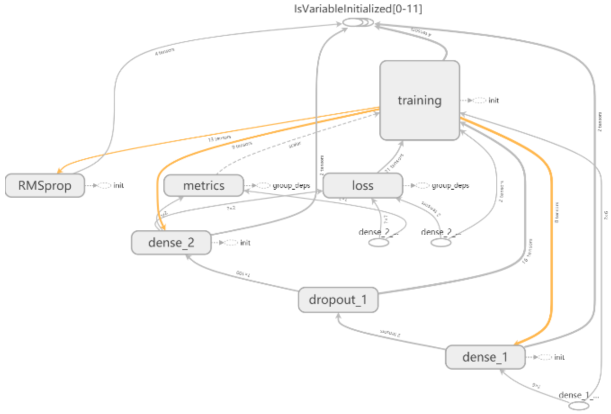


Figure 4: The basic structural of Keras network

Here are two models in Keras, Sequential model and the model class used with functional API. In this project, it uses neural network to differentiate the normal and potential rookies in NBA which is a single output classification, therefore the Sequential model is chosen for starting network building. The dense is the full connection layer, it could represent the input layer as different weights and work with activations. The function of activations is adding non- linear parameter to dense, otherwise, the output of each layer will be linear which means it cannot model non-linear problems. Suppose the layer has a weight matrix w, a bias vector b and an activation, the formula of dense can be represented as output = activation(dot(input,kernel) + bias).

Sigmoid, Tanh, ReLU and softmax are 4 frequently-used activation functions and they have a different applicative scene.

(1) Sigmoid

The formula of sigmoid is $f(z) = \frac{1}{1+exp(-z)}$ and the curve of it is shown below

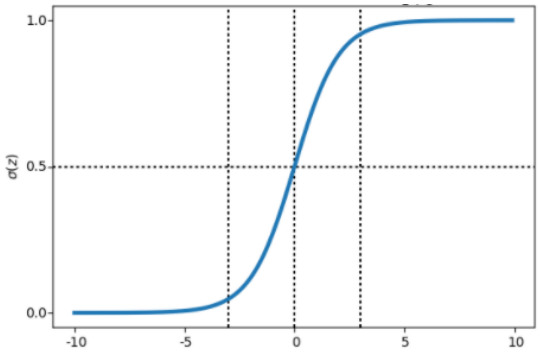


Figure 5: The curve of sigmoid function

The function of sigmoid is the output of neurones in the hidden layer, it could map a real number in (0,1) which means it is suited for two-class classification. The advantage of it is working well when the complexity of eigenvalue is high but the calculation of sigmoid is huge and the vanishing gradient will occur when calculating the backpropagation.

(2) Tanh

The formula of tanh is $f(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ and the curve of it is shown in Fig 6.

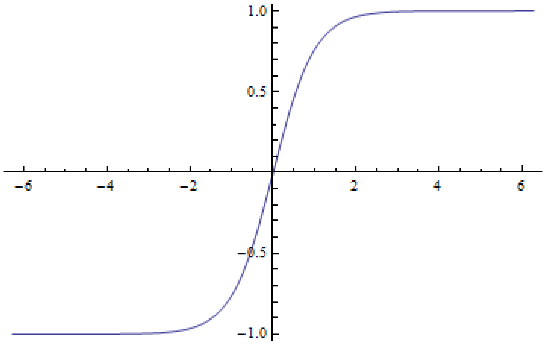


Figure 6: The curve of tanh function

The function of tanh is similar to the sigmoid function but the value range in tanh is [-1,1]. Another difference is the mean value of tanh is 0 which means it has a better application environment.

(3) ReLU

The formula of Rectified Linear Unit(ReLU) is $\Phi(x) = max(0, x)$

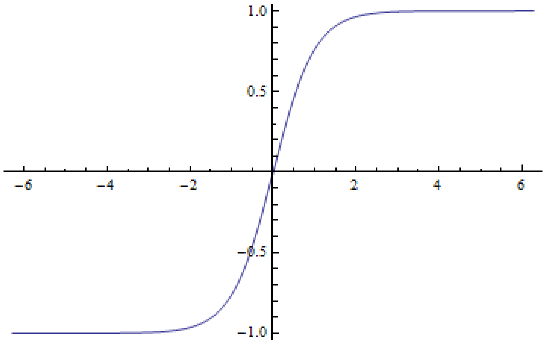


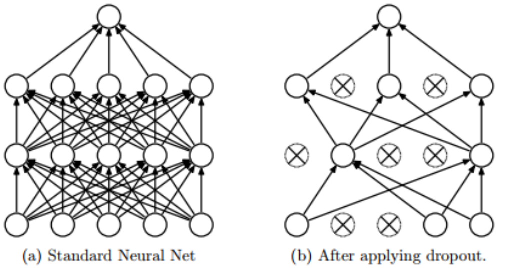
Figure 7: The curve of ReLU function

Compare to sigmoid and tanh, ReLU has around 6 times convergence speed because it no need to do complicated exponent arithmetic [6]. In addition, it could reduce the vanishing gradient. However, part of neurons will not update with tanning when it reaches hard saturation. Therefore, the number of learning rate will influence the ReLU effectiveness.

(4) Softmax

The formula of softmax is $\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_{k'}}}$, the function of it is amplifier the difference between different label, therefore it could be applied in multi-classification problems. Softmax usually set as the activation function of the last layer because it could amplifier the difference, the choice of the first layer will base on the training set. In this project, sigmoid, tanh and ReLU will be tested and the comparison of them will be analyzed in Tensorboard which is a visualizing tool in TensorFlow. It could be called by call back function.

The theory of dropout is set a probability to disconnect the input neuron, it is used for preventing neural networks overfitting because the overfitting will slow the network and influence the predicted result [5]. Therefore, the dropout layer is a regularization technique.


(a) Standard Neural Net

(b) After applying dropout.

Figure 8: The example of dropout layer [5]

Loss function and Optimizers are two main parameters in Keras. There are lots of different types of loss function and optimizers. To increase the training speed of neural network, the choice of these two parameters is significant. For loss function, categorical cross entropy loss function is used to replace the normal mean squared loss function because they have similar character but the learning speed of cross entropy loss function is higher than mean squared. This loss function is suitable for two-class classification, for the multi-classification problem, the sparse categorical cross entropy is better because it could process one hot vector because the different label will be transferred to a sparse matrix. RMSProp (Root Mean Square Propagation) is an unpublished, adaptive learning rate method proposed by Geoff Hinton in Lecture 6e of his Coursera Class [8]. The learning rate will change based on the different parameters which mean it has high speed, in addition, RMSProp has high robustness because the learning rate is hard to convergence. Therefore, the RMSProp is a better choice in optimizers and this conclusion is also found in Ruders paper [6]. In this case, the input layer will be the data after PCA processing and the training set will be shuffled randomly because the input data is sorted. In the training part, the number of validation, epochs and batch size will also influence the speed and accuracy of neural network. Validation is the float between 0 to 1, it will split test set from tanning set. Batch size is the number of samples in each batch when processing gradient descent and the epochs are the rounds of tanning. To prevent train same dataset in each epoch, the shuffle function is set to True to disorganize the sort in each epoch. Tensorboard will recode the choose of different epochs and previous parameters and the accuracy and loss curve will be analyzed to find a better neural network for picking NBA potential rookies.

2.2.3 Fan Map plotting. The collected fans locations of all rookies from 2015 to 2017 are obtained in the data collection part, to statistic number of fans in different countries and states in the USA, a list contains countries and corresponding cities are used to classify and sort the location information. There is a different format of location information, abbreviation and full name of different countries and cities excited, therefore, a list contains the abbreviation name and corresponding name of different countries and cities are collected. A full list with the full name of states and countries of players fans can be obtained by filtering the original data by these two lists. This list is a processed dataset which from the full list of US cities, states and counties. The cleaning process can be checked in city-state-data file in Data folder of GitHub. Tableau is the map generator tool in this project, it could recognize the full name of countries and cities in the processed data. In addition, Tableau will classify the cities into corresponding states in the USA, therefore, the USA fan-map and world fan-map can be plotted.

3 RESULT AND ANALYSIS

3.1 2017-18 all-star players prediction

The first model is to predict who can be the 2017-2018 NBA All-Star Player. And the training data is the game data of all the players in the last 16 years and the All-Star Player in that year are labelled as 1. There are 173 features for each player. Such as point, assist, rebound, block, time, turnover and many advanced data which are got by some formula. After Principal Component Analysis (PCA), it decreased into 48 features. Then compare the result with the real-time All-Star vote ranking. There is something interesting that the model with lower-dimensional input (After PCA) predict is worse than the model with the original input. It is due to the number of eigenvalues change a lot. But the amount of data is not very large, there are only 450 players in NBA. It cannot help a lot and may cause overfitting after the PCA. The model with the original features can do better and the result is pretty similar to the real-time result. Because the All-Star Players are popular everywhere, there is no need to draw their fan map.

Prediction of all star players	
Anthony Davis	Blake Griffin
Chris Paul	Damian Lillard
DeMarcus Cousins	Giannis Antetokounmpo
Jimmy Butler	Joel Embiid
Kyrie Irving	LaMarcus Aldridge
Russell Westbrook	Stephen Currys
Bradley Beal	DeMar DeRozan
James Harden	Kristaps Porzingis
LeBron James	Victor Oladipos

Table 1: Prediction list

3.2 2015-2017 potential all-star in rookies prediction

The second model is to predict the rookies who have the potential to be a future all-star player. As aforementioned, the input of this model is the original game data. The training data is the game data of all the players in their rookie year. The player who has been selected as NBA All-Star Player was labelled as 1. And due to the feature of the neural network, the learning process cannot be seen and the result cannot be explained. So the reliability of this result maybe not very high to those audiences who do not watch the basketball game. The things can be seen are the loss and the accuracy. These two figures are the features of the neural network model in different times of iteration. The activation function of this model is sigmoid, which one is found the best one after the implementation. And the optimizer of this model is RMSprop. It is obvious that the pink line in Fig 9 has the highest value than others and its loss value is the second smallest in Fig 10. The iteration time of the pink line is 35 so the best iteration time is 35 of this neural network model.

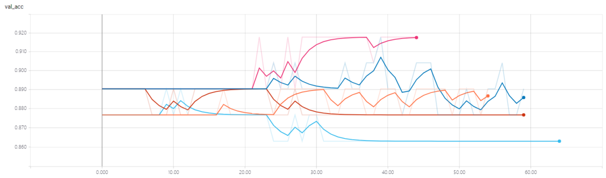


Figure 9: The correlation of the accuracy of validation set and the iterations

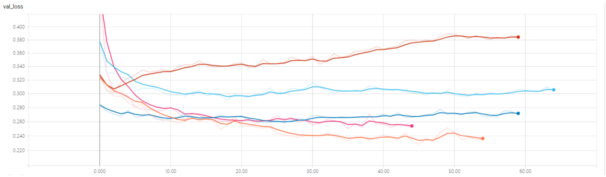


Figure 10: The correlation of the loss of validation set and the iterations

And the result of this model is interesting. There are three names in the result, Karl-Anthony Towns, Kristaps Porzingis, Willie Trill Cauley-Stein. And these three people are both picked from the same year 2015. Which means in this model, there will be no all-star players come from the 2016 NBA draft. These two situations are all existing in the last years. And these three people are both picked in the ten overall pick. Therefore, the rookie who is picked in the front has the higher probability to be a future NBA All-Star player.

3.3 2017-18 First rookies team prediction

The third model is to predict the rookies who were picked in the 2017 draft have the potential to be selected in the 2017-2018 first and second rookie team. It is a kind of three-way classification. The rookie in the first team is labelled as 1 and in the second team is labelled as 2. And the input data is the same as the second model, the only thing changed is the label. As aforementioned, the learning process of this model cannot be seen. But the loss and the accuracy can be shown in the picture. The same as the previous one, when the iteration time is 35, this model can get a good result.

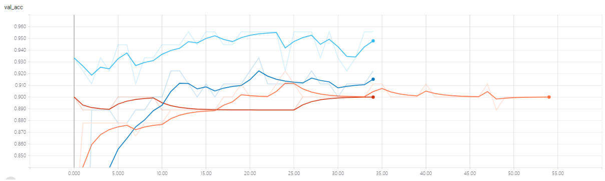


Figure 11: The correlation of the accuracy of validation set and the iterations

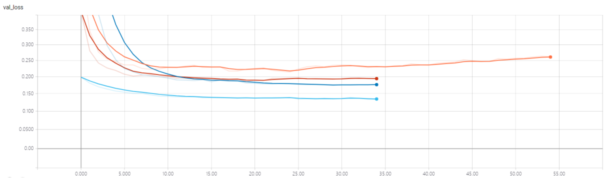


Figure 12: The correlation of the loss of validation set and the iterations

There are only seven names showed in the result. In fact, it will be ten players in the reality. Because the league will select the better one even all of the rookies are not good enough, while the model will only select the good one. Thus, seven names are acceptable. Fig 13 and Fig 14 are fan-maps of Donovan Mitchell, which is an example of the data visualization. And for the figure 13, it is obvious that Donovan Mitchell is very popular in Texas. Which means the local company in Texas can select Donovan Mitchell as their spokesman at a lower price because Donovan Mitchell is still a rookie and cannot get a large business contract. And the local company can get good profit because Donovan Mitchell is one of the future stars of the NBA league depends on the prediction. It is also applicable to the company which wants to popularize their products in Texas. Generally speaking, the location with the highest proportion in the American map is where the rookies team belong to. And it will be different in the second highest proportion or in the world map.

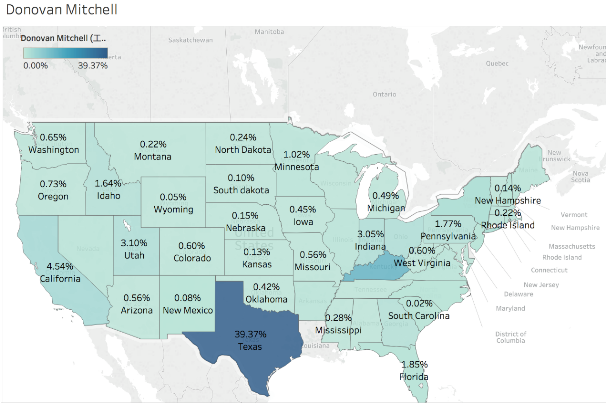


Figure 13: The fan-map (America)

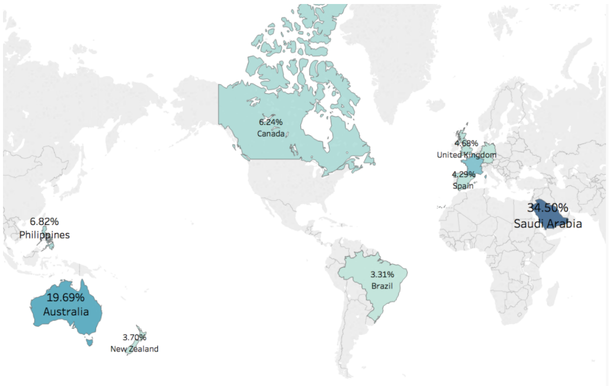


Figure 14: The fan-map (without America)

And the Fig 15 is the five-star chart of players basic game data. It is obvious that Donovan Mitchell is better than the average.

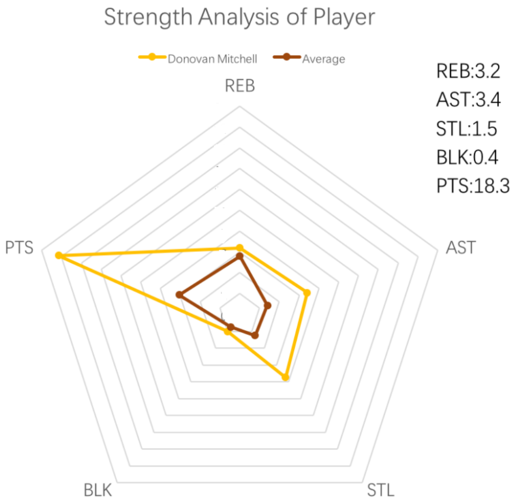


Figure 15: The five-star chart

4 DISCUSSION

4.1 limitaions and future improvements

There are three major limitations about our application.

- The size and quality of datasets.

In this coursework, only the past ten years data were collected for training the neural network because all the data need to be labeled by hand. With about 1000 of the sample, we got a model which could have above 90 percent accuracy for binary classification problems. But it could not provide a stable performance in three-class classification problems. What's more, the collected datasets only contain the basic stats. Lots of advanced data were ignored by us. It can be improved by using MongoDB to store all the player's data. With the help of MongoDB Python API, it would be easier to label the datasets. What's more, we could also customize the dataset by ourself for different purposes.

- The structure of the neural network.

Keras is a really powerful framework for deep learning that can help people without programming experience to build a neural network. All the important details about training a network have been well designed into a user-friendly APIs. It can easy set up a sophisticated network by few line of codes. However, the structure of the neural network in this project is just a simple feedforward network because of our limited knowledge of math and more complicated neural networks. We might need more time to have a better understand how to design a batter structure for different requirements.

- The Static webpage

The webpage we built for visualization of results is just a static page with limited interactions which is less attractive for our potential client. For the frontend visualization, we could use D3.js to create a dynamic and interactive graph to present the result with supporting by MongoDB and RESTful APIs.

4.2 New knowledges

It is really important to review what you have learned from the projects. We would like to go through some new knowledge acquired from the project.

- Docker and virtual machine
  The first of step of doing this project is setting up machines. In this project, A public docker image called deepo was used to quickly set up the virtual machine provided by the university. This makes sure every team member could test our code correctly.
- AWS cloud service
  For the result presentation, we used AWS cloud service to host our website. We also used AWS ES2 instance to train our neural network

5 CONCLUSION

The NBA 2017-2018 All-Star Player the model predicted is similar to last year. But there are still some players who have never been selected as an All-Star Player are predicted to be an all-star this season. And there are only three players who join the NBA league after 2015 have a chance to become an All-Star Player according to the model, and the three players are all from top NBA picks. And for the rookie team of this year, there are seven players are predicted to be selected. And there is only one player who is picked in the second round. Which means the player who is picked in the first round is actually more potent than the player selected in the second round. All of these three predictions can help people to find the potential commercial value before the player become a real superstar. And the fan-map can help people to know the player is popular in which place. The basketball players do not popular in all over the world, maybe they are popular in America. Combined these two results can help the company to find who is the appropriate spokesman for their product and where can the spokesman go to hold a promotion activity. And in the future, when this model is more complete, it can also help the NBA teams to find who is a potential player and help them to make some decisions for the player trading.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[2] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. 2015. ImageNet Classification with Deep Convolutional Neural Networks. (2015). http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

[3] Jason Brownlee. 2016. Introduction to Python Deep Learning with Keras. (2016). https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/

[4] Jeff Gentry. 2014. Twitter clinet for R. (2014). http://geoffjentry.hexdump.org/twitteR.pdf

[5] Alex Krizhevsky Ilya Sutskever Ruslan Salakhutdinov Nitish Srivastava, Geoffrey Hinton. 2014. A Simple Way to Prevent Neural Networks from Overfitting. (2014). http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

[6] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[7] Stanford. 2017. PCA. (2017). http://deeplearning.stanford.edu/wiki/index.php/PCA

[8] Tijmen Tieleman and Geoffrey Hinton. 2012. Neural Networks for Machine Learning. (2012). Retrieved Jan 2, 2018 from http://video.google.com/videoplay?docid=6528042696351994555