Dean&Francis

ISSN 2959-6130

Application of different Model Algorithm in the Prediction of Transfer Fee of Soccer Players

Keying Chen[1*],

Letian Wu[2],

Ziyang Xu[3],

Zheye Liu[4]

[1]Jurong Country Garden School, Nanjing, 212446, China, chenkeying0522@gmail.com
[2]Tsinglan School, Dongguan, 523830, China, tim.wu_25@tsinglan.org
[3]Soochow Foreign Language School, Suzhou, 215131, China, stevenxziyang@163.com
[4]Jingling High School Hexi Campus, Nanjing, 210019, China, 18851068385@163.com

[*]Corresponding author email: chenkeying0522@gmail.com

Abstract:

Soccer is popular worldwide, and the fees for transferring soccer players show how much a player is worth and give us an idea of how well a country's soccer is developing and how a team is being managed. This study aims to investigate the application of different model algorithms in the prediction of the transfer fee of soccer players and find which model is the most accurate. The dataset used for this research is available on the Kaggle website, from the football player's transfer fee prediction dataset. By analyzing the (players' codes and names) football team, position, height, age, the appearance of a player, the number of goals, assists, yellow cards, second yellow cards, red cards, goals conceded, clean sheets, minutes played, days-injured, games-injured, award, current-value, highest-value and position-encoded. Machine learning is commonly used in diverse fields to solve difficult problems that cannot be readily solved in based on computer approaches [1]. This study compares the accuracy of different machine learning algorithms used for predictive analysis of soccer players' transfer fees. For example: Linear regression, Random Forest, Decision tree, K-Neighbors, and Neural network. When finding the relationship between those 21 factors, could help players and teams to make valuable decisions and accurate prediction for the establishment of soccer player market.

Keywords: Machine learning algorithms, Data Analysis, football player's transfer fee

1 INTRODUCTION

Over the last two decades, the economic dynamics have completely been changed, and the amount of money which has been pumped has increased exponentially in player market [2]. Player transfer fees have been going up, with new records being set. One notable record was Neymar's transfer from Barcelona to Paris Saint-Germain for 263 million dollars. Valuing workers has long been a topic of interest in

the literature on labor economics and the earliest work on explaining transfer fees in football originated here. Beginning with Carmichael & Thomas (1993), economists have used regression models to identify determinants of transfer fees [3]. It is no doubt that through the development of football match and industries, such as the popularity of the World Cup, improved and advanced commercial strategies are very needed to catch up with the contemporary situation in the player market. To help players and clubs avoid property damage, machine learning techniques could be used to predict player transfer fees to help them make valuable decisions. However, since the player data is too large for humans to handle, machine-learning data processing techniques can be implemented. First, the work will use different methods and techniques to predict the data. The project will build a Linear regression model, Random Forest, Decision tree, K-Neighbors, and Neuron network to predict the transfer fees of soccer players. Then, blend the images and compare which one is more accurate.

During the transfer window, teams may buy and sell thousands of players, spending millions of dollars, with some top players receiving transfer fees in the hundreds of millions of dollars. In the past McHale & Holmes's (2023) paper, articles often used basic performance metrics, such as goals scored and minutes played to model transfer fees [3]. However, these metrics might not capture the nuances of a player's contribution to the team, particularly for players in different positions. Therefor in this article, the 19 factors that affect a player's value will be more comprehensive. The player price prediction model includes 19 influencing factors and tens of thousands of players. The data analysis is done through Google Colab using Python, a powerful data analysis application software.

2 DATA PRE-PROCESSING

The research imported libraries include Pandas, Numpy, Matplotlib, Seaborn, and Scikit-learn.

Data Preprocessing helps to build up an accurate Machine Learning model. Data pre-processing is the process of transforming the data. It will remove all the NAN, NULL values from our data and bring the data into a format where a machine can parse it easily. This process is also called Data Wrangling. This includes the identification of missing data, noisy data, inconsistent data, and null values.

This process includes instructions: Displaying the first few rows of the dataset, checking five random samples, Check for the shape, Check for missing value, Check for unique value, Check for datatype of the columns, and Getting the statistical summary for the dataset.

Here are 21 parameters. 'Players' code' and 'Name' were firstly dropped, as it displayed duplicated and useless information (certainly they are only used to identify every player).

Table 1: all 21 features(columns) uploaded from the data.

Features	Data description
Player's code	identify different players
Team	the team of the players
Name	name of player
Position	the character of the players in their team
Height	physical height of players
Age	age of players
Appearance	The number of times a player appears on field
Goals	goals scored by the player
Assists	assists to goals
Yellow cards	average number of receiving 1 yellow card per match
Second yellow cards	average number of receiving 2 yellow cards per match
Red cards	average number of receiving 1 red card per match
Goals conceded	average number of goals saved per match
Clean sheets	possibility of not being goaled in 1 match
Minutes played	minutes(time) stayed on field
Days injured	absent days
Game injured	times getting injured during games

ISSN 2959-6130

Award	total award in whole career
Highest value	Highest valuated price in the past
Winger	Is the player a winger
Current value	Valuated price for now (Euro unit)

2.1 Exploratory data analysis (EDA)

Displaying the distribution of the current value from the data can exhibit the shape of the current transfer market. Figure 1 illustrates that the majority of players have current values concentrated at the lower end of the spectrum, with a few outliers possessing significantly higher values. This skewed distribution is critical for the prediction of current value, as it suggests the need for specific data preprocessing and outlier handling techniques. Addressing these aspects will help improve the model's accuracy, ensuring it can effectively predict both the common lower values and the less frequent, higher value

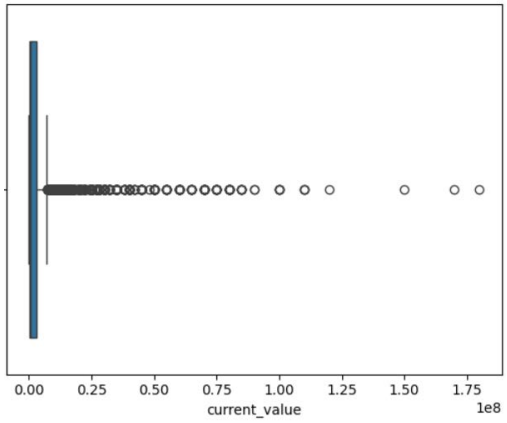

Figure 1: the box-plot chart of the distribution of different players with their current value

The position is divided into four different sectors: DE for defenders, GK for goalkeepers, MI for midfielders, and ST for strikers. The histogram and pie chart below show the distribution of player positions. It is important because each position has distinct roles, skills, and physical attributes that can significantly influence a player's performance metrics, market value, and career trajectory. By categorizing them separately, we can better analyze and understand the specific factors that impact players in each role, leading to more accurate and tailored insights, particularly when predicting current value or other key performance indicators. Corresponding to Figure 2 and Figure 3, defenders are the most common position, followed by midfielders and strikers, with goalkeepers being the least common. This distribution may suggest that teams typically have more players available in defensive and midfield roles, reflecting the need for greater depth in these areas. The relatively lower number of goalkeepers aligns with the fact that each team usually has fewer goalkeepers compared to outfield players. This distribution should be considered when analyzing player performance and predicting current value, as the abundance or scarcity of players in each position could impact their market value.

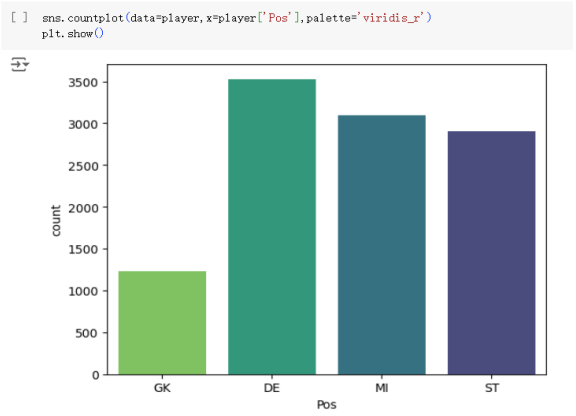

Figure 2: number of each position in the data set

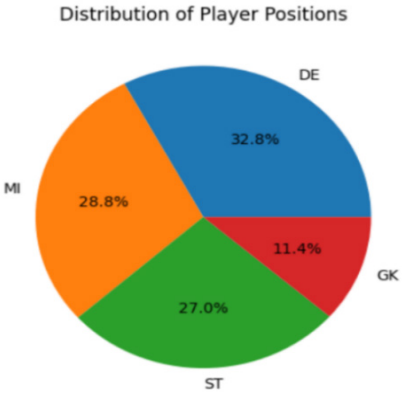

Figure 3: the percentage of each position take up in the data set

Age is another significant factor affecting current value; therefore, the following section presents the relationship between age and current value. Figure 4 and Figure 5 illustrate the distribution of players' ages, with the majority falling between 20 and 30 years old. This age range is

likely where the highest market values are concentrated, as players typically reach their peak performance during these years. Understanding this distribution is crucial for modeling how age influences current value, particularly in identifying the period when players are most valuable.

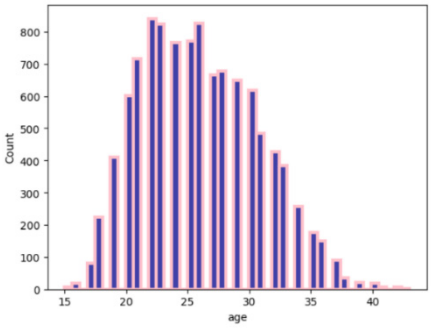

Figure 4: the population of different age groups

Figure 5: the current value for players in different ages

2.2 Dropping abnormal parameters

Figure 6 illustrates and highlights the relationship between all features and current value. There are 19 factors we have. First 'current value' is dropped as it will be meaningless to discuss the correlation with itself. Then we drop 'team_encoded', 'winger' and 'position_encoded' first, as objective factors they obviously have a very insignificant effect on the current value. However, since 'second yellow cards', 'yellow cards' and 'red cards' are also extreme irrelative, we tend to believe that foul markers are considered negatively in an inverse relation. As for height and age, no doubt that they are both in a normal distribution instead of a standard correlation form, it would be no surprise that the correlation index is relatively low.

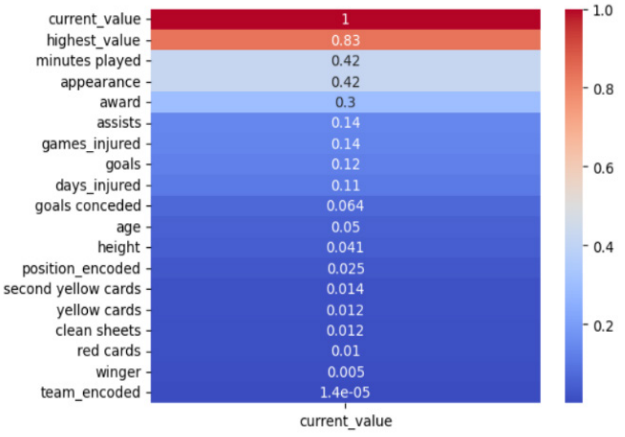

Figure 6: the correlation diagram for each parameter against current value

The following section presents a heatmap, a visual representation that stimulates the correlation between variables. In this heatmap, color intensity indicates the strength of the correlation between different features and the current value, making it easier to identify patterns and relationships at a glance. As shown in Figure 7, the pairs of dark-red parts are 'minutes played' against 'appearance' and 'games_injured' against 'days_injured' relatively. They are 95% correlated. We needed to abandon one parameter from both pairs, otherwise the parameters would be too similar to be meaningless. We keep 'appearance' and 'games_injured' at last.

ISSN 2959-6130

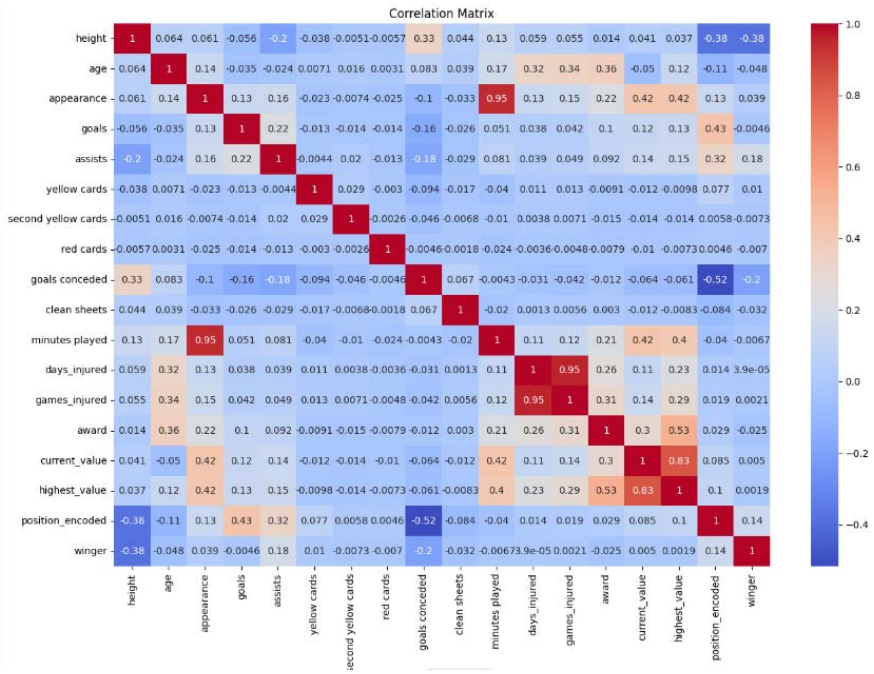

Figure 7: the correlation diagram for each parameter against each parameter

This means until now we had dropped 7 irrelative or meaningless parameters. The parameters left are 'height', 'yellow cards', 'second yellow cards', 'red cards', 'goals conceded', 'clean sheets', 'highest_value', 'age', 'award', 'goals', 'assists', 'appearance' and 'games_injured' at last.

3 IMPLEMENTATION OF CLASSIFICATION ALGORITHMS

Five typical models are picked to take part in the test. They are set to analyze the data from different aspects, in terms of numerical regression, possibilities chain, and correlation classification. It would be necessary to see which type(s) of algorithms will best fit the given data set, to decide the entry point for predictions.

3.1 Preparation for the variables

We separated the columns into independent and dependent variables. The independent variables include all 13 distinguished indexes filtrated by EDA. As shown in Figure 8, Together they were defined as 'Player_Features'to represent Xs. Relatively, column 'current value' (the name in the original data set) was defined as 'Player_Pay' to represent the Y. Due to the huge difference of absolute values between the statistics, all the numbers in the columns are normalized to make sure the result is reasonable

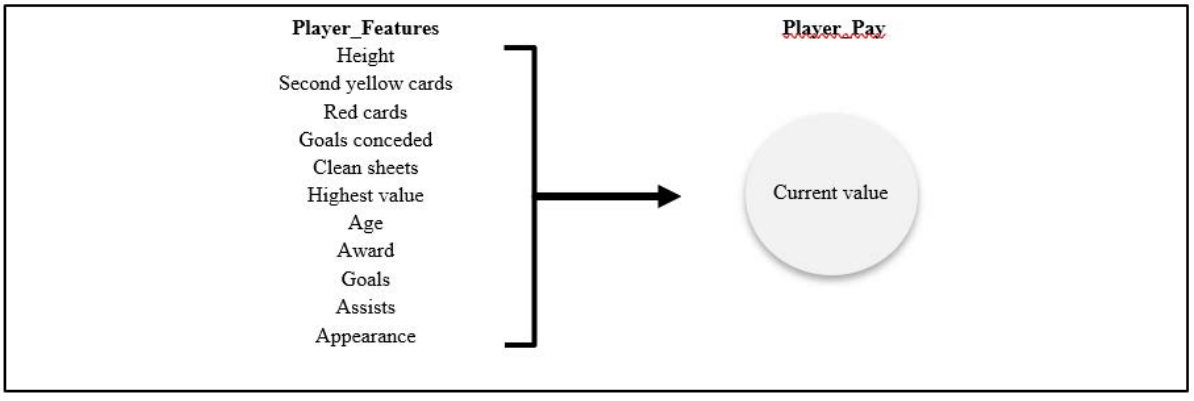

Figure 8: the connection between the independent (Play_Features) and dependent (Player_ Pay) parameters

3.2 Evaluating the models

The use of control variables plays a central role in organizational research due to practical difficulties associated with the implementation of experimental and quasi-experimental designs [4]. Suggest the hyperparameters as the variables, making sure that all the model approximately reach the best result, in order to make a more reasonable comparison.

In order to find the best fitted settings of hyperparameters loaded for every model, we placed different groups of hyperparameters in the relative reasonable way. We rely on the AI algorithm to help us define the best numerical gap for each hyperparameter. For each model the hyperparameters might be different, for example, in Linear regression, 'test_size' and 'random_state' are considered, instead, in Random Forest, 'n_estimator' was needed in addition but 'test_size'. The best final result (output accuracy) for the algorithms will be recorded in terms of R-squared(R^2); higher R^2 value represent high accuracy of the mathematical prediction. We also set a column named 'average R-squared', which is to compare the effectiveness by two different hyperparameters on the R-squared separately.

3.2 1 Linear Regression Analysis

Regression analysis is the process of constructing a mathematical function that has the best fit to a series of data points according to some criterion [2]. We choose linear regression first instead of multiple regression is because that we need to get meaningful information through the test of linearity between each pair of X against Y discretely. According to the visual result of the regression model (Figure 9), it is clear to identify that some parameters show an inversely proportional relationship against the current value. Take parameter 6, 7, 8 (yellow cards, second yellow cards, red cards) as examples, as long as they belong to foul markers, increase in these indexes would cause they players' value to decrease. We can simply define this as a negative performance index for players.

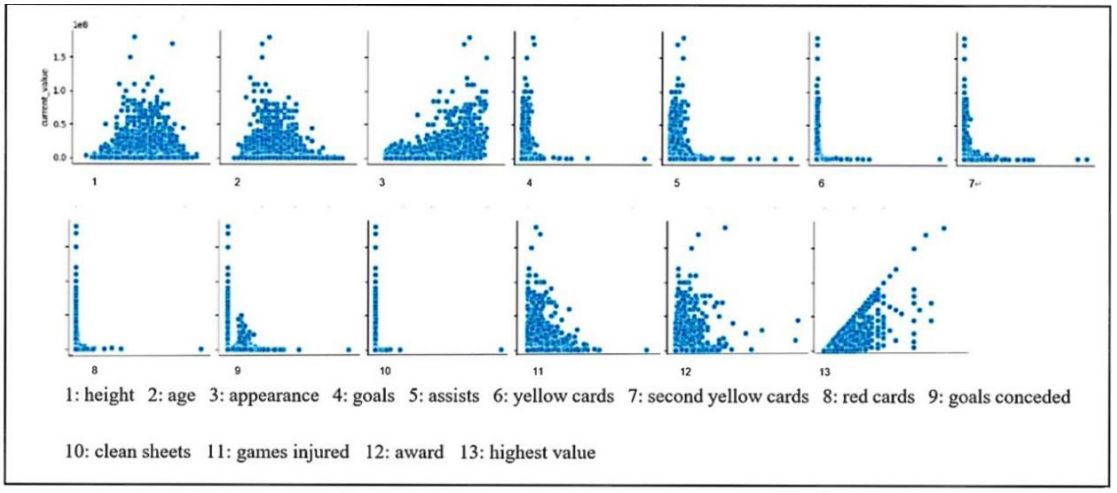

Figure 9: the visualization of separate linear regression in terms of each X (player features) against Y (player pay)

After the classification of the linearity of the parameters, the experiment moved into the process of examination part. For regression model, hyperparameter 'test_size' is defined as 0.20, 0.25 and 0.30. Relatively, 'random_state' for each value of 'test_size' is defined as 41, 43 and 45. They belong to the better presenter in AI test, we managed to find some specific distribution as we list the numbers as best as we can in arithmetic progression. There are 9 groups of hyperparameters are evaluated as shown in Table 1.

Table 2: the final result of linear regression model recorded in terms of R-squared and average R-squared

Hyperparameter(s)		R-squared	Average R-squared
test_size	random_state		
0.20	41	0.7787954997406765	0.7484471267976006
	43	0.6834347269663916	
	45	0.7831111509882093	

ISSN 2959-6130

0.25	41	0.7649090334566191	0.7307671504454346
	43	0.6492225324685577	
	45	0.7781698854120825	
0.30	41	0.7766935967947536	0.7361590935843169
	43	0.6758195004853937	
	45	0.7559641834728033	

The best result is marked with darker area, as shown in Tabel 2, 0.7831111509882093, reserved to 4 significant figures, 0.7831, which is the approximate result of linear regression model.

3.2 2 K-Nearest Neighbors

In those cases where this information is not present, many algorithms make use of distance or similarity among samples as a means of classification. The K-nearest neighbor (Knn) decision rule has often been used in these pattern recognition problems [5]. Depend on different parameters to consider for the value of a soccer player, Knn would be a very simple model to predict with uncompleted data set, it is for sure that despite the prediction won't be extremely accurate, but could be the most efficient one.

There are two main hyperparameters for Knn model, 'n_neighbors' and the p-value. 'n_neighbors' is given value as 1, 3, 5, 10, 20. We would like to see if the number of neighbors would cause a large effect in the classification of the unknown index in terms of soccer market. Therefor the gap is also large, where we hope to get more significant result. In Knn, a database is searched for the most similar elements to a given query element, with similarity defined by a distance function [6]. The 'p' is defined only in quantity 1 and 2, which represent Manhattan distance and Euclidean distance. The reason is that the accuracy of Knn model is dependent on the magnitude of p-value in a large extent. Usually, smaller the p-value is, higher the accuracy. We believe the two pre-defined area would lead to better result.

Table 3: the final result of K-nearest neighbor recorded in terms of R-squared and average R-squared

Hyperparameter(s)		R-squared	Average R-squared
n_neighbors	p		
1	1	0.7073806242580120	0.6940283471922162
	2	0.6806760701264203	
3	1	0.7741777739295469	0.7781744296630022
	2	0.7821710853964574	
5	1	0.7741856489803548	0.7686206107596746
	2	0.7630555725389943	
10	1	0.7703185495065397	0.7654826155989835
	2	0.7606466816914272	
20	1	0.7425148287314671	0.7380371437389844
	2	0.7335594587465016	

The best result is marked with darker area, as shown in Tabel 3, 0.7821710853964574, reserved to 4 significant figures, 0.7822, which is the approximate result of Knn. It is also noticeable that the R-squared seem to reach a peak when n_neighbors=3, at the both side the other values has shown a significant decreasing trend (n_neighbors=1 and 20 have performed relatively undesirable). We assumed that the distribution according to these two parameters (n_neighbors and R-squared) would be a positive-skewed distribution. Withing that we attempted other three values in terms of p=1 with n_neighbors=100, 150, 300, the results are 0.7021, 0.6769 and 0.6081 respectively. This would be a strong proof to our assumption, and it would be also reasonable in logic inference.

3.2 3 Neural Network

Neural network has good self-learning, self-adapting and generalization ability [7]. The neuron network was regraded that it would perform best in the model evaluation. Although unfortunately the it came out that the simulated accuracy for the neuron network is 0, which means it is irrational.

The codes are checked for many times, each step was reprogramed and optimized, however, an expected result was still unavailable. It might be a possible reason that there are some unidentified terms in the original data set that prevent the calculation of the algorithm, for example, English or Chinese vocabularies instead of numbers. Or it is also possible that there is a systematic bug existing and we are not able to find it.

***Annotation**: decision tree and random forest models share almost same hyperparameters, and each of them has 6 or 7 (depend on the definition of the data set) hyperparameters, which is impossible to attempt one by one specifically. Therefor we pre-defined some hyperparameters that would fit both models:*

max_depth=15; min_samples_split=10; min_samples_leaf=10 ; max_features=13; max_leaf_nodes=2

3.2 4 Decision Tree

Decision tree classifiers are regarded to be a standout of the most well-known methods to data classification representation of classifiers [8]. In terms of the pre-defined hyperparameters mentioned in the annotation, the only hyperparameter left is the random state.

Table 4: the final result of Decision Tree recorded in terms of R-squared and average R-squared

Hyperparameter(s)	R-squared	Average R-squared
random_state		
41	0.8065845578919374	0.8129920459810878
43	0.8027982365264307	
45	0.8295933435248952	

The best result is marked with darker area, as shown in Tabel 4, 0.8295933435248952, reserved to 4 significant figures, 0.8296, which is the approximate result of Decision Tree model.

3.2 5 Random Forest

Random forests (Breiman, 2001, Machine Learning 45: 5–32) is a statistical- or machine-learning algorithm for prediction. Random decision forests easily adapt to non-linearities found in the data [9]. In terms of the pre-defined hyperparameters mentioned in the annotation, the two hyperparameters left are the random state and n_estimator.

Table 5: the final result of Random Forest recorded in terms of R-squared and average R-squared

Hyperparameter(s)		R-squared	Average R-squared
n_estimator	random_state		
100	41	0.8661340534407640	0.8650578982160946
	43	0.8629055877667559	
	45	0.8661340534407640	
200	41	0.8637283381620011	0.8618816204893105
	43	0.8622654948324604	
	45	0.8596510284734699	
300	41	0.8679527256763118	0.8636224188847919
	43	0.8623755777903024	
	45	0.8605389531877614	

The best result is marked with darker area, as shown in Tabel 5, 0.8679527256763118, reserved to 4 significant figures, 0.8680, which is the approximate result of Random Forest model. Random Forest achieves increased classification performance and yields results that are accurate and precise in the case of large number of instances [10]. It can be able to overcome the over-fitting problem generated due to missing values in the datasets [10]. compare to Decision Tree model. It is interesting that the highest R-squared score appears when n_estimator equals to 300,

ISSN 2959-6130

however, the results show that when n_estimator equal to 100, the average R-squared is the highest.

4 CONCLUSION

The final target of the research is to measure the accuracy of all the models and use the best model to calculate the suitable transfer fee of the player. The criterion of models is the R-squared number, which is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. The model with larger R-squared is more accurate. The final test result of random forest has the R-square (0.8680) which is the most accurate one. The R-squared of decision tree is also high but the best model should be the random forest. The decision tree is a flowchart-like diagram that visually represents the decision-making process by mapping out potential outcomes and courses of action and the random forest is a model conclude many trees, as a result it can be more accurate. The data basement of the model is also comprehensive. The model got the data of football players of Asian, Africa, Europe and America, as the result, using the model, football league from all over the world can estimate a proper price for players they appreciate. Players can use the model to help make sure whether the transfer fee is suitable.

Model	Best final prediction result (R-squared)
Random Forest	0.8679527256763118
Decision Tree	0.8295933435248952
Linear Regression	0.7831111509882093
K-Nearest Neighbors	0.7821710853964574
Neural Network	NaN

Table 6: the simple conclusion of the best final prediction result in terms of R-squared for all the five models

Acknowledgement

Keying Chen, Letian Wu, Ziyang Xu and Zheye Liu contributed equally to this work and should be considered co-first authors.

REFERENCE

[1] Maulud, Dastan, and Adnan M. Abdulazeez. "A review on linear regression comprehensive in machine learning." Journal of Applied Science and Technology Trends 1.2 (2020): 140-147.

[2] Wang, Guoming. "Quantum algorithm for linear regression." Physical review A 96.1 (2017): 012335.

[3] McHale, Ian G., and Benjamin Holmes. "Estimating transfer fees of professional footballers using advanced performance metrics and machine learning." European Journal of Operational Research 306.1 (2023): 389-399.

[4] Bernerth, Jeremy B., and Herman Aguinis. "A critical review and best-practice recommendations for control variable usage." Personnel psychology 69.1 (2016): 229-283.

[5] Keller, James M., Michael R. Gray, and James A. Givens. "A fuzzy k-nearest neighbor algorithm." IEEE transactions on systems, man, and cybernetics 4 (1985): 580-585.

[6] Batista, G. E. A. P. A., and Diego Furtado Silva. "How k-nearest neighbor parameters affect its performance." Argentine symposium on artificial intelligence. 2009.

[7] Wilamowski, Bogdan M. "Neural network architectures and learning algorithms." IEEE Industrial Electronics Magazine 3.4 (2009): 56-63.

[8] Charbuty, Bahzad, and Adnan Abdulazeez. "Classification based on decision tree algorithm for machine learning." Journal of Applied Science and Technology Trends 2.01 (2021): 20-28.

[9] Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. The Stata Journal, 20(1), 3-29.

[10] Ali, Jehad, et al. "Random forests and decision trees." International Journal of Computer Science Issues (IJCSI) 9.5 (2012): 272.