Predicting Football/Soccer Players Transfer Market Value Based on Linear Regression and Decision Trees ECS784P

Abstract – This paper contains the use of 2 machine learning techniques (Linear Regression and Decision Trees) to predict football players' market values, using the dataset called 'fifa_players'. The original dataset consists of 51 features correlated with the footballers' attributes and their corresponding rating for each attribute. 9 of the features used in this dataset are of type object, with the remaining 42 being numerical variables of the type float64 or int64. The features are compared to each other using Pearson's correlation matrix. This paper will go through the data pre-processing steps, which include feature selection, where many of the features are merged or removed. Additionally, a review of existing literature is included to contextualize our approach, looking at machine learning applications in sports analytics. The report concludes with a comprehensive analysis of the models' performance and a discussion of the results.

Keywords - Linear Regression, Decision Trees, Cross-Validation, PCA, Transfer Market Value

I. INTRODUCTION

The transfer market is a key component in the football world and every team gets involved for a maximum of 12 weeks per year. Teams use this window as an opportunity to buy or sell their players for reasons such as financial gain, improving the team, or helping the player to progress in their career. A player's worth is determined by many features and stats including a player's performance, in-game attributes, age, position, current club, and more.

This Dataset [6] contains over 17,000 entries of football players, attributes, performance metrics, and market values, with all the attributes provided by FIFA, which is the international governing body of association football. These include player names, dates of birth, ages, heights, weights, nationalities, overall ratings, potential ratings and more ratings for other attributes.

Accurately predicting these values can significantly enhance determining the real and true transfer market values of players, ensuring that teams don't overpay for players, and also for teams to not sell a player for a price lower than what they are worth. It also would allow teams to identify players that are undervalued in the market, providing an advantage in acquiring talent at lower costs.

By analysing some of the variables listed above, a machine learning model can identify patterns and correlations that human analysis might overlook, providing a more accurate, objective, and data-driven valuation of players. In terms of young players, who aren't as developed, the model also takes into account each player's potential to combat this possible issue.

OBJECTIVES

i. Identify which features highly correlate to a football player's transfer market value.

ii. To use a minimum of two machine learning models on the input features to produce estimations on a football player's transfer market value.

iii. Evaluate both techniques used, through R^2 and mean-squared error values.

iv. To compare the strengths and weaknesses of both regression models used.

v. Identify challenges and improvements to the report and the machine-learning pipeline.

vi. Use common Python libraries to visualize and present the data so it's easy to understand.

vii. Analyse and clean the input dataset features and entries.

II. LITERATURE REVIEW

In this part of the report, I will examine 5 articles to lay the foundational groundwork for our study.

The first report [1] aims to predict football players' market values using FIFA data and machine learning techniques. They use 4 regression models - Linear Regression, Multiple Linear Regression, Regression Tree, and Random Forest Regression. The study shows that the Random Forest Regression model obtained the best performance. The approach shows that these FIFA attributes could be used with machine learning algorithms to accurately predict a player's transfer value. This confirms the proposed methodology of this report, as it confirms the suitability of using Linear regression and decision tree models in accurately estimating the target variable.

The second report [2] specifically focuses on linear regression models to predict the market value of a football player. The performance metrics used in this report are R-squared, Mean Absolute Error, Median Absolute Error and RMSE. The methodology involves data extraction, visualization, dimensionality reduction using PCA, and modelling. They select suitable features upon inspection of the correlation matrix. The accuracy metric used in the report is R-squared, with the linear regression model obtaining 91% 'accuracy' after applying a log transformation to the dataset. This shows that despite the previous report showing that decision trees perform better, it is still possible to obtain a high accuracy whilst using linear regression. This confirms the suitability of using linear regression in this report, as well as using these attributes to accurately predict the value of the player.

The third report [3] uses machine learning approaches in predicting football players' market values. They uses regularised linear models, like Lasso and ridge regression, which are used to penalise large coefficients. This helps in reducing overfitting. Furthermore, the report discusses using decision tree algorithms. This confirms the suitability of the regularisation models I have used in this report. It also confirms the suitability of comparing the two models that I have used.

The fourth report [4] explores the predictability of professional football players' transfer values using player skills and characteristics. This is done through 3 machine learning models: Linear Regression, Support Vector Regression, and Random Forest Regression. This confirms the proposed methodology of this report, as it confirms the suitability of using Linear regression and decision tree models in accurately estimating the target variable. The report confirms my results, as well as the use of the GridCV algorithm for optimising the accuracy. Similar to the report, I have also seen that the decision tree algorithm performed the best, which provides some validity to my results.

The final report [5] aims to predict the player's transfer values from the 2018/19 season. It uses linear regression models, using performance metrics and variables like age and height. Additionally, it introduces factors like social media influence, highlighting its significance alongside other factors like player performance.

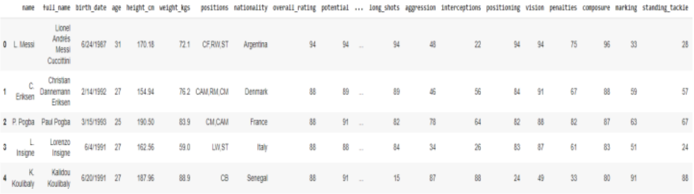
III. DATA MANAGEMENT & PROCESSING

External Libraries Used

- **NumPy:** a Python Library used for executing mathematical operations on multidimensional arrays and matrices. It is designed to handle large datasets.
- **Pandas:** A Python Library that provides data structures and functions that allow data cleaning, analysis and visualisation easier. It has 2 data structures: the DataFrame and the Series, both of which allow for handling missing data, as well as other analytical features on .csv files.
- **Scikit-learn or sklearn:** a Python library that provides both supervised and unsupervised learning algorithms for both regression and classification problems. The library can be used for model fitting for machine learning algorithms like Linear regression and Decision Trees.
- **Matplotlib:** A Python library used to provide visual graphs like histograms and box plots. The library allows you to customise the size, colour and labels of the plot.
- **Seaborn:** a Python library similar to the above, that produces statistical graphs like scatter plots.

Data Source and Description

The initial dataset has a total of 51 features and 17,954 rows. Many of the features, specifically the ones of type object, were removed, as they couldn't be used, such as: "name," "full_name", "birth_date", "nationality", "positions", "body type", "national_team", "national_team_position" and "preferred_foot". The remaining variables are continuous and numerical, meaning that they could be used. All the attributes are given as ratings out of 100, the player wages and values are given in euros, and height is in centimetres.



	name	full_name	birth_date	age	height_cm	weight_kgs	positions	nationality	overall_rating	potential	...	long_shots	aggression	interceptions	positioning	vision	penalties	composure	marking	standing_tackle
0	L. Messi	Lionel Andrés Messi Cuccittini	6/24/1987	31	170.18	72.1	CF,RW,ST	Argentina	94	94	...	94	48	22	94	94	75	96	33	28
1	C. Eriksen	Christian Dannemann Eriksen	2/14/1992	27	154.94	76.2	CAM,RM,CM	Denmark	88	89	...	89	46	56	84	91	67	88	59	57
2	P. Pogba	Paul Pogba	3/15/1993	25	190.50	83.9	CM,CAM	France	88	91	...	82	78	64	82	88	82	87	63	67
3	L. Insigne	Lorenzo Insigne	6/4/1991	27	162.56	59.0	LW,ST	Italy	88	88	...	84	34	26	83	87	61	83	51	24
4	K. Koulibaly	Kalidou Koulibaly	6/20/1991	27	187.96	88.9	CB	Senegal	88	91	...	15	87	88	24	49	33	80	91	88

***Figure 1**: Data sample from the original dataset*

Missing Data

Datasets having missing values are common when dealing with large datasets. Whilst preprocessing the data, we need to ensure that there are no missing values, as they may impact the model's performance, and the models would not even be able to process the data. ***df. isnull().sum()*** can be used to identify the sum of missing values per feature in the dataset. Some ways to deal with missing data are:

- Replacing the missing values with the mean for the column.

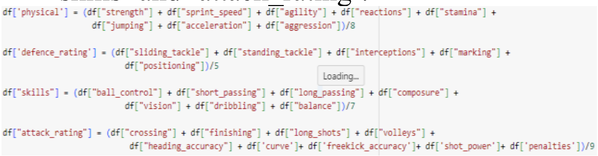
- Replacing the missing values with the mode of the categorical value.
- Completely removing the rows where there are missing values.
- Replace the missing values using the K-Nearest Neighbours approach. It identifies the 'k' nearest points based on some distance metric and imputes values based on these neighbours.

In the project dataset, 214 values are missing from the feature ‘value_euro’, and 209 from the feature ‘wage_euro’. I replaced the missing values with the mean value for that feature. Since the number of missing values is small compared to the total dataset size, using the mean is suitable. Furthermore, after applying a logarithmic transformation to the target variable, the distribution is relatively normal, further confirming the suitability of using the mean.

Feature selection

The dataset originally contained 51 features, but I reduced this by the following:

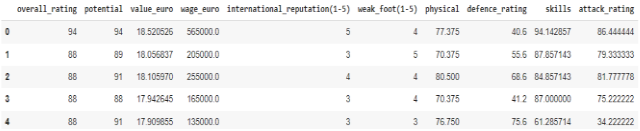
- The object type columns listed above are dropped as they are not useful and cannot be processed by the model.
- Goalkeepers had to be removed as upon inspection, none of the attributes were related to goalkeeper strengths.
- I grouped, merged and averaged some of the features into ‘physical’, ‘defence_rating’, ‘skills’ and ‘attack_rating’.



```
df['physical'] = (df["strength"] + df["sprint_speed"] + df["agility"] + df["reactions"] + df["stamina"] +
                  df["jumping"] + df["acceleration"] + df["aggression"])/8

df['defence_rating'] = (df["sliding_tackle"] + df["standing_tackle"] + df["interceptions"] + df["marking"] +
                        df["positioning"])/5

df["skills"] = (df["ball_control"] + df["short_passing"] + df["long_passing"] + df["composure"] +
                        df["vision"] + df["dribbling"] + df["balance"])/7

df["attack_rating"] = (df["crossing"] + df["finishing"] + df["long_shots"] + df["volleys"] +
                        df["heading_accuracy"] + df['curve']+ df['freekick_accuracy']+ df['shot_power']+ df['penalties'])/9
```

***Figure 2**: Merging similar features.*

- ‘release_clause_euro’ is also dropped as in most cases, this value is the same as the players’ transfer market value. Also, ‘national_rating’ is dropped because most of the players in the dataset do not have values entered for this feature (over 15,000), despite initially seeing that it has a strong correlation.
- Finally, features ‘age’, ‘height_cm’, ‘weight_kgs’ and ‘national_jersey_number’ are also dropped due to their negligible correlation with a player’s transfer market value.
- The final version of the dataset (not including ‘value_euro’ as this is the target variable) has only 9 features, which are further reduced to 7 after applying PCA, which are ready to be trained using a machine learning algorithm. These 9 features can be seen to mostly have a strong positive correlation relationship with the target variable. It can be assumed that the features with a weak correlation (0.5-1) with the target variable (defence_rating, and weak_foot(1-5), are the ones to be dropped after PCA



	overall_rating	potential	value_euro	wage_euro	international_reputation(1-5)	weak_foot(1-5)	physical	defence_rating	skills	attack_rating
0	94	94	18.520526	565000.0	5	4	77.375	40.6	94.142857	86.444444
1	88	89	18.056837	205000.0	3	5	70.375	55.6	87.857143	79.333333
2	88	91	18.105970	255000.0	4	4	80.500	68.6	84.857143	81.777778
3	88	88	17.942645	165000.0	3	4	70.375	41.2	87.000000	75.222222
4	88	91	17.909855	135000.0	3	3	76.750	75.6	61.285714	34.222222

***Figure 3**: Dataset after feature selection (Before PCA)*

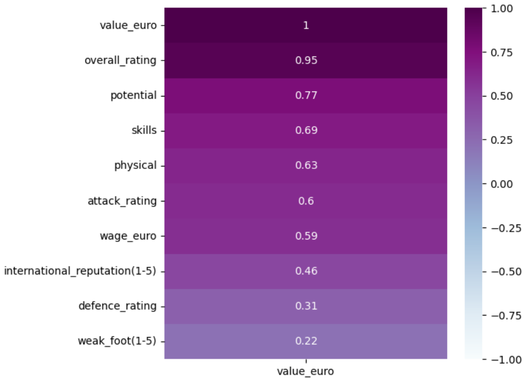


***Figure 4**: Correlation between features and target.*

Normality Testing

We also need to ensure and check whether the distribution of the player’s transfer market value has a normal distribution (a symmetric bell-shaped curve). This is desired as a linear regression model assumes that the residual errors (differences between observed and predicted values) are normally distributed. If the assumption holds, the least squares estimates are unbiased, and the estimation of the coefficients is considered efficient (having the smallest variance).

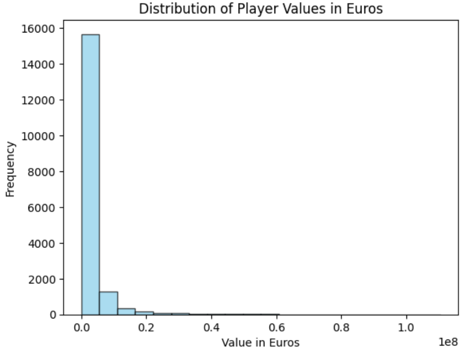


***Figure 5**: Histogram of value_euro.*

From this Histogram, we can see that the distribution is heavily right-skewed, meaning that most of the player values are concentrated in the lower end of the euro value spectrum, with very few players having high values. The bins are concentrated towards the left of the plot, indicating that there are very few high-value outliers.

To address this, a logarithmic transformation is applied to the 'value_euro' column, as shown in the second histogram. The logarithmic transformation is a common technique used to reduce skewness in positively skewed data.

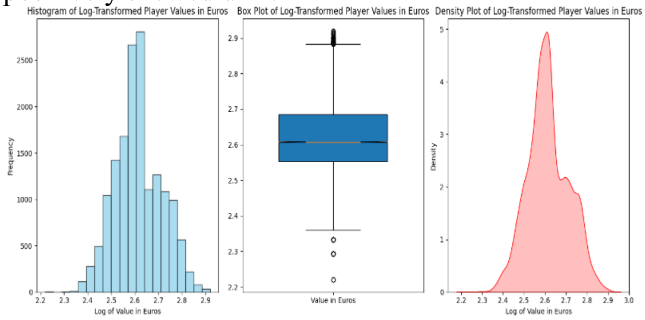


***Figure 6**: Histogram of log-transformed value_euro*

The second histogram shows the distribution of the log-transformed 'value_euro' values. This plot indicates a more symmetric distribution of player values, which is less skewed than the original data. The distribution is more symmetric and is particularly useful for handling outliers, as it reduces the impact of the extreme values on the analysis.

The box plot of the log-transformed values shows a distribution that is more compact and less skewed, with fewer outliers. The median is more centred, and the interquartile range (the box) is more symmetric about the median.

The density plot for the log-transformed values shows a relatively smooth curve that approximates the normal distribution, further confirming that the log transformation helped normalise the data.

IV. METHODOLOGIES

Predicting the transfer value of players given an input dataset where you have both the features (such as player statistics) and the target values (the transfer values) is a **supervised learning** problem. In supervised learning, the goal is to learn a mapping from inputs (features) to outputs (targets), given a labelled dataset. This can also be considered a **regression problem**, as the target variable is continuous, as it can take on any numerical value.

The accuracy of the model is measured by comparing the predicted results of the test set and comparing them with the actual results. This accuracy is evaluated using mean squared error (MSE) and R^2 score.

The R^2 score represents the percentage of the variance in the dependent variable that is predicted from the independent variable. In other words, it shows how well the data fits the regression model. An R^2 score of 1 indicates that the regression predictions perfectly fit the data. An R^2 score of 0 means that the model does not explain any of the variance in the target variable.

The MSE measures the average of the squares of the errors. The error in this case is the difference between the actual value and the predicted values. A lower MSE value indicates a model that accurately predicts the data, while a higher MSE indicates a model that performs poorly in predicting the target values.

Linear Regression Algorithm

Linear Regression is a supervised learning algorithm. This algorithm is used to find a linear equation that best predicts the dependent variable based on the values of the independent variables. This equation is shown as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

In this equation, it can be seen that:

- y: dependent variable
- $x_1, x_2, \ldots x_n$ are the independent variables
- β_0 is the y-intercept
- $\beta_1, \beta_2, \ldots \beta_n$ are the independent variables' coefficients
- ϵ is the error term

Linear regression finds the coefficients (β) that minimise the difference between the predicted values and the actual values. This would also make the MSE low.

Linear regression suits our problem of predicting football players' transfer market values. It can be used here as it is used for predicting outcomes that are continuous numbers, such as a player's market value Linear regression is effective when there is a linear relationship between the features (such as age, potential, skill ratings, etc.) and the target variable (the player's market value).

Based on the correlation matrix seen before, it can be seen that several of the features do have a strong relationship with the target variable. As a result, Linear regression would be suitable for displaying this strong relationship as is computationally efficient and less prone to overfitting when the correct model is chosen, and the assumptions of linear regression are met.

Decision Tree Algorithm

Decision trees are a supervised learning method that can be used in both regression and classification problems. Decision trees can catch non-linear relationships between features and the target variable. Constructing a decision tree involves recursively splitting the training set into subsets based on the feature that results in the greatest reduction in variance. The process of splitting continues until a criterion is met, like when it reaches a maximum specified depth. These models can also provide insights into the importance of different features, which can be valuable for understanding what attributes contribute most to a player's market value.

In this report, I test both the DecisionTreeRegressor and the RandomForestRegressor. The difference between the two is that the DecisionTreeRegressor uses a single tree to make the predictions. It is easier to read and interpret, as it is only 1 single tree, however, it is prone to overfitting, making it perform worse on the test set compared to the other algorithm. RandomForestRegressor constructs multiple decision trees and outputs the average prediction of the individual trees, which should make it produce more accurate results. As it uses multiple trees, it reduces the risk of overfitting the data, compared to using a single tree. However, observing the trees would be harder to interpret than using a single tree, and it would have a higher computational cost, as you would have to train multiple trees on the input data.

V. ANALYSIS, TESTING AND RESULTS

Analysis & Testing

Following the dataset's preprocessing, I tested the two regression algorithms on the dataset. Utilizing scikit-learn's default settings for hyperparameters, these algorithms were applied to the refined dataset. The target variable is the players' transfer market values, while seven selected features were predictors.

80% of the dataset was assigned to the training dataset, and 20% was assigned to the testing dataset. Since accuracy is inherently a classification term, in our regression problem accuracy is determined by how well or close a model's predictions are to the actual value. I used both the R^2 value, and the mean squared error value (MSE) as metrics to determine the accuracy of each model.

Cross-validation is a robust method, as the dataset is divided into k folds, which in this case is 5. At each fold, the model gets trained on k-1 folds and gets tested on the remaining folds. This process is repeated k times, and each fold is used once. The cross-validation score is calculated as the average.

Regularisation is a technique used that prevents overfitting (when a model performs badly with unseen data). It occurs when a model is too complex or when it has too many parameters. Regularisation addresses this by adding a penalty, where the model minimises the MSE.

The 2 regularisation techniques I used are Lasso and ridge. Here we use LassoCV and RidgeCV, which use cross-validation to determine the best value for alpha, which for Lasso is the magnitude of the penalty term which is the absolute value of the magnitude of the coefficients and for ridge it is proportional to the square of the coefficient values.

For both algorithms, I used cross-validation, and regularisation was used only for linear regression, as it doesn't apply to decision trees. Decision trees are non-linear models that do not use coefficients in the same way linear models do. Based on this, regularisation does not apply to decision trees.

For both of the Decision Tree algorithms, a cross-validation technique was used called Grid Search Cross-Validation (GridCV). To avoid overfitting, we need to determine the optimal value for the max_depth of the tree, as well as the minimum amount of sample. This works through multiple combinations of parameter values, cross-validating as it goes to determine which parameters give the best performance.

Results

	R^2 value (Accuracy)	**MSE**
Linear Regression	0.88201	0.0011473
	0.87957	0.0011899
Linear Regression (CV)	0.88163	0.0011496
Linear Regression (LassoCV)	0.88198	0.0011476
	0.87958	0.0011898
Linear Regression (RidgeCV)	0.88201	0.0011473
	0.87958	0.0011899
Decision Tree (DecisionTreeRegressor) (GridCV)	0.94320	0.00055234
	0.91610	0.00082903
Decision Tree (RandomForestRegressor) (GridCV)	0.98716	0.00012481
	0.93751	0.00061744
All values are to 5.s.f,		

***Figure** 7: R^2 & MSE values from each algorithm (pink=test results, orange=training results)*

Linear Regression: Analysis of Results

Based on the table above, it can be seen that all the Linear regression algorithms provide similar results. These algorithms produce a testing accuracy (R^2 value) of approximately 87.96%, an

MSE of 0.119, and a training accuracy (R^2 value) of approximately 88.2% and an MSE of 0.115.

- The basic form of Linear regression performs reasonably well, indicating that the features have a linear relationship with the player's market value.
- The cross-validation results are very close to the initial train/test split results, which suggests that the model's performance is consistent across different subsets of the data and there isn't much variance in the folds used for CV. This confirms the model's reliability.
- Both cross-validated regularised versions of linear regression (LassoCV and RidgeCV) results are very close to the initial train/test split results. This would suggest that the input features are all relevant and there isn't much multicollinearity. Therefore, regularization wouldn't change the performance significantly, hence the similar results.

Decision Tree: Analysis of Results

Based on the table above, the DecisionTreeRegressor algorithm has an R^2 value of 0.91610 on the test set, indicating a very good fit and a fit better than the linear regression algorithms. This means that it accurately explains 91.61% of the variance in the target variable (player's transfer market value). The Mean Squared Error (MSE) for the Decision Tree is 0.00082903, which is relatively low, signifying that the predictions are quite close to the actual values.

For the Random Forest algorithm, the results demonstrate an impressive R^2 value of 0.93751, suggesting the model explains approximately 93.75% of the variability in the target variable. This is a notable improvement over both the linear regression models and the single Decision Tree. This indicates that the RandomForestRegressor model provides the most accurate and robust model. Furthermore, the Mean Squared Error (MSE) for Random Forest is 0.00061744, underscoring the model's precision in making predictions that are very close to the actual values. The combination of a high R^2 value and a low MSE shows the Random Forest's efficacy, making it the best and most accurate model in predicting football players' transfer market values.

VI. CONCLUSION

This project aimed to create and compare 2 main models that could accurately predict a player's transfer market value based on their attribute ratings given by FIFA. This was done through the application of Linear Regression and Decision Tree algorithms. Overall, both models showed high accuracies (highest accuracy for each model being 93.8% for Decision Trees (Random Forest) and 88% for Linear Regression (both regularisation models)). Through feature selection and PCA, we can determine that the attributes in the dataset provided by FIFA can be used to accurately predict a player's transfer market value.

It can be concluded that for this supervised regression problem, using a decision tree algorithm, more specifically the Random forest Algorithm, proved to provide more accurate results, providing a higher R^2 value and lower MSE. There may be reasons why this algorithm performed better than linear regression. One reason may be because decision trees are non-linear models, meaning they can capture more complex patterns in the data, that linear models may not be able to capture. This is particularly useful when predicting the transfer market value, which can be influenced by complex interactions between features. Another reason may be that decision trees are less sensitive to outliers compared to linear models. All the project's objectives were successfully met.

This project also has some challenges and limitations. Initially, the dataset contained missing values, especially in the features 'value_euro' and 'wage_euro'. This was a minor setback that needed to be dealt with and was overcome by replacing these values with the mean of the column, which may have contributed to slight inaccuracies. Another problem was the cleaning of data which proved to be time-consuming as the original dataset contained over 50 features, which I eventually reduced to 7. Included in this problem is the handling of non-numerical data that couldn't be applied to the models.

Despite these challenges, they were all overcome, and the project could be considered a success. The process of feature selection, despite its challenges, led to a more manageable and effective dataset. This is evident as the 51 features were merged, combined and removed to produce a more concise dataset of 9 features, becoming 7 after applying PCA. In both algorithms and particularly in Decision trees, it can be seen that a high accuracy was achieved in predicting a player's transfer market values. This is evident through the high R^2 values and the low MSE values, both of which indicate a model that has been trained well in accurately predicting the target variable.

There are also limitations to do with this report. The dataset is derived from FIFA attributes, which might not consider all factors influencing a player's market value, such as real-world performance, injuries, and market demand. Furthermore, despite both Linear Regression and Decision Trees being used, these models might oversimplify the complex and non-linear relationships in the data. The decision tree's performance, though superior, may still not capture the entirety of the market's dynamics.

Using this dataset in my opinion is more beneficial than using a dataset based on in-game stats. FIFA-provided stats are more authentic as they come directly from the governing body of football, ensuring a higher reliability compared to in-game stats that might not accurately reflect real-world performances. This is because many valuable players may not appear as valuable according to in-game stats. FIFA's dataset is likely to be more inclusive, as it presents attributes like players' skills and potential that in-game stats may overlook. These features would appear to describe a player's all-round game in detail for all outfield positions on the pitch, making it easier to compare players objectively.

This project could also be extended through the use of more sophisticated machine learning techniques such as deep learning, which may improve the accuracy of the models used. Furthermore, incorporating real-time data on player performances and market trends could dynamically refine the model's predictions. This would require a large dataset that includes a variety of player performance stats, not only goals and assists, as these metrics would only apply to attacking players. Other metrics such as passing accuracy or number of interceptions would be useful in determining the transfer value of other positions like defenders or midfielders – positions that don't heavily focus on goals or assists. The predictive model could be employed by football clubs for strategic planning in transfer markets, identifying undervalued or overvalued players. They could use the model to determine any bargains available in the transfer market or to determine what price they should sell a player for.

VII. REFERENCES

[1]https://www.researchgate.net/publication/358871715_Predict_the_Value_of_Football_Players_Using_FIFA_Video_Game_Data_and_Machine_Learning_Techniques

[2]https://www.semanticscholar.org/paper/Player-Performance-Prediction-in-Football-Game-Pariath-Shah/acd456aec9e9b2b43f8be18265244c0d147d74f7

[3]https://arno.uvt.nl/show.cgi?fid=161188

[4][2206.13246] Prediction of Football Player Value using Bayesian Ensemble Approach (arxiv.org)

[5]https://www.researchgate.net/publication/331929212_Determinants_of_Transfers_Fees_Evidence_from_the_Five_Major_European_Football_Leagues

[6]https://www.kaggle.com/datasets/maso0dahmed/football-players-data/data