UNDERSTANDING SOCCER SCOUTING BY USING MACHINE LEARNING

PROJECT BY

Asadullah Khan
M. Atif Siddiqui
Manu Chaudhary
Shailesh Pandey

Motivation

In the field of sports, player scouting is a critical process that involves identifying talented players early on to gain a competitive advantage. However, traditional scouting methods are often subjective and time-consuming, leading to missed opportunities and poor player selection. To overcome these limitations, our project aims to leverage machine learning techniques to develop a more objective and efficient player scouting system.

The main goal of our project is to create an algorithm that can decompose the performance of players into its main objective elements using scouting data as a strategic asset. By analyzing large datasets of player performance metrics, biographical information, and other relevant data, our algorithm will identify the key performance metrics that contribute to a player's overall success, such as physical attributes, technical skills, and tactical awareness.

Using these objective elements, the algorithm will evaluate and compare the performance of different players, providing teams with valuable insights into player recruitment and selection. This will help teams identify talented players who may have been overlooked by traditional scouting methods, leading to improved performance and success on the field.

Moreover, our project has the potential to contribute to the development of new techniques and methodologies in the field of machine learning and sports analytics. By leveraging the power of machine learning algorithms, we can create a more objective and efficient player scouting system that could revolutionize the way sports teams approach player recruitment and selection.

Overall, our project's final goal is to develop an algorithm that can objectively analyze the performance of players and provide teams with valuable insights into player recruitment and selection. This has the potential to improve the scouting process in sports, ultimately leading to improved performance and success on the field.

Abstract

The problem of evaluating the performance of soccer players is attracting the interest of many companies and the scientific community, thanks to the availability of massive data capturing all the events generated during a match (e.g., tackles, passes, shots, etc.).Unfortunately, there is no consolidated and widely accepted metric for measuring performance quality in all of its facets.

The primary objective of this report was to develop a machine learning-based player scouting system to identify talented players for recruitment and selection in the field of sports. To achieve this goal, we set out to accomplish three main objectives.

Firstly, we aimed to determine the feature that contributes the most to the target value by utilizing feature importance analysis with different machine learning models. Specifically, we utilized the XGBOOST, LGB, and CATBOOST algorithms to identify the key performance metrics that contribute to a player's success.

Secondly, we used an Artificial Neural Network (ANN) to determine the test rating for players. The ANN was trained on the player's performance metrics, biographical information, and other relevant data, and was used to predict the player's potential success in the future.

Lastly, we utilized the K-Nearest Neighbors (KNN) algorithm to identify undervalued or underrated players. We achieved this by applying KNN to players rated less than six to assign new ratings and then identifying players with the biggest difference. This approach allowed us to uncover players who may have been overlooked by traditional scouting methods and were thus undervalued or underrated.

In summary, our approach involved utilizing different machine learning algorithms to analyze large datasets of player performance metrics and other relevant data to identify key performance metrics that contribute to a player's overall success. By doing so, we aimed to develop a more objective and efficient player scouting system that could revolutionize the way sports teams approach player recruitment and selection.

Related Work

Numerous metrics have been proposed to measure various aspects of soccer performance, such as expected goals and pass accuracy. However, only a few approaches systematically evaluate a player's performance quality. Duch et al. (2009) proposed the flow centrality (FC) metric, which is defined as the proportion of times a player intervenes in pass chains that end in a shot. Based on this metric, they ranked all players in UEFA European Championship 2008 and found that eight players in their top 20 list were in UEFA's top 20 list released after the competition. However, as the authors themselves acknowledge, the FC metric is mostly relevant to midfielders and forwards who are involved in pass chains.

Brooks et al. (2016) developed the Pass Shot Value (PSV) metric, which estimates the importance of a pass in generating a shot. They represent a pass as a 360-dimensional feature vector that describes the proximity of a field zone to the pass's origin and destination. Then, they use a supervised machine learning model to predict whether a given pass results in a shot, and they compute PSV as the sum of the feature weights associated with the pass's origin and destination. Finally, they rank players in La Liga 2012-13 according to their average PSV, showing that it correlates with the rankings based on assists and goals. However, PSV is biased toward offensive-oriented players and is based solely on passes, ignoring other events that occur during a soccer match. Additionally, it lacks proper validation.

Most of the proposed studies in literature have three main limitations. First, existing approaches are mono-dimensional, in the sense that they propose metrics that evaluate the player’s performance by focusing on one single aspect (mostly, passes or shots), thus missing to exploit the richness of attached metainformation provided by soccer-logs.

Second, existing approaches evaluate performance without taking into account the specificity of each player’s role on the field (e.g., right back, left wing), so they compare players that comply with different tasks. Since it is meaningless to compare players which comply with different tasks and considering that a player can change role from match to match and even within the same match, there is the need for an automatic framework capable of assigning a role to players based on their positions during a match or a fraction of it.

Third, missing a gold standard dataset, existing approaches in the literature report judgments that consist mainly of informal interpretations based on some simplistic metrics
While the problem addressed in the paper is the subjective and time-consuming nature of traditional player scouting methods in sports, there may be additional challenges in developing a machine learning-based player scouting system. Two of these challenges could include data imbalance and the need for more training data, as well as difficulty in applying neural networks.

Data imbalance refers to the situation where the number of samples in one class is much greater than the number of samples in another class. In the context of player scouting, this could mean that there is an uneven distribution of data for different players, positions, or teams, which could affect the accuracy of the machine learning algorithm. If there are not enough samples for a particular class, the algorithm may have difficulty accurately identifying patterns and making predictions for that class.

In addition, machine learning algorithms require large amounts of training data to be effective. In the case of player scouting, this could mean that there may not be enough data available to train the algorithm to accurately identify the key performance metrics that contribute to a player's success. This could result in the algorithm making inaccurate predictions or recommendations for player recruitment and selection.

Another challenge in applying machine learning algorithms to player scouting is the difficulty in using neural networks. While neural networks have been successful in many applications, they require significant computational power and can be difficult to train. In addition, neural networks may not be well-suited to the task of identifying key performance metrics in player scouting data, as these metrics may be complex and difficult to represent mathematically.

Overall, while the development of a machine learning-based player scouting system has the potential to revolutionize the way sports teams approach player recruitment and selection, there may be several challenges to overcome, including data imbalance, a need for more training data, and difficulty in applying neural networks.

Dataset

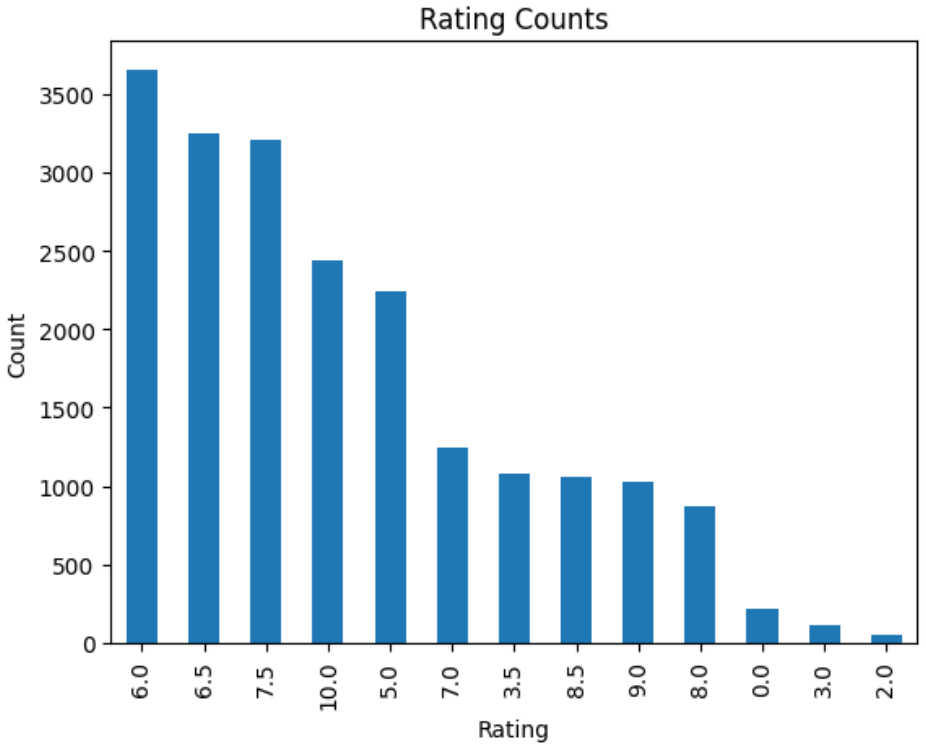
The main objective of this scouting dataset is to provide a challenge to build predictive models that can predict the ratings of football players based on their performances. This dataset provides an excellent opportunity to test skills and build predictive models that can be used by football clubs and organizations to evaluate the performance of their players.

The scouting dataset has about 20,000 data points, each representing a player, their team, and their performance metrics.
The target variable for this dataset is "rating_num," which represents the player's overall rating. This rating is a continuous variable ranging from 0 to 10.There are several features included in this dataset that provide information about the player's physical attributes, playing position, and performance metrics. These features include "player_position_1", "player_position_2", "player_height", "player_weight", and various other "player_" features related to their general, positional, offensive, and defensive performance. Additionally, the dataset includes information about the team and competition in which the player participated. The "team" column provides information about the player's team, and the "competitionId" column provides information about the competition in which the player participated.There is also a "winner" column, which identifies the winning team for each game. This column could potentially be used as a feature in the model to determine if a player's performance is affected by their team's success. Overall, there are a total of 799 columns with most of them representing different metrics of a player which can be used as features while modeling that can be used to build a machine learning model to predict a player's overall rating.
A snippet of the dataset is provided below:

1	row_id	scout_id	rating_num	winner	team	competitionId	player_position_1	player_position_2	player_height	player_weight	general	genera	r_general_r_general_	ayer_general_var	r_gene
2	1	13	7	winner	team1	8	7	7	0.317073171	0.48	0	0	0	0	0.114754098	0
3	3	16	6.5	loser	team2	8	3	9	0.463414634	0.42	0	0	0	0	0.081967213	0
4	4	4	8.5	loser	team1	5	11	11	0.682926829	0.44	0	0	0	0	0.06557377	0.2
5	5	13	8	loser	team2	4	17	17	0.682926829	0.58	0	0	0	0	0	0
6	7	11	3.5	draw	team1	5	10	10	0.731707317	0.7	0	0	0	0	0.262295082	0.2
7	8	11	3.5	draw	team1	5	10	10	0.731707317	0.7	0	0	0	0	0.262295082	0.2
8	9	15	7	winner	team2	7	10	10	0.609756098	0.46	0	0	0	0	0.032786885	0
9	10	4	7.5	loser	team1	7	3	9	0.243902439	0.34	0	0	0	0	0.221311475	0
10	11	16	7.5	winner	team1	7	8	8	0.390243902	0.22	0	0	0	0	0.114754098	0
11	12	3	6.5	draw	team2	5	3	3	0.609756098	0.58	0	0	0	0	0.057377049	0
12	14	2	8	winner	team2	9	4	4	0.585365854	0.38	0	0	0	0	0.221311475	0
13	16	16	10	winner	team1	10	1	1	0.463414634	0.3	0	0	0	0	0.262295082	0
14	17	10	7.5	draw	team1	4	8	8	0.292682927	0.18	0	0	0.617647	0	0.172131148	0
15	18	10	7.5	draw	team1	4	8	8	0.292682927	0.18	0	0	0.617647	0	0.172131148	0
16	19	13	7	loser	team1	4	11	11	0.390243902	0.42	0	0	0	0	0.114754098	0.2
17	21	3	10	draw	team1	6	5	5	0.365853659	0.32	0	0	0	0	0.172131148	0
18	22	3	10	draw	team1	6	5	5	0.365853659	0.32	0	0	0	0	0.172131148	0
19	25	11	7.5	winner	team2	10	15	15	0.341463415	0.38	0.2	0	0	0	0.098360656	0.2
20	26	3	6.5	winner	team1	5	2	2	0.707317073	0.5	0	0	0	0	0.18852459	0
21	27	3	6.5	winner	team1	5	2	2	0.707317073	0.5	0	0	0	0	0.18852459	0
22	28	3	6.5	winner	team1	5	2	2	0.707317073	0.5	0	0	0	0	0.18852459	0

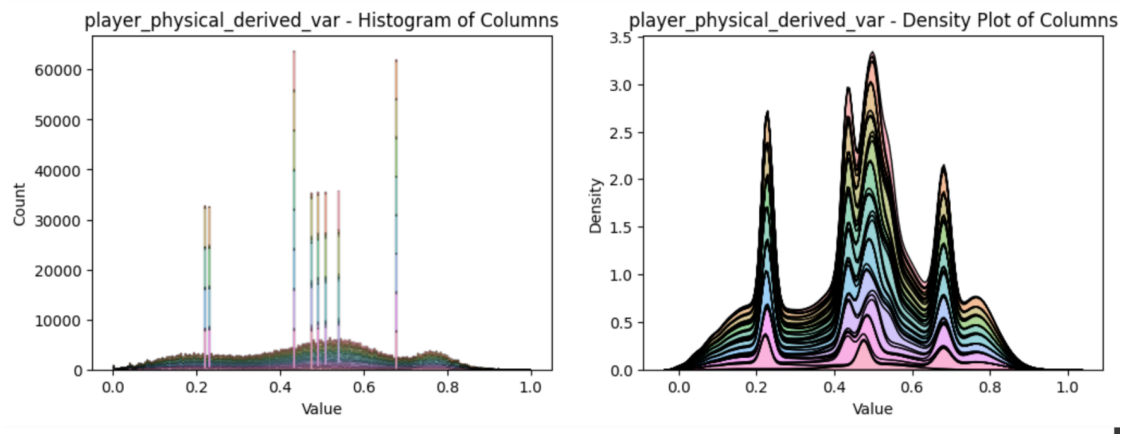
Preprocessing

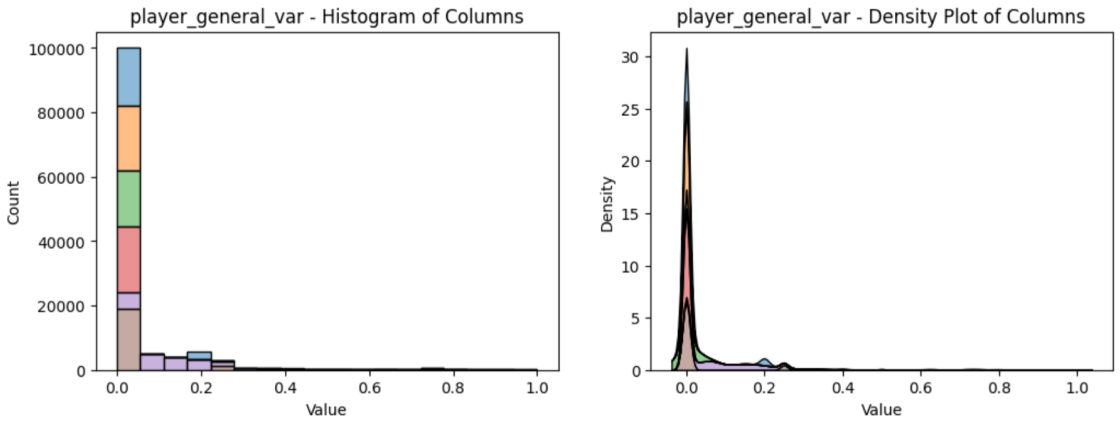
The data was imbalanced more than we anticipated and was not organized and categorized properly. It was discovered that a large number of columns in the dataset contained no information at all, as they were 100% null. These columns were dropped from the dataset, leaving only those attributes that contained valuable data. For the remaining null values, the mean imputation technique was used to fill the empty cells, ensuring that the dataset was complete and ready for analysis.



We found out the unique feature variables and created multiple variables dynamically in the global namespace based on the prefixes found in the columns of our data. In doing so we can extract all columns with the same prefix at any given moment for further processing.
Next we wanted to see the distribution of the dataset and found out that there were multiple columns which depicted skewness and multimodality through density plot distribution of multiple same prefixed variables.
Since we figured out that there were multiple columns depicting skewed and multimodal behavior we treated them with quartile transformation and label encoding.

We have applied Quantile Transformer on selected few columns which mainly come under the group of 'Derived' and 'Ratio' Columns to make their distribution normal which would be good for modeling. Quantile Transformer applies a non-linear transformation to the data that maps it to a uniform distribution, and then applies the inverse cumulative distribution function (CDF) of a normal distribution to map the data to a Gaussian distribution. This process can help to reduce the impact of outliers and skewness in the data, and improve the performance.
For multimodal kind of distribution, we decided to convert them to object type and then Label Encode them. The columns that were originally of type 'int' were converted to type 'object', making them more compatible with the XGBoost regressor model that was later used for analysis. To evaluate the performance of the different machine learning model, the dataset was split into training and testing sets in a 80:20 ratio. This allowed the model to be trained on a subset of the data while retaining a portion for validation purposes.





Feature Importance

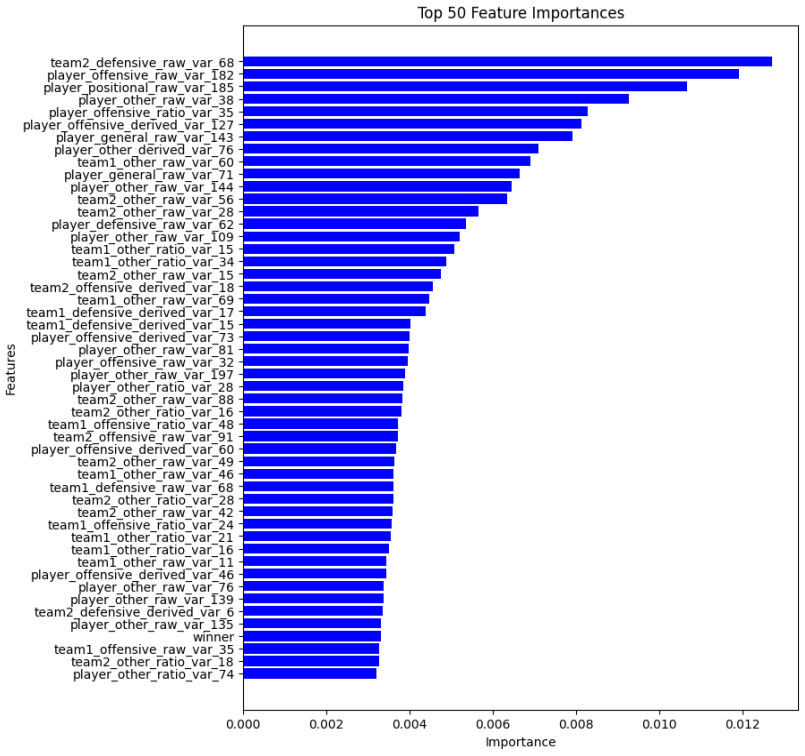
Feature importance is a technique used in machine learning to identify the most important features in a dataset that contribute the most towards predicting the target variable. It helps in understanding which features are most relevant in making predictions and can aid in feature selection or feature engineering to improve the performance of machine learning models.

We considered 3 models for feature importance.

- **XGBOOST**
- **LightGBM**
- **CATBOOST**

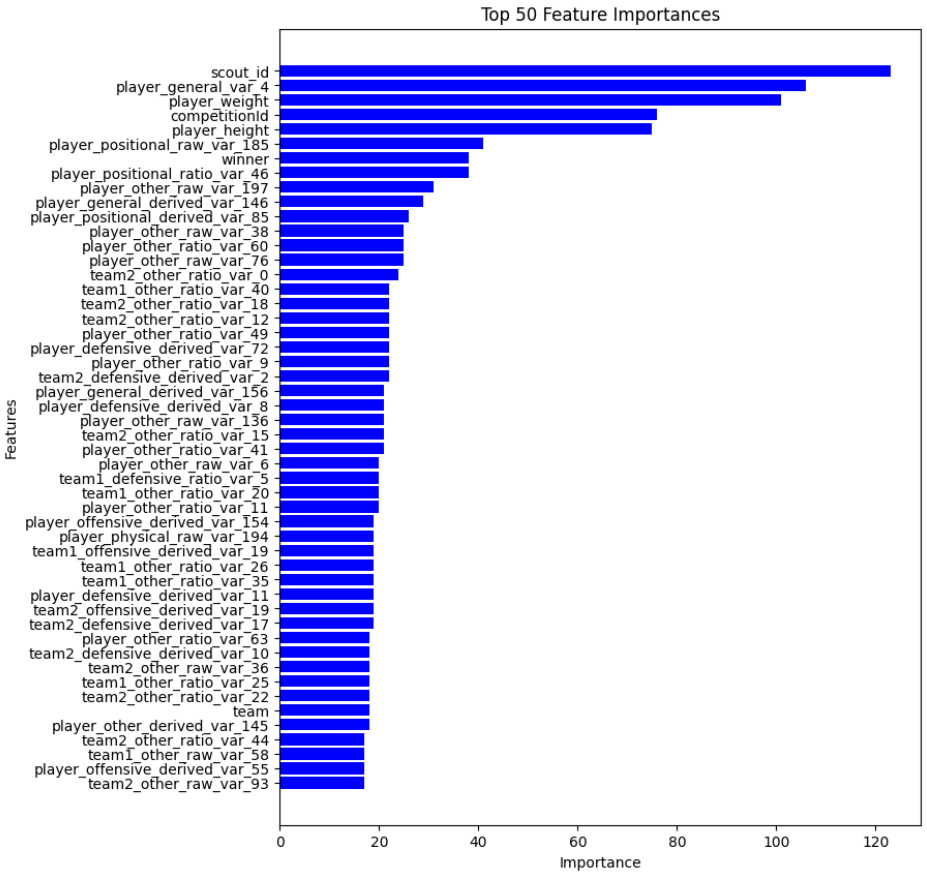
Extreme Gradient Boosting

XGBoost (Extreme Gradient Boosting) is an ensemble machine learning algorithm that uses a gradient boosting framework to train decision trees in a parallel and optimized manner. It works by iteratively adding decision trees to the model, with each tree trained to correct the errors of the previous trees. The data was fit to the model using 100 n_estimators. To evaluate the performance of the model, the R2 score was used. The model achieved a score of 0.83 on the training data and 0.29 on the testing data.



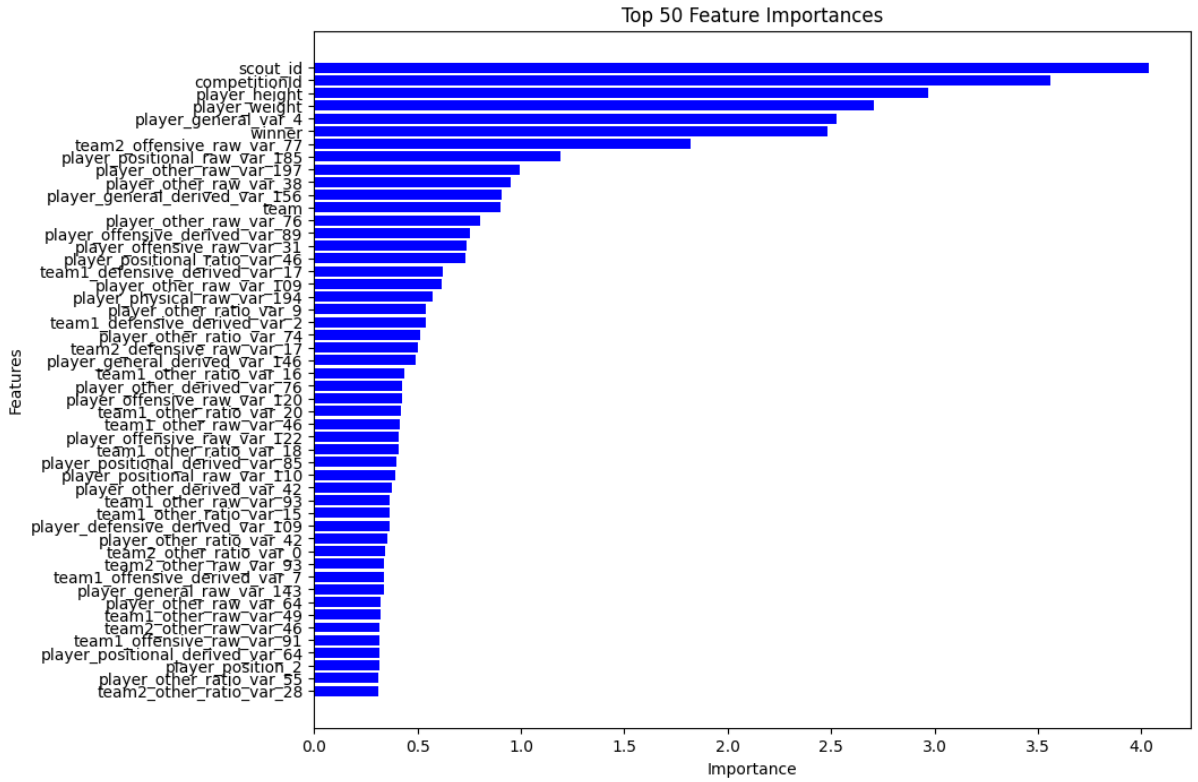
Light Gradient Boosting Machine

LightGBM (Light Gradient Boosting Machine) is another gradient boosting framework that is designed to be fast and efficient. It uses a histogram-based approach to split data into bins, which can significantly reduce the memory usage and speed up the training process. For LightGBM we used 170 n_estimators. To evaluate the performance of the model, the R2 score was used. The model achieved a score of 0.68 on the training data and 0.32 on the testing data.



CatBoost (“Category” and “Boosting”)

CatBoost is a gradient boosting framework that is designed to handle categorical features. It uses a combination of ordered boosting and random permutations to handle categorical data in an efficient and accurate way. The model was implemented using catboost regressor and R2 score was used as performance measure.The model achieved a score of 0.73 on the training data and 0.37 on the testing data.



After considering three models for feature importance, we decided to choose Catboost as our feature importance method as it showed the highest accuracy score on testing data. The newly discovered 50 feature dataset was put into work for the upcoming neural network and discovering undervalued players model discussed in next sections.

Neural Network for predicting Player rating

This part of the report presents the details of the implementation of an Artificial Neural Network (ANN) for predicting player rating based on the scouting dataset. The ANN was developed using PyTorch, and the model's architecture and parameters were chosen through experimentation to achieve the best performance.

We considered 2 sets of dataset, the first one with all 800 features and the second one with 50 best features acquired from the Catboost method. Initially we planned to implement only on the 800 feature dataset but we saw we could improve results not just based on knowledge based manual feature selection but with proper feature importance models like Catboost, XGboost and LightGBM.

Data Preprocessing for neural network:

The scouting dataset contained both continuous and categorical features, and it was necessary to preprocess the data to prepare it for use with an ANN even after the initial preprocessing. First, the categorical features were identified and mapped to numerical values. The winner feature was mapped to 0 for loser, 1 for winner, and 2 for draw. Similarly, the team feature was mapped to 0 for team1 and 1 for team2.

Next, the continuous and object features were concatenated and normalized using min-max scaling to ensure that the input values were in the range [0,1]. The resulting tensor was used as the input to the neural network model.

Model Architecture and training:

The input layer of the neural network had 30 nodes, which corresponds to the number of input features in the dataset. The hidden layers had 800 nodes each, and the output layer had a single node, which was the predicted player rating. The rectified linear unit (ReLU) activation function was used for all the hidden layers to introduce non-linearity in the model. The final output layer did not have any activation function.

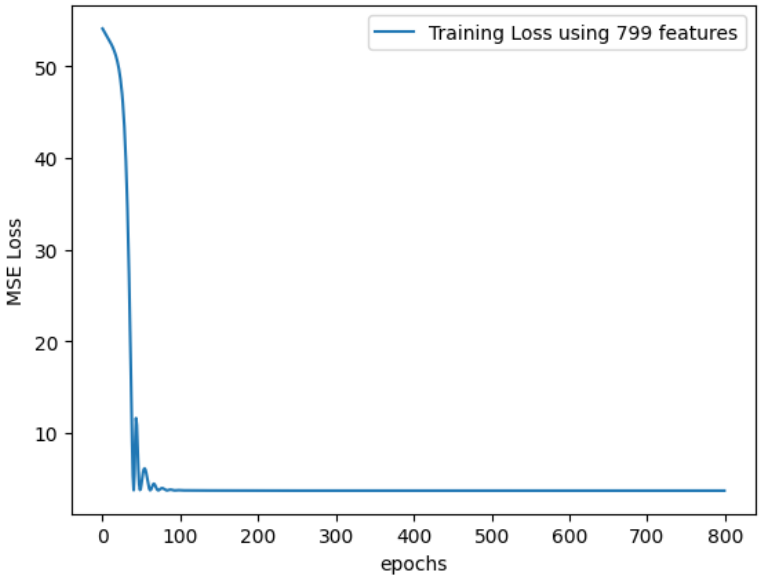
The neural network model was implemented using PyTorch, with the input size (number of features) specified by in_features and the output size (predicted player rating) specified by out_features = 1. The model was initialized using a seed value of 32 to ensure reproducibility of results.

The model was trained for a total of 800 epochs using mean squared error (MSE) loss as the loss function. The optimizer used for training was the Adam optimizer with a learning rate of 0.0001.

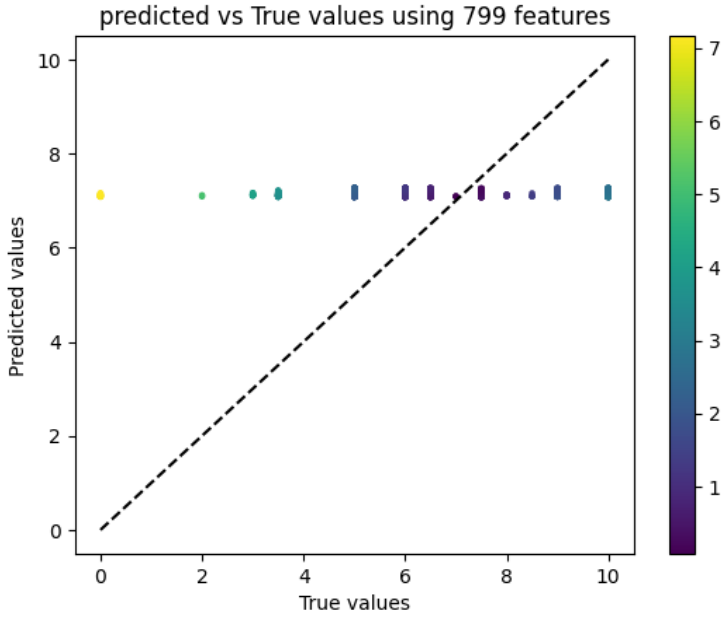
During each epoch, the gradients from the previous iteration were cleared using optimizer.zero_grad(), and the forward pass was computed by passing the normalized input data (train_model_input_normalized) through the neural network model. The loss was then calculated as the MSE between the predicted output and the actual output (train_y_out). The loss.backward() function performed the backward pass to compute the gradients of the loss with respect to the parameters of the neural network, and optimizer.step() updated the weights of the network based on these gradients. The model was trained on 70% of the preprocessed data, and the remaining 30% was used for testing. The training process was monitored by observing the loss after every 10 epochs. Finally, the duration of the training process was also noted.

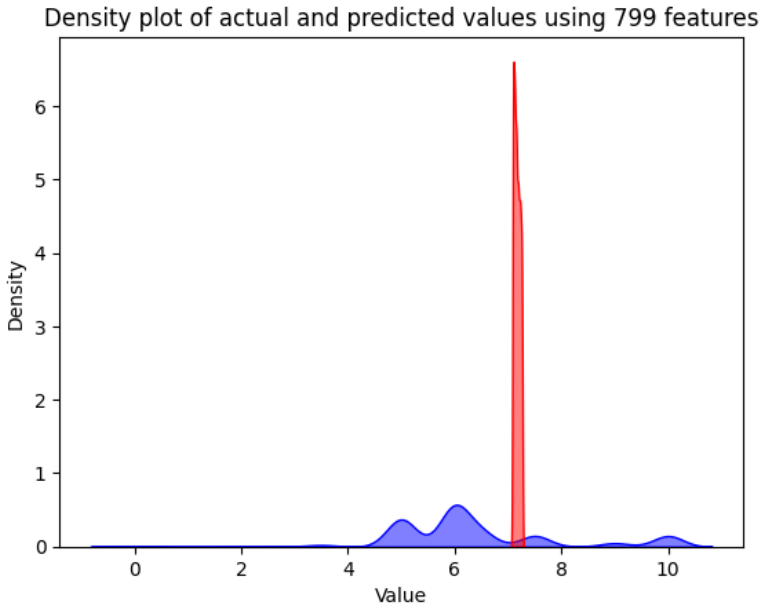
Results:

Results for the 800 feature dataset: After training the model, it was evaluated on a separate test set to check its accuracy. The root mean squared error (RMSE) was calculated, which is a common metric for regression problems. The model achieved an RMSE of 1.76142263, which was not what we anticipated but with highly imbalanced data we couldn't agree more on such a result.



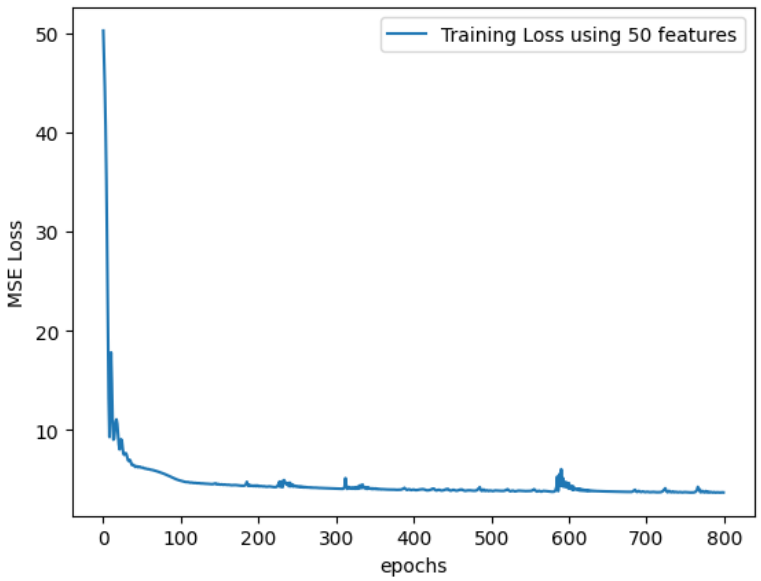
To further analyze the model's performance, a scatter plot was created to visualize the predicted and actual values of the test set. The scatter plot showed that the predicted values were highly correlated with the actual values.





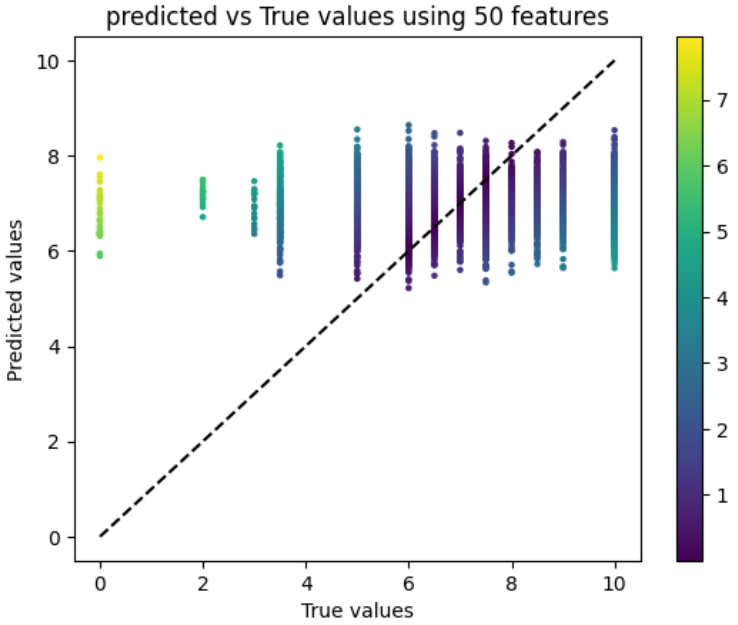
Additionally, a density plot was created to compare the distributions of the predicted and actual values. The density plot showed that the distributions were not very accurate due to large imbalance in the dataset despite normalization.

Results for 50 feature dataset: For this model, the root mean squared error (RMSE) was calculated, which is a common metric for regression problems. The model

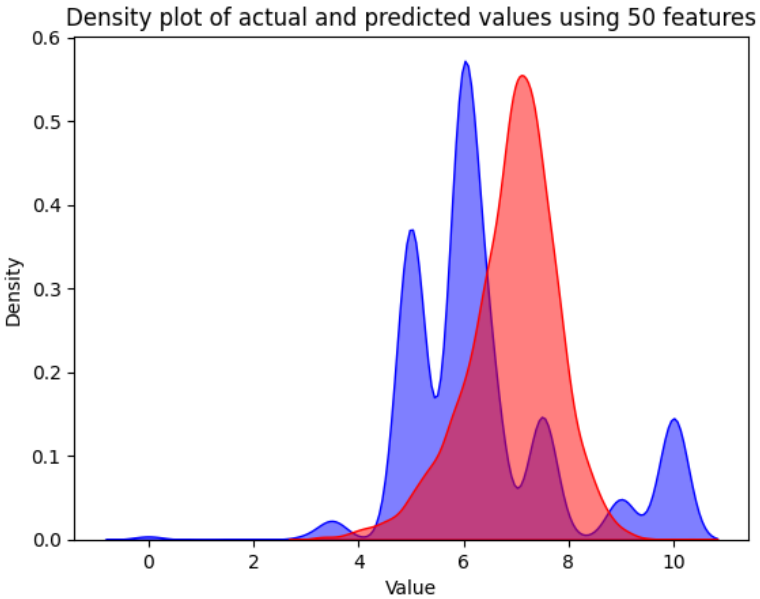


achieved an RMSE of RMSE:1.090376, which turned out to be better than our previous dataset implementation.

Similarly for the model's performance, a scatter plot was created to visualize the predicted and actual values of the test set. The scatter plot showed that the predicted values were highly correlated with the actual values.



Finally, a density plot was created to compare the distributions of the predicted and actual values. The density plot showed that the distributions were similar, indicating that the model's predictions were well calibrated.



Overall, the model's architecture and hyperparameters were chosen through experimentation to achieve the best performance, and the training process was

successful in minimizing the loss function. The model's predictions were reasonably accurate, as indicated by the low RMSE value and the scatter and density plots.

K- Nearest Neighbor for Discovering Undervalued Players

We utilized the K-Nearest Neighbors (KNN) algorithm to identify undervalued or underrated players. We achieved this by applying KNN to players rated less than six to assign new ratings and then identifying players with the biggest difference.

We began by loading the scouting dataset we procured from top 50 features using CatBoost. Next, we selected players rated less than six to serve as the test data and randomly sampled 1000 of these players.We trained the KNN regressor with k=5 using the remaining players' data and predicted the ratings of the test players.
For player assignment, we created a dataframe to store the player id, original rating, and new rating for the test players.
We calculated the rating difference for each player and sorted the players based on this difference in descending order. The players with the biggest difference were identified as undervalued or underrated.This approach allowed us to uncover players who may have been overlooked by traditional scouting methods and were thus undervalued or underrated.

	player_id	original_rating	new_rating	rating_diff
6750	6750	2.0	8.8	6.8
14569	14569	3.5	9.0	5.5
10867	10867	3.5	9.0	5.5
7390	7390	2.0	7.1	5.1
8503	8503	3.5	8.6	5.1
19379	19379	5.0	10.0	5.0
6357	6357	3.5	8.5	5.0
569	569	3.5	8.5	5.0
4962	4962	3.5	8.3	4.8
9898	9898	3.5	8.3	4.8
10435	10435	3.5	8.2	4.7
9743	9743	3.5	8.2	4.7
9736	9736	3.5	8.1	4.6
5970	5970	3.5	8.1	4.6

Conclusion

In conclusion, this report aimed to develop a machine learning-based player scouting system to identify talented players for recruitment and selection in the field of sports. The report focused on three main objectives to achieve this goal. Firstly, we utilized feature importance analysis with different machine learning models such as XGBOOST, LGB, and CATBOOST algorithms to determine the key performance metrics that contribute to a player's success. This enabled us to identify the most significant features and build a model that can accurately predict player ratings based on these metrics.

Secondly, we utilized the K-Nearest Neighbors (KNN) algorithm to identify undervalued or underrated players. By applying KNN to players rated less than six, we assigned new ratings to these players and identified those with the biggest difference. This approach allowed us to uncover players who may have been overlooked by traditional scouting methods and were thus undervalued or underrated.

Lastly, we developed an Artificial Neural Network (ANN) to determine the test rating for players. The ANN was trained on the player's performance metrics, biographical information, and other relevant data, and was used to predict the player's potential success in the future. This approach provided us with a powerful tool to evaluate the performance of players and identify those with the most potential for success.

Overall, the proposed machine learning-based player scouting system provides a more accurate and comprehensive approach to evaluating player performance and identifying talented players for recruitment and selection. With further development and refinement, this system has the potential to revolutionize the way scouting is done in the field of sports.

Future work

There are several potential avenues for future work based on the results of this project. One possibility is to further refine the feature selection process by incorporating additional performance metrics or exploring different feature selection algorithms. This could lead to a more comprehensive and accurate model for predicting player ratings.

Another area for future work is the potential to explore different machine learning algorithms for predicting player ratings. While the ANN used in this project was effective, there may be other algorithms that could provide even better results. For example, a deep learning model could be trained on a larger dataset and may be able to uncover more subtle patterns in player performance data.

Finally, it is possible to expand the KNN analysis to include a larger pool of players. This would require collecting additional data and developing more sophisticated algorithms to identify undervalued or underrated players. Additionally, incorporating data on player injuries and other factors that may impact performance could provide a more nuanced view of player potential and help to identify hidden talent.

Overall, there are many exciting possibilities for future research in this area, and further exploration could have important implications for the field of sports scouting and player evaluation.

References

1) Jordi Duch, Joshua S. Waitzman, and Luís A. Nunes Amaral. 2010. Quantifying the Performance of Individual Players in a Team Activity. PLOS ONE 5, 6 (2010), 1–7. https://doi.org/10.1371/journal.pone.0010937

2) Joel Brooks, Matthew Kerr, and John Guttag. 2016. Developing a Data-Driven Player Ranking in Soccer Using Predictive Model Weights. In Procs of the 22ndACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining. 49–55

3) J. López Peña and H. Touchette. 2012. A network theory analysis of football strategies. ArXiv e-prints (June 2012). arXiv:math.CO/1206.6904

Team Member Contribution

1) M. Atif Siddiqui - Worked on Data Preprocessing, KNN Implementation and report
2) Asadullah Khan - Worked on Pre Processing, Data visualization and report.
3) Shailesh Pandey - Worked on Feature Importance model and Nueral Network
4) Manu Chaudhary - Worked on Neural Networks implementation