PerfMatch Final Report

Özge Bülbül [*1] Sümeyra Koç [*1] Zeynep Yıldız [*1]

Abstract

This project explores a comprehensive approach to evaluating and predicting player performance in volleyball using machine learning techniques. By employing models such as logistic regression, support vector machines (SVM), random forest classifiers and multilayer perceptron (MLP), we aimed to achieve high accuracy and balanced evaluation metrics in predicting team success and individual player contributions. Data was collected via web scraping from GitHub repositories, followed by rigorous preprocessing and normalization to create a clean, consistent dataset.

Our methodology focused not only on overall success prediction but also on assessing detailed player metrics like kill and point contributions. This analysis, combined with the careful selection of significant features, provides actionable insights for users. To enhance usability, we developed a user-friendly frontend interface, allowing coaches and decision-makers to seamlessly access and apply the results.

While the project demonstrates strong predictive capabilities, it has limitations, such as not accounting for player positions or roles, which we aim to address in future iterations. This comprehensive effort bridges data science with practical application, setting a foundation for further innovation in sports analytics.

1. Introduction

In sports analytics, machine learning (ML) has emerged as a transformative tool for optimizing performance and strategy through data-driven insights. This project addresses the problem of predicting whether a newly formed volleyball team will win or lose, while also estimating the individual contributions of selected players to the team's success.

Motivated by our country's significant achievements in volleyball, we analyzed volleyball datasets and explored a variety of machine learning models. To evaluate the accuracy of our predictions, we adopted a novel approach by testing our models with the rosters of existing teams, allowing us to assess their real-world applicability and refine our methods further.

By integrating player-specific contribution estimations and experimenting with multiple ML models, we aim to provide a comprehensive analysis of how individual performance and team dynamics influence match outcomes. Our approach not only predicts the likelihood of victory but also delivers actionable insights for building successful teams and strategies. The backend of this project, which handles data processing and ML model integration, can be found here (bac, 2024). The frontend, which provides an interactive interface for exploring insights and predictions, is available here (fro, 2024). Through this study, we contribute to the growing field of sports analytics, demonstrating the potential of ML in enhancing decision-making and performance optimization.

2. Related Work

Machine learning has become an effective tool for predicting match outcomes and analyzing performance in team sports like volleyball. A study conducted on NCAA volleyball datasets highlighted that player-level features provide higher accuracy compared to team-level statistics (Sanghvi & Others, 2021). Metrics such as serve percentages, block efficiency, and assists modeled through individual player performances were shown to form a strong foundation for predicting team success.

Additionally, the literature shows that Artificial Neural Networks (ANNs) and Decision Trees are commonly used in volleyball match predictions. While ANNs are effective at learning complex relationships, they are limited in terms of interpretability. In contrast, Decision Trees provide more understandable results but are more sensitive to issues like data imbalance. Inspired by this study, we developed pre-

[*]Equal contribution [1]Department of Artificial Intelligence, University of Hacettepe, Ankara, Turkey. Correspondence to: Özge Bülbül <ozgebulbul@hacettepe.edu.tr>, Sümeyra Koç <sumeyrakoc@hacettepe.edu.tr>, Zeynep Yıldız <zeynepyildiz21@hacettepe.edu.tr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

dictive models using both team and player datasets in our project.

Another study by Lalwani et al. (2022) on Brazilian SuperLiga volleyball data employed powerful models such as Support Vector Machines (SVM) and Deep Neural Networks (DNN) combined with explainability methods like SHAP and ProtoDash (Lalwani & Others, 2022). The study found that SVM achieved the highest accuracy and emphasized the importance of factors such as home-court advantage and previous performance in determining match outcomes. These models not only improved prediction accuracy but also facilitated a better understanding of the key factors influencing team performance. Inspired by this study, we also experimented with the SVM model and chose it for our project due to its superior results. Additionally, we calculated the contribution of each player to the team's performance, allowing us to analyze team dynamics in greater detail.

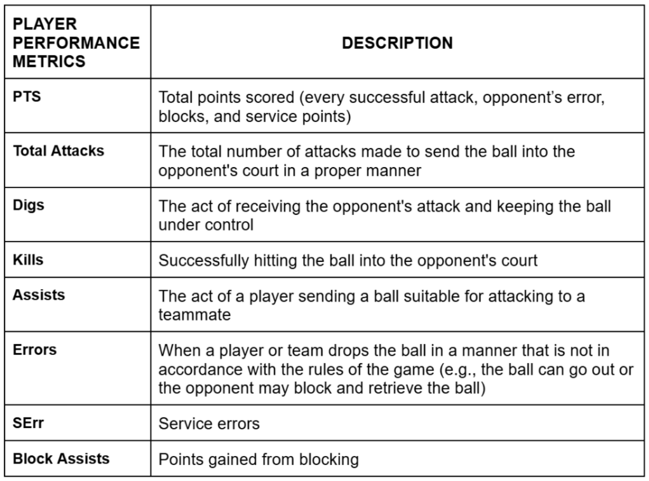
In our project, machine learning models are used to predict the contribution of players to team success. Adopting a similar approach, (Nguyen et al., 2022) used machine learning and deep learning methods to predict the performance and popularity of NBA players. In this study, the players' statistical data and team performances were analyzed to predict the likelihood of players being selected as All-Stars.

(Komar et al., 2023) analyzed men's and women's volleyball Super League matches played in Turkey and Italy between 2013 and 2020 using data mining methods and artificial neural network models. By examining the statistical data of 4,474 matches, they identified key variables influencing match outcomes. Notably, their findings on determining variables affecting match results and associating these variables with player performance provide a solid foundation for our project.

3. The Approach

3.1. Datasets

The datasets used in this project consist of historical volleyball match performance data sourced from a publicly available repository (dat, 2020). Both datasets contain the same features. One contains data related to the teams, while the other contains data related to the players who play for those teams. Player performance metrics (features) of both datasets are showed in the figure 1 The original datasets obtained through web scraping were 160,000 rows in size. After preprocessing and balancing, 30,000 rows remained in each dataset. Due to the need for fast processing, the datasets were stored in Parquet format.



PLAYER PERFORMANCE METRICS	DESCRIPTION
PTS	Total points scored (every successful attack, opponent's error, blocks, and service points)
Total Attacks	The total number of attacks made to send the ball into the opponent's court in a proper manner
Digs	The act of receiving the opponent's attack and keeping the ball under control
Kills	Successfully hitting the ball into the opponent's court
Assists	The act of a player sending a ball suitable for attacking to a teammate
Errors	When a player or team drops the ball in a manner that is not in accordance with the rules of the game (e.g., the ball can go out or the opponent may block and retrieve the ball)
SErr	Service errors
Block Assists	Points gained from blocking

Figure 1. Player Performance Metrics

3.2. Predicting Success of Team

The primary goal of predicting team success involved using historical data to identify the patterns and key features contributing to match outcomes. The team dataset provided comprehensive metrics for each team, such as total assists, kills, blocks, and win/loss statuses, which were used as input features for our models. The success label was defined as "win" (1) or "loss" (0), based on the match outcome (with the threshold 0,5 to determine the label).

We employed three machine learning models to predict team success:

- Random Forest Classifier (RF): Chosen for its robustness and ability to handle both categorical and numerical data effectively. The model also provides feature importance rankings, which helped in identifying critical factors influencing team success.
- Support Vector Machines (SVM): Used with a radial basis function (RBF) kernel, as it performed well in capturing non-linear relationships within the dataset.
- Logistic Regression (LR): A baseline model to compare the performance of more complex algorithms.

Each model was trained using an 80/20 train-test split. The hyperparameters for each model were optimized using grid search, focusing on metrics like accuracy, precision, recall, and F1-score.

The models were evaluated based on their ability to predict the match outcomes of known teams. The SVM model achieved the highest accuracy, outperforming Random Forest and Logistic Regression shown in table 1 . Precision and recall scores were also higher for the SVM, indicating its suitability for this classification problem. We finished this step off by saving our models as joblib files for later use.

Table 1. Algorithm Accuracies for Team-Based Models

Algorithm	**Accuracy**
Random Forest Classifier	86%
Support Vector Machines	87%
Logistic Regression	86%

3.3. Predicting Contribution of Players

After predicting the team success, we concentrated on the work related to building a successful team. We decided to make an AI program that predicts the contribution of players to the performance of the team. After applying a bar plot based on the success column of player dataset, we observed that the data was imbalanced. Although we will not be predicting success of teams, we still want the players of successful and unsuccessful teams to be balanced, because ultimately, the goal of this project is to ensure the success of the team. When Random Undersampling is applied, the player dataset size reduced from 30,800 rows to 26,500 rows, but it is still enough data.

Next, players' actual contributions are calculated using the team dataset. The method is as follows: for a given player, the match they played for their relevant team at that specific time is extracted from the team dataset, then the player's data is divided by the team's data and multiply by 100.

$$\frac{Player'sMetric}{Team'sMetric} \times 100$$

This is possible because the features in both datasets are identical, and for most players, the match record of the team they played for is available in our team dataset. Some players didn't have the match data for the team they played for, which was due to the match containing missing values. Those players are removed from the dataset, which reduced the player data to about 24,000 rows. This way, we were able to calculate the actual contribution values for each player performance metric (assists, kills, digs, etc.) of every player and saved them into a separate dataset.

As a result, we have a dataset containing the player's contribution to the team for each player performance metric (this dataset includes the same features as the player and team datasets, with the addition of a ”ratio” at the end) and we will use this as a data set from now on. The learning and prediction process is as follows: Before deciding which player performance metrics to predict, it was necessary to find the best learning models for each player performance metric. Therefore, we iterated through the columns of the new dataset containing the ratios using a loop, selecting each one as the target value (redefining the training and test datasets each time) and running the learning models. In just one experiment, we obtained eight different models for each player performance metric. More details will be discussed in the Experiment Results section, but MLP, Random Forest, and SVR (a regression extension of the SVM algorithm) learning models were used, and cross-validation was applied to improve generalization.

The results are as follows: The MLP model performed better than the other learning models for most of the player performance metrics. However, when applied with the Cross Validation method, Random Forest produced very high R-squared and low MAE values. Therefore, Random Forest was selected to predict the features where it outperformed the MLP model, while the MLP learning model (without cross-validation) was chosen for the remaining features.

Next, the selection of the player performance metrics to be predicted was undertaken. The details of this process are explained in the ”Selecting Player Performance Metrics” subsection of the Experimental Result section. As a result, Kills, Errors, Total Attacks, Digs, and PTS player performance metrics were selected to be presented to the user. The Random Forest model with Cross Validation predicts the Digs and Errors player performance metrics, while the MLP model predicts the remaining metrics. The evaluation metrics for the relevant player performance metric are provided in the table 2. The user will be able to learn the predicted contributions of the relevant players to the team by simply entering the player names into the program.

Table 2. Evaluation Metrics for Player Performance Metrics

Feature	**MSE**	**RMSE**	**R^2**
Kills	12.75	3.57	0.88
Errors	32.07	5.66	0.97
Total Attacks	10.61	3.25	0.89
Digs	47.55	6.89	0.84
PTS	11.99	3.46	0.84

3.4. Testing Model Output

In this project, the primary goal was to predict whether a newly formed team would be successful. However, since there was no prior data available on the performance of such teams, making a direct prediction was not feasible. Therefore, a different approach was also developed for the testing phase.

Initially, the players of an existing team, whose win/loss status was known, were processed into the appropriate format and fed into the model. The model's predicted outcome (win/lose) was then compared with the team's actual result to evaluate the model's accuracy. However, performing this process for a sufficient number of teams during the testing phase would have been time-consuming and labor-intensive.

To overcome this, an alternative, systematic method was devised to streamline the testing process.

The following steps were implemented during the testing phase:

- **Labeling the Data:** Four years of team and player roster data were utilized. For each year, the win ratio of each team was calculated. Teams with a win ratio greater than 0.5 were labeled as “win (1),” while those with a win ratio below 0.5 were labeled as “lose (0).” These labels were used to represent the overall yearly performance of each team.

- **Segmenting the Data:** In addition to labeling the teams, the data was further segmented into datasets based on yearly performance. This process resulted in three types of datasets:

  - **Winning Teams:** Containing teams with a win ratio greater than 0.5.
  - **Losing Teams:** Containing teams with a win ratio less than 0.5.
  - **Combined Dataset:** Including all teams, regardless of win/loss status.

  This segmentation was repeated for each year, creating a total of 12 datasets (4 years × 3 types of datasets). These datasets were then used to evaluate the model’s performance under different conditions.

- **Processing Player Features:** For each year, the features of the players in each team’s roster were aggregated by calculating the average of all player features. This aggregation allowed individual player data to be converted into team-level data, producing an “average player profile” for each team. As a result, a single row was generated for each team, representing its overall performance for that year.

- **Testing the Model:** The average player profiles were fed into the model, and the model’s predicted results (win/lose) were compared with the actual labels assigned to each team based on their yearly performance. This approach enabled an efficient evaluation of the model’s accuracy and provided reliable results for team-based success predictions.

By implementing this method, the success prediction of newly formed teams was simulated effectively, and the testing process was made more manageable and systematic.

3.5. User Interface

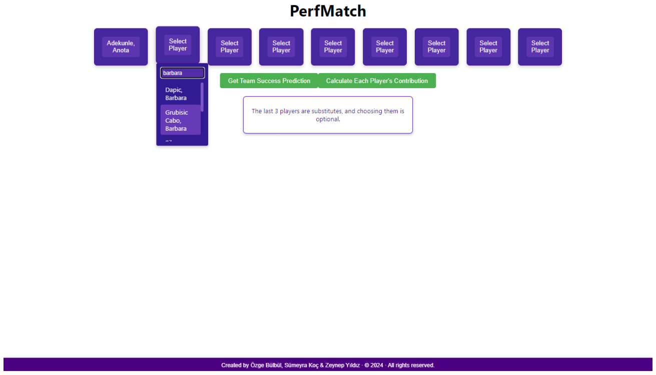
The user interface (UI) for this project is designed to allow users to interact with the model and visualize the results of predictions based on player and team performance data. The UI aims to provide a simple, intuitive experience for users to access the results of the model’s analysis. The frontend of the system is developed with HTML, CSS, and JavaScript to ensure compatibility across different platforms and devices.

The main page features two primary buttons, each corresponding to a different prediction task:

- **Predict Team Success:** This button initiates the process of predicting the success of a newly created team. When selected, the system uses the input features to generate a prediction about the overall success of the team, based on historical performance data.

- **Predict Player Contribution:** This button allows users to predict the contributions of individual players to the team’s success. Users can select players from a list, which will be used to calculate their performance metrics, such as PTS and Kills.

For ease of use, the player selection process includes a **soft search** option as shown in figure 2 . This feature allows users to quickly find players by typing part of the player’s name. As users type, the interface dynamically filters and displays player names that match the input. For example, if a user types ”B,” all players whose names contain the letter ”B” will appear in the search results, making it easier for users to find the player they are interested in without having to scroll through the entire list.

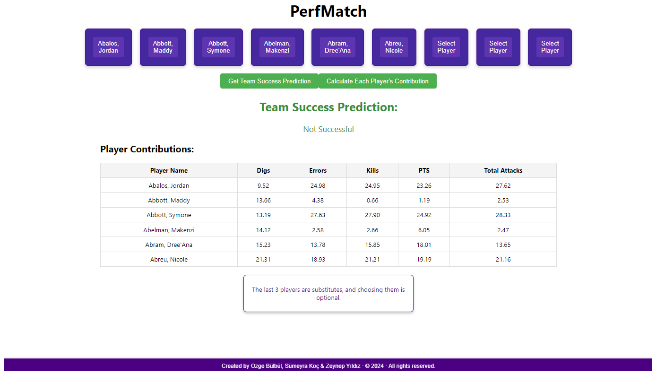


The last 3 players are substitutes, and choosing them is optional.

Created by Özge Bülbül, Sümeyra Koç & Zeynep Yıldız · © 2024 · All rights reserved.

Figure 2. UI Layout showing the player search feature.

The layout is clean and user-friendly, designed to ensure that users can easily navigate between different prediction tasks and view results. A photo of the interface, illustrating the two main buttons and the search results are shown in figure 3



Team Success Prediction:

Not Successful

Player Contributions:

Player Name	Digs	Errors	Kills	PTS	Total Attacks
Abalos, Jordan	9.52	24.98	24.95	23.26	27.62
Abbott, Maddy	13.66	4.38	0.66	1.19	2.53
Abbott, Symone	13.19	27.63	27.90	24.92	28.33
Abelman, Makenzi	14.12	2.58	2.66	6.05	2.47
Abram, Dree'Ana	15.23	13.78	15.85	18.01	13.65
Abreu, Nicole	21.31	18.93	21.21	19.19	21.16

The last 3 players are substitutes, and choosing them is optional.

Created by Özge Bülbül, Sümeyra Koç & Zeynep Yıldız · © 2024 · All rights reserved.

Figure 3. UI Layout showing the player search feature.

This user-friendly design aims to reduce the complexity of interacting with the model, allowing users to efficiently make predictions based on the team or player data provided. With the integration of a soft search and clear buttons for each task, the interface supports both novice and experienced users in generating meaningful predictions about team success and player performance.

4. Experimental Result

4.1. Parameter Tuning

First, it was necessary to find the hyperparameters that yielded the best results for the learning models would be used. For both prediction problems, the SVM with the RBF kernel function gave the best results. The worst results were obtained with the sigmoid kernel function. This is because the dataset contains linear relationships, and the sigmoid kernel typically performs well with datasets separated by non-linear, complex boundaries. The RBF kernel is more compatible with the dataset.

For the regression problem, the hidden layer size parameter of the MLP learning model are tuned using the ”Kills” feature. This was because ”Kills” is one of the features that the model learned best, and based on the correlation matrix, it is the most correlated feature with all other features. The tuning process involved first testing simple dimensions with a single hidden layer, such as (50,), (100,), and (300,), and then testing more complex architectures with two hidden layers that can learn more intricate relationships, such as (50, 50), (100, 50), and (100, 100). MLP model performed best with (100,) according to the figure 4 and figure 5. This is because fewer layers mean fewer parameters, which enables faster training and better generalization.

In the cross-validation method applied for the regression problem, the value of k=10 produced the best results (again, tested on the ”Kills” feature) according the figure 6.

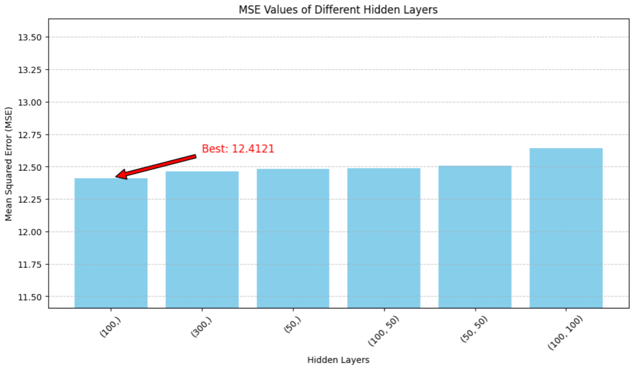


Figure 4. MSE-Hidden Layers Bar Plot

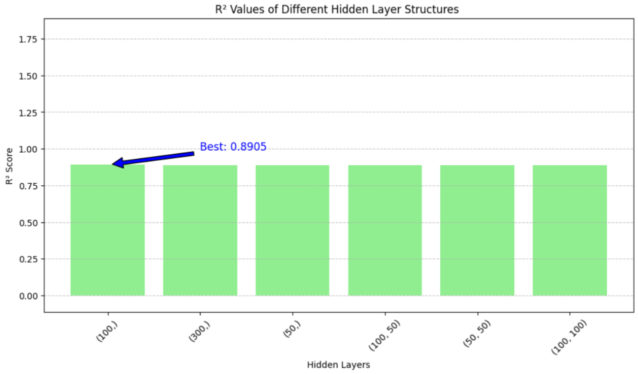


Figure 5. R2-Hidden Layers Bar Plot

Random Forest learning model produced an R² result that reached nearly 90% when cross-validation was used, and the MAE value significantly decreased compared to the version without cross-validation. This change was not observed in the other models (MLP and SVM). The reason for this could be the hyperparameter optimization performed during cross-validation. When CV is used, the hyperparameters of the Random Forest model (such as the number of trees, depth, sample size) can be tested on different data layers, thus achieving better fit. This reduces the MAE and increases R². However, MLP and SVM generally require more complex and delicate parameter tuning.

4.2. Predicting the Success of Newly Created Teams Using a Team Dataset

This section presents the performance analysis of the proposed approach based on experiments conducted using datasets from 2016 to 2019. Initially, the general accuracy of the model across all teams was evaluated. The results are summarized in Table 3.

Although the general accuracy demonstrates moderate per-

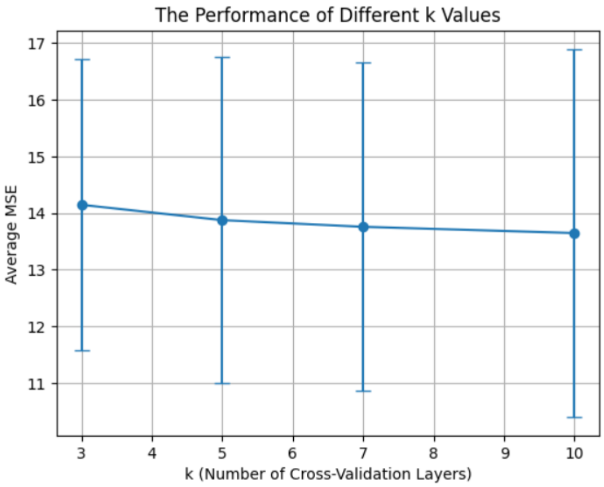


Figure 6. K-MSE Error Bar

Table 3. General Model Accuracy (2016-2019)

Year	Accuracy
2016	0.7006
2017	0.6526
2018	0.6416
2019	0.6536

formance, it does not meet expectations. To better understand the model's strengths and weaknesses, the dataset was divided into two subsets: teams that won more than 50% of their matches (*win datasets*) and teams that won less than 50% (*lose datasets*). The accuracy for each subset was evaluated, and the results are shown in Table 4.

Table 4. Accuracy for Win and Lose Datasets (2016-2019)

Year	Lose Teams Accuracy	Win Teams Accuracy
2016	0.8696	0.7188
2017	0.9200	0.4844
2018	0.9296	0.3731
2019	0.9028	0.3333

An analysis of the dataset sizes reveals that the number of data points for *lose datasets* is slightly higher than that for *win datasets*. This is summarized in Table 5.

Table 5. Dataset Sizes (2016-2019)

Year	Lose Teams Size	Win Teams Size	Total Size
2016	(163, 3)	(165, 3)	(328, 3)
2017	(174, 3)	(159, 3)	(333, 3)
2018	(170, 3)	(164, 3)	(334, 3)
2019	(181, 3)	(151, 3)	(332, 3)

The table indicates that the number of data points for *lose datasets* is slightly higher than for *win datasets*, but this difference cannot be considered a significant imbalance. Other factors may also contribute to the model's higher accuracy for losing teams.

4.2.1. Discussion and Limitations

1. **High Performance on Losing Teams:** The model performs exceptionally well in predicting losing teams, achieving over 90% accuracy in most years. This indicates that the model can effectively identify patterns associated with losing teams.

2. **Low Performance on Winning Teams:** The low accuracy for winning teams suggests that the features used may not sufficiently capture the attributes of successful teams. Additionally, the success of winning teams may depend on more complex relationships that the model struggles to generalize.

3. **Recommendations for Improvement:** - *Alternative Models:* Exploring alternative algorithms or ensemble methods could enhance the model's ability to generalize patterns associated with winning teams.

The experimental results demonstrate that the model excels in predicting losing teams but struggles with winning teams. Improving the model's performance on winning teams requires better feature engineering and optimization techniques. With these improvements, the overall accuracy and balance of predictions are expected to increase significantly.

4.3. Selecting Player Performance Metrics

Due to the learning model not performing well on every feature and some features being unsuitable for contribution prediction, there was a need to select the features to be predicted. In other words, we will choose the player performance metrics that will be presented to the user in our program's user interface. However this will be based on previous research and solid reasoning.

In the paper (Komar et al., 2023) that works with a similar dataset and investigates the contribution of player performance metrics to the overall success of a team, it was found that the SErr feature had no contribution to the outcome. Our model also did not learn the SErr feature well — the MAE score for predicting that feature was 168. The same research shows that attack-related features have a very high contribution to the overall success of the team. According to this paper, the Total Attacks, Kills, and Assists metrics are highly effective in determining team success. Also, when we applied an artificial neural network model to the datasets and assessed each feature's contribution to team success, we found that the PTS feature contributed the most. Following that, attack-related features ranked next, which supports the findings of the relevant paper. Since the ultimate goal of predicting a player's contribution to the team is to ensure the overall success of the team, we will directly apply the results of this paper to our regression problem. In other words, the predictions of players' contributions to the team's SErr player metric will not be presented to the user; instead, the predictions of their contributions to the PTS, Kills, Attacks, and Assists player performance metrics will be presented.

However, there was an issue with the Assists metric, as no model could learn it correctly — the MAE value was 328. As seen in the figure 7, this metric has a different distribution, which is why our models couldn't learn it.

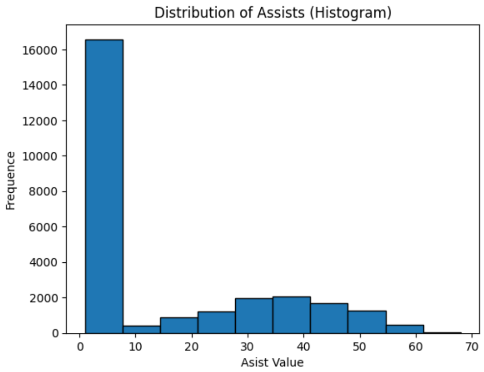
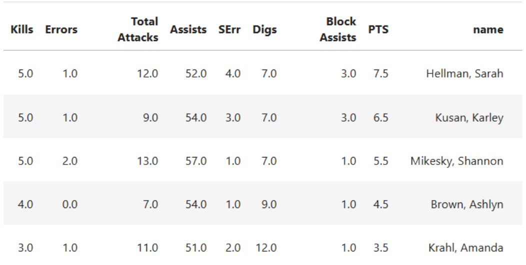


Figure 7. Assists Distribution

The reason for this distribution is that in volleyball, almost all assists come from the setter (Kulüp, 2024). We observed many examples of this in our player dataset, as seen in the figure 8, where, despite having very low values for other metrics, some players have over 50 assists (meaning they contributed almost all of the team's total points).



Kills	Errors	Total Attacks	Assists	SErr	Digs	Block Assists	PTS	name
5.0	1.0	12.0	52.0	4.0	7.0	3.0	7.5	Hellman, Sarah
5.0	1.0	9.0	54.0	3.0	7.0	3.0	6.5	Kusan, Karley
5.0	2.0	13.0	57.0	1.0	7.0	1.0	5.5	Mikesky, Shannon
4.0	0.0	7.0	54.0	1.0	9.0	1.0	4.5	Brown, Ashlyn
3.0	1.0	11.0	51.0	2.0	12.0	1.0	3.5	Krahl, Amanda

Figure 8. Some Players Who Is Setter

As a result, although it has a high contribution to team success, we could not include it among the player performance metrics to be presented, as it is not a suitable metric for the problem of this report. It could be the subject of problems like predicting the setter in a team or estimating the percentage chance of a player being a setter. However, these problems fall outside the scope of this report.The Block Assists metric has the same issues as the Assists metric. The histogram distribution is irregular, and our model couldn't learn it either — the MSE value was 89. The reason for this is, again(Kulüp, 2024), the presence of specific players in the team who perform the Block Assists. Therefore, this metric also falls outside the scope of the problem addressed in this report.

Conclusions

In this study, we explored the prediction of team success and player contributions using machine learning models applied to volleyball datasets. Several models were tested, with each yielding different results based on the problem at hand. The SVM with the RBF kernel function proved to be the most effective for both prediction tasks, while the MLP model showed optimal performance with a single hidden layer, and the Random Forest model significantly improved its accuracy with cross-validation. The analysis revealed that the model was particularly successful in predicting the performance of losing teams, with accuracy rates exceeding 90% for most years. In contrast, the model struggled with predicting winning teams, suggesting that the features used might not be sufficiently capturing the nuances of successful teams. Addressing this limitation through enhanced feature engineering and data balancing could improve the overall accuracy of the model.

Additionally, we addressed the challenges associated with predicting player contributions, particularly in relation to the Assists and Block Assists metrics, which showed irregular distributions due to the role of specific players, such as the setter. Despite this, the model performed well in predicting attack-related metrics, such as Kills and PTS, which are crucial to team success.

Finally, the implementation of a user interface allowed for a practical application of the model's predictions, enabling users to predict both team success and individual player contributions. The user-friendly interface, which includes prediction buttons and a soft search option for player selection, provides an intuitive way for stakeholders to utilize the model's predictions in real-world scenarios.

In future work, we aim to incorporate positional data of players into the model. This addition will allow for a more detailed understanding of player roles on the court, which could enhance the accuracy of the predictions, particularly when analyzing player contributions in specific contexts, such as defense or attack. Exploring alternative models, improving feature selection, and addressing the challenges of predicting specific metrics like Assists and Block Assists could further refine the predictions. Moreover, extending the approach to other sports or domains could further validate the model's generalizability and potential impact.

References

Volleyball ml dataset repository, 2020. URL https://github.com/threewisemonkeys-as/volleyball-ml. Accessed December 2024.

Perfmatch - ain311 project, 2024. URL https://github.com/zgeblbl/perfmatch-ain311project. Created November 2024.

Perfmatch frontend, 2024. URL https://github.com/zgeblbl/perfmatch-frontend. Created December 2024.

Komar, E., Eğrioglu, E., and Semiz, K. Türkiye ve İtalya voleybol süper ligleri 2013-2020 İstatistik verilerinin veri madenciliği yöntemleriyle analizi. *Avrasya Spor Bilimleri Araştırmaları*, 2023. URL https://dergipark.org.tr/tr/download/article-file/2926530.

Kulüp, V. Voleybol pozisyon numaraları ve temel bilgiler, 2024. URL https://voleybolkulup.com.tr/voleybol-pozisyon-numaralari-temel-bilgiler/?utm_source=chatgpt.com. Accessed: 2024-12-28.

Lalwani, V. and Others. Explainable ai for predicting superliga volleyball match outcomes. *arXiv:2206.09258*, 2022. URL https://arxiv.org/abs/2206.09258.

Nguyen, N. H., Nguyen, D. T. A., Ma, B., and Hu, J. The application of machine learning and deep learning in sport: predicting nba players' performance and popularity. *Journal of Information and Telecommunication*, 2022. doi: 10.1080/24751839.2021.1977066.

Sanghvi, M. and Others. Ncaa volleyball match outcome prediction using machine learning techniques. In *Proceedings of ICAIW*, 2021. URL https://ceur-ws.org/Vol-2992/icaiw_wdea_2.pdf.