Machine Learning methods for Volleyball Match Prediction

Atharv Sonwane, Atharv Kirtikar, Himay Patel, Vedant Shah [*]

Abstract

In this study, we propose two datasets for volleyball match data and explore how various Machine Learning models can be used to predict the outcome of a particular match. Data was collected for over 300 teams playing in college level leagues across the United States of America. We scraped individual player performance data as well as team data for seasons from 2016 to 2019. A number of features related to volleyball were processed and used as input to our model. The data was processed in three different ways - Cumulative average (CMA), Simple moving average (SMA) and Exponentially weighted mean (EWM). A number of models were then evaluated for accuracy and F1 score. The accuracy of the model was highest when Logistic Regression with Feature Selection was used on a data set of the Exponentially Weighted Mean of features while including individual player data. This work also explores how player data affects model performance and ultimately shows that machine learning models can perform significantly better than random baselines and thus have the potential to be used in real world scenarios.

1 Introduction

The significance of data analysis in sport has been proliferating with the turn of the centuary. A plethora of sports data is available for us to analyze and convert that into knowledge for use in various industries. This is accomplished by exploiting various machine learning models. Volleyball is played across all levels, national or international, and even college level. Even though the betting market, the main instigator for collection of performance data, is not that prevalent in volleyball, we are able to collect data from organizations such as NCAA (National Collegiate Athletic Association) and FIVB (Fédération Internationale de Volleyball). Presently there are very few studies that have exploited machine

[*] alphabetical order

learning for use in predictional analyses of this sport. This predictional analysis is useful for the coaches to improve their players' performance and to decide which starting line-up is statistically most effective against a particular team. It can also help the team customize their playing style for a particular opponent by analyzing historical data.

In this work we propose two datasets taken from different sources for prediction of outcomes of Volleyball games using previous statistics about the opposing teams. Novelties of our work include the extremely large size of the NCAA dataset provided, our evaluation of multiple different ML models for volleyball game prediction and our exploration of how including player data effects model performance.

2 Related Works

One of the earliest studies to consider AI in the analysis of sports performance was done by Lapham and Bartlett in 1995[1]. The paper concluded that with the use of artificial intelligence techniques in the decision making process, benefits could be reaped in the future. An advent of such studies began after that and many different AI techniques were applied to different sports. K.D. Peterson in 2018[2] predicted the performances of elite track and field sprinters by means of the dynamic, nonlinear mathematical method of recurrent neural networks (RNNs). Their dataset consisted of three years of National Collegiate Athletics Association (NCAA) Division I competitions. Leicht, Gómez and Woods (2017)[3] predicted the match results for Basketball in the Olympics. Classification Tree was chosen over Linear regression, even though it had a lower accuracy, because it provided more flexibility by having greater solving capabilities in non-linear phenomenon.

Machine Learning techniques were used by Tümer and Koçer (2017)[**?**] where they were able to successfully predict league standings of teams in volleyball with 98 percent accuracy using a multi-layer perceptron. They used match results and match location (home/away) as input features to their model. In order to improve the training process, dynamical programming was used in conjunction with a k-nearest neighbour classifier to detect jumps in training data and choose appropriate intensity intervals in the training process. Wenninger, Link and Lames (2020)[**?**] analysed the performance of machine learning models in application to beach volleyball data.

Dataset	Features
FIVB	Set Wise Duration, Set Wise Scores, Attacks, Blocks, Serves, Opposition Errors, Errors, Digs, Receptions, Set, Total Points
NCAA	Kills, Errors, Attacks, Hit Percentage, Assists, Aces, Service Errors, Digs, Reception Errors, Block Solos, Block Assists, Block Errors

Table 1: Features used in both Datasets

3 Dataset

We prepared two different datasets for our study - one consisting of team and player wise performance in the FIVB 2019 Women's World Cup and another similar dataset made up of records from NCAA tournament matches for four seasons from 2016 to 2019. To create these datasets we used the Python libraries BeautifulSoup for HTML parsing and pandas for data wrangling.

3.1 FIVB Women's 2019 World Cup

Data was scraped for a single edition of the tournament, consisting of 11 teams playing in a round robin format. As a result, there were a total of 66 matches. The scraper used the python used the Python library requests to get the HTML source from each web page, and the library BeautifulSoup to extract relevant information.

First, a table containing a list of all women's matches and their outcomes was scraped from the site.

Next, statistics for each match was scraped. This data was scraped from the women's schedule section in FIVB's official website. The scraped data for each match includes the following variables: Match Number, Team Names, the duration and scores for each set, and the number of Attacks, Blocks, Serves and Errors for the match.

Data for each individual player's performance over the tournament was also scraped.

The cumulative values of these variables for all matches (excluding the match for which the result is to be predicted) were calculated for each team to give us the final parameters to be used in our model. These can be seen in Table 1. As the dataset was comparatively small, these cumulative features were calculated during model application itself.

3.2 NCAA Women's Volleyball

The NCAA is a college level tournament which is played by about 320 - 340 (varying) teams each year. The tournament consists of group stages followed by playoffs. However, as we are using cumulative features in our model, the format of the tournament can be ignored.

We collected data for matches over 4 seasons amounting to about 9000 data points per year and about 32600 data points for all the years combined in the final trainable datasets. Each match has records for all the features given in Table 1 as for both teams and each player. The NCAA data was more difficult to collect due to the sheer size of the data. Data for individual player performances and team performances were located on different pages. As a result, scraping was done in two stages.

The first part involved scraping for each team. A table consisting of participating teams with hyperlinks to each team page was obtained by selecting sport, year, and number of teams in a form on stats.ncaa.org. After we manually downloaded the HTML source of the form page, a team scraper extracted the hyperlinks for each team using BeautifulSoup. It then downloaded the HTML and extracted the required tabular data for each team web page using pandas. The data extracted was the game by game performance of the team over the season.

The second part involved scraping each player's statistics. Each team page had a hyperlinked list of team players, so the player scraper accessed each individual player's page and scraped the necessary data.

As the amount of data to be collected was large, processing of the data was done separately. Each individual team data file was first cleaned, during which the it was formatted to be usable for the model. Certain irrelevant features were dropped.

After cleaning, the data was processed in three different ways. Cumulative average (CMA) involved taking the mean of all statistics in matches up till the match under consideration as shown in Equation 1. Simple moving average (SMA) involves taking the average from only the last k matches as shown in Equation 2. Exponentially weighted mean (EWM) uses the formulation show in Equation 3 to take a weighted average with diminishing weights allotted to older matches.

$$\mathrm{CMA}_t = \frac{1}{t}\sum_{i=0}^{t-1} x_i \tag{1}$$

$$\mathrm{SMA}_t^k = \frac{1}{t}\sum_{i=t-k}^{t-1} x_i \tag{2}$$

$$\mathrm{EWM}_t^{\alpha} = \begin{cases} 0 & \text{if } i = 0 \\ \alpha x_t + (1-\alpha)\mathrm{EMA}_{t-1} & \text{if } t > 0 \end{cases} \tag{3}$$

In the final dataset, for each of the processed tables we have the following format -

1. Each row denotes a match
2. Cumulative match wise features for both teams are included
3. Cumulative features for top 12 players (in terms of games played) for both teams are included.
4. The result of match 0 (Team A win) or 1 (Team B win) is included

Note that the cumulative features do not include the stats of the match for which we are predicting the result. The aim is use these features to predict that match's result. This means that models trained on this dataset can be used in real world scenarios where we only know information about teams leading upto a the match we want to predict the results for.

Our dataset is made up of two parts a with-player dataset which contains the accumulated player information along with aggregate information for the team and a without-player dataset which consists of only the team information. We performe tests on both to see how much difference is made by including player data. Our reason for trying this is that we feel player data may provide further information about the makeup of the team (is it made of aggresive or defensive players / how do different players gel together to bolster team performance) which cannot be extracted from aggregate team data.

We have also prepared a dataset for the combined data of all the four years. This dataset has been further modified by making combined(yearwise) datasets with simple moving averages, cumulative averages and exponential averages over all the years. Although we havent used this for our analysis, we feel that it can be used in future work.

4 Methodology

We set up an evaluation pipeline which uses Stratified K-Folds cross-validation for test train split which is a variation of k fold cross validate where in balance of each class is preserved while creating the test train split. It is invariant to class label as it depends upon the data set ordering. Since our match prediction problem boils down to binary classification problem, having imbalanced classes during training may have biased the model towards a particular outcome, which is why we used Stratified version of K Fold Cross Validation.

In regular k-fold CV, the training set is split into k smaller sets and a model is trained using k-1 of the folds as training data and validated using the remaining part of the data. This gives a evaluation of the model robust to anomalies due to random seeding.

The metrics we use are Accuracy and F1 score. Since our problem is inherently balanced (there will be equal number of win and loss outcomes in the dataset), we do not use metrics such as ROC AUC Score. While predicting matches we care about how many positives (wins) and negatives (losses) we predicted correctly and accuracy does exactly this. F1-score is used when the False Negatives and False Positives are crucial and balances both recall and precision. We use these metrics to get a complete idea about the model's performance and not be biased in our analysis of the result. While experimenting we found that both these metrics tended to agree.

We evaluate the following models on the data with default parameters to establish a baseline. The results are given in table

- XGBoost
- Random Forest
- Logistic Regression
- Linear Support Vector Classifier
- Decision Tree

From these preliminary results we see that **Logistic Regression** and **Random Forest** generally perform better than the other methods giving close to 80% accuracy. After establishing this, we use evaluate which features work best with these models using two features selection methods - select k best features based on the ANOVA correlation coefficient and selecting features based on the importance the classifier assigns to them while training.

We feel that this is a critical part of our work since our dataset of combined team and player data has 338 features in total. Since many of these are likely

Model	CMA F1 Score	CMA Accuracy	SMA F1 Score	SMA Accuracy	EWM F1 Score	EWM Accuracy
XGBoost	73.88%	73.12%	74.64%	73.74 %	78.01%	77.30%
Random Forest	74.00%	73.12%	75.13%	73.92%	78.31%	77.10%
Logistic Regression	73.09%	72.60%	76.08%	73.63%	78.27%	77.62%
Linear SVC	72.43%	72.00%	75.13%	72.84%	77.79%	77.10%
Decision Tree	65.17%	64.47%	65.93%	65.16%	69.53%	68.61%
MLP	69.98%	69.11%	70.42%	69.75%	75.16%	74.51%

Table 2: Baseline Model Performances on Datasets With Player Data

to be correlated (team performance will be highly correlated with player performance), such feature selection techniques can drastically reduce the number of features being used. One basic method which was tried was Variance Thresholding, however we found that this did not reduce the feature set size even with high values of variance threshold. One reason for this not working while other methods do is that the high correlation between features may not be explicit when computing individual correlations but may arise out of group associations (team feature being associated with the group of player features).

For training and evaluation we make use of the scikit-learn [4] framework which provides standard implementation of mutiple ML methods and evaluation techniques.

5 Results

Here we present the performance of various standard Machine Leaning models trained on both with-player dataset and without-player dataset where both were processed in different forms of averaging as described in section 3. The baselines results presented in Table 2 and Table 3 are averages of performances of the particular configuration across all 4 seasons of NCAA data.

We can see that most of the models perform better on the without-player dataset. This maybe due to the highly correlated nature of the with-player dataset. We also observe that EWM seems to be a much better averaging method as opposed to SMA and CMA consistently giving better performance on all models. In terms of Models, Logistic Regression and Random Forest are the best performers. Overall we see that models give significantly better accuracy than a random choice (50%).

For the first feature selection method (selecting k best features based on ANOVA score), to evaluate the optimal number of features we evaluated performance for size of feature set ranging from 10 to 310 taking steps of 50 at a time. Results of this can be found in Figure 1.

Model	CMA F1 Score	CMA Accuracy	SMA F1 Score	SMA Accuracy	EWM F1 Score	EWM Accuracy
XGBoost	73.49%	73.02%	74.68%	73.75%	78.34%	77.58%
Random Forest	74.73%	73.96%	76.03%	75.18%	79.25%	78.51%
Logistic Regression	76.36%	75.46%	77.34%	76.49%	80.67%	79.99%
Linear SVC	76.39%	75.46%	77.39%	76.46%	80.70%	79.99%
Decision Tree	66.24%	66.04%	67.45%	66.64%	71.33%	70.57%
MLP	73.16%	72.57%	74.52%	73.67%	78.23%	77.61%

Table 3: Baseline Model Performances on Datasets Without Player Data

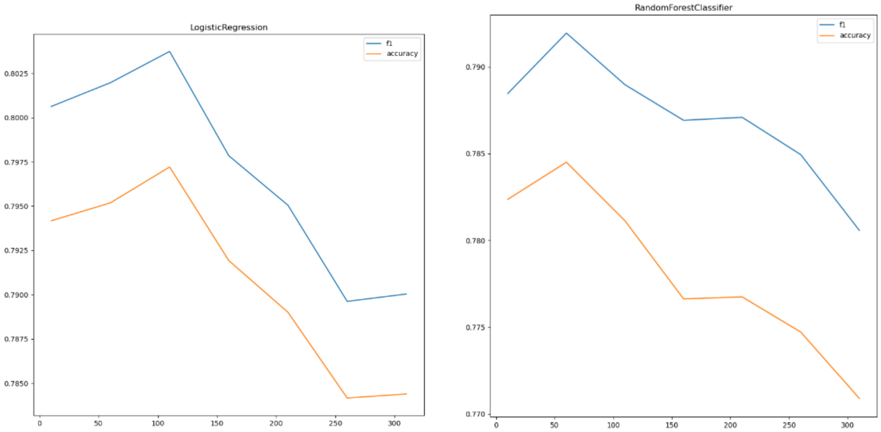

Figure 1: Model Performance vs Number of Features Selected

Model	Features	F1 Score	Accuracy
Random Forest	8	78.14%	77.72%
Logistic Regression	10	80.37%	79.72%

Table 4: Model Performances on Without-Player Dataset using Feature Selection

Model	Features	F1 Score	Accuracy
Random Forest	44	79.23%	78.55%
Logistic Regression	102	80.65%	80.01%

Table 5: Model Performances on With-Player Dataset using Feature Selection

Results of the second method of feature selection (based on importance given to feature by the model) can be found in Table 4 and Table 5. We find that Logistic Regression with a set of 102 features for With-Team Dataset gives 80.65% accuracy which was the highest overall accuracy of any configuration. In fact, while the baseline scores show that the adding player data does not improve performance, we see that with feature selection amongst the extended the dataset, we *can* improve performance. While Logistic Regression shows a marked improvement with reduction of the number of features, Random Forrest Classifier shows similar performance to that on the original dataset. We also experimented with hyper-parameter tuning however found that no significant gains in performance could be observed.

6 Conclusion

After trying a number of models, we find that Machine Learning models can be used to predict the outcomes of volleyball matches with reasonable accuracy. We also see that player data does not have as big an impact as we had anticipated. Our models data set has a large number of data points on which we applied a number of models. Out of these, Logistic Regression with Feature Selection used on a data set of the Exponentially Weighted Mean resulted in over 80% of matches being called correctly.

Our work, however, is limited and can be significantly expanded in the future, allowing us not only to predict a match outcome but also individual player performances. In addition, our EWM models used a fixed α of 0.2 and our SMA models used a fixed window of 10 previous games. These hyper-parameters may be tweaked in the future to obtain an even better result.

Models like these can significantly improve the way coaches can choose and analyse their teams, as the players chosen are taken into consideration by our model. The positions of players are only indirectly considered by our model based on the points scored, but can also be directly be used as a parameter.

Finally, our results and those of others may be compared to the results of human experts in the game, which would result in further improvements in the future Overall, there is much scope for expansion in the area of match analysis using Machine Learning.

References

[1] A.C. Lapham and R.M. Bartlett. The use of artificial intelligence in the analysis of sports performance: A review of applications in human gait analysis and future directions for sports biomechanics. *Journal of Sports Sciences*, 13(3):229–237, 1995. PMID: 7563290.

[2] Kyle Peterson. Recurrent neural network to forecast sprint performance. *Applied Artificial Intelligence*, 32:1–15, 08 2018.

[3] Shaoliang Zhang, Alberto Lorenzo, Carl T Woods, Anthony S Leicht, and Miguel-Angel Gómez. Evolution of game-play characteristics within-season for the national basketball association. *International Journal of Sports Science & Coaching*, 14(3):355–362, 2019.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.