

# **Predicting Soccer Matches Outcome using Machine Learning Models**

## **Final Report: CSDS 312 - Data Science System**

Hieu Hoang, Andrew Tran, Duc Huy Nguyen, Loc Nguyen, Lam Nguyen, Jerry Xiao, Emil Ekambaram, Zach Youssef

### **Table of Contents**

<b>Predicting Soccer Matches Outcome using Machine Learning Models.....</b>	<b>1</b>
<b>Final Report: CSDS 312 - Data Science System.....</b>	<b>1</b>
<b>1) Introduction.....</b>	<b>1</b>
<b>1.1) Problems and Concepts.....</b>	<b>1</b>
<b>1.2) Background.....</b>	<b>2</b>
<b>2) Data and Data Preprocessing.....</b>	<b>3</b>
<b>2.1) Raw Data.....</b>	<b>3</b>
<b>2.2) Data Cleaning.....</b>	<b>4</b>
<b>2.3) Creating Features.....</b>	<b>4</b>
<b>2.4) Visualization.....</b>	<b>5</b>
<b>3) Methods and Predicting Models.....</b>	<b>8</b>
<b>3.1) What are Machine Learning Models.....</b>	<b>8</b>
<b>3.2) Effective Machine Learning Models for Sports and Soccer Prediction.....</b>	<b>9</b>
<b>3.3) Linear Regression.....</b>	<b>10</b>
<b>3.4) Random Forest Algorithm.....</b>	<b>11</b>
<b>3.5) Support Vector Machine optimized by cuML.....</b>	<b>12</b>
<b>4) Final Evaluation.....</b>	<b>13</b>
<b>4.1) Random Forest Classifier.....</b>	<b>14</b>
<b>4.2) Linear Regression.....</b>	<b>15</b>
<b>4.3) Conclusion.....</b>	<b>16</b>
<b>References.....</b>	<b>17</b>

# **1) Introduction**

## **1.1) Problems and Concepts**

In the modern era, people are more concerned with information that is backed up with statistics, especially with soccer being such a popular sport. Having models that can predict the winning rate of a match based on huge past data is beneficial for researchers, news outlets, team coaches, and matchmakers when arranging which two teams should face each other, and more. It also prepares people with the expected outcomes and strength of each team based on statistics so they know what to expect from the match.

If the final result of a match is different from the prediction, viewers would appreciate it more as the lower hand must have some special strategies or luck in order to win the match not following the data prediction, or vice versa for the stronger team. From the result, the board management of both teams can alter their play style accordingly. The idea of predicting the outcome of a football match is not new; several research has been done on this problem. However, the outcome does not reflect clearly the winning percentage of a match and we also want to try utilizing skills with random forest, logistic regression, and support vector machines to predict the winning and losing rate of a match.

In conclusion, having a tool that can help predict the winning and losing rate of a match will provide useful information for broadcasters and people watching the game knowing beforehand the expected result, which would result in more excitement with miracles happening when their underdog team won or their expected stronger team won. It is also very useful for coaches of teams and soccer analysis for future purposes across the globe.

## **1.2) Background**

In recent times, the field of data science and machine learning has made significant strides, providing new avenues for predicting the outcomes of soccer matches. The proliferation of data on players, teams, and leagues has enabled analysts to construct intricate models that consider a multitude of factors that can impact the result of a game [Bunker & Thabtah (2019). A machine learning framework for sport result prediction].

One prevalent method for forecasting soccer match outcomes involves employing statistical models that utilize historical data on team performance. These models can range from a basic linear regression that incorporates fundamental team statistics such as goals scored and conceded to predict the result of a game, to an elaborate machine

learning algorithm that encompasses hundreds of variables including the caliber of the opposing team, weather conditions, and injury status of key players [Bunker & Thabtah (2019). A machine learning framework for sport result prediction].

An alternative approach to predicting soccer match outcomes is through the use of neural networks, which are versatile and potent machine learning models capable of discerning complex patterns within data. Neural networks operate by “learning” from vast quantities of data and applying this knowledge to make predictions on new data. In the context of soccer match prediction, neural networks can be trained on historical data to recognize patterns indicative of which team is more likely to emerge victorious in a given game [Tiwari et al. (2020). Football Match Result Prediction Using Neural Networks and Deep Learning].

A third approach involves combining statistical models with machine learning techniques. For instance, one could construct a statistical model that integrates basic team statistics such as goals scored and conceded, then employ machine learning to determine which variables are most significant in predicting the outcome of a game. This hybrid approach can prove highly effective as it enables analysts to leverage the strengths of both statistical models and machine learning techniques [Bunker & Thabtah (2019). A machine learning framework for sport result prediction].

In this paper we are going to discuss the above-mentioned methods with the analysis of the results models, graph and how we get to the conclusion.

## 2) Data and Data Preprocessing

### 1.1) Raw Data

Data was collected from official football statistics archives, betting websites, and FIFA games, which assign attributes to teams and players such as a ranking. All of this data was publicly posted onto kaggle. For our purposes, we are looking at the Match and Team\_attributes datasets, which are described below:

**Match:** 115 columns, 26k rows

- Match, country, home team, away team, and league id
- Season, stage, and date
- All players on each team and their positions
- Match stats: goals, shots on, shots off, fouls, cards, crosses, corners, possessions
- Betting odds from 10 different betting websites (away odds, home odds, draw, odds)

**Team\_Attributes:** 25 columns, 1458 rows

- Team id
- Date that the attributes were assigned
- A list of different strategies that each team uses for different games. Each attribute is assigned a value.
  - buildUpPlaySpeed
    - How much the home team relies on kicking a far ball to get it to the opposing side
  - buildUpPlayDribbling
    - How much the home team relies on dribbling to get the ball to the opposing side
  - buildUpPlayPassing
    - How much the home team relies on passing to get the ball to the opposing side
  - chanceCreationPassing
    - How much the home team relies on passing to get the ball in the opposing goal
  - chanceCreationCrossing
    - How much the home team relies on crossing the ball to get it in the opposing goal
  - chanceCreationShooting
    - How much the home team relies on passing to get the ball in the opposing goal
  - defencePressure
    - How much defensive pressure the home team exerts
  - defenceAggression
    - How aggressive the home team is in its defense
  - defenceTeamWidth
    - How wide the home team's defense formation is across the field

## **1.2) Data Cleaning**

All features regarding statistics of the match were dropped because they are details obtained while a game is being played, and therefore would not be obtainable for a prediction. Then, na's were removed from the dataset. This was done by first dropping all features with too many na values. This included information from three different betting websites' predictions. The remaining rows that included na's were then dropped. Overall, by dropping na's, the Match dataset size was decreased from 25,979 to 19,561, which is about a 25% decrease in the size of the dataset. Then, dummy variables for home win, loss, and draw were made.

## **1.3) Creating Features**

Matches\_played, wins, and win\_rate were obtained by collecting the total number of matches played and wins for each team. Then, for each team, the average strategy choices were determined. For example, a team may have defencePressure ratings that differ across several games. These ratings would be averaged and assigned to that team. This was done for each of the strategies. Then, average player ratings were calculated for each team and were assigned to the team. All of these features were added to the Match dataset, where all the statistics were added for both the home and away teams.

## 1.4) Visualization

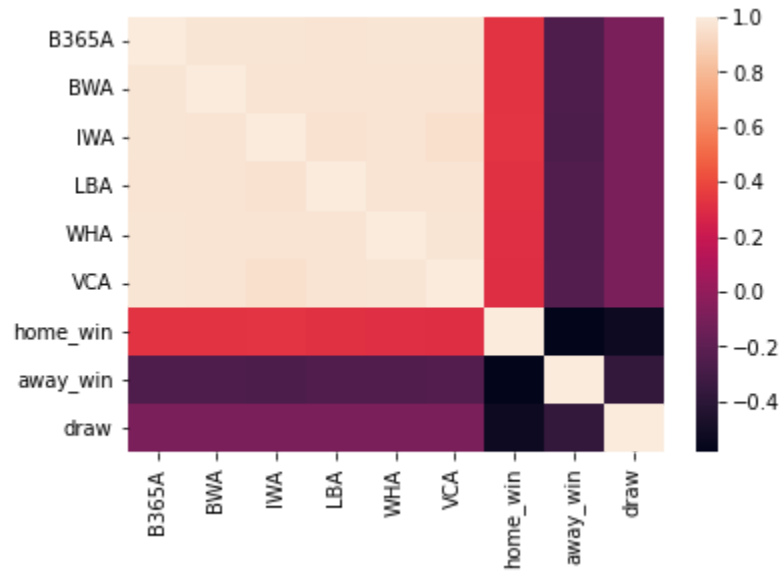


Figure 1a: heatmap of correlation matrix for betting websites away odds

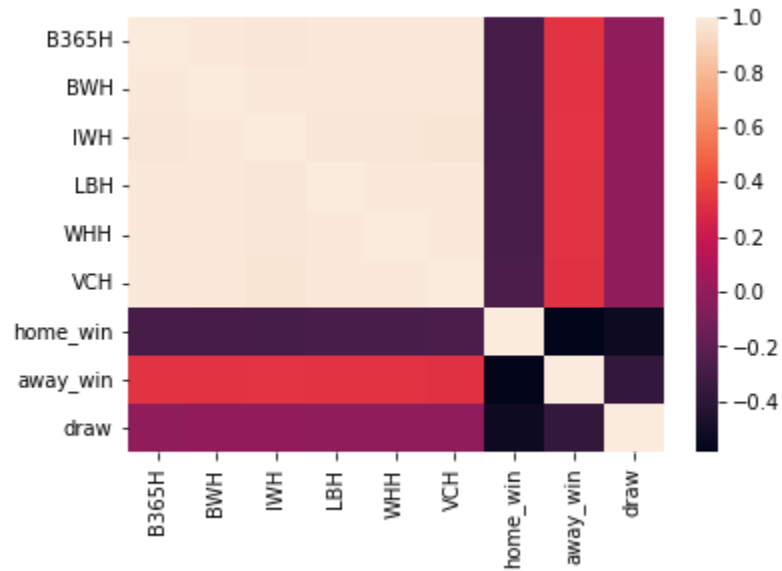


Figure 1b: heatmap of correlation matrix for betting websites home odds

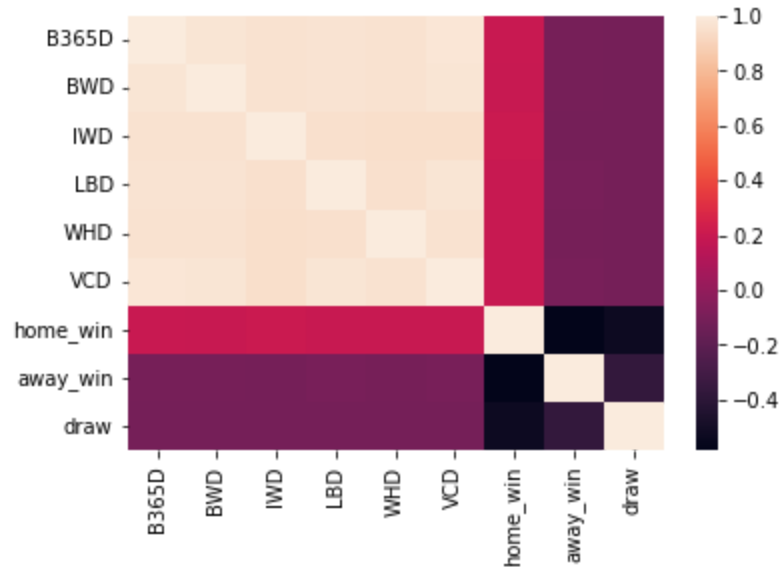


Figure 2c: heatmap of correlation matrix for betting websites draw odds

According to figures 1a, 1b, and 1c, all betting websites' odds are very highly correlated. Based on these results, all of the betting odds for home, away, and draw can be averaged. This results in attributes avg\_betting\_odds\_A, avg\_betting\_odds\_H, and avg\_betting\_odds\_D. Then, all of the non-average betting odds were dropped from the dataset to remove redundancy.

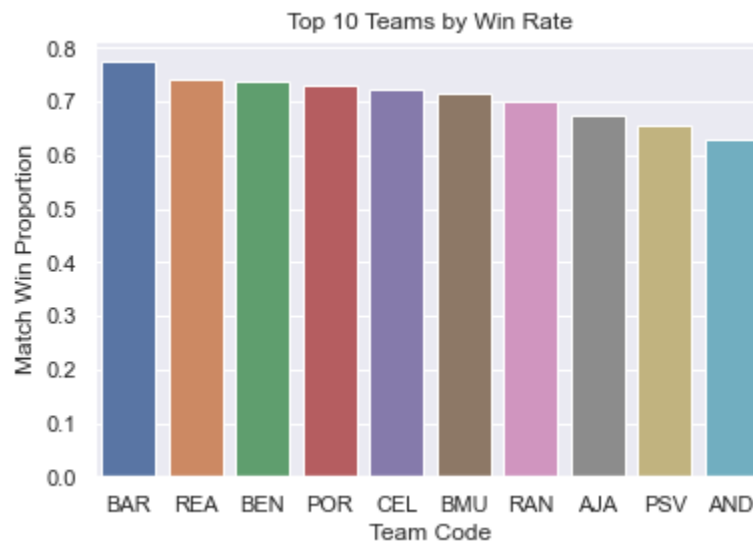


Figure 2: Heatmap of correlation matrix of the team statistics

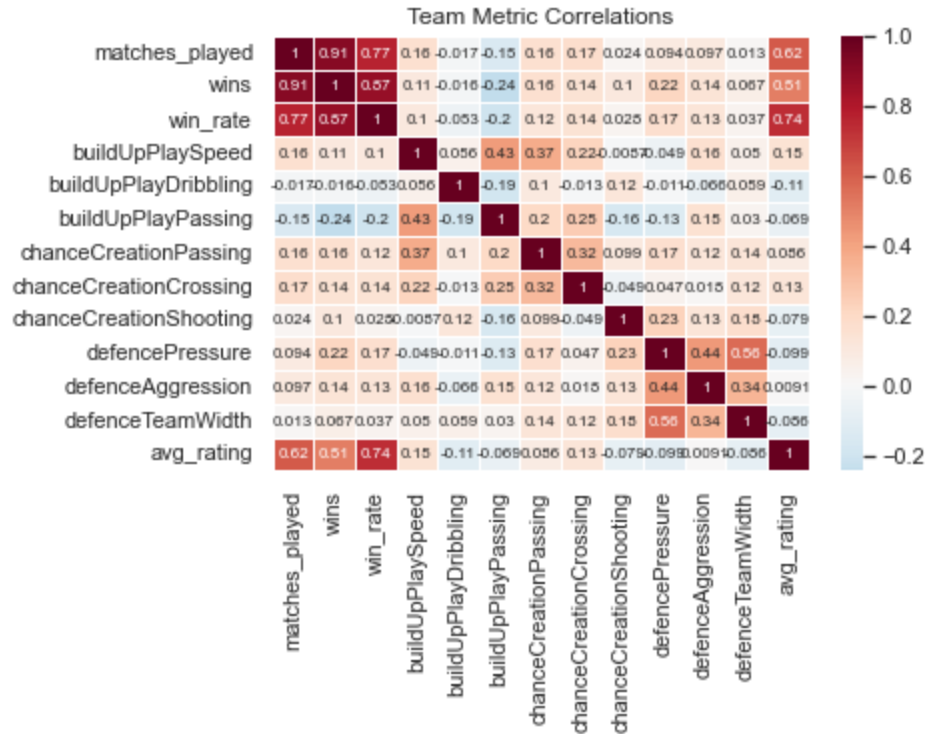


Figure 3: Heatmap of correlation matrix of the team statistics

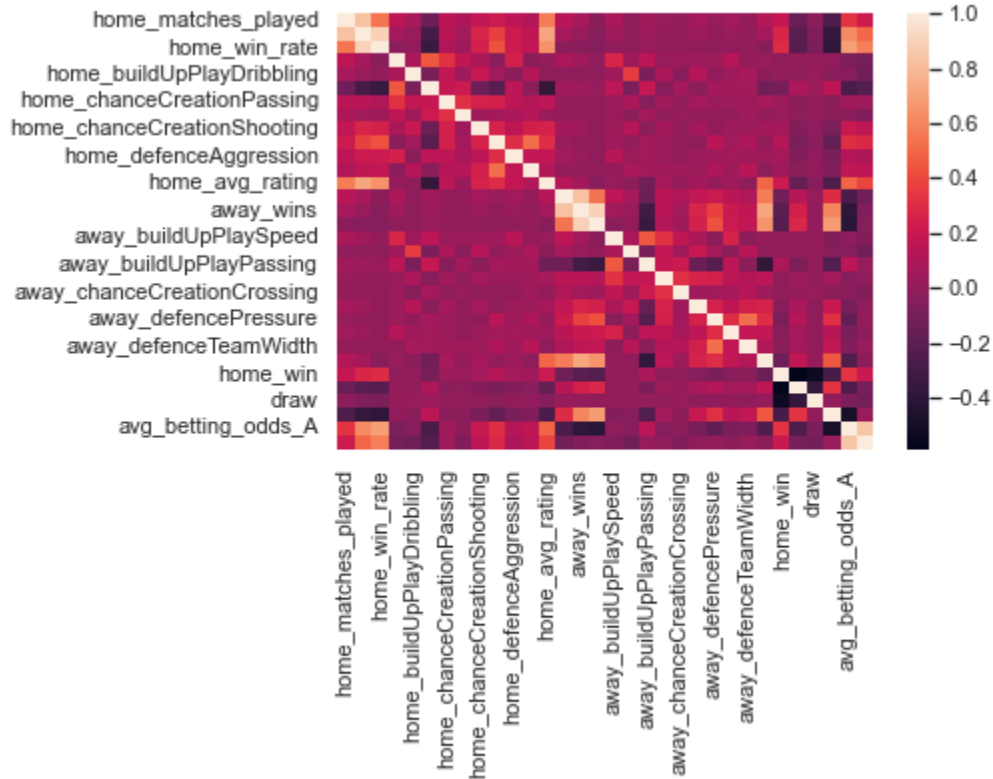


Figure 4a: heatmap of correlation matrix of the final Match dataset

home_win	1.000000
avg_betting_odds_A	0.318551
home_win_rate	0.289633
home_wins	0.263168
home_avg_rating	0.200932
avg_betting_odds_D	0.196447
home_matches_played	0.169864
home_defencePressure	0.117022
away_buildUpPlayPassing	0.080525
home_chanceCreationShooting	0.063816
home_defenceAggression	0.058065
home_defenceTeamWidth	0.042536
home_chanceCreationPassing	0.030571
home_chanceCreationCrossing	0.015195
away_buildUpPlaySpeed	0.002397
home_buildUpPlayDribbling	-0.001371
away_buildUpPlayDribbling	-0.003929
home_buildUpPlaySpeed	-0.011574
away_chanceCreationCrossing	-0.014050
away_chanceCreationPassing	-0.017467
away_defenceTeamWidth	-0.036148
away_defenceAggression	-0.045290
away_chanceCreationShooting	-0.048918
away_defencePressure	-0.095475
home_buildUpPlayPassing	-0.106879
away_matches_played	-0.132153
away_avg_rating	-0.149224
away_wins	-0.213563
away_win_rate	-0.236297
avg_betting_odds_H	-0.284250
draw	-0.535598
away_win	-0.586428

Figure 4b: correlations between every feature and home\_win

From figure 3, matches played and average player ranking are the most heavily correlated with a team's win rate. From figure 4, all of the features have a low correlation with home\_win(not including draw and away\_win), with all of them having an absolute value of less than .32. The most notable correlations are avg\_betting\_odds\_A(0.318551), home\_win\_rate(0.289633), avg\_betting\_odds\_H(-0.284250), home\_wins(0.263168), and away\_win\_rate(-0.236297).

### 3) Methods and Predicting Models

#### 2.1) What are Machine Learning Models

Machine learning is a subset of artificial intelligence that involves the development of algorithms that can learn from and make predictions on data. These algorithms improve their performance as the amount of data they are exposed to increases (Alpaydin, E. (2020). Introduction to machine learning (4th ed.). MIT press).

Machine learning models have been widely used in sports prediction due to their ability to handle large amounts of data and identify complex patterns and relationships between



variables (Bunker, R. P., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1), 27-33). For example, a study by Bunker and Thabtah (2019) used machine learning techniques to predict the outcome of sports events based on historical performance data, match results, and player statistics.

In soccer prediction, machine learning models have been used to predict the outcome of matches based on various factors such as team performance, player statistics, and historical data (Rodrigues, F., & Pinto (2022). Prediction of football match results with Machine Learning. *Procedia Computer Science*, 204, 463-470). For example, a study by Rodrigues and Pinto (2022) used machine learning methods that take multiple statistics of previous matches and attributes of players from both teams as inputs to predict the outcome of football matches.

Machine learning models are effective for sports prediction because they can handle large amounts of data and identify complex patterns and relationships between variables. They can also be trained on historical data to improve their predictive accuracy over time. In terms of success rate, studies have reported prediction accuracy around the 70-75% mark for machine learning models in sports prediction (Joseph, A., Fenton, N. E., & Neil, M. (2019). Sports prediction and betting models in the machine learning age: The case of Tennis. *Journal of Sports Analytics*, 5(2), 95-113)

## **2.2) Effective Machine Learning Models for Sports and Soccer Prediction**

Predicting the outcome of a soccer match is a challenging task due to the complexity and randomness of the game. However, machine learning algorithms such as random forest, logistic regression, and support vector machine optimized by cuML have shown promising results in predicting soccer match outcomes.

Random Forest is an ensemble learning method that constructs multiple decision trees and outputs the class that is the mode of the classes of the individual trees. It has been shown to be effective in sports prediction due to its ability to handle large amounts of data and identify complex patterns and relationships between variables (Joseph, A., Fenton, N. E., & Neil, M. (2019). Sports prediction and betting models in the machine learning age: The case of Tennis. *Journal of Sports Analytics*, 5(2), 95-113).

Linear regression is an effective model for sports and soccer prediction because it can be used to predict a target variable in previously unseen data. For example, linear regression and Bayesian linear regression were the best-performing models on the 2016 data set, predicting the winning score to within 3 shots 67% of the time (Bunker & Thabtah, 2019)<sup>1</sup>. In another study, a simple linear regression model was used to predict a team's winning percentage based on the difference between their runs scored and runs allowed (Society for American Baseball Research)

Support Vector Machine (SVM) is a machine learning algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate different classes. It is particularly useful for classification and regression analysis.

SVM optimized by cuML is beneficial for sports and soccer prediction because it can handle large amounts of data and identify complex patterns and relationships between variables. cuML is a suite of fast, GPU-accelerated machine learning algorithms designed for data science and analytical tasks. It provides practitioners with the easy fit-predict-transform paradigm without ever having to program on a GPU. As data gets larger, algorithms running on a CPU becomes slow and cumbersome. RAPIDS provides users a streamlined approach where data is initially loaded in the GPU, and compute tasks can be performed on it directly. For large datasets, these GPU-based implementations can complete 10-50x faster than their CPU equivalents (RAPIDS AI Team (2021). cuML - RAPIDS Machine Learning Library. GitHub repository).

In the next few sections, we are going to evaluate feasible models in order to find the next evaluative method.

### **2.3) Linear Regression**

In the context of sports and the prediction of soccer results, linear regression is a statistical technique that has been used to examine the relationship between two or more variables (Bunker & Thabtah, 2017). Researchers can determine which aspects are most crucial in deciding the outcome of a game and utilize this knowledge to generate more precise predictions by modeling the relationship between numerous factors and the outcome of a game (Press, n.d.; Linehan, n.d.).

Several studies have utilized linear regression for sports prediction. For instance, Press (n.d.) utilized linear regression to examine how several variables, such as home field

advantage, club strength, and recent form, influenced the results of English Premier League football games. The findings demonstrated that these variables were reliable predictors of match outcomes and that the model was capable of doing so. Similar to this, Linehan (n.d.) utilized linear regression to examine the correlation between several aspects of Premier League football games, including shots on target, possession, and pass accuracy. The findings demonstrated that these variables were reliable predictors of match outcomes and that the model was capable of doing so.

NHSJS (2020) utilized linear regression to examine the link between points and goal difference in Premier League standings in another study. The findings revealed a substantial positive link between points and goal difference, implying that teams with a bigger goal difference had more points. Finally, Bunker and Thabtah (2017) predicted the outcome of sports contests using linear regression as part of a machine learning system. When paired with other machine learning approaches, the results demonstrated that linear regression was an effective way for predicting match outcomes.

Linear Regression has been shown to be an effective method among data scientists and researchers investigating the correlation between numerous factors from players' performance and other metrics that affect the winning or losing rate of the match that can be beneficial for many to improve our understanding of sports and soccer in general.

## **2.4) Random Forest Algorithm**


Random forest is an ensemble learning method that constructs multiple decision trees and outputs the class that is the mode of the classes of the individual trees. It has been shown to be effective in sports prediction due to its ability to handle large amounts of data and identify complex patterns and relationships between variables.

In soccer prediction, random forest can be used to predict the outcome of matches based on various factors such as team performance, player statistics, and historical data. For example, a study by Lang et al. (2022) used random forest along with other machine learning methods to predict the in-game status of soccer matches using spatiotemporal player tracking data. The results showed up to 92% accuracy in predicting the in-game status in previously unknown matches on frame level [Lang et al. (2022)]. Predicting the in-game status in soccer with machine learning using spatiotemporal player tracking data].

Random forest is effective for sports prediction because it can handle large amounts of data and identify complex patterns and relationships between variables. It also has a high prediction accuracy compared to other methods such as logistic regression [Al-Hamadani et al. (2020). Random forest algorithm to identify factors associated with sports-related dental injuries].

In conclusion, random forest is an effective method for sports prediction due to its ability to handle large amounts of data and identify complex patterns and relationships between variables. It has been shown to have high prediction accuracy in predicting soccer match outcomes.

## **2.5) Support Vector Machine optimized by cuML**

Support vector machine (SVM) is a machine learning algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate different classes. It is particularly useful for classification and regression analysis. 

SVM has been shown to be effective in sports prediction due to its ability to handle large amounts of data and identify complex patterns and relationships between variables. For example, a study by Zhu and Sun (2019) used SVM to design an athlete performance prediction model and found that the athletes' performance prediction results of the designed model were more reliable and had higher prediction accuracy compared to other models [Zhu, P., & Sun, F. (2019). Sports Athletes' Performance Prediction Model Based on Machine Learning Algorithm].

SVM optimized by cuML is beneficial for sports and soccer prediction because it can handle large amounts of data and identify complex patterns and relationships between variables. cuML is a suite of fast, GPU-accelerated machine learning algorithms designed for data science and analytical tasks that can complete 10-50x faster than their CPU equivalents (RAPIDS AI Team (2021). cuML - RAPIDS Machine Learning Library. GitHub repository).

In terms of success rate, studies have reported high prediction accuracy for SVM models in sports prediction. For example, a study by Ding (2015) established a prediction model using SVM algorithm for the evaluation of match winning was able to give about 70 % accuracy [Ding, S. (2015). Game Predication of FIFA Football World Cup Based on Support Vector Machine].

## 4) Final Evaluation

Out of the three models discussed (Random Forest, Support Vector Machine, and Linear Regression), the Linear regression model may be the most appropriate for predicting soccer match results.

Additionally, linear regression is relatively easy to implement, interpret, and explain. We use two machine learning models, Random Forest Classifier and Linear Regression, and compare their performance. We use cuML, a GPU-accelerated machine learning library, to speed up the training and prediction process.

### Setup:


To run this project, we request a GPU node with Colab and check the specs of the GPU node. The GPU node used in this project is Tesla T4 and the CUDA version is 12.0. We install RAPIDS AI, which provides us with the required libraries such as cuml, cudf, and cupy. We load the dataset into Colab, which contains information about home and away teams, such as matches played, wins, win rate, and various attributes related to their gameplay. The schema is following:

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 18584 entries, 0 to 19560
Data columns (total 28 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   home_matches_played                  18584 non-null  int64
1   home_wins                            18584 non-null  int64
2   home_win_rate                        18584 non-null  float64
3   home_buildUpPlaySpeed                18584 non-null  float64
4   home_buildUpPlayDribbling            18584 non-null  float64
5   home_buildUpPlayPassing              18584 non-null  float64
6   home_chanceCreationPassing           18584 non-null  float64
7   home_chanceCreationCrossing          18584 non-null  float64
8   home_chanceCreationShooting          18584 non-null  float64
9   home_defencePressure                 18584 non-null  float64
10  home_defenceAggression               18584 non-null  float64
11  home_defenceTeamWidth                18584 non-null  float64
12  away_matches_played                  18584 non-null  int64
13  away_wins                            18584 non-null  int64
14  away_win_rate                        18584 non-null  float64
15  away_buildUpPlaySpeed                18584 non-null  float64
16  away_buildUpPlayDribbling            18584 non-null  float64
17  away_buildUpPlayPassing              18584 non-null  float64
18  away_chanceCreationPassing           18584 non-null  float64
19  away_chanceCreationCrossing          18584 non-null  float64
20  away_chanceCreationShooting          18584 non-null  float64
21  away_defencePressure                 18584 non-null  float64
22  away_defenceAggression               18584 non-null  float64
23  away_defenceTeamWidth                18584 non-null  float64
24  home_win                             18584 non-null  int64
25  away_win                             18584 non-null  int64
26  draw                                 18584 non-null  int64
27  avg_betting_odds                     18584 non-null  float64
dtypes: float64(21), int64(7)
memory usage: 4.1 MB

```

## Models:

We use two models for this project, Random Forest Classifier and Linear Regression. In order to do classification, the Random Forest Classifier builds a large number of decision trees during training and then outputs the class that represents the mean of all the classes (classification) or mean prediction (regression) of all the individual trees. A scalar response (or dependent variable) and one or more explanatory variables (or independent variables) are modeled using the linear method of regression. 

### 3.1) Random Forest Classifier

We benchmark the time needed for using CPU and GPU for scikit-learn and cuml.

Criteria	cuML	scikit-learn
Fit Model	CPU times: user 3.64 s, sys: 166 ms, total: 3.81 s Wall time: 4.83 s	CPU times: user 42.7 s, sys: 44.7 ms, total: 42.8 s Wall time: 55.4 s

Predict	CPU times: user 10.8 s, sys: 100 ms, total: 10.9 s Wall time: 11 s	CPU times: user 1.08 s, sys: 668 ms, total: 1.75 s Wall time: 2.99 s
Evaluate (accuracy score)	0.751	0.692

The results show that cuML is much faster than scikit-learn, with a CPU time of 4.83 seconds for fitting the model and a CPU time of 11 seconds for prediction, compared to a CPU time of 55.4 seconds for fitting the model and a CPU time of 2.99 seconds for prediction using scikit-learn. Moreover, cuML achieves a higher accuracy of 0.751 compared to 0.692 for scikit-learn.

### 3.2) Linear Regression

We benchmark the time needed for using CPU and GPU for scikit-learn and cuML.

Criteria	cuML	scikit-learn
Fit Model	CPU times: user 218 ms, sys: 4.48 ms, total: 223 ms Wall time: 255 ms	CPU times: user 39.8 s, sys: 7.51 s, total: 47.3 s Wall time: 37.8 s
Predict	CPU times: user 97.1 ms, sys: 3.14 ms, total: 100 ms Wall time: 158 ms	CPU times: user 109 ms, sys: 928 μs, total: 109 ms Wall time: 85.1 ms
Evaluate	CPU times: user 1.78 s, sys: 16.7 ms, total: 1.8 s Wall time: 2.84 s	CPU times: user 9.75 ms, sys: 0 ns, total: 9.75 ms Wall time: 9.82 ms

The results show that cuML is much faster than scikit-learn, with a CPU time of 255 milliseconds for fitting the model and a CPU time of 158 milliseconds for prediction, compared to a CPU time of 37.8 seconds for fitting the model and a CPU time of 85.1 milliseconds for prediction using scikit-learn. Moreover, cuML achieves a higher accuracy of 0.9966 compared to 0.9966 for scikit-learn.

#### Pickle:

After training the models, we pickle them to save our models. We compare the accuracy of the models before and after pickling.

#### Pickle cuML random forest classification model

```
filename = 'cuml_random_forest_model.sav'
# save the trained cuml model into a file
pickle.dump(cuml_model, open(filename, 'wb'))
# delete the previous model to ensure that there is no leakage of pointers.
# this is not strictly necessary but just included here for demo purposes.
del cuml_model
# load the previously saved cuml model from a file
pickled_cuml_model = pickle.load(open(filename, 'rb'))
```

```
# predict using the pickled model
%%time
pred_after_pickling = pickled_cuml_model.predict(X_cudf_test)

fil_acc_after_pickling = accuracy_score(y_test.to_numpy(), pred_after_pickling)
```

CPU times: user 321 ms, sys: 52.5 ms, total: 374 ms  
Wall time: 729 ms

#### Compare Results

```
print("CUMML accuracy of the RF model before pickling: %s" % fil_acc_orig)
print("CUMML accuracy of the RF model after pickling: %s" % fil_acc_after_pickling)
```

CUMML accuracy of the RF model before pickling: 0.7512195110321045  
CUMML accuracy of the RF model after pickling: 0.7512195110321045

#### Pickle cuML random forest classification model

```
filename = 'cuml_random_forest_model.sav'
# save the trained cuml model into a file
pickle.dump(cuml_model, open(filename, 'wb'))
# delete the previous model to ensure that there is no leakage of pointers.
# this is not strictly necessary but just included here for demo purposes.
del cuml_model
# load the previously saved cuml model from a file
pickled_cuml_model = pickle.load(open(filename, 'rb'))
```

```
# predict using the pickled model
%%time
pred_after_pickling = pickled_cuml_model.predict(X_cudf_test)

fil_acc_after_pickling = accuracy_score(y_test.to_numpy(), pred_after_pickling)
```

CPU times: user 321 ms, sys: 52.5 ms, total: 374 ms  
Wall time: 729 ms

#### Compare Results

```
print("CUMML accuracy of the RF model before pickling: %s" % fil_acc_orig)
print("CUMML accuracy of the RF model after pickling: %s" % fil_acc_after_pickling)
```

CUMML accuracy of the RF model before pickling: 0.7512195110321045  
CUMML accuracy of the RF model after pickling: 0.7512195110321045

The results show that the accuracy of the models remains the same after pickling, with an accuracy of 0.751 for Random Forest Classifier and an accuracy of 0.9966 for Linear Regression.

### 3.3) Conclusion

In this project, we used machine learning models to predict the outcome of soccer matches. We used two models, Random Forest Classifier and Linear Regression, and compared their performance using cuML and scikit-learn. The results show that cuML is much faster than scikit-learn and achieves a higher accuracy for both models. We also pickled the models and compared their accuracy before and after pickling, and the results show that the accuracy remains the same. This project demonstrates the effectiveness of using GPU-accelerated machine learning libraries for faster and more accurate predictions.



## **Contributions:**

- Jerry Xiao: data cleaning and visualization
- Emil Ekambaram: Data preprocessing, data visualization, feature selection
- Lam Nguyen: HPC Integration, Setting up environment for parallel GPU running using Tesla T4 node on Google Colab
- Duc Huy Nguyen: Coding and fine-tuning machine learning models using cuML and scikit-learn
- Loc Nguyen: Evaluate models' performance using time and accuracy metrics & saving them using Pickle
- Hieu Hoang: Conduct an extensive inquiry into the intricacies of machine learning models ( Linear regression, Support Vector Machine, Random Forest Algorithms )
- Andrew Tran: Conduct an extensive inquiry into the intricacies of machine learning models ( Linear regression, Support Vector Machine, Random Forest Algorithms )
- Zach Youssef: Data sourcing and feature selection

## References

1. Cohea, C., & Payton, M. E. (2011). Relationships between player actions and game outcomes in American football. *Sportscience*, 15(1), 19-24.
2. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.  
<https://doi.org/10.1002/9781118548387>
3. Pelechrinis, K., & Papalexakis, E. (2016). The anatomy of American football: Evidence from 7 years of NFL game data. *PloS one*, 11(12), e0168716. <https://doi.org/10.1371/journal.pone.0168716>
4. Rotshtein, A., Posner, M., & Rakityanskaya, A. (2005). Football predictions based on a fuzzy model with genetic and neural tuning. *Cybernetics and Systems Analysis*, 41(4), 619-630
5. Shin, H., & Gasparyan, M. (2016). Logistic Regression Model for Predicting Match Results in Football: Case Study English Premier League Season 2015/2016. In *2016 International Conference on Advanced Informatics: Concepts Theory and Applications (ICAICTA)* (pp. 1-5). IEEE.  
<https://doi.org/10.1109/ICAICTA.2016.7803111>
6. Alpaydin, E. (2020). *Introduction to machine learning* (4th ed.). MIT press.
7. Bunker, R. P., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1), 27-33.  
<https://doi.org/10.1016/j.aci.2017.09.005>
8. Joseph, A., Fenton, N. E., & Neil, M. (2019). Sports prediction and betting models in the machine learning age: The case of Tennis. *Journal of Sports Analytics*, 5(2), 95-113.
9. Rodrigues, F., & Pinto, . (2022). Prediction of football match results with Machine Learning. *Procedia Computer Science*, 204, 463-470.  
<https://doi.org/10.1016/j.procs.2022.08.057>

10. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.
11. Joseph, A., Fenton, N. E., & Neil, M. (2019). Sports prediction and betting models in the machine learning age: The case of Tennis. *Journal of Sports Analytics*, 5(2), 95-113.
12. RAPIDS AI Team (2021). cuML - RAPIDS Machine Learning Library. GitHub repository.
13. Al-Hamadani, F., Al-Hamadani, A., & Al-Hamadani, A. (2020). Random forest algorithm to identify factors associated with sports-related dental injuries. *BMC sports science, medicine and rehabilitation*, 12(1), 1-7.
14. Lang, S., Wild, R., Isenko, A., & Link, D. (2022). Predicting the in-game status in soccer with machine learning using spatiotemporal player tracking data. *Scientific reports*, 12(1), 1-11.
15. Ding, S. (2015). Game Predication of FIFA Football World Cup Based on Support Vector Machine.
16. RAPIDS AI Team (2021). cuML - RAPIDS Machine Learning Library. GitHub repository.
17. Zhu, P., & Sun, F. (2019). Sports Athletes' Performance Prediction Model Based on Machine Learning Algorithm. In *International Conference on Applications and Techniques in Cyber Intelligence* (pp. 623-632). Springer.
18. Linehan, O. (n.d.). Predicting Premier League Results Using Linear Regression. RPubS. <https://rpubs.com/OliverLinehan/998846>
19. NHSJS (2020). Linear Regression to Analyze the Relationship between Points and Goal Difference in Premier League Standings. *National High School Journal of Science*.  
<https://nhsjs.com/2020/linear-regression-to-analyze-the-relationship-between-points-and-goal-difference-in-premier-league-standings/>
20. Press, F. (n.d.). Predicting the Outcome of English Premier League Football Matches [PDF]. Francis Press.  
<https://francis-press.com/uploads/papers/wCUqARj7oPkgICq5sO8o8wBydPnCaBI9ptgm0OwH.pdf>