Lien : <https://towardsdatascience.com/machine-learning-algorithms-for-football-prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8>

**Machine Learning Algorithms for Football Predictions**

**Using statistics from Brazilian championship data**

[Matheus Kempa](https://medium.com/@matheuskempa?source=post_page-----51b7d4ea0bc8--------------------------------)

[Matheus Kempa](https://medium.com/@matheuskempa?source=post_page-----51b7d4ea0bc8--------------------------------)

[Sep 21, 2020·11 min read](https://towardsdatascience.com/machine-learning-algorithms-for-football-prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8?source=post_page-----51b7d4ea0bc8--------------------------------)



Photo by [Willian Justen de Vasconcellos](https://unsplash.com/@willianjusten?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

**Abstract**

This article evaluated football/Soccer results (victory, draw, loss) prediction in Brazilian Football Championship using various machine learning models based on real-world data from the real matches. The models were tested recursively and average predictive results were compared. **The results showed that logistic regression and support vectors machine yielded the best results, exhibiting superior average accuracy performance in comparison to others classifiers (KNN and Random Forest), with 49.77% accuracy (logistic Regression), almost 17% better than a random decision (benchmark) which has 33% of success chance**. In addition, a ranking of the features’ relative importance was made to orient the use of Data.

**Project Overview**

Football/Soccer is a sport that is very present in people life’s, people use to watch, play, and also bet. Thinking about betting, we clearly can see that football is a very unpredictable sport, and it does not acquire serious research to prove that. In the premier league 2015/2016 season, we had a very unexpected champion, and their [probability for the title at the beginning of the season was one to five thousand](https://www.dn.pt/desporto/leicester-campeao-a-aposta-mais-improvavel-da-historia-britanica-5152206.html#:~:text=Segundo%20Joe%20Crilly%2C%20da%20William,paga%20mais%20alta%20na%20hist%C3%B3ria).

So, the prior objective of this project is to create a supervised machine learning algorithm that predicts the football matches results based on the statistics of the matches. Thus it will be possible to evaluate the difficulty level of prediction.

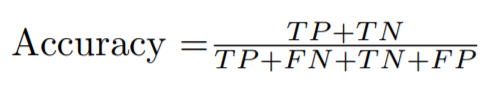
**Problem Statement**

This project aims to:

* Web scrapping robot to pick all the information of the matches
* Automatize the process of Web scrapping to all the season matches
* Create a supervised machine learning model to predict the outcome of the matches
* Evaluate the models

**Metrics**

In classification problems, is common to use accuracy, as an evaluation metric. As our outcome prediction is a multi-class problem, it’s not going to be necessary to use other metrics.



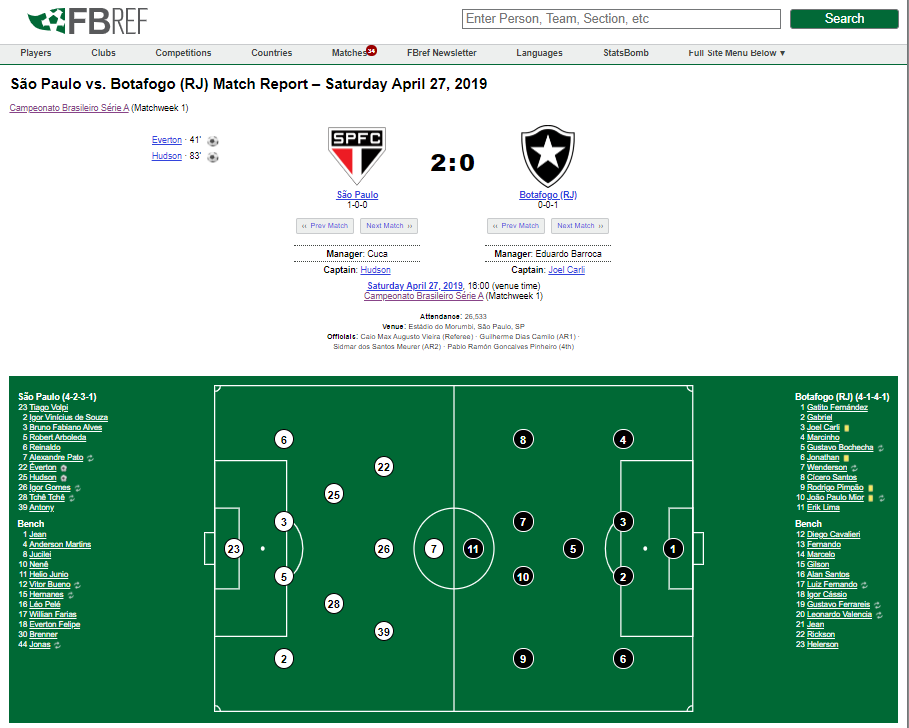
Accuracy formula. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

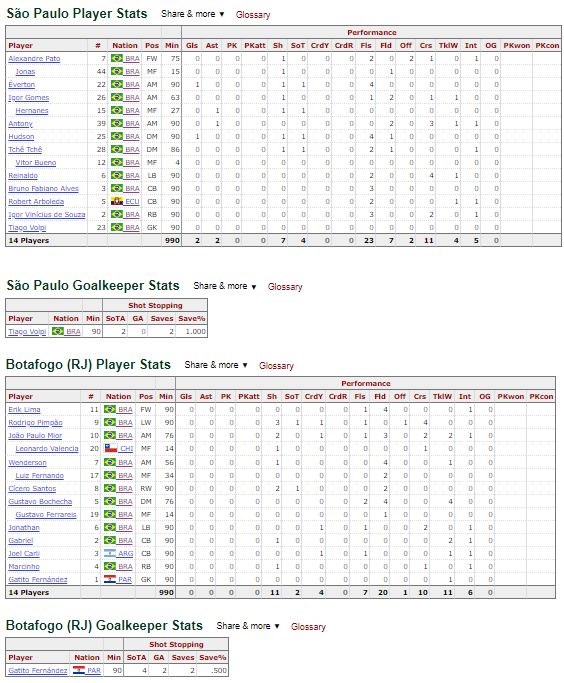
Where TP are the true positives, FP are the false positives, TN are the true negatives and FN are false negatives.

**ETL and Data Exploration**

**Web-Scrapping**

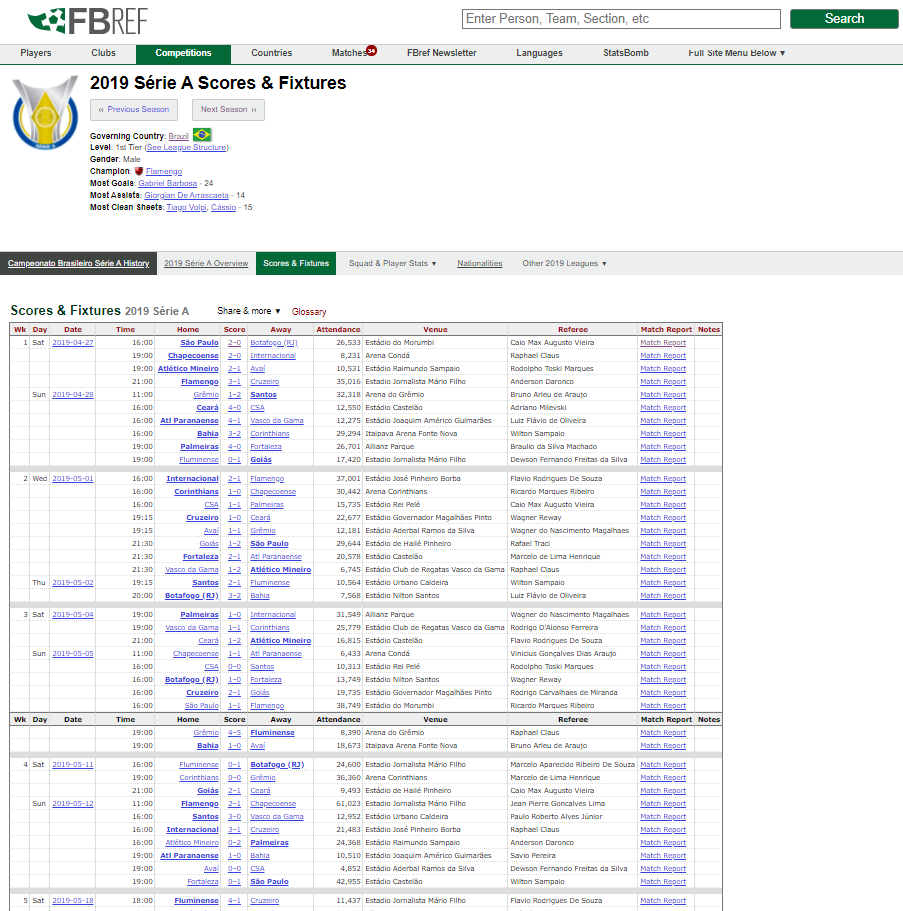
However, before exploring the collected data, it’s crucial to understand how this information was collected. So, this part will reach since the web-scrap robot developed for the final analysis of the whole database treated. The first and the second image below show how displays the page that the data was collected. So step number **one** is ***Web-scrap the main page, picking the football data, and creating a Data frame with all information combined.***





[Webscrapping Match Page](https://fbref.com/en/matches/e1867e7b/Sao-Paulo-Botafogo-RJ-April-27-2019-Serie-A). Source: [FBREF](https://www.sports-reference.com/sharing.html)

Now that we have the code ready to pick data from the matches it is necessary to create another code to collect all the season matches URLs so that it will be an automatized robot to do this task.



[Season Games](https://fbref.com/en/comps/24/3320/schedule/2019-Serie-A-Scores-and-Fixtures): Source: [FBREF](https://www.sports-reference.com/sharing.html)

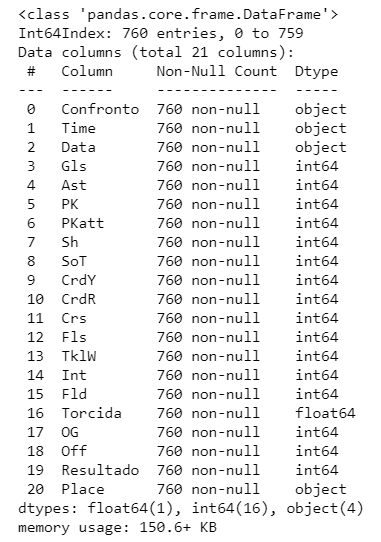
All the codes will be attached to the [Git repository](https://github.com/Matheuskempa/My_Udacity_Capstone).

Before the data were clean and ready to be used for analysis, the below steps were done:

* Select columns: Selected columns that hadn’t had a lot number of null values.
* Problems: The data collection came with some “problems” as the total of all player team’s statistics, so it was needed that these lines were expelled from the data.
* Group by match and team: Because the data collected were from the players of the match, it was necessary to group all the statistics of the player’s team to matches and teams.
* Append the Result: Because the data collected were from the player’s table, it did not bring the result of the match, so it was necessary to create a code that appends this result to the Data frame.
* Place: It’s was necessary to create a code that shows which team played at home and away.

**Data**

Now it’s time to evaluate the data collected so that it could be possible to create a prediction model to the results. These were the columns that our cleaned data had:



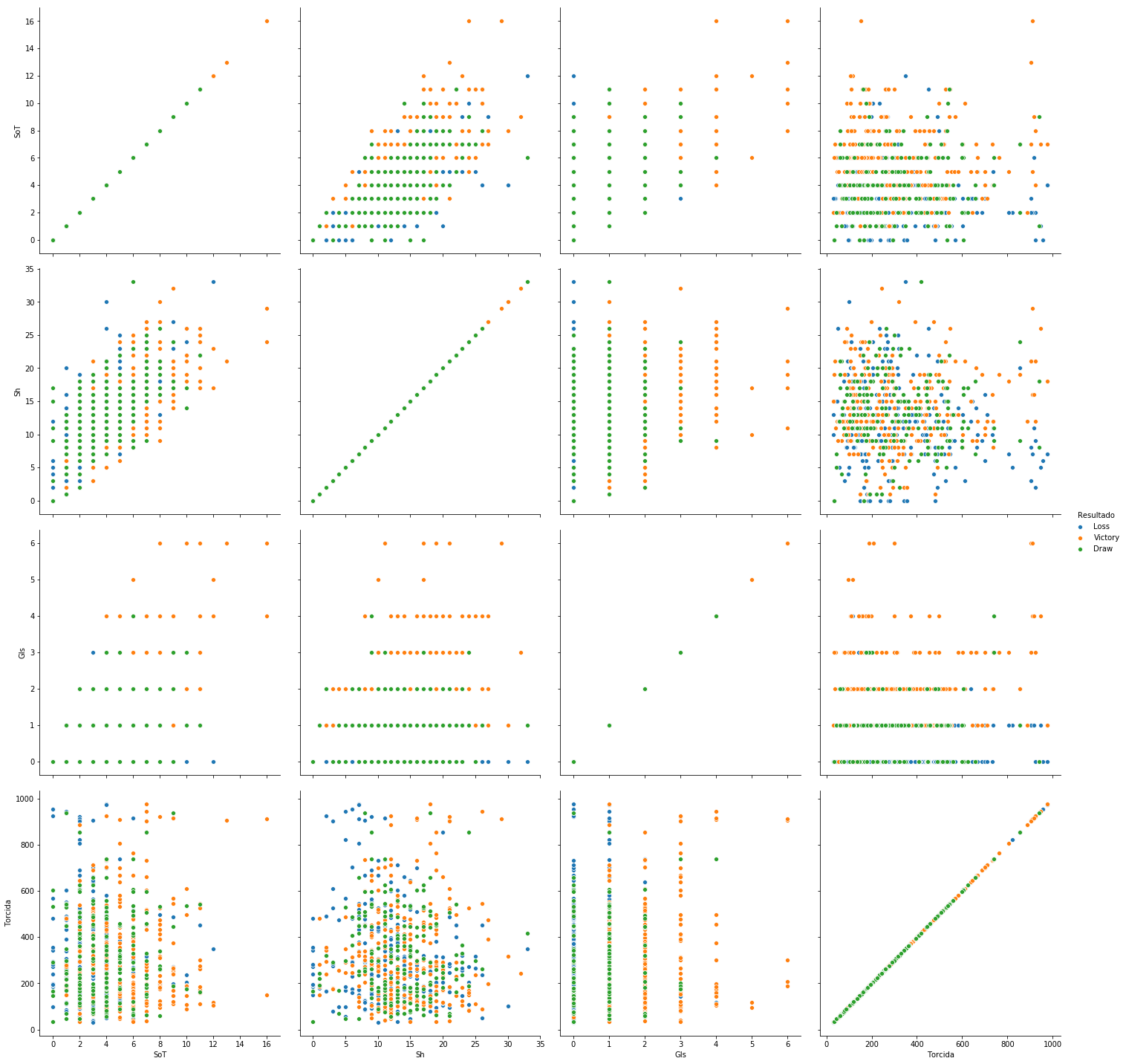
Columns info. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

The meaning of all the variables:

* Confronto: Match
* Time: Team
* Data: Date of the match
* Gls: number of Goals in the match
* Ast: Assists
* PK: Penalty Kicks Made
* PKatt: Penalty Kicks Attempted
* Sh: Shots Total
* SoT: Shots on target
* CrdY: Yellow Cards
* CrdR: Red Cards
* Crs: Crosses
* Fls: Fouls Committed
* TklW: Tackles Won
* Int: Interceptions
* Fld: Fouls Drawn
* OG: Own Goals
* Off: Offsides
* Resultado: Result of the match (Victory, Loss, Draw)
* Place: Home/ Away
* Torcida: Crowd

*Observation: Every match had to lines one for the home team, and another for the away team.*

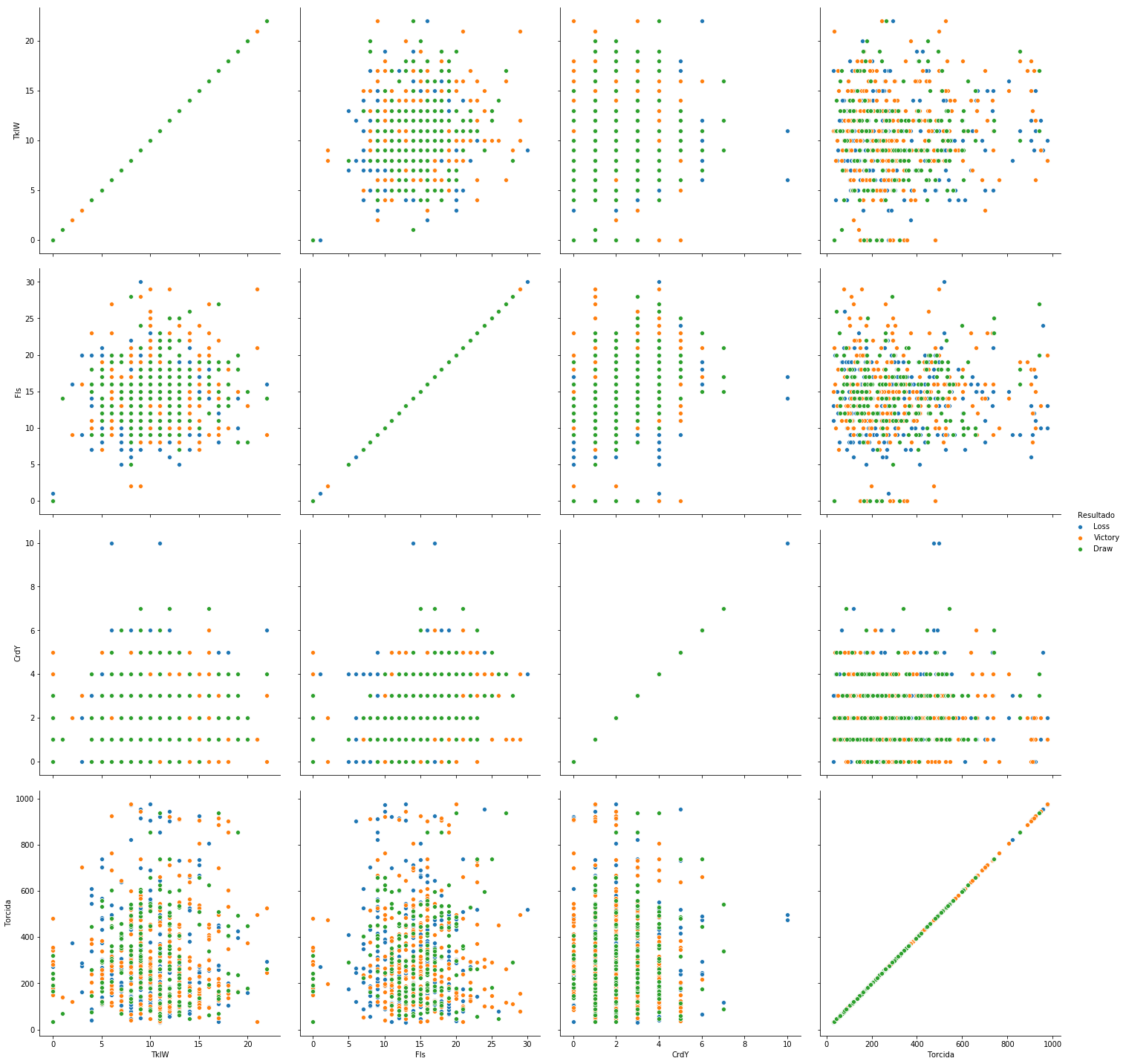
It’s good to check how these variables relate to each other so that it was created to approaches: Attack and Defense. In the code, it was also provided with a function that does this approach by the team and places played (Home/Away).



General Attacking Data. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

For the attacking approach, it was selected 4 variables: “SoT”, “Sh”, “Gls”, “Torcida”. Looking at the data it’s possible to see how difficult is to solve this problem because we can’t find patterns looking at it. But it might be possible to conjecture some hypotheses different from the usual as:

* A high Torcida(football crowd) might not be related to a more significant number of goals.
* A low Torcida(football crowd) might be related to a high number of SoT(Shot on target).



General Defense Data. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

With these graphs, it’s possible to see how our data is related, how the columns are related to each other, thus looking at 3 variables at the same time is possible to see the level of difficulty to separate this data. For the defense approach, it was selected 4 variables: “TklW”, “Fls”, “CrdY”, “Torcida”. And as in attack analysis, it was not possible to find any patterns on the Defense Data either.

However looking at Columns Info Figure, there are more variables available than those shown on graphs. And it’s clear that at least 17 collected variables, can influence the match outcome therefore also on the performance of the prediction. But because football is a lot more complex, more variables were needed to have a good predict a match outcome, so in the next part, it will generate some other variables.

**Methodology**

**Data Pre-Processing**

As it’s not possible to have the statistic of the games before the result of the match, it’s necessary to create some new variables that would be available before the games. So in order to solve this dilemma, it is necessary to generate a mean for all the variables, this means will contain all the games before the correspondent game, by this way when a team plays on September 18, the code will provide a mean of all the variables available to all the games before this exactly game.

**It’s going to be created some variables that try to show the computer the sequence of points for every team in the last 5 games, 3 games, and for the last game also**. For every victory, the team the code summed 3 points, for a draw summed 1 point and for a loss summed no point, 0 points. In this way, it would be possible to see if the team comes from victories, draws, and losses.

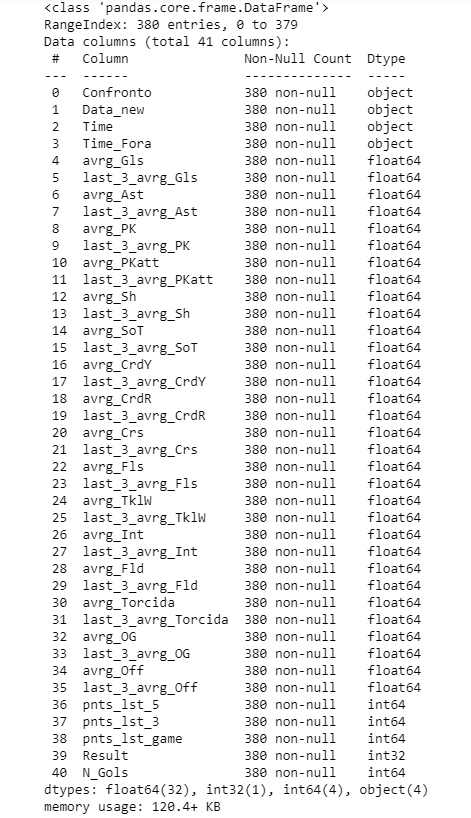
Since these treatments were done, it was found a way to inform the machine of the “place” of the match. Because of this variable (place) and as the championship has 2 turns, the first turn may be in the Home of a team, and if the first game of the first championship turn it was on their home, the second necessarily has to me Away, or in other words, not him their stadium. And in Football matches, this is a very important variable. So the way designed to consider this variable was: Taking all data with all the variables created for the home teams(Place = Home) for every match, and then, subtract by the same dataset(the same variables) but from visiting teams (Place = Away). By this way it could be possible to generate a Database that the data would basically say:

* **If Variables = Negative Values**: The visiting team has had better performances in this variable in the past games than the visitant/opponent. It’s possible to know that because, for example: picking the variable ”Average of Shots”, if the home team has a value P, then the visiting team has X, if P is higher than X the output value of the subtraction of both (HOME — AWAY) will be positive, otherwise if P is lower than X the value will be negative, showing that the team does not have better performances on that variable in the past games played.

Recapping:

1. **Problem:** The variables were only available after the match and for our model to perform it’s necessary to have all data features data before the correspondent match so that it was created a season average for every variable of the season and more, some moving averages for each variable of the Dataset too.
2. **Problem:** Insert the sequence variables: That problem was solved by summing all the points that the team had had in the past 3, 5 games, and also the last game. For every victory, the team summed 3 points, for a draw 1 point and for a loss 0 points.
3. **Problem:** Show to the machine who is the home and the away team. To solve this we subtracted the home team variables results by the visitant variables results for every match. Showing in this way if the team home is in any way superior or inferior to the visitant team.

The final Database had 41 columns and 380 lines, and looked like this:



Final Data. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

Coming at this point our database has finally been treated and by now it’s possible to run some models and check their performances

**Implementation**

To run we dropped the first four variables and “Gls” variable either also was used MinMaxScalar on all the features variables. As can be seen on [Sklearn](https://scikit-learn.org/), this estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one. Moreover were used all these 4 models:

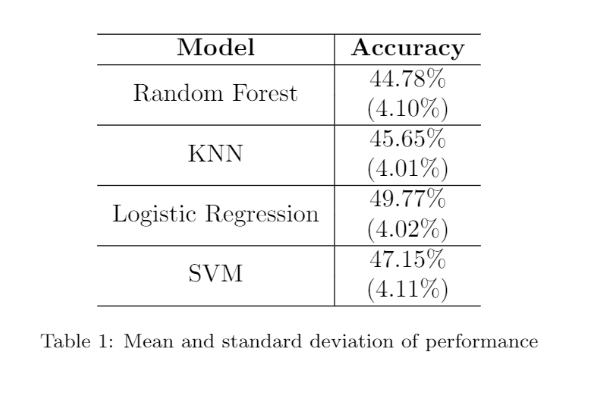
* **Support Vector Machine:** is a supervised learning model, that always aims to increase the distance between the points so that it is possible to classify the classes, so, SVM separates data maximizing the margin between the classes.
* **Random Forest:** As the name says, random forest creates several decision trees and groups them into one “Forest” of trees, taking a sample with size m bootstrap of the columns that represent the explanatory variables when partitioning the tree in each node of it. The final decision vote will be given by the majority vote for classification problems (and the average, in a regression problem).
* **KNN:** is a simple model that performs classification based on the class of its k nearest points based on a distance metric.
* **Logistic Regression:** Basically, logistic regression is a multiple linear regression whose result is “squeezed” in the interval [0, 1] using the sigmoid function.

In order to run all these models we split the Database randomly using the library ”train test split” from scikit-learn. For the **test, it was used 30% of the Data**. **Moreover, the algorithm was randomly played through 1000 times**, **for all four models**. In this way it was possible to check the standard deviation and the accuracy of all models. All models were used by the default values, not exploring the parameters of each specific model.

**Results**

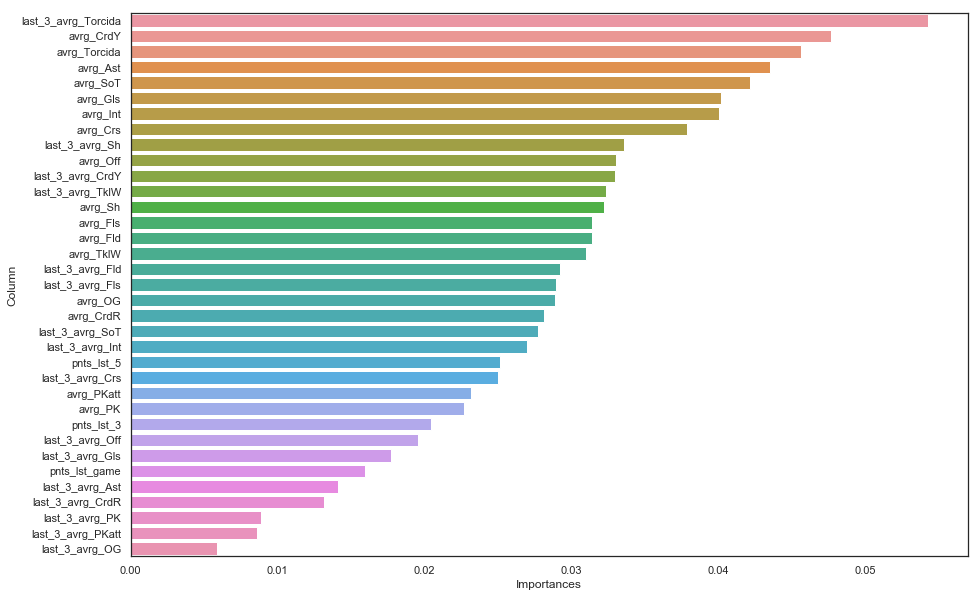
**Model Evaluation and Validation**

As can be seen in the table below, the results weren’t too expressive. The algorithm achieved almost 50% of accuracy, but, considering that the random prediction probability of success is 33% (Victory/Draw/Loss), it is a good signal, showing that the algorithm was able to identify some patterns after all.



Mean and Standard Deviation. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

After all 1000 times, the ***Logistic Regression had the best results,*** with one of the lowest standard deviation rates and the highest accuracy between all four models. SVM performed better than the others, but had the highest standard deviation, maybe showing that the model tends to vary more than other models. Also were created a feature ranking using Random Forest features importance. As can be seen in the figure bellow, the football game crowd from the last 3 matches, the yellow card numbers, and also football game crowd from all the season were influencing more on the prediction outcome.



Feature Ranking. Source: My Code on [github](https://github.com/Matheuskempa/My_Udacity_Capstone)

**Justification**

It’s weird to see logistic regression performed much better than the other models, mainly because as it was seen, data clearly it did not have patterns and it was a complex problem and surprisingly a very complex problem was solved, by a ”linear” approach, maybe the simplest here. One of the reasons for this may be because of the parameters of each model, that were not explored in this approach. Because SVM has a lot of kernels, trying different kernels might be a solution to reach a better performance for this model. Another strange point is that the ”points sequence” was one of the lasts variables to interfere with the prediction. As far as I am concerned this variable should not be ignored and maybe there are more ways to show these patterns to the machine.

**Conclusion**

**Reflection**

This article proves that football prediction is still a very hard task, it still needs more variables to help with the prediction of the results. However, we can see by this article that a machine learning algorithms can already ”think” on which team bet and can still be more accurate than people that do not know about the games having almost 17% of advantage in the prediction when comparing to the probability of a random prediction.

**Improvement**

For the future, I suggest to investigate and find more variables that could be useful, as injuries for example, or more details of the players of each team, maybe FIFA or Pro Evolution game data could help to bring more information inside of the Base. Another thing that could be done for the future is on predicting the number of goals for of each team, this is more complex because it depends on the results predicted and they must conciliate with it, for example, it couldn’t be two goals for the home team and two goals for the way team if the result was predicted as a victory of the home team. So, maybe, this article can be a source of inspiration for the creation of better and complex models in the future.

**[Matheus Kempa](https://medium.com/@matheuskempa?source=post_sidebar--------------------------post_sidebar-----------)**

Always learning

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F7fa5bc03b175%2F51b7d4ea0bc8&operation=register&redirect=https%3A%2F%2Ftowardsdatascience.com%2Fmachine-learning-algorithms-for-football-prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8&user=Matheus%20Kempa&userId=7fa5bc03b175&source=post_sidebar-7fa5bc03b175-------------------------follow_card-----------)