Revista Brasileira de Computação Aplicada, November, 2020

DOI: 10.5335/rbca.v12i3.10234
Vol. 12, Nº 3, pp. 16–32
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ORIGINAL PAPER

Positional analysis of Brazilian soccer players using GPS data

Randal Gasparini [1] and Alexandre Álvaro [2]

[1]Universidade Federal de São Carlos, [2]Universidade Federal de São Carlos

*gasparini.randal@gmail.com; alvaro@ufscar.br;

Received: 2019-11-11. **Revised:** 2020-06-10. **Accepted:** 2020-07-25.

Abstract

The professional soccer is always changing and is constantly searching tools and data to help the decision-making, providing tactics and techniques to the team. In Brazil, this sport goes in the same way and the investments are considerable. The One Sports is a company that captures GPS data from professional soccer players of some Brazilian teams. This set of data has a lot of features and the One Sports asked if it was possible to predict the ideal position of a player. Then, was firmed cooperation between an academic study and a commercial company. This work finds to understand the proposed methods and techniques to predict the ideal position of the soccer player, using machine learning algorithms. The database has more than one million tuples. It was submitted to the preprocessing step, which is fundamental, because it generated new features, removed incomplete and noisy data, generated a new balanced dataset and delete outliers, preparing the data to execution of the algorithms k-NN, decision trees, logistic regression, SVM and neural networks. With the purpose to understand the performance and accuracy, some scenarios were tested. There were poor results when executed multiclass problems. The best results come from binary problems. The models k-NN and SVM, specifically to this study, had the best accuracy. It is important to note that SVM spent more than six hours to finish your execution, and k-NN used less than one and a half minute to end.

Keywords: Classification; GPS; Machine Learning; Soccer

Resumo

O futebol profissional está sempre mudando e está constantemente buscando ferramentas e dados para ajudar na tomada de decisões, fornecendo táticas e técnicas para a equipe. No Brasil, esse esporte segue o mesmo caminho e os investimentos são consideráveis. A One Sports é uma empresa que captura dados de GPS de jogadores profissionais de futebol de algumas equipes brasileiras. Esse conjunto de dados tem muitos recursos e o One Sports questionou se era possível prever a posição ideal de um jogador. Em seguida, foi firmada uma cooperação entre Universidade e a empresa. Com isso, este trabalho procura entender os métodos e técnicas propostos para prever a posição ideal do jogador de futebol, usando algoritmos de aprendizado de máquina. O banco de dados possui mais de um milhão de tuplas. Foi submetido à etapa de pré-processamento, que é fundamental, pois gerou novos recursos, removeu dados incompletos e ruidosos, gerou um novo conjunto de dados balanceado e excluiu outliers, preparando os dados para execução dos algoritmos k -NN, decisão árvores, regressão logística, SVM e redes neurais. Com o objetivo de entender o desempenho e a precisão, alguns cenários foram testados. Os resultados foram ruins quando executados problemas de várias classes. Os melhores resultados vêm de problemas binários. Os modelos k -NN e SVM, especificamente para este estudo, tiveram a melhor precisão. É importante observar que o SVM passou mais de seis horas para concluir sua execução e k -NN usou menos de um minuto e meio para finalizar.

Palavras-Chave: Classificação; GPS; Aprendizado de Máquina; Futebol

1 Introduction

The professional soccer level and all involved parties like the coach, the medical department, and the players themselves, is in constant evolution. The continuous search for better physical conditions of the athletes and a better tactical positioning are justifications for the constant investment in the data analyzes and the search of standards for the soccer players (Di Salvo et al., 2007, Bourke, 2003).

The evolution of the technology permits to evaluate the performance of the professional soccer player and also to monitor them during games. Some equipment like GPS (*Global Positioning System*), muscular monitor, and cardiac frequency meter are some devices for monitoring the athletes (Okazaki et al., 2012).

Specialized equipment collects individual data for after processing, to acquire relevant and strategic information for the technical team. Artificial intelligence can help by providing the support of specific algorithms that allow the creation of systems to decision support. The studies in machine learning allow certain computational techniques to be trained from selected inputs, making it possible to predict the most suitable output for a given set of features. It is also possible to perform analyzes using non-linear methods. The application of the reducing the dimensionality of non-linear data allows the identification of patterns. With this data it is possible to analysis through of the axes and distances of the points, getting predictions for coordinates not yet analyzed, similar to machine learning (Roweis and Saul, 2000).

The goal of this study is to answer the question:

What is the machine learning algorithm has bigger reliability to predict the tactical position of a Brazilian soccer player?

There was elected the machine learning algorithms: K-nearest neighbors algorithm (K-nn), decision trees, logistic regression, support vector machines (SVM) and neural networks. Before the execution of them, all data was preprocessed, eliminating redundant, incomplete and noisy data, deleting outliers and generating new features.

With the purpose to understand the performance and accuracy, some scenarios were tested. There was poor results when executed multiclass problems. The best results come from binary problems. The models k-NN and SVM, specifically to this study, had the best accuracy. It is important to note that SVM spent more than six hours to finish your execution, and k-NN used less than one and half minute to end. These expend time results consider the technique of cross validation.

2 Related Work

The video analysis is very applied, which obtains the movement and distribution of the players. It is a method usable for the data collection. In the nineties, a more rudimentary process was already based on this same technique. The data was collected through the repetition of games recorded on video tapes. Observers noted all possible information, such as kicks, fouls, long passes, among others. In parallel, physical analyzes were performed, such as race tests, speed and anaerobic of athletes[1]. All these data were confronted, obtaining valuable information about the team and its players (Meyer et al., 2000, Pappalardo et al., 2020).

More recently, the same technique is used, but the process is more automatic and autonomous. The capture takes place through modern cameras installed in strategic points and of good visual amplitude in the stadium. This method is called the Automatic Tracking System (ATS). After the capture, a software analysis all the collected material and calculations the variance, based on the positions obtained from the images (Barros et al., 2007, Di Salvo et al., 2007).

The decision to use the ATS needs to consider the quality of the equipment and the strategic choice to install them, or else, the accuracy may be unreliable Barros et al. (2007).

The capture of player data can also use Global Positioning System (GPS) equipment. This technology is based on the determination of points obtained through satellites. A transmitter, located on Earth, sends the coordinates and gets the positioning. This process is sequenced, capturing information about the players movement. The equipment is carried in their clothing during physical activities and in official games. The collected information is processed in a specific software, generating quantitative data about the athletes (Barbero-Álvarez et al., 2010, Hennig and Briehle, 2000, Hennessy and Jeffreys, 2018).

The models are different, but both can determine the athlete data. Information like distance, positional team organization, individual characteristics, and others can be collected. Both capture methods has data accuracy reliable, if the provided technical conditions required for each technology are respected. In this way it is safe to choose any of the methods (Edgecomb and Norton, 2006).

Considering the installation process of the equipment to use the technologies, it is necessary to consider that the ATS requires investment and preparation, since a specific set of cameras is necessary. The installation can be complex, since the fixing points are strategic and need to have great visual amplitude (prozone, 2016, Barros et al., 2007). When using GPS sensor, it demands that the equipment be coupled to each athlete's clothing, and needs of compatible technology and software for data interpretation (Edgecomb and Norton, 2006).

Considering both technologies, it is important to understand the application scenario to decide what equipment to choice. For example, the training is usually performed in reduced fields [2], ensuring a better use of the anaerobic conditions of the athletes. Games are always performed using official measures, then the

[1] ability to repeat the race in the maximum acceleration range several times without considerable loss of speed (Sienkiewicz-Dianzenza et al., 2009)

[2] usually in the training centers

cameras are only adopted in these stadiums, where there is the contracted technology for image capture (Dallaway, 2014).

Analyzing the volume data, it is possible the capture in a frequency from 16 to 25 Hz. The volume of data collected at this frequency is sufficient to generate significant data (Mitchell et al., 2013).

The collected data need to be very well characterized, allowing to identify the correlations. The identification of the attributes is fundamental to the success of the application. The outputs are always conditioned to the quality of the input data. For GPS applications in soccer, relevant data is associated with speed, acceleration and intensity (Dallaway, 2014).

Edgecomb and Norton (2006) studied the monitoring of Australian soccer athletes [3] using GPS and ATS. In addition to understanding the general behavior of athletes during a match, the objective was to obtain data on collective and individual positioning and displacement, then to compare the monitoring technologies. The study demonstrated that it was possible to track all the actions of the players in the field, allowing the collection of data in a considerable volume. It is a sporting modality different from the one proposed in this study, but it demonstrates the versatility of the technologies.

It is also necessary to consider how data are applied and analyzed. The data analysis can be obtained and processed the information in real time (during the game). Another possibility is the post-game data analysis. Aughey and Falloon (2010) observed that for Australian football, the margin of error is bigger when the application occurs during the game.

When analyzing field football the processing og the captured data needs to identify patterns, looking for similar characteristics. It is possible to define, for example, that players playing in the midfield have the characteristic of covering a larger area of the field in distance traveled. It is still possible to observe, within this same group, that the "lateral socks" act with a greater intensity, when compared with the "central socks". These characteristics can be observed in the distance and acceleration attributes, respectively (Dallaway, 2014).

Hennessy and Jeffreys (2018) brings an account of the GPS technology used for monitoring athletes from the beginning to the advances nowadays. It presents a series of metrics that can be extracted from the data collected by the GPS, such as total distance, speed zones, impact metrics, among others. Finally, it presents the limitations of technology for this application and envisions advances for the future.

Strauss A (2019) using data from GPS to quantify the internal and external match demands of semi-elite level female soccer players. The paper aims to describe the magnitude of change of these variables within and between matches over the course of a tournament to determine the effect of player fatigue mainly related to variables around the running intensities.

[3] sports similar to rugby but with specific adaptations and rules, which differentiates it from other sports

Pappalardo et al. (2020) presented a PlayeRank, a data-driven framework that offers a principled multi-dimensional and role-aware evaluation of the performance of soccer players. The used four seasons dataset and discover interesting patterns about the nature of excellent performances and what distinguishes the top players from the others.

Currently, football clubs in Brazil have been using more and more technologies to monitor their athletes, both in training and in games. Clubs with greater purchasing power use several technologies and techniques to monitor players, namely: GPS monitor and accelerometer, sleep monitoring, jumping platform, photo-cell platform, dehydration monitoring, CK monitoring (Creatine Kinaze), among others. However, most clubs have serious financial deficiencies and this impacts the wide adoption of technologies available on the market.

3 Application

The GPS data used in this study are real and were made available by OneSports. The data is from the first division of Brazilian professional football from the whole 2018 year.

The data provided was previously labeled with certain football positions. These are detailed in the Table 1.

Table 1: Positioning of the soccer players that will be used in this study (Scaglia et al., 1996)

Position	Description
Striker	Player who receives the ball in the attacking field of his team and has the objective to score the goal
Right Back	Player on the right side of his team. Its principal function is to connect the ball to leave the defensive sector and reach the midfield players. Has characteristics of sprinters, so can do the connection with strikers
Left Back	Like as the right side player, however, it acts on the left side
Attacking Midfielder	Midfield players with objective to support the strikers, in addition to pressure on the opponent's ball
Midfielder	Acts in the central of the field, providing support to the attack and defense
Defensing Midfielder	Like the attacking midfielder, but provides direct support to the defense of its team
Defender	Players specifically designated to defend their team and eliminate the risk of an opponent's goal

The database has a total of 3.281.948 tuples. It was observed inconsistent sets values. It was necessary to remove this data because they may confuse the models. The clear is detailed in the Table 2 (Faceli et al., 2011).

Table 2: Original database cleaning

Cause	Data	Removed Tuples
Noisy data	Distance player is zero	978.197
Inconsistent data	Label is null	6.789

The reason for the capture of inconsistent data can not be defined in this study, because there are no deterministic elements that demonstrate the cause.

Other problems were found in the dataset: calibration data[4] of the GPS equipment, test data, training data, and not relational data to foreign keys. The selection of these data was chosen because they are capable of inducing errors in machine learning models (Faceli et al., 2011).

The elimination affected 1.190.782 tuples. It is a big portion but it is an important step and it allows safer results.

Any tuple of the database contains the instant detail of the player, generating a sequence of information with low variations of data for each attribute. These data allow to understand the behavior of the player on the field. To get collective information, the data is grouped, generating new data. Table 3 shows this rearrange.

Table 3: Equal data tuples grouping

Information	Total
Number of games	30
Number of teams	22
Number of players	73
Number of fields	16

Each player holds a defined position in your team and with this information, the database is labeled. The quantification of the tuples per position is available in the Table 4.

It is possible to observe in the Table 4 the existence of unbalanced classes. Usually this feature has a negative impact on the predictive models, since the classifier can become biased. However, it is important to consider that real problems it is common to find not balanced situations, because the data has a natural oscillation. A classic example is the algorithm that predict a particular disease in a specific group of patients. Commonly a class will be predominant, of the sicks or the healthy, varying with the analyzed group. This example clearly demonstrates the need to train the algorithm with unbalanced bases, where classes are naturally not uniform (Batista et al., 2004).

It will be proposed in this study a reduction of the numerical disparity found between the classes. Will be applied to the balancing on the data. It will be used the replication technique of the minority classes, increasing the quantity of them, except for the class "midfielder", which is so inferior. It will be totally removed, since there is no numerical significance of this class in the universe of the problem, and may cause inductions in the algorithm (Batista et al., 2004).

Table 4: Number of tuples of each classes in the database

Label Position	Total of Tuples
Striker	200.438
Right Back	121.340
Left Back	35.240
Attacking Midfielder	407
Midfielder	337.945
Defensing Midfielder	273.398
Defender	222.014
Total of Tuples	1.190.782

The choice of attributes relevant to this study is directly associated with the principles of speed, distance, and positioning of the players in the field. All the attributes are correlated with the data which these characteristics. The selected attributes are shown in Table 5. All columns and data analyzed are in the database named *GPS*.

Table 5: Selected attributes based on velocity, distance and positioning in a time period

Feature	Data Type	Correlation
Position	Text	Has the the player's position
Distance	Decimal	Has the the player's distance
Speed	Decimal	Has the player's speed
Acceleration	Decimal	Has the the player's acceleration
Latitude	Decimal	Store the relative data player's latitudinal position in the field
Longitude	Decimal	Store the relative data player's longitudinal position in the field
Instant_time	Time	Has the capture time of the tuple. Its variations is in seconds

The preprocessing is a fundamental step because it adjusts the dataset to obtain better use of the machine learning algorithm. The preprocessing is focused on steps as the exclusion of attributes not relevant to the problem, elimination of noise, conversions, and features transformations (Faceli et al., 2011).

The standardization, transformation, and validity of attributes are keeping, adopting the same symbology and equivalent information for the problem.

All the selected attributes have the same importance

[4] Process to sync the equipment with satellites and adjusts the longitudinal and horizontal positioning in relation to the soccer field and its four corners. It is important to get the best accuracy (Yeh et al., 2006)

for this study, however latitude and longitude there are fundamental characteristics for the analysis proposed. Its importance is essential to understand the positional behavior of the athlete in the field, so these coordinates allow to know each movement performed by the athlete during the professional game. Their importance and apparent impossibility to discard them, it was observed in the data stored in the database. The same athlete has a divergence of its pattern of the data between a game and another. This situation is due to the fact of the games in sixteen different fields since the teams alternate the place of their games.

All field registered in the system, the latitudes and longitudes of its four corners are stored. With this information it is possible to follow the positioning of the athlete in the field, once a rectangle with their references and limits is formed. GPS information does not repeat itself to any distinct point on the planet, then the coordinates of each field will be unique. The first transformation was to reposition all players within the same field, guaranteed so that there is no influence of the latitude and longitude variation caused by different fields.

The Maracanã field, located in the city of Rio de Janeiro, was chosen for the athlete's replacement since it has official measures and can be easily mapped by the tool *Maps Google*. The mapping can be visualized in Fig. 1.

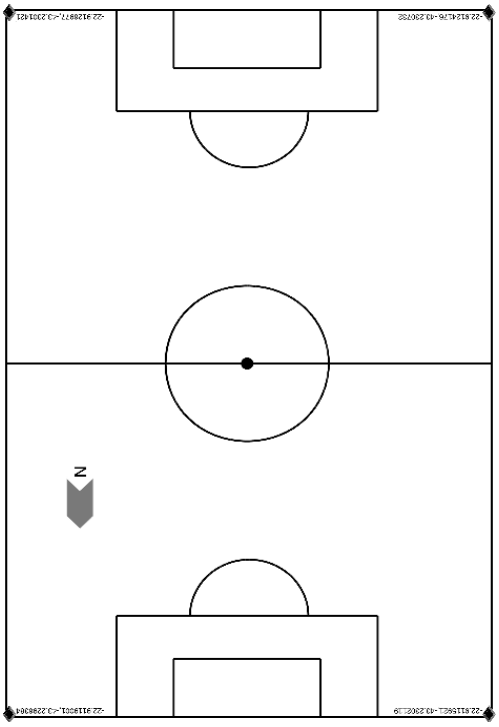

Figure 1: Latitudinal and longitudinal mapping of the Maracanã field

The GPS coordinates of the stadiums that received games were obtained again (using Maps Google). This process is essential for repositioning players in a single field (Maracanã).

After all the fields mapped, the upper corner was chosen to the north of each stadium as a reference for the calculations. The objective is to obtain two new data for each tuple: distance and angle of the athlete; both related to the chosen corner of the field.

To define the distance of the player from the base point of the field, the following Python function was implemented:[5]:

```
def distance(self,latit1,longit1,latit2,longit2):
import numpy as np
R = 6373.0 # approximate radius of the Earth in KM
lat1 = np.deg2rad(latit1)
lon1 = np.deg2rad(longit1)
lat2 = np.deg2rad(latit2)
lon2 = np.deg2rad(longit2)
dlon = lon2 - lon1
dlat = lat2 - lat1
a = np.sin(dlat / 2)**2+np.cos(lat1)*np.cos(lat2)*
np.sin(dlon / 2)**2
c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a))
distance = R * c
return round(distance*1000,1)
```

To know the position of the athlete, the distance is not sufficient. It is necessary to obtain also its angle relative to the reference point that was determined. For each tuple of the database a new attribute was calculated using the Python function:

```
def calc_angle(self,lat1,lon1,lat2,lon2):
import numpy as np
bearing = np.arctan2(np.sin(lon2-lon1)
*np.cos(lat2),
np.cos(lat1)*np.sin(lat2)-np.sin(lat1)
*np.cos(lat2)*
np.cos(lon2-lon1))
bearing = np.degrees(bearing)
bearing = (bearing + 360) % 360
return bearing
```

Considering the number of tuples and knowing that it was necessary to calculate new attributes, the pre-processing step consumed a total of 34 hours to execute. In the end, the angle and distance data were obtained for each tuple of the database, generating two new attributes.

Using the position of the player in the field of the origin (game) and knowing the coordinates of the four corners of the destination field (Maracanã), it was necessary to calculate the new positions of the players in the destination field (latitude and longitude). To calculate these data, it was used the library *geopy* of Python. It supports the algorithms based on geocodes.

The upper corner was chosen to the north direction of the Maracanã as a reference for the new positions, following the same pattern adopted. All tuples in the database were processed again, calculating the new coordinates of the player concerning the Maracanã stadium. At the end of this step, two new features were

[5] algorithms adapted from public functions and Python documentation

generated. Its names in the database are *new_latitude* and *new_longitude*. The Python function used for repositioning is demonstrated below:

```
def reposition(self,lat1,lon1,distance,bearing):
from geopy
import Point from geopy.distance
import distance, VincentyDistance
#convert the distance to meters
distance = distance / 1000;
location =
(VincentyDistance(kilometers=distance).destination
(Point(lat1, lon1), bearing))
lat2 = location.latitude
lon2 = location.longitude
return lat2, lon2
```

In the above code is used the VincentyDistance function. Created by Thaddeus Vincenty, it calculates the distance between two distinct points, based on the terrestrial globe. Its margin of error is a maximum of 0.5 mm. The geopy library (Python) supports this function (Vincenty, 1975).

The GPS coordinates are important for the analysis proposed, and all the athletes were repositioned in the same field, this feature is equivalent, making it more reliable. However, the preparation of the dataset and the features, an inconsistency was observed, can induce the algorithm to predict error. All games recorded in the database are composed of two forty-five minute periods. Each team occupies one side of the field each time. At the end of this, both sides invert. Analyzing from the perspective of only one team, when the inversion occurs the areas of the field have different positions. The defender starts to act in the position of the attacker. The inverse is true too. This applies to all positions, except the midfield players, who have less impact on this change, considering the occupied area. There is no label of the first and second time in the database, and there are no records of which side of the field the team played. It is necessary to consider that the new coordinates may generate some confusion for the machine learning algorithm.

Trying to find general information about the positioning of the players in the field, it was created a new feature. Considering that all were repositioned in the same stadium, the angle and distance of all tuples were calculated concerning the field medium. These new features have less impact on the data because it is possible to define the distance of the player in the field. The Fig. 2 exemplifies the situation.

It is important to note that changing the field side does not impact the distance calculated from the center, however, the angle is still affected when teams change goal side.

After the pre-processing steps, new features were obtained. It is important to understand the relevance and find the correlation between them. To find the best set of features, it was adopted the technique *wrapper*. Using the libraries in *Python*, the data were submitted to the algorithm to find the best set of attributes for the problem in question. Fig. 3 shows the accuracy obtained for each set of attributes.

Fig. 3 shows the best accuracy as six features. The set is: *distance*, *speed*, *acceleration*, *angle_corner*,

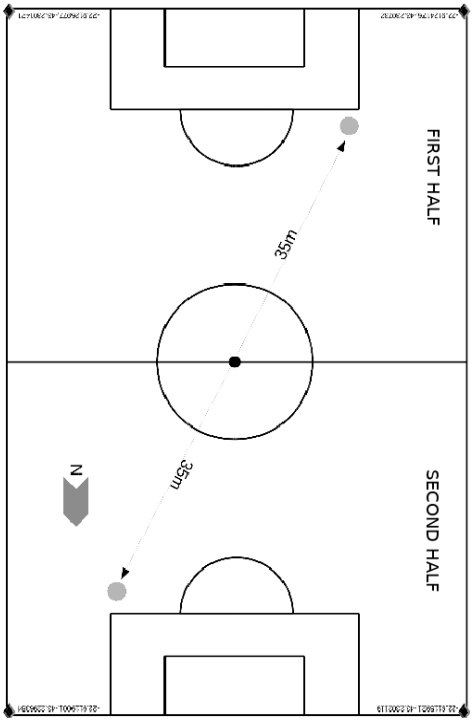

Figure 2: Distance from center of the field to midfield during first and second half

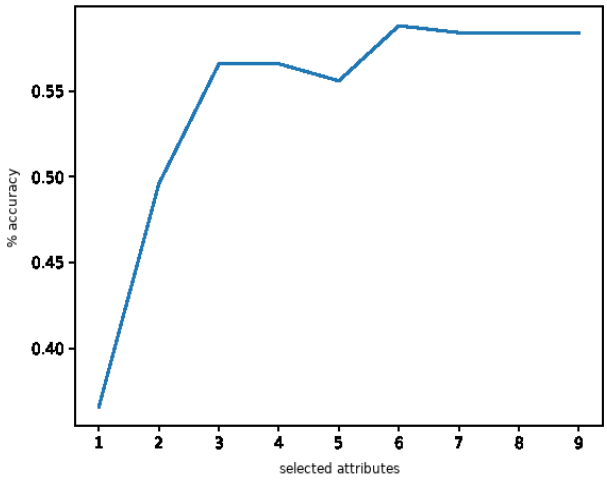

Figure 3: *Wrapper* in *Python*

distance_corner e *distance_middle*.

The choice of the *wrapper* is justified by the number of features and possibles combinations. It is a method computationally more costly, but it was faster and more reliable alternative if compared with other methods.

When different player positions are compared, the data variance is visible. If the feature scaling is adopted, this characteristic can be lost, since the data will have a reduced scalar interval. For the preservation of these characteristics, standardization was adopted.

An important step is a graphical visualization because it helps to understand the data and its correlations. It is also possible to understand its spread, identifying *outliers*.

The Principal Component Analysis (PCA) allows the reduction of the dimensionality of a dataset. The matrices T e P' is generated, from X. The two new matrices created are smaller, considering the original (X). The PCA application allows the rotation of axes, changing the graphical perspective on the data (Hair and Anderson, 2005, Wold et al., 1987).

Fig. 4 shows all data based on two-dimensional visualization after the application of the PCA.

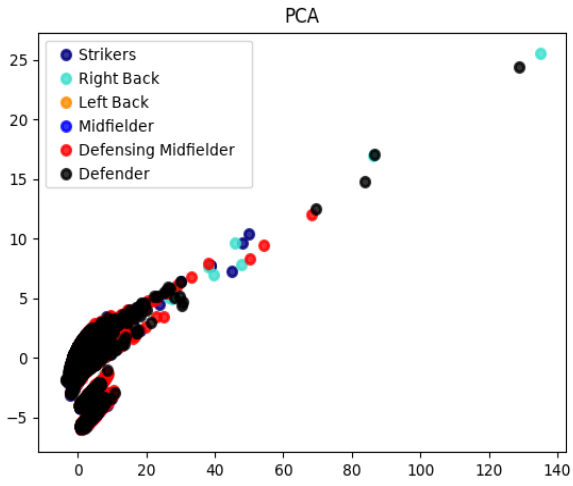

Figure 4: Data reduced to 2 applying using PCA

Fig. 5 shows all classes based on three-dimensional visualization after PCA application.

To identify the existence of possible outliers, the Figs. 6 and 7 shows the spreading rate of the data for the "midfield" and "striker" classes, respectively.

Based on the Figs. 5 to 7 it is possible identify *outliers*. It there are *outliers* it is necessary to eliminate them.

Adopting the six features selected, tests were applied to identify what is the data with a higher impact on outliers. After several runs, it was observed that the "distance" is the attribute with influence more significantly on the non-standard points. The strategy of eliminating tuples for data identified as *outliers* in the

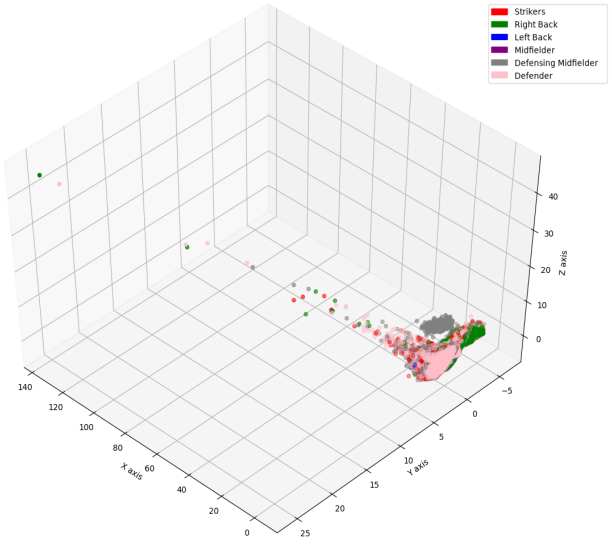

Figure 5: All classes reduced to 3 dimensions applying PCA

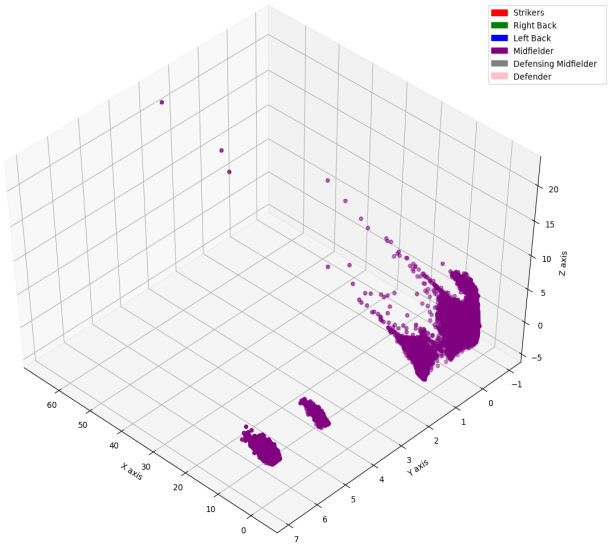

Figure 6: Midfielder class reduced to 3 dimensions using PCA

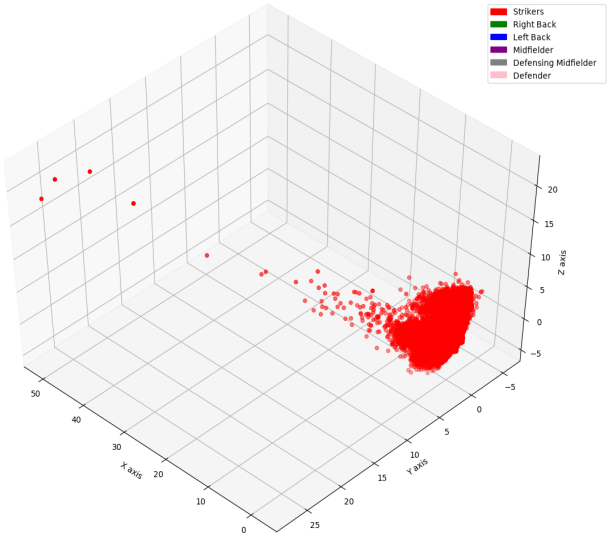

Figure 7: Striker class reduced to 3 dimensions using PCA

"distance" was adopted. The method used was based on the interquartile ranges, where the data must have a value between 75% and 25% percentiles, adopting the equation shown in Eq. (1).

$$IQR = Q_3 - Q_1, \tag{1}$$

where: Q_1 is the lower quartile and Q_3 is the upper quartile.

Finally finding outliers:

$$[Q_1 - 1.5.IQR, Q_3 + 1.5.IQR], \tag{2}$$

The target value needs to into the first and second limits obtained in Eq. (2)[6].

The Fig. 8 shows all classes, the Fig. 9 shows only midfielder and the Fig. 10 shows only strikers.

After elimination of the outliers, it is possible to view the significant data decrease. It is possible too view axis rotation, promoted by PCA and the classes overlap, specifically in Fig. 8.

The new configuration data is available in the Table 6.

The attacking midfielder class was eliminated because your cardinality is low. There was a total reduction of 99,372 records because they were considered potential outliers.

After the process of removing inconsistent data, it is possible to observe in the Table 6 the numerical difference between classes. While the midfielder has a

[6]Interquartile formula and Python encoding obtained from the site https://www.datacrucis.com/research/find-outliers-in-an-array.html em 12/12/2017.

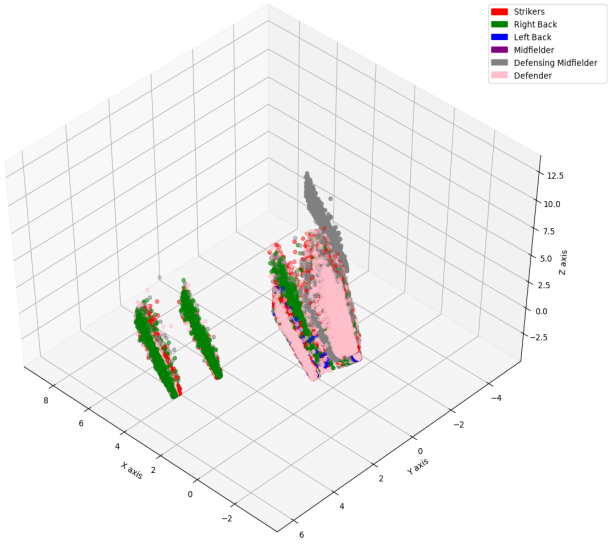

Figure 8: All classes in 3 dimensions adopting PCA and without outliers

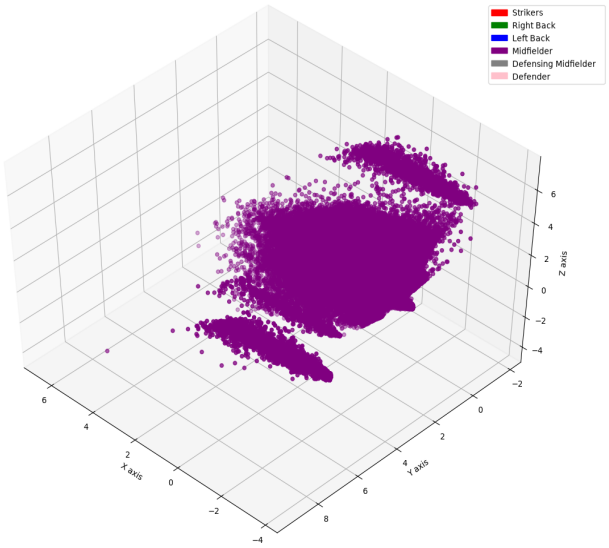

Figure 9: Midfielder in 3 dimensions adopting PCA and without outliers

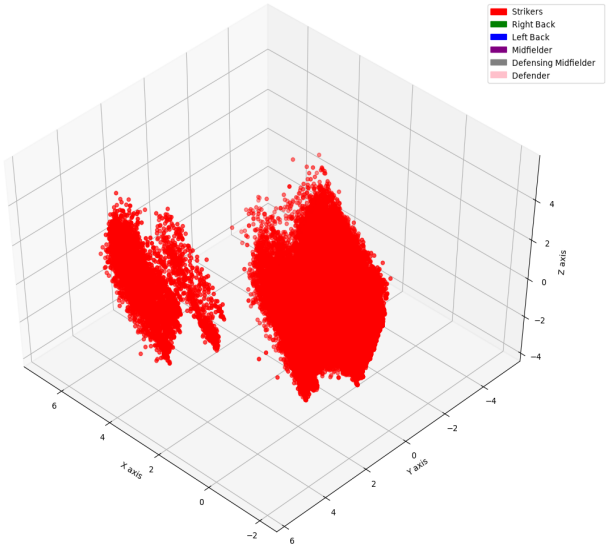

Figure 10: Strikers in 3 dimensions adopting PCA and without outliers

Table 6: Number of tuples of the player's classes in the database after the outliers elimination

Position	Tuples
Strikers	187.525
Right Back	109.887
Left Back	31.586
Attacking Midfielder	0
Midfielder	297.617
Defensing Midfielder	254.388
Defender	210.407
Total	1.091.410

total of 297.617 tuples, the left-back class has 31.586 tuples, a difference of approximately 942.24%. The disparity is a favorable condition for the prediction algorithm. The left-back is the set minority and it is possible to induce the predictions, defining new samples as this class. To reduce the numerical disparity„ the dataset was balanced. The ideal value of tuples per class is the number of 297.617 records, corresponding to the midfielder. The strategy used was to replicate the real data of the other classes. At the end of the balancing, all labeled positions had the same amount of tuples, totaling 1.785.702 records in the table "GPS" (Batista et al., 2004).

The balancing generated an addition of 694.292 tuples in the original data. This is an increase of 57.20%. The process has a considerable impact on the dataset. It may have positive impacts or generate results with underfitting or overfitting characteristics. To understand the scenario and to analysis, the results, the application of the algorithms used unbalanced and balanced data.

The original data were pre-processed as the below list:

- noise elimination;
- inconsistent data elimination;
- features transformations;
- standardization;
- elimination of attacking midfielder class;
- elimination of outliers;
- balancing.

The pre-processing was important to ensure more reliability for prediction models. The final visualization of the data is in Fig. 11 using the PCA reduction. After all stages, there was so much overlap.

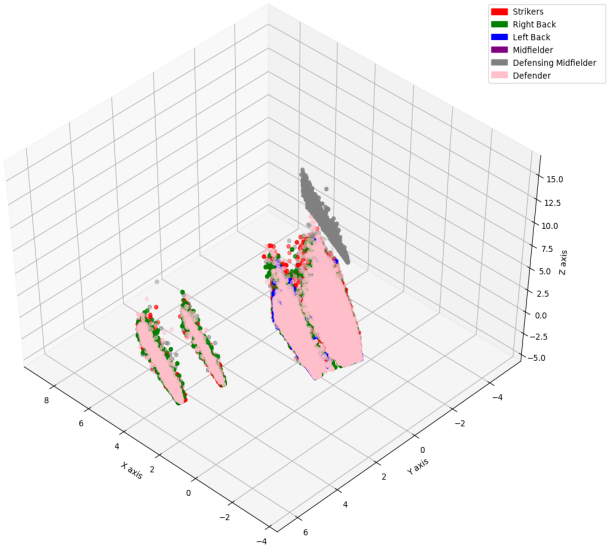

Figure 11: All balanced classes without outliers and reduced to 3 dimensions using PCA

A large addition of tuples was generated by the balancing. It is important to observe if it will have situations of overfitting or underfitting in the models. This situation should be monitored over the study.

3.1 Experimental Methodology

The experimental methodology is based on the execution using *10-fold*. The objective is to find the algorithm with the best hit rate, adopting the dataset proposed. The execution and the application will be for multi-class and binary problems (only two classes involved).

All algorithms were developed using *Python* language with support for *MySQL* ® database. The running was in the web environment, using the Apache server. The versions of the software used are available in the Table 7.

To avoid biased results, all executions used the preprocessed and random database.

To support the execution of the algorithms a tool was developed in *Python* with *HTML*. Its purpose is to act as a facilitator of the many executions required

Table 7: Software versions applied

Product/Service	Version
Apache	2.0
MySQL®	5.7.20
Python	2.7.13

during all the steps involved.

The tool is composed of two main files: *index.py*, which generates the interface and *functions.py*, which is responsible for the functions and integrations with the libraries in *Python*.

The python libraries used were: *mysqldb*[7], *matplotlib*[8], *numpy*[9], *mpl_toolkits*[10] and *sklearn*[11].

Considering that a multi-class problem is academically relevant because the results can be analyzed, it is necessary to consider the coach experience. He needs to have previous knowledge of soccer. It is relevant to consider the possibility of uncertainty between two ideal positions, where the coach's doubt persists, and thus a machine learning algorithm can help in this task. In this work, executions involving only two classes will be called a "binary problem".

Only will be compared the classes with correlations, because they may generate questions in the coach about the ideal positioning of the player.

To better understand the case, the algorithms for two opposing classes will be compared, more specifically striker and defender.

The classes for comparison are:

- Right Back and Right Left
  Both act on the sides of the field, but on opposite sides
- Midfielder and Defensing Midfielder
  Both act in the middle of the field, however the first in the central field and the second more recessed
- Striker and Defenser
  The striker has a characteristic of attacking the opposing goal, while the defender needs to defend its own goal. These positions are opposite, but this analysis is important because it helps to understand the behavior of the algorithms.

Table 8 shows the number of tuples to each execution with the unbalanced dataset.

The balanced base has a cardinality of 297.617 tuples for each class.

The applied algorithms were *k*-NN, decision trees, logistic regression, SVM, and artificial neural networks.

The balancing of the database has increased cardinality. All executions of the experimental methodology will adopt the balanced and unbalanced dataset. This was adopted to better understand the algorithms and their results. It is important to understand the impact of balancing on the problem.

One or more functions have been created to algorithms, adopting the libraries in *Python*. Other auxiliary functions help execution tasks.

All algorithms generated four distinct results. The executions occurred for the multi-class and binary problems, with the balanced and unbalanced dataset.

[7]http://mysql-python.sourceforge.net/MySQLdb.html
[8]https : //matplotlib.org/
[9]http://www.numpy.org/
[10]http://matplotlib.org/1.4.3 /mpl_toolkits/index.html
[11]http://scikit-learn.org/stable/

Table 8: Number of tuples to each execution with the unbalanced dataset

Class 1	Quantity	Class 2	Quantity	Total
Right Back	109.887	Left Back	31.586	141.473
Midfielder	297.617	Defensing Midfielder	254.388	552.005
Striker	187.525	Defender	210.407	397.932

4 Execution and Results

4.1 *k*-NN

The *k*-NN algorithm is a model where is extremely important to choose the correct value to *k*. It is important because is the number of neighbors and impact the prediction class. To find the best *k*, a search algorithm was adopted. It was executed repeatedly with different values for *k*. The value start was 3 and finished in 49. Each execution of the algorithm, *k* was incremented two. The ideal value obtained for *k* was 3. Fig. 12 shows the evolution search.

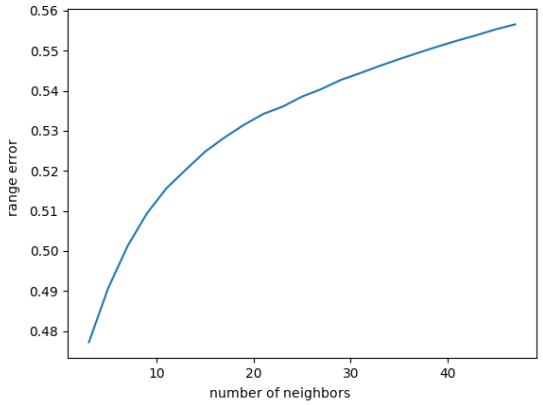

Figure 12: Search for best *k* to *k*-NN algorithm adopting balanced dataset

A low value for *k* ensures that only the closest neighbors are considered. Distance weights were not adopted, then all neighbors have the same importance in voting. Tests with different values for *k* were not performed. Fig. 12 illustrates a lower performance whenever its value is increased.

The hit rate, using the balanced dataset, was 52.27%, consuming approximately one and a half minutes to execution. But biased data was found, turning this result unreliable. The balancing technique consists of replicating the tuples of the minority classes. The cross-validation process was adopted and considering that *k*-NN works its classification by a distance of the nearest neighbors. The balancing, for this method, creates bias. The balancing caused duplicated samples. This process generated a "clone" causing, for many times, the zero distance as the best result. To confirm this theory, a search for the best *k* was improved, starting by 1. Fig. 13 shows the search evolution.

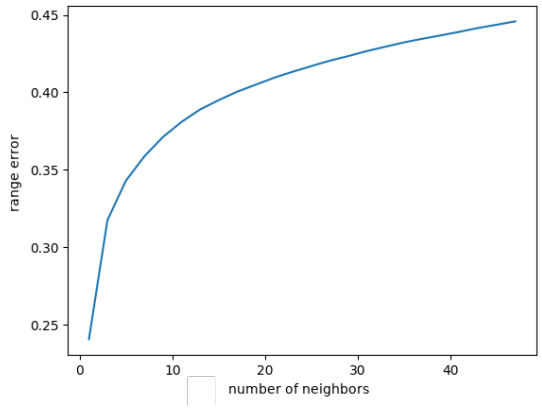

Figure 13: Search the best *k*, with *k* >= 1 and *k* <= 45, for balanced dataset

For the balanced dataset, the best *k* is 1, making it a *1*-NN algorithm. Thus, the balanced dataset will not be considered in the analysis of the *k*-NN. Only the result of the *k*-NN for the unbalanced base will be considered. Fig. 14 shows the search for the best *k*. The hit rate, for *k* equals 9, is 41.85%. The time consumed to run was approximately one minute.

It is a relatively simple algorithm, but its run time was positive, consuming less than two minutes for both scenarios. The hit rate can be considered relevant, since there was more than 40% accuracy, considering five distinct classes in the database.

4.2 Decision Trees

This model is impacted by the large volume of tuples, then the composition of the tree became extensive. Fig. 15 shows its assembly, however with a decreased number of data (only to demonstrates the graphical illustration about the tool developed to support this study).

The balancing does not turn biased the decision trees algorithm. The execution on the balanced dataset hit 14.80% in approximately seven minutes. On the unbalanced dataset, the hit rate reached 13.40% after

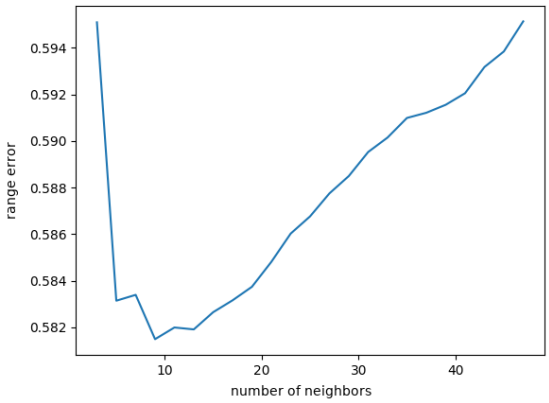

Figure 14: Search for the best *k* to*k*-NN algorithm adopting balanced dataset

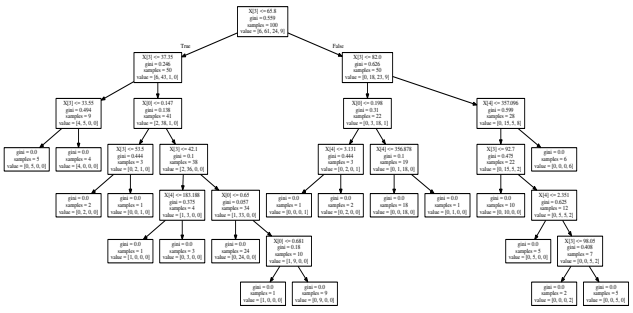

Figure 15: Decreased demonstration of the decision tree adopting partial data

approximately five minutes.

The trees working with a finite and simplified separation of their structure. A multi-class problem with similar samples, like this, turn a complex problem. Its hit rate was 14.80% for the balanced base against 13.40% for the unbalanced base. The difference in the hit rates can be considered as insignificant, totaling only 1.4%.

In the first moment, the results obtained with the decision trees are low relevance.

4.3 Logistic Regression

The logistic regression is based on the statistical method inferring categorical outputs. Its model allows us to work with multi-class problems, as applied, however, its best results are obtained in scenarios limited to two classes.

To illustrate the sigmoid model $f(z)$, only two classes were selected: defender and striker. Knowing that the classifier assumes an output based on $0 \leq f(z) \leq 1$, the Fig. 16 shows the function based on the balanced data.

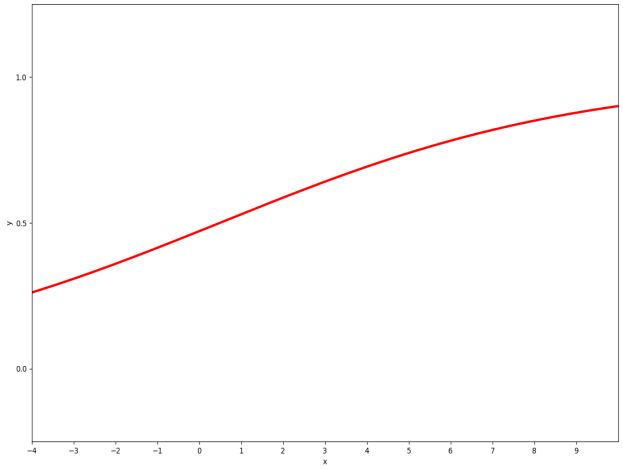

Figure 16: Chart of the sigmoid function to defender and striker classes

Based on a sigmoid function, the Fig. 17 shows the scattering of the two classes and their classifications in the model. How can see, the accuracy will be low.

This algorithm has a better performance when compared with decision trees, but the logistic regression did not exceed it. For the balanced dataset, the time consumed was approximately ten minutes. The hit rate was 17.42%. Repeating the execution, but adopting the unbalanced base, the total time was approximately seven minutes, generating a hit rate of 16.11%. Considering the balancing process, the gain was only 1.31 %.

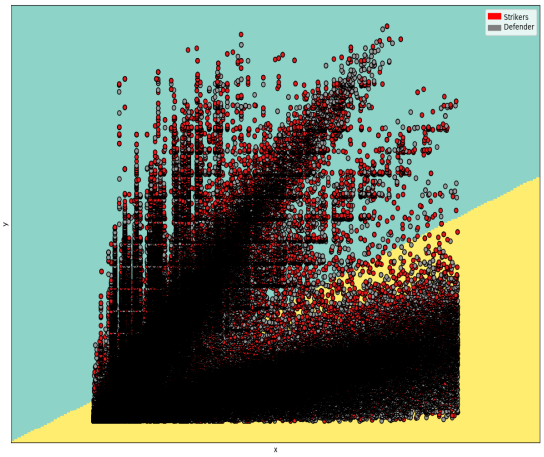

Figure 17: Separation of the defender and striker classes in logistic regression

4.4 SVM

The SVM model required a high computational cost, as expected because this algorithm works with binary problems. The execution of the multiclass problem is possible but it is necessary for the repetition of the classification process. The finish occurs only at the end of the class confrontation, always in alternating pairs. For the binary problem, the execution time was considerably shorter.

The execution of the SVM algorithm was considerably extensive, consuming approximately fourteen hours and nineteen minutes. Time size is justified by the complexity and choice of models adopted. The results returning a hit rate of 56.11%, for the balanced dataset. Adopting the same method for an unbalanced dataset, its execution was approximately 11 hours and forty-two minutes, generating a hit rate of 42.14%.

The SVM kernel adopted was RBF - Radial Basis Function. His choice is justified because it works with numbers based on distance. To calculate the interval between the points, the Euclidean method was used. Fig. 18 shows the application of the SVM algorithm for the right and left-back classes. The graphic seeing is easily for binary problems. To improve the legible visualization, partial sampling was selected, allowing us to understand the separation created.

4.5 Neural Network

For the backpropagation neural network, was adopted a model with the first layer and six neurons, corresponding to the selected number features. There are two hidden layers, the first with twelve neurons, followed by another with six. The output layer is

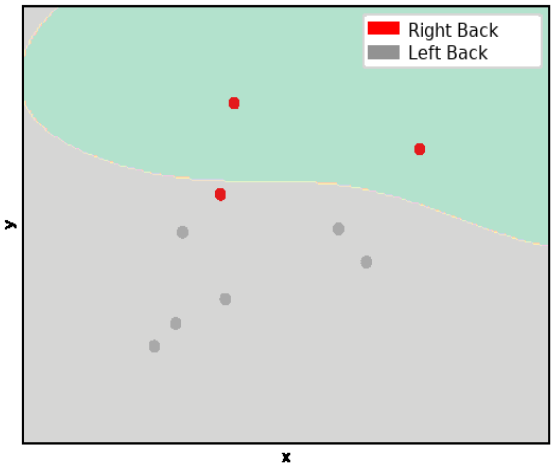

Figure 18: Separation of classes using SVM with kernel RBF

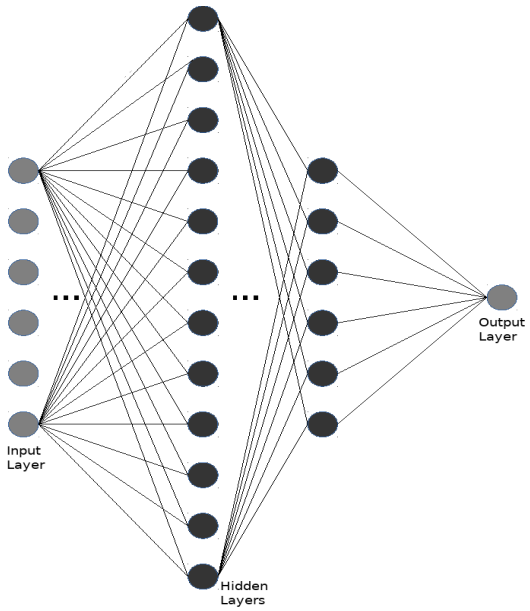

Figure 19: Layout of backpropagation neural network adopted

responsible for defining a single class. Fig. 19 shows the layout of the neural network.

The neural network algorithm consumed approximately twenty minutes to processing. Its hit rate was 16.66%. For the unbalanced base, the processing consumed approximately fourteen minutes. Its hit rate was 10.06%. The results are has a low hit rate but with acceptable execution time.

Table 9: Hit rates of the algorithms - multiclass problem

Algorithm	Hit rate(%)	Execution Time≈
k-NN - balanced	52.27	1.5 min
k-NN - unbalanced	41.85	1 min
Decision trees - balanced	14.80	7 min
Decision trees - unbalanced	13.40	5 min
Logistic Regression - balanced	17.42	10 min
Logistic Regression - unbalanced	16.11	7 min
SVM - balanced	56.11	14 h 19 min
SVM - unbalanced	42.14	11 h 42 min
Neural Networks - balanced	16.66	20 min
Neural Networks - unbalanced	10.06	14 min

Table 9 shows all the results. The SVM model obtained the best hit rates for the problem. Its performance was slow and the computational cost was high. It needs to adaptable to a multi-class problem and its efficiency was drastically affected.

4.6 Binary Problem

It is unusual for a coach to question about what is the ideal position of a football player, considering all possible positions. It is necessary to consider a coach as a prepared and experienced person about soccer. But it is possible to question about two different positions. Machine learning models can be potentially positive to assist him. To understand better, the algorithms were executed again. The dataset was limited with only two player target positions, running the algorithms with balanced and unbalanced data.

All executions adopted the same patterns, models, and values. The values of k for the algorithm k-NN was not changed. The positions confronted and their details are in Section 3.1.

Table 10 shows hit rates and execution time.

Table 10 shows the best way for the proposed problem. Considering the same dataset, but proceeding a binary analysis of the player's positions, the results collected are more expressive. The last column contains the average value of the runtime because the same algorithm was executed three times.

The values obtained for k-KNN using the balanced dataset were discarded again because they are biased.

The analyzes of the columns with the player's positions in the Table 10, for all algorithms applied in the experimental methodology, the best hit rate is obtained between right back and left back. The two positions act on opposite sides of the field and half of the attributes are related to the positioning, it is possible the existence of a biased situation. However, when analyzing the game dynamics and the fact that the dataset has information from the first and second time, where the teams alternate the sides of the field,

Table 10: Hit rates and execution time - binary classes

	Right Back x Left Back	Midfielder x Def. Mid.	Striker x Defenser	Execution Time≈
k-NN balanced	97.29%	65.36%	68.96%	0.7 min
k-NN unbalanced	91.21%	64.76%	66.31%	0.5 min
Decision trees balanced	63.91%	40.85%	35.86%	4.5 min
Decision trees unbalanced	67.39%	41.63%	35.46%	3 min
Logistic regression balanced	62.76%	49.78%	51.60%	0.5 min
Logistic regression unbalanced	77.67%	53.61%	53.10%	0.4 min
SVM balanced	96.36%	59.25%	60.33%	6 h 23 min
SVM unbalanced	91.50%	59.27%	60.54%	4 h 47 min
Neural networks balanced	71.28%	58.51%	54.87%	5 min
Neural networks unbalanced	78.16%	57.63%	55.29%	3.5 min

there is an equivalence of information, as shown in the Fig. 20.

The angle and distance relative to *corner_1* are consistent, but when teams change sides, the positions are reversed. The distance from the midfield is also maintained and does not induce the results.

Analyzing other players, the backs are positions more consistently because they alternate between defense, middle, and attack, but always in the sidelines. This virtual space makes smaller variations on the data, but with a clear separation in the classes. It is necessary to consider the other features, which are based on speed and distance since backs move more and with less intensity. Adopting all features, merging velocity and positioning, the algorithms were able to predict with greater accuracy if the player has characteristics of acting on the left or right side (Di Salvo et al., 2007, Scaglia et al., 1996).

The processing results to predict midfielder and defender midfielder returned low accuracy if compared with left back and right back. Analyzing these positions, backs acts in the side field, preserving a defined space. The midfielder acts centralized. This zone has intense movement and it is not possible to delimit the area, as shown in the Table 1. The center field is a shared region and the players taken actions during the game progresses. The prediction of the algorithms was affected because three features are based on positioning.

The results rates oscillated into 64.76% to 40.85%.

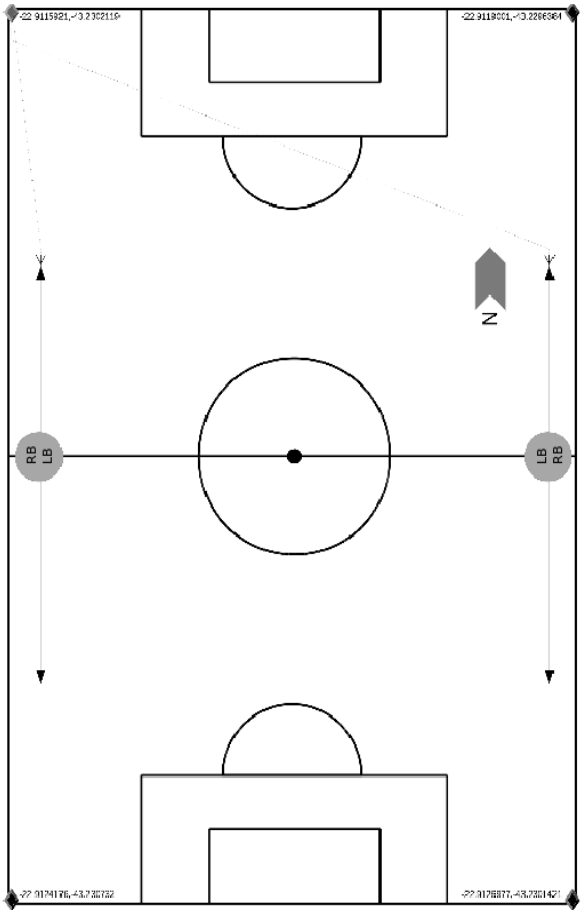

Figure 20: LB - Left back, RB - Right Back. Equivalence of angle and positioning for the sides

Considering the best result, it is possible to accept its relevance, specifically if compared to the multi-class problem.

The last execution occurred between the defender and striker. They act on opposite sides of the field and with different objectives, but both have some similar characteristics. The two positions have direct action with the goal, favor, and against. They run shorter distances in the field, however, their muscular explosions are bigger. The data generated for these two distinct positions have similarities, which makes the prediction more difficult (Di Salvo et al., 2007, Scaglia et al., 1996).

Hit rates ranged from 66.31% to 35.46%, as shown in the Table 10. The best result evidences the relevance of this study, considering to the positions with similarities data.

Analyzing execution time for the multi-class and binary problems, SVM was the algorithm with the highest computational cost. It is important to consider the technique used was cross-validation. This process is exhausting, repeating the training step. Considering the only prediction, the training runs only once, impacting in a shorter time to predict new inputs. The *10*-fold, proposed in this study, is computationally exhaustive for the SVM when considering the time consumed by the other algorithms.

An important comparison to be analyzed is the balancing technique. In preprocessing, it was adopted, however, the cardinality was reduced considerably. All algorithm was executed double times, always with a balanced and unbalanced dataset. The goal is to understand the behavior and how much gain or loss was computed.

For the multi-class problem, balancing acted positively. All the results were better when the technique was adopted. The average gain was 5.82%.

In the binary problem, there were losses when the balancing was adopted. On average for all runs, there was a loss of 2.15%. For the binary problem applied in the experimental methodology, balancing is not recommended. The only exception applies to the SVM algorithm, which has performed better or relatively equal to the unbalanced base.

The binary problem is closer to reality. The unbalanced dataset was unfavorable. Its adoption is not indicated in this study. Real problems do not have gains with the balancing technique (Batista et al., 2004).

In a situation where it would be appropriate to apply the multi-class problem, the SVM with balancing has the best hit rate. The unbalanced *k*-NN would be the second most indicated algorithm. It is necessary to remember that the first consumed more than fourteen hours, against only one and a half minutes of the second.

For the binary problem, a simple average hit rate was calculated. Table 11 shows the calculated averages.

Considering the preprocessing applied and other applied techniques, the algorithm most indicated, in this study, is *k*-NN without balancing. The second best is the SVM with balancing. It is important to remember the computational cost is very divergent between two models.

Table 11: Best hit rates for binary problem

Algorithm	Hit Rate
k-NN - Balanced	77.20%
k-NN - Unbalanced	74.09%
SVM - Balanced	71.89%
SVM - Unbalanced	70.44%
Neural Networks - Unbalanced	63.69%
Neural Networks - Unbalanced	61.55%
Logistic Regression - Unbalanced	61.46%
Logistic Regression - Balanced	54.71%
Decision Trees - Unbalanced	48.16%
Decision Trees - Balanced	46.87%

Two different problems were approached: multi-class and binary. In both, the two best algorithms were *k*-NN and SVM.

Analyzing the complexity level, the models are adaptive and divergent. *k*-NN is based on the distance technique. It is extremely simple and efficient, because the selected attributes have positional correlations, influencing the data spreading. SVM needs high processing. It was adopted the RBF kernel, which is also based on distance and may have favored the positive result. Is the results were expressive.

5 Discussion

Analyzes by GPS and video are adopted in professional sports, like soccer. Machine learning was integrated with GPS data, challenging the union of both. The goal was to understand the results in predicting to ideal positioning of professional players in Brazilian soccer.

The main goal was to understand how the machine learning algorithm has greater reliability to infer the tactical position of a professional player of Brazilian soccer. To find the most reliable answer, several techniques and models were applied.

The execution of algorithms was adjusted for the current problems. However, the models were directly impacted by preprocessing. New features were generated from the current dataset.

Considering the six features used in the algorithms, three were generated from the preprocessing. It was possible to get new features from analyzing the problem and dataset. It needed to reposition the players in a single field because the games occurred in different places. This process was not sufficient to understand the player's behavior since it was not possible to identify on which side the athlete was acting, because there was no identification of the first and second-time game. The solution was to create new features. It was possible to generate independent variables without correlation with the acting side.

The execution of the machine learning algorithms was so important, however, the preprocessing has a big impact on the results.

It is possible to conclude, specifically in this study, that the preprocessing step is more impacting than the application of machine learning algorithms.

The multiclass application was not successful. It returns low hit rates is not compatible with the reality of Brazilian professional soccer.

Analyzing hit rates in the binary problem, the result levels are satisfactory and can be considered acceptable to the proposal. The algorithms k-NN and SVM obtained the best hit rates: 74.09% and 71.89% respectively.

The best performances were with models k-NN and SVM with RBF kernel. It is important to remember that both worked their predictions based on Euclidean distances. This is an important feature present in both algorithms and may be responsible for higher hit rates.

The balancing applied in preprocessing and used during the execution of the algorithm, its positive impact varied according to the combination test. It is possible to observe that it is a significant gain, but there are not enough elements to recommend or refute the use of this technique in this study.

6 Conclusion

This study was focused on two aspects: the academic approach and the solution to a real problem related to a company. It show that it is a possible academy and business work together, and cooperation brings positive results for both. This is the first contribution of this study.

Considering the original dataset features were not enough, some techniques were applied to generate new data, getting better hit rates. In this way, there was a contribution to the techniques that allowed us to obtain new features.

It was possible to observe that a multiclass problem may not be relevant in Brazilian professional soccer. The results demonstrate a low hit rate when all player positions are included in the predictions of cross-validation. Another contribution is a recommendation on the class comparison, limiting them to a binary problem.

The algorithms executed in the binary problem it is recommended to use the algorithms k-NN or SVM. It is important to remember that the first model is simpler and its execution time consumes only a few seconds by cross-validation.

Acknowledgments

We are grateful to One Sports to provide all GPS data to execute this study.

References

Aughey, R. J. and Falloon, C. (2010). Real-time versus post-game gps data in team sports, *Journal of Science and Medicine in Sport* **13**(3): 348 – 349. https://doi.org/10.1016/j.jsams.2009.01.006.

Barbero-Álvarez, J. C., Coutts, A., Granda, J., Barbero-Álvarez, V. and Castagna, C. (2010). The validity and reliability of a global positioning satellite system device to assess speed and repeated sprint ability (rsa) in athletes, *Journal of Science and Medicine in Sport* **13**(2): 232–235. http://dx.doi.org/10.1016/j.jsams.2009.02.005.

Barros, R. M., Misuta, M. S., Menezes, R. P., Figueroa, P. J., Moura, F. A., Cunha, S. A., Anido, R. and Leite, N. J. (2007). Analysis of the distances covered by first division brazilian soccer players obtained with an automatic tracking method, *Journal of Sports Science and Medicine* pp. 233–242. Available at http://hdl.handle.net/11449/69706.

Batista, G. E. A. P. A., Prati, R. C. and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor. Newsl.* **6**(1): 20–29. http://doi.acm.org/10.1145/1007730.1007735.

Bourke, A. (2003). The dream of being a professional soccer player, *Journal of Sport and Social Issues* **27**(4): 399–419. http://dx.doi.org/10.1177/0193732503255478.

Dallaway, N. (2014). *Movement profile monitoring in professional football*, Master's thesis, University of Birmingham. Available at http://etheses.bham.ac.uk/5044/.

Di Salvo, V., Baron, R., Tschan, H., Calderon Montero, F., Bachl, N. and Pigozzi, F. (2007). Performance characteristics according to playing position in elite soccer, *International journal of sports medicine* **28**(3): 222–227. https://doi.org/10.1055/s-2006-924294.

Edgecomb, S. and Norton, K. (2006). Comparison of global positioning and computer-based tracking systems for measuring player movement distance during australian football, *Journal of Science and Medicine in Sport* **9**(1): 25 – 32.

Faceli, K., Gama, J., Lorena, A. and De Carvalho, A. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*, Grupo Gen - LTC. Available at https://books.google.com.br/books?id=4DwelAEACAAJ.

Hair, J. and Anderson, R. (2005). Rl: Black, wc análise multivariada de dados. 5ª edição.

Hennessy, L. and Jeffreys, I. (2018). The current use of gps, its potential, and limitations in soccer, *Strength and Conditioning Journal* **40**: 1. 10.1519/SSC.0000000000000386.

Hennig, E. and Briehle, R. (2000). Game analysis by gps satellite tracking of soccer players, *Archives of Physiology and Biochemistry* **108**(1-2): 44–44.

Meyer, T., Ohlendorf, K. and Kindermann, W. (2000). Longitudinal analysis of endurance and sprint abilities in elite german soccer players, *Deutsche Zeitschrift für Sportmedizin* **7**(8): 271–277. Available at https://www.researchgate.net/publication/

282684850_Longitudinal_analysis_of_endurance_and_sprint_abilities_in_elite_German_soccer_players.

Mitchell, E., Monaghan, D. and O'Connor, N. E. (2013). Classification of sporting activities using smartphone accelerometers, *Sensors* **13**(4): 5317–5337. https://doi.org/10.3390/s130405317.

Okazaki, V. H. A., Okazaki, F. H. A., Dascal, J. B. and Teixeira, L. A. (2012). Ciência e tecnologia aplicada à melhoria do desempenho esportivo, *Revista Mackenzie de Educação Física e Esporte* **11**(1). Available at http://cev.org.br/biblioteca/ciencia-tecnologia-aplicada-melhoria-desempenho-esportivo.

Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D. and Giannotti, F. (2020). *Metadata record for: A public data set of spatio-temporal match events in soccer competitions*. https://doi.org/10.6084/m9.figshare.9711164.v2.

prozone (2016). Find your sports data. Available at http://prozonesports.stats.com.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**(5500): 2323–2326. http://dx.doi.org/10.1126/science.290.5500.2323.

Scaglia, A. J. et al. (1996). Escolinha de futebol: uma questão pedagógica, *Motriz* **2**(1): 36–43. https://doi.org/10.5016/6513.

Sienkiewicz-Dianzenza, E., Rusin, M. and Stupnicki, R. (2009). Resistência anaeróbica de jogadores de futebol, *Fitness & performance journal* (3): 199–203. http://dx.doi.org/10.3900/fpj.8.3.199.p.

Strauss A, Sparks M, P. C. (2019). The Use of GPS Analysis to Quantify the Internal and External Match Demands of Semi-Elite Level Female Soccer Players during a Tournament, *Journal of Sports Science and Medicine* **18**(1). Available at https://pubmed.ncbi.nlm.nih.gov/30787654/.

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations, *Survey Review* **23**(176): 88–93. http://dx.doi.org/10.1179/sre.1975.23.176.88.

Wold, S., Esbensen, K. and Geladi, P. (1987). Principal component analysis, *Chemometrics and Intelligent Laboratory Systems* **2**(1): 37 – 52. https://doi.org/10.1016/0169-7439(87)80084-9.

Yeh, T. K., Wang, C. S., Lee, C. W. and Liou, Y. A. (2006). Construction and uncertainty evaluation of a calibration system for gps receivers, *Metrologia* **43**(5): 451. Available at http://stacks.iop.org/0026-1394/43/i=5/a=017.