NOVA IMS Information Management School

MDSAA

Master's Degree Program in

Data Science and Advanced Analytics

Development of a Market Value Metric – Made in Benfica

Rodrigo Vicente Pimenta

Internship Report

Presented as a partial requirement for obtaining the Master's Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

DEVELOPMENT OF A MARKET VALUE METRIC – MADE IN BENFICA

by

Rodrigo Vicente Pimenta

Internship report presented as a partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a Specialization in Business Analytics.

Orientador: Pedro da Costa Brito Cabral

Julho 2023

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisboa, Julho 2023

ACKNOWLEDGMENTS

I would like to thank the Sport Data Science team at Benfica and specifically Bruno and Miguel for the availability they had with me during the project.
I would also like to thank Professor Pedro Cabral for having guided this work and for the brevity and availability he has always shown to me.
I would also like to thank my girlfriend Catarina for her patience and unconditional support throughout this process.
Last but not least, I would like to thank my family, mother, father, sister and brother for their support and for helping me with everything I needed, a special thanks to the Pimentinha family.

ABSTRACT

This thesis focuses on the analysis of player transfers in football, with a specific emphasis on Benfica, the Portuguese club known for its significant investments in player acquisitions. The study aims to determine a market value that aligns closely with actual transfer values while identifying the key features and factors that play a crucial role in this context. By understanding these factors, the research aims to predict the market value of the players, in order to help the club's decision-making process. To achieve these objectives, the data was collected from various leagues, including the Portuguese, English, and French leagues, which are of interest to Benfica's scouting department. The TransferMarket website served as the data source, with data extraction performed using the *BeautifulSoup* library in Python. The analysis phase involved exploring and identifying the most relevant variables and their impact on calculating the market value using three modeling techniques: Decision Trees, Random Forest, and XGBoost. This process was conducted twice, once for the global dataset and once specifically for Benfica's purchases, allowing for a comparison between the two scenarios. The conclusions revealed that players who have previously played in the Premier League or Eredivisie and are international players for their respective national teams are highly prioritized by Benfica. Furthermore, factors such as the time remaining on a player's contract and their age were identified as important variables in determining market value. The Random Forest model exhibited the best performance, with a higher r-squared value (0,919 for Benfica dataset and 0,953 for the Global dataset) and a lower Mean Absolute Error (3818086,11 for Benfica dataset and 4038199,56 for the Global dataset) for both the TransferMarket and Actual Transfer Value problems, respectively. This study points to the significance of player transfers in professional football and highlights the potential benefits of employing data-driven approaches for decision-making by clubs. By incorporating these methodologies and leveraging comprehensive information, clubs can make more informed decisions and improve their overall strategic planning.

KEYWORDS

Sports Analytics; Market Value; Data Analysis; Decision Trees, Random Forest, XGBoost, Decision-Making

INDEX

1. INTRODUCTION 1
2. THEORETICAL FRAMEWORK 5
3. METHODOLOGY 8
    3.1. Web Scrapping 9
    3.2. Organize the Database 10
4. RESULTS 15
    4.1. Data Understanding 15
    4.2. Modeling Phase 22
        4.2.1. Decision Trees 23
        4.2.2. Random Forest 29
        4.2.3. XGBoost 35
5. CONCLUSION 41
BIBLIOGRAPHY 43

LIST OF FIGURES

Figure 1 – Transfers Value (Millions €) that the top leagues spent from 12/13 to 22/23 season (TransferMarket, 2023). 1
Figure 2 – Transfers Value (Millions €) that the top 3 teams in Portugal spent at the last decade (TransferMarket, 2023). 3
Figure 3 – Flowchart of the methodology applied in this project 8
Figure 4 - Numeric Variables' Histograms 16
Figure 5 - Numeric Variables' Box Plots 17
Figure 6 - Correlation Matrix 18
Figure 7 - Categorical Variables' Absolute Frequencies 18
Figure 8 - Numeric Variables' Histograms, after cleaning phase 20
Figure 9 - Numeric Variables' Boxplots, after cleaning phase 21
Figure 10 - Categorical Variables' Absolute Frequencies 21
Figure 11 - R-squared values in Decision Trees with Benfica Data, to check the best scaler.. 24
Figure 12 - MAE values in Decision Trees, to check the best scaler. 25
Figure 13 - R-squared values in Decision Trees with Global Data, to check the best scaler on TM model. 26
Figure 14 - R-squared values in Decision Trees with Global Data, to check the best scaler on ATV model 28
Figure 15 - R-squared values in Random Forest with Benfica Data, to check the best scaler 30
Figure 16 - MAE values in Random Forest, to check the best scaler. 31
Figure 17 - R-squared values in Random Forest with Global Data, to check the best scaler on TM model. 32
Figure 18 - MAE values in Random Forest with Global Data, to check the best scaler on ATV model. 34
Figure 19 - R-squared values in XGBoost with Benfica Data, to check the best scaler. 35
Figure 20 - MAE values in XGBoost, to check the best scaler. 37
Figure 21 - R-squared values in XGBoost with Global Data, to check the best scaler on TM model. 38
Figure 22 - MAE values in XGBoost with Global Data, to check the best scaler on ATV model. 39
Figure 23 - Example of the Final Report 41

LIST OF TABLES

Table 1 - Tuning for Decision Trees Benfica Data (r-squared)..... 24
Table 2 - Top 10 Features for the Decision Trees Model with TM values and Benfica Data... 25
Table 3 - Top 10 Features for the Decision Trees Model, with ATV and Benfica Data..... 26
Table 4 - Tuning for Decision Trees Global Data (r-squared) 27
Table 5 - Top 10 Features for the Decision Trees Model, with TM value and Global Data..... 27
Table 6 - Top 10 Features for the Decision Trees Model, with ATV value and Global Data.... 28
Table 7 - Tuning for Random Forest Benfica Data (r-squared) 30
Table 8 - Top 10 Features for the Random Forest Model with TM values and Benfica Data . 31
Table 9 - Top 10 Features for the Random Forest Model, with ATV and Benfica Data 32
Table 10 - Tuning for Random Forest Global Data (r-squared). 33
Table 11 - Top 10 Features for the Random Forest Model, with TM value and Global Data . 33
Table 12 - Top 10 Features for the Random Forest Model, with ATV value and Global Data 34
Table 13 - Tuning for XGBoost Benfica Data (r-squared) 36
Table 14 - Top 10 Features for the XGBoost Model with TM values and Benfica Data 36
Table 15 - Top 10 Features for the XGBoost Model, with ATV and Benfica Data 37
Table 16 - Tuning for XGBoost Global Data (r-squared)..... 38
Table 17 - Top 10 Features for the XGBoost Model, with TM value and Global Data 39
Table 18 - Top 10 Features for the Random Forest Model, with ATV value and Global Data 40

LIST OF ABBREVIATIONS AND ACRONYMS

MV	Market Value
TM	TransferMarket
ATV	Actual Transfer Value
OHE	One Hot Encoding
MAE	Mean Absolute Error
FIFA	Féderation Internationale de Football Association
UEFA	Union of European Football Associations

1. INTRODUCTION

Football is a collective sport originated in Britain in the 19th century, and since then has suffer a lot of changes, on the rules, players on the pitch and specifically at the audience. According to the International Association Football Federation (FIFA), that is the main organization that manage world football, close to 1.5 billion people watched the final of the Qatar World Cup final (FIFA, 2023). Given that is fair to say that football is one of the most viewed sports in the world.

Football is also an industry that moves a lot of money, specifically in players transfers. The amount of money that clubs are paying on players transfers tend to increase with time. For example, according to Gornic (2019), in the transfer market in 2018 was invested more 10,8% than in 2017. However, in the past 3 years, we lived with a pandemic and the clubs reduced the levels of investment because the incomes (mostly tickets to attend the matches) were dramatically reduced, as we can see on Figure 1. But in general, we can confirm the value the clubs are paying for the players in increasing.

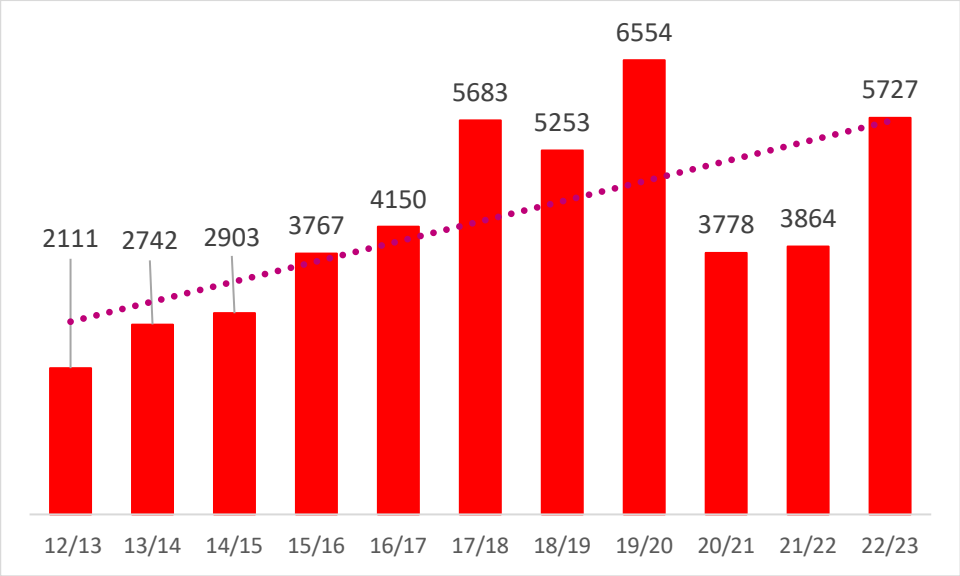

Figure 1 – Transfers Value (Millions €) that the top leagues spent from 12/13 to 22/23 season (TransferMarket, 2023).

This monetary value that the clubs pay in exchange off some players is directly linked to the Market Value (MV) of each player. The MV is an amount that is associated with each player and tends to follow the level of the athlete's career. In other words, if the athletes are preforming great in his team, and the team is getting good results in their games, his MV tends to increase, in the same way, if the player, for some reason, starts to not be on the coach's options or the team results are not that good, his value will tend to go down.

There has also been a relative increase in the price that a club have paid for a player for a transfer. For example, the top 13 most expensive transfers were above 100 million euros (TransferMarket, 2023). 8 of the top 13 most expensive transfers happened on the last 5 seasons. Is very important that the clubs are aware of the MV and the characteristics of a player, so that the club does not spend more than the fair value on players that doesn't have that high valuation. The opposite is also true, the goal of the clubs is to pay a fair value comparing with his MV and characteristics.

Nowadays, all the big clubs have a Scouting Department that is in charge to search for new talent and for certain players that may fit on the team. As an example of this, there is a club in the Eredivisie, the Netherlands Championship, that on season 2018/19, started to used data and technology to evolve the recruitment process, and they said that every club on that league has benefit on getting access to this kind of data because it allows them to benchmark the other clubs' players with their owns (Cooper, 2019). They also do Transfer Marker Analysis, so they can see if the clubs are undervaluing or overvaluing their players, in order to, if makes sense, the club know which is the price they need to spend on a certain player. Lastly, this department can also do some data analysis, so they can identify future potential talents using statistics of the game itself.

Sport Lisboa e Benfica (Benfica), as a big club not just in Portugal but in Europe (according to the club coefficients by the Union of European Football Associations (UEFA), Benfica is in top-20 best clubs in Europe (UEFA.com, 2023)), also have a Scouting Department to comply with all the explained needs and processes.

Benfica is a club based in Lisbon, it was founded in 1904 and has become, in our days, one of the biggest clubs not just in football but also in other sports. Recently the youth categories of football start also to get some good results, not just in Portugal but in Europe also. This is a consequence of the infrastructures and other aspects that the youth and the principal team has, so they can only think about the sport.

One of the more important changes Benfica did, has the creation of a department of Sports Data Science, that is always in communication with several other areas, in order to develop some analysis and give, in a very professionalized mode, all the information the organization need.

Benfica is the club in Portugal that most invest on players transfers, in order to try to get better results at all the competitions that the club is competing (TransferMarket, 2023). These values are increasing since the 13/14 season (Figure 2).

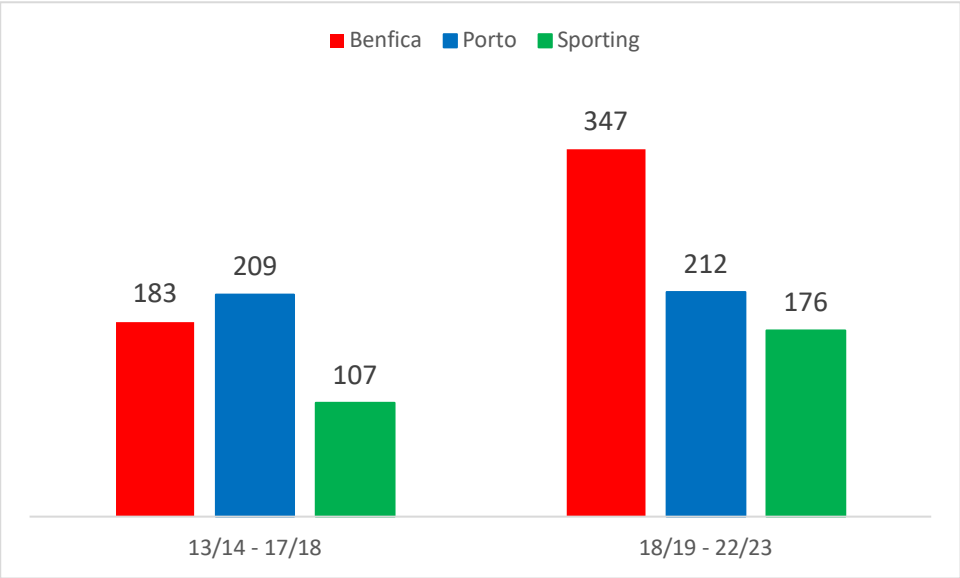

Figure 2 – Transfers Value (Millions €) that the top 3 teams in Portugal spent at the last decade (TransferMarket, 2023).

This project intents to support the Scouting Department in order to give information to provide some analysis and conclusions about the Market Value of the professional football players. Benfica objective is to know the fair value before doing the financial operations.

Currently, Benfica's Scouting Department, as well as most Scouts from the world's biggest clubs, is guided by the MV on the TransferMarket (TM) website, the best-known, quoted and even consensual market value in the world of football.

TransferMarket is a German company that was founded in 2000 and is dedicated to football and the evaluation of the Market Value of several players (https://www.transfermarkt.com). This company calculates market values considering different pricing models, which are still non-public to this day. However, it is known that the main factor of this calculation is the TransferMarket community, because the members can discuss and evaluate the market values of the players. TM opens a forum at the time that they consider appropriate and then the community discuss if a player deserves an update on his MV (TransferMarket, 2021).

These values are usually updated twice during the season. However, if there is a big competition where many players deserve an update, this same update happens. Intermediate changes can also happen, where the most common thing is to update the market values of young players. This is because there was not much data on these players and for some reason, they started to have more minutes on the field and therefore, an update is made on their market value.

According to the TM, the most important factors for calculating the Market Value are: Future promises/expectations, age, performance in clubs and/or in the national team, league level and ranking, reputation/prestige, development potential specific effects of the league, marketing value, number and reputation of interested clubs, level of experience susceptibility to injuries, general "demand" and market "trends", among others (TransferMarket, 2021).

Even though TM is a reference for top clubs and for the community, in terms of Market Value of different players, most of the time this value is far from the value for which transfers are made. Two examples, on the Benfica perspective are:

- João Félix + 57.20 M € of his MV at the time of the transfer
  At the end of the 19/20 season, João Félix transferred from Benfica to Atletico Madrid. At the time of the transfer, and according to TM, his MV was 70 M €, however the transfer was 127.2 M €, leading to an overvaluation by almost 60 M €.
- Morato + 7.10 M € at the time of the transfer.
  Still at the beginning of the 19/20 season in Europe, Morato was transfer from São Paulo (Brazil) to Benfica, in order to join the B team. At the time, his MV was only 50 thousand €, but Benfica paid in total 7.60 M €, thus paying more than 150 times its market value.

The objective of this work focuses on determining a market value closer to the value of players' transfers and to understand which features and factors are more relevant. After doing that, it's possible to look for some examples and try to predict, as best as possible, the market value of some players that are important in this context.

In order to achieve those goals, the first step is to obtain the data. The data that was used in this project was obtained via the TM website. It is necessary to get the data from several leagues, such as, Portuguese, English, France, among others, that Benfica finds interesting in order to get some conclusions on players that the Scouting Department may be interested, now or on the future.

After having the data of the leagues that were consider, it was done some analysis in order to try to understand in what way and which variables are the most important for the calculation of the market value from TransferMarket. Then repeat this process, but restricted to Benfica's purchases, and consequently make a comparison between the two scenarios.

2. THEORETICAL FRAMEWORK

This chapter serves as a tool to understand the theoretical concepts, models and some previous research that are relevant to this project. It also allows to position this study within the knowledge and theoretical perspectives that are already done, leading to a deeper understanding of this project and its implications.

- **Sport Analytics**
  In our days, technology became a tool that can help people in their tasks and jobs. Sports, and football more specifically, are no different. Today all the big clubs use sport analytics in order to get to some better known and justified conclusions. Player tracking data analysis has become increasingly popular as a tool for performance analysis in sports, particularly in the context of football (Caine et al., 2018).
  Sports analytics involves the examination and modeling of sports performance using scientific methods. It encompasses activities such as analyzing structured historical data, employing predictive analytical models based on this data, and utilizing information systems to support decision makers. The goal is to assist organizations in gaining a competitive edge in the realm of sports by providing valuable insights and guidance (Morgulev et al., 2018)
  The world of sports has transformed from a simple recreational activity to one that incorporates scientific principles. Sports analytics has emerged as a combination of data collection, game prediction, and visualization of strategies using various tools and techniques. Its aim is to improve individual player performance and overall team outcomes. The field of sports analysis is anticipated to give rise to numerous new applications, benefiting end users, sports coaches, and sports managers alike (Bhatnagar & Babbar, 2019).
  Sports Analytics can help the coaches and the teams in the decision making in pre- and post-match, however according to Kröckel, P. (2019) some techniques can be useful for real-time decision making. This is important because coaches need to make decisions in a quick time and under pressure, and if theses coaches can have more information, they can make better decisions.
- **Web Scrapping**
  Web Scrapping is the automated extraction of data from websites. In this project it was used the Python library *Beautiful Soup* that provides convenient methods for navigating, searching and extracting data from web pages.
  According to Robot (2022) in one project, about coroner reports to prevent future deaths, before using web scrapping the data scientists needed to manually extract the data, and they only could do 25 cases per hour. Using web scrapping they can increase that number to 1000 cases per hour.
- **Try-except method**
  This method was used in this project in order to avoid abruptly stops. On the web scrapping phase, the internet connection is important, in order to extract the data to the files. However sometimes this connection can have breaks, leading to an exception called *ConnectionError*. By implementing a try-except statement, it's possible to handle effectively potential issues instead of abruptly stopping the program. Failure to appropriately handle exceptions may result in an unexpected termination of the program (Try And Except in Python - Python Tutorial, n.d.).

- **Data Cleaning and Understanding**

  Data preprocessing is a crucial aspect of data mining, involving the preparation and transformation of data into a suitable format for mining procedures. Its objectives include reducing data size, identifying relationships between data elements, normalizing data, and extracting relevant features (Alasadi, & Bhaya, 2017).

  In this project there are some null values that were not supposed to be empty, such as, Market Value, preferential foot, among others. This process of leading with these errors involve identifying data patterns that for some reason have been distorted or simply there are lack of data (Bacao & Lobo, 2007).

  Feature selection is the method to choose a lower number of variables amongst all that are available in the data set. This process is important in order to deal to the curse of dimensionality problem. This problem urges to try to get a commitment between the extra information that a lot of variables would give to the model and the simpler representation of the data (Bacao & Lobo, 2007).

  On the metric features it was used the Pearson Correlation Coefficient that shows if two feature are related, positively, if the coefficient is near to 1.0 or negatively, if the coefficient is near to -1.0 (Turney, 2022). In this kind of projects is important not only know if two variables are highly connected, and if so, if the model would use both those variables it will be redundant, but also to know which variables are highly connected with the objective feature.

  For the non-metric features, it was used the Chi-Squared test of independence. This test is a nonparametric hypothesis and tells if some features are significantly related to the objective variable. If the variable is significantly important, then that variable stays for the model, in other hand, if the feature is not important, then it's dropped (Turney, 2022a).

- **One Hot Encoding**

  This method is crucial when dealing with categorical variables. The input in this method should be an array denoting the values taken on by categorical features. These features after the process would create a binary column for each category and returns a matrix. That matrix has only 0 or 1, in case of that category matches that column. This method will create as much columns as different categories are in one feature (Sklearn.Preprocessing.OneHotEncoder , n.d.).

  There are other methods to encode the categorical variables, such as the label encoding method. In this work was chosen the One-Hot Encoding in order to solve some ordering issues that the label method have and can be misunderstood by some machine learning algorithms (Al-Shehari & Alsowail, 2021).

- **Normalization**

  This method involves adjusting the values of various variables to a common scale. By doing so, it ensures that no singles variable overwhelms the others, allowing the data analysis method to treat the data in a balanced manner (Bacao & Lobo, 2007).

  Min Max Scaler transforms features by scaling each feature to a given feature range (Sklearn.Preprocessing.MinMaxScaler, n.d.). In this project it was used two feature ranges, [0,1] and [-1,1]. The first range is used when the data has positive values, the second one is when it's desirable to have features centered around the value 0.

Standard Scaler transforms features by removing the mean and scaling to unit variance (Sklearn.Preprocessing.StandardScaler, n.d.). This new value is calculated by the next formula $z = (x - u)/s$ where *u* is the mean, *s* is the standard deviation and x is the observation.

Robust Scaler removes the median and scales the data according to the quartile range. Normally the normalization process is done by removing the mean and scaling to unit variance, but outliers can influence those values, so in such cases, the median and the interquartile range give better results (Sklearn.Preprocessing.RobustScaler, n.d.).

- **Tuning the models**

  Hyperparameters are parameters in machine learning models that are set before the training process and cannot be updated during training. These parameters are integral to defining the model's architecture (Yu & Zhu, 2020). Due to their impact on training accuracy and speed, it is crucial to configure them based on prior experience before initiating the training process (Rodriguez, 2018).

- **Train Test Split**

  This method splits the data into random train and test subsets (Sklearn.Model_Selection.Train_Test_Split, n.d.).

  An effective and trusted method, known as the Train/Test Split, involves dividing a dataset into two parts: one for training a machine learning model and another for validation purposes. By reserving a portion of unseen data for testing, this approach provides an unbiased evaluation of the model's performance when it is deployed for real-world predictions. This allows for a reliable estimation of the model's effectiveness in practical situations (Vabalas et al., 2019).

- **Repeated Stratified K-Fold**

  This process split the data in *k* subsets of equal size, then each subset is used for testing and the other ones remained for training (Sklearn.Model_Selection.RepeatedStratifiedKFold, n.d.).

  Using Repeated K-Fold Cross-Validation increase the accuracy of estimating the average model performance, although at the expense of fitting and evaluating a larger number of models (Brownlee, 2020).

- **Performance Measures**

  In this project it was used two different performance measures, R-Squared and Mean Absolute Error (MAE).

  R-Squared or coefficient of determination has as best score 1.0 and that tells that the model makes perfect predictions. But if the result of this coefficient is near 0.0 the predictions are imperfect.

  However, this measure has a lack of context and it's essential to interpret the R-squared with some other measures (Sklearn.Metrics.R2_Score, n.d.).

  And to get other perspective it was used MAE. This coefficient measures the mean absolute error regression loss. It has the advantage that is less sensitive to outliers since it only considers the absolute difference (Sklearn.Metrics.Mean_Absolute_Error, n.d.).

3. METHODOLOGY

In order to achieve all the objectives, the first step is to obtain the data via TM. For that it was used *BeautifulSoup*, a library from Python in order to do web-scrapping.

After having the data of every desired league, it's necessary to proceed to some analysis of the data itself, try to catch some outliers or values that doesn't make sense, dealing with null values and feature selection. Finally, it will be necessary to do one hot encoding, to be easier to deal with the categorical data and to scall the data.

Following this process, the data is ready for the predictive models, where was used three of them, Decision Trees, Random Forest and XGBoost. This process can be seen in Figure 1.

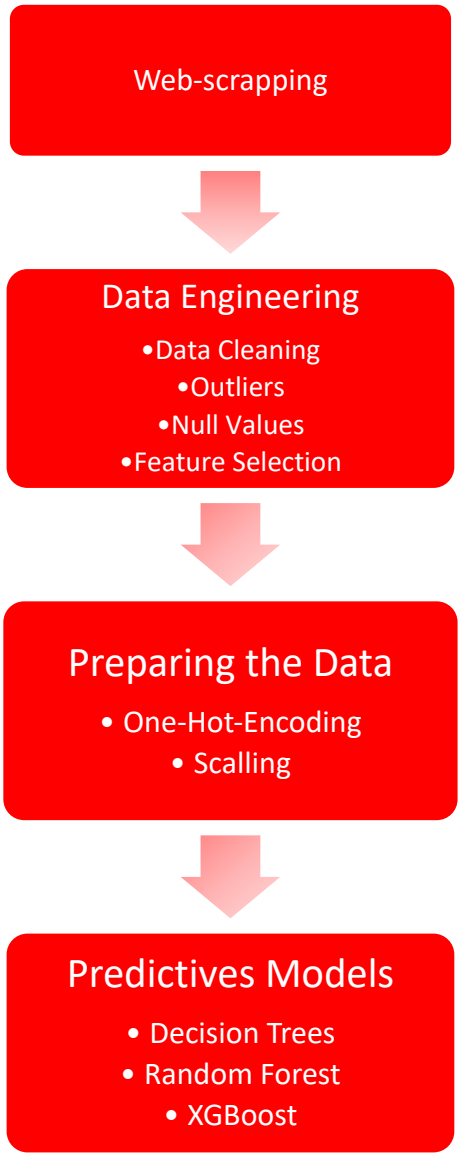

Figure 3 – Flowchart of the methodology applied in this project

3.1. WEB SCRAPPING

As mentioned before, the first sept of this project is to get the data organized in order to do the desired analysis and consequently get some conclusions. The main source of the data will be the TransferMarket website. In order to extract the data, was used the *BeautifulSoup* library from Python.

For the sake of being more organized and for the case that anything went wrong, it was written many functions to get the final database.

Firstly, there is the function ***ligas*** (no attributes). This function takes care of create a CSV with the principals' competitions that are on TransferMarket, given them an Id, without any order relation. On this file is included national leagues and cups, international competitions for club and national teams. The output of this function is a CSV file with the competition Id, the country, the kind of competition and its URL.

Next, there is the function ***df_clubes***, that as attributes have the league URL (*url_liga*) and the clubs CSV. The goal here is to create a CSV with the clubs from the pretend leagues. From the league URL and the library *BeautifulSoup*, is possible to extract the CSV file *clubes*. This file has as columns, the league (Liga), which is divided on country and kind of competition, the name of the club (Clube), the URL of the webpage of the club on TransferMarket (URL) and the league URL (URL_Liga).

Once the clubs from the desired league are on the *clubes* file, the following step is to get the squads of that league. To do that there is the function ***df_jogadores***, which receive a CSV with the clubs and another CSV with the players. This function is organized in a way to get the data of the players by club. That data is the Name (Nome), his Id on TransferMarket (Id), the URL of the TransferMarket website (URL), the club at the time of data extraction (Clube_atual) and an auxiliar name of each player (Nome_aux). This function will be used in each webpage of each club on the *clubes* file, in order to get the data that was mentioned before.

Below that, there will be the function ***df_PI_URL***. This function receives three files, *Players*, *PI* and *URL*. The *PI* file is where are the basic players information, such as, his Id on TransferMarket (Id), Birth Date (Data de nascimento), height (Altura), nationality (Nacionalidade), position (Posição), preferential foot (Pé preferencial), the date that the player join the actual team (Na equipa desde), the date that his contract expires (Contrato até), his outfitter (Equipador), his agent (Empresário), and the club that the player represents now (Clube atual). The *URL* file is a way to the next work to be more organized, because this file has the *URL* of the webpages of the Market Value, Stats, players' career on the national level, players' transfers, the injury history and the position on the field. This function will get all this data in the principal webpage of every player that is in the *Player*'s file. If for some reason the player doesn't have any information on that webpage, for example, his outfitter, that value will be null on the CSV. In the same way if the player doesn't have any career on the national team, the URL, on the *URL* file, will be also null.

The next step is to extract the path from every player in their national team. To do that, there is this function called ***seleção***, which has, as attributes, two CSV files, *URL* and *selecao.* For the next functions, they will be organized in the similar way, it will be done an iteration on the *URL* file, in order to get the URL that it's needed and after that the function extracts the data from that web page. On the case of the national teams' career, the data is obtained are, the national team, which has the country but also

the team, such as, principal national team, under-21, etc., debut on that team, how many games and goals, and the player age at the time of the debut.

Following the same idea as the function before, ***transferencias***, with two CSV files, *URL* and *transferencias*, as attributes, has the job to extract all the transfers from every player. The output of this function will be a file that has as columns the player Id, the Season, the date of the transfer, the clubs that the player left and joins, the market value at the time of the transfer, and the actual transfer value. In this file, in addition to the definitive transfers to a club to another, there is also the loan transfers.

The next function is ***ValorMercado***, which received the files *URL* and *VM* (Market Value). This function has as output a *VM* file which each line is an update of the Market Value of every player. Given that the columns of this file are the player ID, the date of the update, the market value after the update and the appreciation/devaluation compared to the market value before. If it is the first update, such as, it is the first time that a player has a Market Value, then his appreciation/devaluation will be null.

The last function on this web-scrapping phase is ***desempenho***. As the last functions, this receives two CSV files, *URL* and *desempenho*. However, this function gets another attribute, *temos_df*, which is a Boolean that the function will be explain later. This function is the one that consumes more time on the project, just because each line on this *desempenho* file is data from a game from a player, which means that this file will be holding data from every game that is available on TransferMarket that the player, that was on consideration, ever played in his whole career.

The columns of that file are the Player Id, the competition and the matchday, the day of the game on consideration, the home and away teams, the final result affect the player, meaning, if it is a Win, a Draw or a Defeat, the actual final result in terms of goals, the position of the player, goals, assists and own-goals of the player on that game, yellow card, double yellow card, red card, if he was substituted on or off and the minutes that he played. On the cards, and if he is substituted on or off, it shows the minute of that action.

3.2. ORGANIZE THE DATABASE

Now that all the data necessary is in several files, it's time to organize it on an only file. To do that, there is the function called ***make_df*** that receives the market value, stats, national team's career, transfers, personal information and leagues files, as well as the data frame *df*, that will be the output on this function. Each line on this file corresponds a time interval, that matches the new updates of the market value. In other words, each line matches the data from two updates on the market value. In the case that the update is the latest, the time interval will be from that date until the date that the data frame was build. The columns of this file are, player Id, the highest team on the national matter, that the player ever played, his height and age, the first and second nationality, his preferential foot, the months left on his contract, the outfitter, his agent, the club and position on the field, number of wins, draws, defeats, goals, assists, own-goals, minutes, yellow, double-yellow and red card, games as the players goes into the game from the bench, games that the player is substituted and injured, his market value and difference between the last valuation, the top three competitions that the player has more games and finally the date of beginning and end of the time interval in consideration.

This function starts to do some cleaning on the data, for example, transform the information about the cards, such as yellows or red, and substitutions from the minute that the action occurs to do a count, in order to after that do a sum between the two dates. Then the null values on the goals, assist and own-goals columns are filled with the value 0 (zero). And lastly, the minutes column is passed to *float64* type.

Just after all this steps, it starts the build of the main database. This phase is done from an iteration of the market value file, because the final database will have the same structure. The first step is to create an auxiliar data frame that will be restrict the stats data of the player and the time interval that will be consider, in order to add that information to the final database. After the stats data it's done, it's time to search on the national teams' data. If the player, at that time, never had any type of international game from his national team, then this value will be null. On the other hand, if the player has some path on the national team, is consider the highest team, on the final day of the time interval, that this player ever played.

The next thing to do is to get the number of games that player was injured on that period. To do that, we will use the fact that when the player is injured, TransferMarket marks the player position as 'Lesão', which means that the player is injured. After that we must figure out the top three competitions. This is done with a group by on thar auxiliar data frame by competition and with a count followed by a merge on the competition URL and with that we get the top three more disputed competitions on that time interval.

Now it's time to get the club info. To get that information, it will be using the transfers file and do a connection by the beginning data of the time interval and get the final club on that file. Next, from the basic information (*PI*) file we get the position of the player.

After that, it's needed two different lists, results and stats. On the results list will be the number of wins, draws and defeats on the period. On the stats list will be the number of goals, assists, own-goals, minutes, yellow, double-yellow and red cards, games that the player starts on the bench but enters the field, the total games that player were called, and the games that player was injured.

The Market Value and his appreciation/devaluation are withdrawn directly from the market value file. The age, height, nationality, number of months left on the contract, outfitter and the agent info come from the personal information file.

After all this steps, it's added a new line on the database and starts all over again on other period. When there were no more data on the market value file about this player, the function will pass to the next one. Once all players were considered, the final database is finished.

In order to be more organized, there is the function ***dataframe*** that receives the URL of the league and the files of the clubs, players, basic information, URL, stats, national career path, transfers, market value, the final data base and a Boolean. This Boolean has only one job. If equal to *True*, when asked to download the *df* file, if *False*, create a new one. This function calls every one of the functions that were describe before in order to get the final database in a complete and organized way.

The next function is called ***new_league***, which doesn't have any inputs. This function is interactive with the user. It's the function that is called every single time that we want to add a new league to the database. The first question that will appear to the user is if he wants a new data frame or if he already

has one and wants to continue to add more data. If the option of the new data frame is selected, then if there is some data, will be eliminated. Next is asked to the user to put a path to where the user want to create a folder named 'CSV'. This folder is where will be storage all the files that were mentioned before. After that, and if the user chooses to start over, will be created all the data frames with no lines. Next is called the function *dataframe* that will extract and organizes all the data. If there are already some data and the choice was to continue to add more, the function will ask where the files are, in order to continue the extract process. After that the function does the download of all files. If the league URL is already on the clubs table, then the user gets a message saying that the league is all in or part of it on the file. After that message the user has the option to put another league or to end the process. If the league is not on the table, it's called the *dataframe* function. If the user gives some invalid input the function returns a message of 'Input Inválido'.

This process is normally a very time consuming process, so for that reason and in order to be more automized exists the function ***error*** which receives two inputs, *where* and *flag*. *Flag* is a Boolean just like *temos_df* on the *desempenho* function. These Boolean points will check if exists a certain file or if is necessary to create a new one. *Where* is to get the place that the error appears. For all the project when is done a download of the html it's done by the try-except process. This process works in the way of try the code on the *try* part. If this code goes well with no errors, the process will continue as normal. However, if there is an error on that code, it will run the *except* code. On this case there is a specific error *ConnectionResetError*. This type of error occurs when the internet connection is off, or when exist any type of failure on those matters. It is here that the *error* function appears. This function is called every time that any *ConnectionResetError* type error appears, in a way that the process doesn't finish and to be possible to continue if there was an internet failure. This function normally is used on an automatically way, but if for some reason appears a different error, it's possible to call this function. The only two things that are needed are, find where the error was, and if it's needed to create a new file or if it's already one.

The last function is the ***update*** one. This function, as the names tells, is to do an update of the leagues that are already on the data base. This function has as inputs the files of the clubs, player, basic information, URL, stats, national team career, transfers, market value, leagues and a flag. This flag works on a same way as the one on the *error* function, because each file has his own marker, and if there is any error on the download of the HTML, is used the same try-except technique, i.e., it's called the *update* function with the right flag. On every file the update is done on a similar way. This function is written to update league by league, file by file. The first step is to update the data that are already on the file and only then, if it is the case, add more new data. When the *df* is been updated there is a sub-function ***row-df***, that from the file mentioned before returns a line of the *df* table. On this file, if there was not an update on the market value, is only updated the last row of this player. Otherwise, it's closed the line before, and it's opened a new line with the last market value.

Now that all the functions are ready, it's time to do the web-scrapping. The Leagues that were consider were, the top 5, which are, England, Spanish, Italian, Germany and France, Portugal League for obvious reasons, and then the Netherlands, Denmark, Norway, Argentina and finally Brazil, that are some markets that Benfica are used to sell and buy some players.

After the data were downloaded, it was done some data understanding and cleaning.

The first step is to clean the data, meaning that some values must change, for example, replace commas to dots in order to get a float instead of a string.

In this process is also created a new column, *ValorTransferencia*, which will have the transfer value on that period, if it happens. Firstly, the values of this new column are '-'.

In order to fill the *ValorTransferencia* column on the database, it is used a method that when the date of transfer is between the time interval of a certain line, the value on this new column will be the actual transfer value. If there is not a transfer for a player on a certain interval of time, then that value will continue to be '-'. This column will be useful for the analysis between the TransferMarket value of the players, and the actual transfer value that the clubs pay for a player. This analysis will appear next on this work.

After this new column and some initial cleaning, it was decided to do some analysis on the null values. On some columns it's normal to have a lot of null values, such as, the national team, the second nationality, the outfitter, and a few more. On these columns, for now, wasn't done nothing. However, some columns can't have null values. So, it was dropped the rows that were null on the Time left on the Contract, preferred foot, most played competition, club, height and Market value.

Then it was done two groups of analysis, the one with metric features and other with non-metric features. In this part it was used some graphics so the analysis could be done.

As a way to check some correlation between columns, it was used Pearson Correlation matrix. The reason why the Pearson method was chosen is because the objective is only to check the correlation between two columns, but also to check the correlation of every feature with the objective, which is the Market Value column (VM).

For the analysis of the categorical variables, it was done some charts in order to see the frequencies of the low cardinality features, such as, the national team, preferred foot, outfitter and position. After that, and to check all categorical variables it was done an Independence test of Chi-Squared, with an alpha of 0.0001.

Finally, the last step is to do some feature selection based on the previously obtained results.

After all this cleaning process, it was done the visuals again, in order to see the changes after the data cleaning phase.

In order to be able to do some analysis, it was done a One Hot Encoding on the categorical features. To do so, it was used the method OneHotEncoding and pass all the non-metric features that were left after the feature selection.

Before entering on the modelling phase, it was created four different files, that only differs on the scaling method used. It was used the Min Max Scaler, that will put the values between 0 and 1, Min Max Scaler with the values between -1 and 1, the Robust Scaler and the Standard Scaler.

This process was done twice given that fact that there is a more global database, where there are data from every player on the leagues that were mentioned, and a more restrict one, where are only the data from the players thar joined Benfica. In that case the data goes from the begin of the career of

the athlete until the moment that he joins Benfica, in order to try to understand the reasons or the factors that lead to Benfica to want the player.

4. RESULTS

4.1. DATA UNDERSTANDING

Given the fact that the data, from the TransferMarket website, is all available in files, it's time to do some data understanding and cleaning before leading to the model phase.

The first step is to clean the data, so, at the *height* column, it is replaced the comma to a dot, in order to pass the string to float, and replace the '-' to null at the Time Left on the Contract column. After that is created a new column *ValorTransferencia*, which, for now all, the values are '-'.

To fill this new column, it is used the file *transferencias*, that is the file that contains all the transfers of all the players. First, it is done some cleaning on this file too, such as, replace, 'custo zero' to '0', and the commas for dots. Then dropped the lines that on the column of the transfer value appear '?' as unknown, and 'draft'. If the date is also unknown that line is also dropped. For this work it isn't consider the loan transfers, just because, normally this transfer does not involve money, or when involves are low amount of money, so these types of transfer aren't considered. The last thing to do in this file is to replace 'mil €' to '*1000' and 'M €' to '*1000000', in order to get the actual transfer value.

Then it was done two groups of analysis, the one with metric features and other with non-metric features. The first thing done was some histograms for the numeric variables (Figure 4) and notice that on the wins, draws and defeats are some outliers, because there is a big bar near 0 and then only a few cases on the other bars. In order to get a more information it was done numeric variables box plots (Figure 5) and confirmed the described situation. The columns, wins, draws, defeats, goals, assists, own-goals, minutes, yellow, double yellow and red cards, sub in and sub out, called out and games injured have a lot of outliers. The other numeric columns also have outliers but not much as the ones mentioned before.

Figure 4 - Numeric Variables' Histograms

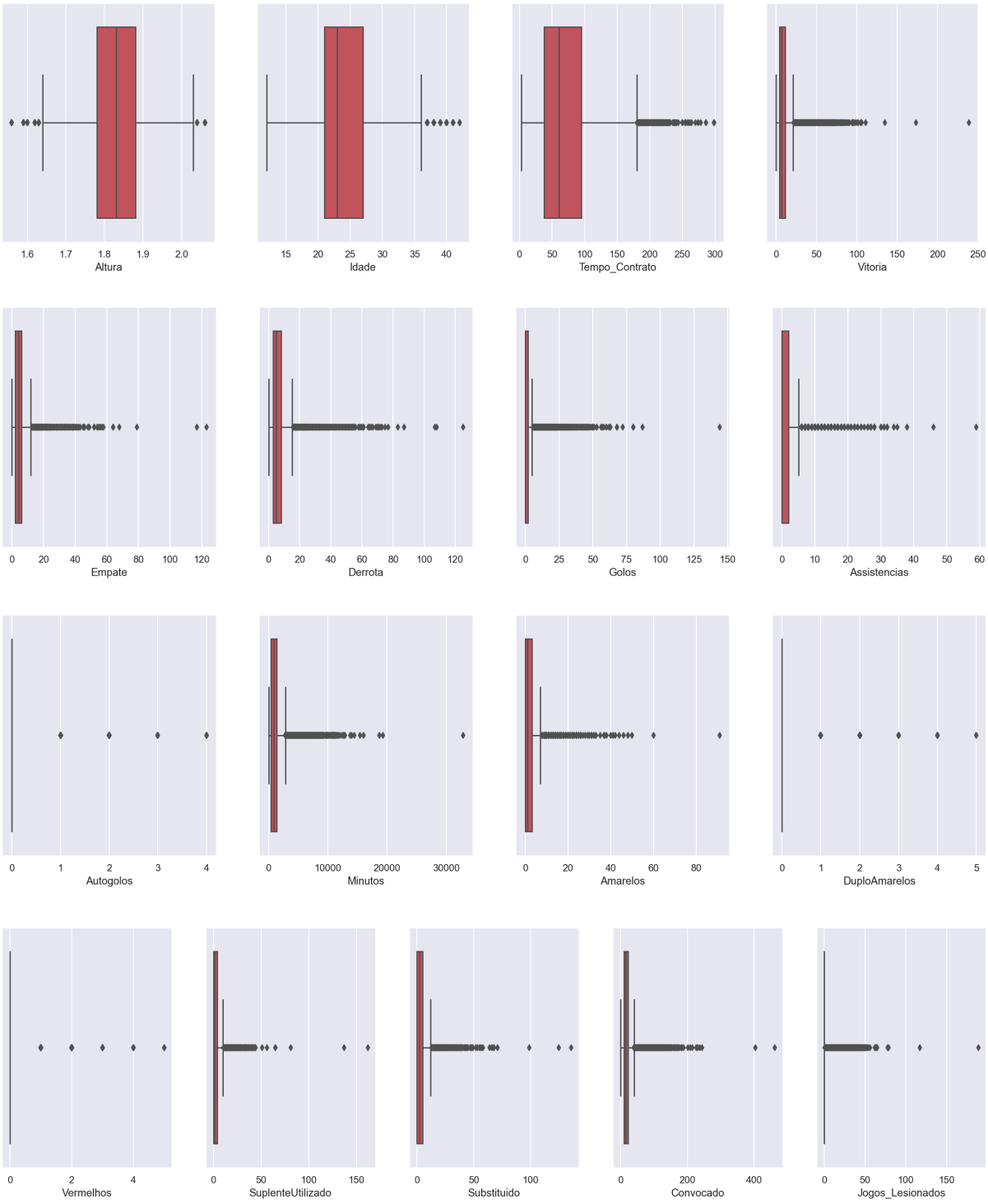

Figure 5 - Numeric Variables' Box Plots

For checking some correlation between columns, it was used Pearson correlation matrix (Figure 6). From that matrix is possible to notice that the column *call out* is correlated with a lot of other columns, such as wins, minutes and draws. It's possible to notice also that the column minutes have a lot of correlation as well. Another thing that this matrix shows, and expected as well, is that the market value has some correlation with the difference between updates.

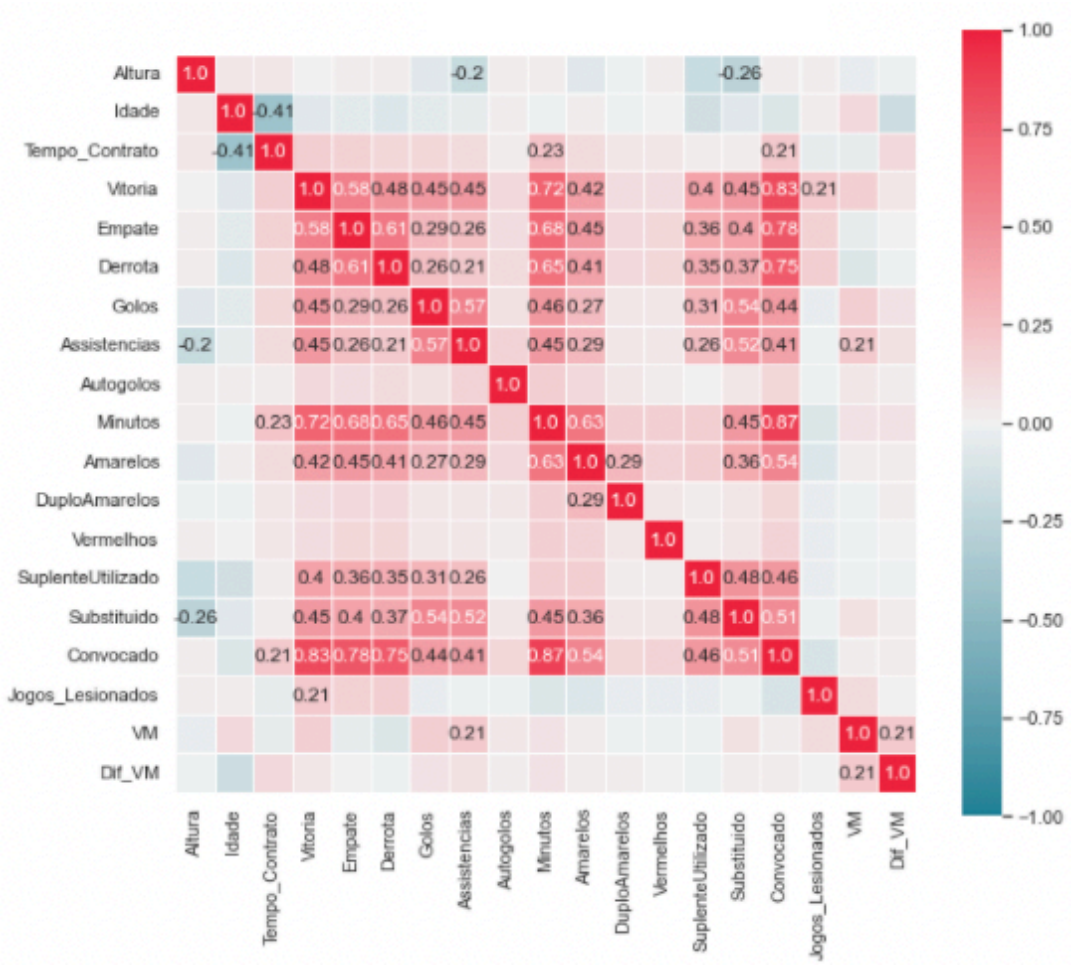

Figure 6 - Correlation Matrix

Now for the categorical variables it was done 4 charts in order to see the frequencies of the low cardinality features (Figure 7), such as, the national team, preferred foot, outfitter and position. Here it's possible to conclude that on the outfitter there is 3 main ones and rest have low cardinality, and the same for the national team, there are the A, under-21 and the under-20 team with the most cardinality. After that, and to do some analysis on all categorical variables it was done an Independence test of Chi-Squared, with an alpha of 0.0001, and the only feature that is not important to the predictor is the third most played competition.

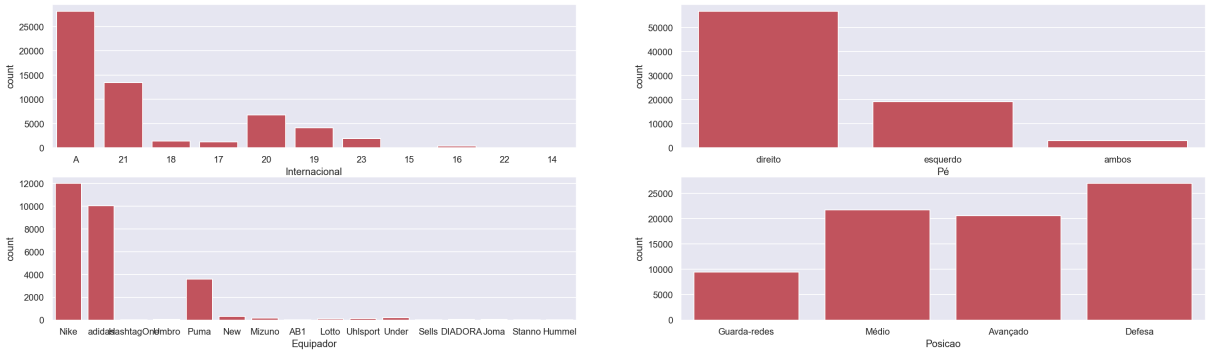

Figure 7 - Categorical Variables' Absolute Frequencies

After all this analysis it was decided to do some feature selection, based on the previously obtained results. Own-goals, double yellow and red cards, and the third most played competition have low

variability, so those features were dropped. Calls up and minutes have a lot of relation with other features and the second nationality has high cardinality and that doesn't support that many on the future model, so those were also dropped. And finally, as shows the graph on Figure 7, the outfitter has a lot of categories that have a low cardinality, and in addition to that there are also a lot of null or unknown values. Given that, the outfitter column was also dropped. So basically the features that are on the model are, player id, his high national team, his heigh, age nationality, preferred foot, time left on the contract, agent, club, position, number of wins, draws, defeats, goals, assists and yellows, the number of games that the player was sub in and sub out, his market value, and difference, the top 2 most played competitions and if occurs the value of the transfer on that period of time. The two dates that determine the time interval were also dropped because those dates will not be needed anymore.

Using the boxplots and the histograms it was done an outlier removal, so at the columns sub out, sub in, draws, goals and yellow cards I dropped the rows that were greater than 10, for the assists greater than 5, wins greater than 20, defeats greater then 15 and time left on the contract 200 months more were dropped as well.

For the categorical variables, it was decided to do little groups, to be an easier analysis and lower the cardinality. On the national team, where appears under-15, 16,17,18,19, 20 or 22, it was grouped to youth-team. Where appears null it was given as "not international". Result will be youth-team, under-21 and 23, A team, and not international. For the agents it was replaced the null values for 'without agent' and did a group as well. If the agent appears on the database less then 300 times, then this agent will group to the 'others' group. The same method was considered on the top 2 most played competitions. If the most played competition appears less then 200 times or the second most played competition appears less then 100 times, then those competitions will appear as "others".

After all this cleaning process it was done the visuals again, in order to see the changes after the data cleaning phase. For the histograms (Figure 8) it's possible to see, especially on the columns that the outliers were remove, a more distributed data, confirmed by the boxplots (Figure 9), that doesn't have outliers as much as before. For the categorical variables, as the data were grouped the low cardinality groups, seems to be a more distributed (Figure 10).

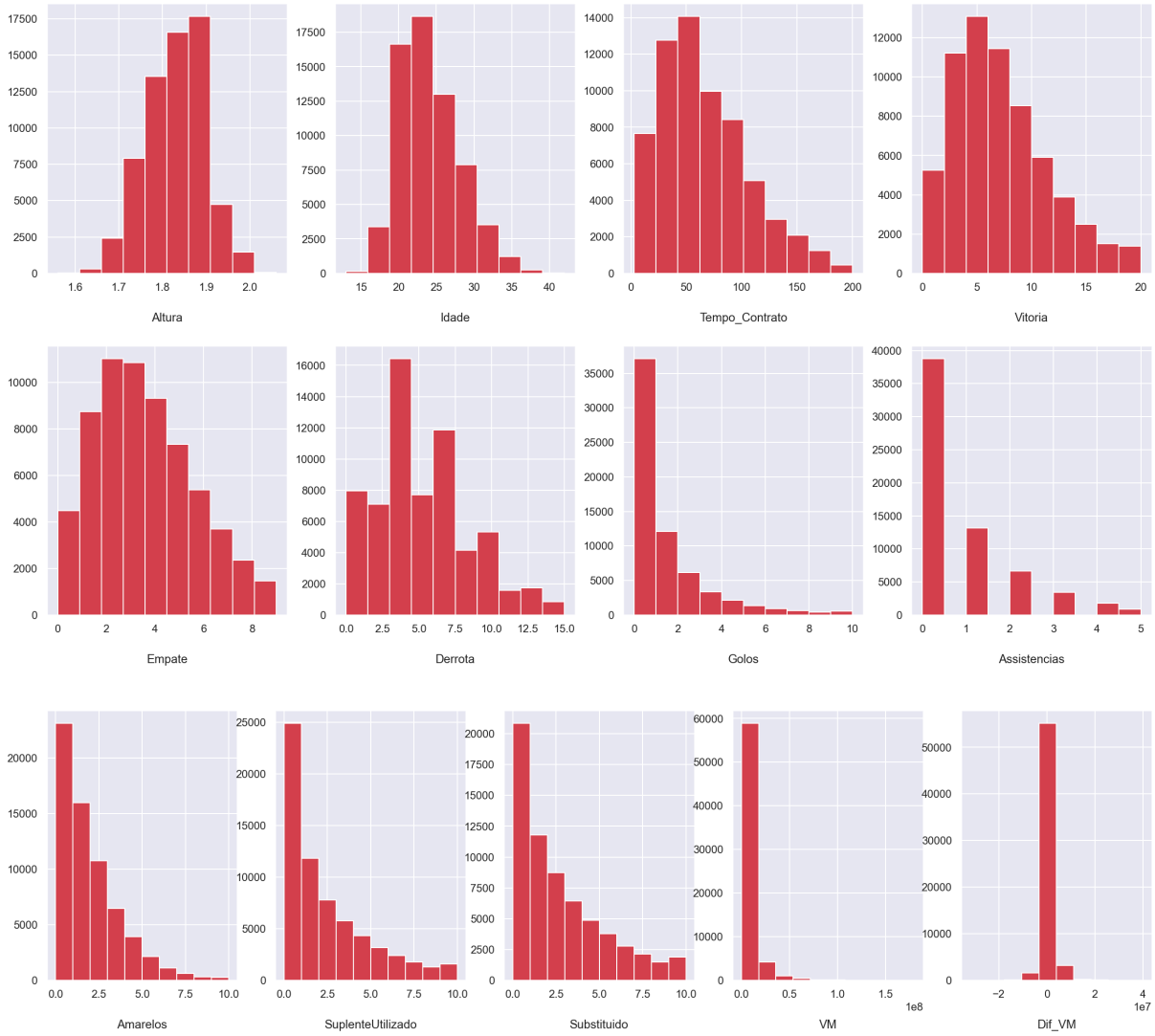

Figure 8 - Numeric Variables' Histograms, after cleaning phase

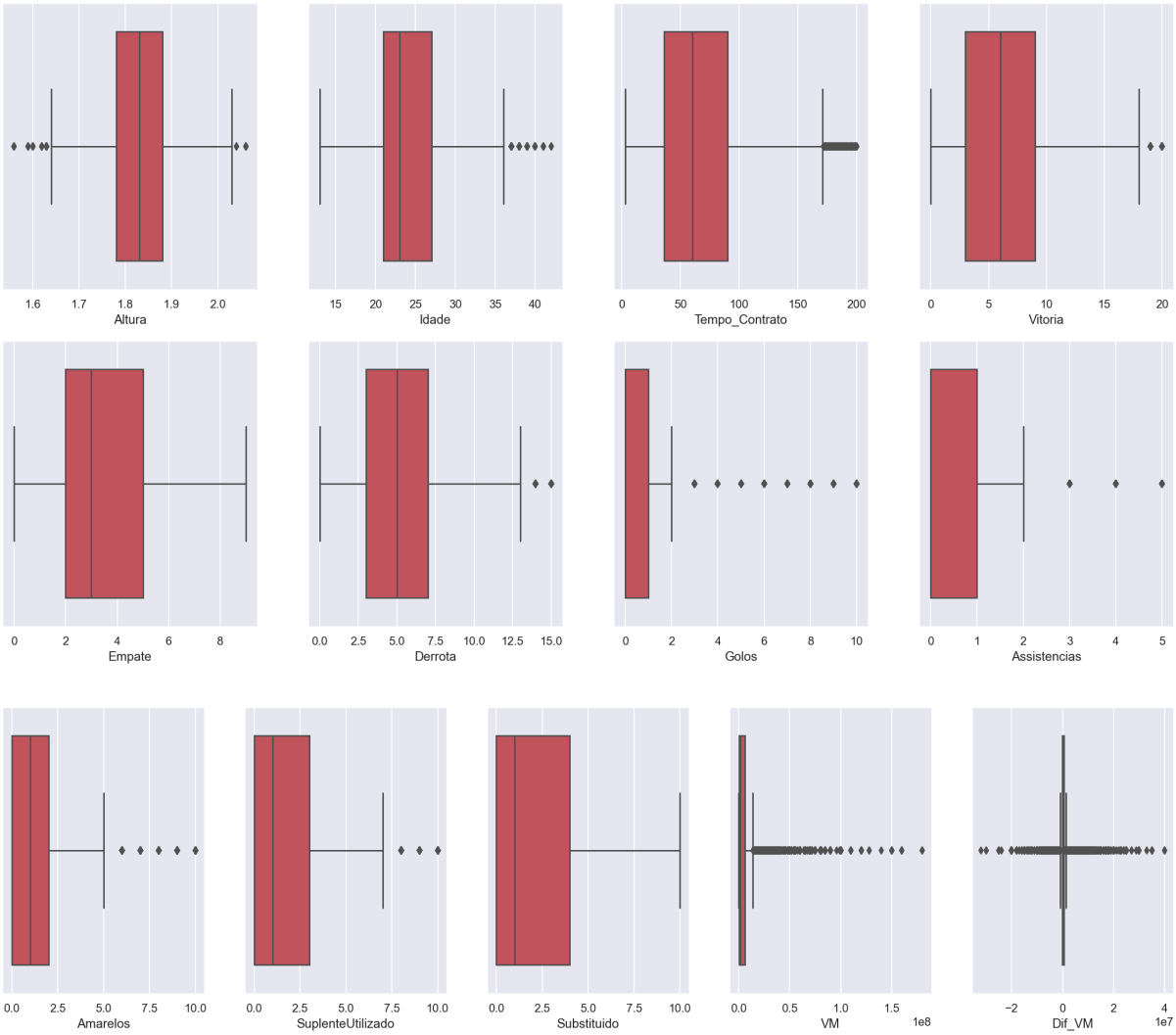

Figure 9 - Numeric Variables' Boxplots, after cleaning phase

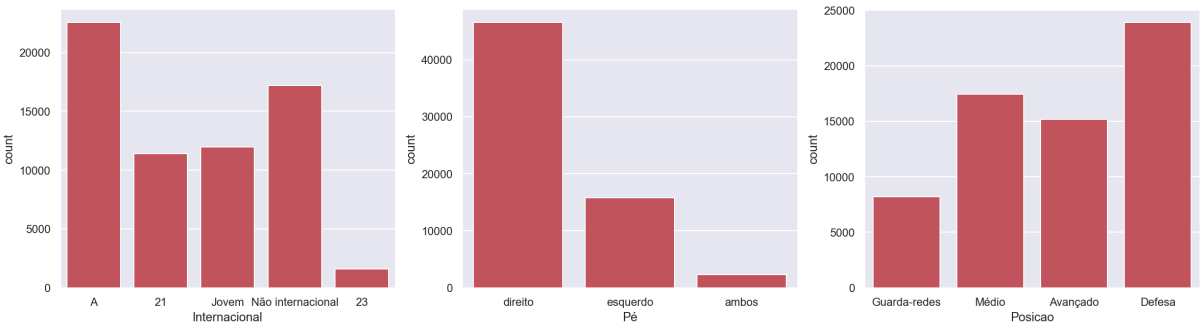

Figure 10 - Categorical Variables' Absolute Frequencies

In order to be able to do some analysis, it was done a One Hot Encoding on the categorical features. This method has a limitation because, this database has some feature with high cardinality, especially the 'Club' column. As this column has a lot of different cases, this one hot encoding will create a column for every single case. So, after the one hot encoding the database will have more then 2000 columns.

After all this steps, there is eight datasets ready for the model phase. Four for the global data, each one with a different scaler and four more the Benfica Universe.

4.2. MODELING PHASE

This modelling phase will be done in the same way through the three models, Decision Trees, Random Forest and XGBoost.

The first step is to find the best scaler. To do that, it was done four models with the default parameters, in order to get four r-squared value. The one that has a higher r-squared value it's the scaler that will be used over this phase.

After that is done a manual tuning of the parameters of the models. It was decided to do a manual one, in order to get the visuals and to be less time-consuming. However, if it was done automatically, for example Grid Search, every possibility would be tested.

On this part is done thirty different models, using Repeated Stratified K-Fold with ten splits repeated three times, for each possibility that is going to be tested on a specific parameter. It was used Repeated Stratified K-Fold because in a regression problem it helps to reduce bias, where this process ensures that each fold has a representative distribution of the objective variable. Other benefit of using this structure is that Repeated Stratified K-Fold increases the robustness by doing repetitions multiple times with different randomizations, which will lead to average out any variation caused by the random selection of the training and validation test. Then is calculate every r-squared and is given the max value, the mean and the standard deviation for every possibility. Here is given also a boxplot of the results of the r-squared in order to have a visual for easier interpretation. After all the possibilities were tested, the one that have the higher mean of the r-squares is the one that is chosen. If the mean is equal, then is chosen the one that have the top max value.

When the tuning is done, it's time to pass to the predictions. The first step here is to do a train test split. Here it was chosen that the test dataset would have 30% of the size of all data set, and the other 70% would go to the training set. On this phase it's done one hundred times the model, using also Repeated Stratified K-Fold, but this time with ten splits and ten repeats. After doing the Repeated Stratified K-Fold the output will be the mean of the r-squared of the validation and train datasets with the respective standard deviation, and a column with the predictions. This column is calculated with the mean of the values from the different models.

This process is also done twice for each model. One for the global database and other one for the Benfica Universe.

Finally, it was done also some models in order to try to get closer to the Actual Transfer Value (ATV) relatively to the TM Market Value. To do so it was used the same tuning for the model but in terms of the scaler, it was decided to try again the 4 different scalers. The big difference here is the metric that was used to check the results and to compare the alternatives in terms of scalers. Given the fact that most of the values that were predicted were a bit far away for the Actual Transfer Value, in absolute terms, the r-squared was giving some bad results. In other words, if the distance between a prediction and an ATV was like 5 million euros, the absolute difference is very high, but in this context, that difference is not that problematic. So, to solve this problem it was decided that on this ATV models the metric that will be used is the Mean Absolute Error (MAE).

4.2.1. Decision Trees

For the manual tuning, it was decided to check what was the best value for the following parameters (sklearn.tree.DecisionTreeRegressor, n.d.):

- **Splitter** which is the strategy that the model will use to split each node. It can be “best”, where the best split is chosen and “random”, where the chosen one is the best random split.
- **Max_depth** which is, as the name said, the maximum depth of the tree. This parameter must be an integer.
- **Min_samples_split** which is the minimum number of samples required to split an internal node. These values could be integers or floats, but it was only considering integers, in order to control the minimum number of samples in each node.
- **Min_samples_leaf** is the parameter where it’s defined the minimum number of samples that a leaf node must have. There only be considered a split point at any depth if the number of leaves is at least this value. This parameter can also be an integer or a float. As the one before it only was considering integers.
- **Min_weight_fraction_leaf** is the one that has the minimum weighted fraction of the total of weights that is required to be a leaf node. If there is any node that have a lower weight than this value, that node won’t be consider. This parameter only accepts floats.
- **Max_features** is the number of features to consider when it’s searching for the best split. In this project were consider the options “auto” which means that the maximum features are the actual number of features, “sqrt”, which only consider the squared root of the number of features and “log2”, which like the last one, only consider the logarithm base 2 of the number of features.
- **Criterion** which is the function that measures the quality of a split. It was considered the options “squared_error” for the mean squared error, “friedman_mse” which uses mean squared error with Friedman’s improvement and “poisson” which uses reduction in Poisson deviance to find splits.

Given that, it was done two tuning processes, one for the “global” data another of the Benfica Universe Data. In each one of those processes, it was done two predictions. One with the objective to find the Market Value with the Values from TransferMarket, and another with the Actual Transfers Values.

4.2.1.1. Benfica Data

As mentioned before, the first step in every model, and specifically in Decision Trees, is to check what is the best scaler for the model.

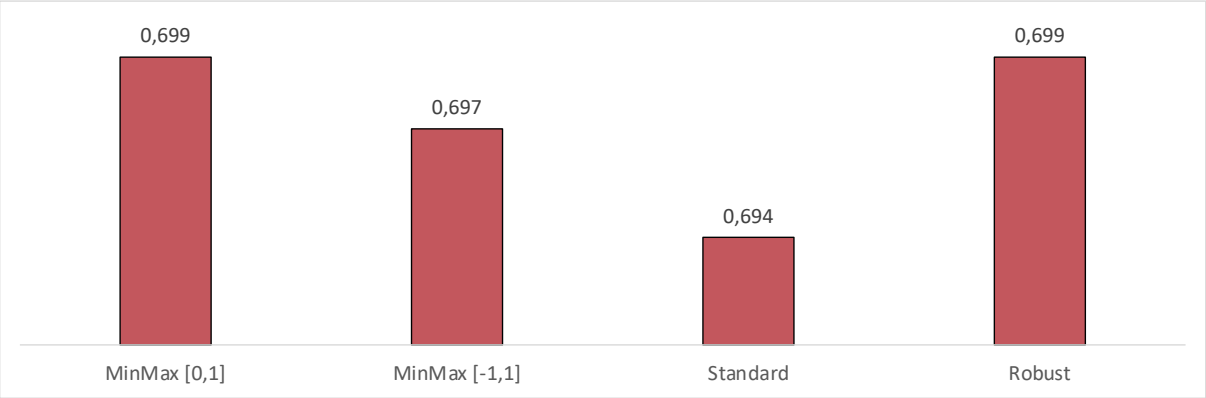

Figure 11 - R-squared values in Decision Trees with Benfica Data, to check the best scaler.

As a result of Figure 11 it is possible to notice that MinMax between 0 and 1, and Robust are with the higher r-squared, so the decision was to proceed with the MinMax dataset.

After the scaler were chosen, it's time for tuning the model (Table 1).

Table 1 - Tuning for Decision Trees Benfica Data (r-squared)

Decision Trees		Benfica
splitter	best	0,647
	random	0,775
max_depth	50	0,796
	100	0,796
	120	0,796
	150	0,796
min_samples_split	2	0,796
	10	0,76
	50	0,721
	100	0,712
min_samples_leaf	1	0,796
	5	0,761
	10	0,636
	50	0,267
min_weight_fraction_leaf	1,00E-06	0,796
	1,00E-05	0,796
	0,0001	0,796
max_features	auto	0,796
	sqrt	0,558
	log2	0,556
criterion	squared_error	0,796
	friedman_mse	0,794
	poisson	0,761

With the information on the table above, for the Benfica Universe the values that were chosen for this Decision Trees model are the ones that appears in read in Table 1.

In this model the r-squared for the train data set was **1.0** and for the validation set was **0.611**. The other results that appear in this phase is the most important features for this model that are in Table 2.

Table 2 - Top 10 Features for the Decision Trees Model with TM values and Benfica Data

Top	Feature
1	Premier League Player
2	Brazilian Player
3	Bundesliga Player
4	Time Left on the Contract
5	Serie A Player
6	Eredivisie Player
7	International for the First National Team
8	Portuguese Player
9	Height
10	Age

As mentioned before, it was done again some models in order to analyze the ATV. The tuning will be the same, but the scaler could be different.

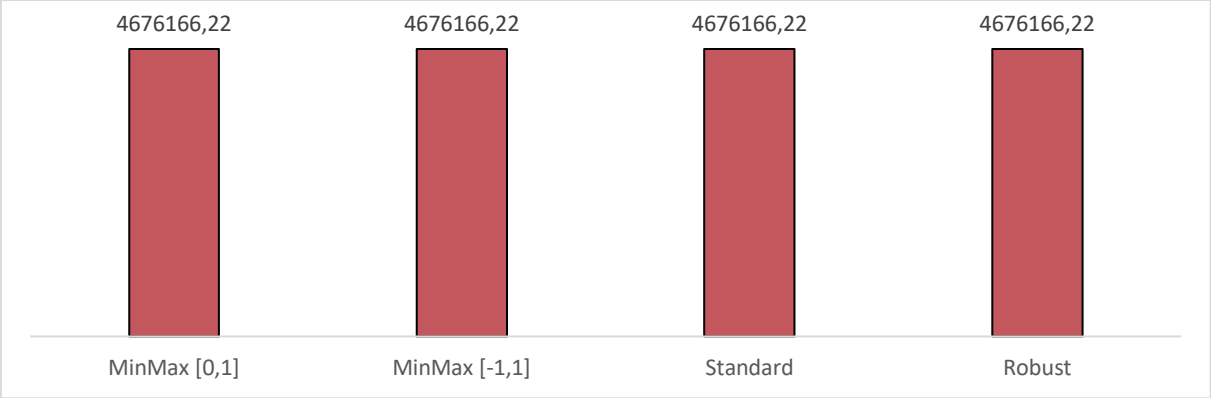

Figure 12 - MAE values in Decision Trees, to check the best scaler.

As Figure 12 shows the MAE is the same for the four scalers and with that information it was decided to keep the Min Max between -1 and 1.

In the table below, table 3 it can be seen the top 10 features to this model.

Table 3 - Top 10 Features for the Decision Trees Model, with ATV and Benfica Data

Top	Feature
1	Premier League Player
2	One Competition Player
3	Serie A Player
4	Uruguayan Player
5	Netherlands National Cup Player
6	International for the First National Team
7	Argentinian Player
8	International for the U-23 National Team
9	French Player
10	German Player

As shown on the figure 11 the MAE of this model is **4676166,22** million euros.

4.2.1.2. Global Data

For this Global Dataset the Scaler that was chosen was the Min Max between -1 and 1, as the Figure 13 shows.

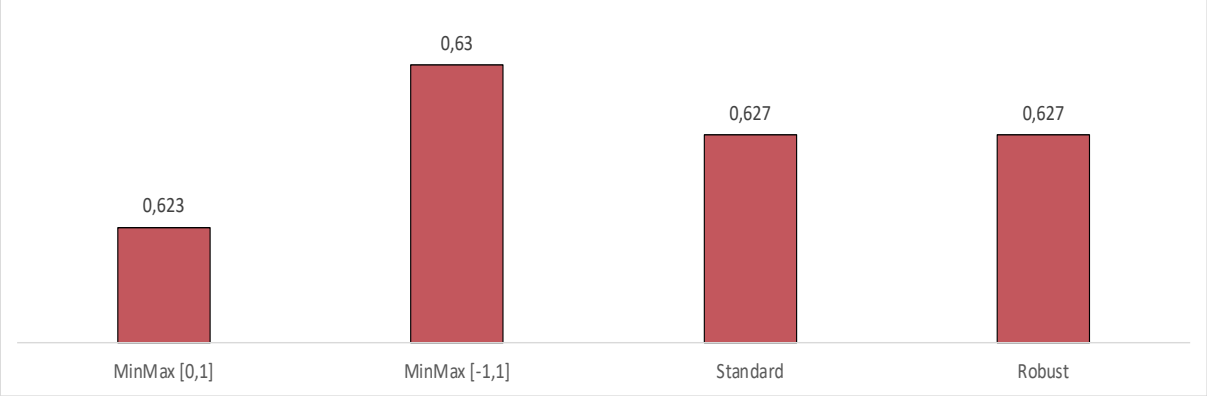

Figure 13 - R-squared values in Decision Trees with Global Data, to check the best scaler on TM model.

In terms of tuning the model the table 4 has the results with the different options and with the r-squared of those options.

Table 4 - Tuning for Decision Trees Global Data (r-squared)

Decision Trees		All
splitter	best	0,627
	random	0,676
max_depth	50	0,672
	100	0,675
	120	0,675
	150	0,672
min_samples_split	2	0,675
	10	0,704
	50	0,657
	100	0,622
min_samples_leaf	1	0,704
	5	0,664
	10	0,631
	50	0,521
min_weight_fraction_leaf	1,00E-06	0,704
	1,00E-05	0,704
	0,0001	0,66
max_features	auto	0,704
	sqrt	0,461
	log2	0,334
criterion	squared_error	0,626
	friedman_mse	0,627
	poisson	0,49

This model ends with a r-squared of **0.91** for the train data set and **0.698** for the validation set.

Below there is table 5 with the top 10 features of this model.

Table 5 - Top 10 Features for the Decision Trees Model, with TM value and Global Data

Top	Feature
1	International for the First National Team
2	Premier League Player
3	Age
4	LaLiga Player
5	Time Left on the Contract
6	Bayern Player
7	Serie A Player
8	Player managed by Family Member
9	Barcelona Player
10	Bundesliga Player

In terms of ATV, and similar to the Benfica Data, here in Global Data, the MAE is also equal for the four scalers, as shows Figure 14, and it was decided to keep using the Min Max between -1 and 1 as a scaler.

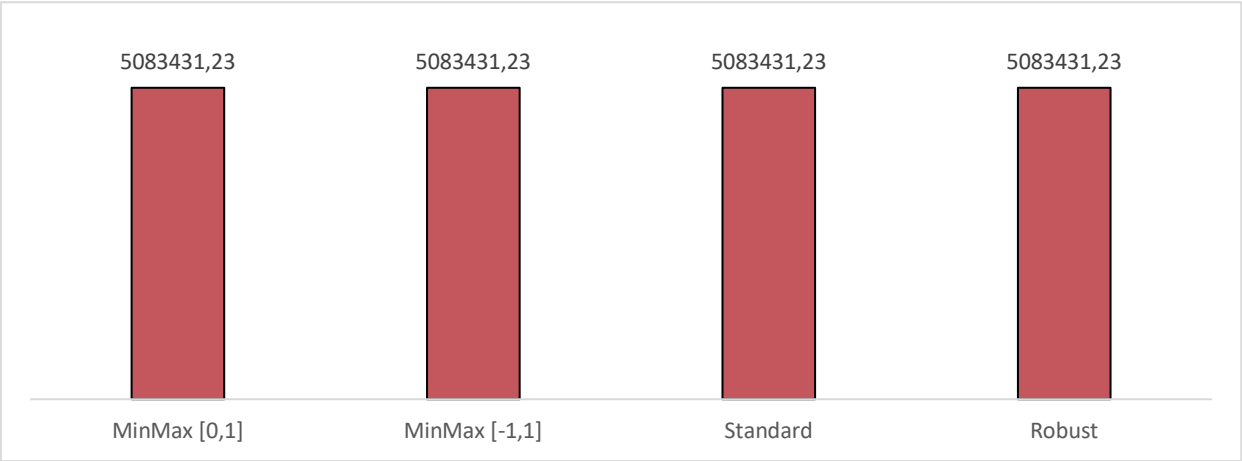

Figure 14 - R-squared values in Decision Trees with Global Data, to check the best scaler on ATV model

In the same way that was done before, on table 6 is the top 10 features with Global Data in terms of ATV.

Table 6 - Top 10 Features for the Decision Trees Model, with ATV value and Global Data

Top	Feature
1	Premier League Player
2	International for the First National Team
3	Player that uses both feet
4	Player that as second most played competition has Premier League
5	Brazilian Player
6	Time Left on the Contract
7	Age
8	Argentinian Player
9	French Player
10	Bundesliga Player

As shown in figure 14 the MAE for this model is **5083431,23** million euros.

4.2.2. Random Forest

The hyper-parameters that were subject of analysis were (Sklearn.Ensemble.RandomForestRegressor ,2023):

- **Bootstrap** samples are used when building tress. If False, the whole dataset is used to build each tree.
- **N_estimators** is the number of trees in the forest.
- **Min_samples_split** is the minimum of samples required to split an internal node. If the value is an integer, then that value is the minimum samples required. If it is a float, then the value is the percentage of the number of samples.
- **Min_samples_leaf** is the minimum number of samples required to be at a leaf node. If the value is an integer, then that value is the minimum samples required. If it is a float, then the value is the percentage of the number of samples.
- **Min_weight_fraction_leaf** is the minimum weighted fraction of the total weights required to be at a leaf node. This value is always a float.
- **Max_features** is the number of features to consider when looking for the best split. This value can be "auto" if the max_features is equal to the total number of features, can be "sqrt" if the max_features is equal to the squared-root of total number of features, can be "log2", if the max_features is equal to the logarithm of base 2 of the total number of features, can be a float and then the max_features will be the percentage of the number of features or, finally, can be an integer and the max_features would be that value
- **Max_depth** is the maximum depth of the tree. This value can be None then the nodes are expanded until all leaves are pure, or can be an integer and the maximum depth would be that number
- **N_jobs** is the number of jobs that can run in parallel, such as, fit, predict, apply, among others.
- **Max_leaf_nodes** is the maximum leaves that a tree could have. It can be an integer or if it's Null, then the number of leaf nodes are unlimited.
- **Min_impurity_decrease** is a number that means that a node will be split if this split gets a decrease of the impurity greater or equal to this value.
- **Warm_start** when set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, on the other hand, if it's set to False, just fit a whole new forest.
- **Max_samples** is the number of samples from the dataset to train each base estimator. If None, then all the samples will be drawn, if it is an integer, then max_samples will be that value, and finally, if it is a float the number of samples will be max_samples $\times$ the number of samples in the dataset.
- **Criterion** is the function that measures the quality of a split. It can be "Squared_error" for the mean squared error, "Friedman_mse" which uses the mean squared error with Friedman's improvement score for potential splits, "Absolute_error" for the mean absolute error, and finally "Poisson" which uses reduction in Poisson deviance to find splits. Given the fact that using "Absolute_error" is significantly slower than using "Squared_error", the mean absolute error option wasn't considered.
- **Ccp_alpha** is the complexity parameter used for Minimal Cost-Complexity Pruning. The subtree with the largest cost complexity that is smaller than ccp_alpha will be chosen.

4.2.2.1. Benfica Data

As was done in a previous model, the first step is to check what is the scaler that adjust better for this model.

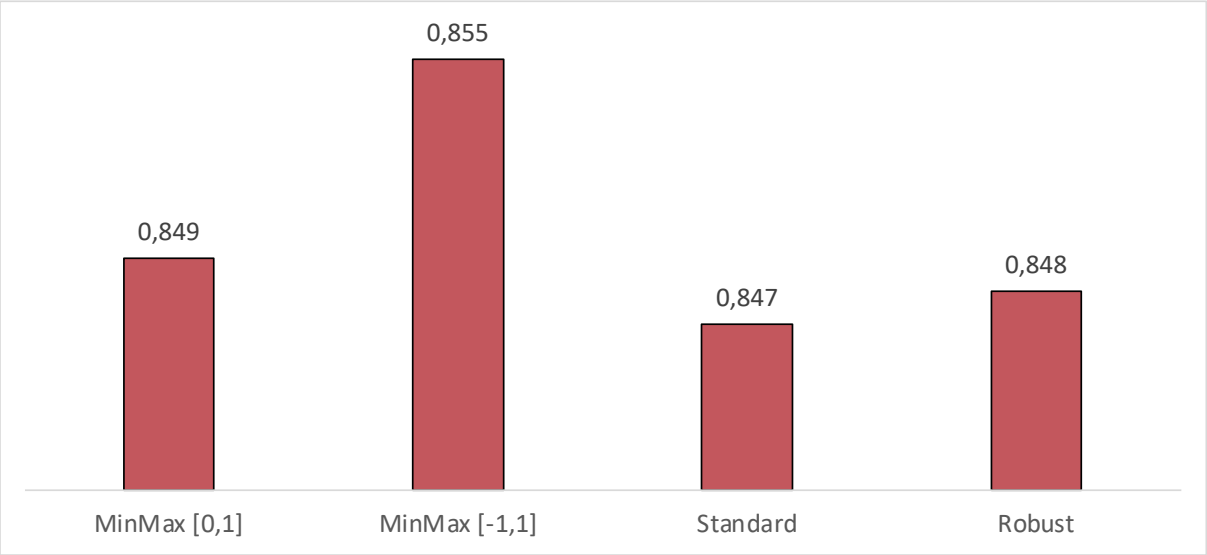

Figure 15 - R-squared values in Random Forest with Benfica Data, to check the best scaler

As can be seen in Figure 15 the scaler that will be used in this model is the MinMax between -1 and 1.

For the tuning process in the table 7 there is the parameters that were selected and final options for the values for each parameter.

Table 7 - Tuning for Random Forest Benfica Data (r-squared)

Random Forest		Benfica	Random Forest		Benfica
bootstrap	TRUE	0,87	n_jobs	2	0,873
	FALSE	0,838		10	0,873
n_estimators	150	0,872		50	0,873
	200	0,872	max_leaf_nodes	None	0,873
	250	0,873		2500	0,873
min_samples_split	2	0,873	min_impurity_decrease	0,2	0,873
	5	0,862		0,5	0,873
	10	0,838		0,7	0,873
min_samples_leaf	1	0,873	warm_start	TRUE	0,873
	10	0,559		FALSE	0,873
	50	0,211	max_samples	50	0,631
min_weight_fraction_leaf	0	0,873		100	0,779
	0,2	0,257		0,2	0,681
	0,4	0,183		0,5	0,835
	0,6	nan		None	0,873
max_features	auto	0,873	criterion	squared_error	0,873
	1	0,775		friedman_mse	0,873
	sqrt	0,793		poisson	0,873
	log2	0,778	ccp_alpha	0	0,873
max_depth	10	0,867		0,005	0,873
	50	0,873		0,0005	0,873
	150	0,873		0,05	0,873

This model ends with a r-squared of **0.921** for the train data set and **0.919** for the validation set.

Bellow, on table 8 are the top 10 more important features for this model.

Table 8 - Top 10 Features for the Random Forest Model with TM values and Benfica Data

Top	Feature
1	Age
2	Premier League Player
3	Height
4	Bundesliga Player
5	Time Left on the Contract
6	International for the First National Team
7	Player that as second most played competition has Netherlands National Cup
8	Serie A Player
9	Yellow Cards
10	Eredivisie Player

For the ATV the Figure 16 shows the MAE for each scaler.

Figure 16 - MAE values in Random Forest, to check the best scaler.

According to Figure 16, on this model the scaler that was chosen was Min Max with values between 0 and 1.

On table 9 can be seen the top 10 most important features for this model.

Table 9 - Top 10 Features for the Random Forest Model, with ATV and Benfica Data

Top	Feature
1	Premier League Player
2	Assists
3	Serie A Player
4	Age
5	Time Left on the Contract
6	Uruguayan Player
7	Argentinian Player
8	Draws
9	One Competition Player
10	Portuguese Player

As shown in Figure 16 the MAE for this model is **3818086,11** million euros

4.2.2.2. Global Data

In the Figure 17 can be seen the r-squared values between the 4 different scalers that were consider.

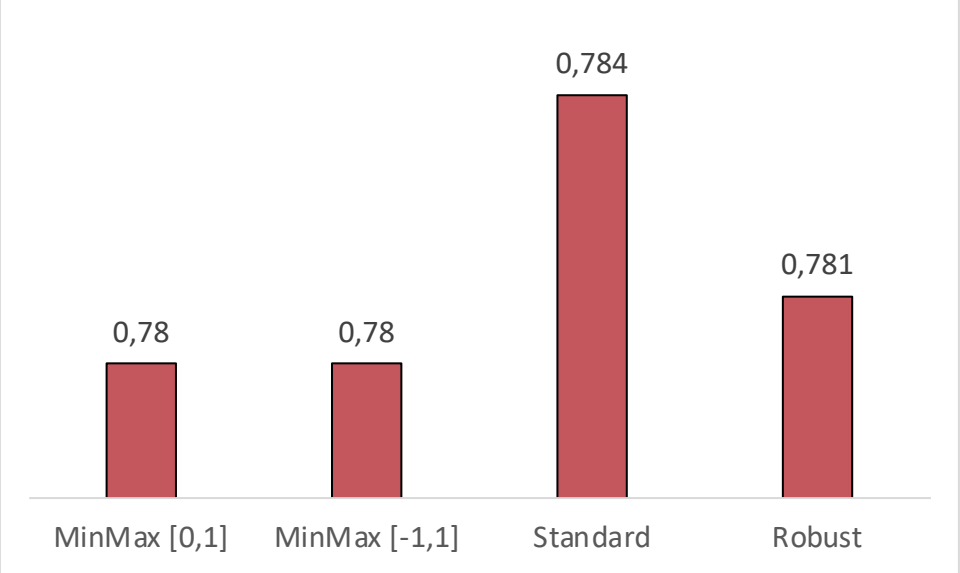

Figure 17 - R-squared values in Random Forest with Global Data, to check the best scaler on TM model.

As shows the Figure 17, the scaler that will be used is Standard.

For the tuning process, the table 10 shows the parameters and the values that were chosen for this model.

Table 10 - Tuning for Random Forest Global Data (r-squared).

Random Forest		All	Random Forest		All
bootstrap	TRUE	0,783	n_jobs	2	0,784
	FALSE	0,65		10	0,784
n_estimators	50	0,78		50	0,784
	100	0,783	max_leaf_nodes	None	0,784
	150	0,784		2500	0,763
min_samples_split	2	0,784	min_impurity_decrease	0	0,784
	5	0,775		0,2	0,784
min_samples_leaf	1	0,784		0,5	0,784
	10	0,66		0,7	0,784
	50	0,516	warm_start	TRUE	0,784
	100	0,466		FALSE	0,784
min_weight_fraction_leaf	0	0,784	max_samples	50	0,267
	0,2	0,138		100	0,322
	0,4	0,028		0,2	0,661
	0,6	nan		0,5	0,728
max_features	auto	0,784		None	0,784
	1	0,005	criterion	squared_error	0,784
	sqrt	0,546		friedman_mse	0,784
	log2	0,322		poisson	0,588
max_depth	None	0,784	ccp_alpha	0	0,784
	10	0,631		0,005	0,782
	50	0,768		0,0005	0,782
	150	0,768		0,05	0,782

This model ends with a r-squared of **0.953** for the train data set and **0.953** for the validation set.

On table 11 can be seen the top 10 most important features for this model.

Table 11 - Top 10 Features for the Random Forest Model, with TM value and Global Data

Top	Feature
1	International for the First National Team
2	Premier League Player
3	Time Left on the Contract
4	Age
5	Height
6	LaLiga Player
7	Goals
8	Wins
9	Serie A Player
10	Bundesliga Player

In terms of ATV, the Figure 18 shows the MAE among the 4 different scalers.

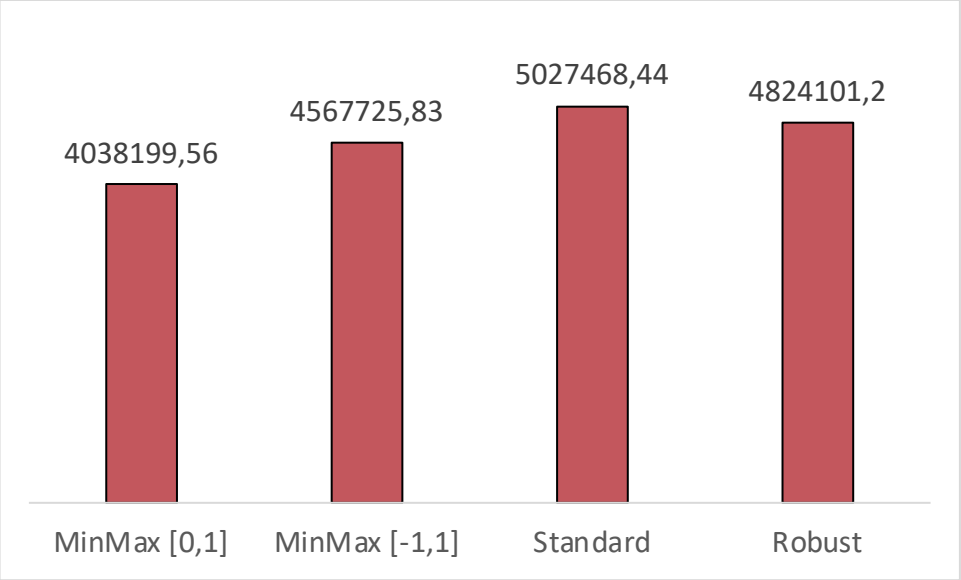

Figure 18 - MAE values in Random Forest with Global Data, to check the best scaler on ATV model.

The scaler that was chosen to proceed with the analysis in this model and with Global Data was Min Max between 0 and 1.

Table 12 have the top 10 features for this model with this dataset.

Table 12 - Top 10 Features for the Random Forest Model, with ATV value and Global Data

Top	Feature
1	Premier League Player
2	International for the First National Team
3	Age
4	Time Left on the Contract
5	Defeats
6	Wins
7	Height
8	Assists
9	Benfica Player
10	Sub out

As can be seen in Figure 18, the MAE result for this model is **4038199,56** million euros.

4.2.3. XGBoost

The parameters that were chosen to go through the tuning process are (Rithp, 2023):

- **N_estimators** is the number of trees in the model. Normally if this value is high then, the model performance improves, however it can also lead to overfitting.
- **Max_depth** is the parameter that controls the maximum depth of the trees in the model. A larger value on this parameter results in more complex models, which can lead to overfitting. In other hand, a smaller value results in simpler models, but also can lead to underfitting.
- **Eta (learning rate)** controls the step size at which the optimizer makes updates to the weights. A smaller learning rate value leads to a slower but more accurate updates, while a higher value results in faster but less accurate updates.
- **Subsample** is the parameter that controls the fraction of observations used for each tree. A lower value results in smaller and less complex models, which has as benefit the prevents overfitting. But a higher value in this parameter leads to a larger and more complex models, which can lead to overfitting.
- **Colsample_bytree** controls the fraction of features used for each tree. A smaller value results in smaller and less complex models, which will help prevent overfitting. A larger value in this parameter leads to a larger and more complex models that can result in overfitting.

4.2.3.1. Benfica Data

Finally, the last model the process will start the same with the step when is checked what is the scaler that adjust better for this model.

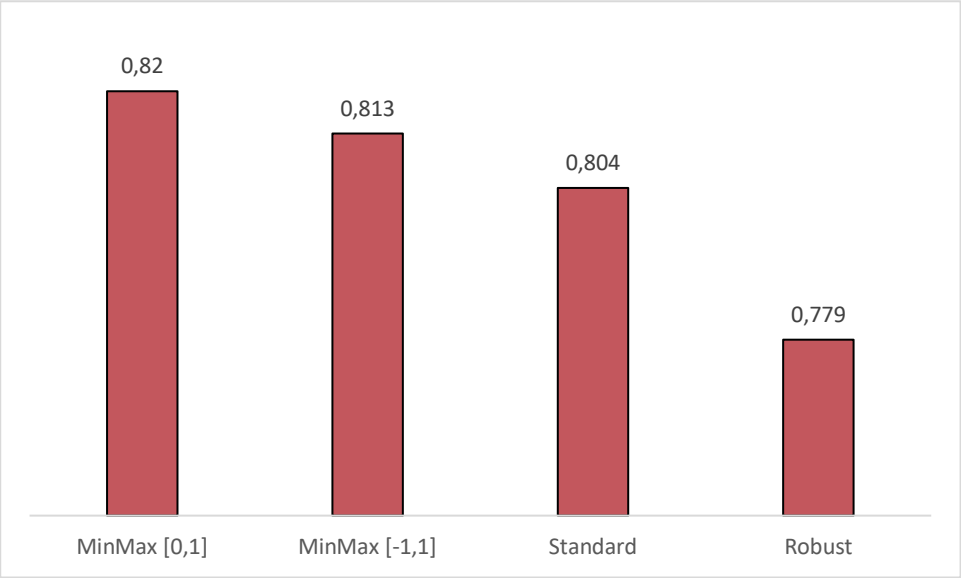

Figure 19 - R-squared values in XGBoost with Benfica Data, to check the best scaler.

As can be seen in Figure 19 the scaler that will be used in this model is the MinMax between 0 and 1.

For the tuning process in the table 13 there is the parameters that were selected and final options for the values for each parameter.

Table 13 - Tuning for XGBoost Benfica Data (r-squared)

XGBoost		Benfica
n_estimators	100	0,797
	200	0,797
	300	0,797
	400	0,797
max_depth	1	0,691
	3	0,788
	4	0,816
	6	0,797
	8	0,775
eta	0,3	0,816
	0,1	0,825
	0,01	0,801
	0,001	0,168
subsample	0,5	0,801
	0,8	0,827
colsample_bytree	1	0,825
	0,2	0,824
	0,5	0,84
	0,8	0,837

This model ends with a r-squared of **0.999** for the train data set and **0.844** for the validation set.

Bellow, on table 14 are the top 10 more important features for this model.

Table 14 - Top 10 Features for the XGBoost Model with TM values and Benfica Data

Top	Feature
1	Premier League Player
2	Not International
3	Eredivisie Player
4	Player that as second most played competition has England National Cup
5	Brazilian Player
6	Bundesliga Player
7	Player that as second most played competition has Netherlands National Cup
8	International for the First National Team
9	Goalkeeper
10	Argentinian Player

For the ATV the Figure 20 shows the MAE for each scaler.

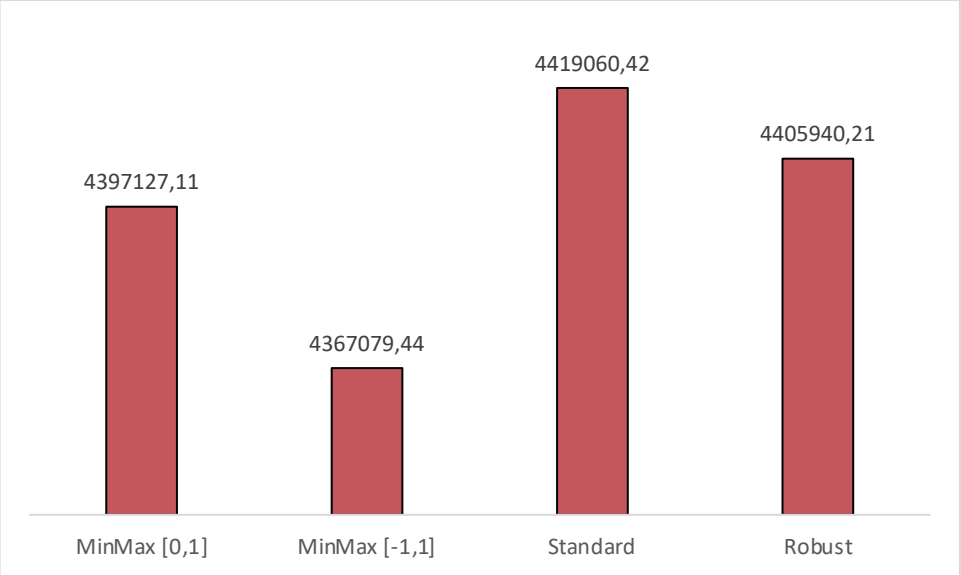

Figure 20 - MAE values in XGBoost, to check the best scaler.

On this model the scaler that was chosen was Min Max with values between -1 and 1.

On table 15 can be seen the top 10 most important features for this model.

Table 15 - Top 10 Features for the XGBoost Model, with ATV and Benfica Data

Top	Feature
1	Premier League Player
2	Serie A Player
3	International for the Youth National Teams
4	International for the First National Team
5	Portuguese Player
6	Argentinian Player
7	Age
8	Yellow Cards
9	Assists
10	Uruguayan Player

As shown in figure 20 the MAE for this model is **4367079,44** million euros

4.2.3.2. Global Data

In the Figure 21 can be seen the r-squared values between the 4 different scalers that were consider.

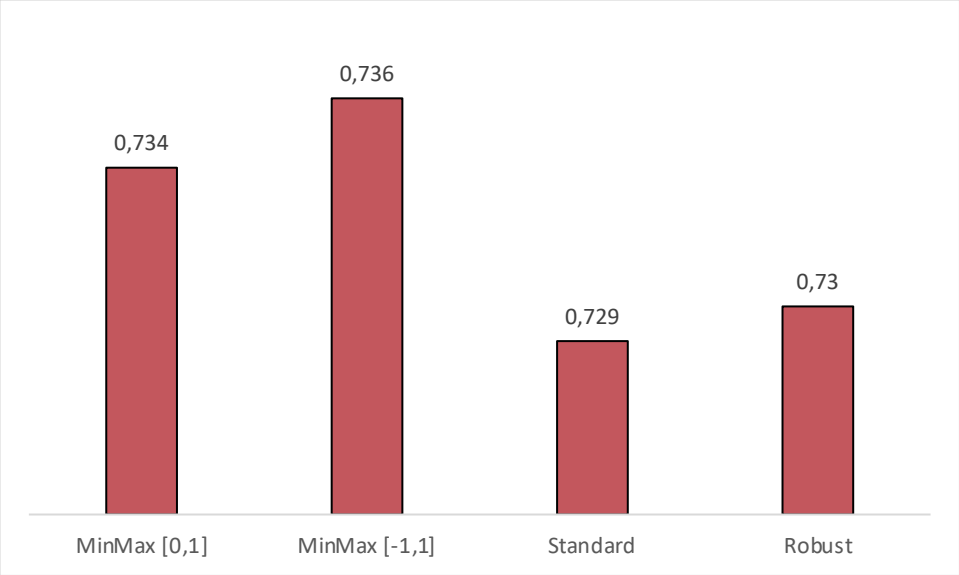

Figure 21 - R-squared values in XGBoost with Global Data, to check the best scaler on TM model.

As shows the figure 21, the scaler that will be used is Min Max between -1 and 1

For the tuning process, the table 16 shows the parameters and the values that were chosen for this model.

Table 16 - Tuning for XGBoost Global Data (r-squared).

XGBoost		Benfica
n_estimators	1000	0,8
	1500	0,804
	2000	0,805
max_depth	1	0,527
	4	0,779
	6	0,8
	8	0,796
eta	0,3	0,8
	0,1	0,787
	0,01	0,683
	0,001	0,351
subsample	0,5	0,768
	0,8	0,8
colsample_bytree	1	0,8
	0,2	0,776
	0,5	0,794
	0,8	0,801

This model ends with a r-squared of **0.967** for the train data set and **0.819** for the validation set.

On table 17 can be seen the top 10 most important features for this model.

Table 17 - Top 10 Features for the XGBoost Model, with TM value and Global Data

Top	Feature
1	Premier League Player
2	Real Madrid Player
3	International for the First National Team
4	Man. City Player
5	Atlético Madrid Player
6	Barcelona Player
7	Bayern Player
8	PSG Player
9	Egyption Player
10	Juventus Player

In terms of ATV, the figure 22 shows the MAE among the 4 different scalers.

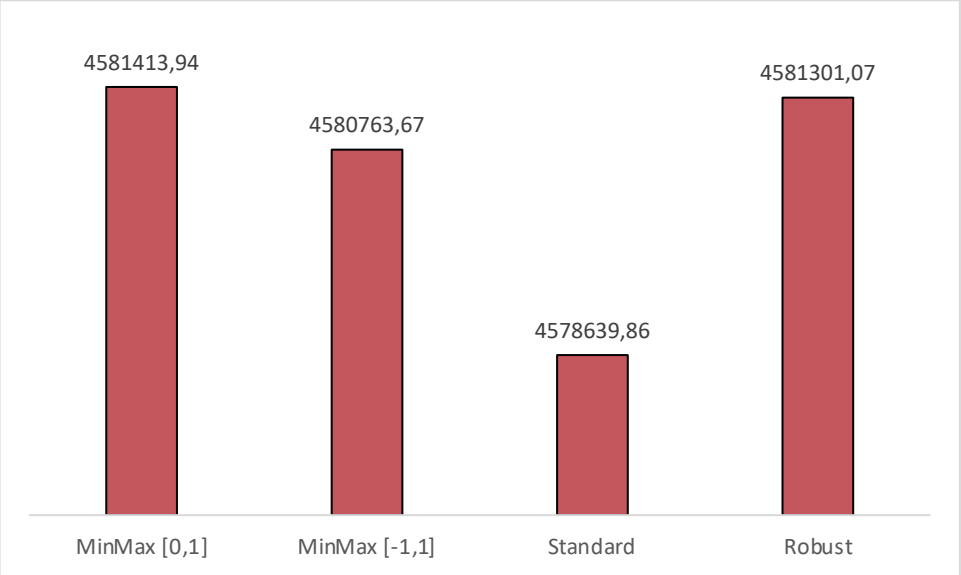

Figure 22 - MAE values in XGBoost with Global Data, to check the best scaler on ATV model.

The scaler that was chosen to proceed with the analysis in this model and with Global Data was Standard.

Table 18 has the top 10 features for this model with this dataset.

Table 18 - Top 10 Features for the Random Forest Model, with ATV value and Global Data

Top	Feature
1	Premier League Player
2	Player that as second most played competition has Premier League
3	Benfica Player
4	Dortmund Player
5	Player managed by Gestifute
6	Leicester Player
7	Shakhtar D. Player
8	International for the First National Team
9	Central African Republic Player
10	LaLiga Player

As can be seen in Figure 22, the MAE result for this model is **4578639,86** million euros.

5. CONCLUSION

This work has the objective to try to get a market value closer to the value of players' transfer and to understand which factors are more relevant in that prediction.

In terms of the players that are transfer to Benfica, the top features that this club has as priority are players that played on **Premier League**, **Eredivisie** and players that are **international for the first national team**. In terms of the model that adjust better to this dataset is the **Random Forest** with a r-squared of **0,919** in the test set on the TM problem and **3818086,11** of MAE on the Actual Transfer Problem.

Referencing to the more global dataset, the factors that are more important for the calculation of the market value are players that played on **Premier League**, **international for the first national team**, the **time left on the contract** and the player's **age**. For the model that perform better results, it was **Random Forest** with a r-squared of **0,953** for the TM values and **4038199,56** of MAE on the ATV problem.

On Figure 23 it is possible to see an example of the final report that was deliver at Benfica. It has some filters that the user can choose, for example, the position, the club that the player represents now, among other. At the *Predictions* column, if the cell is red that means that the prediction is 75% or move above of the actual MV, and if it is blue, it means that is 75% below that value.

Nome	Data inicio	Data fim	VM	Predictions	Transfer Value	Predictions TV
Aaron Cresswell	14/09/11	07-02-12	0.20M	0.55M		4.85M
Aaron Cresswell	07/02/12	22-06-12	0.50M	1.14M		6.14M
Aaron Cresswell	02/03/13	12-07-13	0.75M	1.25M		5.78M
Aaron Cresswell	08/01/14	12-08-14	1.50M	2.02M	4.75M	
Aaron Cresswell	12/08/14	12-02-15	4.00M	6.11M		18.22M
Aaron Cresswell	12/02/15	01-07-15	7.00M	6.84M		18.05M
Aaron Cresswell	09/02/16	01-08-16	10.00M	11.80M		18.28M
Aaron Cresswell	01/08/16	20-02-17	12.00M	13.78M		13.67M
Aaron Cresswell	20/02/17	28-06-17	12.00M	13.09M		17.19M
Aaron Cresswell	28/06/17	02-01-18	12.00M	14.56M		12.99M
Aaron Cresswell	02/01/18	28-05-18	12.00M	14.10M		19.31M
Aaron Cresswell	28/05/18	19-12-18	12.00M	13.96M		9.12M

Figure 23 - Example of the Final Report

To conclude this project, the predictions about the Actual Transfer Values didn't have as good results as the TransferMarket Values, and one of the reasons of that is that only near 10% of the dataset has Actual Transfer Values which lead to a lack of information to train the model.

Another limitation is the use of the One Hot Encoding (OHE) for the Club feature. This can be an issue because in this dataset there are many different clubs and doing an OHE leads to a many different columns that will be very time consuming for the predictions.

In addition to that there are some information gaps in some features that could be more important, such as the time left on the contract. In this project it was only consider the newest contract of every player, but sometimes when the player is with 6 moths or less left in his contract his market value tends to decrease because the club doesn't want to lose the player without a monetary compensation. This aspect is not in consideration in this project.

Other way to improve this work is to get more features related to the game. Here we have goals, assists, the minutes that player plays, the number of cards, and others. If it was possible to have the tackles, key-passes, blocks, goalkeeper defenses, among others, some players in the midfield and defensive zone on the field could benefit and his predictions could be more precise, more like the forwards that already have the goals and the assists.

The Player's transfers are very important in the professional football universe, and given that, these approaches could bring a lot of benefits to the clubs, in order to get a final decision more wisely and with more information. This is an important strategy and an area of investigation to be followed to allow decision-making based on the best evidence.

BIBLIOGRAPHY

Al-Shehari, T., & Alsowail, R. A. (2021). An Insider Data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine Learning Techniques. Entropy, 23(10), 1258. https://doi.org/10.3390/e23101258

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, *12*(16), 4102-4107.

Bacao, V. F., & Lobo, V. (2007). Fundamentals of data preparation and pre-processing. *NOVA-IMS, Lisboa*.

Bhatnagar, R., & Babbar, M. (2019). A systematic review of sports analytics. *ResearchGate*. https://www.researchgate.net/publication/332406802_A_systematic_review_of_sports_analytics

Brownlee, J. (2020). Repeated k-Fold Cross-Validation for Model Evaluation in Python. *MachineLearningMastery.com*. https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/

Caine, M. P., Skinner, J., Dogan, S., Kondoz, A. M., Peter, T., Axtell, E., Birnie, M., & Smith, B. J. (2018). Player Tracking Data Analytics as a Tool for Physical Performance Management in Football: A Case Study from Chelsea Football Club Academy. *Sports*, *6*(4), 130. https://doi.org/10.3390/sports6040130

Cooper, A. (2019). AZ Alkmaar - Combining Data and Technology to Evolve Recruitment Processes. *Stats Perform*. https://www.statsperform.com/resource/az-alkmaar-combining-data-and-technology-to-evolve-recruitment-processes/

Definição de valor de mercado - Análise do valor de mercado - Forum | Page 1. (2021). Transfermarkt. https://www.transfermarkt.us/definicao-de-valor-de-mercado/thread/forum/484/thread_id/3314

Gornic, P. (2019). Global Market Transfer Report Confirms Football Industry Growth. World Football Summit. https://worldfootballsummit.com/global-market-transfer-report-confirms-football-industry-growth/

Kröckel, Pavlina. (2019). Big Data Event Analytics in Football for Tactical Decision Support.

One Month On: 5 billion engaged with the FIFA World Cup Qatar 2022™. (2023). FIFA. https://www.fifa.com/tournaments/mens/worldcup/qatar2022/news/one-month-on-5-billion-engaged-with-the-fifa-world-cup-qatar-2022-tm

Rithp. (2023, January 29). The main parameters in XGBoost and their effects on model performance. *Medium*. https://medium.com/@rithpansanga/the-main-parameters-in-xgboost-and-their-effects-on-model-performance-4f9833cac7c

Robot, S. (2022, August 31). *Scraping Websites for Academic Research and School Projects*. Scraping Robot. https://scrapingrobot.com/blog/scraping-website-for-academic-research/

Rodriguez, J. (2018). Understanding hyperparameters optimization in deep learning models: Concepts and tools.

Saldos de transferência (receitas e despesas). (2023). Transfermarkt. https://www.transfermarkt.pt/transfers/transfersalden/statistik/plus/0?sa=1&saison_id=2012&saison_id_bis=2023&land_id=&nat=&kontinent_id=6&pos=&w_s=&plus=0

sklearn.ensemble.RandomForestRegressor. (2023). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

sklearn.metrics.mean_absolute_error. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

sklearn.metrics.r2_score. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

sklearn.model_selection.RepeatedStratifiedKFold. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedStratifiedKFold.html

sklearn.preprocessing.MinMaxScaler. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

sklearn.preprocessing.OneHotEncoder. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

sklearn.preprocessing.RobustScaler. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

sklearn.tree.DecisionTreeRegressor. (2023). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

Transfer records. (2023). Transfermarkt. https://www.transfermarkt.us/statistik/transferrekorde

Try and Except in Python - Python Tutorial. (n.d.). https://pythonbasics.org/try-except/

Turney, S. (2022). Pearson Correlation Coefficient (r) | Guide & Examples. *Scribbr*. https://www.scribbr.com/statistics/pearson-correlation-coefficient/

Turney, S. (2022a). Chi-Square Test of Independence | Formula, Guide & Examples. *Scribbr*. https://www.scribbr.com/statistics/chi-square-test-of-independence/

UEFA.com. (2023). *Club coefficients | UEFA Coefficients | UEFA.com*. https://www.uefa.com/nationalassociations/uefarankings/club/#/yr/2023

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. PLOS ONE, 14(11), e0224365. https://doi.org/10.1371/journal.pone.0224365

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*