Estimating the most important football player statistics using neural networks

Universiteit Utrecht

Bachelor Artificial Intelligence, Utrecht University (7.5 ECTS)

Name student: Kasper Goes
Supervisor : Dr. M. (Thijs) van Ommen
Sudent Id number : 6529283

Abstract

Objectively knowing when an individual football player is good can be a difficult task, as football is a team sport and therefore the number of factors that must be taken into account are huge. Human judgements in these situations are therefore often mislead. Studies have shown that neural networks are a useful tool for these situations, due to their ability to extrapolate complex relations between input and output. In this thesis, we ask which player statistics are most important for determining a player's performance in football. We present two neural networks that attempt to do this by looking at the defending and attacking statistics separately. As a measure of strength of a team we took the numbers of goals the team was expected to score. Results from testing the algorithm showed better results for the attacking model than the defending one. Overall the results looked promising but there is still room for improvement. Future study will definitely need to take into account a player's competition strength in order to make a better judgement about a player's objective strength.

Acknowledgements

I would like to take this opportunity to thank my supervisor Thijs van Ommen for the many helpful comments he provided during the coding process. I would also like to thank the football club AZ, in particular Barend Verkerk (Head of Data Science AZ) for giving me the opportunity to be a part of a unique project and to work at a very special and interesting environment.

Contents

Chapter 1

Introduction

Knowing what makes a football player good poses an interesting challenge due to the fact that the sport is so popular and widespread and the amount of money involved in the sport. However, getting an answer to this question is a difficult problem because football is a team sport, so in order to measure the performance of an individual there are a huge number of factors that must be taken into account that are hard to be quantitatively valued or modeled. This is why it is so hard for football clubs to objectively assess the quality of a player. Hence, all football clubs have to mainly trust the subjective opinion of scouts to determine how good a player is and whether to buy/sell a player. The Dutch professional football club AZ has the ambition to change this old-fashioned scouting process and to come up with a solution to this problem that is objective. According to AZ, even the best scouts and coaches have a bias and they are never able to see everything. The idea is that algorithms are less biased and are able to see more and therefore might offer a better solution. For this thesis I worked together with AZ to try to get closer to that solution. Because the club is one of the market leaders when it comes to football analytics a lot of data was available. Over 200 statistics are tracked for each player per game. They made me aware of the problem that they know when a player has a good specific stat by comparing it to other players, but that it is hard to determine which of the statistics are the most important for assessing how good a player is. The expectation is that machine learning will have an answer to this problem.

In the domain of machine learning Artificial Neural Networks (ANNs) are perhaps the most commonly applied approach among machine learning mechanisms to sport prediction problems [2]. This is because of their strong capability to find complex non-linear relations between inputs and outputs. This is important because most relationships in sport science are unfortunately non-linear. Using ANNs we try to answer the following research question.

RQ: What player statistics are most important for determining a player's performance in football?

This investigation was intended to determine which statistics offer the most information and qualify for performing the role of explanatory variables in the neural models. After studying the statistics I noticed that they could be divided into either attacking or defensive statistics. Because of the difference in nature of these two statistics it made sense to split them up to eventually be able to get an indication

of an attacking and defending score of a player.

The approach to see which player statistics were the most important for determining a player's performance was as follows: because it is much easier to objectively assess a team's (amount of goals, or matches won for example) than player performance the idea is to determine which player statistics have a high importance to a team's performance and deduce from that the most important player statistics. The research will take into account the last three seasons from eleven of the top divisions in Europe because that was the available data.

To my knowledge, no prior studies have looked at inferring the most important player statistics by looking at team statistics in football. The most similar study I good find was done on estimating an NBA player's impact on his team's chances of winning using a linear regression model [3].

Chapter 2

Theory

2.1 Artificial neural networks

In this section a short summary of Artificial neural networks (ANNs) is provided. ANN have been proven to be highly capable for recognizing patterns between input data and a desired output. They are for example used for the best self-driving cars [5]. The power of neural networks stems from their ability to combine relationships. ANNs can aggregate simple linear relations to complex non-linear relations.

ANNs function in a similar way that biological brains do [1]. In both there are neurons which are connected to each other through synapses. In our brain, input signals for example aroma molecules reach our noses. Then the different molecules (input) are transformed depending on the nature of neurons to eventually get a smell (output). This process is mimicked in ANNs.

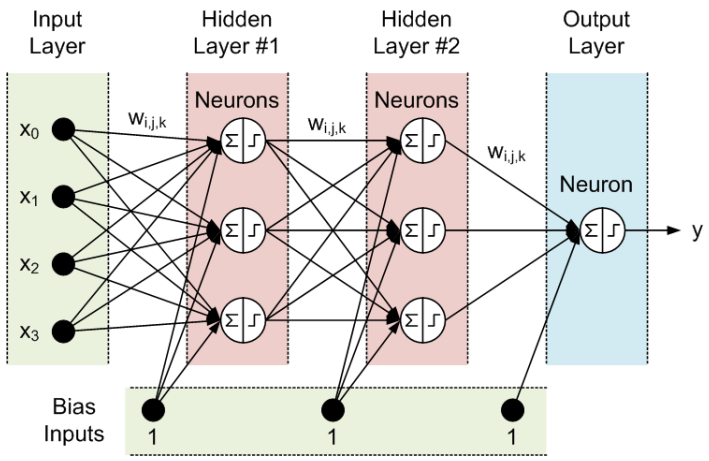

Figure 2.1: A visual representation of an Artificial neural network which has four input nodes two hidden layers and one output node

A neural network consists of an input layer, hidden layer(s) and an output layer. In figure 2.1 [6] you can see a visual representation of an ANN, with its respective layer(s). The values of a layer are passed through to its next layer by taking a weighted average of the nodes plus a bias and filling that into an activation function. The different layers can have one or more neurons per layer depending on the problem. Each node can also have any activation function, but it is conventional for each node in layer to have the same activation function.

A neural network can learn to estimate the best parameters by using a function

that evaluates how good the fit is. After that, the optimal weights and biases can be estimated by using an optimisation algorithm. Although the structure of a ANN and a human brain are similar, in order for an ANN to estimate the correct parameters it needs a lot more data than a human [7].

2.2 Permutation importance

One of the most basic questions we might ask of a ANN model is: what features in the model are the most important? This concept is called feature importance. Because ANNs are a black box in the sense that while it can approximate any function, studying its structure won't give you any insights on the structure of the function being approximated. So to find out which features have the most impact on the output we need to use another method. There are multiple ways of figuring out the feature importance of a neural network, here we have chosen for the permutation method because it is fast to calculate, widely used and understood and consistent with properties we would want a feature importance measure to have.[1]

Permutation importance is calculated after the model has been trained. So our input will generate the same output. Then in order to find out the importance of a column we randomly shuffle a single column, leaving the target and all other columns in place, i.e one column of the data is no longer corresponding to the real world. We should expect less accurate predictions with one randomly shuffled column. Then we compare the accuracy of our ANN with one shuffled column and without the shuffled column. The performance deterioration measures the importance of the column we just shuffled. If we perform this process for every column it is possible to understand and compare the importance of each column.

[1] https://www.kaggle.com/dansbecker/permutation-importance

Chapter 3

Data

In order to understand choices made in the methodology later it is important to first understand the data we are working with. The raw data in this case study is provided by StatsBomb[1]. StatsBomb is a market leader when it comes to the football analytics field. They collect player event data in two ways: by analysing video images automatically and by recording specific events by hand. AZ buys the raw player event data from StatsBomb and processes it to fit their requirements. The dataset that will be worked with in this project consists of player football statistics from the following eleven European competitions: Premier League, Champions League, Eredivisie, Ligue 1, Bundesliga, La Liga, Serie A, Liga Nos, Bundesliga and the Keuken Kampioen Divisie. The datasets contain all the data from the last two seasons, dating back to the season 2019/2020. In total it contains 7.515 matches. After AZ has processed the data for each match of every player in the team 206 stats are present. The total list of statistics can be found in the appendix.

3.1 Important statistics

Below I will explain some important statistics that might be hard to interpret without further explanation. It is important to understand these statistics to be able to understand later why we prefer one over the other for our models. All of the following statistics are commonly used in the football analytics field, but are all calculated by AZ from the raw player event data provided by StatsBomb. The source of the explanation for these statistics comes directly from AZ, the source is not public and therefore there is no reference.

xG
xG is an abbreviation for 'expected goals', it measures the quality of a shot on goal based on several variables such as shot angle, distance from goal, angle to goal and the defenders in front of the shot. An xG rating of 0.1 means a player is expected to score one out of ten shots from that specific situation. The xG is calculated by using a logistic regression model.

xA
xA is an abbreviation for 'expected assist'. An assist is the last pass to the scorer of a goal. Expected assists measured the likelihood that a given pass will become a pass

[1] https://statsbomb.com/

that leads to a goal. It considers several factors including the type of pass, whether the pass is a through pass, the location of the pass end-point and the length of the pass. Adding up a player's expected assists gives us an indication of how many assist a given player of a team will make. xA is better indication of how good a player is at making assists than real assist because for making an assist you are as a player reliant on a teammate scoring a goal after your pass. So a player could make ten amazing passes, but have his striker miss all ten. On the other hand a player could make one meaningless pass in the middle of the field and have that player score a goal. If we would only look at assists the second player would perform better, this is ofcourse not what we want. xA corrects for this.

xT
xT is an abbreviation for 'expected threat'. Expected threat is the total added probability of scoring from all passes/dribbles combined based on the start and end location of each pass/dribble. Only successful passes/dribbles are included in this calculation. Also this only considers the increase in scoring chance within the five subsequent events. xT is calculated by using a Markov Model.

xPVScoring/Conceding
xPV is an abbreviation for 'Possession value'. Every action that a player makes increases or decreases the probability of making a goal for their own team, at the same time that same action also increases or decreases the probability for the opposing team to make a goal. When we look at the probability of our own theme making a goal we use the terms xPVScoring and we use xPVConceding for our opponent. In calculating this increase or decrease in probability we take the ten subsequent events into account instead of the five like we did with xT.

Zones
Over 65 of the 205 stats contain the term Zone. A Zone refers to a specific part of the field. The field is divided into a grid structure as seen in figure 3.1 below.

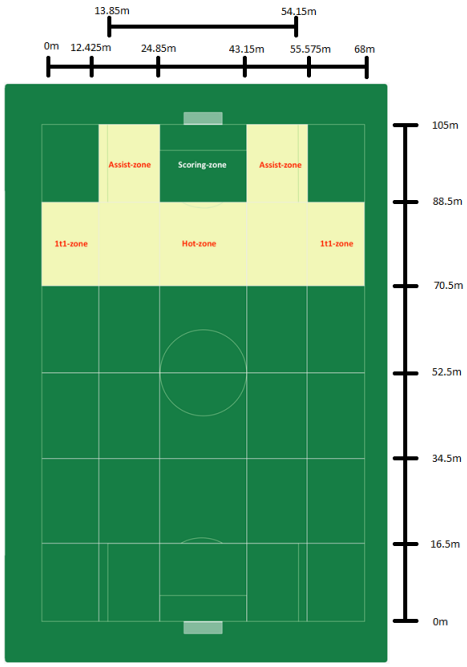

Figure 3.1: A top view of a Eredivisie sized football field showing the grid structure and the zone names.

The boxes closest to the goal have been given specific names: Scoring-zone, Assist-zone, Hot-zone and 1s1-zone as can be seen below. Creating these zones creates the ability to display actions with more specificity. For a striker it could for example be very important to score well on the dribblesHotZoneToAssistZone Statistic. Looking at these specific regions of the field is made possible because of the creations of these zones.

After defining the zones of a field we can start to take a look at the important zones for a specific role, so called 'KeyZones'. You can imagine that for a left back different boxes in figure 3.1 are important than for a right winger. Each position has its own respective KeyZones that are most important for its role.

Chapter 4

Methodology

4.1 Software

The software that is used to conduct this experiment is Python within Jupyter Notebooks and Google Colaboratory. Data manipulation and modelling is done using the packages Pandas, Matplotlib, Sklearn and Keras with Tensorflow as backend.

4.2 Transforming the data

The data currently has the statistics of each player. Because we will be trying to fit the the player data to the teams performance, we need to make sure that the player data is aggregated per team. This is done by transforming the data. First all the statistics of the goalkeepers need to be removed because they have no impact on the xG and clutter the data. All statistics except the percentages can be summed for the team. We do this by grouping the statistics based on matchid and teamid. After this percentages need to be recalculated because the base has changed. For example, the statistic completedpassesPerc is calculated per player in the original data. Adding up the percentages of each player will not result in the correct percentages of the team. To calculate the correct percentages for the team, we need to first calculate the total completedPasses and passes for the team and divide these. After this we are left with the aggregated team statistics for each team for each match.

4.3 Artificial neural network

4.3.1 Determining the output

The output of our ANN should capture the performance of the team in question. Is it very important to pick the statistic which captures this performance the most accurately. One might assume that the amount of points that a team scores is a good performance indicator. But you have to take into account that football is a very stochastic game, therefore a team can win a match despite being a worse opponent. Because of the stochastic element in football we would like to use a better performance indicator than the amount of points a team has won. Another more precise statistic to consider would be the amount of goals scored by a team. This is more accurate than the amount of points won because it captures the dominance of a team better. If a team wins by 6-0 or by 2-1 they get the same amount of

points even though the team was much more dominant in the first game. So we can conclude that the amount of goals scored is a better estimate of a team performance than the amount of points won. But there is a performance indicator which is even better than the amount of goals scored, namely the amount of expected goals (xG). xG is more accurate because it takes into account lucky hits and unfortunate misses a team makes. For example, a striker of a team can miss two easy tap-ins. The amount of goals suffers from this, but xG does not.

So we can conclude that xG is the best indicator of team performance. For the attacking model we can use the xG of the entire team and for our defending model we can use the xG of their opponent, because the amount of expected goals that get scored against you is an indication of the team's defending strength.

4.3.2 Determining the input for attacking and defending

After transforming the dataset into team statistics we are ready to split our dataset into the attacking and defending input sets. To be eligible to the attacking or defending input set the features have to meet a few criteria. The reason for this will be explained further. These criteria are:

- Don't bear too much resemblance to the output
- Indicate player performance more than team dominance
- Say something about their respective fields (defending/attacking)

Don't bear too much resemblance to the output
The goal of our model is to learn relevant statistics for AZ. This means that the statistics need to differ from the information that is already contained in the xG. An example of a statistic that violates this principle is goalAttempts. Although the amount of goal attempts might be a very good predictor of the xG, the stat has no added value for AZ. Because it is self explanatory that a player that has a lot of goalAttempts scores a lot. This gives no extra information besides the xG and is therefore not useful. To identify the statistics that might bear too much resemblance to the xG, the correlation for each input statistic was calculated to the xG. For some of the statistics it was trivial that they had to be removed, for some it was much harder. This was especially the case for xA and xT, who had a correlation of 0.73, 0.71 respectively. After long deliberation it was decided that xA was too similar to the xG and had to be dropped but that xT could stay. The reason for this is that xA only looks at two subsequent actions, while xT looks at five.

Indicate player performance more than team dominance
Because this tool will eventually be used for determining a player's performance and not a team performance we need to pick statistics that indicate the quality of a player rather than indicate the performance of a team. An example might help explain this. If we take a look at the statistic actionsInHotZone which describes the amount of actions a player has made in the HotZone. This statistic is too reliant on a team's strengths compared to its opponent, since if your team is much better it is expected to have a higher possession of the ball and have the more frequently on its opponent half, therefore making the amount of actions that occur in the HotZone higher. One could say that a good player is playable more often and therefore

has more actions in the HotZone than a bad player. Although this is true, we still thought that the statistic had more influence from that team than from the player. This is sometimes a gray line, that's why the team of data and football experts at AZ was very useful. They helped to pick the statistics that indicate a player's performances. When possible the statistics with a percentage were picked because they are not so reliant on the team's strength.

Say something about their respective fields (defending/attacking)
For the statistics that are left over we have to decide if they belong to the attacking or defending category. In the case of xPVScoring/Conceding this is easy, for some this is more difficult. These choices were again made in consultation with the team of data and football experts at AZ.

After this process we were left with the two input sets attacking and defending. The last thing that was done before finalizing the two input sets was to check for multicollinearity. Multicollinearity occurs when independent variables in a regression model are correlated. This is a problem because independent variables should be as independent as possible. If the degree of correlation between two input variables is high enough, it can cause problems with the fitting of the model and interpreting the results. Let's say variables A and B are highly correlated. A neural network after training could only use variable B for predicting the output and ignore variable A. This could then be interpreted as if only variable B has any meaning. When in reality since both variables are so highly correlated they capture the same information. So the neural network only needs one variable to have the information from both. This is why it is important to check for multicollinearity before creating your ANN. The statistics with a correlation higher than 0.8 were removed.
We were now left with 29 statistics to capture the attacking part and 23 statistics For the defending part. A list of these stats can be found in table 4.1 and 4.2 along with their correlation to the xG.

Attacking statistics	Correlation
xT	0.71
completedCrossesLate	0.44
xPVDribblesScoring	0.42
completedThroughPasses	0.37
dribblesHotZoneToAssistZone	0.31
completedPassesPerc	0.30
dribblesHotZoneToScoringZone	0.30
dribbles1t1ToAssistZone	0.27
completedSecondaryActionsPerc	0.27
completedCutBackPasses	0.26
completedPassesUnderPressurePerc	0.25
dribbles1t1ToHotZone	0.25
dribblesAssistZoneToScoringZone	0.23
sideSwitchInitiated	0.22
completedCrossesEarly	0.15
regainPossessionDuelOutsideKeyZones	0.12
possessionRegainInPlayWithin7Sec	0.09
dribbles1t1ToScoringZone	0.08
xPVDefendingInKeyZonesScoring	0.07
xPVDefendingOutsideKeyZonesScoring	0.07
attackingHeadersOnTargetPerc	0.05
regainPossessionInterceptionOutsideKeyZones	0.03
xPVPassesOutsideKeyZonesScoring	-0.01
possessionLostOpponentHalf	-0.02
xPVGoalAttemptsScoring	-0.03
attackingDuelsWonPerc	-0.10
possessionLostOwnHalf	-0.19
xPVPassesInKeyZonesScoring	-0.23

Table 4.1: Input statistics for the attacking model along with their correlation to the xG.

Defensive Statistics	Correlation
oppActionsInBoxResponsible	0.65
oppActionsInKeyZones	0.44
completedCrossesAgainst	0.40
oppPassesToKeyZones	0.36
oppPassesPerDefendingAction	0.34
possessionLostOwnHalf	0.23
regainPossessionInterceptionInKeyZones	0.13
regainPossessionDuelInKeyZones	0.07
possessionLossWithin7Sec	0.06
xPVPassesInKeyZonesConceding	0.05
xPVPassesOutsideKeyZonesConceding	0.02
blockedShotsResponsiblePerc	0.01
xPVDefendingInKeyZonesConceding	0.00
regainPossessionInterceptionOutsideKeyZones	-0.05
xPVGoalAttemptsConceding	-0.05
xPVDefendingOutsideKeyZonesConceding	-0.07
possessionLostOpponentHalf	-0.09
defensiveAirDuelsInKeyZonesWonPerc	-0.10
possessionRegainInPlayWithin7Sec	-0.10
completedSecondaryActionsPerc	-0.11
completedSecondaryPassesPerc	-0.13
regainPossessionDuelOutsideKeyZones	-0.16
xPVDribblesConceding	-0.17

Table 4.2: Input statistics for the defending model along with their correlation to the xG.

4.3.3 ANN implementation

To figure out which statistic is the most important for determining the xG, the ANN first needs to be trained and tested on the dataset. The attack and defending ANN were both trained on the same dataset which amounted to 4.810 matches for the training, 1.202 for the validation, and 1.503 for the test set. The activation function chosen for both neural networks is ReLU because it is simple, fast and empirically it seems to work well [4]. For the loss function we experimented with mean squared error and mean absolute error. Mean absolute error performed better so that one was chosen as our loss function. For the hidden layers, we looked at two hidden layers versus one hidden layer. In both models one hidden layer performed better. The amount of epoch was chosen by starting at 5 epochs and iterating with steps of 5 until epoch 200 where It became clear that after that more epochs lead to overfitting. The best results found for both models around 120 epochs. Overall the amount of epochs and the neuron configuration did not have a big impact on the error measure. An overview of the hyperparameters used for each model can be found in table 4.3.

	Attacking	Defending
Activation Function	ReLU	ReLU
Loss function	mae	mae
Epochs	120	120
Total layers	3	3
Hidden layers	1	1
Neurons	29-7-1	23-5-1
Learning rate α:	0.001	0.001

Table 4.3: The hyperparameters that were used for the attacking and defending model

4.3.4 Creating a baseline error

After calculating an error measure for your trained model you would like to know how well your model performed. But how do you know if your results are any good? You need a basis for comparison of results, a meaningful reference point to compare with. This is called a baseline result. Because we are making a regression model and mean absolute error as loss function, we will use the mean xG to compute our baseline result. First we compute the mean xG by calculating the average xG over all matches in the dataset. Then for every match we calculate the difference of its xG and the mean xG. The average of this difference is our baseline result. Finally we calculate the permutation feature importance, for explanation see section 2.2. To check the performance of weights calculated by the permutation feature importance, the list of weights was ran through the entire player dataset and the top ten players were picked for both models. After a quick testing for the defending model, surprisingly the offensive positions seemed to perform best. Because we wanted to evaluate how good the defending model was, we decided to only select defenders because it was easier to determine their defensive strength.

Chapter 5

Results

The baseline mean absolute error (mae) is 0.487. This means that given the average xG the absolute error is 0.487 off the real xG.

5.1 Results for the attacking model

The mae for the attacking model is 0.272. this means that on average the model is 0.272 expected goals of of the actual xG. Figure 5.1 shows the error over time during training and figure 5.2 shows the error distribution of the model.

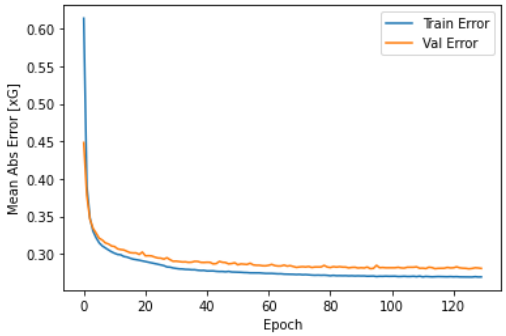

Figure 5.1: The training and the validation mean absolute error plotted for each iteration during the training phase of the attacking model.

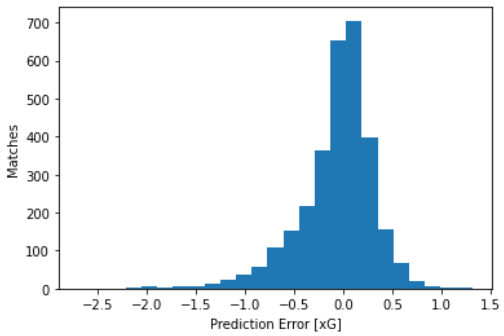

Figure 5.2: Bar chart of the prediction error of the attacking model

In table 5.1 the permutation importance of the attacking model is depicted. As you can see, the xT is by far the most important statistic with a weight of 0.22. The next statistic xPVGoalAttemptsScoring has a weight that is more than four times less important. After the statistic completedThroughPasses the statistics become close to irrelevant very fast. Some statistics even have a small negative weight which means that in those cases the predictions on the shuffles data happend to be more accurate than the real data.

In table 5.2 you can see the top ten attacking players calculated by using the permutation weights of the attacking model. The results in this table look very promising. It is difficult trying to objectively evaluate this table, because one of the reasons we are making this model is to figure out objectively what the best players are. To get an indication of the accuracy of the model I asked for help from the football experts at AZ. So keep in mind that the following statements about

the top ten players are somewhat subjective. It is hard to deny that most of the players on the list are considered one of the best in the world. Keep in mind that this list only account for the attacking capability of the players. As seen Lionel Messi is at top of the list. Lionel Messi is widely considered to be the best attacking player of all time. At number three is Leonardo Spinazzola of the stars of the 2021 European Championship. The next player is Neymar Junior, the player with the most expensive transfer in history of the sport.[1] After Neymar comes Jack Grealish, he is consired on of the best players in the Premier League right now. At number six in the list is Kevin de Bruyne, according to AZ he is currently the best midfielder in the world. The second player in the ranking is Ángel Di María. Although he is a very good player, he is not considered as one of the top ten attacking players in the world. It is very surprising to see Dušan Tadić, Pedro Gonçalves and Ruggero Mannes appear so high in the rankings because they are not considered as elite attacking players. Especially Ruggero Mannes who plays in the Dutch second league (Keuken Kampioen divisie). Although this looks strange, it makes sense, because it is easier to get good statistics playing in a 'bad' league compared to the other players in the database because your opponents here are much weaker. This concept is also applicable to Ángel Di María, Dušan Tadić, Pedro Gonçalves, who all don't play in good but not elite competitions.

Atacking statistic	Weight
xT	0.222818
xPVGoalAttemptsScoring	0.054230
xPVDribblesScoring	0.028103
completedThroughPasses	0.014408
dribblesHotZoneToScoringZone	0.004697
completedCrossesLate	0.003423
possessionLostOpponentHalf	0.002739
completedPassesUnderPressurePerc	0.002086
completedCrossesEarly	0.001778
xPVPassesInKeyZonesScoring	0.001310
dribblesAssistZoneToScoringZone	0.001224
xPVDefendingInKeyZonesScoring	0.001202
completedPassesPerc	0.001186
xPVPassesOutsideKeyZonesScoring	0.000768
dribbles1t1ToHotZone	0.000704
regainPossessionDuelOutsideKeyZones	0.000690
xPVDefendingOutsideKeyZonesScoring	0.000678
sideSwitchInitiated	0.000640
regainPossessionInterceptionOutsideKeyZones	0.000580
completedSecondaryActionsPerc	0.000356
attackingHeadersOnTargetPerc	0.000207
possessionRegainInPlayWithin7Sec	0.000170
completedCutBackPasses	0.000133
dribblesHotZoneToAssistZone	0.000097
dribbles1t1ToAssistZone	0.000068
possessionLostOwnHalf	-0.000035
dribbles1t1ToScoringZone	-0.000123
attackingDuelsWonPerc	-0.000404

Table 5.1: Depicts the importance of each feature calculated by permutation feature importance of the attacking model.

[1] https://www.transfermarkt.co.uk/statistik/transferrekorde

	Name	Score
1	Lionel Messi	0.1095
2	Ángel Di María	0.1090
3	Leonardo Spinazzola	0.1019
4	Neymar Junior	0.1017
5	Jack Grealish	0.1001
6	Kevin de Bruyne	0.0941
7	Jordi Alba	0.0938
8	Dušan Tadić	0.0931
9	Pedro Gonçalves	0.0891
10	Ruggero Mannes	0.0884

Table 5.2: The top ten attacking players, calculated by using the permutation feature importance of the attacking model.

5.2 Results for the defending model

The mae for the defending model is 0.353. this means that on average the model is 0.353 expected goals of of the actual xG, remember that when we say xG in the defending context we are talking about the xG of the opponent. Figure 5.3 shows the error over time during training and figure 5.4 shows the error distribution of the model. As you you can see, the bar graph in figure 5.4 is less steep than that of the attacking model. This makes sense since the error of defending model is higher and therefore makes more mistakes. Looking at the height of the bars, it looks like the model tends to underestimate the xG more often than to overestimate it.

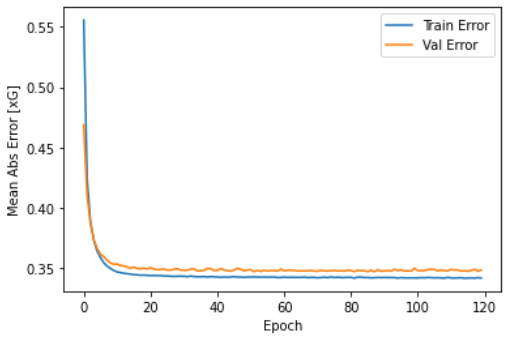

Figure 5.3: The training and the validation mean absolute error plotted for each iteration during the training phase of the defending model.

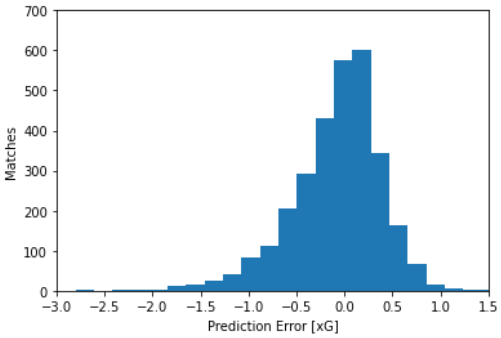

Figure 5.4: Bar chart of the prediction error of the defending model.

Defending statistic	Weight
oppActionsInBoxResponsible	0.188494
completedCrossesAgainst	0.009092
possessionLostOpponentHalf	0.008426
oppPassesPerDefendingAction	0.004537
possessionLossWithin7Sec	0.002694
oppPassesToKeyZones	0.002645
completedSecondaryPassesPerc	0.001923
possessionLostOwnHalf	0.001846
defensiveAirDuelsInKeyZonesWonPerc	0.001768
xPVPassesOutsideKeyZonesConceding	0.001274
oppActionsInKeyZones	0.001112
xPVDribblesConceding	0.000979
regainPossessionInterceptionOutsideKeyZones	0.000714
xPVDefendingOutsideKeyZonesConceding	0.000665
regainPossessionDuelOutsideKeyZones	0.000593
xPVGoalAttemptsConceding	0.000445
regainPossessionDuelInKeyZones	0.000358
regainPossessionInterceptionInKeyZones	0.000184
completedSecondaryActionsPerc	0.000170
possessionRegainInPlayWithin7Sec	0.000018
blockedShotsResponsiblePerc	0.000000
xPVPassesInKeyZonesConceding	-0.000471
xPVDefendingInKeyZonesConceding	-0.000491

Table 5.3: Depicts the importance of each feature calculated by permutation feature importance of the defending model.

In table 5.3 the permutation importance of the attacking model is depicted. Just as in the attacking model it is clear that there is one statistic that rises far above the others when it comes to importance. oppActionsInBoxResponsible is more important than the next most important statistic. This is even more extreme than the attacking model. The statistics become close to irrelevant very fast. Just as in the attacking model some statistics even have a negative importance.

	Name	Score
1	Pontus Jansson	-2.1934
2	Luís Neto	-2.2941
3	Nicolás Otamendi	-2.3251
4	Aritz Elustondo	-2.3716
5	Mads Juel Andersen	-2.4313
6	Sergio Ramos	-2.5184
7	Zouhair Feddal	-2.5184
8	Toby Sibbick	-2.5376
9	Jan Vertonghen	-2.5422
10	Lucas Hernandez	-2.5513

Table 5.4: The top ten defending players, calculated by using the permutation feature importance of the defending model.

In table 5.4 you can see the top ten defending players calculated by using the permutation weights of the attacking model. Notice that the all the scores are negative because they have a negative correlation to the xG. In trying to objectively evaluate the table to the best of my ability, I again asked for help from football experts from

AZ. They concluded that the results are nowhere near the best defending players. The best performing defending player according to the model Pontus Jansson is considered an mediocre central defender. In the attacking model we concluded that some player's attacking strength was overestimated because of their relative weak competition strength. This is not the case for Pontus Jansson who plays in the first league of England (Premier League). The overestimation did occur for six players in the top ten though. Luís Neto, Nicolás Otamendi, Mads Juel Andersen, Zouhair Feddal, Toby Sibbick and Jan Vertonghen all play for the first Portuguese league (Liga Nos) or the second League in England (EFL Championship) which are relative weak competitions compared to the other competitions in the database.

Chapter 6

Conclusion

In this thesis we asked which player statistics are most important for determining a player's performance in football. To answer this question two neural networks were constructed to determine a player's attacking and defending performance. The ANNs were trained with 6.120 matches.

When looking at the results it is clear that the attacking model performed much better than the defending model. The mean absolute error was 0.272, 0.353 respectively. Compared to the 0.487 baseline results it is clear that both models outperformed the baseline error, but that the attacking model performed much better. A possible explanation for this is that most of the events captured are on ball events. Because most of defence happens when players don't have possession over the ball the stats aren't able to accurately capture defending.

After looking at the permutation importance weights it seems like xT, xPVGoalAttemptsScoring, xPVDribblingScoring and completedThroughPasses are the best indicators of a players attacking performance and that oppActionsInBoxResponsible is the best indicator for a players defending strength.

Chapter 7

Discussion

7.1 Limitations

Several limitations should be noted. First of all, it has to be noted that although we've focused on picking statistics that indicate a player's strength and not his team's strength, it's inevitable that in the statistics we've chosen the team's strength also played a role, no matter how small. This could especially be the case for the defending model, where one could argue that the most important statistic for the defending model oppActionsInBoxResponsible is more of an indicator of a team's strength than that of in individual player a player's strength because the stats depicts the quantity of actions a team has allowed in their own penalty arena. The worse your team is, the more often your opponent will come to this area. With this stat removed the model performed much worse (mae of 0.42). If one would classify the statistic oppActionsInBoxResponsible as a team's performance indicator it would confirm the argument we had arrived at in the conclusion that defending is just really hard to convert into statistics.

Second is that the results of the models are very generalized. The attacking and defending statistics are seen as equal for every player, when in reality this is of course not the case. A striker needs other strengths than a right-midfielder in order to be considered good at attacking. The same logic applies to the defensive side where a midfielder needs other strengths than left-back in order to be considered good in defending. Although the models lack this specificity they can still provide useful information for determining the general strength of a player.

The third limitation is that the models do not take into account the strength of the player's competition. This definitely should be taken into account because the stronger the league the player plays in, the more impressive it is to perform well. The same logic applies to players that play in a weak league, where a good performance is less impressive. This effect was definitely visible when we took a look at the ten best players according to our model, where very mediocre players secured a spot in the top ten only because they played in a weak competition. This difference of league strength could be compensated for by calculating a competition factor based on the strength of the league and factoring this in with the players attacking score. Without the competition factor it is only possible to compare players from the same league or from leagues with the same strength.

Last I would like to point out that it is very hard to evaluate the performance of the models because it is hard to know if the model perceives something accurate that people are not seeing or that the model is just performing bad. I want to explain this

further with an example: it is clear that Lionel Messi is a better attacking player that a striker from the second Dutch division because the two are so far apart in attacking quality. This is much harder to determine when we are comparing player A and player B with similar quality, from the same team for example. Let's say that player A is considered the better player according to the media but our model says that player B is better. Is our model seeing something that the media is missing or is our model making mistakes that we are not aware of?

7.2 Future research

As stated before, a competition strength factor needs to be calculated in order to get a more accurate representation of a player's strength. This could be a research project on its own. A possible way of assessing a competition's strength is by looking at how different competitions perform against each other in international tournaments.

A problem with our model is that it is very general as stated before. When buying a new player, a club often looks for a certain position. Therefore it would be useful to be able to see what player statistics are important for a certain position.

Bibliography

[1] S Agatonovic-Kustrin and Rosemary Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research". In: *Journal of pharmaceutical and biomedical analysis* 22.5 (2000), pp. 717–727.

[2] Rory P Bunker and Fadi Thabtah. "A machine learning framework for sport result prediction". In: *Applied computing and informatics* 15.1 (2019), pp. 27–33.

[3] Sameer K Deshpande and Shane T Jensen. "Estimating an NBA player's impact on his team's chances of winning". In: *Journal of Quantitative Analysis in Sports* 12.2 (2016), pp. 51–72.

[4] Jianli Feng and Shengnan Lu. "Performance analysis of various activation functions in artificial neural networks". In: *Journal of physics: conference series*. Vol. 1237. 2. IOP Publishing. 2019, p. 022030.

[5] Sorin Grigorescu et al. "A survey of deep learning techniques for autonomous driving". In: *Journal of Field Robotics* 37.3 (2020), pp. 362–386.

[6] Raqeeb H Rajab and Hussain H Ahmad. "Analysis of Thermosiphon Heat Pipe Performance Using an Artificial Neural Network". In: *Journal of The Institution of Engineers (India): Series C* 102.2 (2021), pp. 243–255.

[7] Anthony M Zador. "A critique of pure learning and what artificial neural networks can learn from animal brains". In: *Nature communications* 10.1 (2019), pp. 1–7.

Appendix A

Appendix

Table A.1: All available statistics

Statistics	
xG	compId
personId	positionId
starting	substitutedIn
substitutedOut	redCard
positionChanged	minutesPlayed
xPVScoring	xPVConceding
xPV	xGBuildUpFixed
xPVPassesOutsideKeyZones	xPVPassesOutsideKeyZonesScoring
xPVPassesOutsideKeyZonesConceding	passes
completedPasses	completedPassesPerc
passesToPreAssistZone	passesTo1t1Zone
passesToHotZone	passesToAssistZone
passesToScoringZone	passesToKeyZones
completedLongPassesForward	sideSwitchInitiated
sideSwitchReceived	sideSwitchDangerInitiated
sideSwitchDangerReceived	secondaryPasses
completedSecondaryPasses	completedSecondaryPassesPerc
secondaryActions	completedSecondaryActions
completedSecondaryActionsPerc	possessionLost
possessionLostOwnHalf	possessionLostOpponentHalf
attackingDuels	attackingDuelsWon
attackingDuelsWonPerc	keyPasses
xA	xPVPassesInKeyZonesScoring
xPVPassesInKeyZonesConceding	xPVPassesInKeyZones
xT	passes1t1ToHotZone
passes1t1ToAssistZone	passes1t1ToScoringZone
passesHotZoneToAssistZone	passesHotZoneToScoringZone
passesAssistZoneToScoringZone	progressivePassesInKeyZones
passes1t1ToHotZonePerc	passes1t1ToAssistZonePerc
passes1t1ToScoringZonePerc	passesHotZoneToAssistZonePerc
passesHotZoneToScoringZonePerc	passesAssistZoneToScoringZonePerc
completedThroughPasses	actionsInFlankingZone
actionsIn1t1Zone	actionsInHotZone

Table A.1 – continued from previous page

Statistics	
actionsInAssistZone	actionsInScoringZone
actionsInKeyZones	actionsInOppBox
crosses	completedCrosses
completedCrossesPerc	crossesLate
completedCrossesLate	completedCrossesLatePerc
crossesEarly	completedCrossesEarly
completedCrossesEarlyPerc	crossesHigh
completedCrossesHigh	completedCrossesHighPerc
crossesLow	completedCrossesLow
completedCrossesLowPerc	cutBackPasses
completedCutBackPasses	completedCutBackPassesPerc
dribbles	completedDribbles
completedDribblesPerc	xPVDribblesScoring
xPVDribblesConceding	xPVDribbles
dribbles1t1ToHotZone	dribbles1t1ToAssistZone
dribbles1t1ToScoringZone	dribblesHotZoneToAssistZone
dribblesHotZoneToScoringZone	dribblesAssistZoneToScoringZone
progressiveDribblesInKeyZones	dribbles1t1ToHotZonePerc
dribbles1t1ToAssistZonePerc	dribbles1t1ToScoringZonePerc
dribblesHotZoneToAssistZonePerc	dribblesHotZoneToScoringZonePerc
dribblesAssistZoneToScoringZonePerc	avgPassLengthReceived
avgPassWidthReceived	avgPassDistanceReceived
longPassesForwardReceived	throughPassesReceived
subjectedToFoul	goalAttempts
shots	shotsOnTarget
shotsOnTargetPerc	shotsOnTargetSaved
shotsOnTargetSavedPerc	shotsOnTargetToRebound
shotsOnTargetToReboundPerc	shotsOnTargetToCorner
shotsOnTargetToCornerPerc	attackingHeaders
attackingHeadersOnTarget	attackingHeadersOnTargetPerc
xGAvgPerAttempt	xG_{05}
$xG_{51}0$	$xG_1 0_2 0$
$xG_2 0_4 0$	$xG_4 0_1 00$
$xG_{05}Perc$	$xG_{51}0Perc$
$xG_1 0_2 0Perc$	$xG_2 0_4 0Perc$
$xG_4 0_1 00Perc$	goals
addedGoalValue	xPVGoalAttemptsScoring
xPVGoalAttemptsConceding	xPVGoalAttempts
goalsInsideOppBox	goalsInsideOppBoxPerc
goalsOutsideOppBox	goalsOutsideOppBoxPerc
goalsByShot	goalsByShotPerc
goalsByHeader	goalsByHeaderPerc
xPVDefendingOutsideKeyZonesScoring	xPVDefendingOutsideKeyZonesConceding
xPVDefendingOutsideKeyZones	oppPassesToKeyZones
defensiveDuelsOutsideKeyZones	regainPossessionDuelOutsideKeyZones
regainPossessionDuelOutsideKeyZonesPerc	indexLeaveFieldEvent
sideSwitchAgainst	oppPassesInResponsibleZones

Table A.1 – continued from previous page

Statistics	
defendingActionsInResponsibleZones	oppPassesPerDefendingAction
xPVDefendingInKeyZonesScoring	xPVDefendingInKeyZonesConceding
xPVDefendingInKeyZones	keyPassesInitiatedAgainst
keyPassesReceivedAgainst	xAAgainst
oppActionsInKeyZones	defensiveDuelsInKeyZones
regainPossessionDuelInKeyZones	regainPossessionDuelInKeyZonesPerc
regainPossessionInterceptionInKeyZones	crossesAgainst
completedCrossesAgainst	completedCrossesAgainstPerc
foulsConcededOutsideBox	goalAttemptsAgainst
goalAttemptsOnTargetAgainst	blockedShots
blockedShotsResponsible	blockedShotsResponsiblePerc
xGAvgPerAttemptAgainst	$\mathrm{xG}_{05}Against$
$\mathrm{xG}_{51}0Against$	$\mathrm{xG}_{1}0_{2}0Against$
$\mathrm{xG}_{2}0_{4}0Against$	$\mathrm{xG}_{4}0_{1}00Against$
$\mathrm{xG}_{05}AgainstPerc$	$\mathrm{xG}_{51}0AgainstPerc$
$\mathrm{xG}_{1}0_{2}0AgainstPerc$	$\mathrm{xG}_{2}0_{4}0AgainstPerc$
$\mathrm{xG}_{4}0_{1}00AgainstPerc$	goalsAgainst
goalsAgainstResponsible	oppActionsInBox
oppActionsInBoxResponsible	clearancesInBox
defensiveAirDuelsInKeyZones	defensiveAirDuelsInKeyZonesWon
defensiveAirDuelsInKeyZonesWonPerc	penaltiesConceded
regainWithGoalAttemptInCounter	involvedInGoalAttemptInCounter
possessionLossWithGoalAttemptInCounterAgainst	possessionLossWithin7Sec
possessionRegainInPlayWithin7Sec	passesUnderPressure
completedPassesUnderPressure	completedPassesUnderPressurePerc
tackles	completedTackles
counterPressFouls	counterPressInterceptions
attackingAerialDuels	attackingAerialDuelsWon
attackingGroundDuelsWon	defensiveDuels
defensiveAerialDuels	defensiveAerialDuelsWon
defensiveGroundDuelsWon	indexEnterFieldEvent