Utrecht University

Clustering soccer players: investigating unsupervised learning on player positions

Gijs Wijngaard

5671833

Bachelor Thesis Artificial Intelligence
7.5 ECTS

Thesis Supervisor: Tejaswini Deoskar

Contents

Abstract

In this study, we investigate the clustering capability of two unsupervised learning clustering methods: K-means and Expectation Maximization (EM). We train the methods on soccer match data of the Spanish competition La Liga, which contains matches from 2004 to 2019. We classify both clustering methods with soccer player positions to visualize a correlation between player positions using Principal Component Analysis (PCA). In these visualizations, we use 4 and 11 clusters that correspond to player positions in the field. To interpret K-means and EM, we use purity and the silhouette score. Results show that K-means classifies the data better than EM. With the use of feature selection methods Laplacian score and correlation mean, we increase the performance of K-means by 37%. We see that a cluster size of 8 clusters has the best separability, which suggests that there are 8 different types of soccer players on the field during a match.

Chapter 1

Introduction

As more and more computer power became available in the past decades, computing statistics using large amounts of data became more mainstream. This rise influenced the use of data science within the field of artificial intelligence as one of the strategies in optimizing data variables.

Over the past years, we also see more and more machine learning techniques have been applied to sports data. Clubs started analyzing their players using statistics and use machine learning techniques to maximize player performance. Sports data analysis can vastly improve team performance and help coaches make the right decisions. A famous example of this is the book and Oscar-nominated movie adaptation Moneyball [11]. This book is based on real events and tells the story of a coach that uses statistics to optimize team performance, even with a small budget. That team then continues to win various competitions.

In this thesis, we will compare two machine learning clustering techniques on soccer data: K-means and Expectation Maximization. We will determine which technique clusters the data the best using 2 cluster evaluation methods: purity and the silhouette score. We conclude that K-means fits the data better. Next to that, we calculate the best cluster size and see that the best way to separate the data is into 8 clusters. At last, we will determine the best and worst features in our data for these clusters using the silhouette score, Laplacian score and the correlation mean. From this, we see that we can increase our cluster performance by 37%.

With this thesis, we gain insightful results on how each player does differ from another player and what important attributes they can have in doing so are. More specifically, this thesis contributes to the knowledge of comparison of two unsupervised algorithms, two evaluation methods and

two unsupervised feature selection methods, based on soccer data. In short, we have answered how we can use unsupervised algorithms to predict and analyze player positions.

We first start with explaining about the data we used and how we cleaned it. We will talk about some preliminary analysis we have done and state the size of data. Then, in the subsection Methodology, we talk about the specific details of all the methods and algorithms we used. We go more in-depth in on the math and its use-cases and why we have used it for our thesis. After that, we will talk about our outcomes and what we have found in the data. We will state and visualize interesting results and put down all the calculations we have done. At last, we will conclude our thesis and state interesting insights we got.

The data we will be using for the results in this thesis is open-sourced data from StatsBomb.com. The data contains events that happen within a match, such as passes, shots on goal and throw-ins. Next to that, we use data from the game of FIFA for preliminary analysis and conclude that good soccer players do not get more passes than worse players in a match.

Next we will focus on the difference between supervised and unsupervised learning. This is because our thesis is primarily focussed on unsupervised learning methods and scores, and to gain understanding of the aforementioned methods used in the thesis some understanding of the difference is needed. After that, we introduce the methods we will be using in this thesis.

1.1 Difference between supervised and unsupervised learning

A commonly used technique in finding correlations in data is the use of supervised learning. Methods include, for example, linear regression, where we fit a linear line onto the data such that we can infer an average of the data. Then, with the use of new data, new outcomes can be predicted. Quite some research has been done with supervised learning on soccer analysis, as will be explained in the related work subsection.

Next to supervised learning, a large field within the domain of machine learning is unsupervised learning. This field is about algorithms to recognize patterns without the use of outputs of the data. A subfield of unsupervised learning is clustering methods. Clustering methods can separate the data by calculating several mean values within the data and label each data accordingly.

When working on supervised machine learning algorithms, we can use

unsupervised learning to explore the data beforehand. An example of this is Principal Component Analysis (PCA) [20], which we can use for visualization of the data. We will use PCA in this thesis for the same purpose. Another example is that for some supervised algorithms, data can be too large to compute in a given timeframe. In such situations, feature selection is needed. Unsupervised techniques that help supervised learning with preprocessing include removing features based on similarity [18] and low-variance [2]. In our thesis, we will use 2 unsupervised learning methods that also focus on detecting and removing bad features.

1.2 Unsupervised learning methods used

2 clustering methods we will use in this paper are K-means [14] and Expectation Maximization [6]. K-means focusses on calculating clusters that are on average the same size, while also classifying each data point as either belonging to the cluster or not. This binarity results in clusters that are visually clearly separable, as each cluster tends to have its own boundaries in a vector space.

On the other hand, silhouette analysis tends to separate clusters using a normal (gaussian) distribution. This distribution means that every data point has a chance of belonging to a cluster. Here, the data point belongs to the cluster with the highest percentage. This results in cluster boundaries that are much less defined, as for every dimension the cluster distribution is different. Visualizing such clusters in 2 dimensions do result labels all over the place, as we will see in the results section.

To compare both clustering techniques, we will be using the cluster evaluation methods purity and silhouette analysis to define which method performs best on the data. Purity can be classified as a supervised cluster method, as it defines its score based on real data. We use it for telling which cluster outcome of machine learning methods correlates to which actual label. The score is then the calculation for each cluster which cluster has the most similarity with the actual label.

Next to that, we will also assess the features of the data based on feature selection methods. In particular, we will use Laplacian scores and correlation means to define how good each feature contributes to an unsupervised learning method. From this analysis, we can deduce what features make that we can separate player positions.

1.3 Related work

Previous work within the field of soccer analysis dominantly contains papers with supervised learning methods. Some notable examples include prediction of soccer outcomes using Monte Carlo [24], pass analysis [28] using spatial reasoning and summarizing soccer matches using SVM and neural nets [30], amongst others. Interestingly, some research has been done focussing on prediction of games for profit, by competing against betting offices [25] [4].

Studies that are closely related to the subject of player classification, is work by Decroos et al. [5], Liu et al. [13] and Mahfuz et al. [15]. In Decroos' work, individual players are analysed using a similar event dataset in comparison to ours to determine the overall strategy the coach imposes on the team. Instead of single events, it calculates event streams and deduces phases during a soccer game.

Work by Liu et al. focusses on classifying players from a video stream of the match. It also uses Expectation Maximization (EM) clustering. Liu uses EM on the different types of player images and overcome the problem that in the dataset not every player cluster has the same size.

The study by Mahfuz et al. focussed on unsupervised learning techniques used in this paper. Mahfuz primarily used K-means and Expectation Maximalization to do overall analysis of player performance. The analysis was focussed on training data and drills and did not include any matches. The study however fails to draw any notable conclusions.

Overall, we can conclude that the related work is dominantly based on supervised learning. This thesis will fill the gap here and contributes the research on applying unsupervised learning techniques on event data of soccer matches.

Chapter 2

Data

In this section, we elaborate on the 2 different datasets we use in our research. For both datasets, we state the size of the data and explain some general information about the dataset. We will also briefly touch on what preliminary particularities we noticed in the data. Both datasets are open-source, and we can download them freely. The first dataset from StatsBomb we have used for the unsupervised models and other methods. The Fifa dataset we used only for preliminary analysis.

2.1 StatsBomb

The data of StatsBomb Open Data [26] contains a sample of 809 matches, all of which are from the Spanish soccer competition La Liga. For each match, it defines what sort of events happened in a match. Some events, for example, are passes, shots and dribbles. A full list of all events is in appendix A. The events in this dataset we end up using for the unsupervised learning methods. The data initially contains data from the following competitions: FA Women's Super League, FIFA World Cup, La Liga, NWSL and Women's World Cup. The vast majority of the data consists of matches of the Spanish competition La Liga, which is why we focus on that competition.

Within that data, we have matches that happened in 2004 until 2019. Not every match within every year of La Liga is made available by StatsBomb. From the 809 matches, we continue with 452 matches that we use. In those 452 matches, 1626804 events have happened, on average 3559 per match. We again cleaned the data such that the columns we are using in the end consists of *player*, *position*, *location*, *type* and *related_events* (see table 2.1).

Type	**Player**	**Position**	**Location_x**	**Location_y**
Pass	Iniésta	Left Center Midfield	58.7	7.9
Dribble	Fernando Torres	Right Center Forward	77.1	42.1
Ball receipt*	Adrián González	Right Defensive Midfield	60.6	24.2
Dribble	Paulinho	Left Center Midfield	77.0	49.0
Carry	ter Stegen	Goalkeeper	13.4	35.8

Table 2.1: Sample of StatsBomb data.

To obtain datasets for the use of clustering player positions, we defined the favourite position of each player based on what position would appear most in the matches he played. To get 4 clusters, we would convert each position of the player to an unspecific label, namely goalkeeper, defender, midfielder or attacker. The same we would do when clustering with 11 clusters (see appendix B for the specific classification).

For clustering, each event contains data about the player that did the event. For each event, we counted the occurrences to gather the attributes, which are equal to the type of event. We averaged the attributes out over the sum of all attributes per player to gather a percentage for each attribute. All empty attributes were assigned 0. We also converted the location to a mean x and y coordinate. We standardized the data by removing the mean and scale it to unit variance. We ended up with 1847 players (rows) each having 25 attributes (columns). In this dataset, there are 635 midfielders, 589 defenders, 498 forwards and 125 goalkeepers.

2.2 Fifa

This data [22] is on the online video game of FIFA. The data originates from the EA sports game FIFA19. The data is received from a Kaggle dataset [8], initially containing 89 columns and 18206 rows. For the preliminary analysis, we use the columns overall and position (see table 2.2). The overall column states the players' overall score. This score is defined as a weighted average of all scores in the data that define how good a player performs [3]. The position column states the players' most used position.

First, we analyzed ways in how we could correlate both datasets to discover what insights we could receive. One insight we tried to discover was how passes of soccer players would correlate with its overall score on Fifa. With the help of network visualizations, it is possible to map how often each player passes to another player within a match (figure 2.1) and within

Player name	Overall statistic	Position
Lionel Messi	94	Forward
Cristiano Ronaldo	94	Forward
Virgil van Dijk	86	Defender
Frenkie de Jong	81	Midfielder

Table 2.2: Sample of Fifa data.

a team during the year (figure 2.2). We did this by calculating for each pass if the related events contain a ball receipt from another player. If true, there is an out-degree between them. Edges size is equal to the number of passes; node size is its FIFA overall score. There was almost no correlation, so interestingly good players do not get the ball more often.

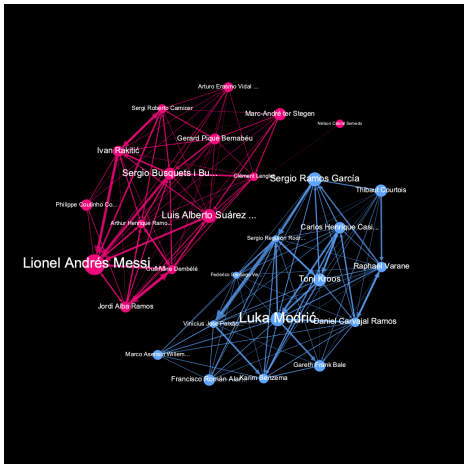

Figure 2.1: Network visualization of a single match. Red is Real Madrid, Blue is FC Barcelona

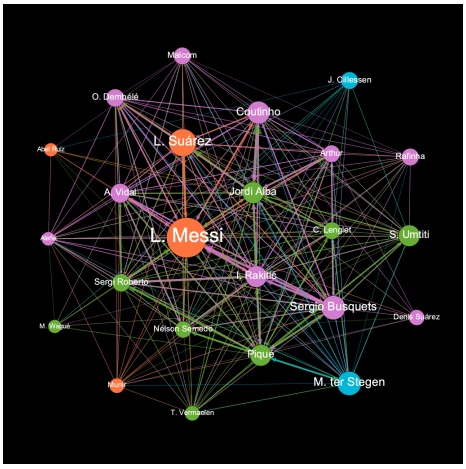

Figure 2.2: Network visualization of a whole team within a year. Blue = goalkeepers, green = defenders, pink = midfielders, orange = attackers

Chapter 3

Methodology

In this thesis, we have used several machine learning and statistical methods to gain insights. In this section, we explain some of these methods, its uses and go more in-depth on the math behind it. We also explain what its use case is and why.

In total, we have used 2 unsupervised clustering methods, namely K-means and Gaussian Expectation Maximization. We also used Principal Component Analysis for visualizing the data. At last, we used 4 scoring methods for evaluating purposes: purity, silhouette score, Laplacian score and correlation mean. We start by explaining the unsupervised clustering methods.

3.1 Unsupervised clustering methods

K-means [14] and Expectation Maximization Algorithm (EM) [6] are both clustering methods, capable of separating large amounts of data in separate clusters. Apart from separation, they help with understanding the data, detecting the outliers and fine-graining the features, amongst others.

Both algorithms try to separate the data to the nearest centroid defined by that cluster. First, you specify the number of clusters. Then, on the first iteration, the mean of centroids are defined randomly. Afterwards, they are defined based on the mean of all the points classified as that cluster. With k-means, we minify the sum of squared errors for the sum of clusters, for each iteration:

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\left\|x_i - c_j\right\|^2 \tag{3.1}$$

K-means and EM, however, differ in how clusters overlap or not. K-means is a hard clustering method; each data point belongs 100% to a single cluster. With EM, clusters can overlap; data points can be in multiple clusters. In EM, each cluster has a multivariate normal distribution $\langle \mu, \mathrm{cov} \rangle$ (Gaussian mixture) instead of the centroids location. These are defined randomly on the first iteration. Each point's i probability is then calculated based on the Bayes rule and the data points location in the distribution:

$$P(c_j \mid x_i) = \frac{P(x_i \mid c_j)P(c_j)}{\sum_{k=1}^{m} P(x_{ki} \mid c_k)P(c_k)} \quad P(x_i \mid c_j) = \frac{1}{\sqrt{2\pi \mathrm{cov}_j}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\,\mathrm{cov}_j}\right)$$

$$P(c_j) = \frac{\sum_{k=1}^{n} P(c_j|x_{ki})}{n}$$

This is called the Expectation step. Next to this step we have the Maximization step. In this step we have the μ and cov, which will be recomputed for each iteration:

$$\mu_j = \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{P(c_j \mid x_i)}{\sum_{m=1}^{n} P(c_j \mid x_{mki})} x_{ki}$$

$$\mathrm{cov}_j = \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{P(c_j \mid x_i)}{\sum_{m=1}^{n} P(c_j \mid x_{mki})} \prod_{l=1}^{m} (x_{ki} - \mu_{lj}) \tag{3.2}$$

Again, we continue until a maximum number of iterations, which we decide beforehand. In this case, we settled with 30000 iterations. Since we are recomputing the $P(c_j)$ on each iteration as well, clusters do not tend to keep the same size, where K-means clusters do. This difference in size leads to different size clusters for each cluster in EM.

Next we will shortly touch on Principal Component Analysis (PCA) [20]. In the thesis we used PCA to visualize the data from 23 dimensions into 2 dimensions, to plot it in a scatterplot. In PCA, we first calculate the covariance matrix $\mathrm{cov}(a, b) = \sum_{i=1}^{n} \frac{x_{ai} \cdot x_{bi}}{n}$, then we solve in the determinant $\det(\mathrm{cov} - \lambda \mathrm{I}) = 0$ to get eigenvalues λ, then we find the eigenvectors by solving $\mathrm{cov}\, e_i = \lambda_i\, e_i$ for every eigenvector i and eigenvalue (principal component). We have 2 dimensions, thus 2 eigenvalues, and we center the original data and project it to each dimension: $\prod_{i=1}^{d} (x - \mu)^T e_i$. The result is x' in 2 dimensions.

3.2 Cluster evaluation methods

To evaluate the k-means and EM clusters we used two statistics: purity [16] and silhouette analysis [23]. Both are used as cluster statistics and evaluate

the performance of a cluster. They differ, however, in the way we use them.

We use purity for knowing how good a cluster performs and uses the actual labels to calculate the correlation between the predicted and actual labels. In this way, purity is a supervised learning statistic since the actual labels of the data are needed. Because cluster algorithms do not use the actual data to create clusters, purity calculates for every actual label the most classified corresponding label and divides that by the total amount:

$$\frac{\sum_{i=1}^{k} max_j \mid c_i \cap t_j \mid}{N} \tag{3.3}$$

In this way, the algorithm with the best purity is the best in classifying the actual data correctly.

Next to this classification, we use silhouette analysis for predicting the best algorithm as well as calculating the best size of clusters. Silhouette analysis calculates how similar data is to data in their cluster in comparison to data in other clusters. No actual labels are needed, and thus silhouette can be applied to unsupervised clustering algorithms. For each point x_i belonging to cluster c and closest cluster d we calculate:

$$\frac{\left(\sum_{k=1}^{m} \|x_i - x_{dk}\|^2\right) - \left(\sum_{j=1}^{m} \left\|x_i - x_{cj}\right\|^2\right)}{\max\left(\sum_{j=1}^{m} \left\|x_i - x_{cj}\right\|^2, \sum_{k=1}^{m} \|x_i - x_{dk}\|^2\right)} \tag{3.4}$$

We then take its mean and get the silhouette coefficient (score) for all the clusters. The silhouette score is a value between -1 and 1, where values > 0 mean the data fits the best within the current cluster.

3.3 Feature selection methods

To verify the results of column and feature selection we have done with the silhouette score, we use a method that focusses on identifying good and bad features in unsupervised data, named the Laplacian score [9]. Laplacian scores score features based on how they preserve each cluster. They seek which features are best in containing two close data points close.

Laplacian scores use a graph where nodes are data points, and edges of each node are its 5 nearest neighbours based on the euclidean distance between the two. We define a weight matrix based on this graph, measuring the similarity between each neighbour if two nodes are connected. The similarity is calculated for every x_i and x_j based on $e^{-\frac{\left\|x_i - x_j\right\|^2}{1}}$ if they are

connected, otherwise 0. We also calculate the degree-matrix of each node to other nodes. The result is a matrix where the diagonal contains the degree of the node, and the rest equals 0. We then calculate the graph Laplacian (L), which is degree-matrix (D) minus the weight matrix, for each feature. With the graph Laplacian we can calculate the Laplacian score (LS) for each feature f based on its normalized feature g (mean removed), where 1 equals a vector of 1's:

$$g = f - \frac{f^T D\, 1}{1^T D\, 1} 1 \quad LS = \frac{g^T L\, g}{g^T D\, g} \tag{3.5}$$

The result is that features that contribute to a small distance between data points have a small Laplacian score, and thus are useful features. Nodes that have a small distance to others have a higher degree and thus a higher degree matrix.

At last, we used the correlation mean. Correlation mean is a method where we cross-correlate two matrices of cluster centres and take the mean to see how much the clusters differ we remove a feature. The hypothesis is that useful features do change the positions of a cluster a lot, while with bad features removed, the cluster tend to keep the same position. Since two matrices are not the same dimension, the removed-feature clusters get an added column of values of 0. Correlation is based on the dot product between the rows of the matrices and summed up. Lastly, we calculate a mean based on all features.

3.4 Software Implementation

For the software implementation of PCA, K-means, Expectation Maximization and the silhouette score, we use the Scikit-learn python library [21]. For the implementation of the Laplacian score, we use the Scikit-feature python library [12]. For the purity score, we define a custom function using Numpy [19] and Scikit-learn's contingency matrix. We base the correlation mean method by using SciPy's function $correlation2d$ [29] and taking the mean. For all the plots in this thesis, we use Matplotlib [10]. For the data preparation and preliminary analysis, we rely on Pandas' data frames [17]. Also, we did some preliminary analysis with R. Lastly, we built the network visualizations in the preliminary analysis with Gephi [1] and used the Fruchterman-Reingold Algorithm [7] for the layout.

Chapter 4

Results

In this section, we put all results we found during analysis of the data. Specifically, we analyze and talk about the results we found and why it is the case we found it. We also state some preliminary conclusions.

4.1 Clustering using 4 clusters

The data we analyzed is 25-dimensional (see appendix A). To visualize this data, we used PCA to reduce the dimensionality to 2 dimensions. For the first clustering task, we set the number of clusters to 4. The number 4 corresponds to the different group of positions in a soccer field, namely the positions of goalkeeper, defender, midfield and forward.

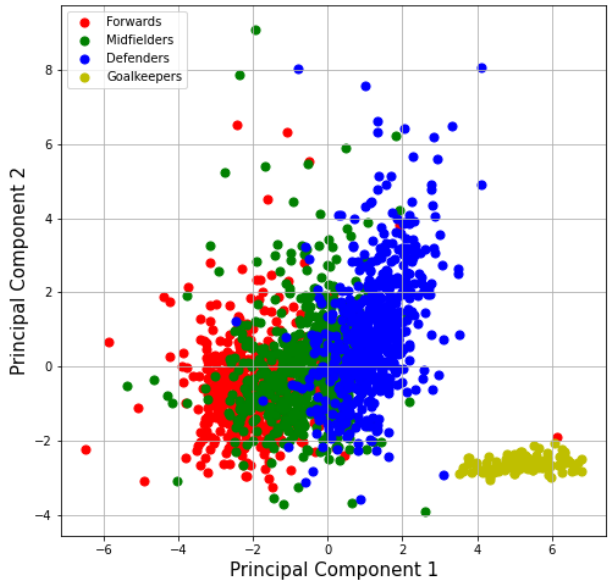

Figure 4.1: Dataset dimensionality reduction using PCA

In Figure 4.1, we can see that the data clusters do overlap, except for the goalkeepers. An explanation for this is that goalkeepers tend to stay on one location, and tend to have quite different actions in comparison to other players on the field. We also see some outliers at $y > 4$ and $x < -4$. Since it is a dimensionality reduction, it is a reduced image of the actual dataset, and variables may be more widespread.

We now visualize K-means and Expectation Maximization (EM) algorithms. In both methods, we used 4 clusters, as we try to model the clusters of the 4 global positions on the field. We trained both methods on all the data with 30000 iterations. The outcome is displayed in figure 4.2 and figure 4.3.

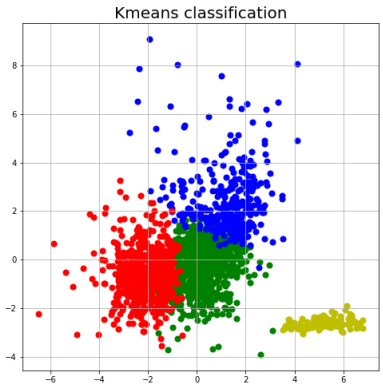

Figure 4.2: K-means clustering with 4 clusters

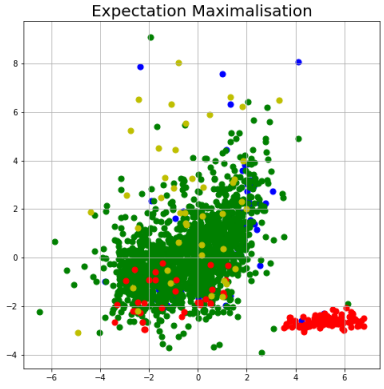

Figure 4.3: Expectation Maximization clustering with 4 clusters

We see that the k-means algorithm tends to keep the cluster size the same, as expected. The EM-algorithm, however, tends to cluster most of the data in 1 group, as explained in the methodology section. Next to that, the variance of all clusters is much more significant, as all players belong to one cluster is much more widespread.

The evaluation gives the results displayed in table 4.1.

	purity	**silhouette score**
K-means	0.6621	0.1347
EM	0.4277	0.2950

Table 4.1: Results of K-means and EM on 4 clusters

We see that k-means does a better job at classifying the right clusters in comparison to EM, according to the purity score. However, the silhouette score is close to 0 and suggests it might be better to take a different number of clusters.

4.2 Clustering using 11 clusters

Next up, we use a high number of clustering, namely 11. This number of clustering corresponds to the number of positions of players on the field. The goal is if it is possible to separate all player positions using these two

algorithms. We separate the clusters with the most common soccer player setting: 1 goalkeeper, 4 defenders, 4 midfielders and 2 attackers [27]. We train the models on the data and again use 30000 iterations. We visualize the data in 2 dimensions using PCA first, see figure 4.4.

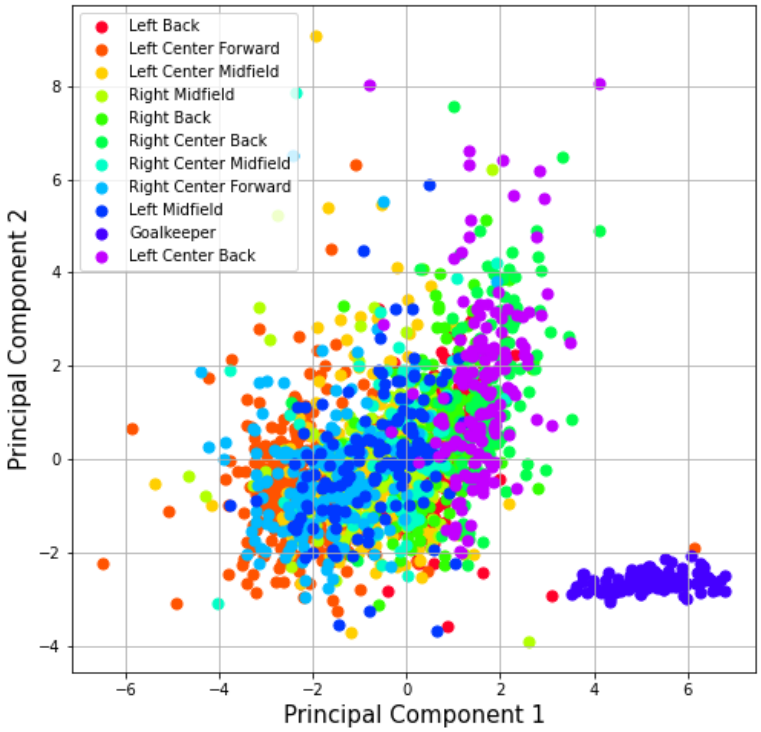

Figure 4.4: Dataset dimensionality reduction using PCA

We see that there is almost no separable data, except for the goalkeepers which again do have their cluster. When performing our two algorithms, K-means and EM, we get the results in figure 4.5 and figure 4.6.

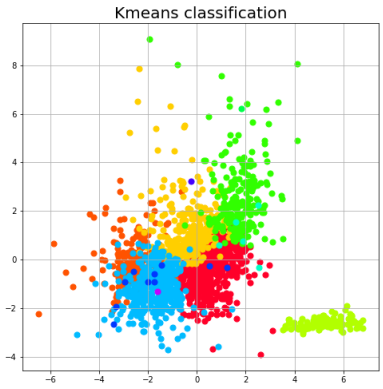

Figure 4.5: K-means clustering with 11 clusters

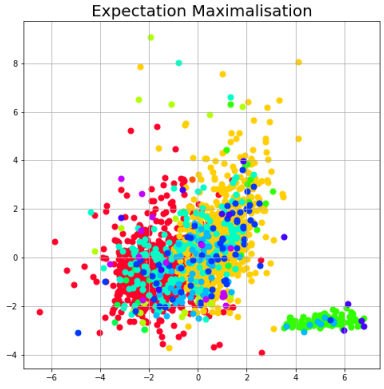

Figure 4.6: Expectation Maximization clustering with 11 clusters

Now, we can see that Expectation Maximization tend to cluster the data somewhat better than with 4 clusters, as we did previously. Next to that, we again see in EM that some clusters are way more extensive than others. We compute the purity and silhouette scores given these algorithms and show results in table 4.2.

	purity	**silhouette score**
K-means	0.3649	0.1161
EM	0.3037	-0.0398

Table 4.2: Results of K-means and EM on 11 clusters

Both the purity and silhouette scores are in all cases way worse than using only 4 clusters. We can see that the individual player's positions are not clusterable at all. The data does not differ enough between the different clusters to separate the players into position categories.

We now know that K-means is a better algorithm. We also know that 4 clusters work better than 11 clusters. For every number of clusters, we train the model and calculate the silhouette scores. We display the result in table 4.3.

We see that a cluster size of 8 has the highest silhouette score, meaning 8 clusters does create the best clusters where each cluster is the most apart from other clusters. When applied to our dataset, this means that we can

Clusters	5	6	7	8	9	10
Silhouette score	0.1302	0.1415	0.1334	0.1485	0.1464	0.1261

Table 4.3: Results of K-means on 5-10 clusters

separate players on a soccer field best in 8 different positions. We continue with 8 clusters and try to use some feature analysis to increase the silhouette score. Also, we continue using K-means, as it gives better purity scores than Expectation Maximization does.

4.3 Feature selection

We now calculate the least and most important features based on two approaches. First, for every feature in the dataset, we run the K-means algorithm without the feature to determine its importance. Next, we pick the highest K-means silhouette score and use that score as a dataset for the next run. We do that iteratively until convergence. Iteratively until convergence means that when it has converged, it is not possible to get a higher silhouette score by removing yet another column from the dataset. We get table 4.4.

Removed columns	-	9	9, 4	9, 4, 0
Silhouette score	0.1485	0.1550	0.1662	0.2046

Table 4.4: Results of silhouette score on least important columns

Thus, by removing features in columns 9, 4 and 0, we can increase the separability of the clusters to a score of 0,2046. This score indicates that we should drop the columns of *Duel* (indicates a duel in a game) *Carry* (a player controls the ball) and *50/50* (2 players challenge to recover a ball) as they do not contribute to any cluster separability. The range of least importance is as in the order above. In the same way we can track important features by lowering the silhouette score. We get table 4.5.

Removed columns	-	16	16, 18	16, 18, 17	16, 18, 17, 20
Silhouette score	0.1485	0.1118	0.1074	0.0929	0.0921

Table 4.5: Results of silhouette score on interesting columns

This indicates that columns *Offside*, (offside infringement) *Own Goal For* (own goal is scored by the opponent), *Own Goal Against* (own goal is scored), *Pressure* (pressure is applied to opposing player) are the most important columns (in that order). Moreover, these columns divide a player from being a defender instead of a midfielder.

Another approach that we use is the Laplacian score to verify the most and least interesting columns. We get table 4.6.

Columns	16	10	18	13	17
Laplacian score	7.9120e-09	1.0960e-05	0.0023	0.0032	0.0043

Table 4.6: Results of Laplacian score on interesting columns

Thus *Offside*, *Own Goal For* and *Own Goal Against* are again amongst the most important columns. Lastly, with the least important columns given by Laplace, we compare the dot product (correlation) between a K-means with the full dataset and a K-means with a dataset where we discard where 1 least important feature. We get table 4.7.

Columns	0	11	3	8	5
Laplacian score	0.0386	0.0331	0.0313	0.0302	0.0294
Correlation mean	0.0971	0.1090	0.0990	0.1006	0.1002

Table 4.7: Results of Laplacian score on least important columns

Chapter 5

Discussion and Conclusions

The main findings of this study were that the amount of 8 clusters best separate player types. With the use of silhouette analysis on K-means clustering, we saw that all other cluster sizes scored worse separability of different types of players. From this, we can say that given a soccer match, we classify each player performance as either one of the 8 types. Thus, there are only 8 types of players in a soccer game, in contrast to 11 or 4 player positions. We conclude that some positions within soccer are redundant or perhaps versatile.

From our Principal Component Analytics (PCA) visualizations, we see that goalkeepers are uniquely defined and do not correlate with other players in the field. PCA makes visual that the location of players tends to correlate with places of players in the clustering. However, we do not see this result back in the feature selection.

Some essential features we got from the Laplacian score and correlation mean that most define each cluster separability and deviance of player positions, are offside infringement and own goals scored. Offside infringement, on the other hand, is often associated with attackers that are too far forward. There is a smaller chance that any other position in the field is that much forward during a phase to perform an offside infringement. Own goals scored is a statistic that is often more attributed to defenders and goalkeepers. These positions tend to stay near their own goal and have more possibility in scoring an own goal. We can not directly attribute other notable features, such as own goal against and pressure, to any player position. This attribution might be interesting for further research.

On the contrary, columns such as duel, carry, and 50/50 are not crucial in defining cluster separability. This cruciality seems logical; any player can

perform these 3 features in the game: players can control the ball or try to recover the ball with another player. Each player tends to have some duels with other players in the game, however usually goalkeepers tend to have the least amount of duels and 50/50's. Interestingly, we can deduce that goalkeepers do have some events that are classified as such in the game. Otherwise, these columns would not be classified as not relevant.

We see that cluster sizes have about the same size according to the labels. Expectation Maximization tends to neglect evenly sized clusters, and this might have influenced why we saw that K-means outperformed Expectation Maximization. K-means was able to classify 66% of the data correctly when using 4 labels.

With the use of feature selection, we see that we can increase the performance of the K-means algorithm by 37%. This score indicates that feature selection is essential when fine-tuning a model, whether supervised or unsupervised. Prior studies also tell us this. The scientific conclusion is that fine-tuning parameters and feature selection play a significant role in changing machine learning algorithms to adapt to specific problem domains.

Another valuable insight is the one we saw in our preliminary data analysis in the data section: why is it the case that the best players do not get the ball more. You could argue that since they are defined as better based on their skills, having the ball more would benefit the team performance.

Further research includes inspecting outliers within each cluster. These outliers correspond to players that clustering models labelled as playing in 1 position when the data suggest they are playing in a different position. Soccer teams and coaches might benefit from placing these players in different positions to accommodate their behaviour more. Or, coaches can instruct these players to focus on performing more as their allocated position to streamline the performance of the team better.

References

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009. URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

[2] Deng Cai, Chiyuan Zhang, and Xiaofei He. "Unsupervised feature selection for multi-cluster data". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2010, pp. 333–342.

[3] Leonardo Cotta et al. "Using fifa soccer video game data for soccer analytics". In: *Workshop on large scale sports analytics*. 2016.

[4] Martin Crowder et al. "Dynamic modelling and prediction of English Football League matches for betting". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 51.2 (2002), pp. 157–168.

[5] Tom Decroos, Jan Van Haaren, and Jesse Davis. "Automatic discovery of tactics in spatio-temporal soccer match data". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 223–232.

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[7] Thomas MJ Fruchterman and Edward M Reingold. "Graph drawing by force-directed placement". In: *Software: Practice and experience* 21.11 (1991), pp. 1129–1164.

[8] Karan Gadiya. *FIFA 19 complete player dataset*. 2018. URL: https://www.kaggle.com/karangadiya/fifa19/.

[9] Xiaofei He, Deng Cai, and Partha Niyogi. "Laplacian score for feature selection". In: *Advances in neural information processing systems*. 2006, pp. 507–514.

[10] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[11] Michael Lewis. *Moneyball: The art of winning an unfair game*. WW Norton & Company, 2004.

[12] Jundong Li et al. "Feature selection: A data perspective". In: *ACM Computing Surveys (CSUR)* 50.6 (2018), p. 94.

[13] Jia Liu et al. "Automatic player detection, labeling and tracking in broadcast soccer video". In: *Pattern Recognition Letters* 30.2 (2009), pp. 103–113.

[14] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[15] Rehana Mahfuz, Zeinab Mourad, and Aly El Gamal. "Analyzing Sports Training Data with Machine Learning Techniques". In: *Methods* 9 (2016), pp. 0–038104976.

[16] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008, pp. 356–358.

[17] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.

[18] Pabitra Mitra, CA Murthy, and Sankar K. Pal. "Unsupervised feature selection using feature similarity". In: *IEEE transactions on pattern analysis and machine intelligence* 24.3 (2002), pp. 301–312.

[19] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.

[20] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[21] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[22] *Player FIFA database*. URL: https://sofifa.com/.

[23] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[24] Havard Rue and Oyvind Salvesen. "Prediction and retrospective analysis of soccer matches in a league". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 49.3 (2000), pp. 399–418.

[25] Jeffrey Alan Logan Snyder. "What actually wins soccer matches: Prediction of the 2011-2012 Premier League for fun and profit". In: *University of Washington* (2013).

[26] *Statsbomb Open Data*. URL: `https://statsbomb.com/resource-centre/#data`.

[27] Sam Tighe. *Breaking Down the 10 Most Popular Formations in World Football*. Oct. 2017. URL: `https://bleacherreport.com/articles/1417306-breaking-down-the-10-most-popular-formations-in-football#slide1`.

[28] Vincent Vercruyssen, Luc De Raedt, and Jesse Davis. "Qualitative spatial reasoning for soccer pass prediction". In: *CEUR Workshop Proceedings*. Vol. 1842. 2016.

[29] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `https://doi.org/10.1038/s41592-019-0686-2`.

[30] Hossam M Zawbaa et al. "Machine learning-based soccer video summarization system". In: *International Conference on Multimedia, Computer Graphics, and Broadcasting*. Springer. 2011, pp. 19–28.

Chapter 6

Appendix A: Event types - StatsBomb data

Event type	**Event description**
50/50	2 players challenging to recover a loose ball.
Bad Behaviour	When a player receives a card due to an infringement outside of play.
Ball Receipt	The receipt or intended receipt of a pass.
Ball Recovery	An attempt to recover a loose ball
Block	Blocking the ball by standing in its path.
Camera On	Signals the stop of the camera to capture gameplay for a replay/video cut.
Carry	A player controls the ball at their feet while moving or standing still.
Clearance	Action by a defending player to clear the danger without an intention to deliver it to a teammate.
Dispossessed	Player loses ball to an opponent as a result of being tackled by a defender without attempting a dribble
Dribble	An attempt by a player to beat an opponent
Dribbled Past	Player is dribbled past by an opponent.
Duel	A duel is an 50-50 contest between two players of opposing sides in the match.
Error	When a player is judged to make an on-the-ball mistake that leads to a shot on goal.
Foul Committed	Any infringement that is penalised as foul play by a referee.
Foul Won	A foul won is defined as where a player wins a free-kick or penalty for their team after being fouled by an opposing player.
Goal Keeper	Actions that can be done by the goalkeeper.
Half End	Signals the referee whistle to finish a match part.
Half Start	Signals referee whistle to start a match period.
Injury Stoppage	A stop in play due to an injury.
Interception	Preventing an opponent's pass from reaching their teammates by moving to the passing lane/reacting to intercept it.

Miscontrol	Player loses ball due to bad touch
Offside	Offside infringement. Cases resulting from a shot or clearance (non-pass).
Own Goal Against	An own goal scored against the team.
Own Goal For	An own goal scored for the team.
Pass	Ball is passed between teammates.
Player Off	A player goes/ is carried out of the pitch without a substitution.
Player On	A player returns to the pitch after a Player Off event.
Pressure	Applying pressure to an opposing player who is receiving, carrying or releasing the ball.
Referee Ball-Drop	Referee drops the ball to continue the game after an injury stoppage.
Shield	Player shields ball going out of bounds to prevent opponent from keeping it in play.
Shot	An attempt to score a goal, made with any (legal) part of the body.
Starting XI	Indicates the players in the starting 11, their position and the team's formation.
Substitution	Change of players in the game
Tactical Shift	Indicates a tactical shift made by the team shows the players' new positions and the team's new formation.

Chapter 7

Appendix B: Conversion of positions

General position	Specific position
Forwards	Center Forward, Right Center Forward, Left Center Forward, Secondary Striker, Right Wing, Left Wing
Midfielders	Center Defensive Midfield, Right Center Midfield, Left Center Midfield, Center Attacking Midfield, Right Defensive Midfield, Left Defensive Midfield, Left Midfield, Right Midfield Center, Midfield, Right Attacking Midfield, Left Attacking Midfield
Defenders	Left Center Back, Left Back, Right Center Back, Right Back, Center Back, Right Wing Back, Left Wing Back
Goalkeepers	Goalkeeper

General position	**Specific position**
Right Center Forward	Right Center Forward, Right Wing, Secondary Striker
Left Center Forward	Left Center Forward, Left Wing, Center Forward
Left Midfield	Left Midfield, Left Attacking Midfield, Left Defensive Midfield
Left Center Midfield	Left Center Midfield, Center Defensive Midfield, Center Attacking Midfield
Right Center Midfield	Right Center Midfield, Center Midfield
Right Midfield	Right Midfield, Right Defensive Midfield, Right Attacking Midfield
Left Back	Left Back, Left Wing Back
Left Center Back	Left Center Back, Center Back
Right Center Back	Right Center Back
Right Back	Right Back, Right Wing Back
Goalkeeper	Goalkeeper