MEF UNIVERSITY

FOOTBALL PLAYER PROFILING USING OPTA MATCH EVENT DATA: HIERARCHICAL CLUSTERING

Capstone Project

Uğurcan Kalenderoğlu

İSTANBUL, 2019

MEF UNIVERSITY

FOOTBALL PLAYER PROFILING USING OPTA MATCH EVENT DATA: HIERARCHICAL CLUSTERING

Capstone Project

Uğurcan Kalenderoğlu

Advisor: Asst. Prof. Dr. Utku Koç

İSTANBUL, 2019

MEF UNIVERSITY

Name of the project: Football Player Profiling Using Opta Match Event Data: Hierarchical Clustering
Name/Last Name of the Student: Uğurcan Kalenderoğlu
Date of Thesis Defense: 09/09/2019

I hereby state that the graduation project prepared by Uğurcan Kalenderoğlu has been completed under my supervision. I accept this work as a "Graduation Project".

09/09/2019
Asst. Prof. Dr. Utku Koç

I hereby state that I have examined this graduation project by Uğurcan Kalenderoğlu which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

09/09/2019
Prof. Dr. Özgür Özlük

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of Uğurcan Kalenderoğlu and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Dr. Utku Koç	……..
2. Prof. Dr. Özgür Özlük	……..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Uğurcan Kalenderoğlu 09/09/2019 Signature

EXECUTIVE SUMMARY

FOOTBALL PLAYER PROFILING USING OPTA MATCH EVENT DATA: HIERARCHICAL CLUSTERING

Uğurcan Kalenderoğlu

Advisor: Asst. Prof. Dr. Utku Koç

SEPTEMBER, 2019, 31 pages

Increasing popularity of data analytics has impacted the sport industry. Dimension of available data and best practices on the usage of data analytics increased as a result of this trend. Player profiling is one of emerging hot topics among those, especially in football. On the other hand, income and expense balance of transfers has been biggest burden on clubs' financials while it should be reverse. Scouting processes are currently dominated by bilateral relations and intuitive comments of scouting staff. It is an important step to transform into data driven decision framework to overcome this situation. It is crucial to replace a player who leave the team with someone who has potential and very close playing style. Player profiling is the first step to do this. The data set used in this project is obtained from Opta – a sport focused data company – and contains all actions performed on-ball at player level from Turkish Super League, English Premier League and German Bundesliga in three seasons between 2015 and 2018. Principal component analysis is applied to the dataset in order to reduce dimensionality to the 15 features which consists of 2469 players and 271 features at the beginning. As a result of this study, it is observed that there are twelve different player clusters within the traditional main positions; three for defenders, four for midfielders and five for forwards. Clubs can enrich and benefit from these clusters in three ways: 1) evaluation of a player style over a period of time and detecting the best role fit 2) analyzing the effect of cluster combination to decide which line-up yields better team results 3) finding the closest match to a player who is subject to replacement.

Key Words: unsupervised learning, hierarchical clustering, football data analysis, player profiling.

ÖZET

OPTA MAÇ VERİSİ KULANARAK FUTBOLCU PROFİLLEME: HİYERARŞİK KÜMELEME

Uğurcan Kalenderoğlu

Tez Danışmanı: Asst. Prof. Dr. Utku Koç

EYLÜL, 2019, 31 sayfa

Veri analitiğinin her alana hükmetmesiyle beraber futbolda da hem toplanan verinin boyutu hem de veri temelli yapılan iyi örneklerin sayısı artmaktadır. Futbolcu profilleme de bu alanlardan en revaçta olanlarından biridir. Kulüplerin finansal sağlığını koruması için transfer gelir gider dengesi en önemli kalem iken; genelde, izlenen yanlış transfer politikaları sonucu en büyük zarar kaynağı olarak dikkat çekmektedir. Bu alanda atılacak en önemli adımlardan biri ise şimdiye kadar geleneksel ve kişisel ilişkiler üzerinden gelişen futbolcu keşif süreçlerinin veri analitiğinden beslenen bir sürece evrilmesi olacaktır. Özellikle takımdan ayrılan bir yeteneğin yerine hem potansiyeli yüksek hem de oldukça benzer oyun stiline sahip adaylar bulmak için futbolcu profilleme doğru bir başlangıç adımı olacaktır. Bu projede, Opta ismindeki spor odaklı veri şirketinin 2015-2018 arasındaki üç sezonda Türkiye Süper Ligi, Almanya Bundesliga ve İngiltere Premier Ligi'ni kapsayan ve topla yapılan tüm hamleleri içeren veri seti kullanılmıştır. 2469 futbolcunun oynadığı tüm maçları içeren ve 271 öznitelik bulunan veri seti, temel bileşen analizi kullanılarak 15 özniteliğe indirgenip hiyerarşik kümeleme algoritması kullanılmıştır. Çalışma sonucunda, ana pozisyonlardaki farklı oyun stillerini temsilen; defans için üç, orta saha için dört, forvet içinse beş olmak üzere toplamda on iki farklı oyuncu kümesi olduğu gözlenmiştir. Kulüpler bu kümelerden üç farklı şekilde faydalanabilir: 1) mevcut oyuncunun yıllar içinde evrildiği roller ve oyuncuya en uygun rolün tespiti 2) farklı oyuncu küme kombinasyonlarının maç sonuçlarına etki analizi sonucu en verimli ilk on birin belirlenmesi 3) transfere konu oyuncuya stil veya rol olarak en yakın adayın bulunması.

Anahtar Kelimeler: güdümsüz öğrenme, hiyerarşik kümeleme, futbol veri analitiği, oyuncu profilleme.

TABLE OF CONTENTS

.

1. INTRODUCTION

Player profiling is an emerging topic in football industry. It can be utilized during the pre-match analysis where players and technical staff discuss how to attack and defend the opposing team in advance. However, its best use would be during scouting and player recruitment processes. The objective of this study is to develop an unsupervised machine learning model that clusters the players based on the playing style by utilizing Opta match event data which is comprised of all on-ball actions performed during the match time. Opta employs three operators to log and check on-ball events for one game. Examples of on-ball events are goals, assists, successful passes and aerial duels lost.

Scouting in football is a very traditional field that is still managed by the managers. Bilateral relations and intuitive comments of scouting staffs dominate the hiring decisions of clubs. Moreover, most of the transfers do not pay off enough and the clubs face financial crisis. Then it becomes easier to spend lots of money to acquire new player in order to reverse the unsuccessful outcomes and gain the financial health back. In contrast, quick and intuitive decisions result in low-performing teams that yield ultimately worse financial results. On the other hand, rising platforms which produce event or image-based football data empowered us to analyze teams and players in a more objective way. It guides the technical and managerial staff through more data driven recruitment processes to attract the best talents. 'The best talent' refers to fitting to the team as much as possible instead of pure individual performance (Decroos et al., 2018). Therefore, traditional positions such as goalkeeper, defender, midfielder or forward are not enough anymore. One needs to develop robust models which cluster the players who have similar type of playing style. Clubs can find feasible alternatives for their needs considering all limitations and expectations, using more detailed clusters.

Metrics which are currently used by football authorities rely heavily on generic and popular ones such as goals, assists, and shots. However, this limits the insights which can be extracted from the data (Decroos et al., 2018). If we evaluate the performance of forward player based on only number of goals he concedes then we would ignore the contribution of player who enabled his teammate to score more than 20 goals thanks to space he created by repressing the defenders of the opposing team. This study aims to

include balanced set of features in order to represent each cluster of playing style that exist in real life.

Finding the right set of clusters of playing styles is not only useful for scouting. It can also guide the coaches when they try to find the best line up to start a match. Ingersoll et al. (2017) presents that the more diverse the team is in terms of culture, the better outcomes they get in UEFA Champions League. The same may apply to the teams which are set up by considering balanced playing styles that are present on the pitch at the same time. According to the recent study by Ven (2018), if the right types of players are mixed to play together, team performance can improve as they cover each other's weaknesses. The combination of two different defenders, one is strong and slow while other one is weak but fast, may yield a better match result for the team (Ven, 2018).

The limited number of studies conducted on football player clustering is due to its complex nature. Football is a complicated sport that valuing an action is hard since goals happen rarely and it is not clear that which action is the most crucial one in scoring a goal. Therefore, most of the academic research in sport analytics are based on the data from baseball, tennis or American football where individual performance is less dependent to other factors and outcomes are clearer (Kerr, 2015).

This study contributes to the existing literature by extending the domain to the football and using the actual player match statistics rather than ratings of player attributes which are given by experts as Kerr (2015) used.

The rest of the paper is organized as follows: The first section that gives introductory background on the domain of the study and brief literature review. Section 2 defines the problem, gives the objectives of the study, and ends up with the project scope. Then, Section 3 summarizes methods, tools and techniques used in the study. Details of data and source are also covered in the same section. Section 4 is dedicated to exploratory data analysis including the descriptive statistics and explanation of the features. Section 5 gives details of dimensionality reduction techniques and review of clustering algorithms applied on the dataset. Finally, Section 6 concludes the study by providing results and insights.

2. PROJECT DEFINITION

In this section, we start with the definition of the problem that we try to propose a solution. Then, we give brief objective and scope of the project.

2.1. Problem Definition

Assessing the player performance is highly dependent on the most visible features such as goals, passes or shoots. Moreover, football is a team play and individual success may not mean high contribution all the time. There can be some players who have better statistics for his position but decrease the total team contribution because of self-focused playing style. Therefore, recruitment departments (or scouting functions) should evaluate the performance of potential candidate compared to players who have similar playing style. Otherwise, transfer would not guarantee the success of the team since pure individual statistics is not an objective way of measuring the real contribution to the team. However, there is no widely accepted objective way of clustering football players although the need is quite clear.

2.2. Project Objective

This project aims to develop a reliable unsupervised clustering model that reveals the sub-positions beyond the traditional ones and assign each player to the most relevant cluster. Such list can be used by scouting teams when they are supposed to find a good fit (e.g. for replacement of a specific player). Evaluation phase can be more objective since categorization is done by an algorithm which is developed by real event data and players are compared with equivalent candidates who have similar playing styles.

2.3. Project Scope

There are four main positions in football: goalkeeper, defender, midfielder and forward. We focus on: defender, midfielder and forward players and exclude goalkeepers since feature set is not rich enough to represent differences among all profiles of them. It is important to note that the model does not assess the performance of players or evaluate the future potential of a candidate. It only does select the top features which explain most of the variance and clusters the players who can be grouped in terms of playing characteristic.

Some players can play at different roles in different matches, but it is assumed that sum of whole season is not affected by those cases.

The style and cluster of players are surely influenced by their teammates who play with closely or the dynamics of the opposition team. This effect is not covered within the scope of this project. Liu et al. (2015) presents a very useful study on this point. They even show that the strength of the opposition team directly affects the playing style of a team and it requires more tactical and technical performance if game is played against, relatively stronger teams (Liu et al., 2015).

3. METHODS

In this section, we give the details of methods and algorithms used in the study. Aim is to clarify the underlying logic behind the algorithms by giving mathematical equations or rule sets.

3.1. Principal Component Analysis

Principal component analysis (PCA) is used to reduce the dimensionality of the dataset by finding the eigenvector decomposition of features where most of the variance lies on. It also helps to control the computational complexity by reducing number of features used in the model.

PCA transforms the existing features and gives reduced set of new columns which explains most of the variance among samples. It is the creation of new coordinate system that actual data points are projected onto. Explained variance is greatest at first principal component variance and it decreases as we go through the last principal component.

Assume that there exists matrix X which includes n rows representing the samples and p columns for a feature set. Existing set of p-dimensional vectors $\mathbf{w}_{(k)} = (w_1,\ldots, w_p)_{(k)}$ is replaced by new vector $\mathbf{t}_{(i)} = (w_1,\ldots, w_l)_{(i)}$ which is given by:

$$t_{k(i)} = x_{(i)} \cdot w_{(k)} \quad \text{for} \quad i = 1,\ldots,n \qquad k = 1,\ldots,l$$

usually satisfying l is less than p and each component of the l-dimensional vector of $\mathbf{t}$ explains the maximum possible variance on the original dataset.

The transformed set of new feature space is accepted to be uncorrelated over samples. It does not mean that all components should be included in the model, first L principal components may be enough to explain the desired level of variance. These are eigenvectors, which are obtained by

$$\mathbf{T}_L = \mathbf{X}\mathbf{W}_L$$

which is n x L matrix.

3.2. Unsupervised Learning: Clustering

Clustering can be thought as a classification of unlabeled data. It groups the very similar samples in a meaningful way when there is no target variable. Frequently used concept behind the clustering algorithms is to minimize the within-group-variance among

samples which belong to the same cluster while maximizing the between-group-variance as much as possible. However, deciding the number of clusters and which measure to use for evaluation of the model is not easy as there is no standard way of that. Interpretation of the results is also another challenge which requires intense domain knowledge. Therefore, a lot of approaches are developed to cluster datasets in different domains. We use the following three clustering algorithms that are suitable for the problem.

3.2.1 K-means Clustering

K-means is a simple clustering algorithm based on the centroid approach that tries to find center points for the clusters first, and then assigns each point to the relevant cluster by checking distance from points to those centroids.

In the case of *n* x *d* matrix that *n* is the number of samples and *d* is the number of features, objective function of the k-means clustering algorithm is to minimize the total of within-cluster variance. Formulation of the objective can be described as follows:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where S_i represents subset of observations, $\boldsymbol{\mu}_i$ is average value of points in S_i, k is number of observations in the S_i and $\|\mathbf{x} - \boldsymbol{\mu}_i\|^2$ is euclidean distance between points. The steps of the algorithms are as follows:

- Step 1: Randomly choosing K points of centroids as an initial cluster centers
- Step 2: **repeat:**
- Step 3: assign each point to the closest cluster centroid by measuring euclidean distance between point and cluster center
- Step 4: calculate the new center of each cluster formed by taking mean of cluster points
- Step 5: **until** centroids remain stable (Tan, Steinbach, Karpatne, & Kumar, 2018, p.497).

3.2.2 Hierarchical Clustering

Hierarchical clustering is a type of clustering algorithm that assumes that there is a hierarchy between clusters. Each cluster is a subset of another big cluster except the top hierarchy which is a single cluster including all points. There are two types of hierarchical clustering algorithms described by Tan et al. (2005):

- Agglomerative: Starts by behaving each point as a separate clusters and then merging closest pairs.
- Divisive: Starts with one single cluster and continue by splitting clusters at each step.

The decision criteria to merge clusters in agglomerative approach or decompose in divisive type is based on a dissimilarity measurement. This measurement requires both a distance metric and a linkage criteria. Two most common distance metrics are:

- Euclidean distance:

$$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

- Manhattan distance:

$$\|a-b\|_1 = \sum_i |a_i - b_i|$$

Each metric is a different way of measuring distance between two different points. Euclidean is the most commonly used one.

Three most common linkage criteria are:

- Complete linkage (maximum): calculate maximum distance between two clusters

$$\max\{\, d(a,b) : a \in A,\, b \in B \,\}$$

- Single linkage (minimum): calculate minimum distance by using closest points in two clusters

$$\min\{\, d(a,b) : a \in A,\, b \in B \,\}$$

- Average linkage: calculate average of sum of distances between all points in two clusters

$$\frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a,b)$$

Each linkage criteria is a measurement of distance between two sets of observation (A and B), clusters in this case.

3.2.3 Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

HDBSCAN is a density based hierarchical clustering algorithm which is developed as a response to limitations and challenges of the previous algorithms. We should have pre-

defined number of clusters in k-means, or heuristic approach to cut the tree in hierarchical clustering. HDBSCAN formulates the problem to optimize the identification of significant clusters by global optimal solution rather than local (Campello et al., 2013).

Campello et al. (2013) defines the main steps of the HDBSCAN algorithm as:

- Step 1: calculate the distance of all elements in **X** to the m_{pts}, where **X** is set of n points and m_{pts} denotes minimum required points to form cluster
- Step 2: minimum spanning tree is calculated (for G_{mpts} - which stands for complete graph which shows distance to the relevant set of object)
- Step 3: Add self edges to the existing trees including the weight associated
- Step 4: Construct a dendogram – tree based diagram to show cluster merges - of current hierarchy
    - Step 4.1: Start with single cluster including all points
    - Step 4.2: Start with highest weighted edge and remove iteratively

It defines and use new distance metric, *mutual reachability distance*. It is calculated as below:

$$d_{\text{mreach}-k}(a,b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a,b)\}$$

where core_k denotes distance between point and its nearest neighbor while d(a,b) is distance between points a and b.

4. EXPLORATORY DATA ANALYSIS

Opta provided full dataset of 3 seasons for Turkey Super League, German Bundesliga and English Premier League. They record a very detailed set of actions of each player for each game played. Granularity level is as follows: Gomis played 80 min against Fenerbahce, conducted 3 shots - only one on target -, have 5 successful and 4 unsuccessful passes in the opposition's half.

4.1. Data

There are two different csv files for each league per season. The first csv files contains statistics of players per each game their team played while second file only contains only sum of team statistics for each game. It would be good to go over the file which contains player statistics for Turkish Super League's 2017-18 season. The dataset consists of 8534 rows (observations) and 271 columns (features). As you can see from the screenshot below, each row represents the statistics of one player for unique game played in each season. For example, row which is highlighted with red box says that "Abdoul Sissoko which is player of Akhisarspor received 44 passes and conducted one shoot with his left foot with 0% accuracy during the game played against Antalyaspor and Akhisarspor was the home team". There are also unique ids for each player and for each match played.

```
In [4]: events.head()
Out[4]:
```

	Date	Match Id	Player ID	Player Surname	Player Forename	Team	Team Id	Opposition	Opposition id	Venue	...	Passes Received	Left Foot Shots	Shooting Accuracy Left Foot	Right Foot Shots	Shooting Accuracy Right Foot	
0	13/08/2017	935593	33160	Abdou Traoré	NaN	Konyaspor	2141	Trabzonspor	383	Away	...	13	0	0.00%	0	0.00%	
1	08/04/2018	935835	55947	Abdoul Sissoko	NaN	Akhisarspor	6796	Alanyaspor	3324	Away	...	9	0	0.00%	0	0.00%	
2	28/01/2018	935750	55947	Abdoul Sissoko	NaN	Akhisarspor	6796	Antalyaspor	381	Home	...	44	1	0.00%	0	0.00%	
3	16/09/2017	935623	55947	Abdoul Sissoko	NaN	Akhisarspor	6796	Kardemir Karabükspor	3073	Home	...	20	0	0.00%	0	0.00%	
4	29/09/2017	935646	55947	Abdoul Sissoko	NaN	Akhisarspor	6796	Fenerbahçe	253	Home	...	31	0	0.00%	0	0.00%	

Figure 1: Representation of first 5 rows of the dataset

4.2. Pre-processing

We focus on defenders, midfielders and forward players, so goalkeeper are eliminated. In order to do that, we should keep only rows which have position id greater than 1 which correspond the positions except goalkeeper. Figure 2 shows that number of players in Turkish Super League by total time they played in three seasons. There are a

very few players who played more than 8000 minutes. It is also worth noting that a lot of footballers played less than 1000 minutes in sum of three seasons.

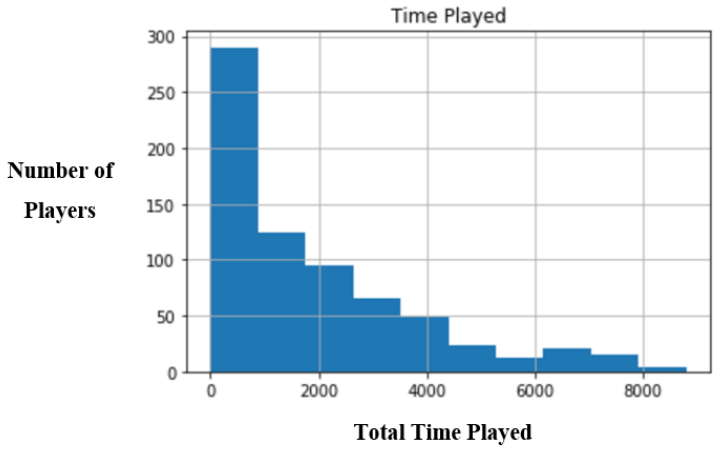

Figure 2: Number of players by time played (minutes)

Figure 3 shows average time played per game versus the number of players in each time interval. It is interesting to have a lot of players whose average game time is greater than 80 minutes. It is probably due to having some players who played just a few games thus average time per game is at such a high level.

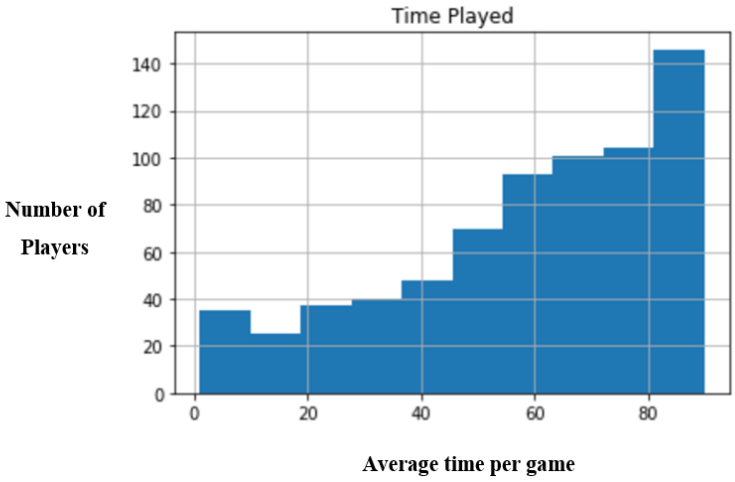

Figure 3: Number of players by avg time per game (minutes)

We excluded a player if he plays less than 20 minutes in a match since it can manipulate the overall results due to limited actions performed. It takes time to get adopted for the ball and other players on the pitch. Therefore some matches are excluded if footballer played less than 20 minutes. The next step is to sum all stats by players and having one row for each player. In order to do that, we should drop the unnecessary

variables and the ones which are not suitable to sum up after group by operation. Those of features which do not add value to the model and not suitable for sum up operation such as:

- Date: date of game
- Match id: unique 6 digit game id
- Team: team name
- Team id: team id number
- Venue: home or away

are dropped. As a result, we have 259 features left including ‘Player ID’, ‘Player Surname’, ‘Player Forename’ and ‘Position Id’ that we need in further steps.

There are a lot of footballers who played less than 1000 minutes in three season as we know from Figure 2 and Figure 3. It means they played less than four games in one season considering one game takes 90 minutes. This is insufficient amount of time to claim that we have enough actions to cluster the style of player. Hence the players who played more than 1000 minutes are only included for the rest of the study.

As we can see from the box plot below (Figure 4), there are outliers for total time on the pitch per player. So we need to convert all variables to the metric which is 'per 90 min.'. Hence we can compare the player who played 2000 minutes and 6000 minutes in a more objective way.

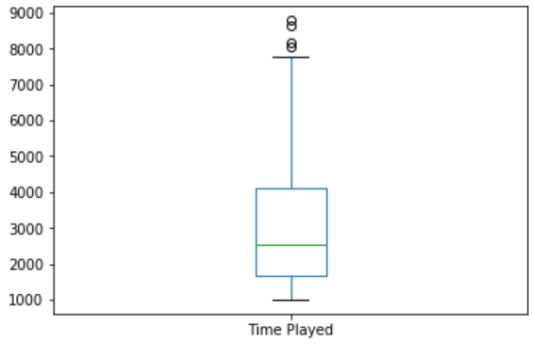

Figure 4: Box plot of total times played in minutes

Measurement of features is different. So, we should scale them in order to prevent the model to treat the biggest feature as a most important one. For instance, goals are in the 0-50 range while key passes range is from 0 to 182. If we model the dataset without scaling, it can behave key passes as a more important feature than goals. On the other hand, if there is a player who has unique skills, we would not want to lose that

differentiation. We can have some clusters only a limited set of players belong to it. Thus MinMaxScaler can be used there. Figure 5 represents first five rows of dataset after MinMaxScaler is applied.

```
#MinMaxScaler is better when dealing with outliers and losing them is not desired
from sklearn.preprocessing import MinMaxScaler
transformer = MinMaxScaler().fit(pgs_STSL_2)
np_array = transformer.transform(pgs_STSL_2)
pgs_scaled = pd.DataFrame(np_array)
```

```
pgs_scaled.head()
```

	0	1	2	3	4	5	6	7	8	9	...	234	235	236	237	238	239	240
0	1.000000	0.684211	1.000000	0.936306	0.742138	0.555556	0.676471	0.382353	0.272727	0.857143	...	0.0	0.0	0.0	0.069318	0.646465	0.803213	0.0
1	0.444444	0.342105	0.653846	0.420382	0.352201	0.228395	0.205882	0.088235	0.272727	0.000000	...	0.0	0.0	0.0	0.218409	0.636364	0.281124	0.0
2	0.533333	0.526316	0.538462	0.484076	0.559748	0.104938	0.088235	0.088235	0.000000	0.000000	...	0.0	0.0	0.0	0.000000	0.141414	0.477912	0.0
3	0.477778	0.473684	0.692308	0.547771	0.440252	0.246914	0.529412	0.529412	0.000000	0.000000	...	0.0	0.0	0.0	0.341136	0.141414	0.477912	0.0
4	0.600000	0.394737	0.884615	0.515924	0.396226	0.209877	0.117647	0.117647	0.000000	0.000000	...	0.0	0.0	0.0	0.586818	0.151515	0.461847	0.0

5 rows × 244 columns

Figure 5: Data Normalization

4.3. Dimensionality Reduction: PCA

We have 244 features after elimination of non-value added ones and PCA can help us to reduce dimensionality in order to have more robust clustering model.

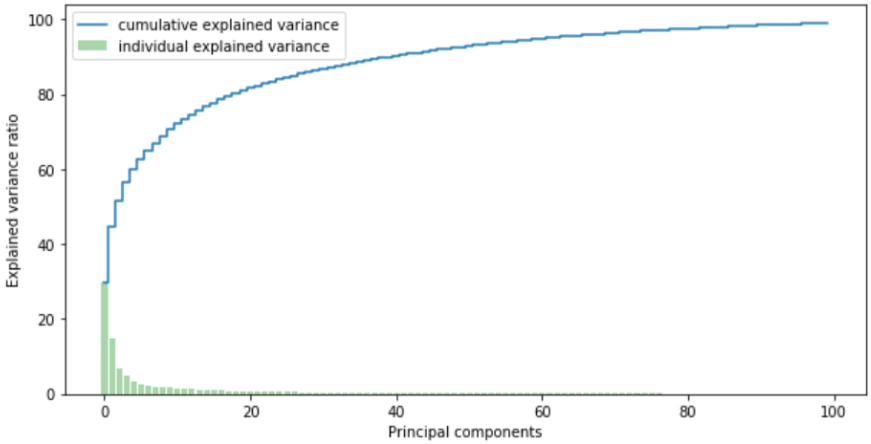

Figure 6: Graph of PCA – explained variance chart

As we can see in Figure 6, the first fifteen principal components explain the 76.2% of the variance among samples. At this point, it would be good to get top features which are highly correlated with PCA components for only ones we selected in previous step.

PC1 is the first principal component which explains 31.5% of the variance by itself. If we calculate the correlation coefficient between PC1 and all remaining features, it turns out that following three features have strong relationship with it:

- Attempts open play on target: total number of attempts from open play on target
- Shots on target inc goals: total number of shots on target (saved), including goals
- Attacking ground duels lost: total number of ground duels lost, including; freekicks for dangerous play and fouls, take on lost, challenge lost and dispossessed, within the attacking half

It is highly correlated with specifications of attacking players and does not differ much among leagues as can be seen in Table 1a, 1b and 1c. It may be logical to combine all leagues and making clusters for whole dataset after checking PC2 -second principal component- as well.

CorrelatedFeatures	Correlation coefficient
Attempts Open Play on target	0.900917
Shots On Target inc goals	0.88504
Unsuccessful Ball Touch	0.882331
Attacking Ground Duels Lost	0.87735
Touches open play final third	0.873889

Table 1a: PC1 for Turkish STSL

CorrelatedFeatures	Correlation coefficient
Attempts Open Play on target	0.889207
Attacking Ground Duels Lost	0.877504
Unsuccessful Ball Touch	0.872558
Shots On Target inc goals	0.871922
Touches open play final third	0.866002

Table 1b: PC1 for Premier League

CorrelatedFeatures	Correlation coefficient
Attempts Open Play on target	0.894729
Attacking Ground Duels Lost	0.883065
Touches open play final third	0.88027
Shots On Target inc goals	0.877855
Attempts Open Play off target	0.863364

Table 1c: PC1 for Bundesliga

The second principal component is highly correlated with features related to corners as illustrated by Table 2. If a player is highly active in using corners then we would expect those features to be higher than average.

It seems that all three leagues have similar dynamics in terms of explaining variance in playing styles although players are different. There is almost no difference among leagues except slight change in sorting. Therefore, we can combine confidently the dataset for three leagues.

STSL	
Correlated Features	Correlation coefficient
Unsuccessful Crosses Corners in the air	0.83182
Unsuccessful Crosses Corners	0.822085
Successful Crosses Corners	0.814943
Successful Crosses Corners in the air	0.812785
Unsuccessful crosses in the air	0.770773
EPL	
Correlated Features	Correlation coefficient
Successful Crosses Corners	0.82264
Successful Crosses Corners in the air	0.818628
Unsuccessful Crosses Corners in the air	0.809824
Unsuccessful Crosses Corners	0.791571
Successful crosses in the air	0.73209
Bundesliga	
Correlated Features	Correlation coefficient
Successful Crosses Corners in the air	0.828173
Successful Crosses Corners	0.826603
Unsuccessful Crosses Corners in the air	0.826319
Unsuccessful Crosses Corners	0.818586
Successful crosses in the air	0.748699

Table 2: PC2 for all leagues

4.4. Model Selection

In order to find the suitable model, we decided to test the performance of clustering algorithms. We transformed the clustering problem into classification by using already existing main position ids. K-means clustering, hierarchical clustering and HDBSCAN

algorithms were applied to the dataset and performance was evaluated based on the accuracy.

All models are deployed on the same dataset. We selected the most reliable players as a reference to label the outcome of clusters and then used them to evaluate the accuracy of models. The reason for doing that is the output of clustering is not meaningful by itself. We need to take some players as a reference and label all players belong to same cluster. A total of 24 players used as reference and player are selected based on the following criteria:

- Taking central role in his position since they may represent the characteristics of position better than who play as wingers
- Play at the same position in almost all matches
- Well known player by the audience

Hierarchical clustering outperformed both k-means and HDBSCAN algorithms. It can be observed from Figure 7, accuracy is 63.3% for hierarchical clustering while it is 53.5% for k-means and 54.6% for HDBSCAN. Hierarchical clustering has also some other advantages: it is stable as you run algorithm multiple times, it is flexible since you can define number of clusters based on the defined metrics and domain knowledge, lastly it is more suitable for uniformly distributed data which is the case for us because data points do not have clear boundaries to form clusters. All these reasons are enough to choose hierarchical clustering in order to perform position specific player profiling.

```
Accuracy scores for each model are as follows:
k-means clustering:  0.535
hierarchical clustering:  0.633
HDBSCAN:  0.546
```

Figure 7: model comparison - accuracy

Below plots represent how data points are distributed based on actual classes and labelled clusters, in both Figure 8 and Figure 9. X-axis is PC1, y-axis is PC2, red points belong to defenders, while greens are midfielders and blues are forwards. It is clear that algorithm works well when distinguishing defensive and attacking styles but label some defenders as midfielders and some midfielders as forward. We observed that most of misclassified defenders are wing backs who mostly support attacks and thus have similar statistics as midfielders in some features. Consequently, having a set of misclassified defenders supports our objective that different player clusters exist within each position.

We expect to have them grouped with midfielders since their style is closer to midfielders than defenders. Same applies for midfielders who grouped with forwards.

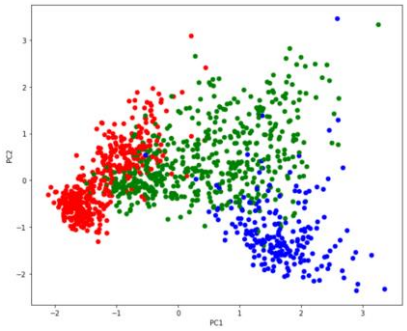

Figure 8: cluster of actual positions

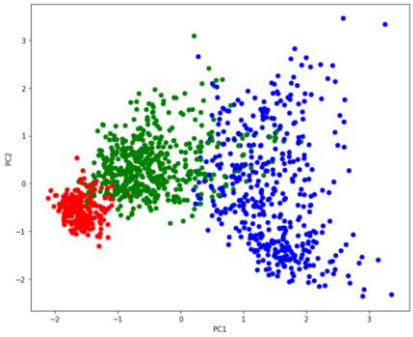

Figure 9: cluster of algorithm output

We can also check the other performance evaluation metrics of hierarchical clustering for each position separately. Table 3 shows the accuracy, precision, recall and F1-score for each position. Calculations are made considering following assumptions:

- true positive: number of actual defenders who also clustered as defender
- false positive: number of players who are clustered as defender but play as midfielder or forward in actual life
- true negative: number of actual midfielders or forwards who also clustered as 'not-defender'
- false negative: number of actual defenders who clustered as 'not-defender'. The same applies for midfielder and forward as well.

Accuracy is above 65% for all and high as 84% for defenders. Subset of PCA components we used for clustering are highly correlated with the features related to attacking. For this reason, it is expected to have high accuracy in defender and forward and lower accuracy in midfielder. On the other hand, some defenders have offensive style of playing and it is possible to classify them as midfielder. Therefore, recall is low for defenders and precision is low for midfielders. Highest recall is obtained in forward position since it is less likely to have an example close to midfielder or defender in this position.

Metric/Position	Defender	Midfielder	Forward
Accuracy	0.84	0.65	0.81
Precision	0.99	0.62	0.47
Recall	0.59	0.57	0.99
F1 score	0.74	0.60	0.64

Table 3: Evaluation metrics of hierarchical clustering

5. RESULTS

We applied PCA one more time to each position separately in order to cluster them in a more robust way since selected features will be more related to the position itself. This resulted in more diversified set of features which enabled us to find the position specific features and clusters. We know that hierarchical clustering is better in clustering for football positions so we use it from now on.

Clustering is a science but interpreting results is an art. Especially deciding the number of clusters requires intense domain knowledge most of the time. There are some metrics such as dendrogram, silhouette and elbow but none of them are deterministic or able to find global optimum value of clusters. All use heuristic approaches and help us to compare alternatives based on distance measures. Therefore we benefited from different sources when deciding number of clusters and explaining the characteristics of players. Scisports' article (Aalbers, 2019) guided us through naming each cluster. We also benefited from 'Understanding roles in Football Manager' (Tactics and Etc., 2018) and 'Football Roles Explained' (Conditional Love, 2013) posts to get insight about unique attributes of clusters and model players given as an example. Finally, we compared the cluster centers and highlighted top features that difference between cluster centers is significant.

First fifteen principal components were used as input to hierarchical clustering after dataset is restricted to only defenders, midfielders and forwards. Explained variances by these components for each position are as follows:

- Defenders: 82%
- Midfielders: 83%
- Forwards: 77%

5.1. Clustering Defenders

One of main advantage of hierarchical clustering is that you do not have to know how many clusters should exists in advance, which is not case for k-means clustering. Dendrogram helps us to decide number of clusters. It is the tree that shows each sample as a leaf at the bottom and groups similar leaves as you move up. So, the top of the tree is single branch which represents all samples as one cluster. The height of the branches,

length of vertical axis, is proportional to the dissimilarity of two clusters fused at that branch. The longer the distance, the more dissimilar the clusters are. So, it would be preferred to cut the dendrogram at the point where marginal gain in terms of increasing dissimilarity begins too low. Alternatively, Hees (2015) proposed the automated cut-off selection to cut the dendrogram. We did not use it as he stated that it is not a good idea to rely only on this method and manual selection is preferable.

Dendrogram for defenders shown in Figure 10 implies that cutting the tree around height of 15 would give most diversified clusters. Silhouette score also drops drastically after 3 clusters as can be seen in Table 4.

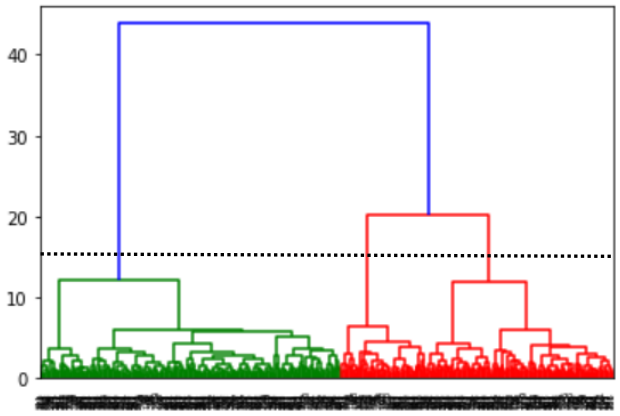

Figure 10: Dendrogram for defenders

# of clusters	Silhouette score
2	0.312
3	0.270
4	0.169
5	0.145
6	0.142

Table 4: Silhouette score for defenders

Observing the clusters, we identified three classes for each defenders: stoppers, wing backs and defensive backs.

Stoppers (240 players): Their main duty is stopping the attacking players of opposing team and winning the ball. They do not contribute much to attacks but get highly engaged role in defensive minutes. Examples are Abdoulaye Ba, Michael Dawson, Steve Cook and Martin Skrtel.

Wing backs (67 players): This role plays very active role in attacks although it is defensive position. He runs near the line, attempts to cross, use corners, gets passes

frequently and give high number of short passes. Some of them might also try shoots. Examples are Caner Erkin, Ozgur Cek, Omer Bayram and David Alaba.

Defensive backs (149 players): They play at wings like wing backs but do not take active role in attacks. Their main focus is stopping opposed wingers making crosses into penalty box. Examples are Gokhan Gonul, Mauricio Isla and Philipp Lahm.

5.2. Clustering Midfielders

Midfielders includes wingers and main distinction is expected to occur between central midfielder roles and winger roles. Dendrogram in Figure 11 shows having 4 clusters would be suitable for this position but silhouette score (Table 5) drops as number of cluster increases. We know that there are at least four distinct clusters in midfielder from the sources mentioned above although algorithm and dataset can not differentiate them well.

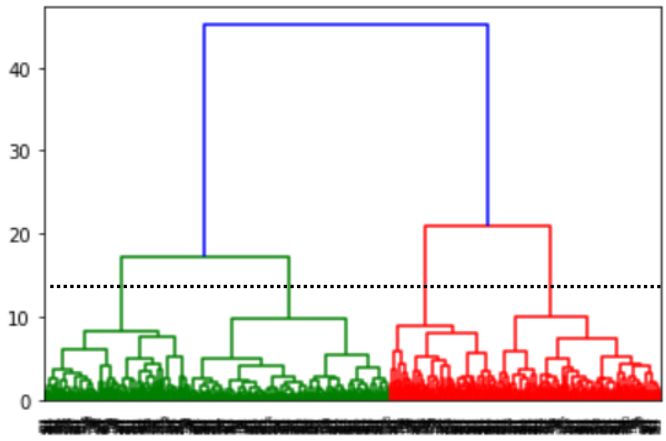

Figure 11: Dendrogram for midfielders

# of clusters	Silhouette score
2	0.430
3	0.363
4	0.314
5	0.261
6	0.219

Table 5: Silhouette score for midfielders

Defensive midfielders (157 players): Players belonging to this group are expected to apply pressure to the ball when it is on opposition. Additionally, they cover wing backs when they are in the attacking duty by positioning themselves between stoppers. Most of them can play as centre back/stopper as well. Examples are Mehmet Topal, Ryan Donk, Mathieu Flamini and Michael Carrick.

Centre midfielders (88 players): These players are bridges between the defense and attack. So they may include different profiles but most of them are assumed to have balanced skills at defensive, supportive and attacking duties. They have high ball touches, pass received and successful passes to teammates statistic per game. Examples are Aaron Ramsey, Paul Pogba and Jack Wilshere.

Set Piecers/Wingers (80 players): They play wide and mostly focused on attacks. Their dribbling skill is higher than players in other clusters and they are fast to create the goal chances without enabling opposing defense to take position. Most of the team crosses are made by players in this cluster. There are also some players who lead the set pieces included in this cluster as well. Examples are David Silva, Arjen Robben and James Rodriguez.

Advanced Playmakers (212 players): The last midfielder cluster are brains behind the attacks while their role in defense is weak. Their position as a location can be attacking midfielder, winger or centre midfielders but their role and playing style differ most of players in the same location. Some of them play behind the centre forwards. They are good at first touch and can try shoots frequently. Examples are Henrikh Mkhitaryan, Kagawa, Franck Ribery and Kevin-Prince Boateng.

5.3. Clustering Forwards

Dendrogram implies that there would be five clusters under the forward position as shown in Figure 12. Silhouette score also rises when number of cluster is five (Table 6).

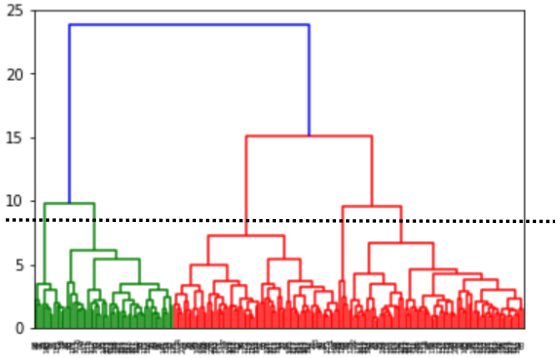

Figure 12: Dendrogram for forwards

# of clusters	Silhouette score
2	0.305
3	0.218
4	0.224
5	0.229
6	0.207

Table 6: Silhouette score for forwards

Winger Forwards (14 players): They play wide like winger midfielders but can also position themselves in the opposing penalty box. If their team is attacking and ball is on the other side of the pitch, they generally threat the opposing defenders and seek a chance to score goal. Examples are Alexis Sanchez, Philippe Coutinho, Dimitri Payet and Memphis Depay.

Inside Forwards (73 players): They are freer versions of attacking midfielders. They play closely with centre forwards and support them by passes. On the other hand, they do not prefer to shoot on from inside box. Examples are Sadio Mane, Raheem Sterling and Jordan Ayew.

Target Men (69 players): This is the hardest group for opposing defenders. They play very close to opposing centre backs and good at taking the ball from pressured area to a place where scoring is more likely. They also pose threat in the aerial positions especially from set plays. Examples are Peter Crouch, Fernando Llorente, Umut Bulut and Laudio Pizarro.

Advanced Forwards (13 players): This is the smallest cluster in the study. This is the ‘wait and capture the moment when opposing defenders cannot catch you’ position. Advanced forwards are good at escaping from opposing defenders and having one-to-one

position with opposing goalkeeper. So highest number of goals are scored by players in this cluster. Examples are Mario Gomez, Harry Kane, Sergio Agüero and Robert Lewandowski.

Shoters/Set Players (45 players): They use every chance to shoot and have higher goals from set plays than other players. Examples Daniel Sturridge, Lukas Podolski, Romelu Lukaku, Moussa Sow and Jamie Vardy.

5.4. Evaluation of Clustering Whole Players

We have twelve clusters in total. In the beginning, we split the dataset into three clusters since we know that there are three main positions after excluding goalkeepers. However, we now know that there are more clusters than three. So we can check that how setting number of cluster equal to twelve or less affects the performance metric. We used thirty-five features which explains 90% of the variance in the dataset. As you can see from the table below, silhouette score increases when number of cluster is five and then continues to drop as each new cluster added.

# of clusters	Silhouette score
2	0.265
3	0.246
4	0.159
5	0.177
6	0.176
7	0.168
8	0.150
9	0.145
10	0.145
11	0.133
12	0.130

Table 7: Silhouette score for all players excl. goalkeepers

This trend in silhouette score supports our argument in Section 4.4. The defenders who play as a wing backs has different playing style than other defenders so algorithm group them with midfielders. Same applies for midfielders who play as a set piecer/winger but grouped with forwards. This is also reason to having low recall for defenders and low precision for forwards given in Table 1. These two positions are transition roles between defender-midfielder and midfielder-forward so entail grouping separately.

6. CONCLUSION

For football positions, the mentality of ‘defenders are only responsible for defense and forwards for scoring’ lost its reputation and simple position information is not enough anymore. Having new and data-driven methods to cluster players for each position is beneficial in multiple aspects: 1) Evolution of the players and development or change in style can be analyzed using this point of view. For example, there may be a player who fits a better to the role of defensive back rather than wing back. Positioning and directing him in such a role may increase his contribution to the team. 2) Managers can also improve the team line-ups by analyzing effect of role combinations in the team. If having two advanced playmakers yields better results than having two central midfielders on the pitch, line-up and transfer decisions can be adjusted accordingly. 3) Replacement decisions can be made more easily by using scouting databases which store style (cluster in this study) information of player and similarity index of all players who have close playing style (in the same cluster for this study) to the one that team looks to replace. Scouting people can save time by watching only potential candidates who satisfy certain filters such as candidate who has at least 95% similarity to player leaving the team. There are a few emerging companies (e.g. SciSports) which already provide this service but it is a pretty limited field yet.

There are some limitations of the study that should be considered in the further research on the same topic. Average time that players have ball is about two minutes so analyzing the such a game using only data of actions performed with the ball can be insufficient. For instance, GPS based tracking data would reveal further insights to cover remaining eighty-eight minutes as stated by Silva et al. (2018). Secondly, there are more features related to attack than defense that results biased model as it gives more weights to attacking features. Moreover, football is a team play and playing style of team – so the teammates – have a significant effect on style of footballer. Another clustering for teams can be conducted and used as input for player clustering. Finally, some players can play at different positions in different games depending on the need of the team and having model based on the total game statistics can change some statistics of these players. Those players might be considered as different observations using detailed positions.

APPENDIX

PYTHON SCRIPT

```
@author: kalenderogluu
"""
#Read the Data
import pandas as pd
import glob

#Read all datasets for Turkish Super League
path = r'pgs_STSL'
all_files = glob.glob(path + "*.csv")

li = []

for filename in all_files:
    df = pd.read_csv(filename, index_col=None, header=0, encoding = "latin-1")
    li.append(df)

pgs_STSL = pd.concat(li, axis=0, ignore_index=True)

#Read all datasets for English Premier League
path = r'pgs_EPL'
all_files = glob.glob(path + "*.csv")

li = []

for filename in all_files:
    df = pd.read_csv(filename, index_col=None, header=0, encoding = "latin-1")
    li.append(df)

pgs_EPL = pd.concat(li, axis=0, ignore_index=True)
```

```
#Read all datasets for German Bundesliga
path = r'pgs_Bundesliga'
all_files = glob.glob(path + "*.csv")

li = []

for filename in all_files:
    df = pd.read_csv(filename, index_col=None, header=0, encoding = "latin-1")
    li.append(df)

pgs_bundesliga = pd.concat(li, axis=0, ignore_index=True)

#New column can be generated to get number of matches played (total min played divided by 90 min.)
pgs_STSL = pgs_STSL.rename(index=str, columns={'Time Played': 'time_played'})
pgs_STSL['avg_game'] = pgs_STSL['time_played'] / 90

pgs_EPL = pgs_EPL.rename(index=str, columns={'Time Played': 'time_played'})
pgs_EPL['avg_game'] = pgs_EPL['time_played'] / 90

pgs_bundesliga = pgs_bundesliga.rename(index=str, columns={'Time Played': 'time_played'})
pgs_bundesliga['avg_game'] = pgs_bundesliga['time_played'] / 90

pgs_STSL.head()
#Divide all relevant columns by avg number of game which is added in the previous step, in order to have 'per 90 min. stat'
pgs_STSL_2 = pgs_STSL.iloc[:,8:252].div(pgs_STSL.avg_game, axis=0).round(2)
pgs_EPL_2 = pgs_EPL.iloc[:,8:252].div(pgs_EPL.avg_game, axis=0).round(2)
pgs_bundesliga_2 = pgs_bundesliga.iloc[:,8:252].div(pgs_bundesliga.avg_game, axis=0).round(2)
```

```
pgs_STSL_2.head()

#MinMaxScaler is better when dealing with outliers and losing them is not desired
from sklearn.preprocessing import MinMaxScaler
transformer = MinMaxScaler().fit(pgs_STSL_2)
np_array = transformer.transform(pgs_STSL_2)
pgs_scaled = pd.DataFrame(np_array)

transformer = MinMaxScaler().fit(pgs_EPL_2)
np_array = transformer.transform(pgs_EPL_2)
pgs_scaled2 = pd.DataFrame(np_array)

transformer = MinMaxScaler().fit(pgs_bundesliga_2)
np_array = transformer.transform(pgs_bundesliga_2)
pgs_scaled3 = pd.DataFrame(np_array)

#Apply PCA
from sklearn.decomposition import PCA
pca = PCA(n_components=15)
principalComponents = pca.fit_transform(pgs_scaled_vc.iloc[:, 5:])

#Apply PCA
from sklearn.decomposition import PCA
pca = PCA(n_components=15)
principalComponents = pca.fit_transform(pgs_scaled_vc.iloc[:, 5:])

#Apply Hierarchical Clustering
from scipy.cluster.hierarchy import linkage, fcluster
hcal_data = pd.DataFrame(km_data)
z = linkage(hcal_data, 'ward')
hcal_data['cluster_labels'] = fcluster(z, 3, criterion='maxclust')
```

```
#Combine cluster labels with initial df
clustered_data_h = pd.concat([cluster_data, hcal_data['cluster_labels']], axis=1)

#Apply HDBSCAN
import hdbscan
hcal_data = pd.DataFrame(km_data)
clusterer = hdbscan.HDBSCAN(min_cluster_size=10)
hcal_data['cluster_labels'] = clusterer.fit_predict(hcal_data)

#Combine cluster labels with initial df
clustered_data_hdb = pd.concat([cluster_data, hcal_data['cluster_labels']], axis=1)

#Create a function to calculate accuracy, precision, recall and F1 score for each position
def eval_metrics(df1):
    tp = df1[(df1['position_id']==1)&(df1['cluster']==1)].count()[0]
    fp = df1[(df1['position_id']==0)&(df1['cluster']==1)].count()[0]
    fn = df1[(df1['position_id']==1)&(df1['cluster']==0)].count()[0]
    tn = df1[(df1['position_id']==0)&(df1['cluster']==0)].count()[0]

    accuracy = (tp+tn) / (tp+fp+fn+tn)
    precision = tp / (tp+fp)
    recall = tp / (tp+fn)
    f1_score = 2*(recall*precision) / (recall+precision)

    print("Accuracy is: ", accuracy.round(2))
    print("Precision is: ", precision.round(2))
    print("Recall is: ", recall.round(2))
    print("F1 score is : ", f1_score.round(2))
#Create dendogram
from sklearn.cluster import AgglomerativeClustering
```

```
import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage(hcal_data,method='ward'))

#Calculate silhouette score

#Convert values of kmeans to array
km_data = cluster_data.values

#Plot silhouette analysis for different number of clusters
from sklearn.metrics import silhouette_samples, silhouette_score

for k in ([2,3,4,5,6]):

    hcal_data = pd.DataFrame(cluster_data)
    z = linkage(hcal_data, 'ward')
    hcal_data['cluster_labels'] = fcluster(z, k, criterion='maxclust')
    labels = hcal_data['cluster_labels']

    # Get silhouette samples
    silhouette_vals = silhouette_samples(km_data, labels)

    # Get the average silhouette score and plot it
    avg_score = np.mean(silhouette_vals).round(4)

    print("For n_clusters =", k,
          "The average silhouette_score is :", avg_score)
#Apply Hierarchical Clustering
from scipy.cluster.hierarchy import linkage, fcluster
hcal_data = pd.DataFrame(cluster_data)
z = linkage(hcal_data, 'ward')
hcal_data['cluster_labels'] = fcluster(z, 3, criterion='maxclust')
```

```
        #Combine cluster labels with initial df
        clustered_data_2 = pd.concat([finalDF_2, hcal_data['cluster_labels']], axis=1)

        #Select features and print cluster centers for these features
        selected_features = ('PC1', 'PC2','PC3')
        print(clustered_data_2.groupby('cluster_labels')[selected_features].mean().round(3)
)
        #Plot cluster centers to visualize clusters
        clustered_data_2.groupby('cluster_labels')[selected_features].mean().plot(legend=T
rue, kind='bar')
        plt.show()
```

REFERENCES

Aalbers, B. (2019, April 17). Player Roles: How to find the right type of player for your team? [Blog post]. Retrieved from https://www.scisports.com/player-roles-how-to-find-the-right-type-of-player-for-your-team/

Campello, R., Moulavi, D., & Saner, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates.

Decroos, T., Bransen, L., Haaren, J. V., & Davis, J. (2018). Actions Speak Louder Than Goals: Valuing Player Actions in Soccer.

Hees, J. (2015, August 26). SciPy Hierarchical Clustering and Dendrogram Tutorial [Blog post]. Retrieved from https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/

Ingersoll, K., Edmund, M., & Sebastian, S. (2017). Heterogeneity and team performance: Evaluating the effect of cultural diversity in the world's top soccer league. *Journal of Sports Analytics 3*, 67–92.

Kerr, M. (2015). Applying Machine Learning to Event Data in Soccer.

Liu, H., Yi, Q., Gimenez, J., Gomez, M., & Lago-Penas, C. (2015). Performance profiles of football teams in the UEFA Champions League considering situational efficiency. *International Journal of Performance Analysis in Sport 15*, 371-390.

Love, C. (2013). Key attributes & Roles [Blog post]. Retrieved from https://steamcommunity.com/sharedfiles/filedetails/?id=121071384

Silva, V.D., Caine, M., Skinner, J., Dogan, S., Kondoz, A., Peter, T., … Smith, B. (2018). Player Tracking Data Analytics as a Tool for Physical Performance Management in Football: A Case Study from Chelsea Football Club Academy. *Sports 2018, 6*, 130; retrieved from *doi:10.3390/sports6040130*

Tactics, E. (2018, May 14). Understanding roles in Football Manager (and real life) (part 1) [Blog post]. Retrieved from https://medium.com/@v_maedhros/understanding-roles-in-football-manager-and-real-life-part-1-73054cfbb303

Tan, P., Steinbach, M., Karpatne, A., & Kumar, V. (2018). Introduction to Data Mining. *Cluster Analysis: Basic Concepts and Algorithms*, 496-515. Retrieved from https://www-users.cs.umn.edu/~kumar001/dmbook/ch8.pdf

Ven, E., & Alfons, A., (2018). Clustering soccer players to find the drivers of soccer team performance.