CS982: BIG DATA TECHNOLOGY

Data Analysis on English Premier League Players

University of Strathclyde
MSc Artificial Intelligence & Applications

Contents

List of Figures

List of Tables

1.Introduction to the dataset

1.1 Football and Premier League

Association football, more commonly known as simply football or soccer,[a] is a team sport played with a spherical ball between two teams of 11 players. (Wikipedia. 2021)

There are 4 kinds of position in general in the football game: Goalkeeper (GK), Defender (DF), Mid Fielder (MF), Forward (FW). When an offence (also called foul) occurs during the game, it is at referee's discretion as whether to display a yellow/red card to the player that commits the offence or not. Normally a yellow card is displayed when the offence is considered intermediate, a red card is displayed when it is considered serious. If an offence is committed in the penalty area, that is the rectangular marked field in front of both goals, a penalty will be given to the victim's team. A penalty is taken by one player at the penalty mark, which is 11m (12 yards) from the goal line. When a ball is kicked into the goal by either team, it is counted as a goal. An assist is made when one player passes the ball to another player, who finishes the ball into the back of the net. Usually there will be lots of passes completed during a game, by which players try to work together to manage to score a goal.

The **Premier League**, often referred to as the **English Premier League** or the **EPL** (legal name: **The Football Association Premier League Limited**), is the top level of the English football league system. Contested by 20 clubs, it operates on a system of promotion and relegation with the English Football League (EFL) (Wikipedia. 2021). As the world's most famous league, EPL could thus provide us ample and high-quality data to analyse, leading to the possibility to find some interesting relationships in it.

First, we will introduce the variables and instances of the dataset in Section 2. Then we will talk about some interesting questions to be tackled later using a few big data technologies in Section 3. We will have a look at the summary statistics for the data being analysed at Section 4. In Sections 5 and 6 we will go deeper into the data by unsupervised analysis method and supervised analysis method respectively. In the last section we will reflect on the methods we used for analysis and see how we can do better in Section 7.

2 Dataset and its variables

The dataset is taken from Kaggle and the name is “The English Premier League (2020 – 21)” (kaggle.com. (n.d.)), which indicates that this is the data from the English Premier League season 2020 – 2021. There are 18 columns in the dataset, some of them are categorical and others numeric. I am going to explain the meaning for each of them in the following part.

1. Name: Name of the player
2. Club: Club of the player
3. Nationality: Nationality of the player
4. Position: Positions for which the player plays
5. Age: Age of the player
6. Matches: Number of the matches played
7. Starts: Number of times the player was in the starting lineup of the match
8. Mins : Number of minutes the player played overall
9. Goals: Number of goals scored by the player
10. Assists: Number of assists given by the player
11. Passes_Attempted: Number of passes attempted by the player
12. Perc_Passes_Completed: Percentage of passes completed by the player successfully.
13. Penalty_Goals: Number of penalty goals scored by the player.
14. Penalty_Attempted: Number of penalties attempted by player.
15. xG: Expected Goals by the player.
16. xA: Expected Assists by the player.
17. Yellow_Cards: Number of Yellow Cards acquired by player throughout the season.
18. Red_Cards: Number of red cards acquired by player throughout the season.

3. Challenges/Problems to be addressed

In this section we will be looking at the problems that are to be addressed using some big data technologies. First by looking at its summary statistics we will gain a general overview of the dataset, the problem to be addressed at this stage is about the age of football players: Are football players more likely to be a young player or players age about 30? What is the distribution of the age of football players look like? Second problem is on the relationship between the tendency of getting booked and the age, we will deal with it using supervised algorithm. Lastly as we know there are in general 4 positions in the football game: Goalkeeper, Defender, Midfielder and Forward. Can we distinguish the position of a player given his data? Is there a data pattern for a particular position, i.e., player of a certain position could possibly behave differently from that of other positions? The last problem is to be tackled using supervised learning algorithm.

4. Summary statistics of the data

Here we are using the summary statistics of the dataset to do some analysis on the age of the players to find out if they tend to be younger or older. As Figure 4.1 shows, the median age is indicated by the yellow line and the red line indicates the mean age. The difference between them is quite small, however median age is still higher than the mean age, which hints that the age is in general symmetrically distributed but still slightly right-skewed. This tells us football potentially prefers younger plyers than older players.

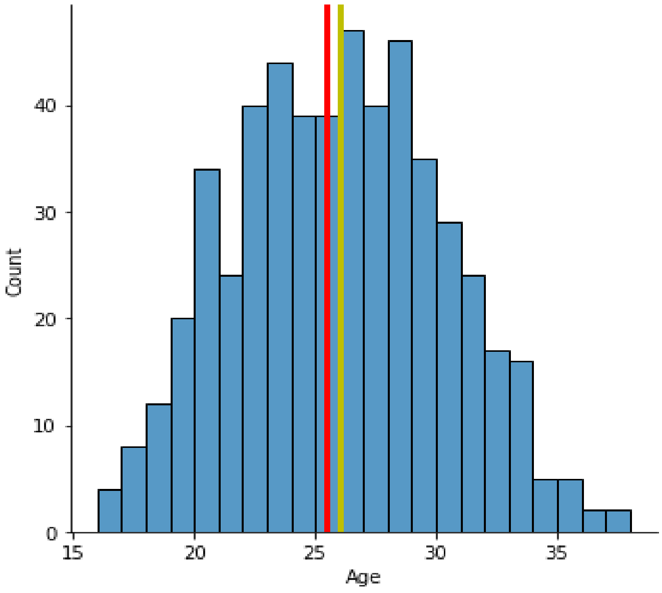


Figure 4.1 Distribution of the age

We can also see the minimum and maximum values of age is 16 and 38, respectively. Both figures are surprising as players start to play professional football at a very early age when they are just allowed to work, and there are players who are persist at the age of 38, which I think is too old for a competitive sport full of physical interactions. However, both are still reasonable enough to be true.

We continue to study the age by looking at the boxplot to get a sense of outliers. In Figure 4.2 we do not see any points outside the upper whisker, which means that the maximum value lies below Q3+1.5*IQR. This shows there is no outlier currently playing football at a professional level. That is very fair to me as I don’t expect older players could still be football professionals in such a highly competitive career.

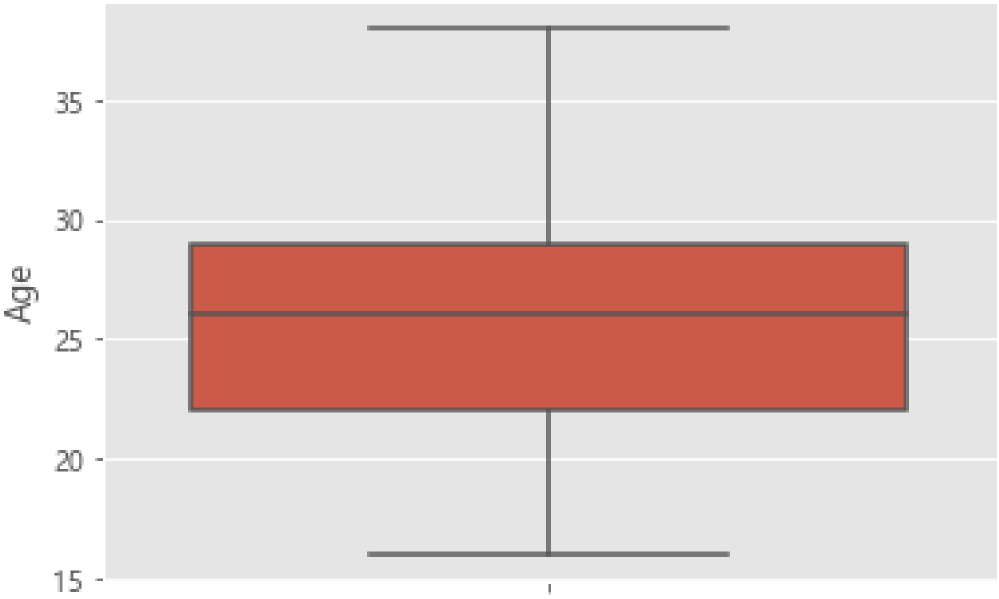


Figure 4.2 Boxplot of the age

5. Unsupervised Analysis Method

5.1 K-Means Clustering

Description:

The k-means clustering algorithm finds centroids that best represent the data using an iterative process. The algorithm starts with a predefined set of centroids, which are normally data points taken from the training data. The k in k-means is the number of centroids to look for and how many clusters the algorithm will find. For instance, setting k to 3 will find three clusters in the dataset (Robert Layton, 2017, p.223)

Rationale:

There are two phases to the k-means: assignment and updating. In the assignment step, we set a label to every sample in the dataset linking it to the nearest centroid. For each sample nearest to centroid 1, we assign the label 1. For each sample nearest to centroid 2, we assign a label 2 and so on for each of the k centroids. These labels form the clusters, so we say that each data point with the label 1 is in cluster 1 (at this time only, as assignments can change as the algorithm runs). In the updating step, we take each of the clusters and compute the centroid, which is the mean of all the samples in that cluster. The algorithm then iterates between the assignment step and the updating step; each time the updating step occurs, each of the centroids moves a small amount. This causes the assignments to change slightly, causing the centroids to move a small amount in the next iteration. This repeats until some stopping criterion is reached. It is common to stop after a certain number of iterations, or when the total movement of the centroids is very low. The algorithm can also complete in some scenarios, which means that the clusters are stable—the assignments do not change and neither do the centroids. (Robert Layton, 2017, p.223 - p.224)

Application: Analysis is conducted on the following columns of the dataset: Age, Yellow_Cards. We want to find out is there a relationship between the above as it is widely known that there is a peak age when people tend to be very temperamental and later, they become less aggressive and get less easily frustrated.

When it comes to the number of clusters, there is difficulty in deciding the best number for it. Robert Layton (2017, p.226) has pointed out that:

> Clustering is mainly an exploratory analysis, and therefore it is difficult to evaluate a clustering algorithm's results effectively. A straightforward way is to evaluate the algorithm based on the criteria the algorithm tries to learn from
>
> In the case of the k-means algorithm, the criterion that it uses when developing the centroids is to minimize the distance from each sample to its nearest centroid. This is called the inertia of the algorithm and can be retrieved from any k-means instance that has had fit called on it:
>
> pipeline.named_steps['clusterer'].inertia_

The value of the inertia should decrease with reducing improvement as the number of clusters [increases], … , Looking for this type of pattern is called the elbow rule (Robert Layton, 2017, p.227-p.228).

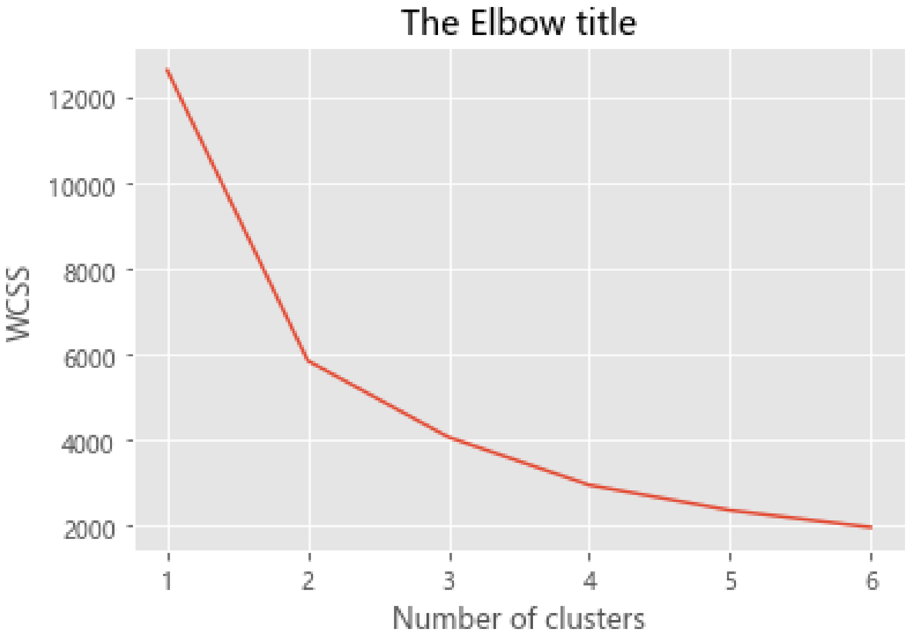


Figure 5.1 The Elbow Curve

By observing Figure 5.1, it is hard to decide the best number of clusters as 2,3,4 all suit for our purpose. Next, we will use another metric called silhouette score to help us make the decision. As we can see in Figure 5.2, when k =2 silhouette score is the greatest.

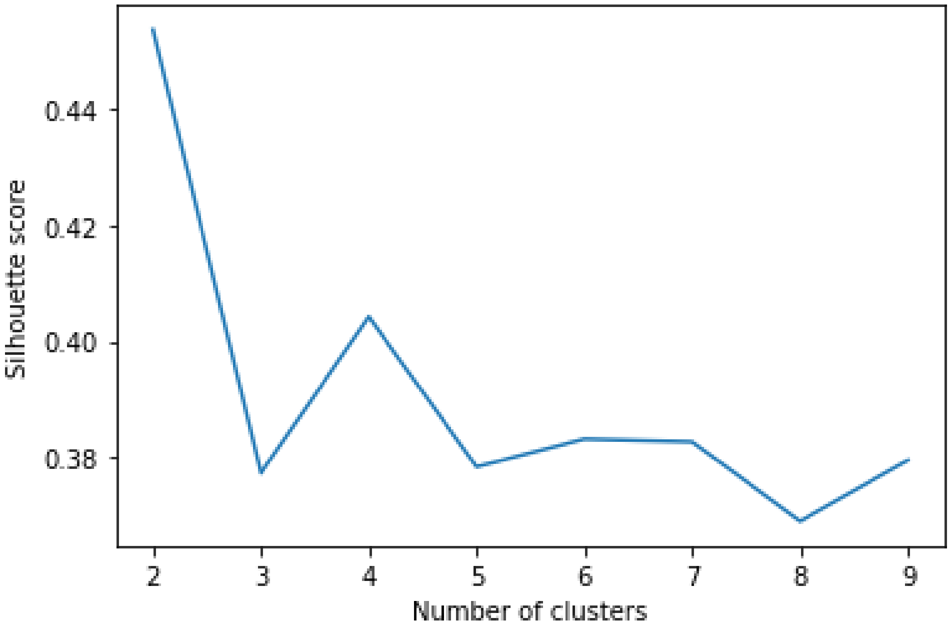


Figure 5.2 Silhouette score

However, it is not ideal to separate the players into 2 groups as when we do so, the players are separated into 2 groups by age without any influence of yellow cards as we can in Figure 5.3.

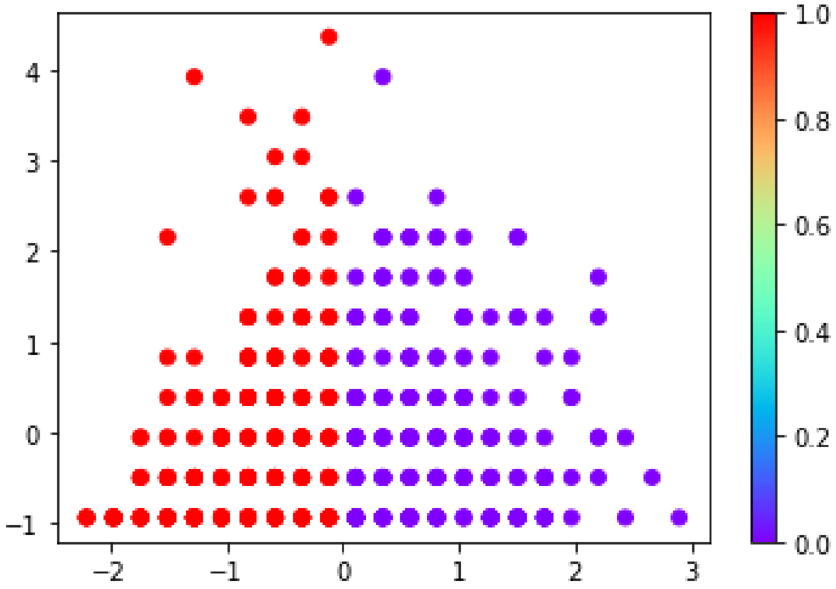


Figure 5.3 Clustering when k = 2

We can see there is a bump in Figure 5.2 when k = 4, so we decide to pick 4 as the number for clusters.

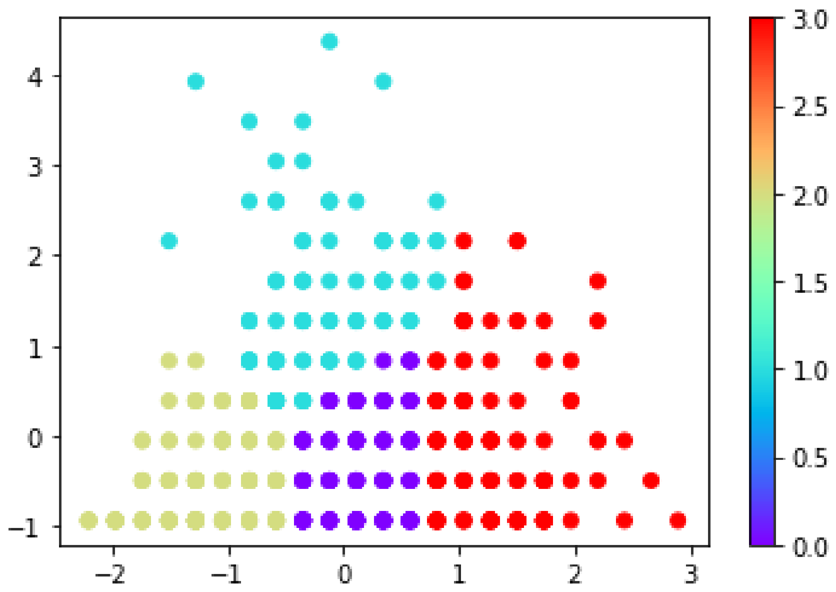
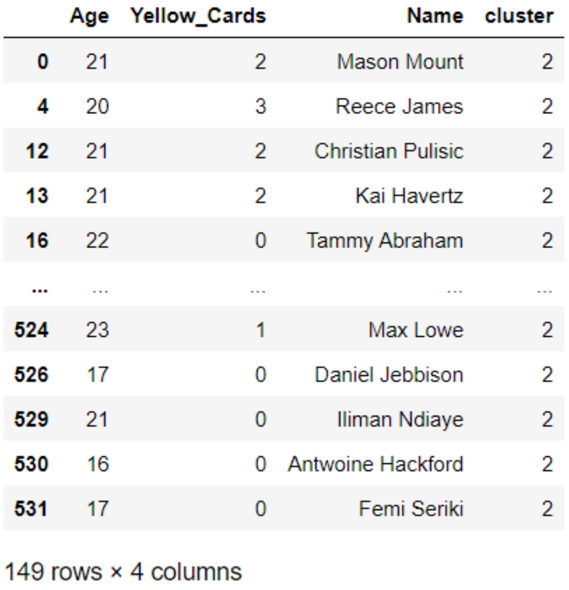


Figure 5.4 Clustering when k =4

Findings: We can see from the clustering that there are 4 kinds of player. The first kind in yellow is relatively young and getting few yellow cards, we can thus conclude they are the young players who behave well and don’t get booked too much. Table 5.1 is the list of them.

Table 5.1 Table of players in the yellow cluster

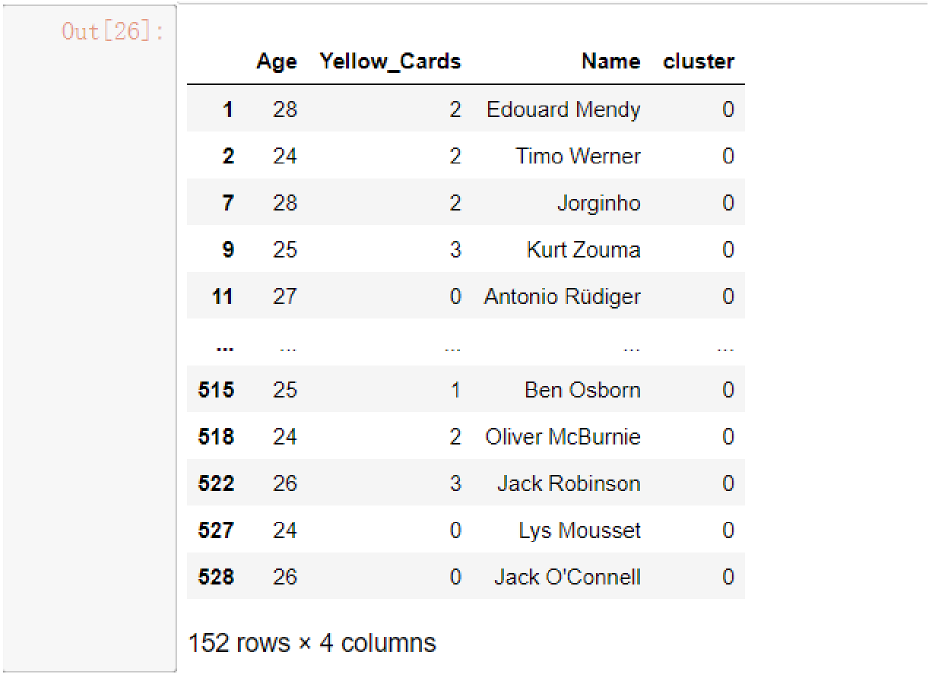


	Age	Yellow_Cards	Name	cluster
0	21	2	Mason Mount	2
4	20	3	Reece James	2
12	21	2	Christian Pulisic	2
13	21	2	Kai Havertz	2
16	22	0	Tammy Abraham	2
...	...	...	...	...
524	23	1	Max Lowe	2
526	17	0	Daniel Jebbison	2
529	21	0	Iliman Ndiaye	2
530	16	0	Antwoine Hackford	2
531	17	0	Femi Seriki	2

149 rows × 4 columns

The second kind in blue is relatively older than the previous ones and they behave well on field as well. Table 5.2 is the list of them:

Table 5.2 Table of players in the blue cluster

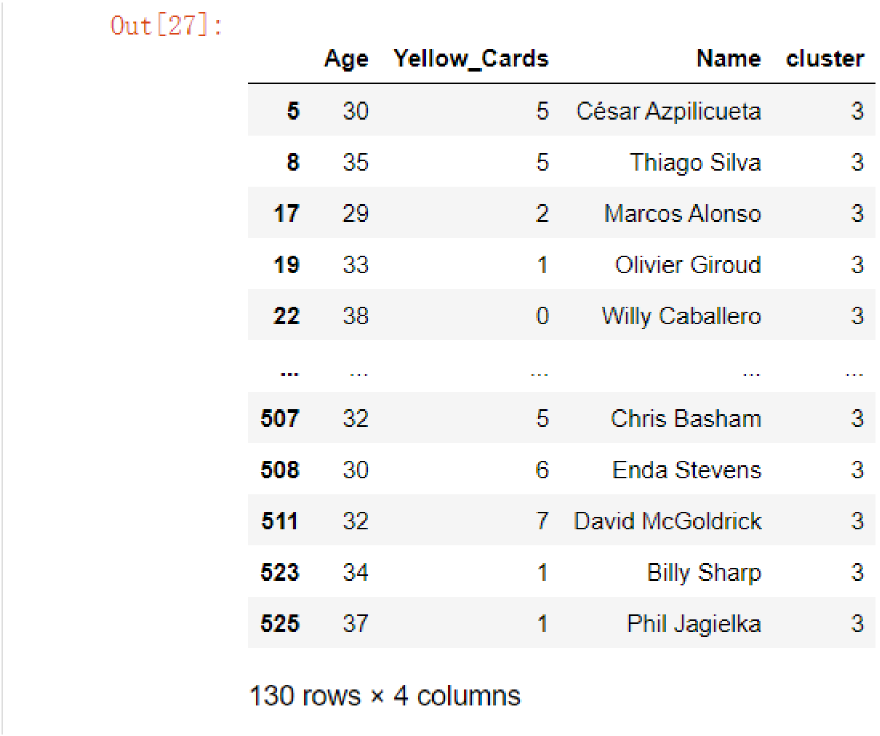


	Age	Yellow_Cards	Name	cluster
1	28	2	Edouard Mendy	0
2	24	2	Timo Werner	0
7	28	2	Jorginho	0
9	25	3	Kurt Zouma	0
11	27	0	Antonio Rüdiger	0
...	...	...	...	...
515	25	1	Ben Osborn	0
518	24	2	Oliver McBurnie	0
522	26	3	Jack Robinson	0
527	24	0	Lys Mousset	0
528	26	0	Jack O'Connell	0

152 rows × 4 columns

The third kind in red is the oldest among the players and they behave quite well too although some of them do get yellow cards sometimes, they are still considered well-behaved. Table 5.3 is the list of them:

Table 5.3 Table of players in the red cluster



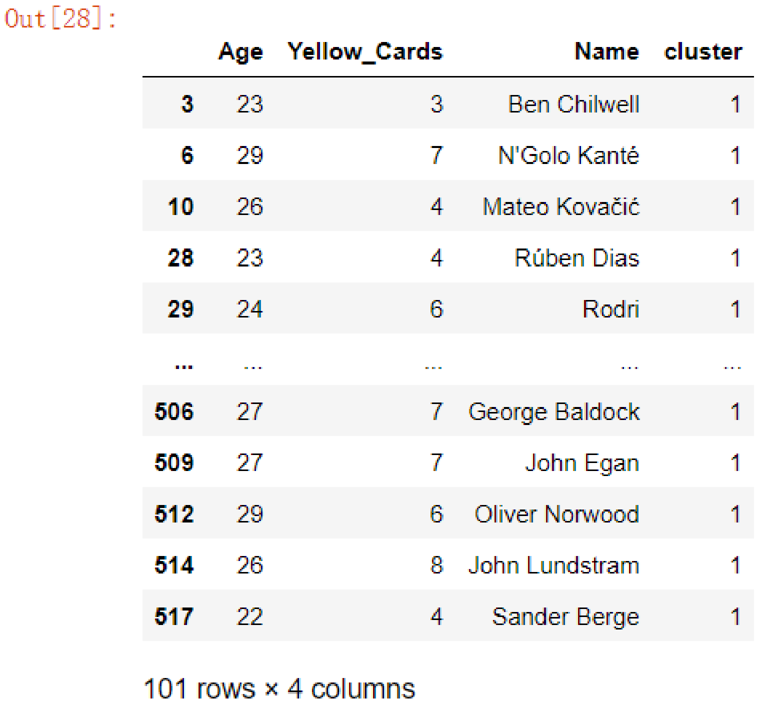
Out[27]:

	Age	Yellow_Cards	Name	cluster
5	30	5	César Azpilicueta	3
8	35	5	Thiago Silva	3
17	29	2	Marcos Alonso	3
19	33	1	Olivier Giroud	3
22	38	0	Willy Caballero	3
...	...	...	...	...
507	32	5	Chris Basham	3
508	30	6	Enda Stevens	3
511	32	7	David McGoldrick	3
523	34	1	Billy Sharp	3
525	37	1	Phil Jagielka	3

130 rows × 4 columns

The fourth kind is of medium age and they get decent amounts of yellow cards, which could possibly because the testosterone level reaches a peak in their body and therefore cause them to be aggressive on the pitch and foul a lot. Apart from the hormone side, it is also possibly because some of them take on more defence job than others and thus commit fouls. Table 5.4 is the list of them:

Table 5.4: Table of players in the green cluster



Out[28]:

	Age	Yellow_Cards	Name	cluster
3	23	3	Ben Chilwell	1
6	29	7	N'Golo Kanté	1
10	26	4	Mateo Kovačić	1
28	23	4	Rúben Dias	1
29	24	6	Rodri	1
...	...	...	...	...
506	27	7	George Baldock	1
509	27	7	John Egan	1
512	29	6	Oliver Norwood	1
514	26	8	John Lundstram	1
517	22	4	Sander Berge	1

101 rows × 4 columns

6. Supervised analysis method

6.1 K nearest neighbors

Description:

> The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training dataset. To make a prediction for a new data point, the algorithm finds the closest data points in the training dataset—its “nearest neighbours.” (Andreas C. Muller, 2017, p.35)

Rationale:

> In its simplest version, the k-NN algorithm only considers exactly one nearest neighbour, which is the closest training data point to the point we want to make a prediction for. (Andreas C. Muller, 2017, p.35)

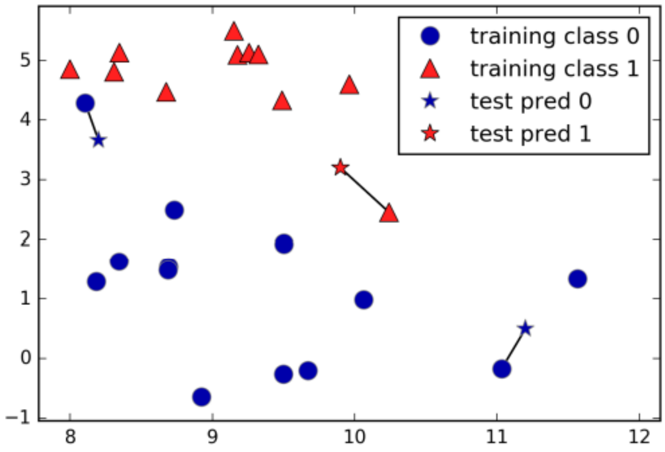


Figure 6.1 K nearest neighbours’ classification when k = 1 (Andreas C. Muller, 2017, p.35)

Instead of considering only the closest neighbour, we can also consider an arbitrary number, k, of neighbours. This is where the name of the k-nearest neighbours algorithm comes from. When considering more than one neighbour, we use voting to assign a label. This means that for each test point, we count how many neighbours belong to class 0 and how many neighbours belong to class 1. We then

> assign the class that is more frequent: in other words, the majority class among the k-nearest neighbours. The following example (Figure 6.2) uses the three closest neighbours: (Andreas C. Muller, 2017, p36)

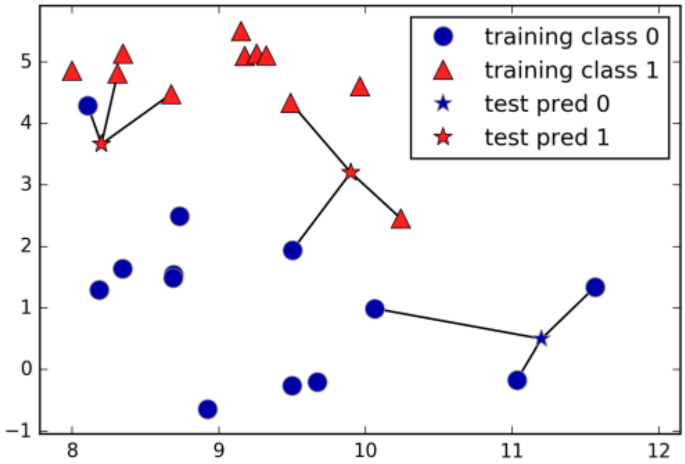


Figure 6.2 K nearest neighbours' classification when k = 3(Andreas C. Muller, 2017, p36)

Application: Analysis is conducted on the numeric columns of the dataset and I have trimmed it a little by only selecting players who play greater or equal than 10 games in the season. Here we want to find out whether are we able to judge a player's position by looking at his numeric data as it is obvious that players of different positions play different roles on the field and thus should have different behaviour pattern.

I turn the position of each player into the corresponding numbers: FW – 3, MF – 2, DF – 1, GK – 0. Also I drop all the categorical data to make it pure numeric. After the dataset is well-prepared for analysis, we need to consider how to set parameters. Robert Layton has pointed out that:

> The nearest neighbour algorithm has several parameters, but the most important one is that of the number of nearest neighbours to use when predicting the class of an unseen attribution. In scikit-learn, this parameter is called n_neighbours. In the following figure(Figure 6.3), we show that when this number is too low, a randomly labelled sample can cause an error. In contrast, when it is too high, the actual nearest neighbours have a lower effect on the result: (Robert Layton, 2017, p.33)

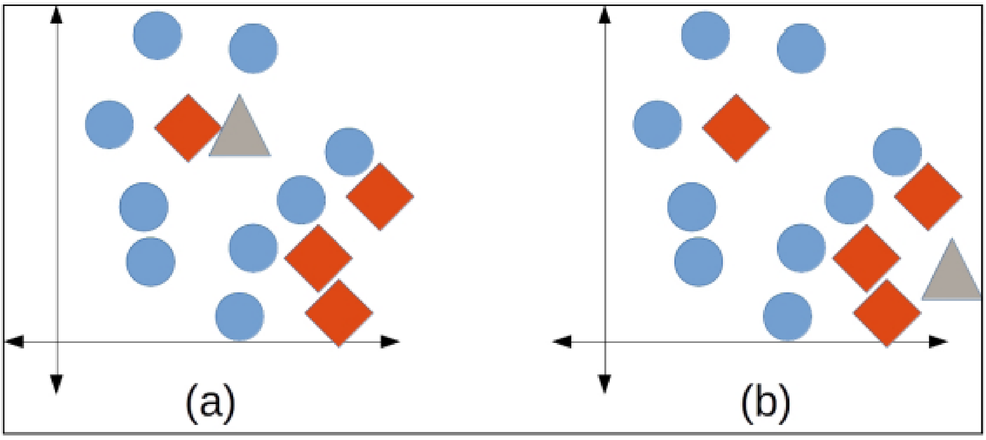


Figure 6.3 Classification Examples of different n (Robert Layton, 2017, p.33)

He continues that:

> In the figure (a), on the left-hand side, we would usually expect the test sample (the triangle) to be classified as a circle. However, if n_neighbours is 1, the single red diamond in this area (likely a noisy sample) causes the sample to be predicted as being a diamond, while it appears to be in a red area. In the figure (b), on the right-hand side, we would usually expect the test sample to be classified as a diamond. However, if n_neighbours is 7, the three nearest neighbours (which are all diamonds) are overridden by the large number of circle samples (Robert Layton, 2017, p.34).

Since the choice of n is so important for the algorithm, we need to make careful decision on the number of it. Here we follow Muller's approach:

> We begin by splitting the dataset into a training and a test set. Then we evaluate training and test set performance with different numbers of neighbours (Andreas C. Muller, 2017, p.38 – p.39).

Figure 6.4 is what I got after taking this approach:

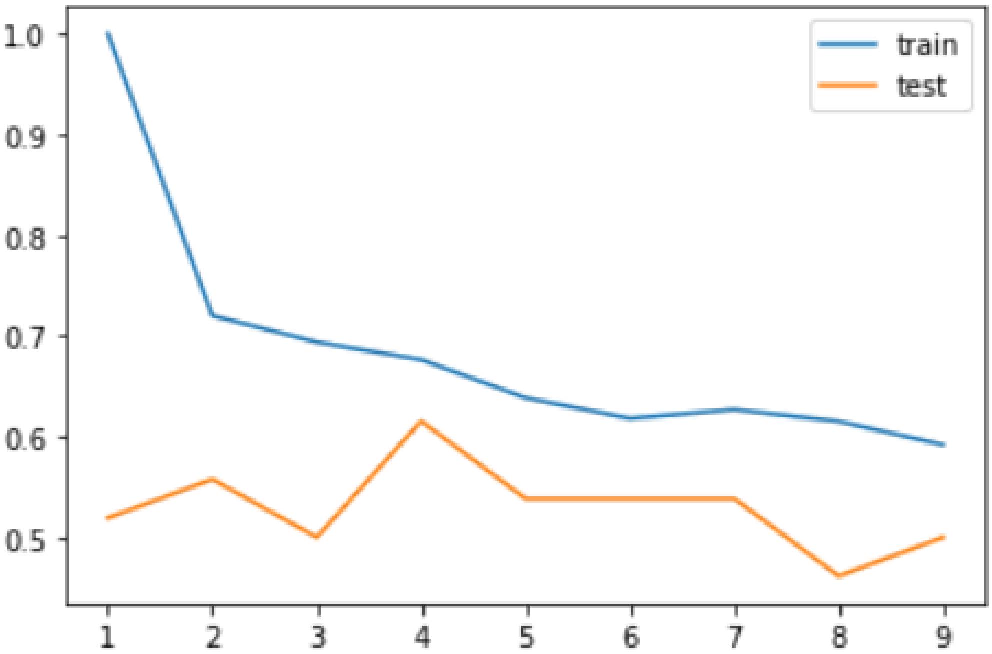
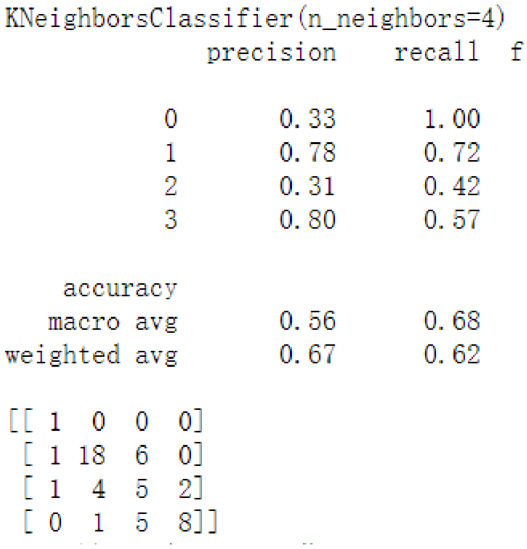


Figure 6.4 Train-Test set accuracy curve

As the Figure 6.4 suggests, we should pick n = 4 as the parameter. It gives us the best performance on train set and test set. Figure 6.5 is the classification report and confusion matrix for n = 4.



```
KNeighborsClassifier(n_neighbors=4)
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         1
           1       0.78      0.72      0.75        25
           2       0.31      0.42      0.36        12
           3       0.80      0.57      0.67        14

    accuracy                           0.62        52
   macro avg       0.56      0.68      0.57        52
weighted avg       0.67      0.62      0.63        52

[[ 1  0  0  0]
 [ 1 18  6  0]
 [ 1  4  5  2]
 [ 0  1  5  8]]
```

Figure 6.5 Classification report and confusion matrix when n = 4

The accuracy on test set is 62%, which did not live up to my expectation as about 90%. Therefore, I would like to try another supervised algorithm to see if it does better than K Nearest Algorithm. The algorithm I pick is Decision Tree.

6.2 Decision Tree Algorithm

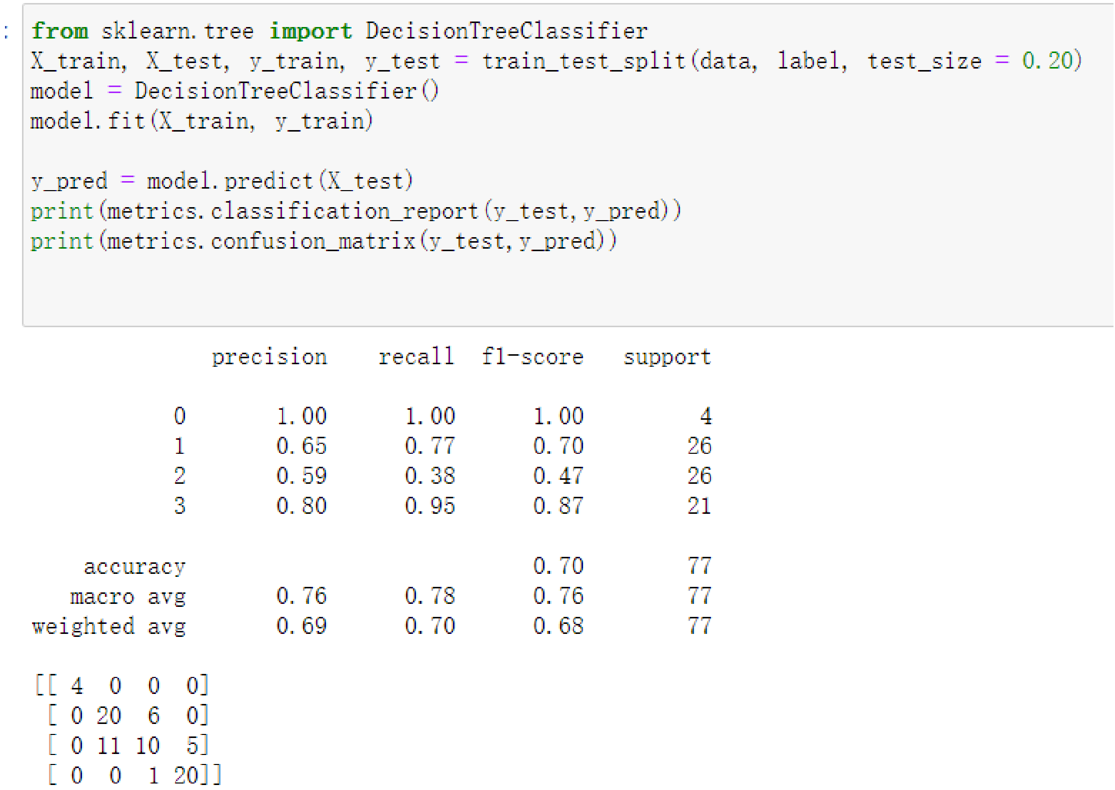
Description:

> Decision trees are a class of supervised learning algorithm like a flow chart that consists of a sequence of nodes, where the values for a sample are used to decide on the next node to go to. (Robert Layton, 2017, p.47)

Rationale:

> The first is the training stage, where a tree is built using training data. While the nearest neighbour algorithm from the previous chapter did not have a training phase, it is needed for decision trees. In this way, the nearest neighbour algorithm is a lazy learner, only doing any work when it needs to make a prediction. In contrast, decision trees, like most classification methods, are eager learners, undertaking work at the training stage. The second is the predicting stage, where the trained tree is used to predict the classification of new samples. (Robert Layton, 2017, p.47)

Application: I simply apply the algorithm to the dataset without too much tuning of the parameter, barely because I want to roughly see the accuracy of this algorithm so that I can make sure the K Nearest Neighbours is doing a decent job in the previous section. Figure 6.6 is the classification report and confusion matrix:



```
from sklearn.tree import DecisionTreeClassifier
X_train, X_test, y_train, y_test = train_test_split(data, label, test_size = 0.20)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(metrics.classification_report(y_test,y_pred))
print(metrics.confusion_matrix(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       0.65      0.77      0.70        26
           2       0.59      0.38      0.47        26
           3       0.80      0.95      0.87        21

    accuracy                           0.70        77
   macro avg       0.76      0.78      0.76        77
weighted avg       0.69      0.70      0.68        77

[[ 4  0  0  0]
 [ 0 20  6  0]
 [ 0 11 10  5]
 [ 0  0  1 20]]
```

Figure 6.6 Classification report and confusion matrix for decision tree algorithm

Decision tree is giving us 70% of accuracy on test set.

Finding from both supervised algorithms: The supervised algorithm can basically tell us what position players play given some data on goals, assists, passes completed and so on of other players. The accuracy can reach above 60% at the best case.

7. Reflection

The first lesson I learned from analysing this dataset is when I was analysing it with unsupervised algorithm, i.e. Kmeans clustering. I used the Elbow Curve and the silhouette score to help me decide what is the best k to choose. These methods indicate me to choose k = 2, which gives me good inertia value as well as good silhouette score. However, I have to take reality into account and realise that k = 2 is not ideal for our purpose. We should not be directed always by any rules and should always take the reality into account because that is why we are doing analysis. The outcome of our analysis should serve to our purposes rather than simply giving the best metrics。

The second lesson is learned through applying supervised algorithm to the dataset, and it is also to do with reality. In the dataset, the position labels are not as simple as I thought it would be. There are not only just 4 kinds of positions, but some of the players can have two positions, which is often the case as some players need to play different positions under manager's instructions. Therefore, we need to deal with this ambiguity or overlap of positions with care. My approach in this case is to pick the position that appears first in one player's position as it should be indicating his main position after taking a closer look at some players positions with my football experience. Another thing I realise after applying decision tree algorithm to analyse the dataset to predict player's position, is that the reason the accuracy of prediction is not so high as I expected is because this ambiguity or overlap that I mentioned earlier. A player can play in a position that is in between forward and midfield, thus his data such as goals, passes, assists can be slightly different from a pure midfielder or forward. Or he could play forward in one game and midfielder the next, which is also confusing for algorithm to judge whether he is a pure forward or midfielder.

The last thing I learned is that, we should do some trimming job for the dataset we are to do analysis on. In this dataset, I pick players who play more or equal than 10 games in the season and drop those who play less than 10. The reason I did this is that those who play less tend to have less informative data and thus make them meaningless to analyse. Their data will mix with some others so that the prediction will be less accurate. For example, a forward player who plays mostly as substitute would not score many and thus his data could possibly look like that of a midfielder or even defender.

8. Appendix A

Environment

Language: Python 3.8.0   IDE: Jupyter Notebook 6.3.0

9. References

Wikipedia. (2021). *Portal: Association football*. [online] Available at: https://en.wikipedia.org/wiki/Portal:Association_football#:~:text=The%20Association%20football%20portal&text=Association%20football%2C%20more%20commonly%20known [Accessed 7 Nov. 2021].

Wikipedia. (2021). *Premier League*. [online] Available at: https://en.wikipedia.org/wiki/Premier_League#:~:text=The%20Premier%20League%2C%20often%20referred [Accessed 7 Nov. 2021].

Müller, A.C. and Guido, S. (2017). *Introduction to machine learning with Python: a guide for data scientists*. Beijing: O’Reilly.

Layton, R. (2017). *Learning data mining with Python : use Python to manipulate data and build predictive models*. Birmingham, UK: Packt Publishing.

kaggle.com. (n.d.). *English Premier League (2020-21)*. [online] Available at: https://www.kaggle.com/rajatrc1705/english-premier-league202021.