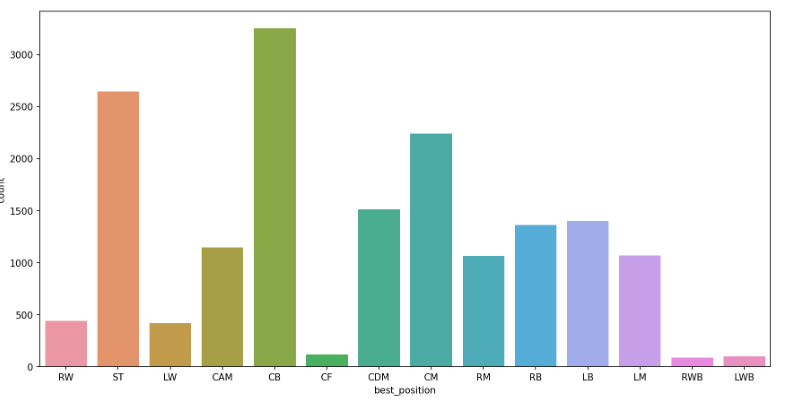
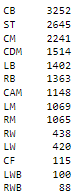
Practical Machine Learning – Project

The project consists in applying supervised and unsupervised models to the Fifa 21 player dataset from Kaggle (<https://www.kaggle.com/datasets/stefanoleone992/fifa-21-complete-player-dataset>).

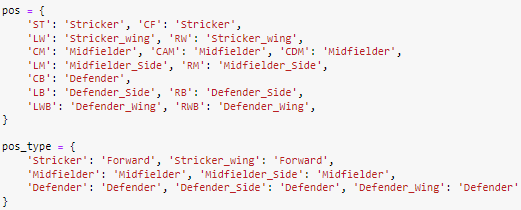
The main goal is to classify players in the position they have on the field. In this project I tried to have two types of labels(positions), the first type are the more specific positions ( Stricker, Stricker\_wing, Midfielder, Midfielder\_Side, Defender, Defender\_Side and Defender\_Wing) and in the second type we have the 3 main roles on the pitch: Forward, Midfielder and Defender. We exclude the goalkeepers because they have other specific stats for they’re roll on the pitch and we will focus only on the 3 mains ones.

The data set has 18944 rows (players) with 106 columns and after we drop some impractical columns and exclude the goalkeepers we remain with 16860 rows and 95 columns. After some more data cleaning we create a new colum called ‘best\_position’ from the column ‘player\_positions’ which has all the positions that the player can play, we take the first one as the best.

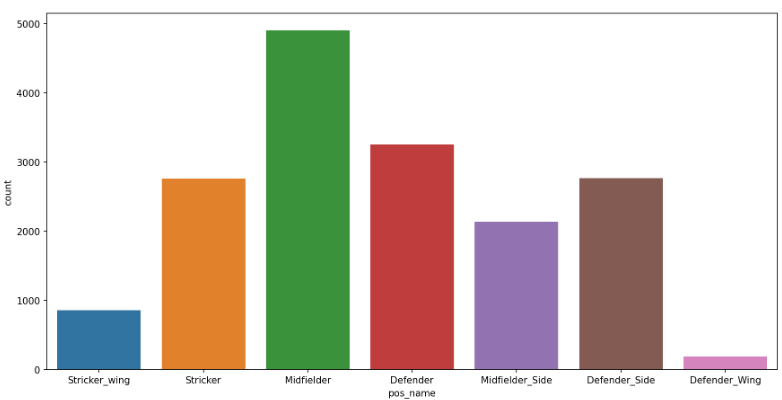
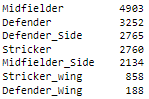
And this is how these specific positions looks like:



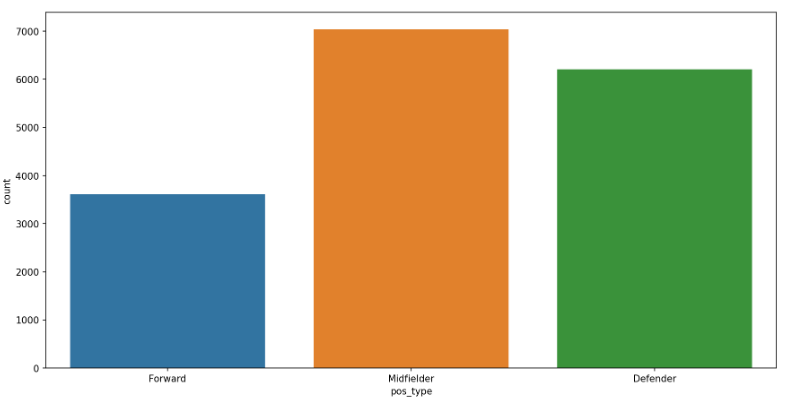
We use this specific positions just to label further using this notations:



And this is how our labels look like:

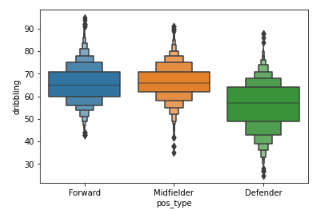
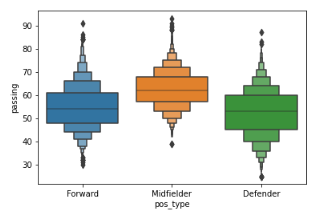


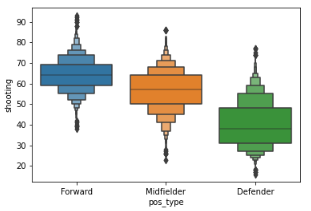
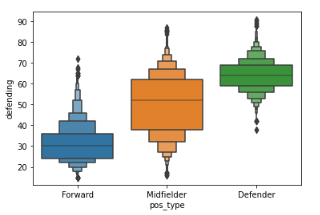
As we can see the main positions are midfielder and defender because at football it is very important not to receive a goal so defense plays a key role in a team. The midfielders have an important role to connect the team and push the ball to the strikers creating opportunities to score.



In this plot we can also see that the number of attackers is significantly lower and the most positions are for midfielders that because they have a huge role in how the team moves from back to front or front to back also the good specific stats like passing to lead their team to victory.

Let’s have a look at their skills:

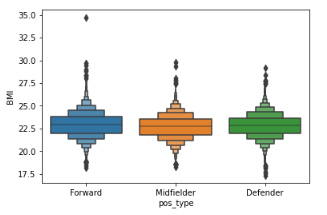


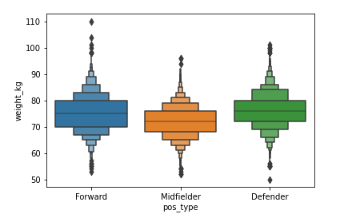
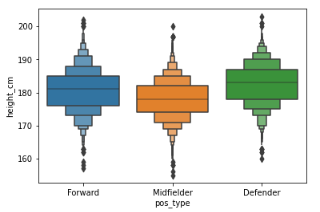
So as I say before a prime skill for the midfielder is that he needs to have good passing and also they are pretty good with dribbling almost good as forward players. Also the defenders have the best skill in defending and the worst in shooting.

I used Body Mass Index as a feature, it is a simple calculation using a person's height and weight.

The formula is: BMI = kg/m2.



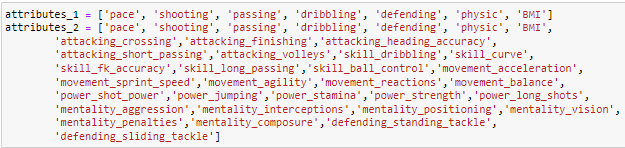
But this index is not showing us some notable difference between roles, even if we look individually at height and weight we can see some differences.

We see that midfielders have less weight than other players and are shorter because they have to run very much back and forth and create spaces between lines.

For the models I will use two types of attributes, the basic ones + BMI (7 attributes) and

the basic ones + BMI + rest of skills (35 attributes).



**Supervised**

After exploratory data analysis I will implement two supervised models on our data to predict their positions.

**KNN**

The abbreviation KNN stands for “K-Nearest Neighbour”. It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.

**Decisional Tree**

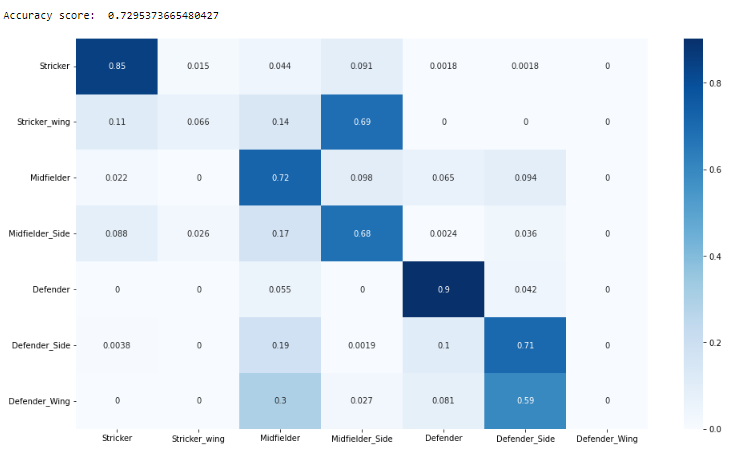
Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.

The first classification will have the 7 attributes and the 7 positions:

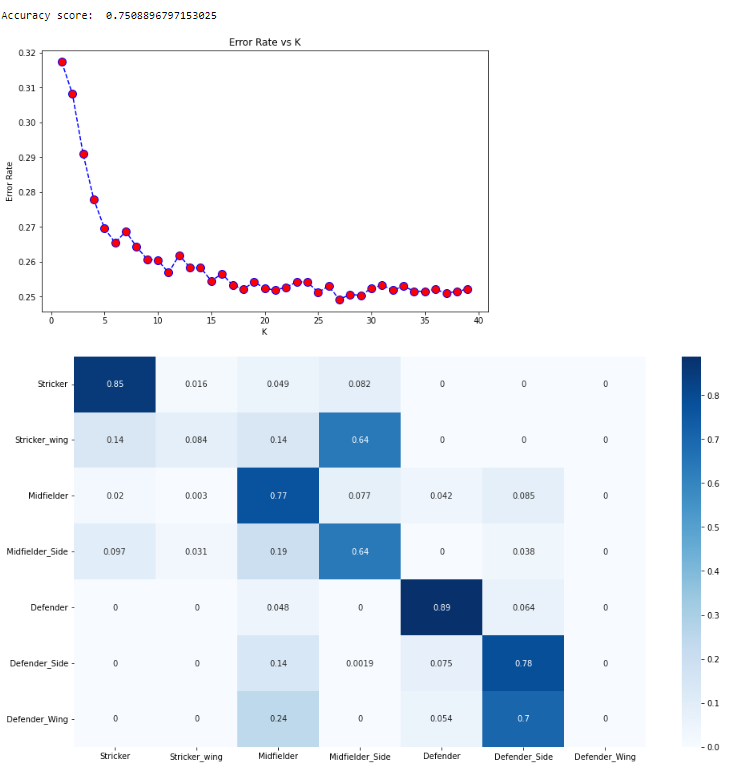




Tree – Gridsearch parameters {'criterion': 'gini', 'max\_depth': 8}

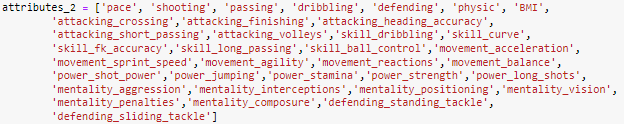


KNN – with Best n\_neighbors: 27

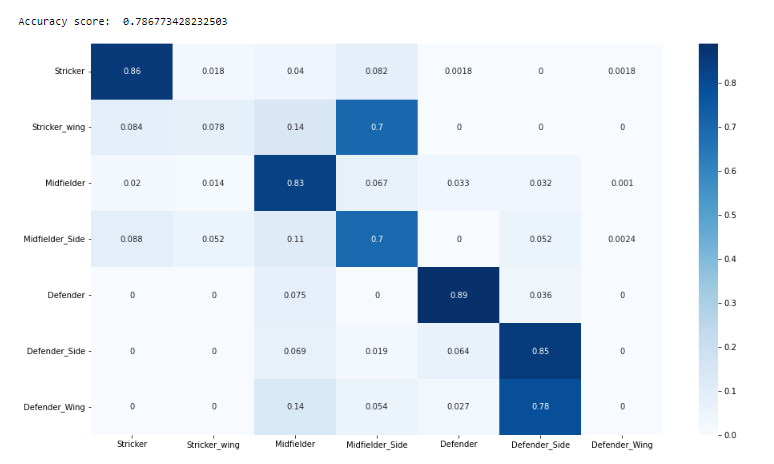


Both scores are really good over 70%, with Knn performing a little better, the algorithm has some little problems with defender\_side and defender\_wing but are very similar role and same thing with midfielder\_side and strciekr\_wing.

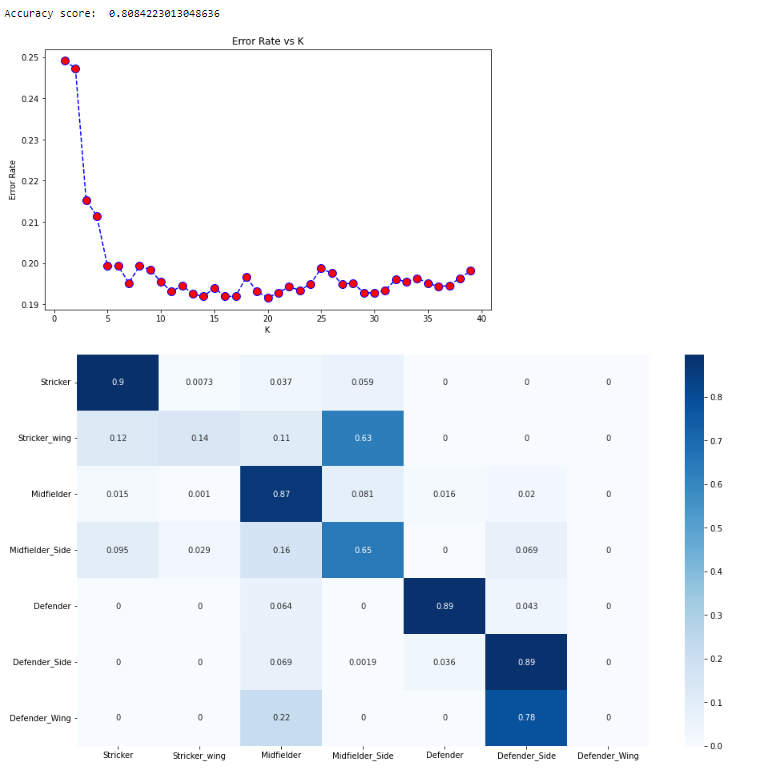
Now it is time to use 35 attributes:



Tree – Gridsearch parameters {'criterion': 'gini', 'max\_depth': 9}



Knn – with Best n\_neighbors: 20

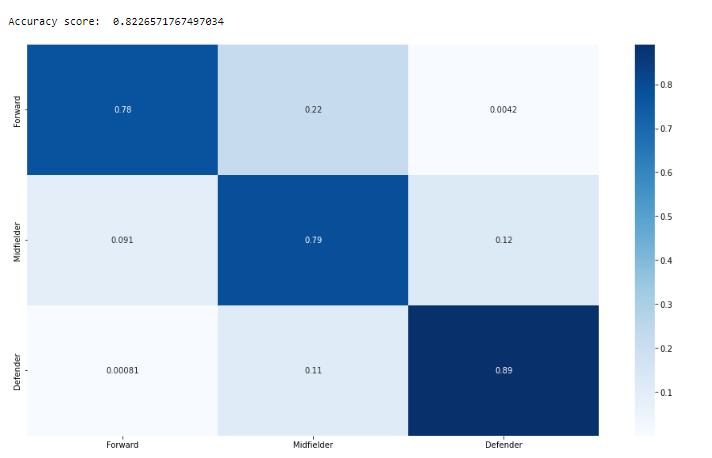


This time with more attributes the algorithm performs even better with over 78% accuracy score, and again KNN has little bit better score than Decisional Tree, we can see in the heatmaps that around midfielder positions are versatile and can misclassify players. Also the stricker\_wing is misclassified with midfielder\_side.

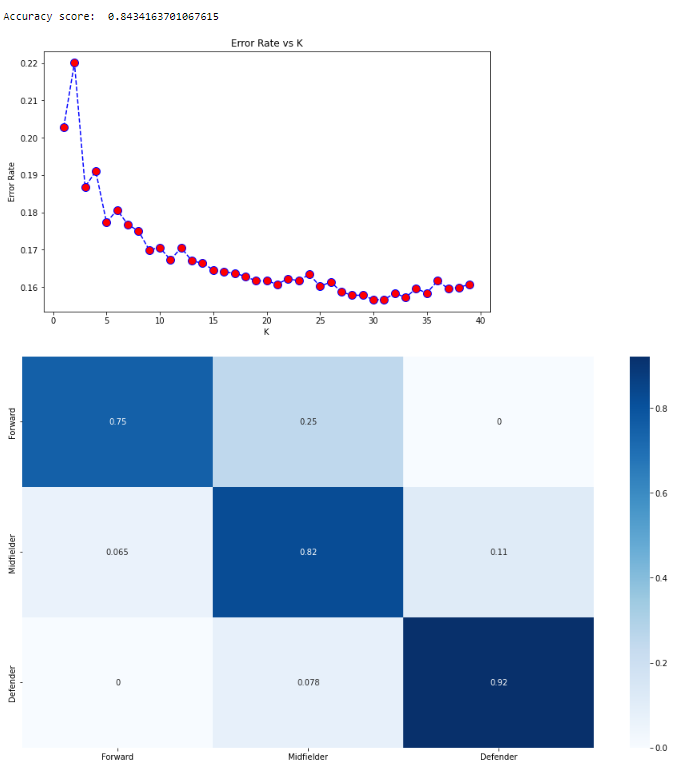
Now is time to use the 3 main positions again with 7 attributes:



Tree - Gridsearch parameters {'criterion': 'gini', 'max\_depth': 8}



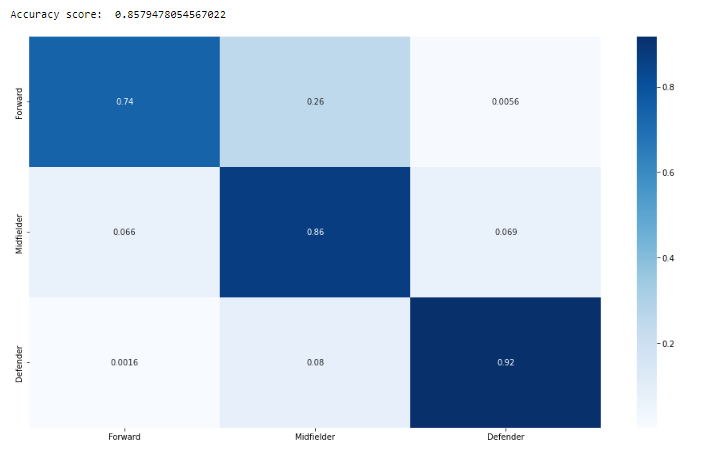
Knn – with Best n\_neighbors: 30



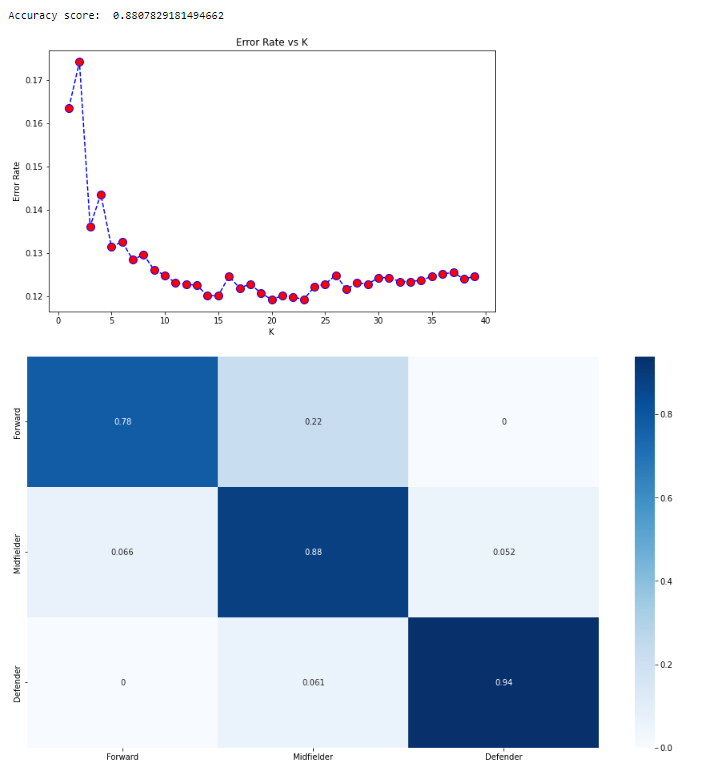
This time performed even better with over 82% accuracy score, and KNN beats again Decisional Tree, we can see that around midfielders are miss classifications because has similar stats as forward players but also with defensive players cause midfielders have important roll in defends but alto in offense.

Let’s see with 35 attributes how it works:

Tree - Gridsearch parameters {'criterion': 'gini', 'max\_depth': 7}



Knn – with Best n\_neighbors: 20

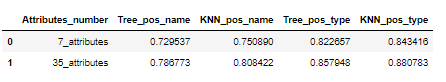


This time with over 85% accuracy score, Knn performs better same confusion between midfielders and forward players because they have similar stats, also I also tried a random decision maker and checked what are the odds of correctly classifying players and given the fact there are 3 classes its around 33% which is not good at all.

**Conclusions**

We can tell that is possible to classify players into roles based on their attributes. We saw that KNN worked better in all cases than Decisional Tree and we understood the importance of working with the data and getting the correct information and creating the right number of classes also more attributes gave us better results.

Pos\_name – 7 positions / pos\_type – 3 main positions

****

****

**Unsupervised**

For the unsupervised part I will implement K-means and Hierarchical clustering

**K-means**

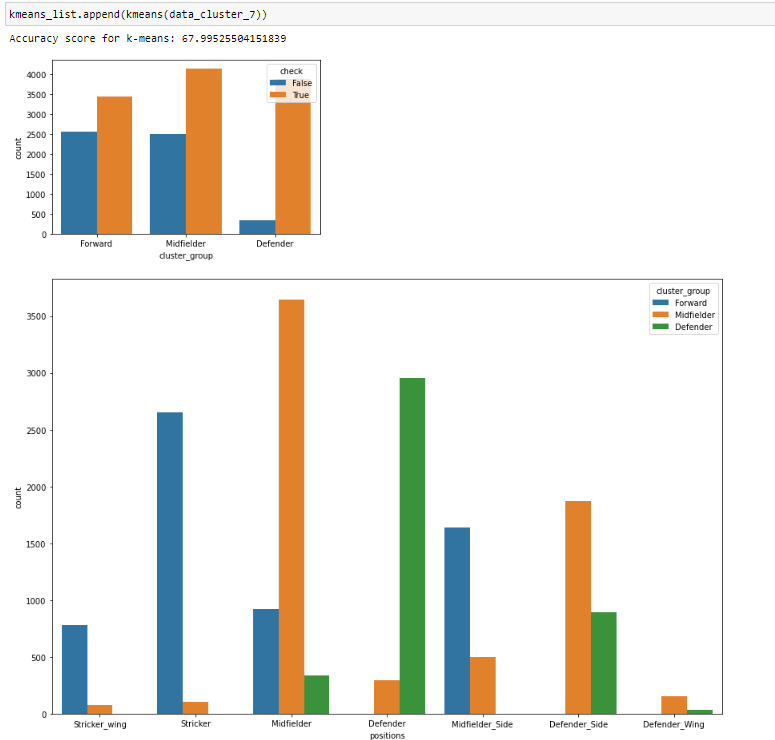
K-means clustering is the popular unsupervised machine learning algorithms, a target number k is defined, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster.

**Hierarchical** **clustering**

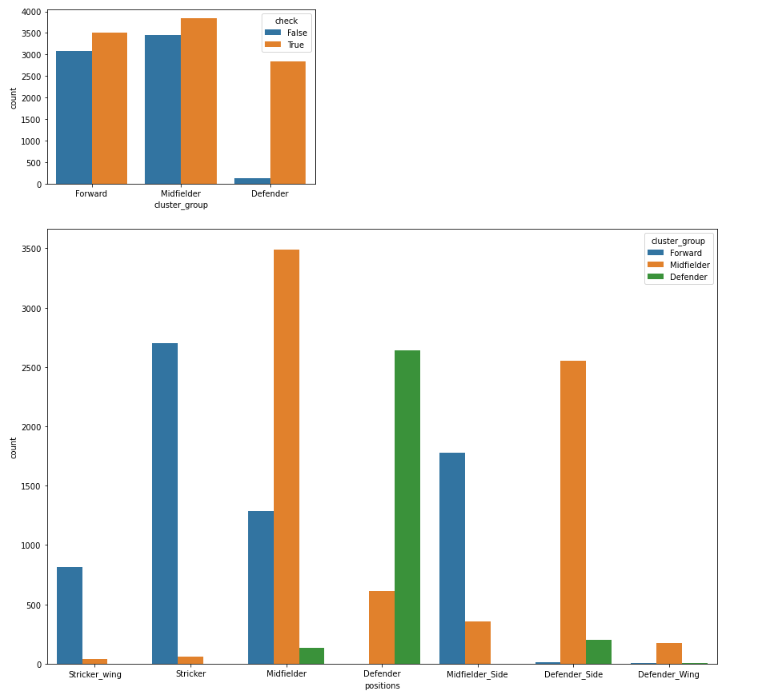
Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

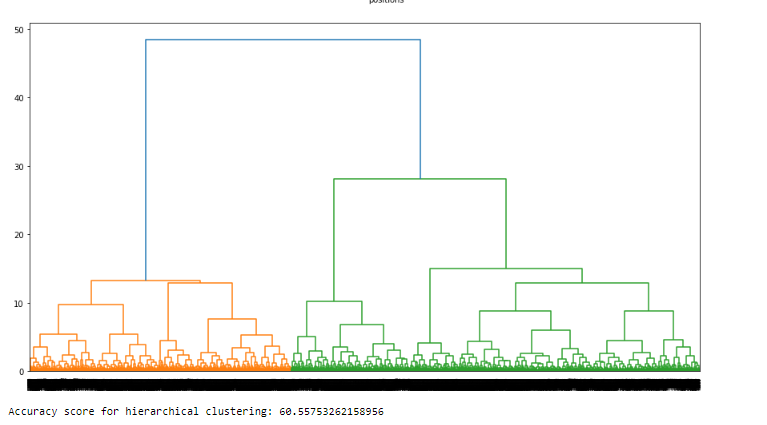
Also as above I want to try with basic attributes and more to see the differences.

K-means



Hierarchical

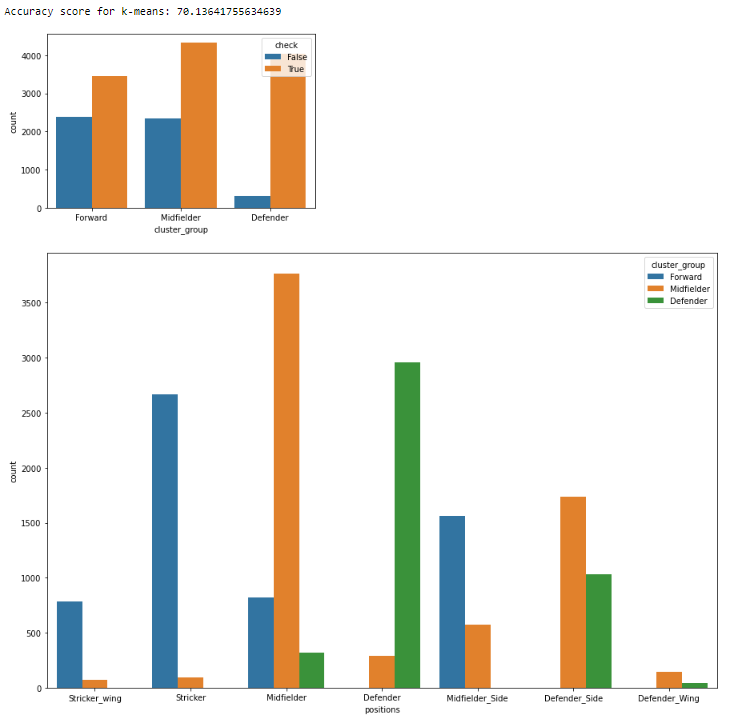
****

****

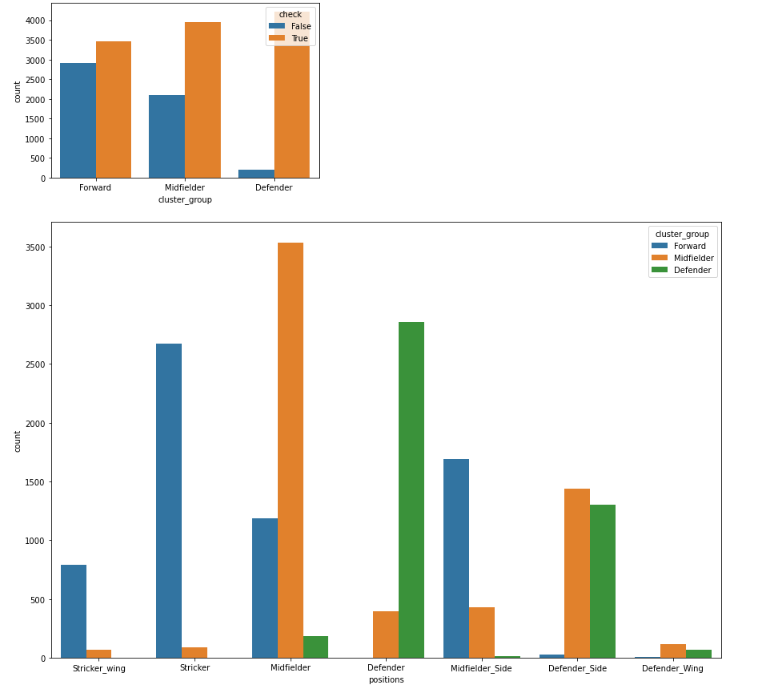
The algorithm performed pretty well for an unsupervised with over 60% accuracy, we can the again as we saw earlier when we tried to classify with supervised methods that there are a big similarity between the forward and midfield positions.

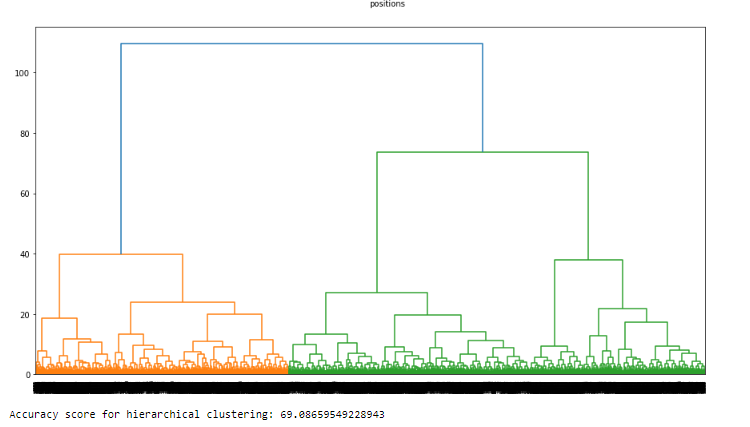
K-means performed better than Hierarchical, how much will be the differences with more attributes?

k-means

****

Hierarchical





This time Hierarchical performed much better almost hitting 70% accuracy score, k-means with a 2% improvement is still better than Hierarchical but no so far one from another.

**Conclusion**

The methods perform really well for unsupervised methods, the data is also not so clustering friendly, we can see that in the supervised methods that has some trouble with similar stats.

