DATA ANALYTICS –1

CLUSTERING PROJECT

SUBMITTED BY:-

Anjali Bhatnagar 2019201012

TASKS UNDER CONSIDERATION

The data that is given to us for this project contains features of about 18k football players. The number of features available to us is 89. The main aim of this project is to perform several visualisation and clustering tasks on the given data and analyse the clusters based on various parameters to gain more insight about clustering approaches and analyse the data efficiently.

Further after we have performed the given 4 tasks we will analyse the clusters formed from each of them and finalise on one approach.

Flow of the Project :

- Data Analysis
- Visualisation of data using Matplot and Seaborn
- Drawing conclusions from the visualizations
- Outlier analysis and Trend analysis using the visualizations
- KMeans Clustering Algorithm
- Analysis of Kmeans
- Hierarchical Clustering and Analysis
- DBSCAN and Analysis
- Analysis results
- Comparison of Clustering Algorithms Used
- Conclusion

DATA DESCRIPTION

Drawing conclusions from the data which we will be useful for data visualization and clustering

The total no of data rows is 18207. The following are the various features of the data that are available to us . However we wont be using all of the data which is given to us for clustering as we will be using the numeric attributes for the purpose of clustering.

Though we cannot use all of these attributes for the clustering problem at hand we can use several of these features for the 1st task at hand I.e the visualization task in which we will use the seaborn and matplot libraries to draw graphs and infer results from it.

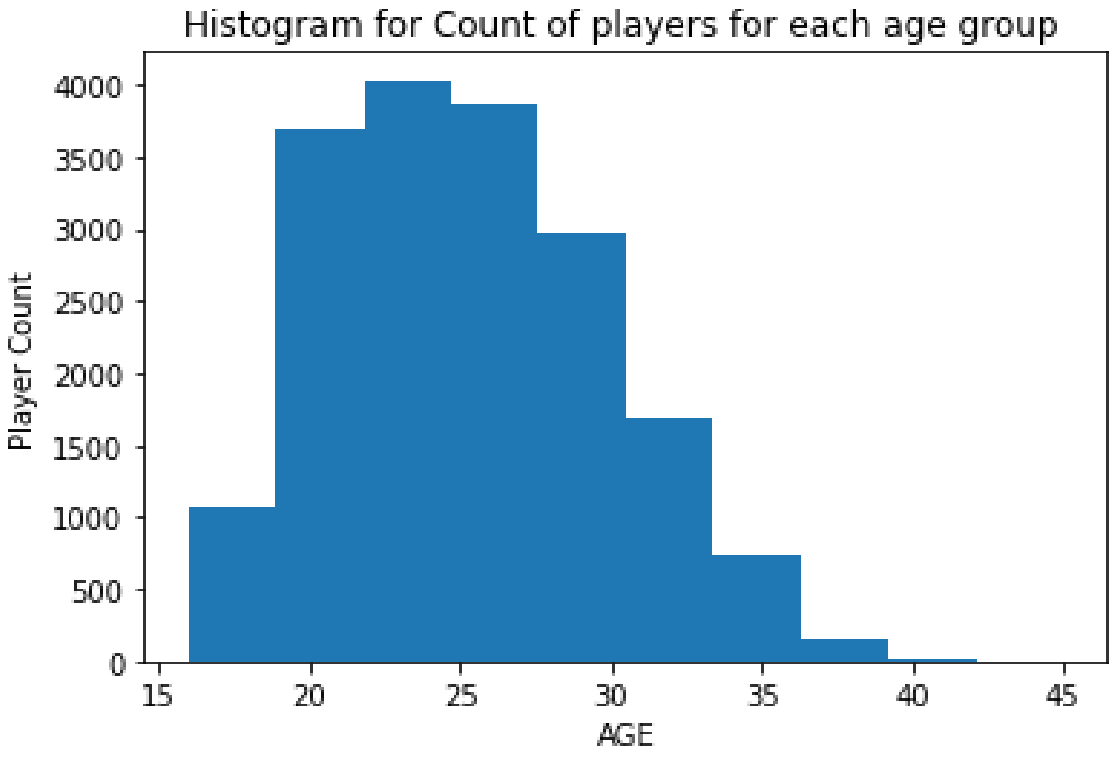
TASK 1 – DATA VISUALIZATION

The major focus of task1 is to derive visualizations from the given football data and use it to draw further inferences from the data. We have used several visualizations to have a better understanding of the problem statement at hand So lets perform the various visualizations as per the flow of the task problems.

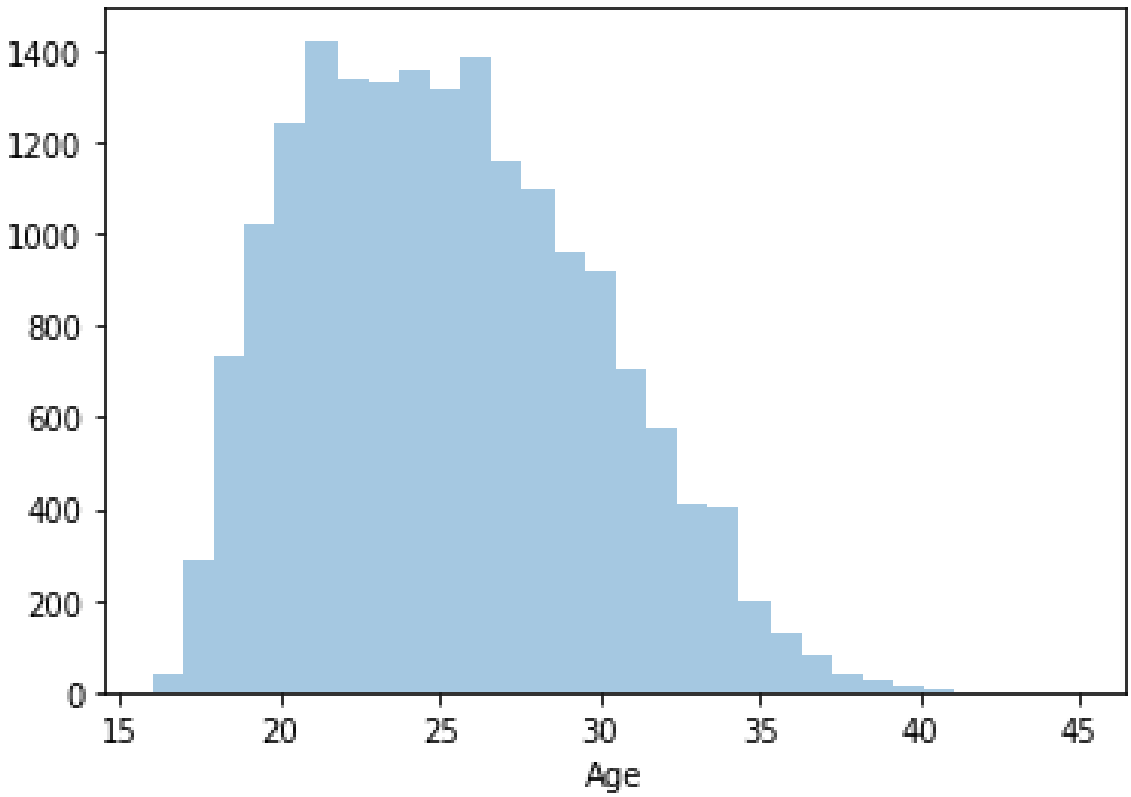
1.1 PLOTTING HISTOGRAMS and BAR GRAPHS FOR VARIOUS ATTRIBUTES

We begin this task with performing histograms for various attributes that we have .

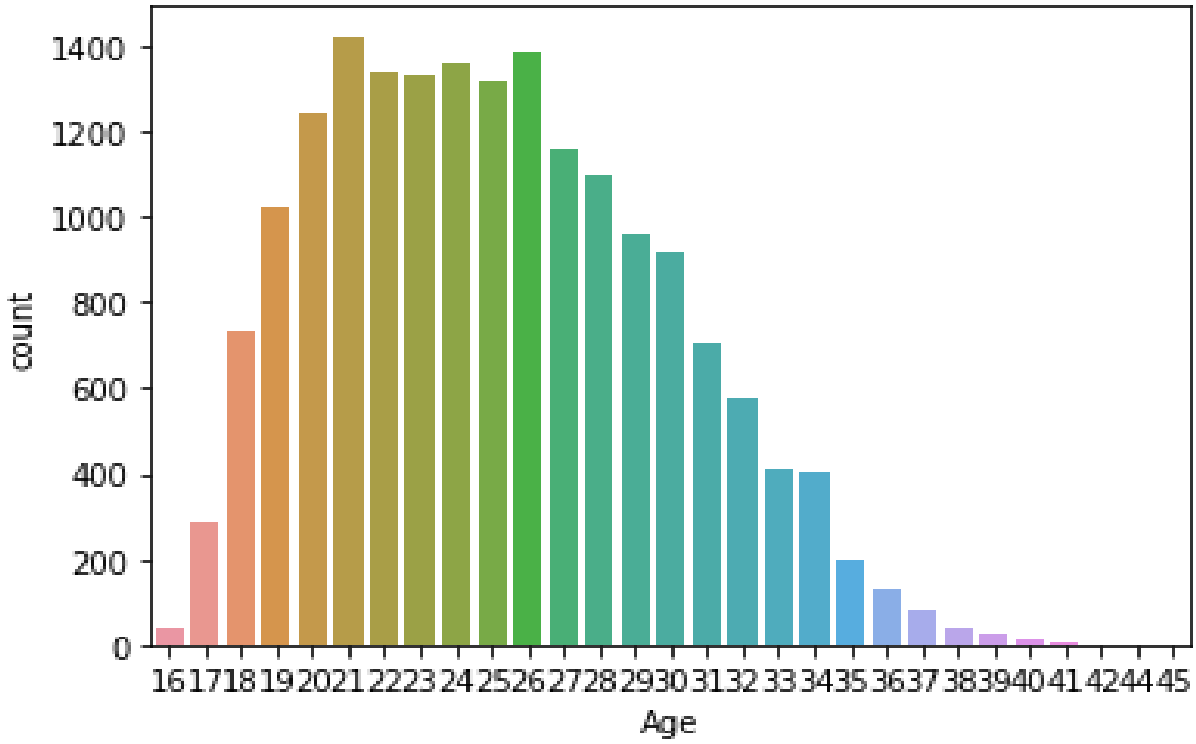
- Plotting Histogram for count of age of Players i.e. how many players fall in a particular age group



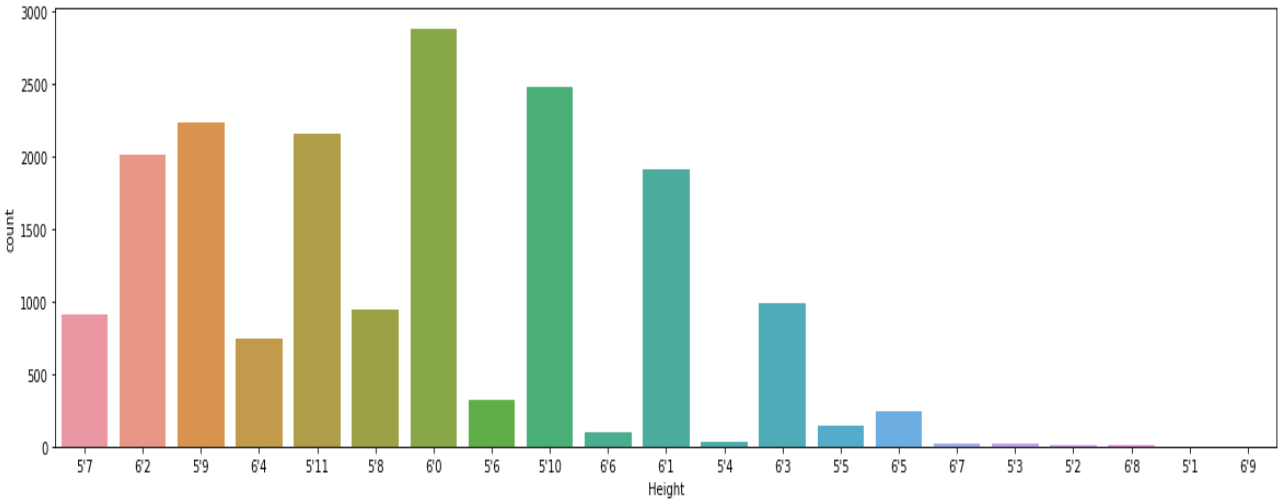
Lets change the number of bins and then see the visualizations :



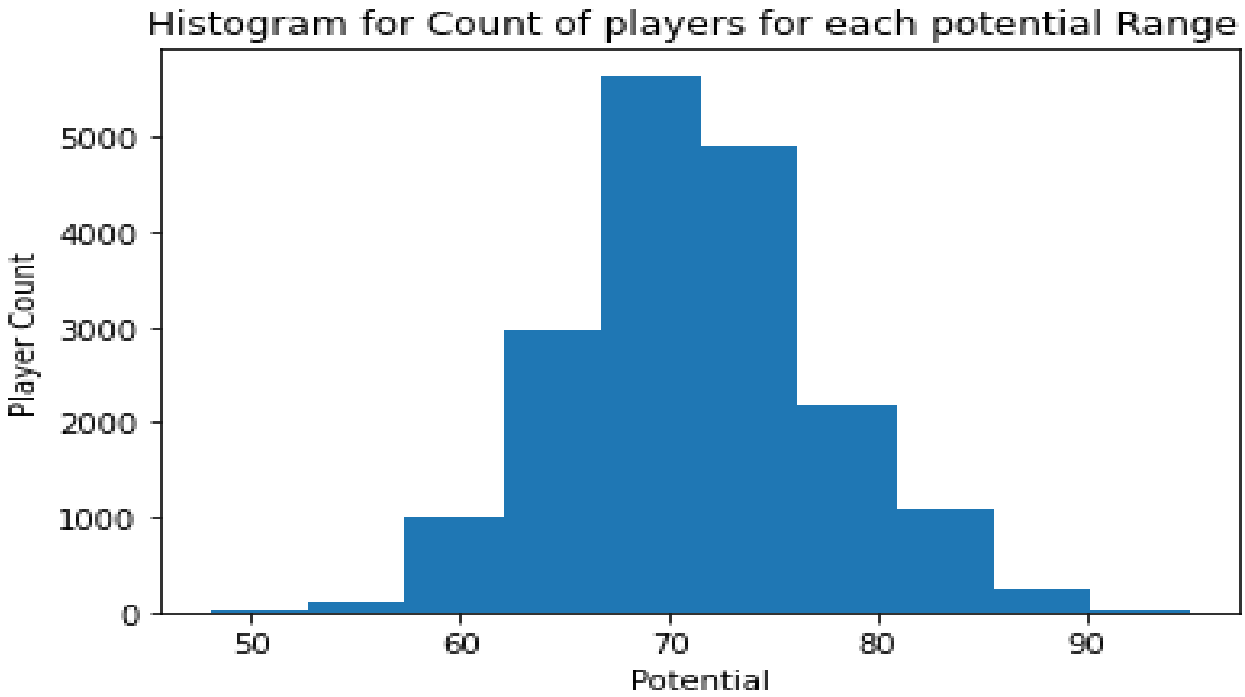
Now lets try to plot a bar graph for the same :



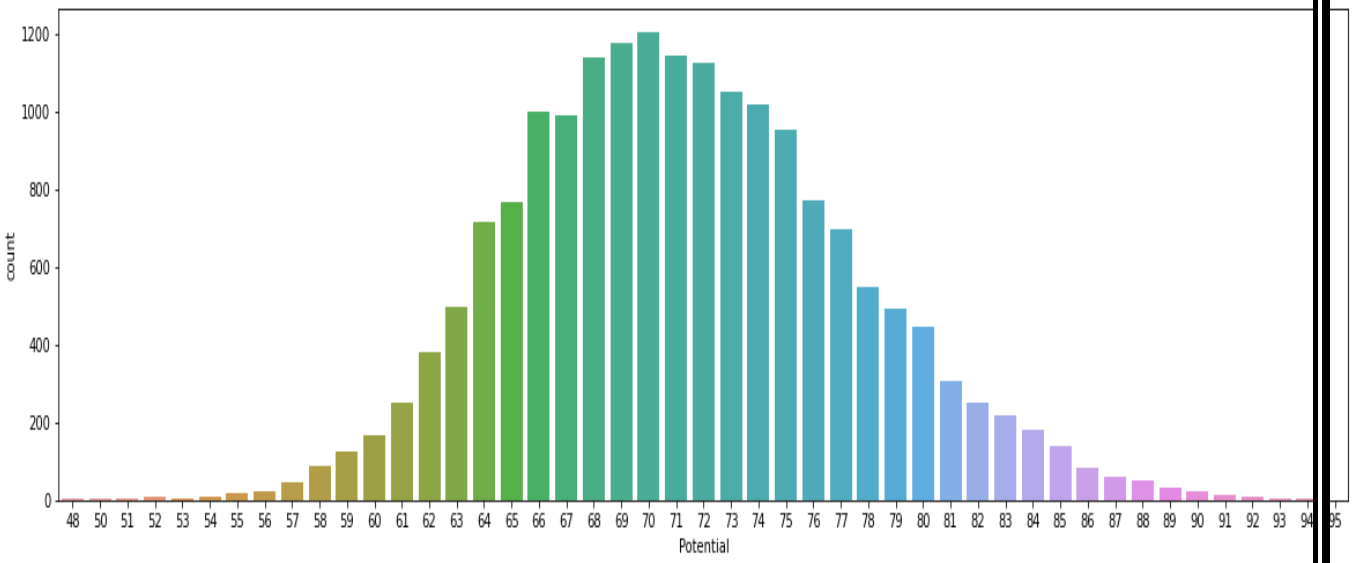
So after plotting for age lets plot for the attribute **height** :



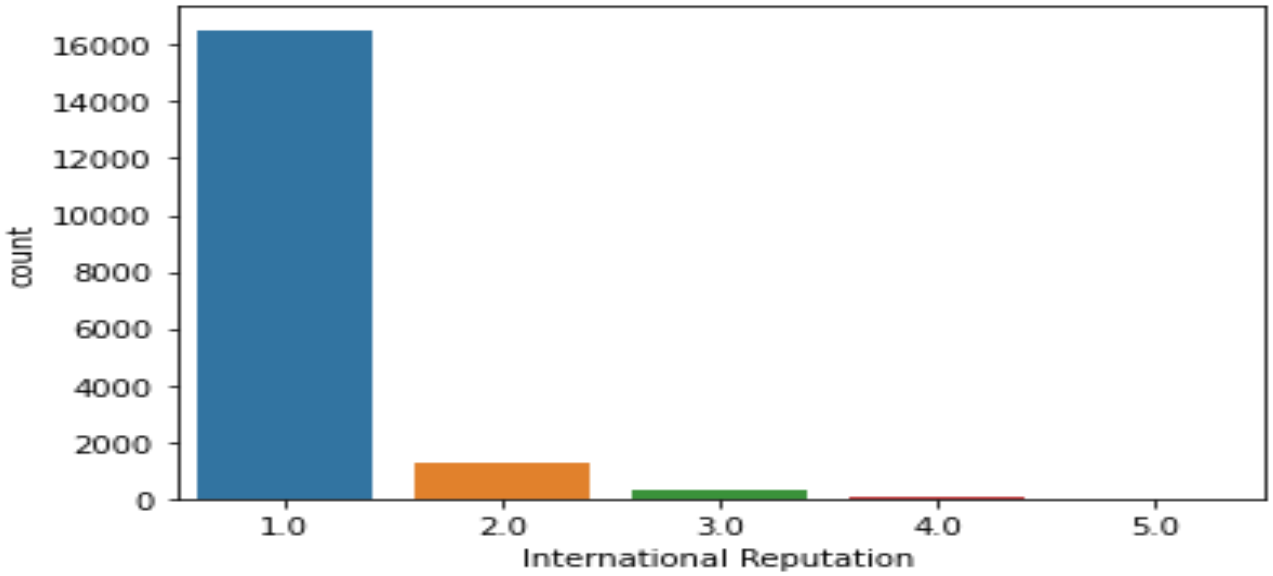
Now we will plot histograms for count of players for each **potential** range



Plotting a bar graph for the same :



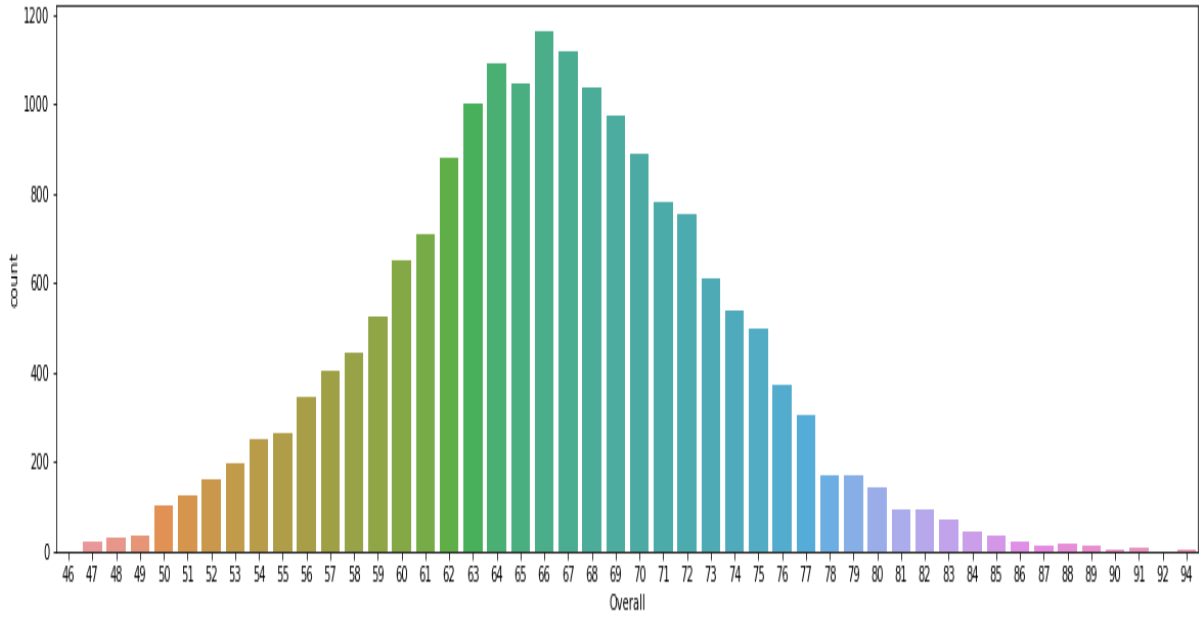
Plotting Bar Graph representing the International Reputation of Players



This graph also shows us that very few players have an international reputation of 4.0 which further suggests for the presence of outliers like Ronaldo and Messi in the football players data.

1.2 Outlier Analysis Using Bar Graphs Based on Values of Attributes

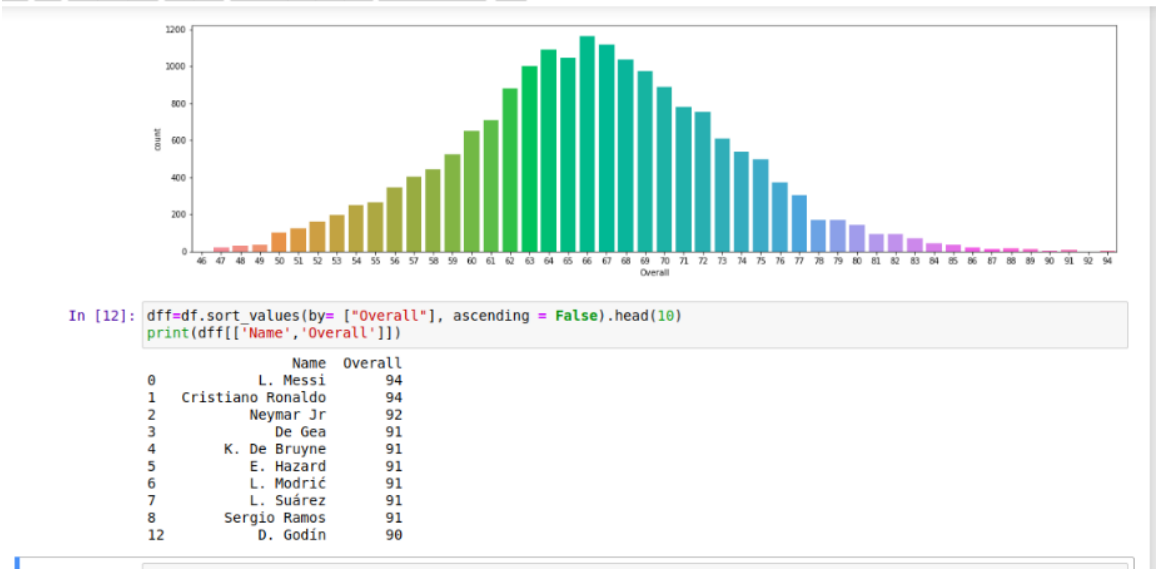
Now lets try to exploit the attribute Overall Ratings which will also show us the presence of outliers amongst the players .



Barcelona captain Messi held a higher base overall (94 rating) than Juventus superstar Ronaldo (93 rating) in FIFA 20. This can be seen from this graph . Hence we can conclude that we can use visualization techniques to find outliers.

Though we can check this out from the data given to us as well.

Lets have a look at this screenshot of the analysis done.

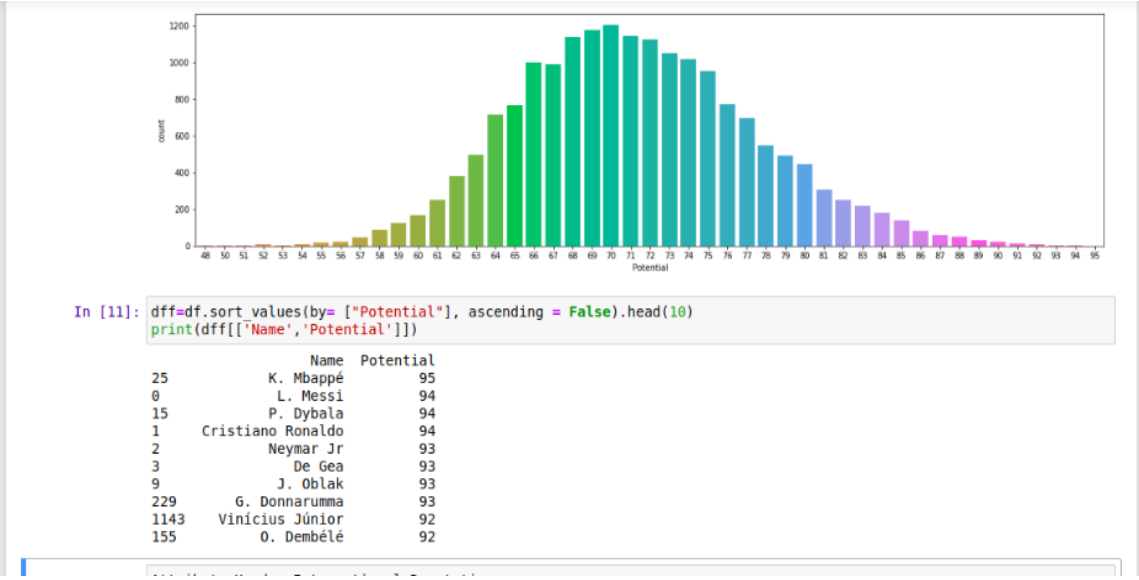


```
In [12]: dff=df.sort_values(by= ["Overall"], ascending = False).head(10)
         print(dff[['Name','Overall']])
```

	Name	Overall
0	L. Messi	94
1	Cristiano Ronaldo	94
2	Neymar Jr	92
3	De Gea	91
4	K. De Bruyne	91
5	E. Hazard	91
6	L. Modrić	91
7	L. Suárez	91
8	Sergio Ramos	91
12	D. Godín	90

Similar analysis for the attribute Potential also tells us about the presence of outliers such as Ronaldo , Messi and Neymar.

Outliers are those data points which show behaviour different from the rest of the lot. They are important in data analysis as they reveal specific information about the data which can be captured to make the full use of the data .

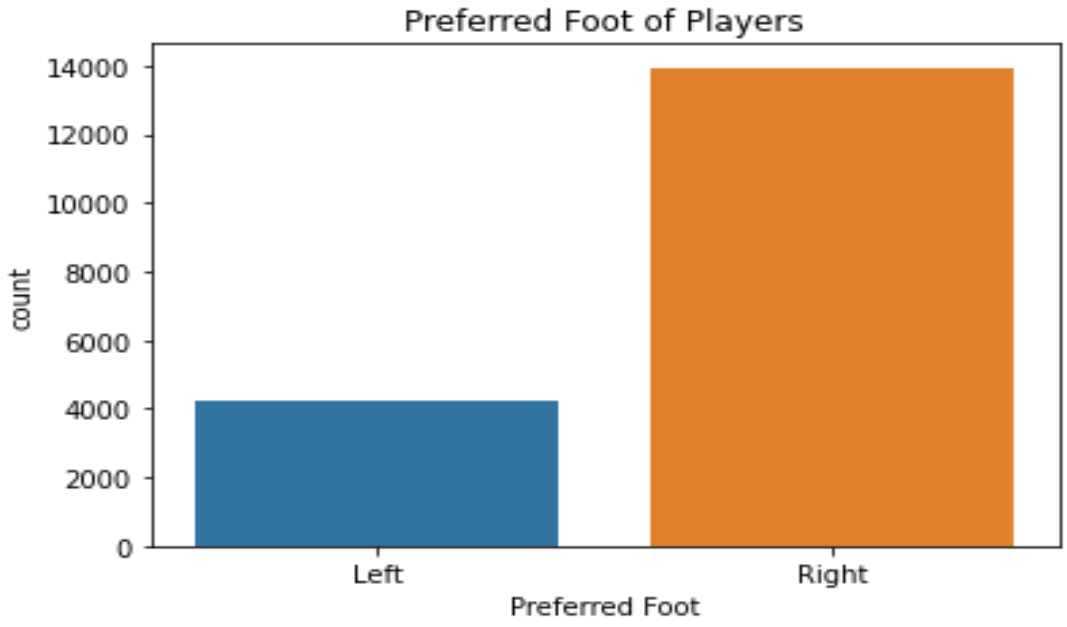



```
In [11]: dff=df.sort_values(by= ["Potential"], ascending = False).head(10)
         print(dff[['Name','Potential']])
```

	Name	Potential
25	K. Mbappé	95
0	L. Messi	94
15	P. Dybala	94
1	Cristiano Ronaldo	94
2	Neymar Jr	93
3	De Gea	93
9	J. Oblak	93
229	G. Donnarumma	93
1143	Vinícius Júnior	92
155	O. Dembélé	92

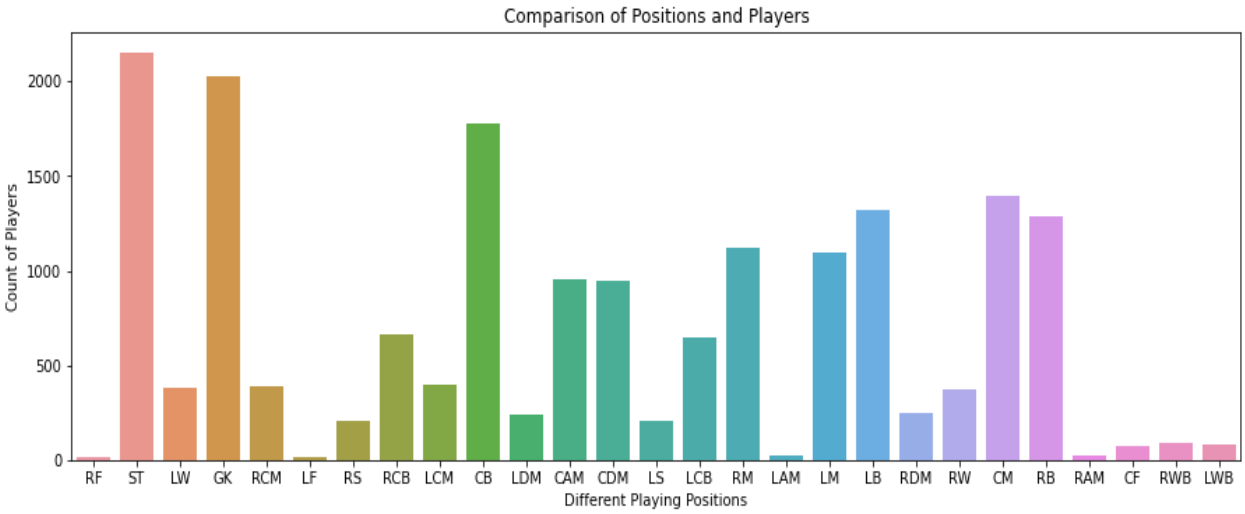
1.3 Analysing the data through visualizations based on specific Attributes of data

Plotting bar graphs for preferred foot of the players:



So from here we can infer that most of the footballers have their right foot as the preferred foot.

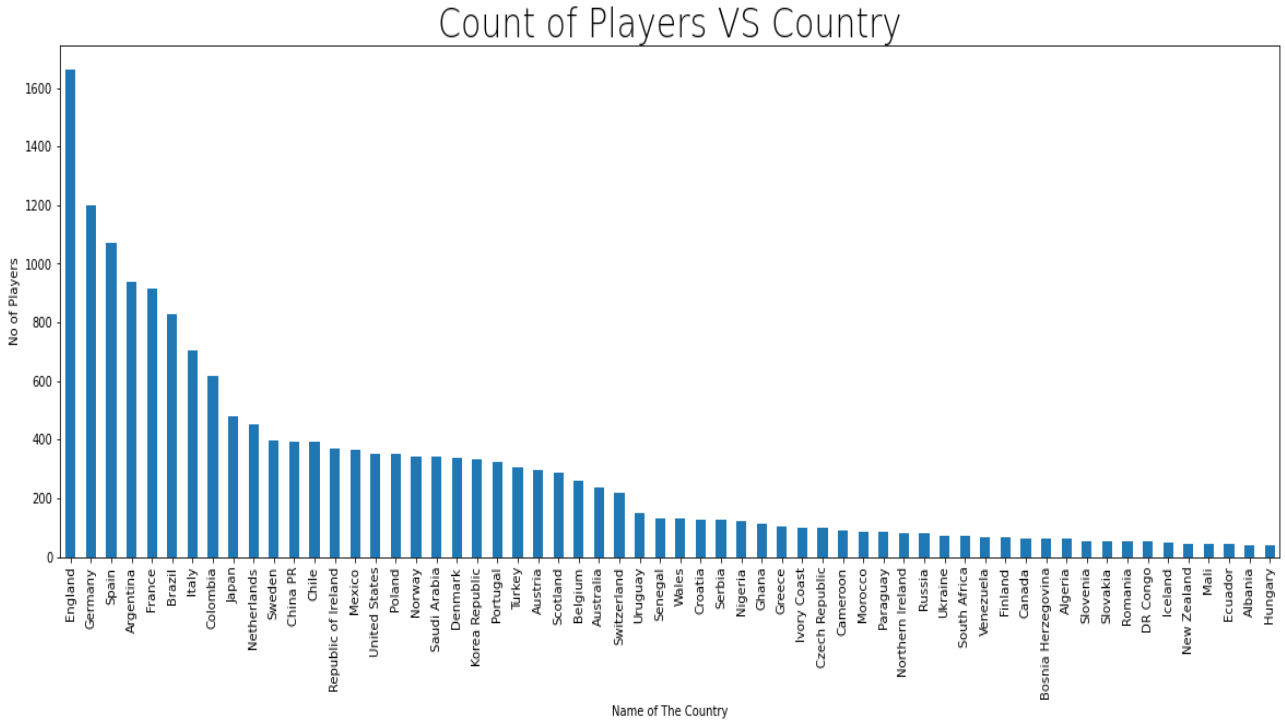
1.4 Analysis of the Positions at which the players play



So from this we get an idea of at which playing positions majority of the players play like we have very less Right forwords and left forwords compared to goal keepers in the given data.

This analysis will definitely be relevant for any team selection as we can see the general trend of players and which position is the favourite among various players.

1.5 Country based analysis of players

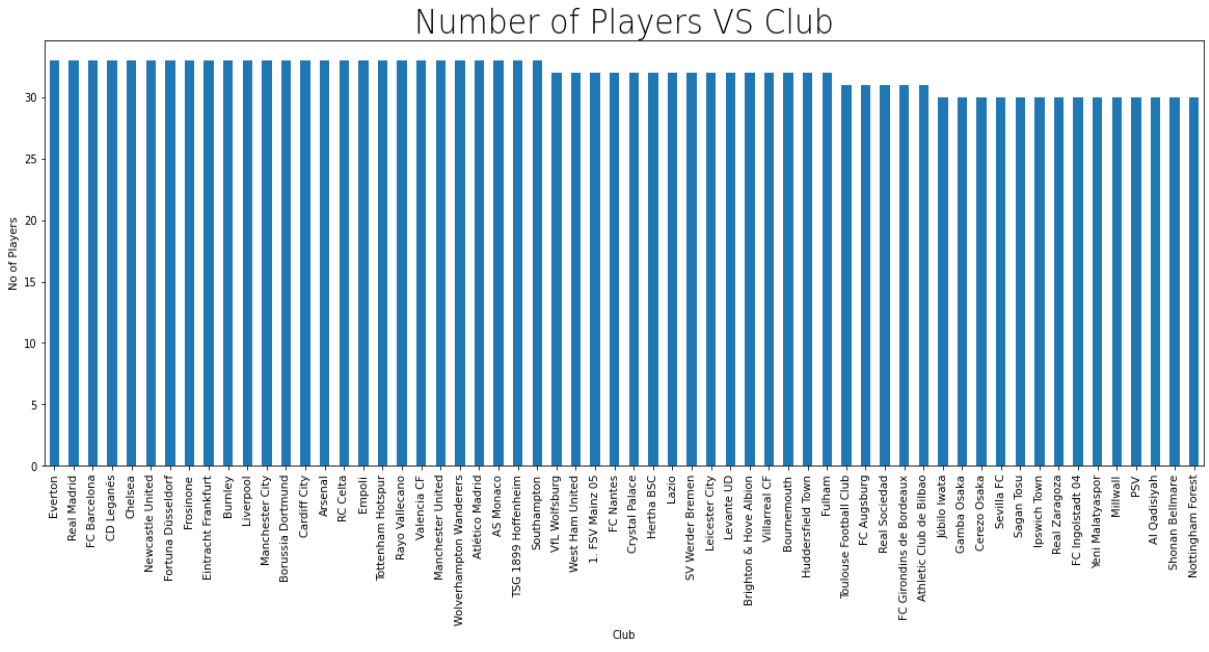


So from here we can see which country has the highest number of footballers.

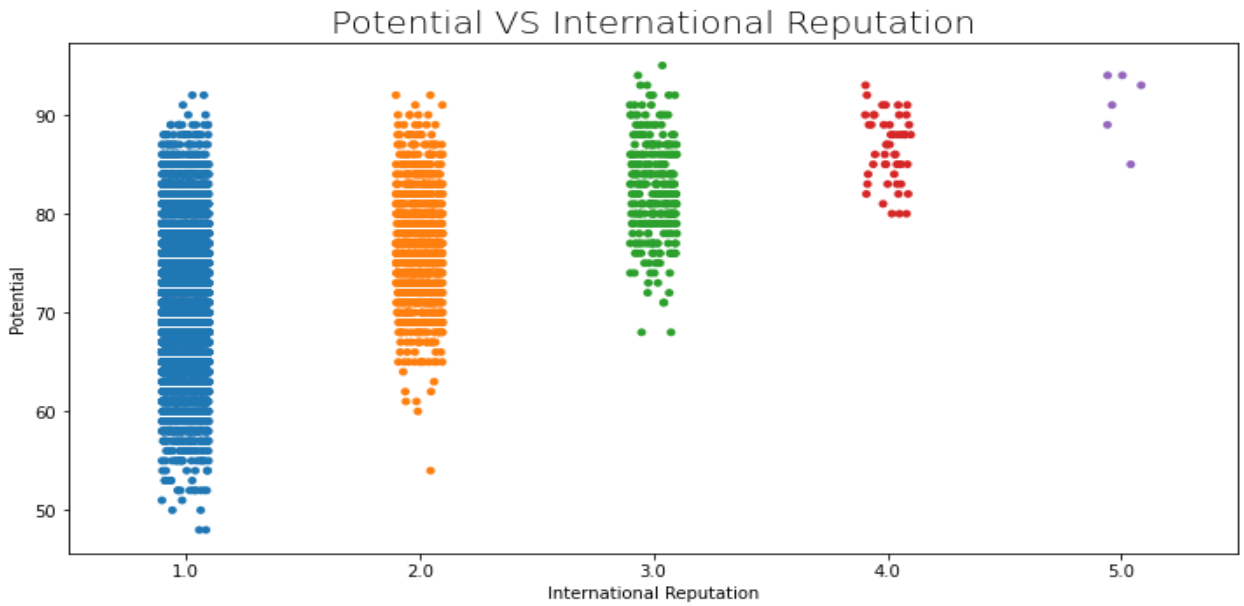
England , Germany , Spain , Argentina , Italy clearly overtake the other countries in this domain.

1.6 Club Based Analysis of Players

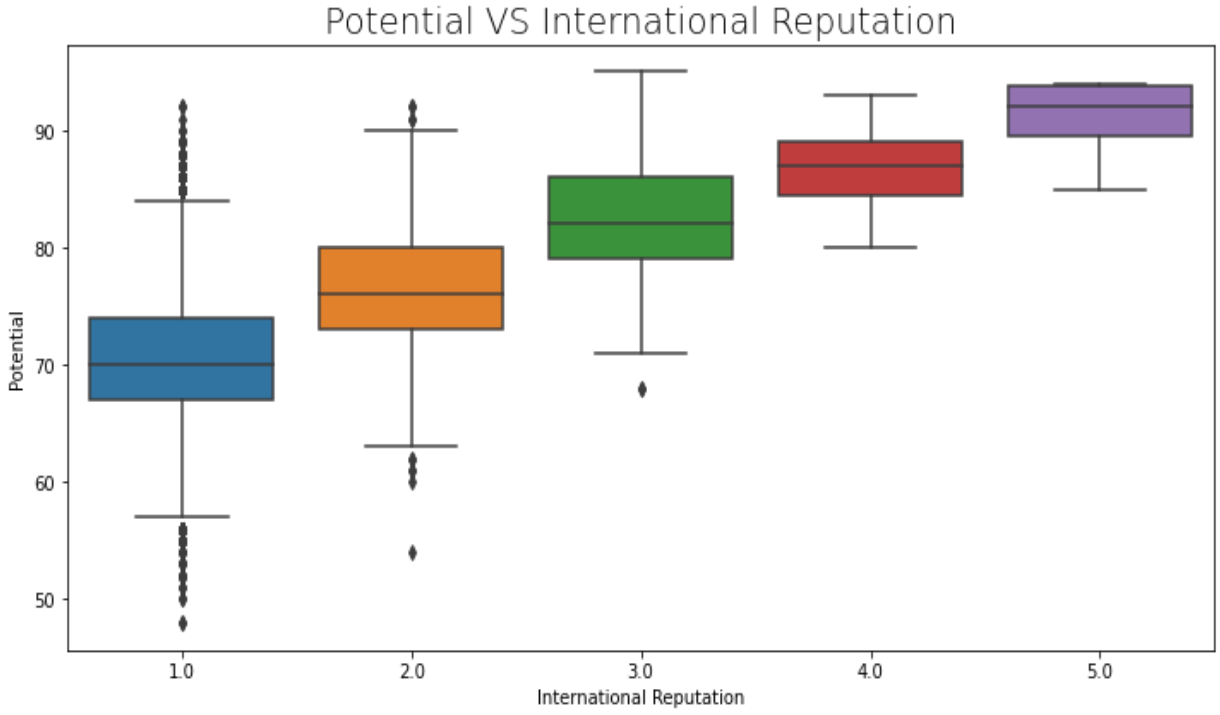
The following graph shows how many players do we have from each club majorly in the given data to be analysed.



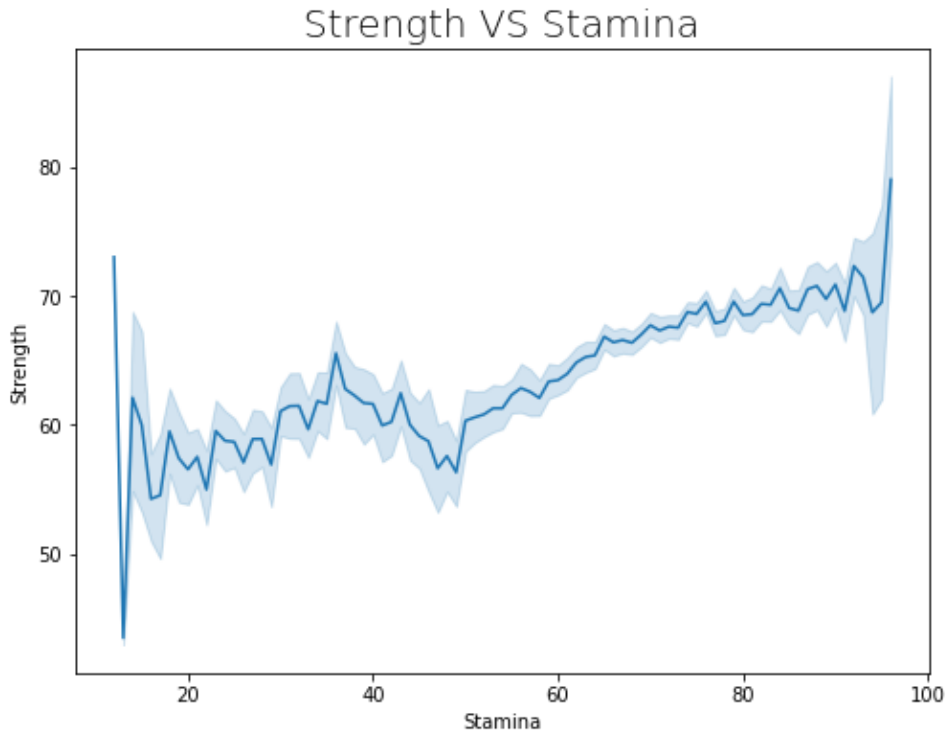
1.7 Plots showing relationship between various attributes of the football players

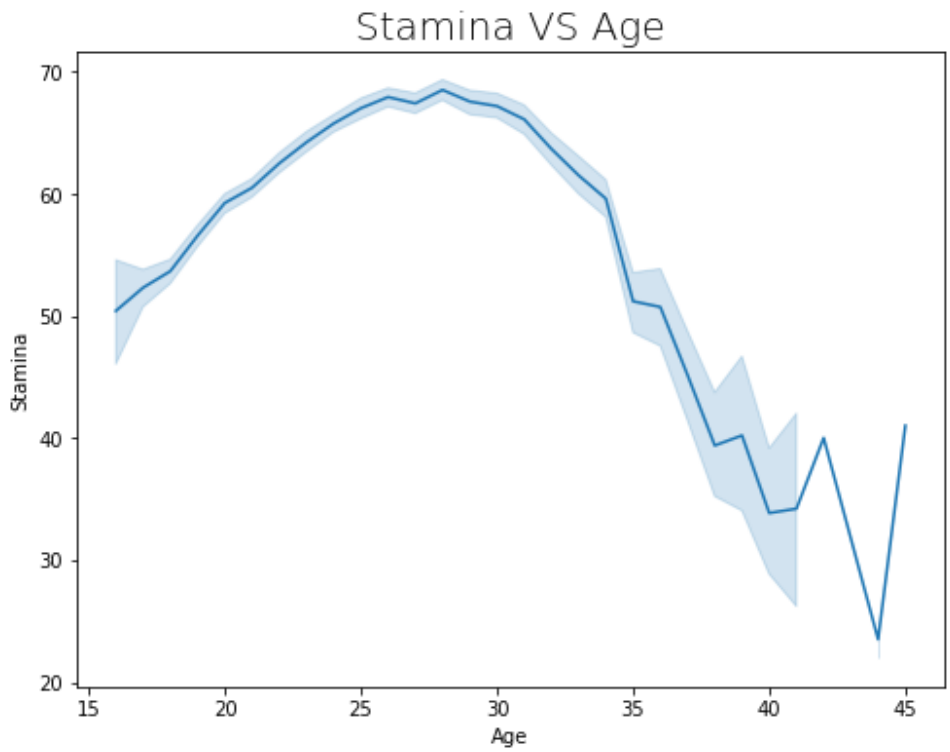


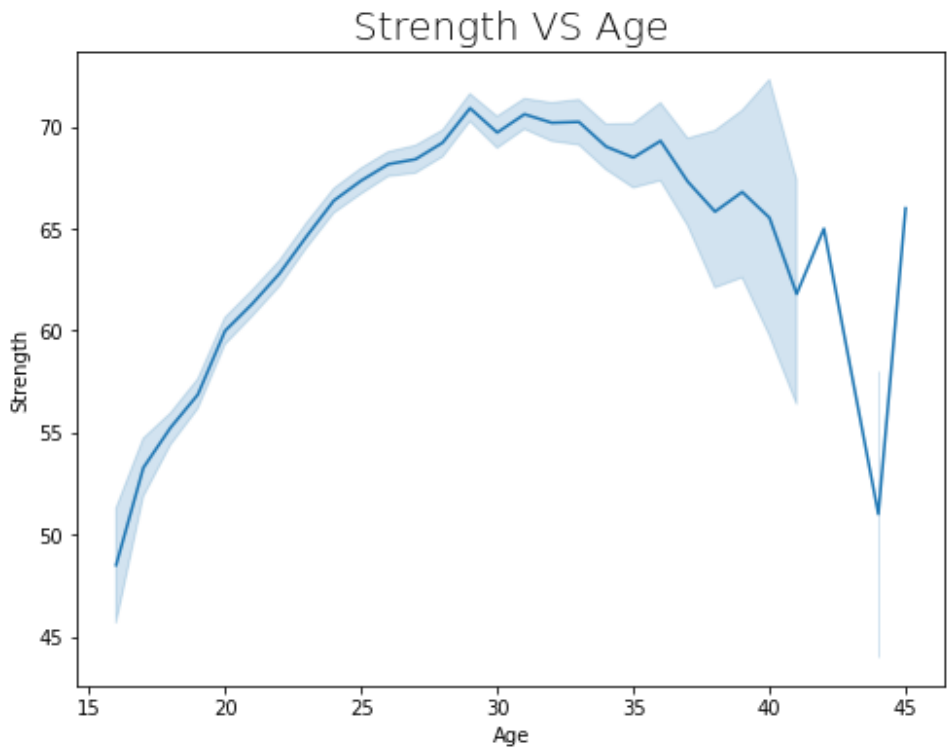
The outliers can be seen from here also as we see very few players have an international Reputation of 5 and above plus such high potential scores. Now we can represent this with the help of box plot also , which will give us a further clarification regarding the outliers.



1.8 Showing the relationship between various attributes and exploiting the dependencies on these factors with the help of visualizations

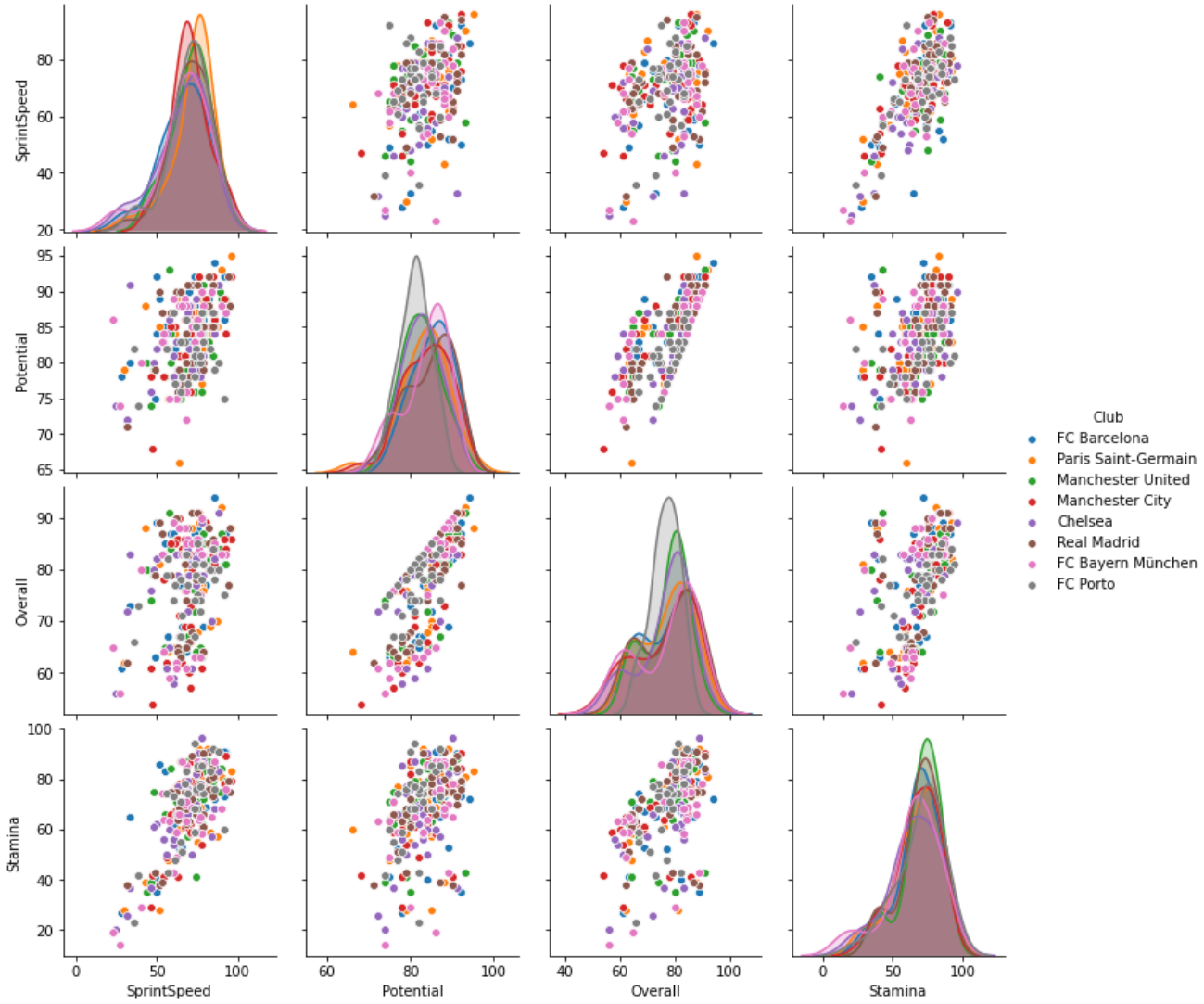






The inference drawn from these graphs is that as the stamina is more in the age group of 20 to 35 and then decreases . Similar conclusions can be drawn from the strength parameter as well. From the strength VS stamina graph we can see that strength and stamina are highly dependent on each other.

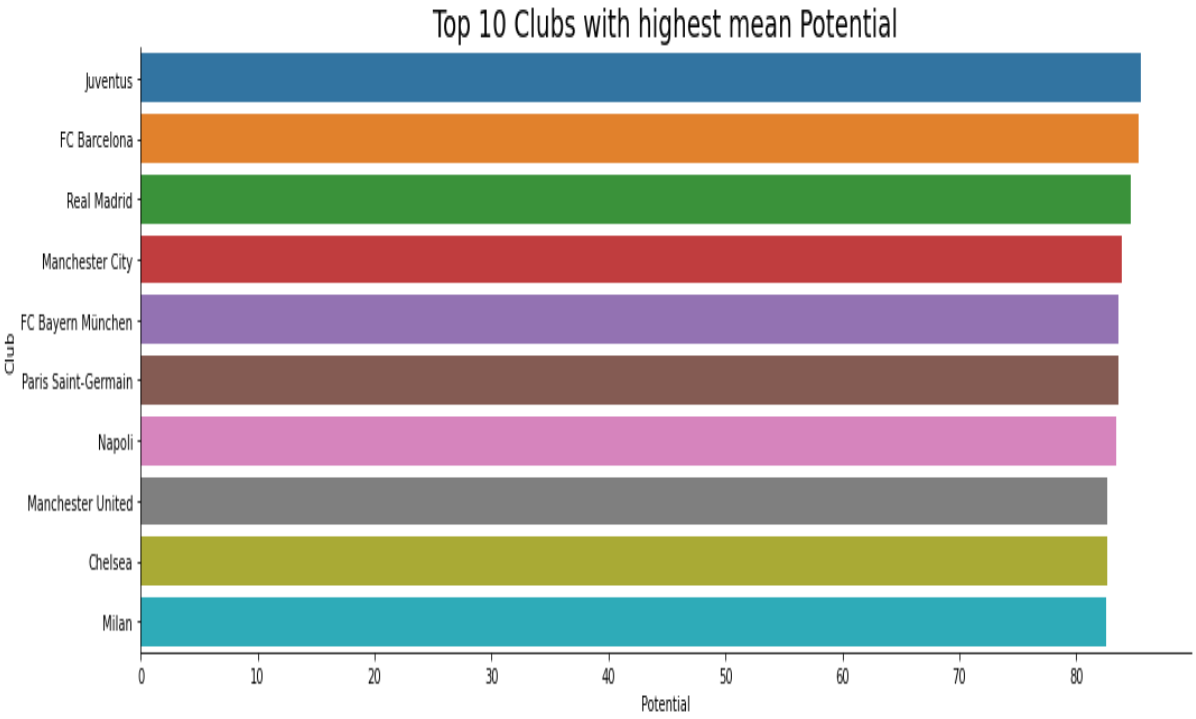
1.9 DISTRIBUTION OF PLAYERS IN DIFFERENT CLUBS BASED ON CERTAIN ATTRIBUTES

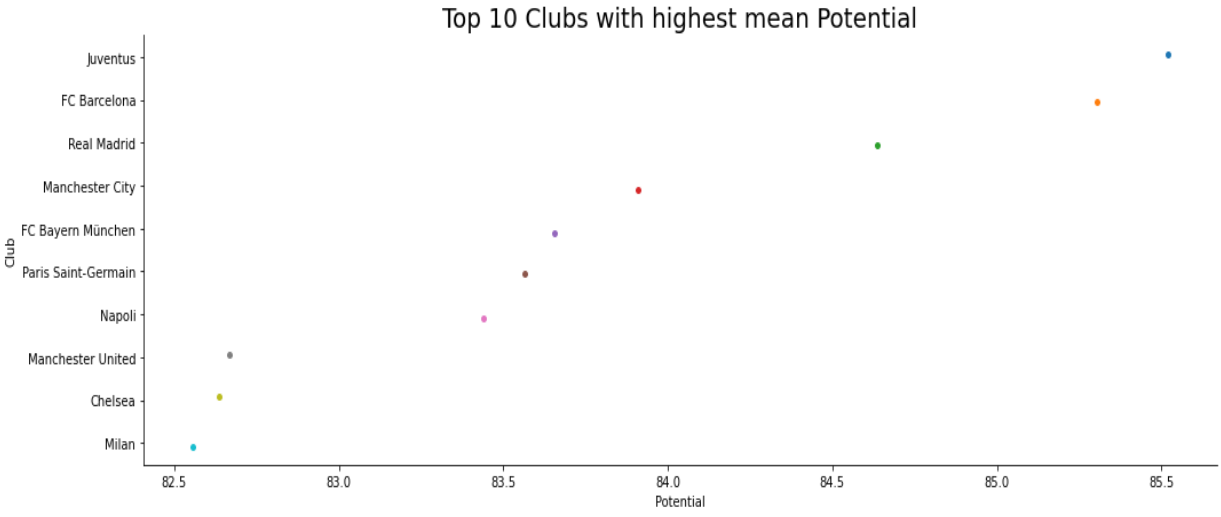


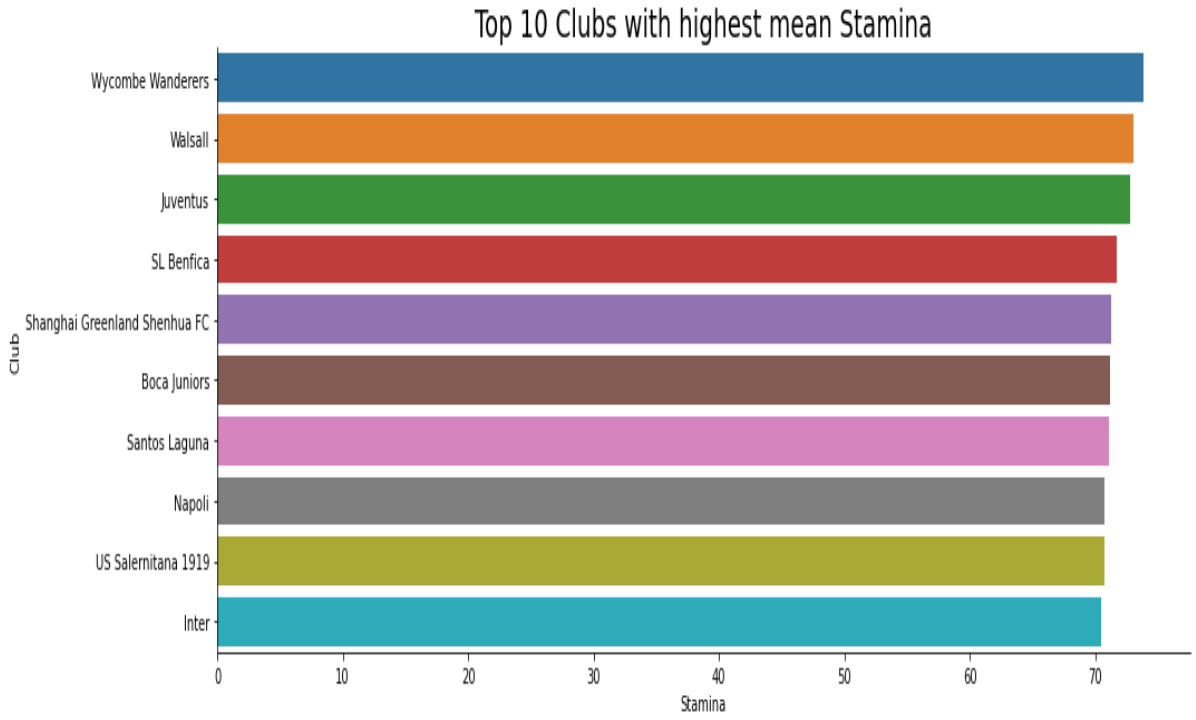
Showing distribution of Players in various clubs based on certain Attributes . As we have many clubs we will be considering only the top few clubs.It gives us a fair idea of the attributes and how much potential the players of these clubs have! Their stamina , rating and Sprint Speed !

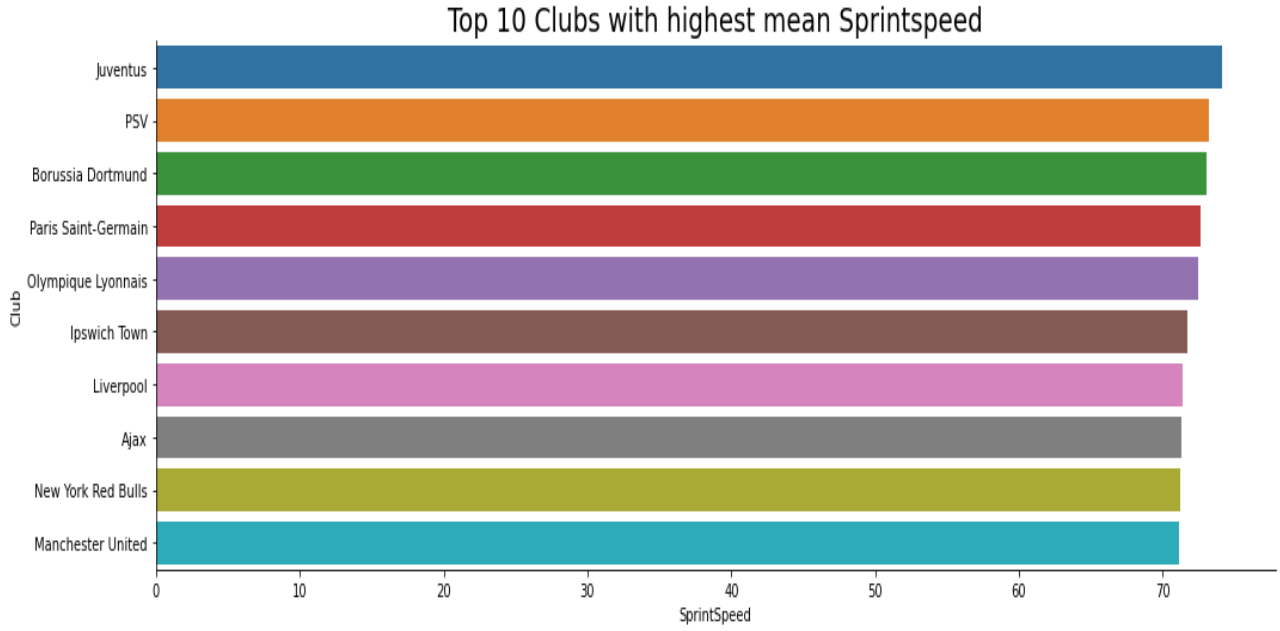
All these factors together play a very crucial factor for any club!

1.10 Analysis of Clubs based on the features









Conclusion of Task 1:

This task clearly helped us to learn about various data visualization techniques and gather various insights about the data which we have represented above.

TASK 2 – K MEANS

TASK 2.1: IMPLEMENTATION OF KMEANS FROM SCRATCH

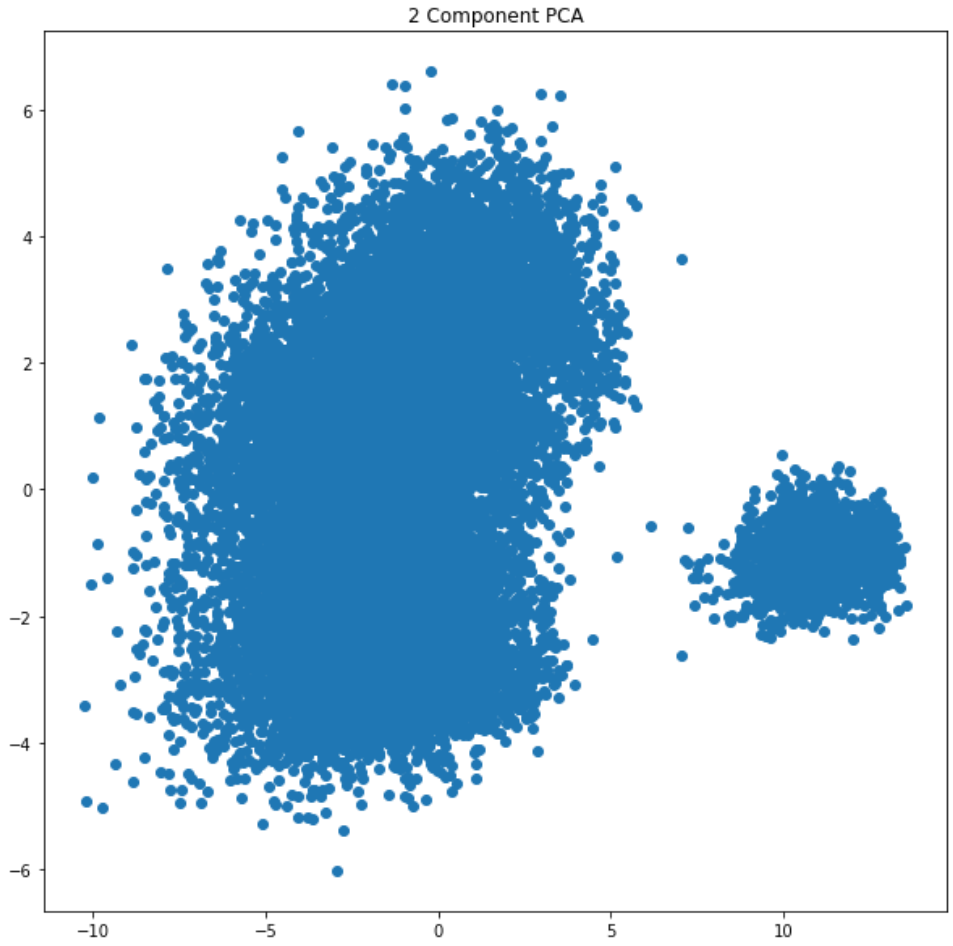
```
In [38]: def kmeans(X,centres,k):
             #find the distance from each of the centre and then add in the list corresponding to which the distance is mini
             clusters=list()

             for i in range(k):
                 clusters.append(list())#clusters formation

             centers_new=copy.deepcopy(centres)
             #print(centers_new)
             for i in X:
                 dis=np.sqrt(np.sum((centres-i)**2,axis=1))
                 dis=list(dis)
                 #we get the distance of each element from each centre
                 ind=dis.index(min(dis))
                 clusters[ind].append(i)
             n=X.shape[0]
             m=X.shape[1]
             #print(n)
             #print(m)
             #print(temp.shape)
             for i in range(k):
                 temp=np.array(clusters[i])
                 centers_new[i] = np.mean(temp, axis=0)
             #print(centers_new)


             return centers_new

In [39]: df=df.dropna()
```

We chose random centres for these clusters and then applied the Kmeans algorithm till the centres converged.

```
while(error!=0.0 and i<max_itr):
    centres=kmeans(X,centres,3)
    error = np.linalg.norm(centres-old)
    print("error",error)
    old=copy.deepcopy(centres)
    i=i+1
```

Now we will run this Kmeans algorithm for various values of k i.e 3, 5, 7 and stated in the 2nd part of Task 2 and plot these values.

As we had many attributes we used PCA for our data processing.
Data after application of PCA:



TASK 2.2: CHOOSE K=3,5,7

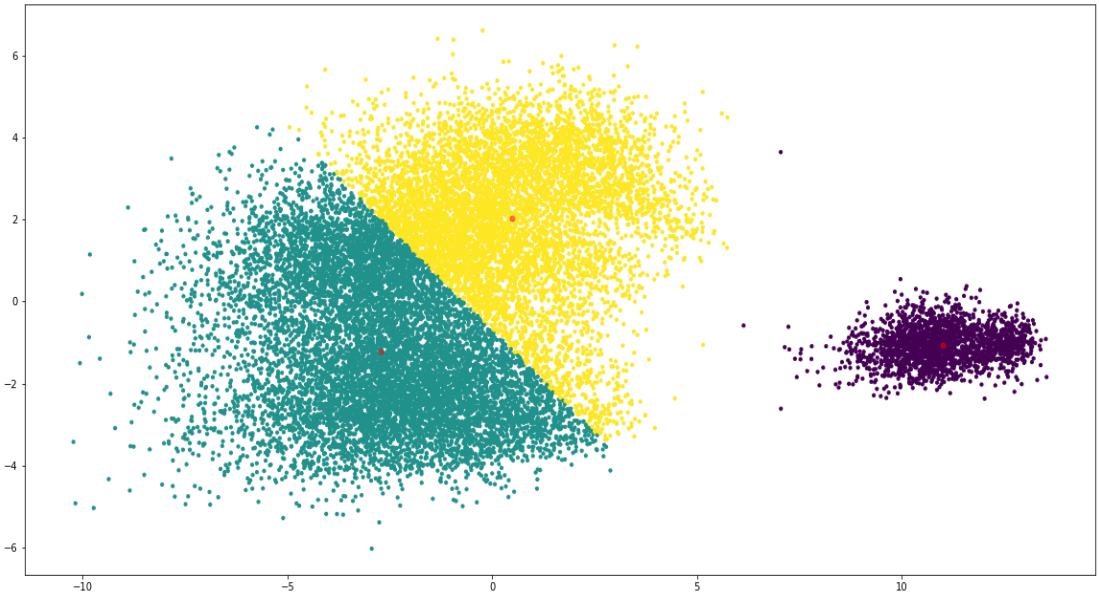
Now we will run this algorithm for different values of k and see the result .

Earlier in the beginning we took all the attributes for the purpose of clustering but later dropped the number to 2 using Principal Component Analysis so that we could easily plot the points and have a better visualization of the results.

Also the time taken i.e the number of iterations taken for kmeans to converge was comparatively higher for the one when we took all the attributes compared to when we used only the principal components.

One more thing that we noticed was that as we used KMeans and the Euclidean Distance Measure the shape of the clusters so formed was spherical.

K=3

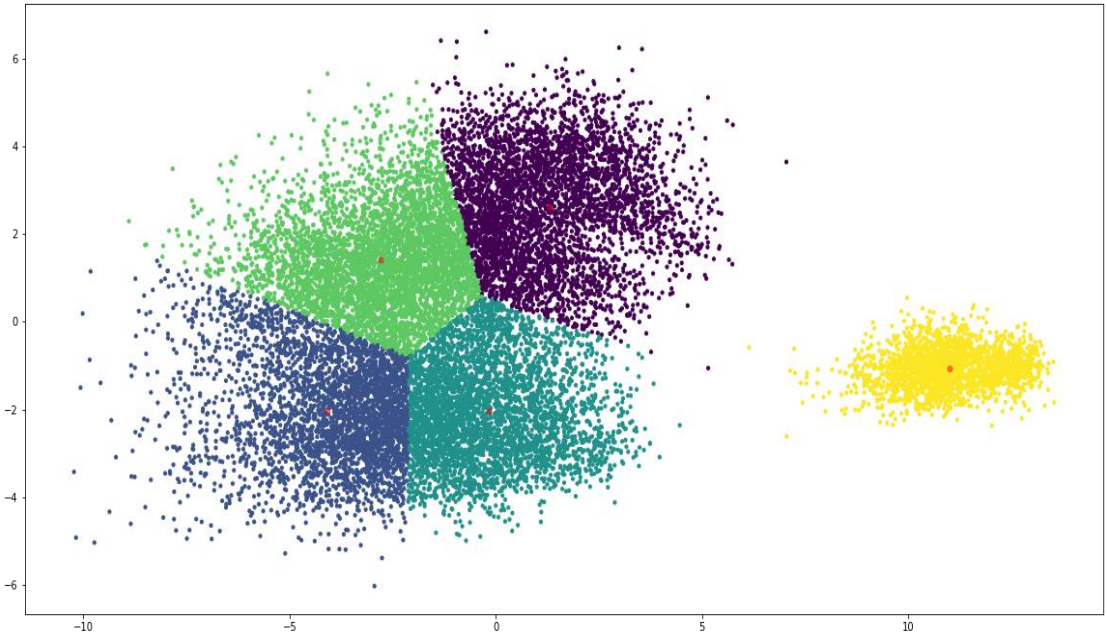


No of clusters : 3

No of iterations it took to converge : 28

The red circles represent the centres of the clusters.

K=5

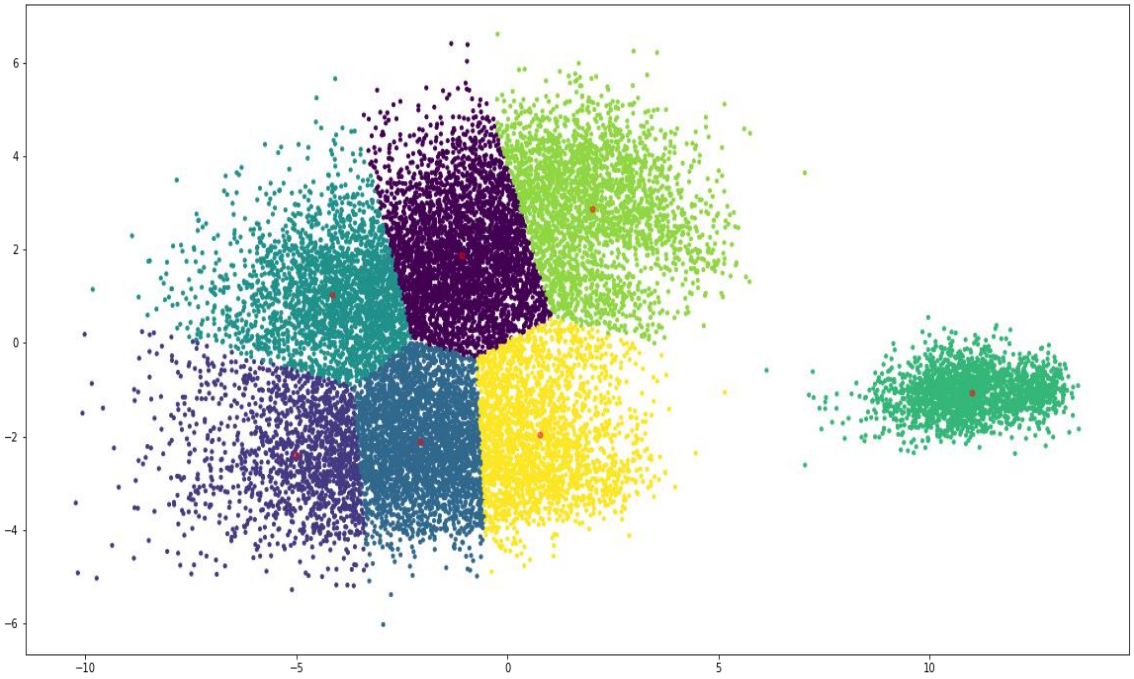


No of clusters : 5

No of iterations it took to converge : 25

The red circles represent the centres of the clusters.

K=7



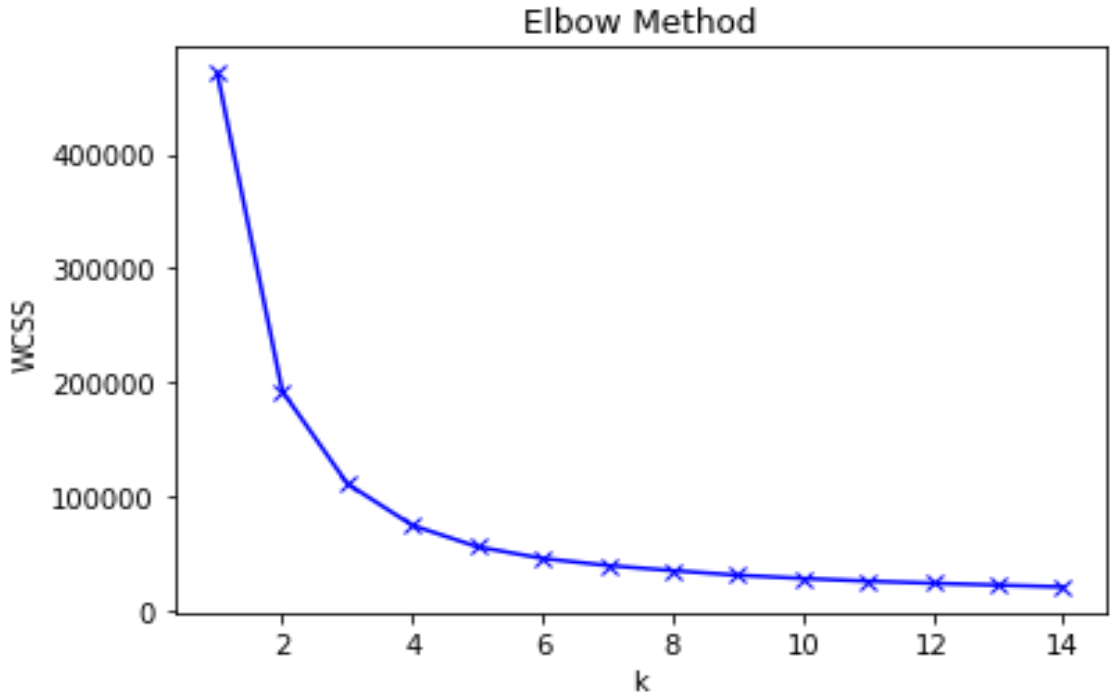
No of clusters : 7

No of iterations it took to converge : 27

The red circles represent the centres of the clusters.

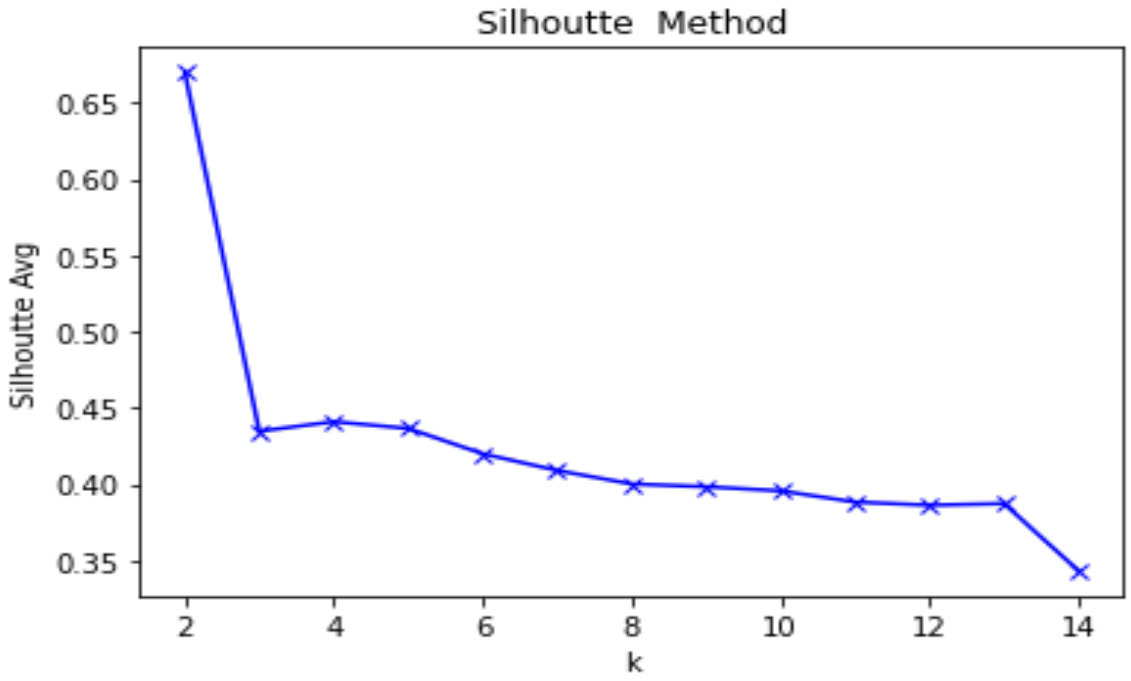
TASK 2.3 : Use elbow method and Silhoutte Score to get optimal number of clusters

Using Elbow Method to find the optimal number of Clusters :



From this the optimal value of k is 4.

Using the Silhoutte Score to find the optimal Value of K :



So from the Silhoutte Method too we get the value of k as 2.

TASK 2.4: ANALYSIS OF CLUSTERS WE GOT IN EACH CASE AND MARKING THE CENTRES

As we saw from above we have marked the centres for all the 3 different values of k .

Now lets analyse these clusters that are formed.

So we will analyse the clusters on the basis of intra-cluster distance and inter-cluster distance.

Intra Cluster Distance: The distance between the points of the same cluster

Inter Cluster Distance: The distance between the centres of different clusters.

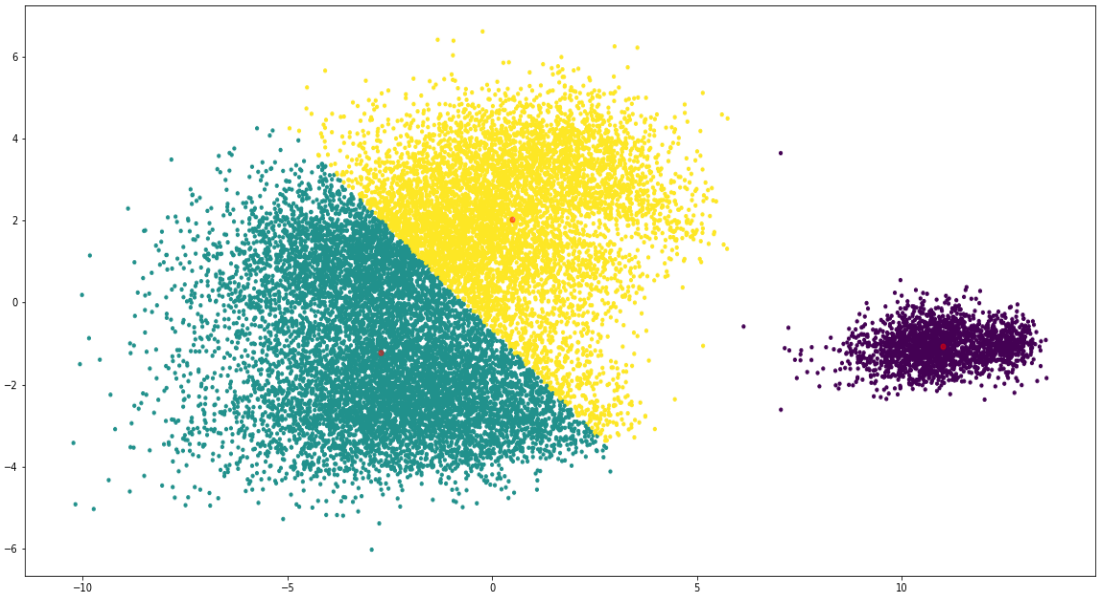
The less the Intra Cluster Distance and the more the Inter Cluster Distance the better is the Clustering Approach.

For k =3

Intra – Cluster Distance : [8.503537156380618, 13.764599590632827, 11.90316356889481]

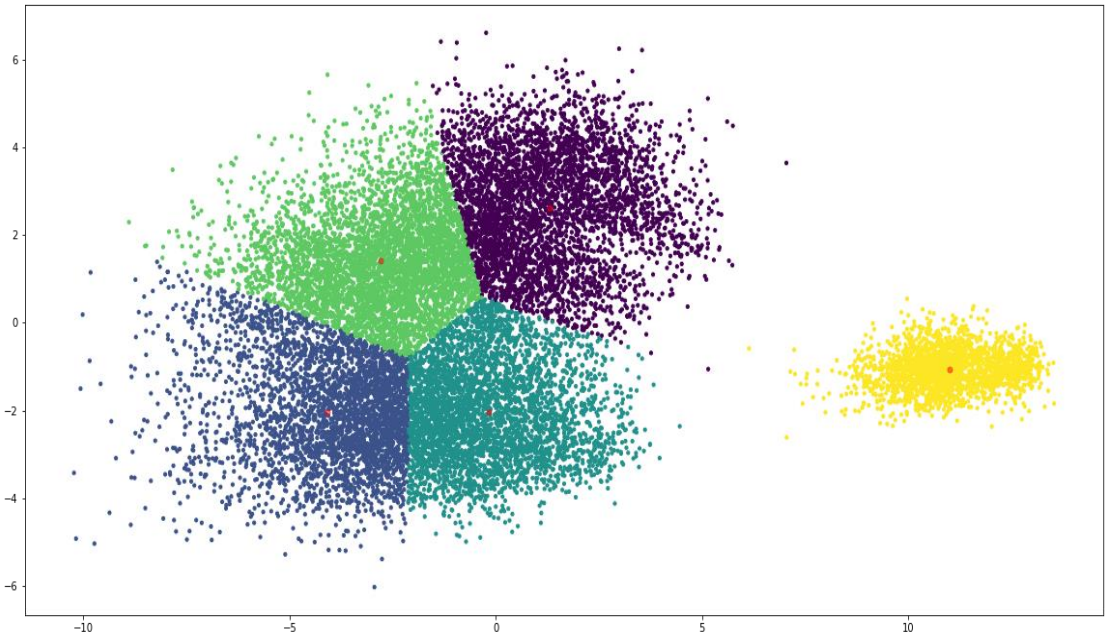
Inter Cluster Distance: 9.747433199000215

Lets have a look at the Clusters Plotted and gather further insights from it.



The 3 clusters seem pretty well formed , where the cluster being represented by purple is strictly different and has a few outlier. Though the clusters represented by green and yellow are quite difficult to be differentiated and the points at the boundary could be equidistant from each of the centroids with a minimal difference.

As said in the task we have plotted the centres of the clusters which can be clearly seen in red colour.



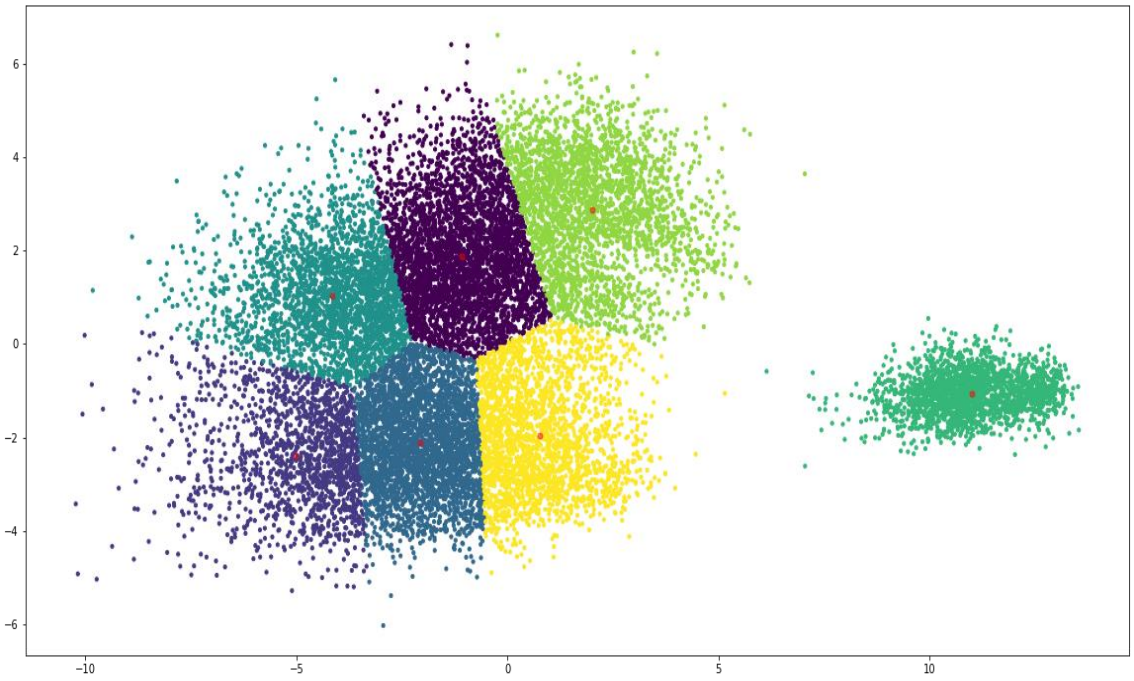
For k =5

Intra – Cluster Distance: [9.89121106198221, 9.94682566554108, 6.811003733415643, 8.72820578412055, 7.5067894484585995]

Inter Cluster Distance: 7.8916465874035

The Clusters represented by colours other than yellow yielding to a lower value of Inter Cluster Distance.

If we compare k=3 and k=5 k=3 is a better estimate !



For k=7:

Intra – Cluster Distance: [6.720538598034652, 8.348112585908922, 6.0827636109284695, 7.616387440447447, 7.5067894484585995, 7.986377287373859, 6.731041907924012]

Inter- Cluster Distance: 7.096691681547637

As we used Principal Component Analysis for the purpose of feature plotting we are seeing the features on the basis of these Principal Components only !

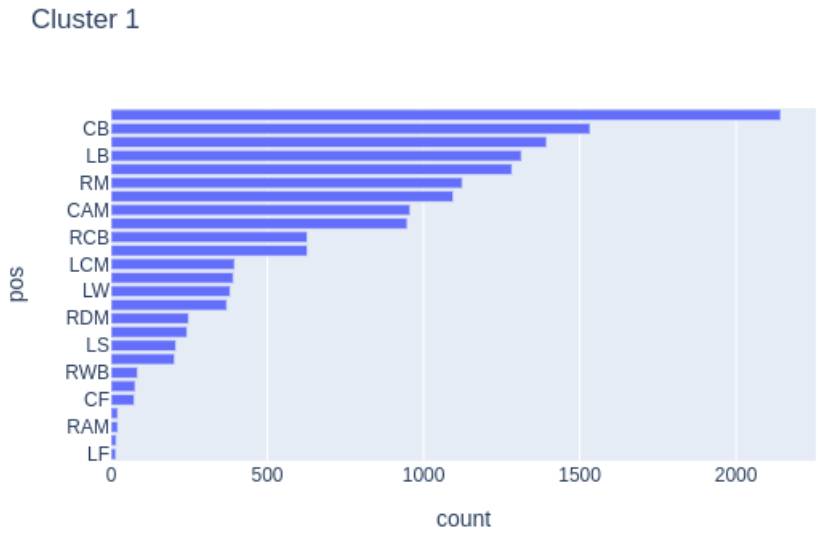
CONCLUSION OF TASK 2 :

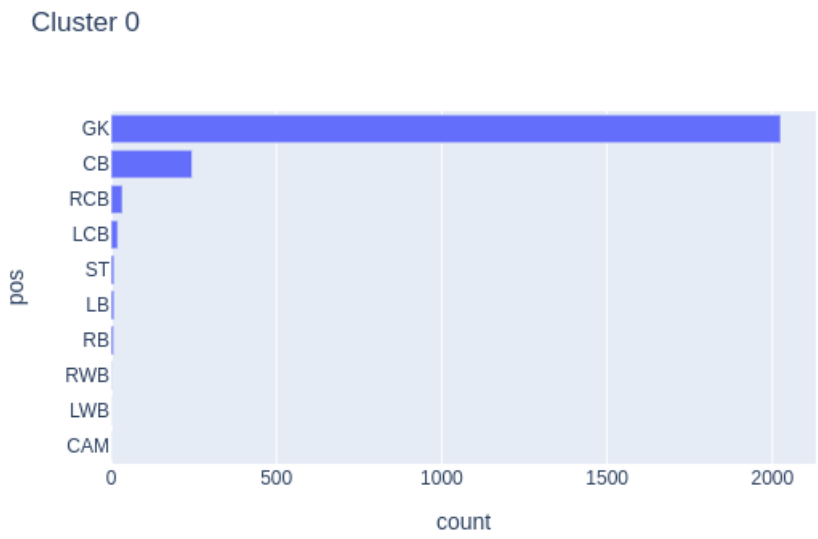
So we got the value of the K to be 2 and 4 . So lets use these values for clustering and then further analyse the results. This time we will not use PCA for clustering but PCA will just be used for the representation of the final clusters.

So lets start with our analysis for K=2

The following figures show the 2 clusters formed and we are representing these on the basis of positions as the position majorly caters to the features of the players and is a good attribute to check for in various clusters.

At the end we will be using the inferences from these results to compare all the 3 clustering methods.



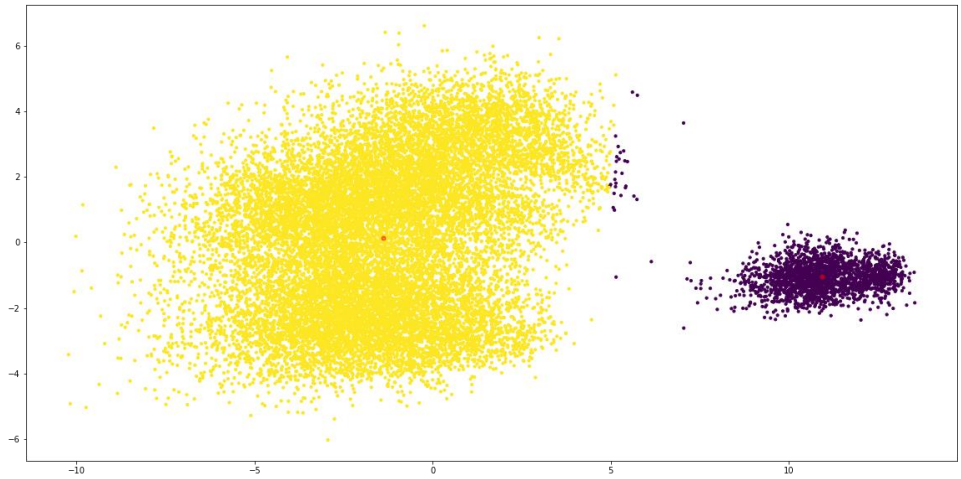


Simply looking at these 2 clusters features :

The clusters beautifully represent the players on the basis of their **positions.(Attribute)**

This also gives us an insight of **the hidden Pattern (trend)** in the data that many other features are directly or indirectly dependent on the position of the players due to which the clusters have a clear reflection in terms of the Position.

Assigning Names to Clusters :The clusters can be named on the basis of particular positions i.e Cluster 0 as Goal Keepers and Cluster 1 for Players other than Goal Keepers.

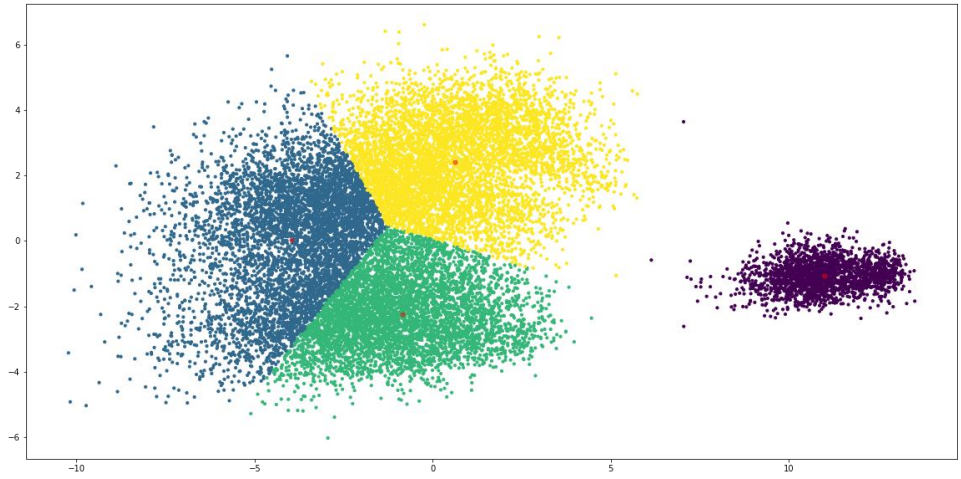


This visualization is done with the help of PCA and correctly shows how we grouped the data into 2 clusters . We see a very few outliers specifically near the yellow cluster the few purple dots which exhibit a bit different properties.

The centres of the clusters are marked by Red dot.

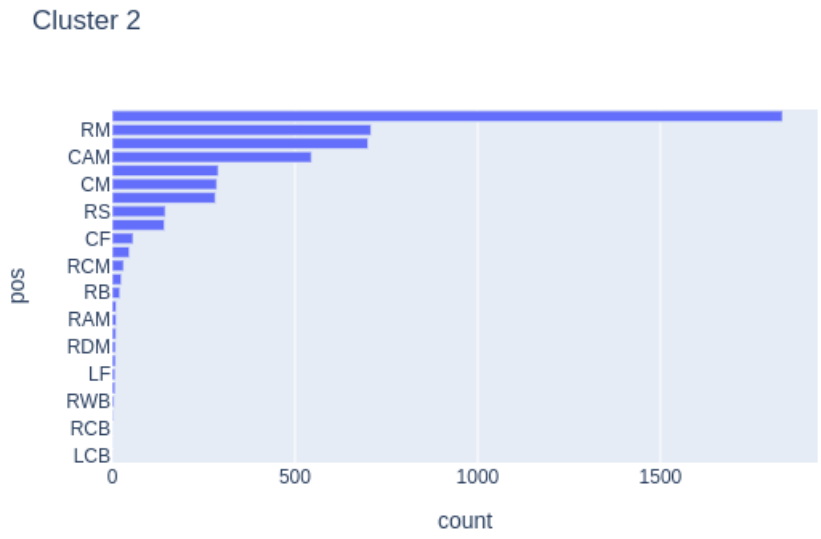
Now lets start the Analysis for K=4

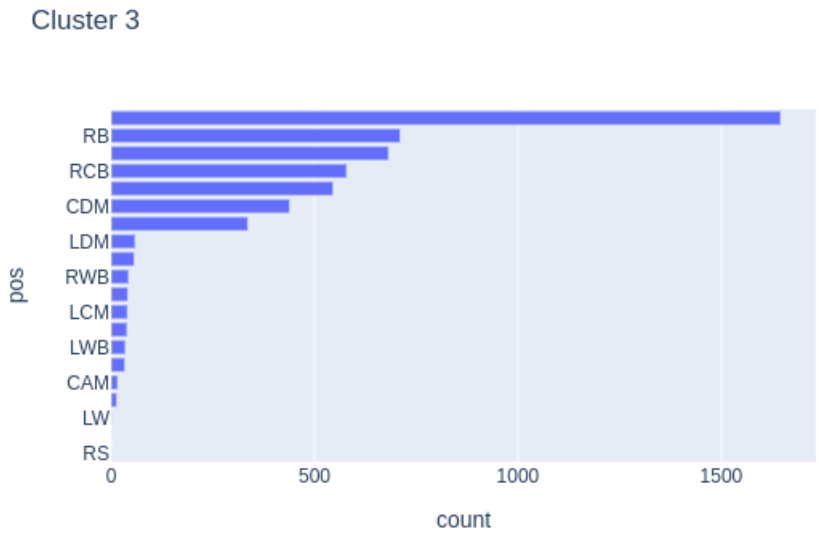
If we begin with the visualization of the clusters :

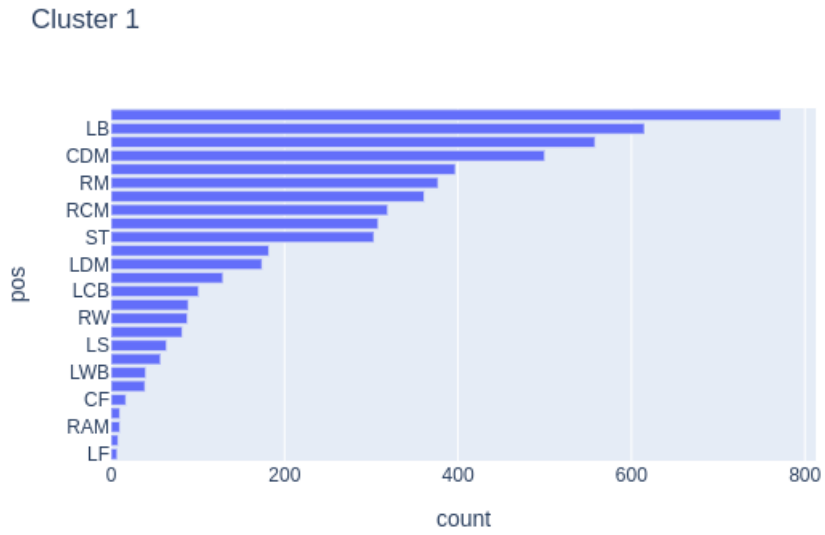


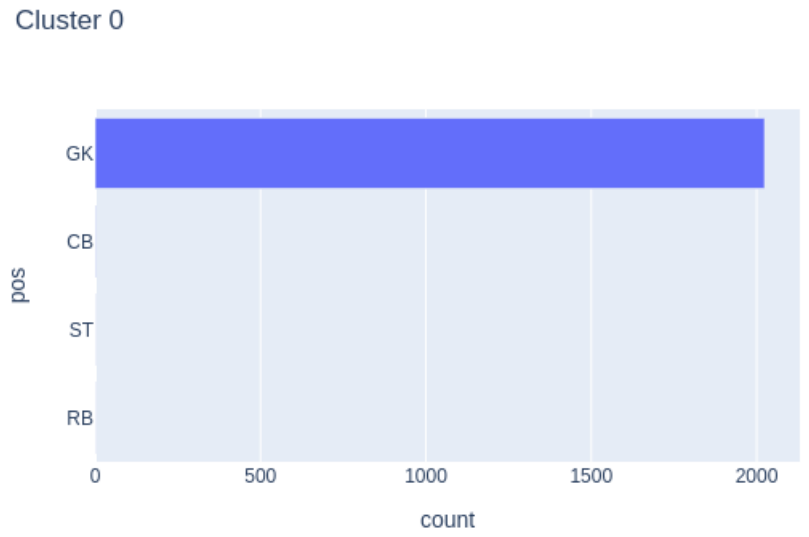
Here the purple outliers of the previous case are now assigned to the yellow cluster possibly because of the change in the position of the centres of the clusters .

Similarly we will also plot the various positions for these players as well and see the clusters formed and name them accordingly . The hidden trends are the same as we discussed for K=2.









So over here we can see that each Cluster is dominated by a particular position of the players and the other positions are just like a few outliers.

What is the most specific thing is when K=4 the Cluster 0 contains all the Goal Keepers and no other outliers as we could see for K=2 .

So we can name the 4 clusters as : Goal Keepers(Cluster 0) and the others on the basis of the other features like CDM, RB,RM i.e Midfielders , Strikers , Defenders depending on the specific attributes though we see some outliers also in these 4 clusters.

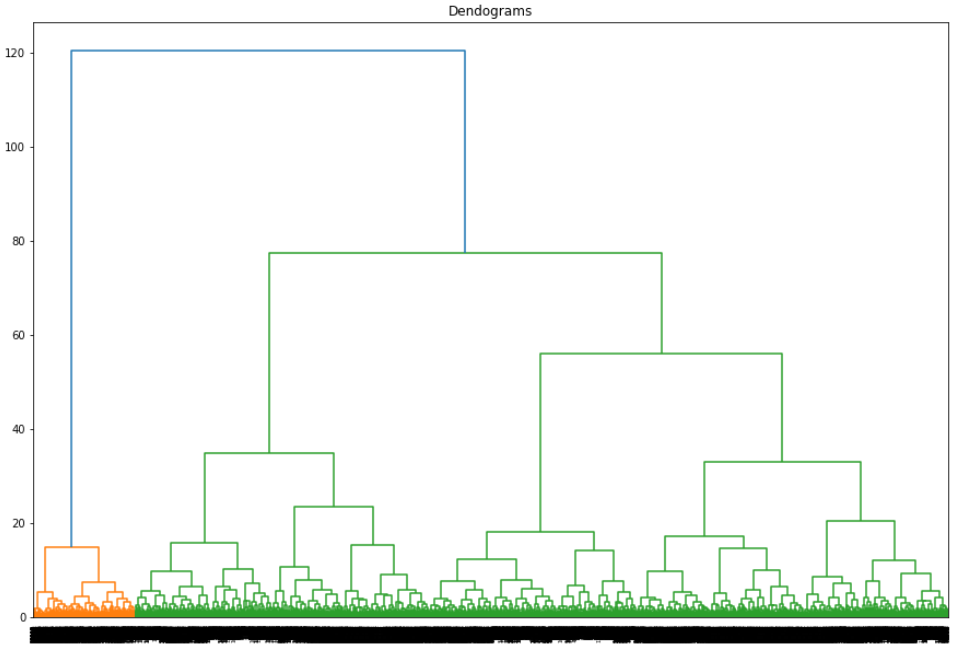
So we can choose K=4 .

TASK 3 : HIERARCHICAL CLUSTERING

3.1 Agglomerative Clustering (Bottom Up Clustering)

The **agglomerative clustering** is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also known as *AGNES* (*Agglomerative Nesting*). The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named *dendrogram*.

We used the inbuilt clustering algo and this is the dendogram for this clustering process.

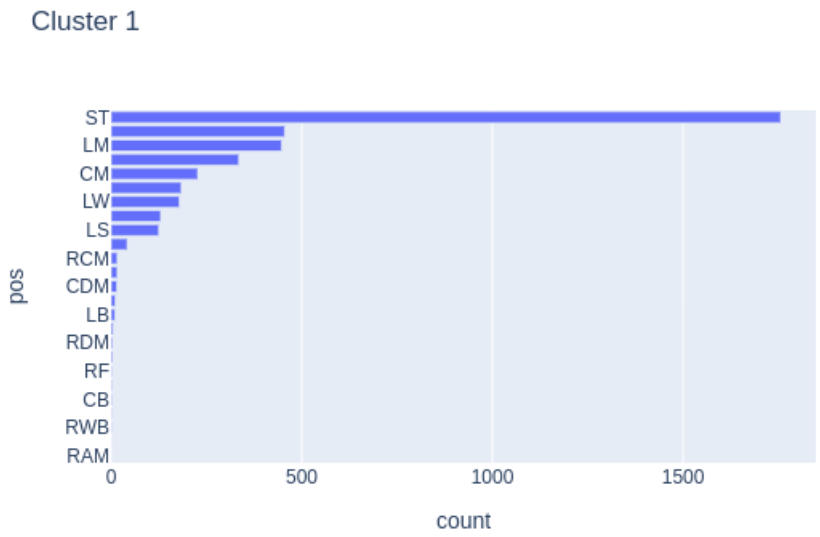


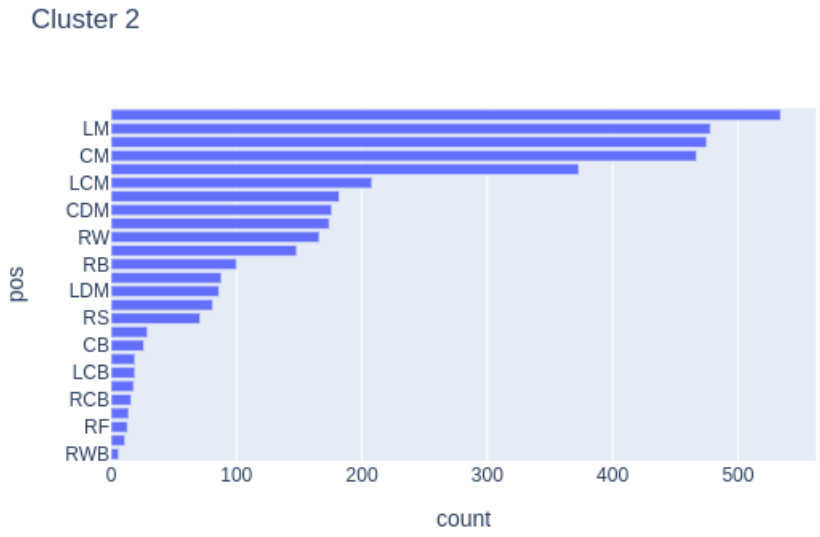
3.2 : DIVISIVE CLUSTERING(Top Down Approach)

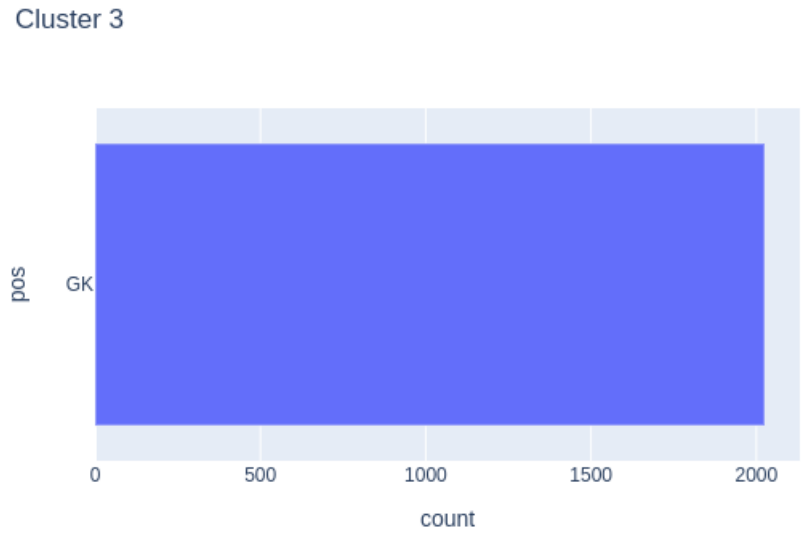
We start at the top with all documents in one cluster. The cluster is split using a flat clustering algorithm. This procedure is applied recursively until each document is in its own singleton cluster.

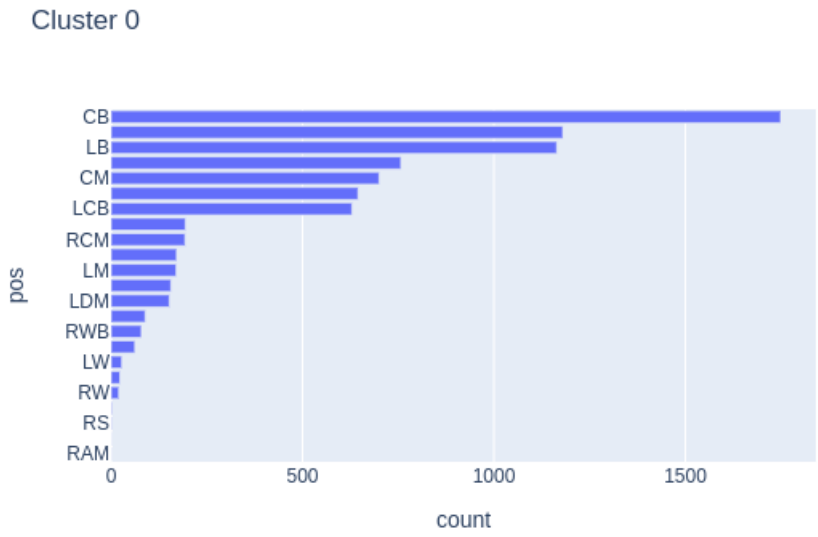
We referred the link shared and tried to analyse how the DIANA Algo works.

3.3 : Analysis of the Hierarchical Clustering



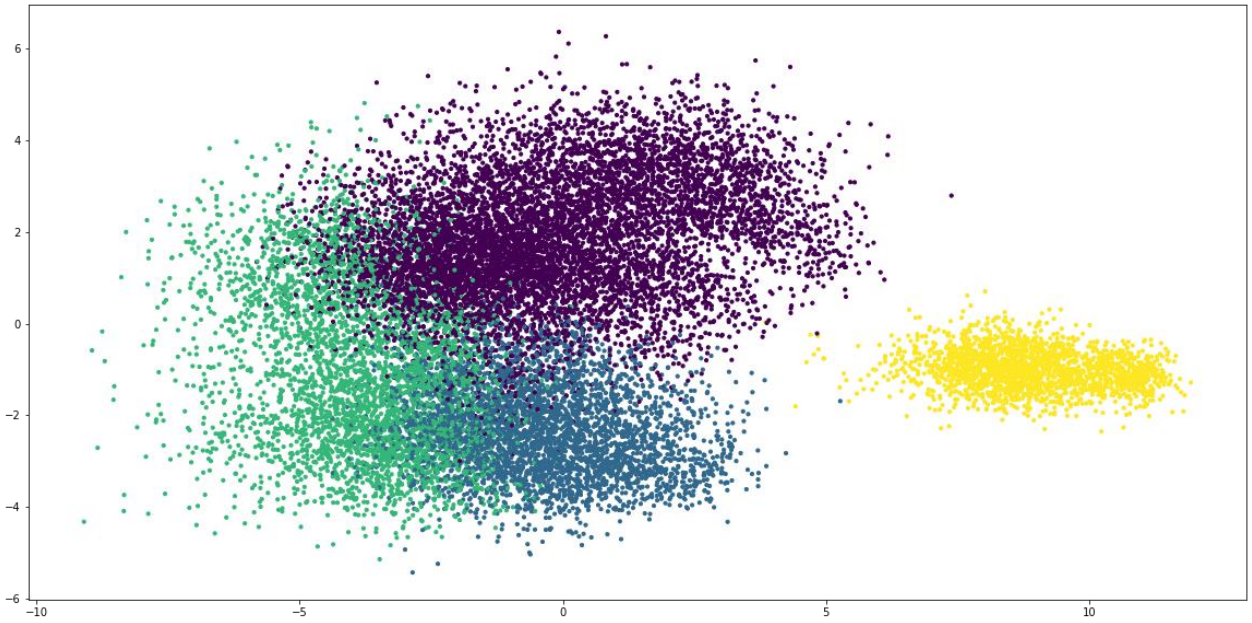






As we saw in KMeans Analysis the same hidden patterns appear here as well , However the difference is that for other clusters (other than Goal Keepers) we can see several features being mixed up here.

So lets have a view at the pictorial representation i.e Using PCA we will visualize the clusters .



So from here we can see an overlap between the different clusters which was not the case with KMeans Clustering . We will look at this aspect again at the end when we will be comparing all the clusters.

Conclusion of Task 3 :

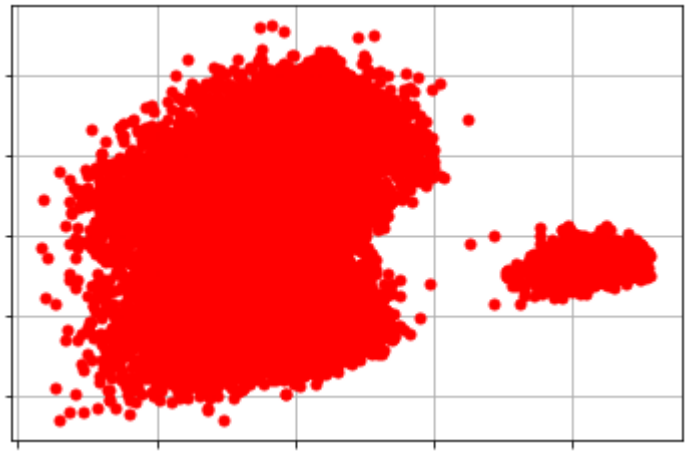
Though hierarchical clustering is a good approach but for our data as we can see there is overlapping of clusters it isn't the best approach .

TASK 4 : DBSCAN

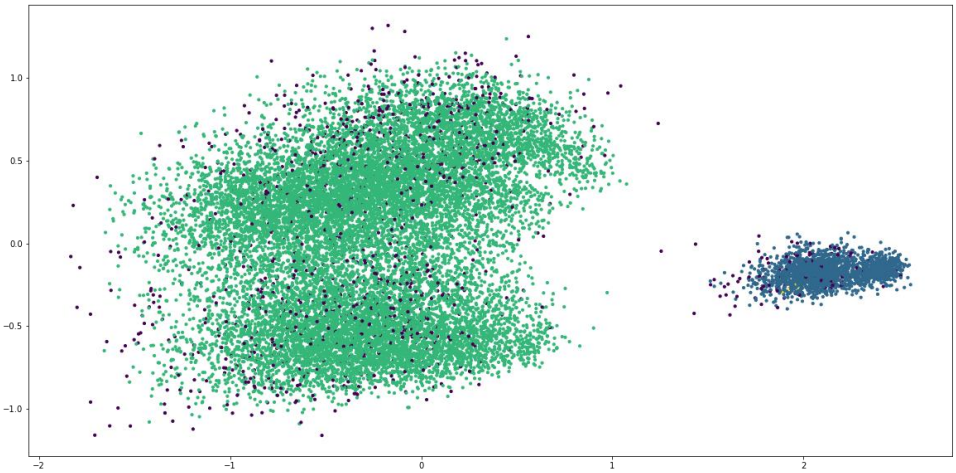
4.1 : Use DBSCAN to cluster the data

DBSCAN is density based scanning algorithm . We used inbuilt DBSCAN to cluster the data. As the number of features were high we used Principal Component Analysis to reduce the number of features to 2 (uses correlation among features to draw results and use 2 principal features)

First we used the pyclustering DBSCAN and we got the following results:



Then we used the inbuilt DBSCAN of sklearn :

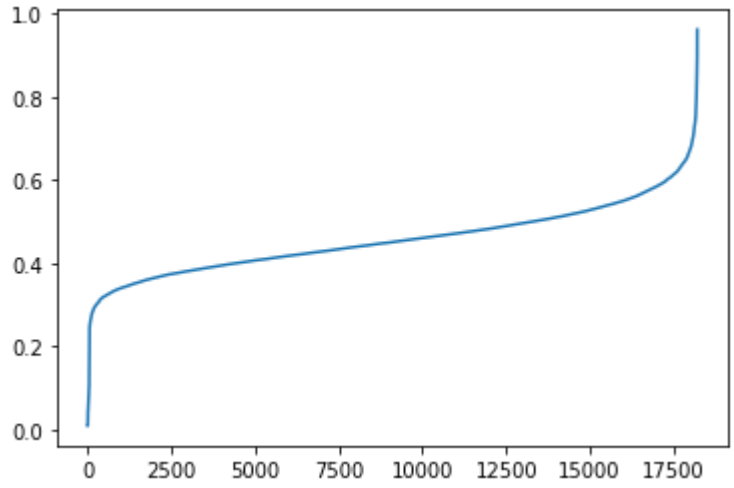


The clusters were not very clear and as it is density based we could see outliers and noise in this clustering approach.

4.2 : Selection of Parameters of DBSCAN (eps and Minpts)

To choose the value of eps for DBSCAN we used the following graph , from which we conclude the value of eps to be 0.6.

DBSCAN

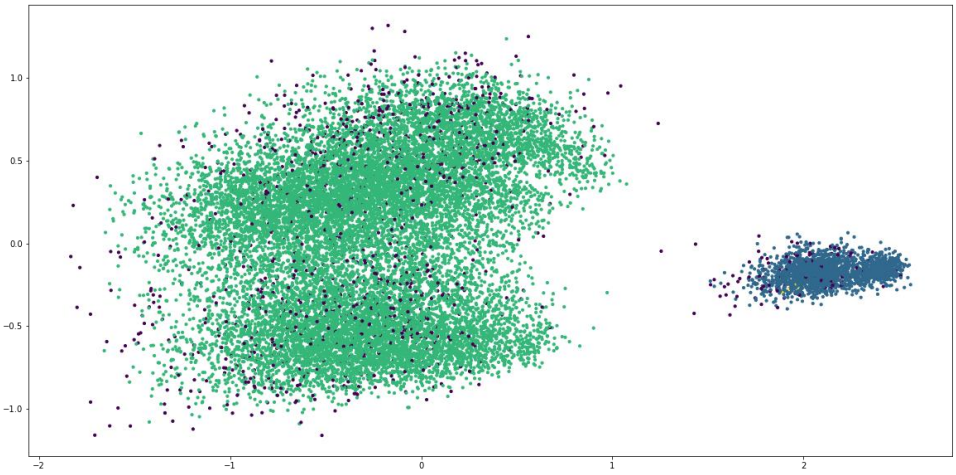


Coming to the next Parameter Selection , we selected the Min no of Samples using the following approach.

```
In [17]: print("-------------------------------------------------------------------------")
         for i in range(1 ,15):
             dbscan_model=DBSCAN(eps=0.6, min_samples=i)
             cluster = dbscan_model.fit(data_scaled)
             #finding unique cluster labels
             len_labels=len(set(cluster.labels_))
             flag=0
             if -1 in cluster.labels_:
                 flag=1
             n_clusters_ = len_labels-flag
             print( 'Min No of Samples = ',i,' : Num of Clusters : ',n_clusters_)
         print("-------------------------------------------------------------------------")

         -------------------------------------------------------------------------
         Min No of Samples =  1  : Num of Clusters :  947
         Min No of Samples =  2  : Num of Clusters :  34
         Min No of Samples =  3  : Num of Clusters :  5
         Min No of Samples =  4  : Num of Clusters :  4
         Min No of Samples =  5  : Num of Clusters :  4
         Min No of Samples =  6  : Num of Clusters :  4
         Min No of Samples =  7  : Num of Clusters :  2
         Min No of Samples =  8  : Num of Clusters :  2
         Min No of Samples =  9  : Num of Clusters :  2
         Min No of Samples =  10  : Num of Clusters :  2
         Min No of Samples =  11  : Num of Clusters :  3
         Min No of Samples =  12  : Num of Clusters :  2
         Min No of Samples =  13  : Num of Clusters :  3
         Min No of Samples =  14  : Num of Clusters :  2
```

Depending on the Number of Clusters we are aiming at we can choose the Minimum Number of Samples.

4.3 : Analyse the Clusters Formed



So from the analysis of DBSCAN we can see the presence of outliers in the clusters formed and the noise is also visible.

Coming to the 2 clusters we can see that the inter cluster distance and the intra cluster distance are as they should be and do justice to their names.

We are not able to inference any new hidden patterns apart from the ones we found out from task 2 and task 3.

Conclusion of the Project

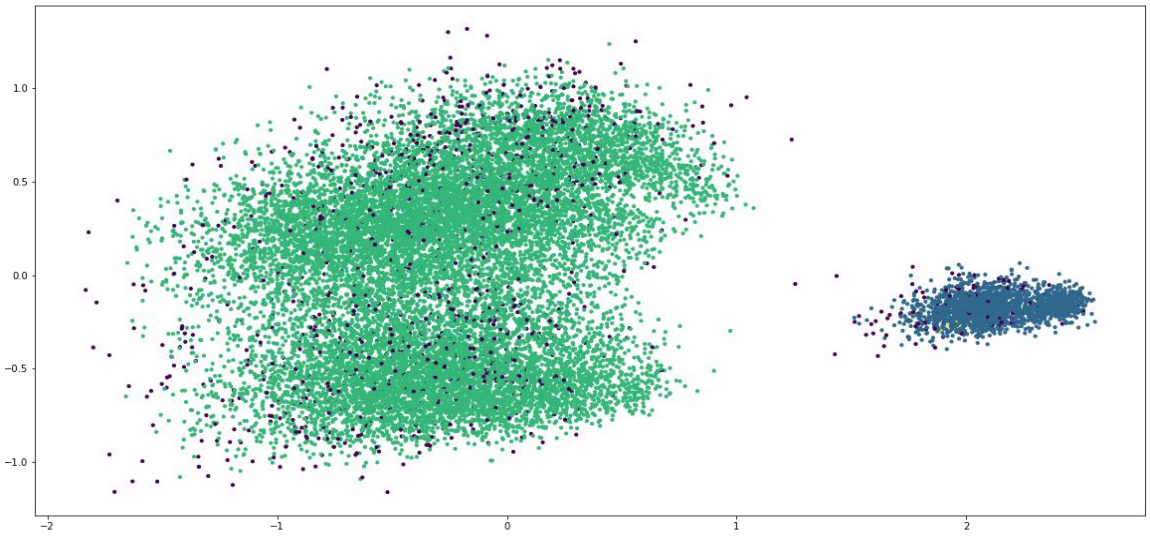
In this project we began with the football data given to us from which we drew visualizations and further extracted the insights from it. We learned how to analyse data , how to effectively view the data , how to process it and then perform analysis on it.

This project basically gave us a huge insight on how we can use visualizations to gain major insights about the data which also revealed several hidden patterns , we came to know about the positions , the factors which affect the position of players , what attributes are shown by the players of a particular club , the count of players , the outliers (Ronaldo , Messi , Neymar) and the list goes on.

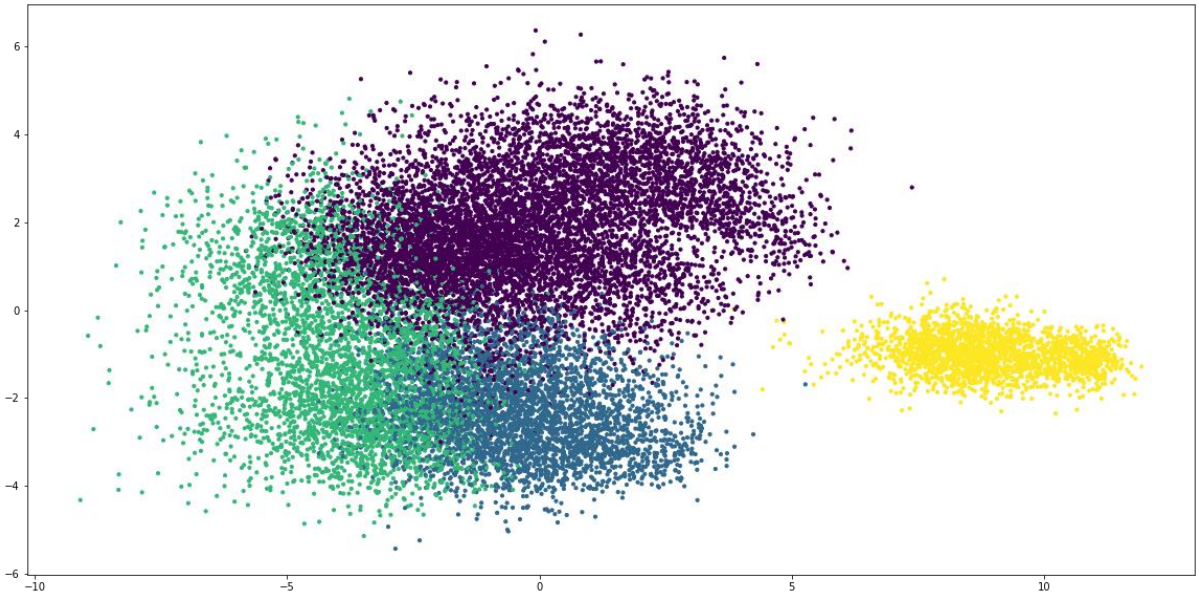
We used different metrics to draw conclusions from the data and analyse it efficiently . The results of the analysis were then applied in determining the best clustering algorithm .

Though at the end of each task we have explained about the conclusions of our task and how each clustering algorithm performed to end with this report we will just show a small overview of our conclusions and inferences drawn:

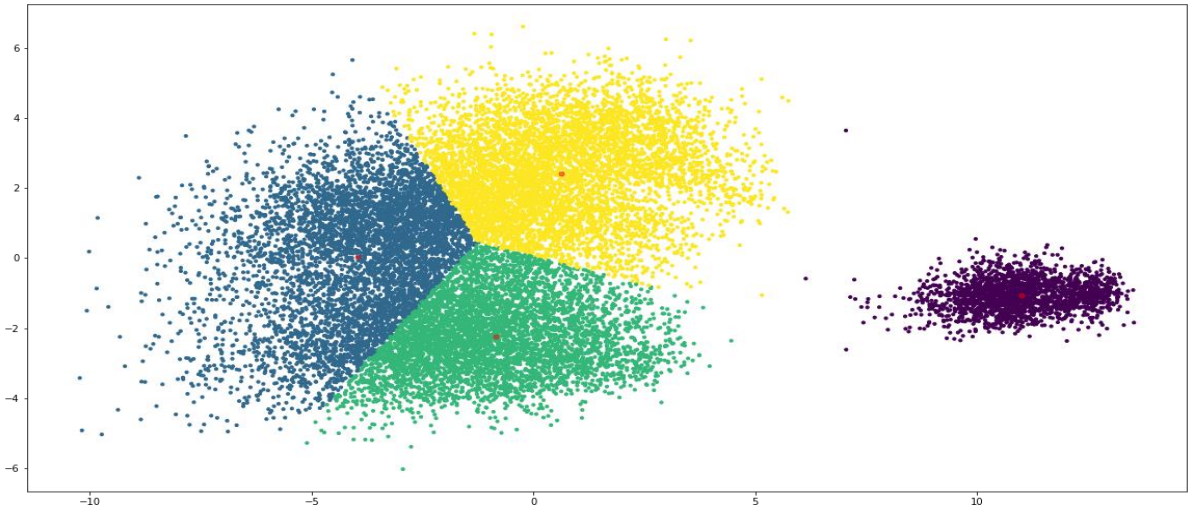
DBSCAN :



Hierarchical Clustering:



K-Means Clustering :



So out of all the clustering algos we choose KMeans Clustering Algorithm as the best for this particular data as firstly it fits the data well , it gives us non overlapping clusters , No of outliers and the frequency of noise is less and also as we concluded from the task 2 it gives us good insight about the hidden patterns and we can also group the clusters (players) based on their playing positions . The intra cluster distance and inter cluster distance for K means is also quite coherent with what it actually should be .

----------Thank You----------