ASSIGNMENT 5 REPORT

—

Team Mates:

B Krupa Kiranmai, *2021201022*

N C S Jagannath, *2021201024*

Football Data Clustering:

1.K-Means Clustering:

The task at hand is to build a machine clustering model to cluster and analyze the cluster with the help of k-means built from scratch.

Dataset:

The dataset consists of 60 attributes.

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	Unnamed: 0	18207 non-null	int64
1	ID	18207 non-null	int64
2	Name	18207 non-null	object
3	Age	18207 non-null	int64
4	Nationality	18207 non-null	object
5	Overall	18207 non-null	int64
6	Potential	18207 non-null	int64
7	Club	17966 non-null	object
8	Value	18207 non-null	object
9	Wage	18207 non-null	object
10	Special	18207 non-null	object
11	Preferred Foot	18159 non-null	object
12	International Reputation	18159 non-null	object
13	Weak Foot	18159 non-null	float64
14	Skill Moves	18159 non-null	float64

15	Work Rate	18159 non-null	object
16	Body Type	18159 non-null	object
17	Real Face	18159 non-null	object
18	Position	18147 non-null	object
19	Jersey Number	18147 non-null	object
20	Joined	16655 non-null	object
21	Loaned From	1290 non-null	object
22	Contract Valid Until	17891 non-null	object
23	Height	18159 non-null	object
24	Weight	18159 non-null	object
25	Crossing	18156 non-null	object
26	Finishing	18159 non-null	float64
27	HeadingAccuracy	18159 non-null	float64
28	ShortPassing	18159 non-null	float64
29	Volleys	18159 non-null	float64
30	Dribbling	18159 non-null	float64
31	Curve	18159 non-null	float64
32	FKAccuracy	18159 non-null	float64
33	LongPassing	18159 non-null	float64
34	BallControl	18159 non-null	float64
35	Acceleration	18159 non-null	float64
36	SprintSpeed	18159 non-null	float64
37	Agility	18159 non-null	float64
38	Reactions	18159 non-null	float64
39	Balance	18159 non-null	float64
40	ShotPower	18159 non-null	float64

```
41  Jumping              18159 non-null  float64
42  Stamina              18159 non-null  float64
43  Strength             18159 non-null  float64
44  LongShots            18159 non-null  float64
45  Aggression           18159 non-null  float64
46  Interceptions        18159 non-null  float64
47  Positioning          18159 non-null  float64
48  Vision               18159 non-null  float64
49  Penalties            18159 non-null  float64
50  Composure            18159 non-null  float64
51  Marking              18159 non-null  float64
52  StandingTackle       18159 non-null  float64
53  SlidingTackle        18159 non-null  float64
54  GKDiving             18159 non-null  float64
55  GKHandling           18159 non-null  float64
56  GKKicking            18159 non-null  float64
57  GKPositioning        18159 non-null  float64
58  GKReflexes           18159 non-null  float64
59  Release Clause       16644 non-null  object
60  Unnamed: 60          27 non-null     object
dtypes: float64(35), int64(5), object(21)
```

Data Preprocessing:

- First dropped the columns that we felt are not required for the task at hand.

Null Values imputation

- Then analyzed the presence of the null values in the dataset.

- We found out 48 rows have most of the columns null values so we dropped them.
- Then imputed the categorical attributes 'club','position','jersey number' as 'missing'

Data cleaning/transformation of categorical data

- The columns preferred foot,work rate , body type,and real face have noise data in themselves.
- So we observed that the noise data corresponds to 30 odd rows so we removed them.
- The columns value and wage correspond to money so we converted them into required float data types by taking M as million and K as thousand .
- The column height was given in feet and inch notation; we converted it into centimeters .
- The values in the column weight were as 123lbs we removed that lbs metric.
- The values in columns international reputation,special,crossing were integers so we converted them to int.
- Then converted all the categorical attributes into numerical by using the label encoder of sklearn.

Data cleaning/transformation of numerical data:

- For identifying the noise that present in the numerical columns we plotted an histogram for every column.
- We identified there were a few columns that belonged to the goal keeper which may possibly act as outliers.
- The shape of the data after the preprocessing was (18124,53)

Model construction and analysis:

1.1 K-Means from scratch:

- Built the k-means algorithm from scratch.
- The K-means function takes the number of clusters and data as the parameters and returns the clusters and their centroids.

- We set the default number of clusters to be 3 and the maximum number of the iterations were 10000.
- We added an extra column as cluster_num to store the cluster of the particular tuple.

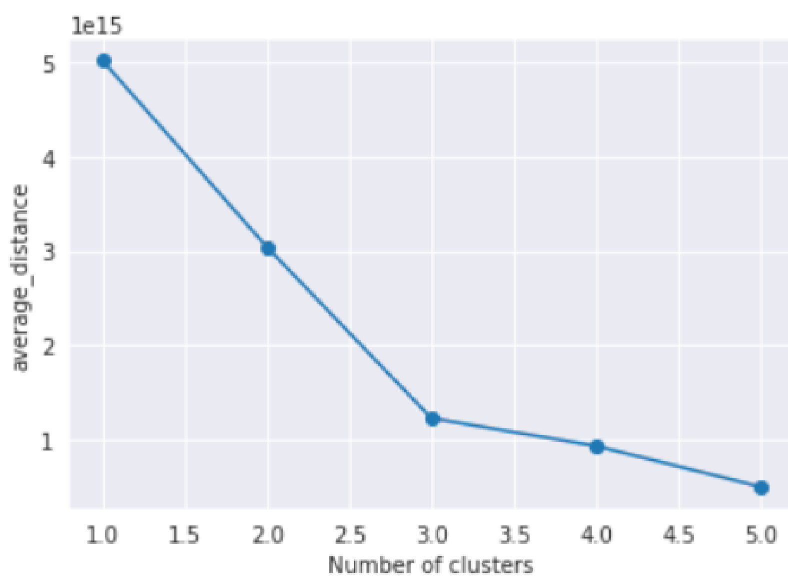
1.2 Clustering the data with different K:

- We used the above constructed k-means algorithm to vary the k=3,5,7 and get the centroids and clustered labels of each tuple.

1.3 Finding optimal Number of clusters:

- Elbow method:
  - Calculated the average distance of the elements from its clusters and plotted it.

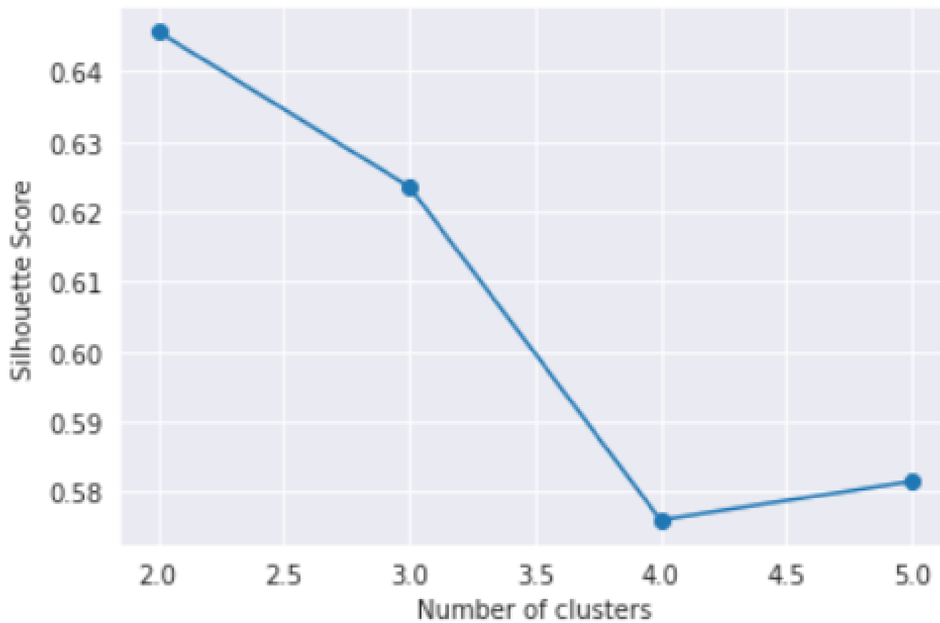
```
#plotting the avg distances.
plt.grid()
plot(clusters,average_distance,'average_distance')
```



from the above we can say 3 was the optimal number of clusters

- Silhouette score:
    - Calculated the silhouette score with the help of the inbuilt function

```
[60] plt.grid()
     plot(clusters[1:],s_score,'Silhouette Score')
```



From the above we can say that the optimal number of clusters are 2

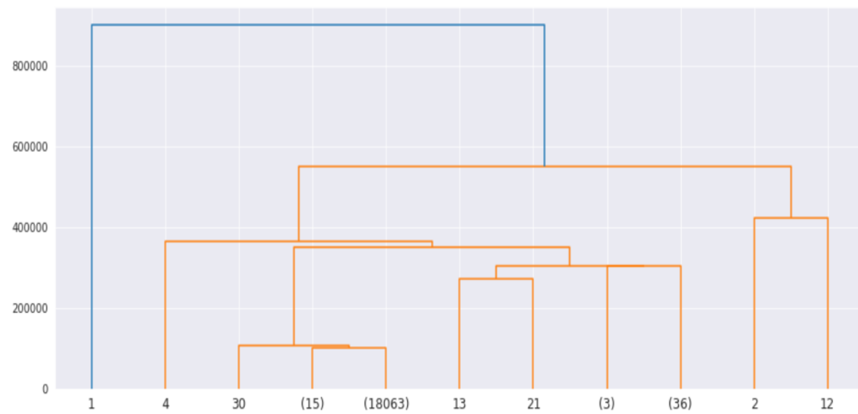
2.1 Agglomerative Clustering:

- Built all the four required agglomerative algorithms with the help of the scipy library.
- The data was preprocessed in the same way as specified above in the document.

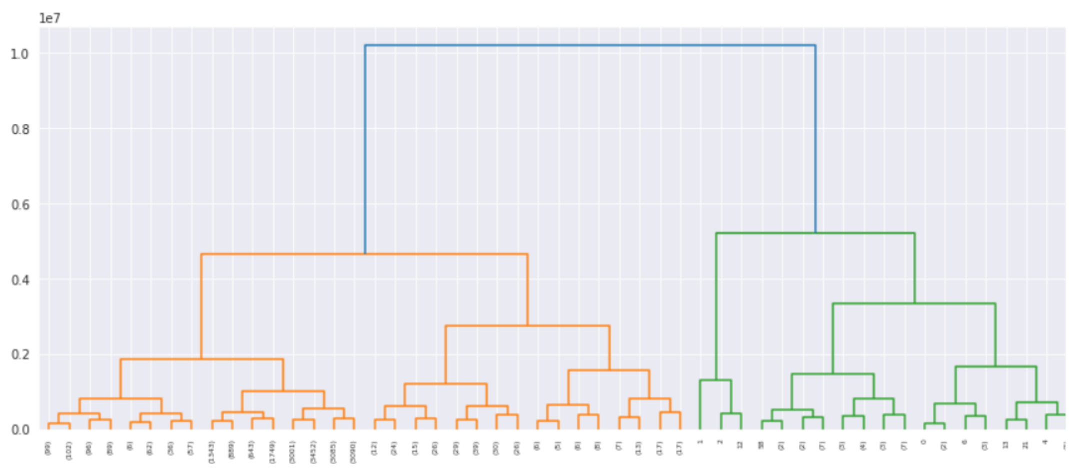
2.2 Dendrograms:

- Single Linkage:

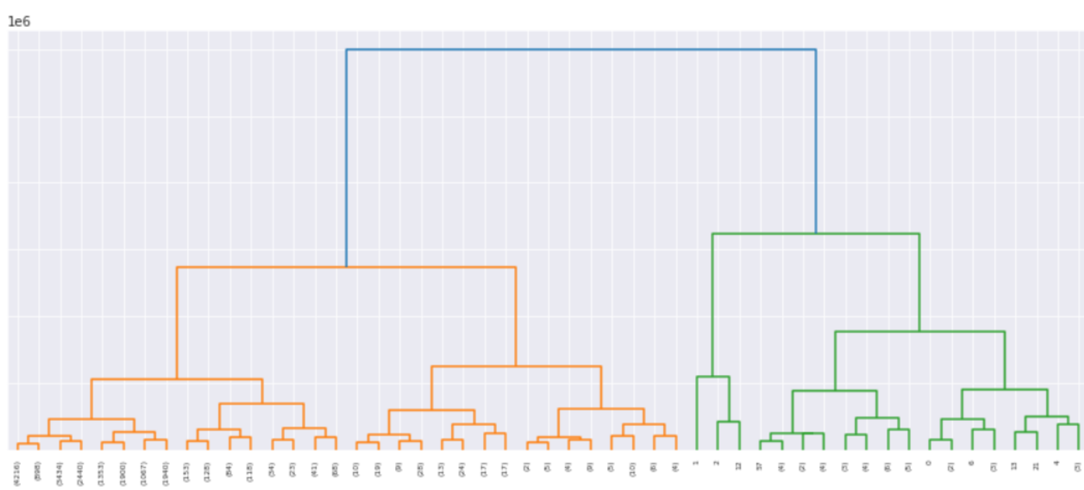
```
dn = dendrogram(Z,p=5,truncate_mode='level')
```



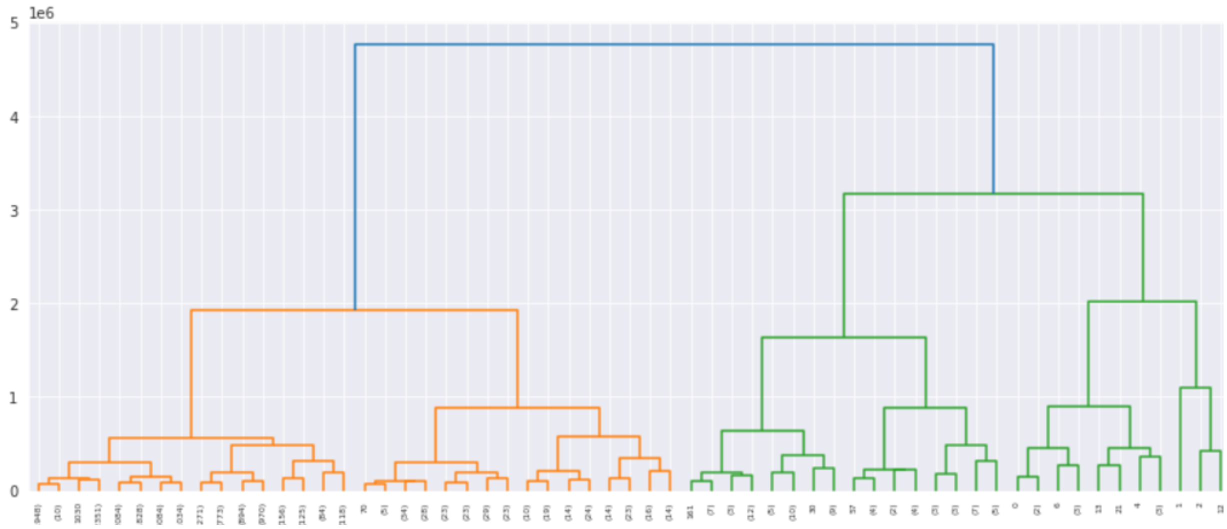
- Complete Linkage



- Average Linkage:



- Mean Distance:



1.4 .Analysis of K-Means From scratch:

- We can have different distance measures such as euclidean distance,manhattan distance.
- So we analyzed the intra class similarity with the distance measure, centroid distance measure and inter class similarity with the centroid linkage distance,for the two different distance measures specified above.
- For three clusters the inter class similarity and intra class similarity of euclidean and manhattan.
  - euclidean-{0: [0.0, 554955.8607390244, 3547464.1920400998], 1: [554955.8607390244, 0.0, 2992649.3143009148], 2: [3547464.1920400998, 2992649.3143009148, 0.0]} {0: 228231.1822998184, 1: 378551.28474512155, 2: 2345208.4456637334}
  - Manhattan-{0: [0.0, 3533118.5071504484, 563044.4908650895], 1: [3533118.5071504484, 0.0, 2970206.480034366], 2: [563044.4908650895, 2970206.480034366, 0.0]} {0: 238539.12676763086, 1: 2344064.246582597, 2: 374744.7285690202}
  - The intra class distances of euclidean are low than compared to manhattan.

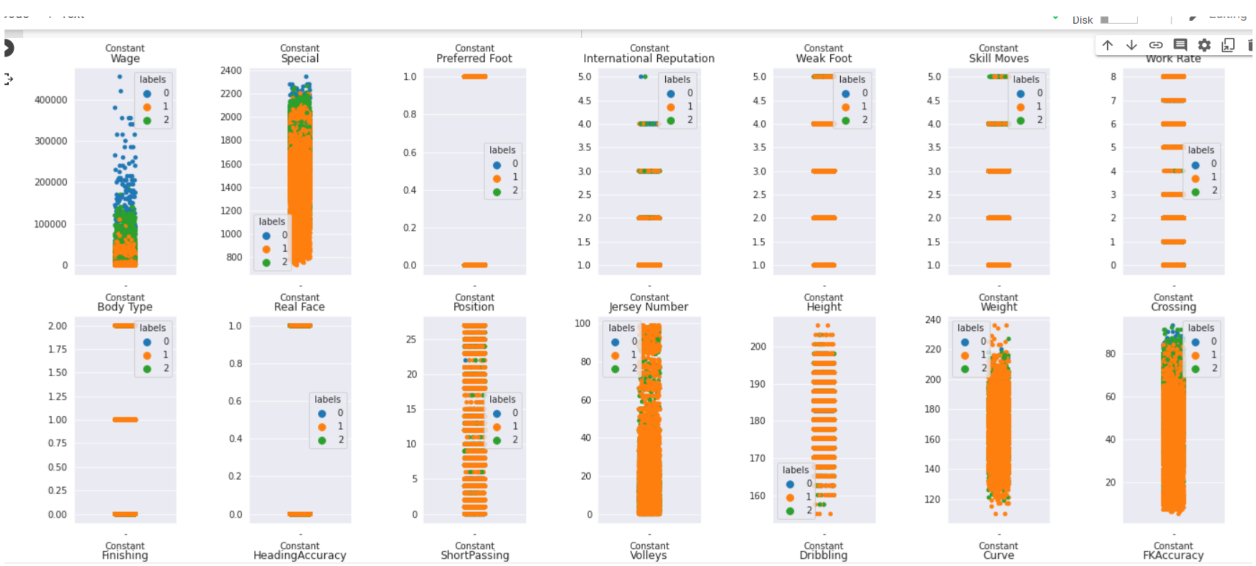
- The counts of the clusters were:

  1.0 11825

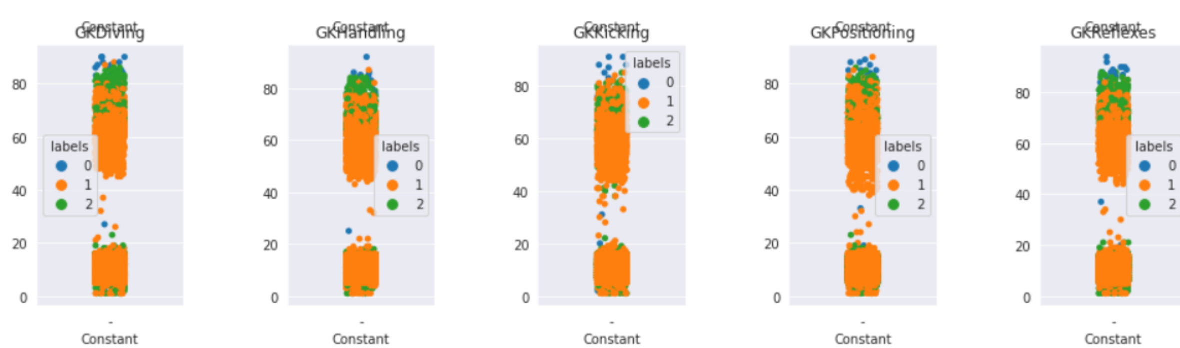
  2.0 6069

  0.0 230

- We visualized the clusters of the data points for each and every column.



- Some pointer we get from the above graph was that the low height has high skill moves.
- The high wage people have good ball control.
- The goalkeeper columns were there is a chance of outliers but the below graph show that they were clustered correctly so we didn't get any outliers on the preprocessed data.



3.Analysis:

- The data that was clustered is as follows:
    - K-means:

      1.0 11825

      2.0 6069

      0.0 230

    - Agglomerative clustering:

      1 18031

      2 77

      3 16

- We clustered the data to three classes. We get the counts of each cluster as above.We can see that the K-means gives better clustering of the data than the agglomerative.
- The hierarchical clustering may be not good for the data set given as each of the attributes have different things to offer than merging into the hierarchy.