DATA MINING

Assignment - 4

Group Members:

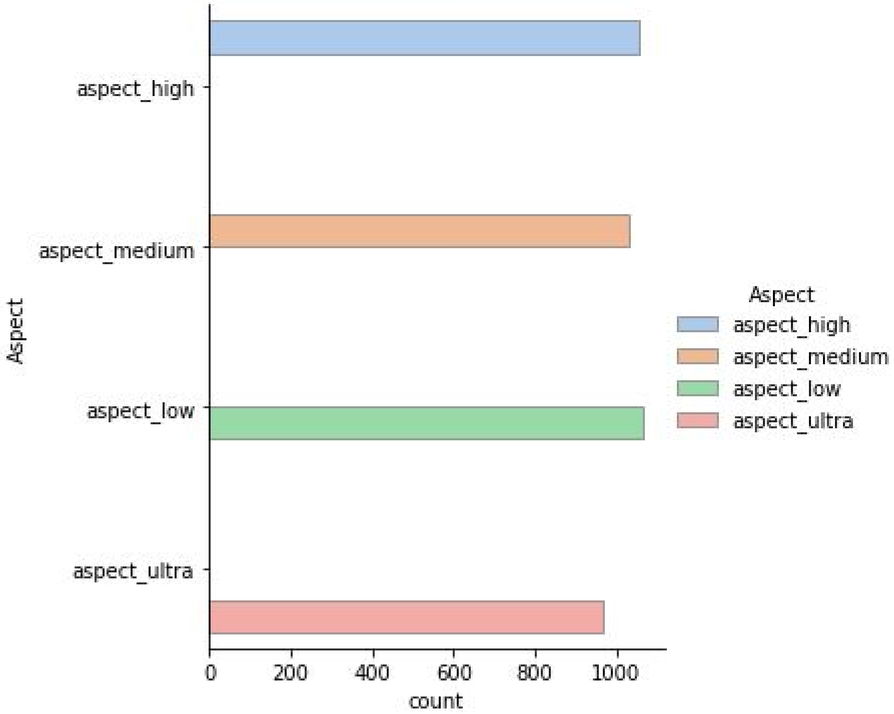
Anuj Verma	-	2017026
Sakshi Saini	-	2017092

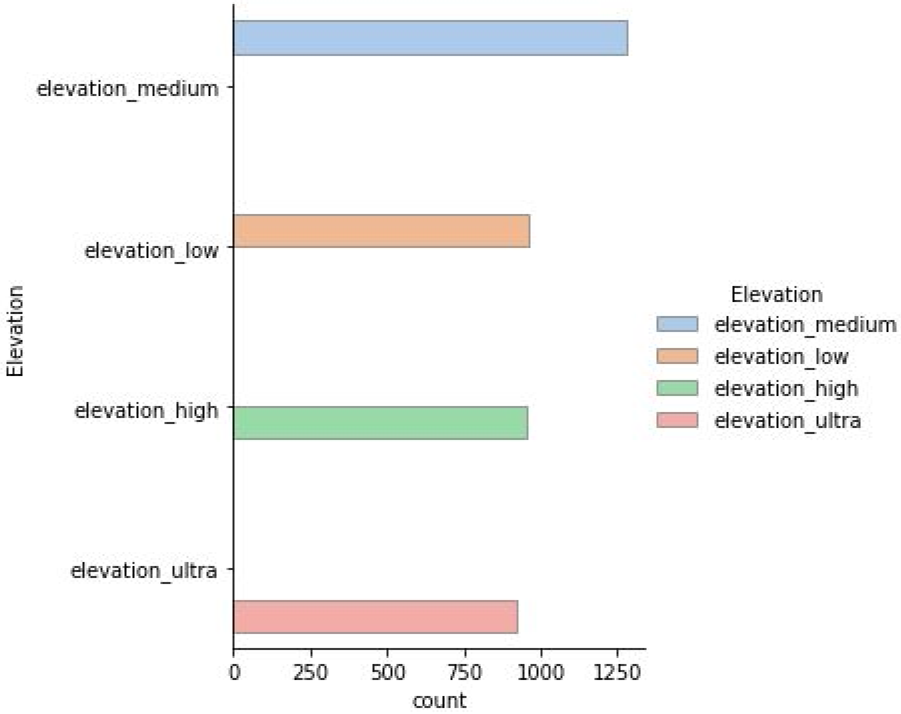
Assumptions:

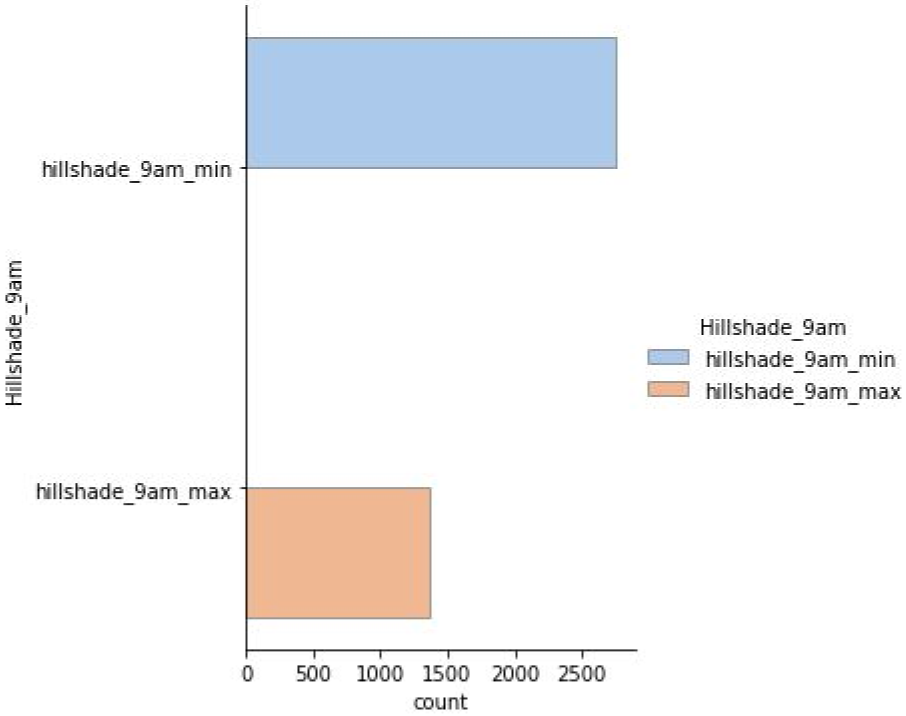
1. Runner function will save the results in 3 files, Kmeans, Kmeans++ and DBScan.
2. Results contains csv files and images for clusters in X
3. test_X.csv should be in the same file.

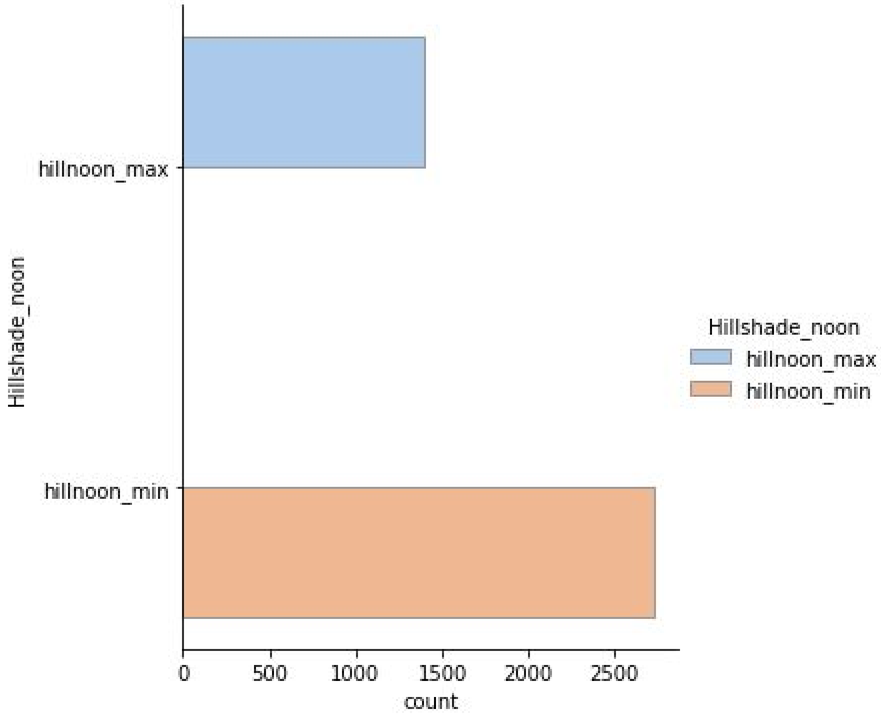
Data Visualization

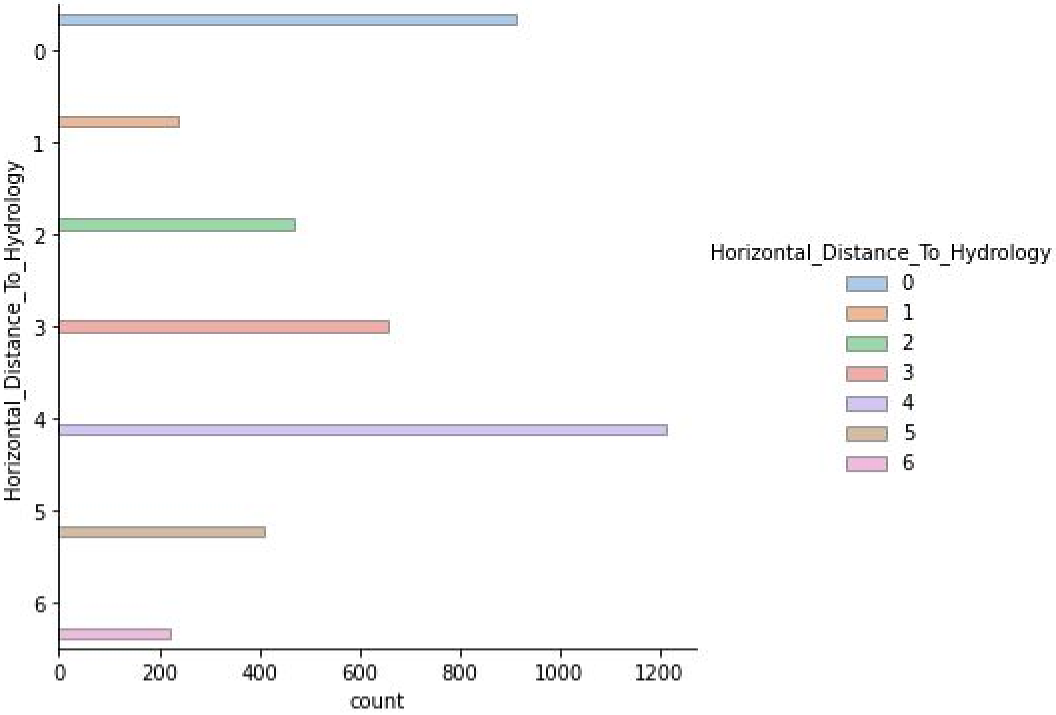
As the data was categorical we plotted the count of each value in each feature.

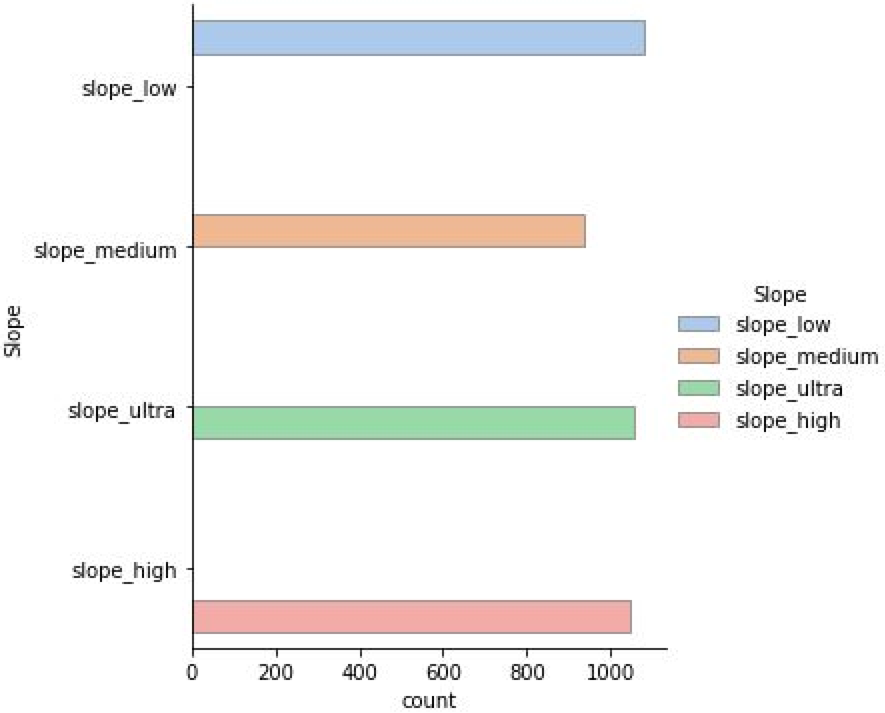


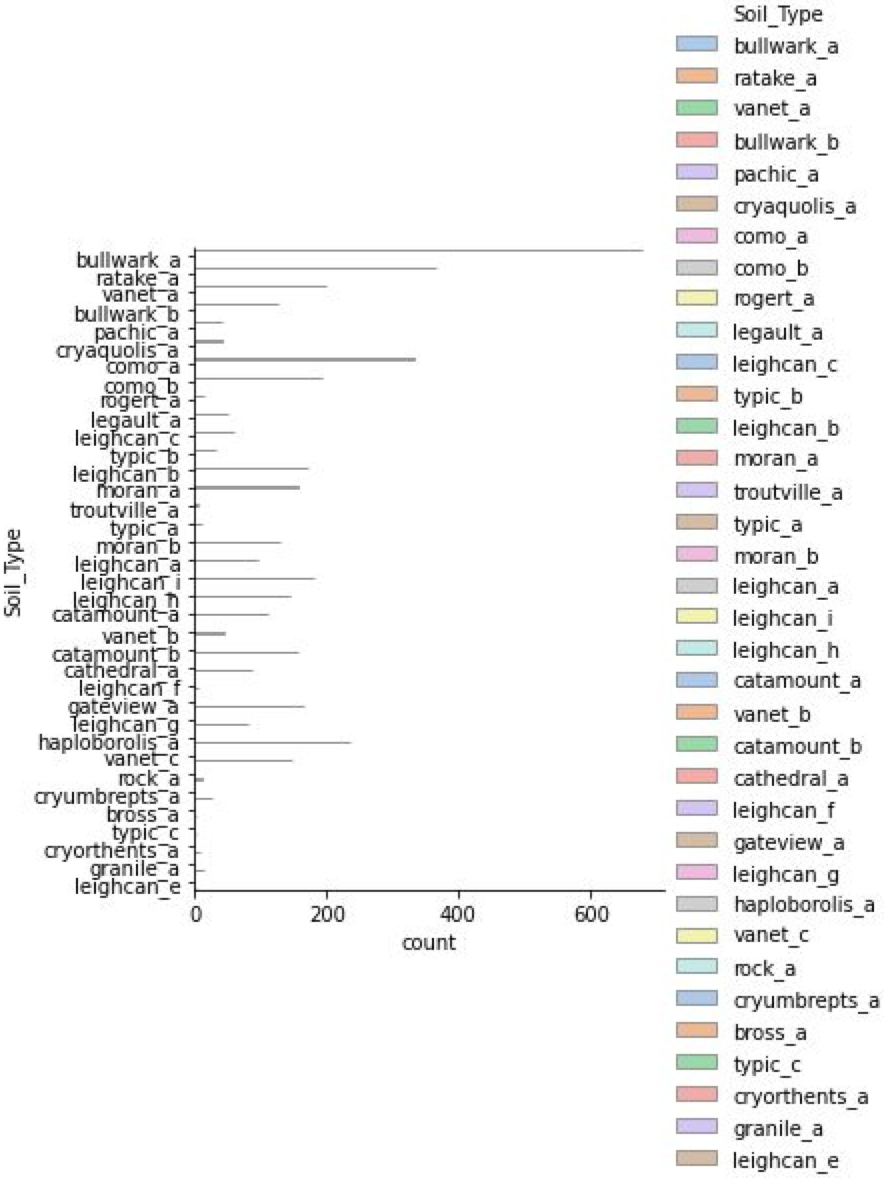


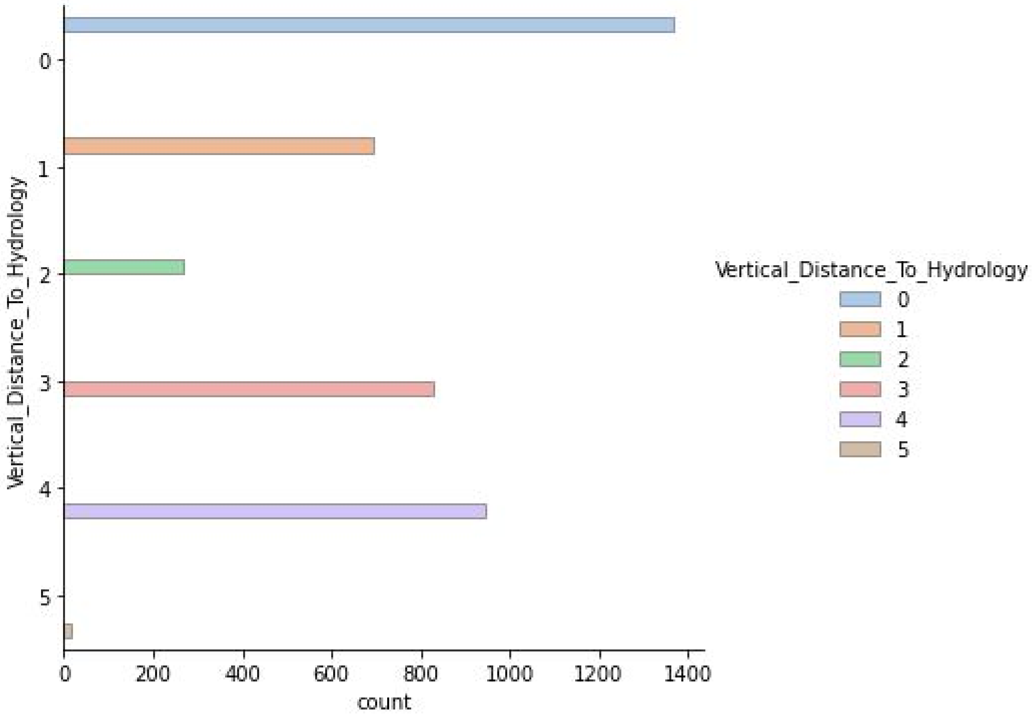

hillshade_9am_min
hillshade_9am_max
Hillshade_9am
Hillshade_9am
hillshade_9am_min
hillshade_9am_max
0
500
1000
1500
2000
2500
count


hillnoon_max
hillnoon_min
Hillshade_noon
Hillshade_noon
hillnoon_max
hillnoon_min
0
500
1000
1500
2000
2500
count

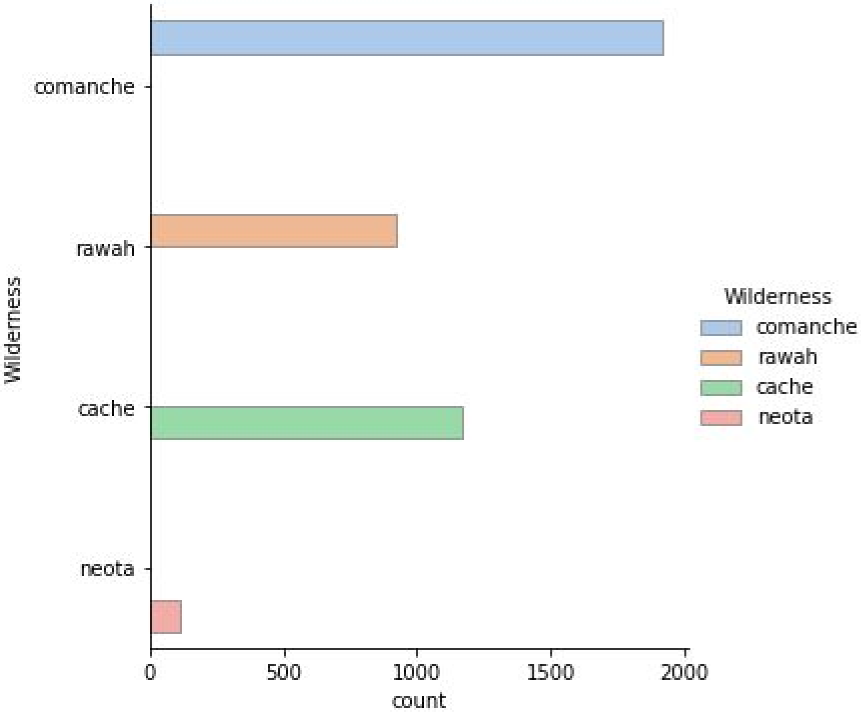

Horizontal_Distance_To_Hydrology
0
1
2
3
4
5
6
0
200
400
600
800
1000
1200
count
Horizontal_Distance_To_Hydrology
0
1
2
3
4
5
6


Slope
slope_low
slope_medium
slope_ultra
slope_high
0
200
400
600
800
1000
count
Slope
slope_low
slope_medium
slope_ultra
slope_high


Soil_Type
bullwark_a
ratake_a
vanet_a
bullwark_b
pachic_a
cryaquolis_a
como_a
como_b
rogert_a
legault_a
leighcan_c
typic_b
leighcan_b
moran_a
troutville_a
typic_a
moran_b
leighcan_a
leighcan_i
leighcan_h
catamount_a
vanet_b
catamount_b
cathedral_a
leighcan_f
gateview_a
leighcan_g
haploborolis_a
vanet_c
rock_a
cryumbrepts_a
bross_a
typic_c
cryorthents_a
granile_a
leighcan_e
0
200
400
600
count



Vertical_Distance_To_Hydrology
0
1
2
3
4
5
0
200
400
600
800
1000
1200
1400
count
Vertical_Distance_To_Hydrology
0
1
2
3
4
5



Wilderness
comanche
rawah
cache
neota
0
500
1000
1500
2000
count
Wilderness
comanche
rawah
cache
neota

Methodology:

Data Pre-Processing

1. Removed the id feature, as it does not give any information.
2. Converted the categorical feature into one-hot encoding because the clustering algorithm needs numerical data to work on.

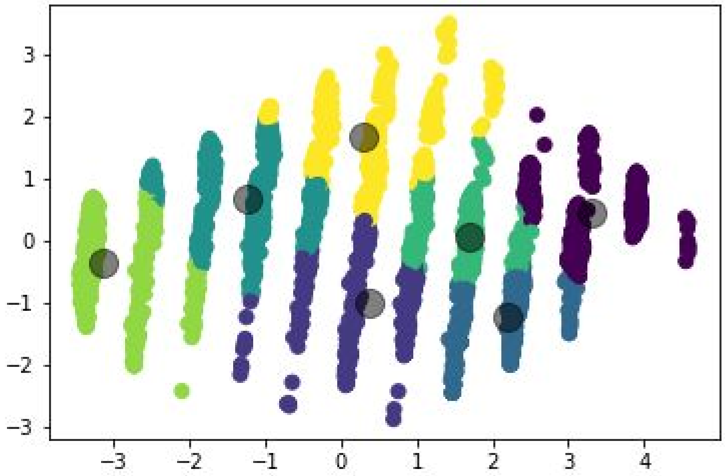
Clustering Algorithms:

1. K-Means (with Random Initialization, max_iteration = 300, n_init = 10)
   a. Centroids

```
Count of each Label:
Counter({5: 1048, 2: 571, 3: 566, 6: 513, 4: 495, 0: 479,
1: 448})

Centers:
[[ 3.29873915  0.45938042]
 [ 0.36495232 -1.00368967]
 [ 2.18972379 -1.22192304]
 [-1.23923953  0.67549584]
 [ 1.70162302  0.06775751]
 [-3.13067819 -0.3382617 ]
 [ 0.29917343  1.67811099]]
```

b. Visualization of K-Means



Gray Points are Centroids

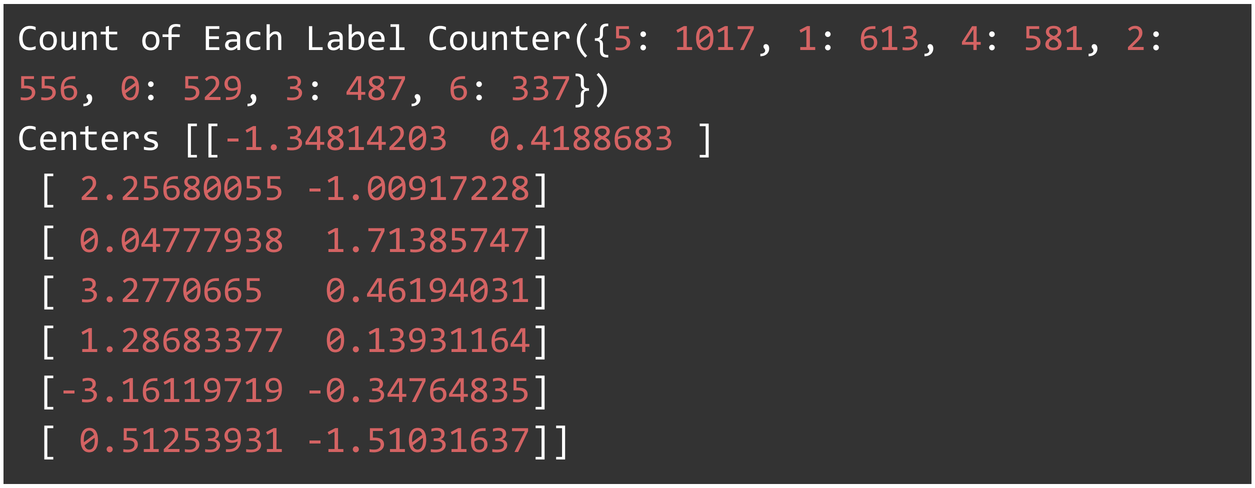
After that we thought to tune the hyperparameter that is the number of iterations and this is what we got.

Sr. No.	Number of Iterations	Differences	Sum Of Differences
1	100	[203, 53, 8, 2, 41, 66, 287]	660
2	200	[206, 54, 19, 1, 39, 62, 303]	684
3	500	[155, 53, 48, 16, 69, 60, 231]	632

4	1000	[203, 57, 53, 15, 37, 66, 327]	758
5	2000	[196, 42, 21, 22, 34, 35, 238]	588
6	5000	[204, 54, 15, 8, 39, 61, 287]	668
7	7000	[223, 37, 26, 11, 36, 37, 276]	646
8	10000	[203, 54, 11, 16, 40, 62, 274]	660

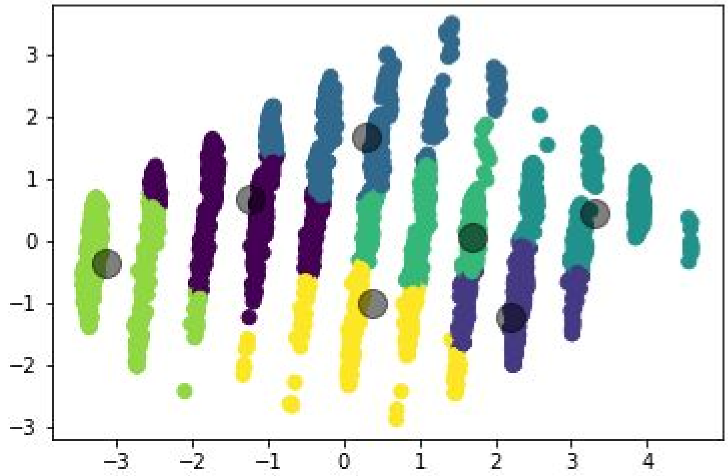
From this table we get that the best model for the Kmeans is on the number of iterations = 2000.

2. K-Means++ (with k-means++ Initialization)
    a. Centroids



```
Count of Each Label Counter({5: 1017, 1: 613, 4: 581, 2:
556, 0: 529, 3: 487, 6: 337})
Centers [[-1.34814203  0.4188683 ]
 [ 2.25680055 -1.00917228]
 [ 0.04777938  1.71385747]
 [ 3.2770665   0.46194031]
 [ 1.28683377  0.13931164]
 [-3.16119719 -0.34764835]
 [ 0.51253931 -1.51031637]]
```

b. Visualization



Gray Points are Centroids

After that we thought to tune the hyperparameter that is the number of iterations and this is what we got.

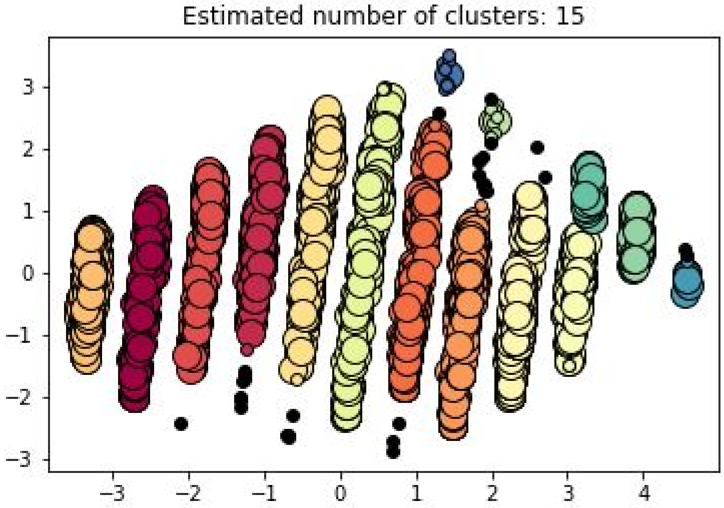
Sr No	Number of iterations	Differences	Sum of Differences
1	100	[208, 53, 16, 11, 39, 66, 315]	708
2	200	[203, 53, 6, 3, 38, 66, 287]	656
3	500	[156, 54, 44, 9, 62, 30, 231]	586
4	1000	[203, 53, 7, 4, 38, 66, 287]	658

5	2000	[192, 42, 23, 15, 35, 31, 238]	576
6	5000	[205, 40, 20, 30, 34, 37, 238]	604
7	7000	[198, 42, 19, 20, 34, 33, 238]	584
8	10000	[203, 56, 11, 7, 39, 58, 282]	656

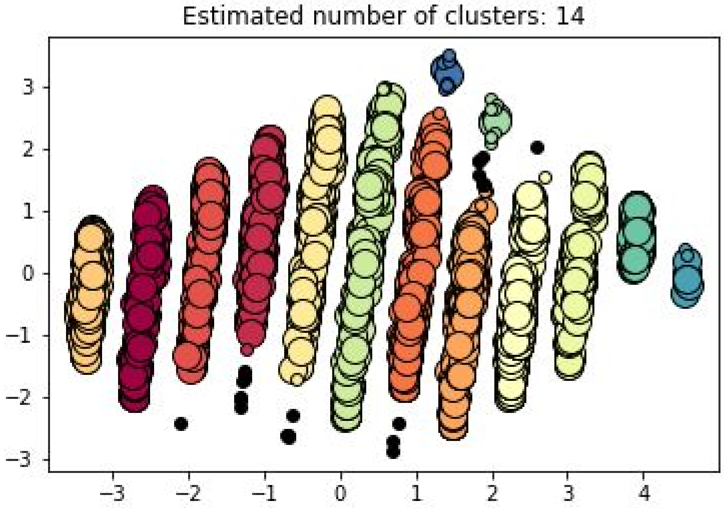
From this table we get that the best model for the Kmeans is on the number of iterations = 2000.

DB Scan:

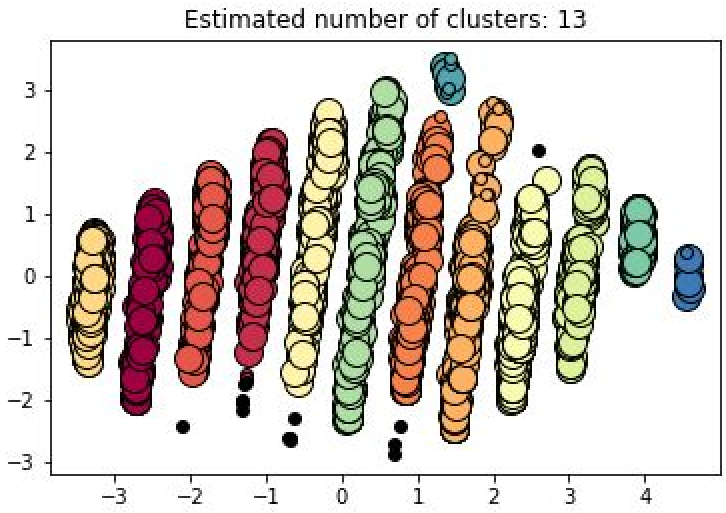
For DB Scan we changed the values of esp to get the **seven** clusters. Here from the images you will get the idea how we tune esp to get the seven clusters.



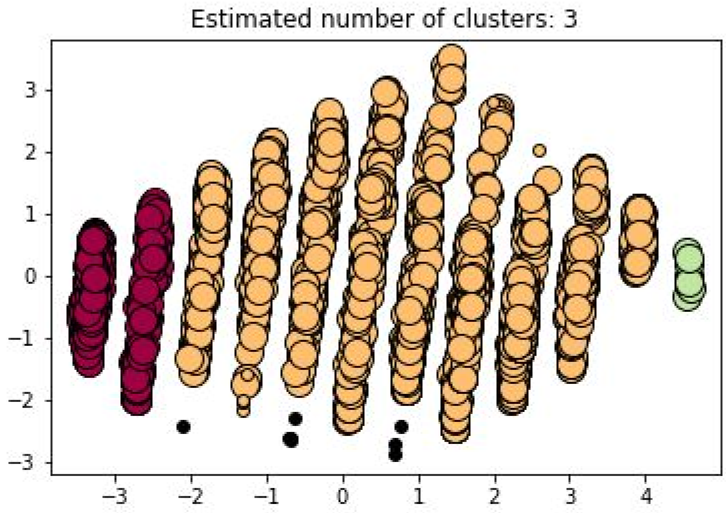
With esp = 0.3 we will get the number of clusters = 15



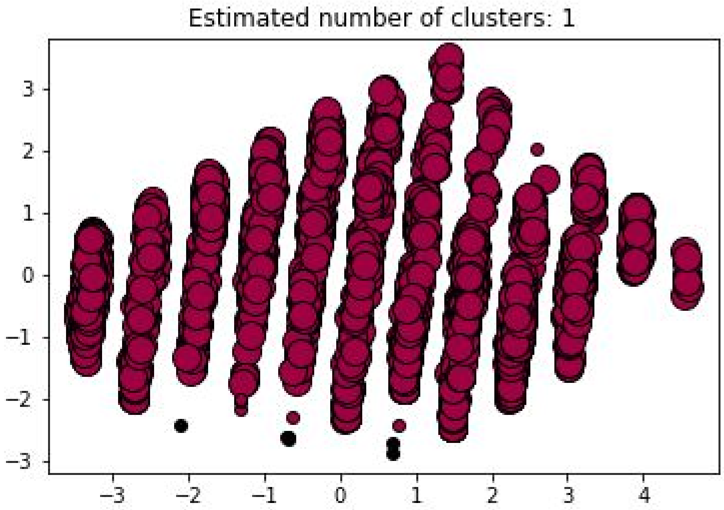
With esp = 0.35 we will get the cluster = 14



With esp = 0.45 we will get the cluster = 13

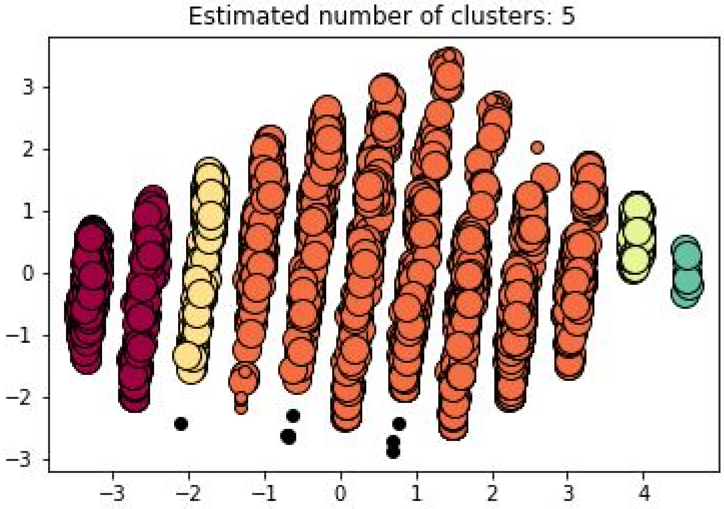


With esp = 0.55 we will get the cluster= 3

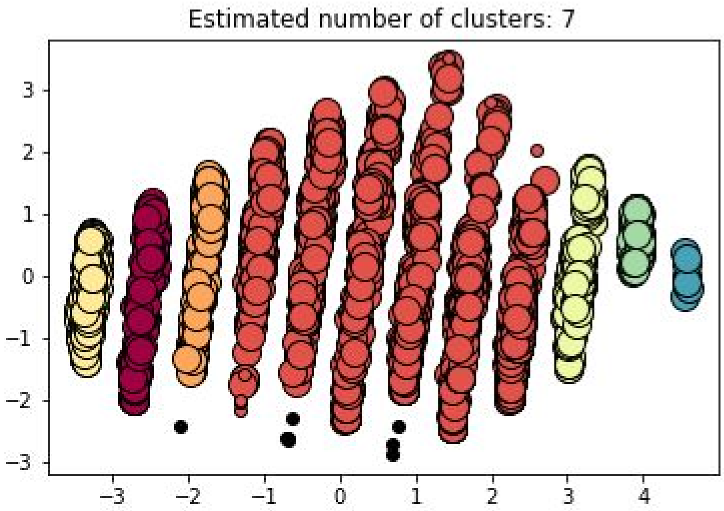


With esp = 0.6 we will get the cluster = 1

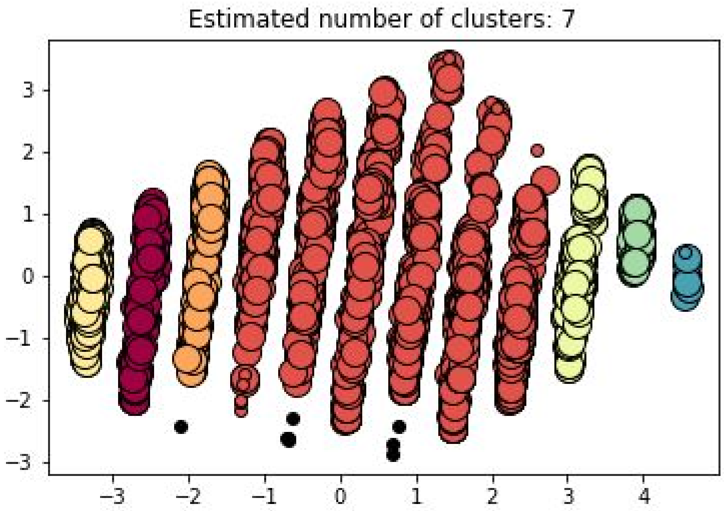
Since the 0.55 and 0.6 are giving the lesser number of the clusters so we thought to take the values less that 0.55
Then here what we got.



With esp = .53 we will get the number of clusters = 5



With esp =0.52 we will get the number of clusters = 7



With esp = 0.51 we will get the number of clusters = 7

So from this we can interpret that esp = 0.51 or esp = 0.52 can give the desired number of clusters.

After this we find the difference between actual and expected and it is that we got with esp = .51 and minimum number of samples = 10

Difference = [532, 526, 394, 318, 286, 138, 1698]
Sum of differences = 3892
Since the difference sum is quite high so here we can interpret that the **DB Scan** is not performing better than **K-means** and **K-means++** for this set of data.

So we thought for doing with **Agglomerative Algorithms**
And this is what we got with the agglomerative algorithm
Differences = [395, 112, 50, 14, 143, 88, 340]
Sum of differences = 1142
So here we can interpret that the agglomerative algorithm is working better than DB Scan.

Learning:

1) Here we learned how to make the data compatible with the clustering algorithm , we did one hot encoding and converted our categorical data to numerical data.
2) We also get to know clustering algorithms are relative, that is it may be possible that a clustering algorithm that is good for one data may not be good for other data.
3) We learn how the esp value can affect the number of clusters.
4) We learn how to tune the model and get the best value out of your model.
5) We learn about the clustering algorithms and their working with our data.