Machine learning Assignment 2

M22CS007

A. Jyothi

1.

Task1 link:
https://colab.research.google.com/drive/1rAS7oWSyUGZkt45DRJk9GNHpT6Mrt7gQ?usp=sharing

A. Kmeans from Scratch

My rollno is 7, so 7%4=3 (11 in binary) and yr>20 so 1. So I got 3 oes in my number. So my dataset is"diabetes.csv".

It has got the following columns

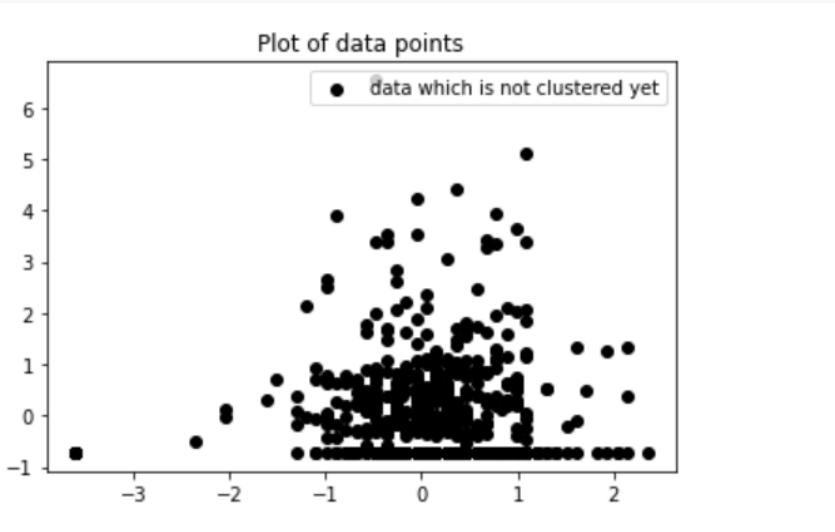
Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome

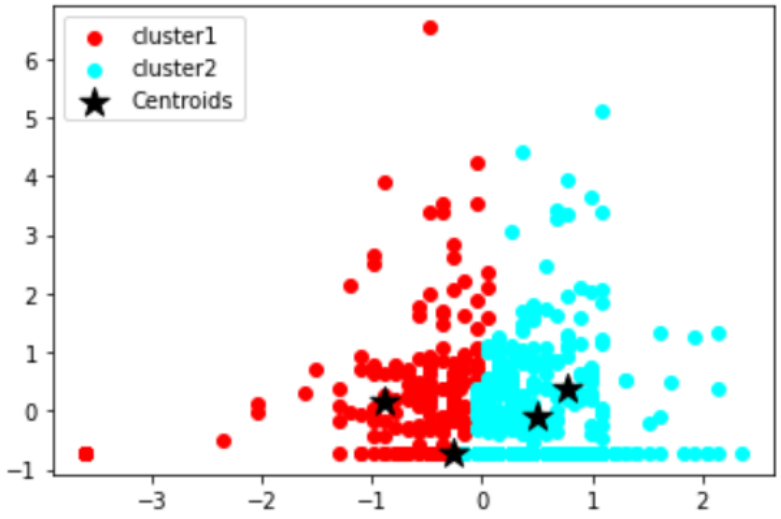
I have randomly taken blood pressure and insulin into account.

And performed kmeans algorithm from scratch. I used 300 iterations

No. of clusters is 2, because the output labels are only 2{0,1}

These are the output plots:



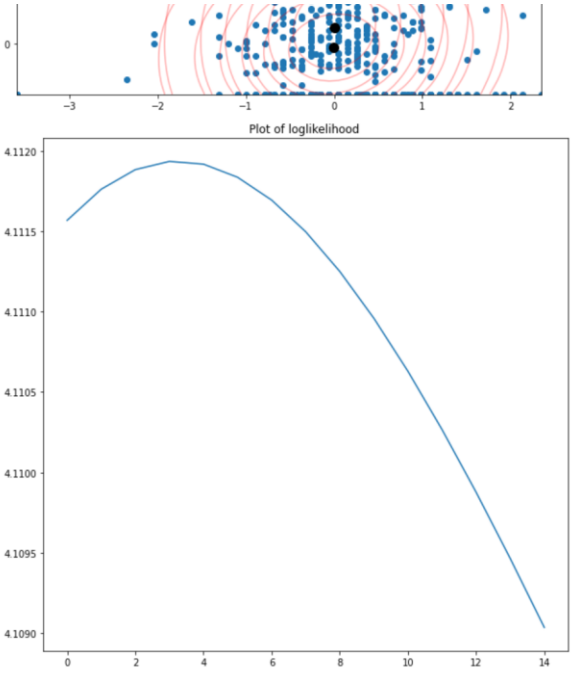



```
{1: array([[-0.47179147,  0.4813591 ],
           [-0.99373735,  0.03549796],
           [-0.26301312, -0.05024456],
           [-0.57618065, -0.29889789],
           [-0.26301312, -0.15313559],
           [-3.60346676, -0.71903627],
           [-0.15862394, -0.16170985],
           [-0.68056983, -0.71903627],
           [-0.68056983,  0.18126026],
           [-0.15862394, -0.29032364],
           [-0.57618065, -0.58184823],
           [-1.20251571,  2.15333837],
           [-0.3674023 , -0.71903627],
           [-0.3674023 ,  1.66460597],
           [-0.68056983,  0.58425013],
           [-0.99373735, -0.41036317],
           [-0.26301312,  0.18126026],
           [-0.05423477,  0.54995312],
           [-3.60346676, -0.71903627],
           [-0.88934818, -0.17885835],
           [-0.15862394, -0.23887812],
           [-3.60346676, -0.71903627],
           [-0.26301312,  2.83927858]
```

This is the output array of clustered data points.

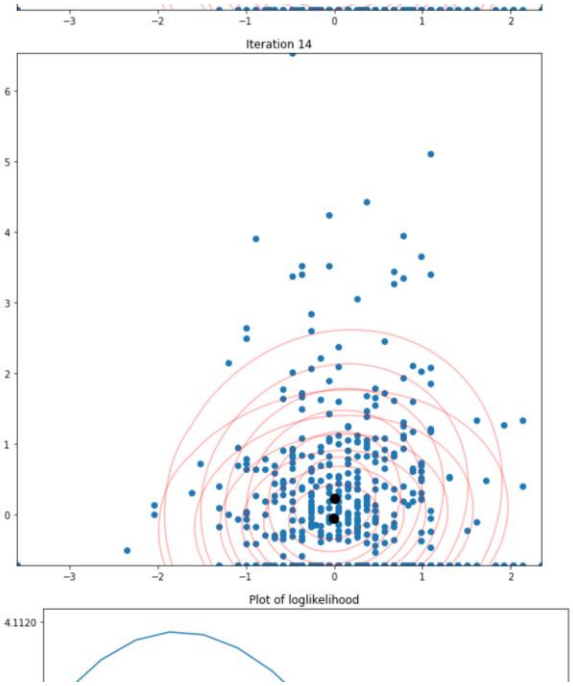
B.GMM

In task1, we use diabetes for both k means and Gmm.



This is the log likelihood function I got for the blood pressure and insulin features

This is the GMM plot for the same features where the iteration was for 15.



C.

K means only considers the Euclidean distance. But the GMM takes mean and covariance features also into account, due to which there is much more clear clustering in gmm. And also, it can be applied for complex data easily. But kmeans cannot give that accurate results

And kmeans uses spherical clustering, due to which the outliers can make the plot so distracted. It will not give correct clustering in case of outliers. But GMM will give correct clustering even if there are outliers, it is robust. It gives spherical, diagonal or oval or any other kind of clusters, based on clustering the data points. That is the main idea of using GMM

So, We use Gmm instead of kmeans.

2. Task 2 link: https://colab.research.google.com/drive/101MXC1Jra9LjX-4Z_Zkys9oSQFbgSAs_?usp=sharing

A. PCA using libraries:

First take the dataset, Since the activity column is not integer and is string, converting to integer.

```
[5]  # import pandas library
     import pandas as sw


     # this is dictionary file
     Activity_modified = {'WALKING': 2, 'WALKING_DOWNSTAIRS':4,'STANDING': 1, 'SITTING':5, 'WALKING_UPSTAIRS':3,'LAYING':6}

     # looking dataframe coc
     # For the values for which the key matches, we are looking for
     coc.Activity = [Activity_modified[it] for it in coc.Activity]
     print(coc)
```

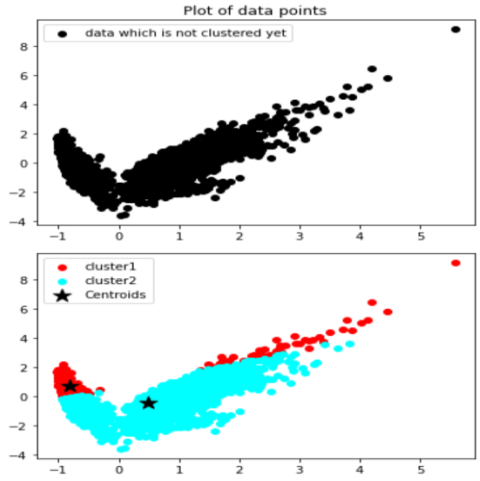
```
     7348         -0.029456          0.080585          0.117440  ...
     7349         -0.098913          0.332584          0.043999  ...
     7350         -0.068200          0.319473          0.101702  ...
     7351         -0.038678          0.229430          0.269013  ...

           fBodyBodyGyroJerkMag-kurtosis()  angle(tBodyAccMean,gravity)  \
     0                           -0.710304                    -0.112754
     1                           -0.861499                     0.053477
     2                           -0.760104                    -0.118559
     3                           -0.482845                    -0.036788
     4                           -0.699205                     0.123320
```

Then I have scaled the features and applied pca

Plotted a heat map, described the pca data. I found 3 principal components in my model'.

B. Applied Kmeans and GMM on the pca data.

```
(5882, 2)
[[-0.96978208  0.39514064]
 [ 1.16510017 -0.43004761]]
```



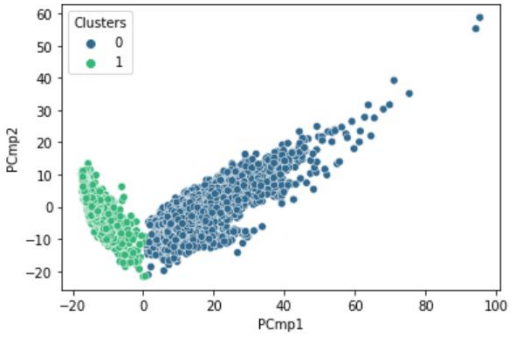
This is the graph for 2 principal components using Kmeans

Using inbuilt library also, I was able to get the same plot

:

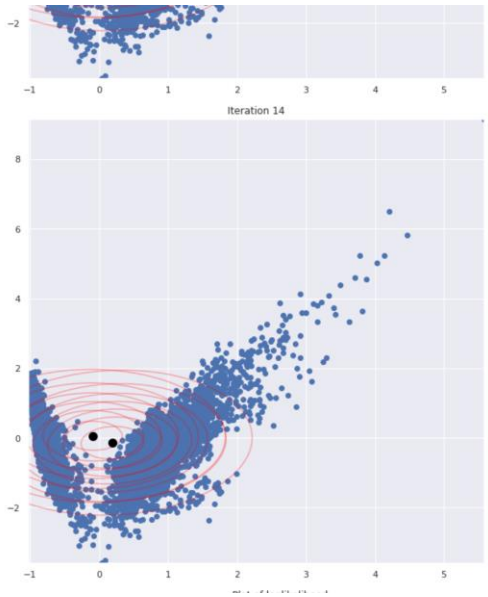
```
coc_pca['Clusters'] = km.labels_
ffd.scatterplot(x="PCmp1", y="PCmp2",hue = 'Clusters',  data=coc_pca
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcd056ac3d0>
```

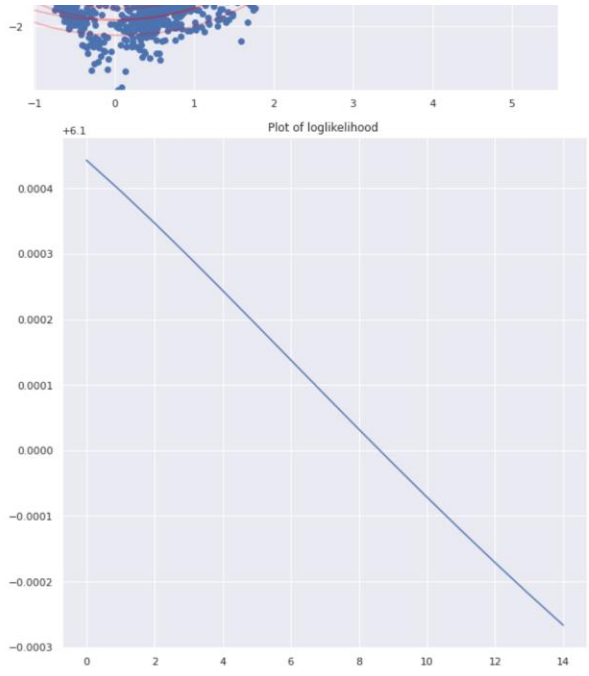



```
[ ]
```

Using GMM from task1, I got this plot:

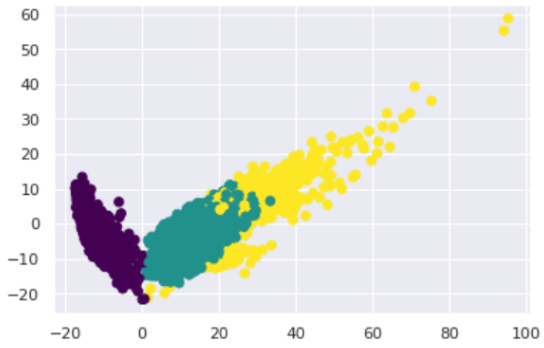


Log likelihood:



GMM using libraries with 3 pcs:

```
plt.scatter(ndarray[:, 0], ndarray[:, 1], c=labels, s=40, cmap='viridi
```




```
]
```

References for the same:

1. https://www.geeksforgeeks.org/reduce-data-dimentionality-using-pca-python/#:~:text=Steps%20to%20Apply%20PCA%20in%20Python%20for%20Dimensionality%20Reduction&text=In%20this%20example%2C%20we%20will,the%20sklearn%20library%20of%20Python.&text=All%20the%20necessary%20libraries%20required,Python3

2. https://youtu.be/K5w7q5BkaTI

3. https://levelup.gitconnected.com/gaussian-mixture-models-gmm-816f549940c5

4. https://www.youtube.com/watch?v=vtuH4VRq1AU

5. https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670