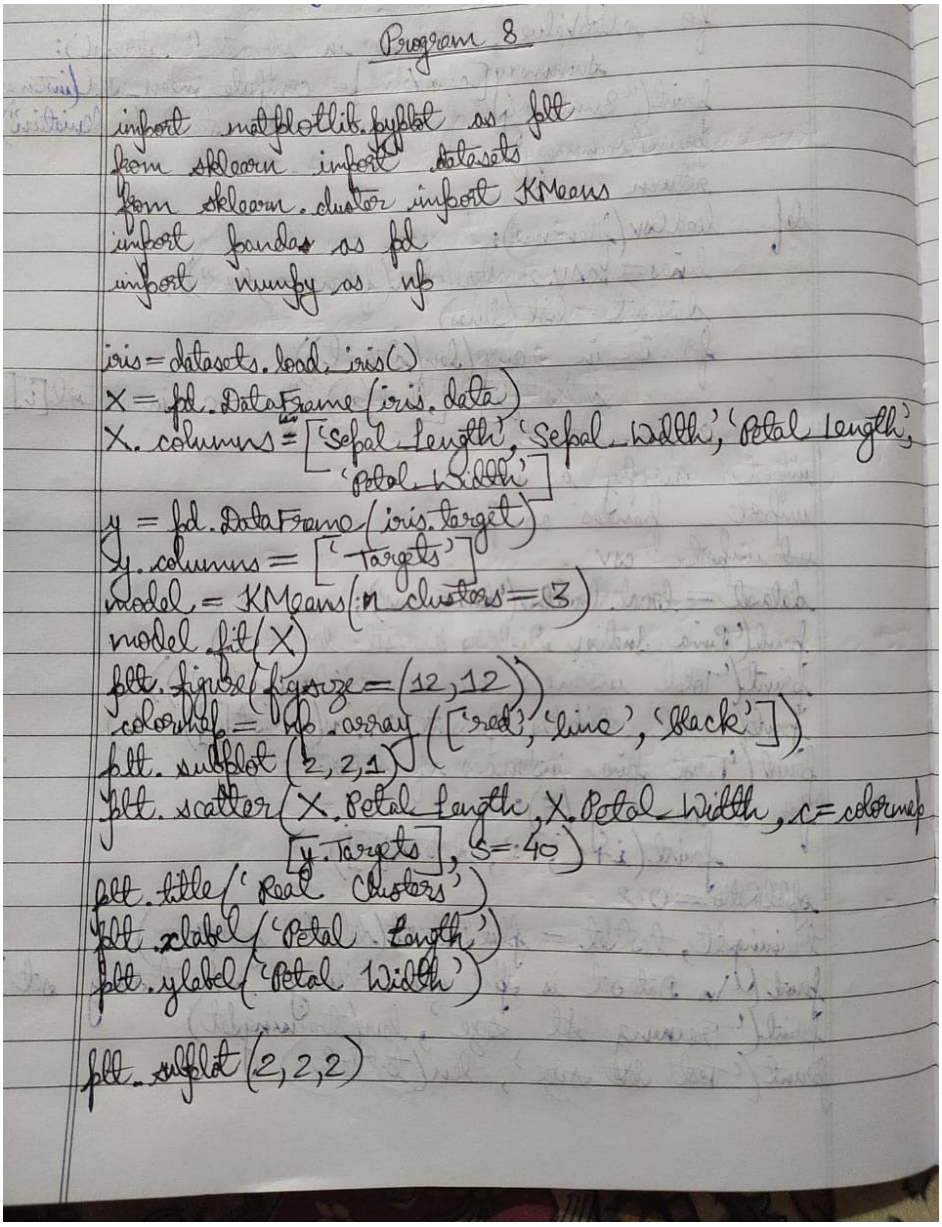
PROGRAM 8

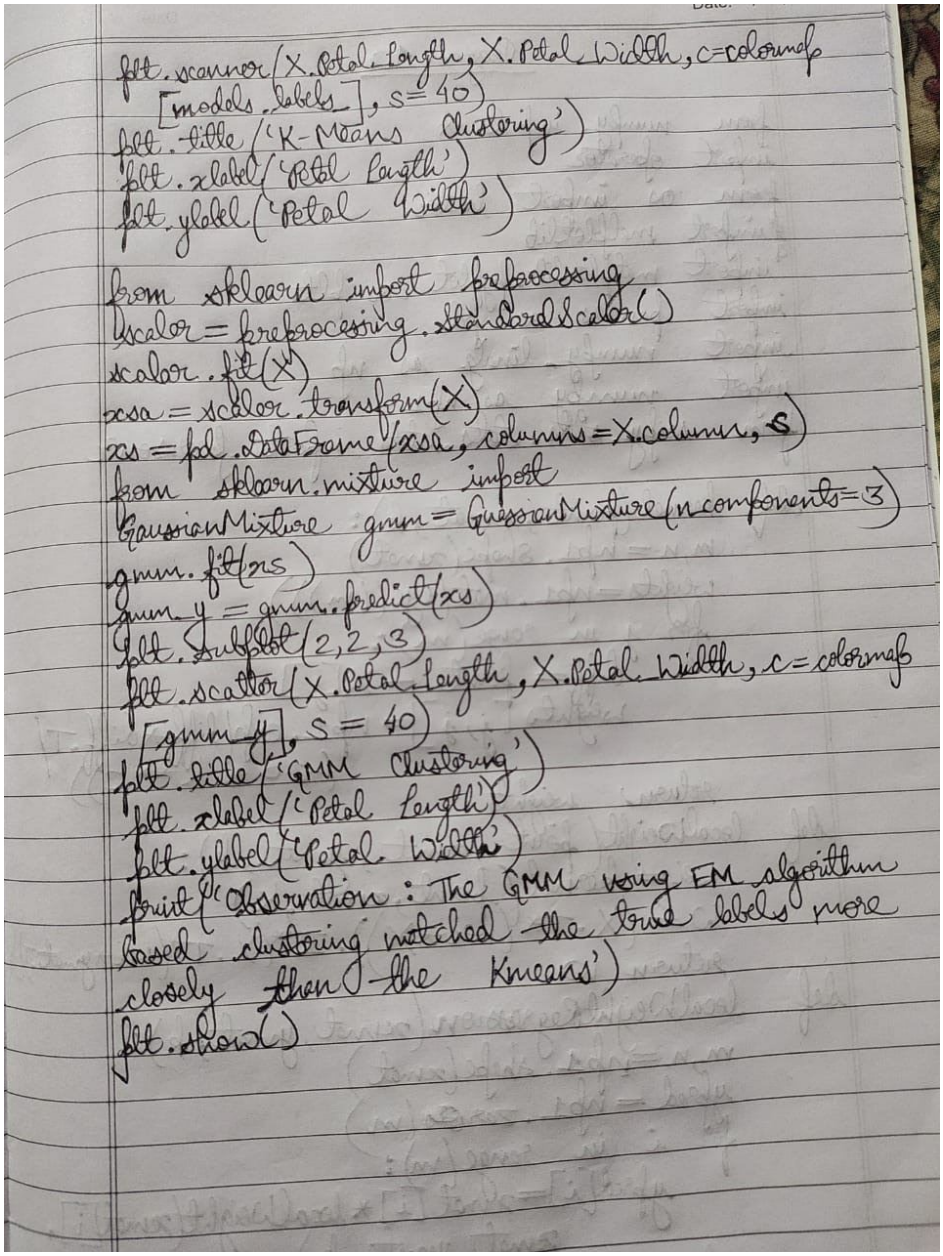


Program 8

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
import pandas as pd
import numpy as np

iris = datasets.load_iris()
X = pd.DataFrame(iris.data)
X.columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length',
             'Petal_Width']
y = pd.DataFrame(iris.target)
y.columns = ['Targets']
model = KMeans(n_clusters=3)
model.fit(X)
plt.figure(figsize=(12,12))
colormap = np.array(['red', 'lime', 'black'])
plt.subplot(2,2,1)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap
            [y.Targets], s=40)
plt.title('Real Clusters')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')

plt.subplot(2,2,2)
```



```
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap
    [models.labels_], s=40)
plt.title('K-Means Clustering')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')

from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
scaler.fit(X)
xsa = scaler.transform(X)
xs = pd.DataFrame(xsa, columns=X.columns, s)
from sklearn.mixture import
GaussianMixture gmm = GaussianMixture(n_components=3)
gmm.fit(xs)
gmm_y = gmm.predict(xs)
plt.subplot(2,2,3)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap
    [gmm_y], s=40)
plt.title('GMM Clustering')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
print('Observation: The GMM using EM algorithm
based clustering matched the true labels more
closely than the Kmeans')
plt.show()
```

OUTPUT

DATASET 1-hear_failure

A2- program no 1 | prog8(output).pdf | (12) WhatsApp | MLL/ | 8b - Jupyter Notel | 8a - Jupyter Notel | Untitled documen

localhost:8888/notebooks/MLL/8a.ipynb

jupyter 8a Last Checkpoint: 11 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [1]:
```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
import pandas as pd
```

In [2]:
```
X=pd.read_csv("heart_failure.csv")
x1 = X['age'].values
x2 = X['anaemia'].values
X = np.array(list(zip(x1, x2))).reshape(len(x1), 2)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [18]:
```
plt.plot()
plt.xlim([0, 100])
plt.ylim([0, 50])
plt.title('Dataset')
plt.scatter(x1, x2)
```

Out[18]: <matplotlib.collections.PathCollection at 0x1d3a4b61f08>

Dataset

Type here to search

ENG IN 17:44 10-12-2020

A2- program no 1 | prog8(output).pdf | (12) WhatsApp | MLL/ | 8b - Jupyter Notel | 8a - Jupyter Notel | Untitled documen

localhost:8888/notebooks/MLL/8a.ipynb

jupyter 8a Last Checkpoint: 12 minutes ago (unsaved changes)

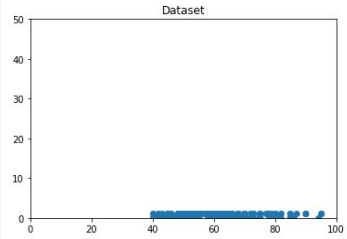
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Out[18]: <matplotlib.collections.PathCollection at 0x1d3a4b61f08>



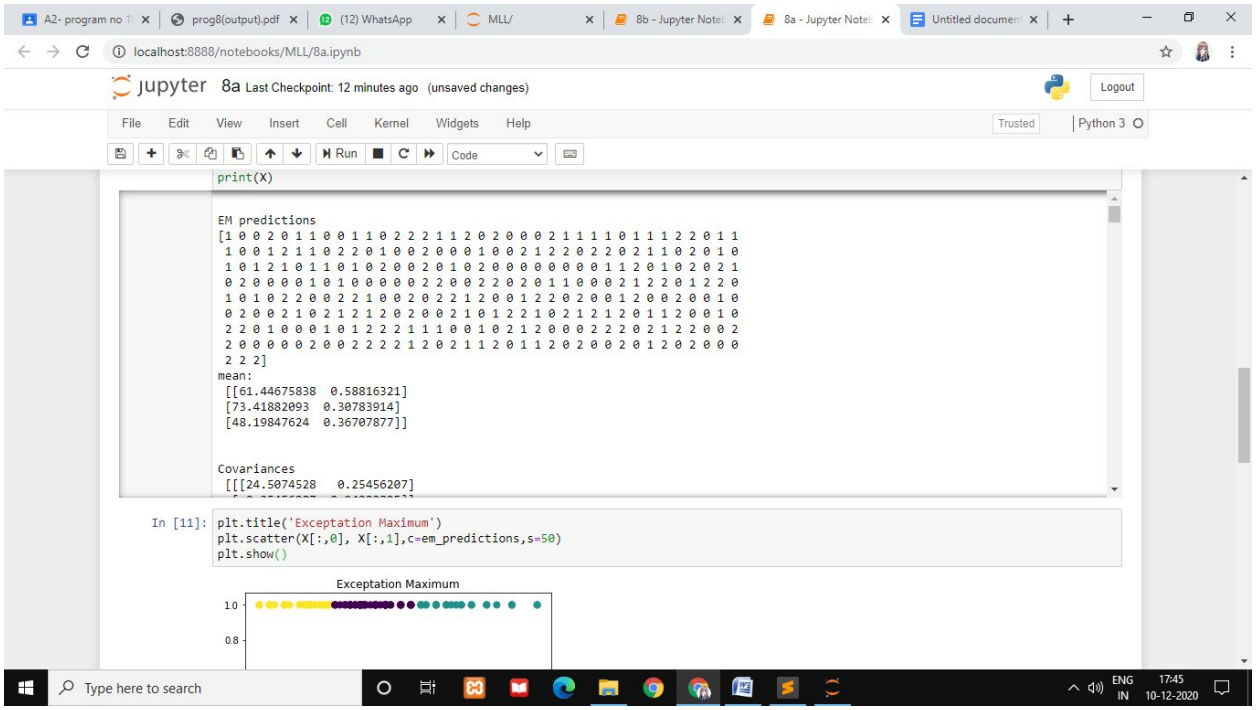
In [8]:
```
plt.show()
```

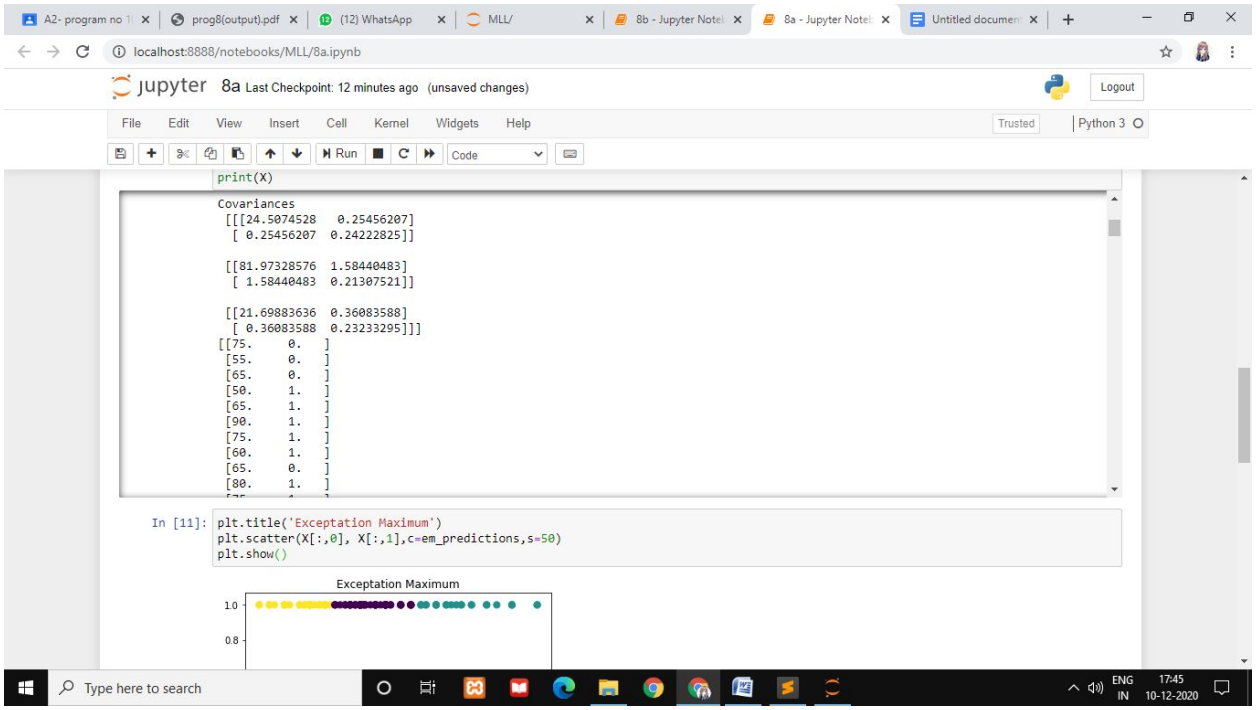
In [9]:
```
gmm = GaussianMixture(n_components=3)
gmm.fit(X)
em_predictions = gmm.predict(X)
```

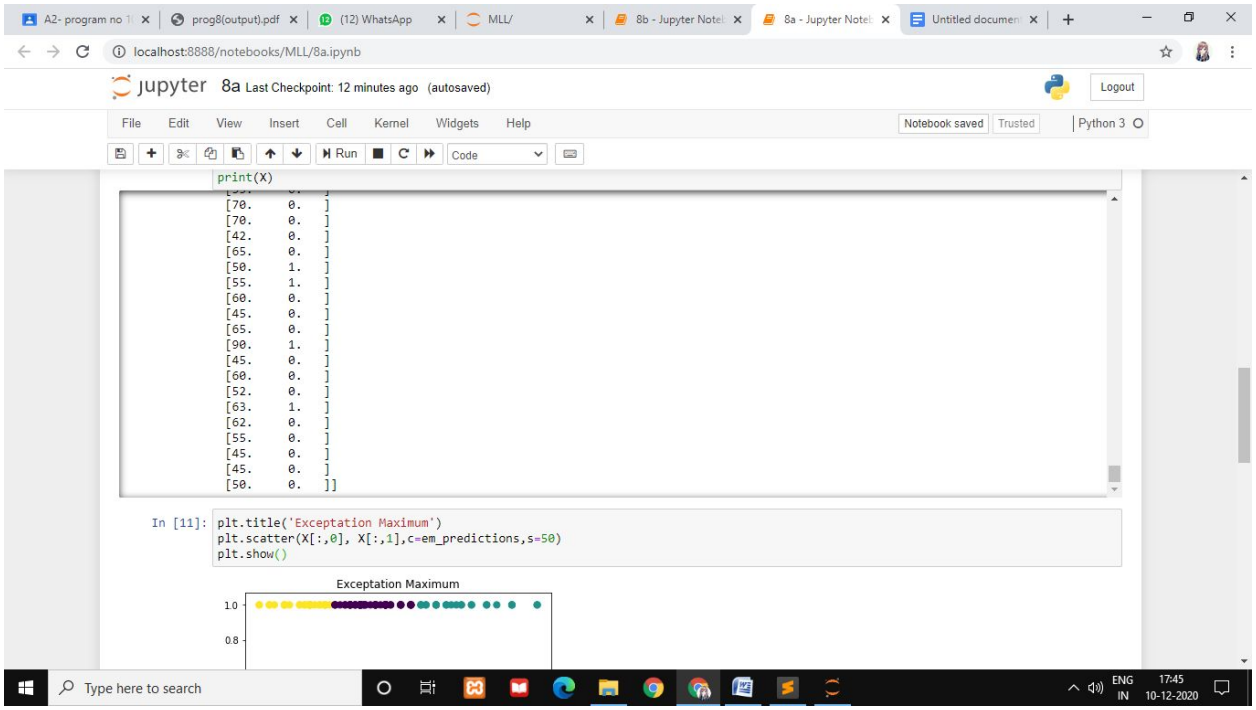
In [10]:
```
print("\nEM predictions")
print(em_predictions)
print("mean:\n",gmm.means_)
print('\n')
print("Covariances\n",gmm.covariances_)
print(X)
```

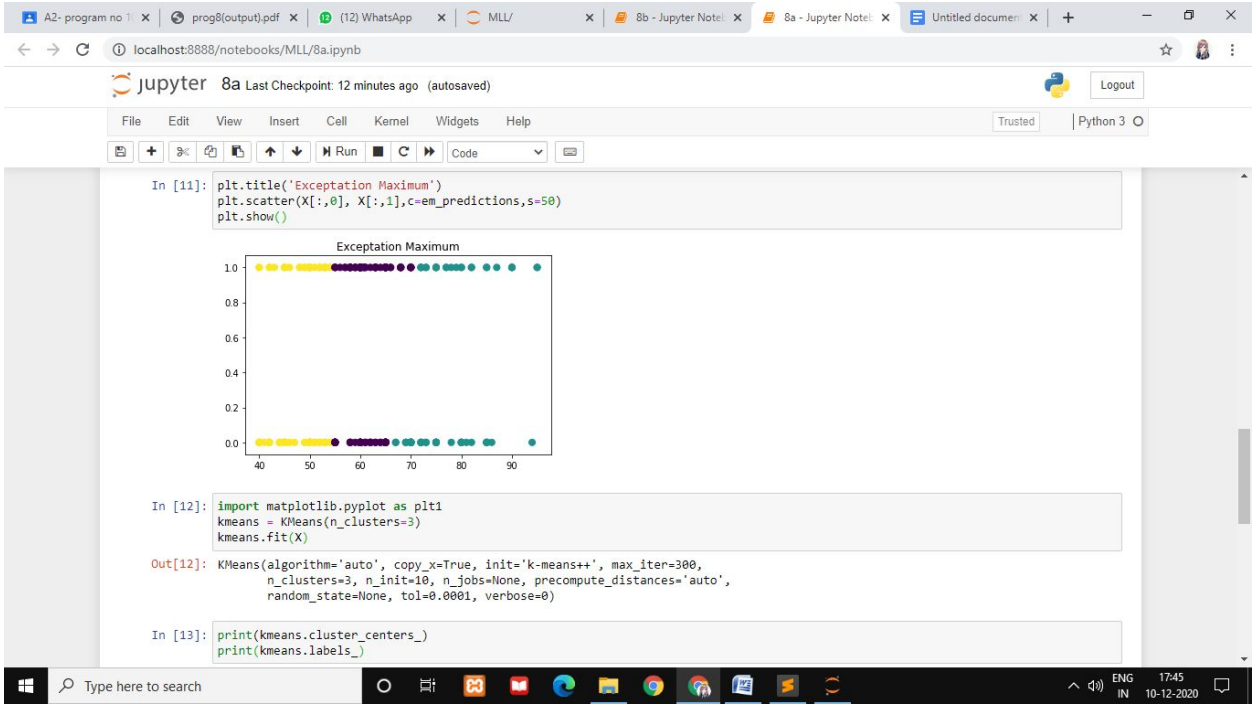
Type here to search

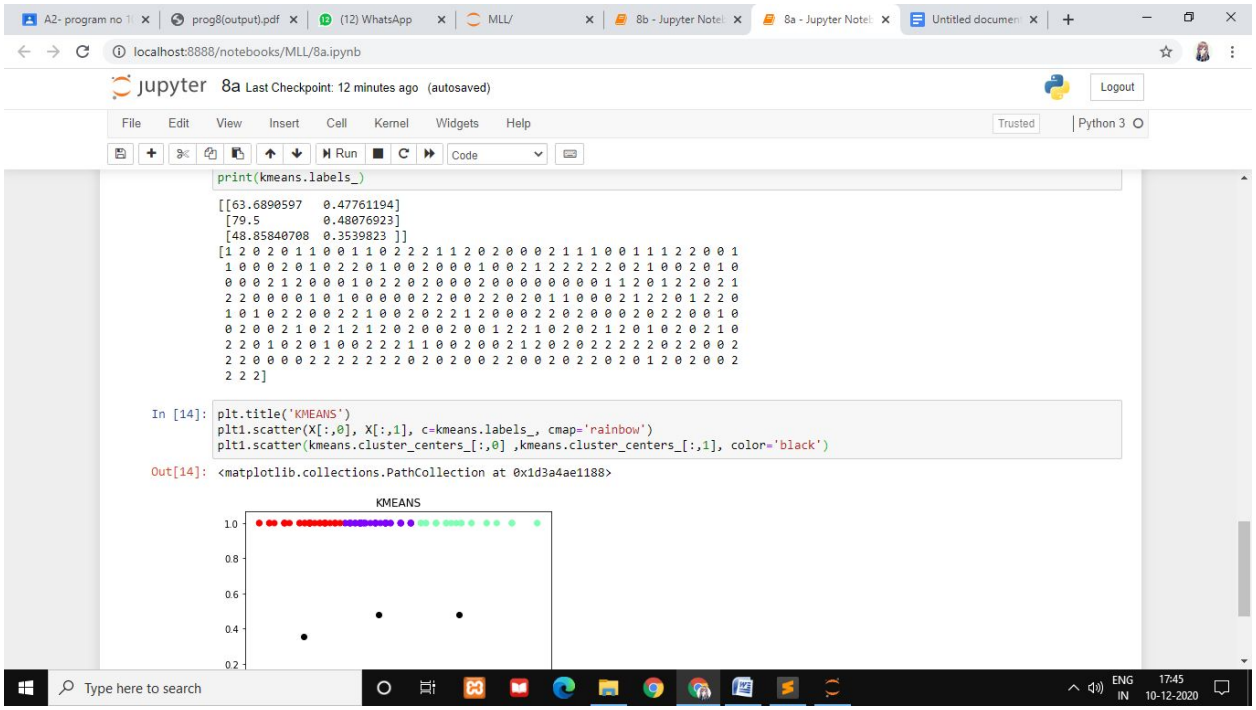
ENG IN 17:45 10-12-2020

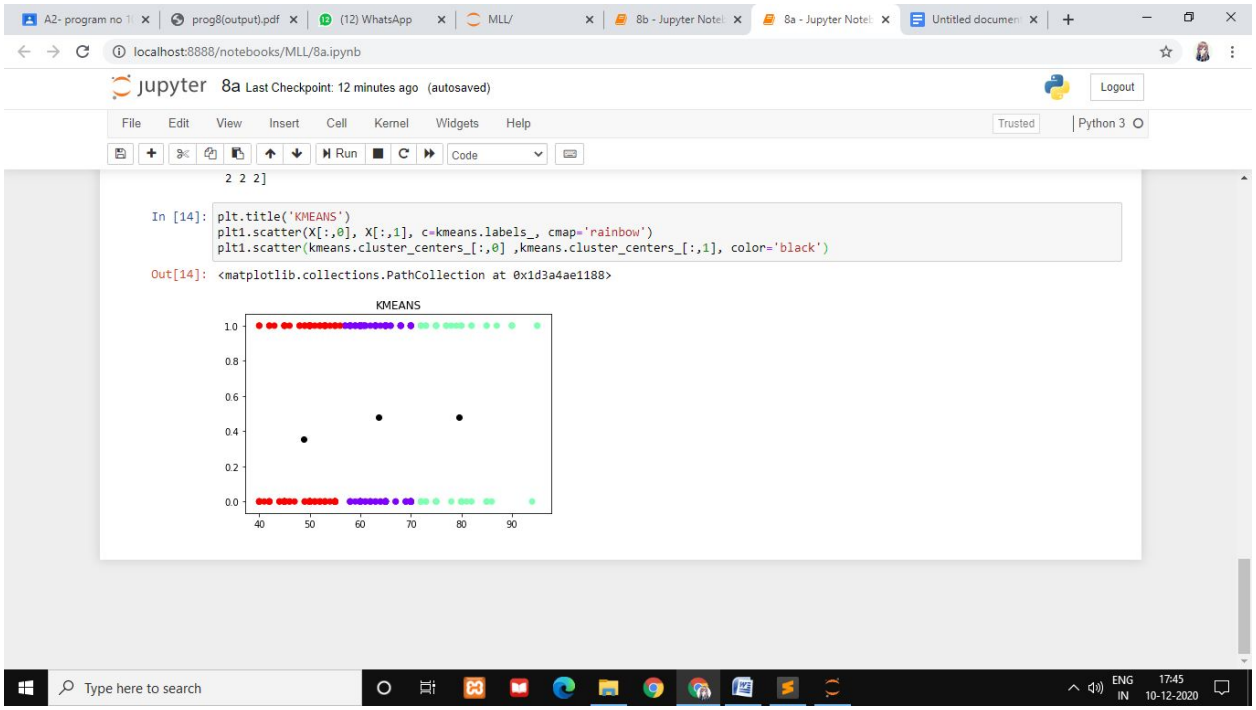

A2- program no 1
prog8(output).pdf
(12) WhatsApp
MLL/
8b - Jupyter Notel
8a - Jupyter Notel
Untitled documen
localhost:8888/notebooks/MLL/8a.ipynb
jupyter 8a Last Checkpoint: 12 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
print(X)
EM predictions
[1 0 0 2 0 1 1 0 0 1 1 0 2 2 2 1 1 2 0 2 0 0 0 2 1 1 1 1 0 1 1 1 2 2 0 1 1
1 0 0 1 2 1 1 0 2 2 0 1 0 0 2 0 0 0 1 0 0 2 1 2 2 0 2 2 0 2 1 1 0 2 0 1 0
1 0 1 2 1 0 1 1 0 1 0 2 0 0 2 0 1 0 2 0 0 0 0 0 0 0 0 1 1 2 0 1 0 2 0 2 1
0 2 0 0 0 0 1 0 1 0 0 0 0 0 2 2 0 0 2 2 0 2 0 2 0 1 1 0 0 0 2 1 2 2 0 1 2 2 0
1 0 1 0 2 2 0 0 2 2 1 0 0 2 0 2 2 1 2 0 0 1 2 2 0 2 2 0 0 1 2 0 0 2 0 0 1 0
0 2 0 0 2 1 0 2 1 2 1 2 0 2 0 0 2 1 0 1 2 2 1 0 2 1 2 1 2 0 1 1 2 0 0 1 0
2 2 0 1 0 0 0 1 0 1 2 2 2 1 1 1 0 0 1 0 2 1 2 0 0 0 2 2 2 0 2 1 2 2 0 0 2
2 0 0 0 0 0 2 0 0 2 2 2 2 1 2 0 2 1 1 2 0 1 1 2 0 2 0 0 2 0 1 2 0 2 0 0 0
2 2 2]
mean:
[[61.44675838 0.58816321]
[73.41882093 0.30783914]
[48.19847624 0.36707877]]
Covariances
[[[24.5074528 0.25456207]
In [11]:
plt.title('Exceptation Maximum')
plt.scatter(X[:,0], X[:,1],c=em_predictions,s=50)
plt.show()
Exceptation Maximum
1.0
0.8
Type here to search
ENG
IN
17:45
10-12-2020


A2- program no 1
prog8(output).pdf
(12) WhatsApp
MLL/
8b - Jupyter Notel
8a - Jupyter Notel
Untitled documen
localhost:8888/notebooks/MLL/8a.ipynb
jupyter 8a Last Checkpoint: 12 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
print(X)
Covariances
[[[24.5074528 0.25456207]
[0.25456207 0.24222825]]
[[81.97328576 1.58440483]
[1.58440483 0.21307521]]
[[21.69883636 0.36083588]
[0.36083588 0.23233295]]]
[[75. 0.]
[55. 0.]
[65. 0.]
[50. 1.]
[65. 1.]
[90. 1.]
[75. 1.]
[60. 1.]
[65. 0.]
[80. 1.]
In [11]:
plt.title('Exceptation Maximum')
plt.scatter(X[:,0], X[:,1],c=em_predictions,s=50)
plt.show()
Exceptation Maximum
1.0
0.8
Type here to search
ENG
IN
17:45
10-12-2020


A2- program no 1
prog8(output).pdf
(12) WhatsApp
MLL/
8b - Jupyter Notel
8a - Jupyter Notel
Untitled documen
localhost:8888/notebooks/MLL/8a.ipynb
jupyter 8a Last Checkpoint: 12 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Notebook saved Trusted Python 3
Run Code
print(X)
[70. 0.]
[70. 0.]
[42. 0.]
[65. 0.]
[50. 1.]
[55. 1.]
[60. 0.]
[45. 0.]
[65. 0.]
[90. 1.]
[45. 0.]
[60. 0.]
[52. 0.]
[63. 1.]
[62. 0.]
[55. 0.]
[45. 0.]
[45. 0.]
[50. 0.]]
In [11]:
plt.title('Exceptation Maximum')
plt.scatter(X[:,0], X[:,1],c=em_predictions,s=50)
plt.show()
Exceptation Maximum
1.0
0.8
Type here to search
ENG IN
17:45 10-12-2020


A2- program no 1
prog8(output).pdf
(12) WhatsApp
MLL/
8b - Jupyter Notel
8a - Jupyter Notel
Untitled documen
localhost:8888/notebooks/MLL/8a.ipynb
jupyter 8a Last Checkpoint: 12 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [11]:
plt.title('Exceptation Maximum')
plt.scatter(X[:,0], X[:,1],c=em_predictions,s=50)
plt.show()
Exceptation Maximum
1.0
0.8
0.6
0.4
0.2
0.0
40
50
60
70
80
90
In [12]:
import matplotlib.pyplot as plt1
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
Out[12]:
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
random_state=None, tol=0.0001, verbose=0)
In [13]:
print(kmeans.cluster_centers_)
print(kmeans.labels_)
Type here to search
ENG IN
17:45 10-12-2020





DATASET 2- Iris Dataset

In [1]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
import pandas as pd
import numpy as np
```

In [2]:
```
iris = datasets.load_iris()
X = pd.DataFrame(iris.data)
X.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']
y = pd.DataFrame(iris.target)
y.columns = ['Targets']
```

In [3]:
```
model = KMeans(n_clusters=3)
model.fit(X)
```

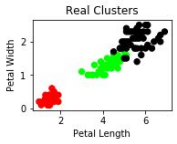
Out[3]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [4]:
```
plt.figure(figsize=(12,12))
colormap = np.array(['red', 'lime', 'black'])
```

```
<Figure size 864x864 with 0 Axes>
```

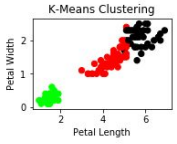
In [5]:
```
plt.subplot(2, 2, 1)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Real Clusters')
plt.xlabel('Petal Length')
```


Out[5]:
```
Text(0, 0.5, 'Petal Width')
```



In [6]:
```
plt.subplot(2, 2, 2)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[model.labels_], s=40)
plt.title('K-Means Clustering')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
```

Out[6]:
```
Text(0, 0.5, 'Petal Width')
```



jupyter 8b Last Checkpoint: 7 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [7]:
```
from sklearn import preprocessing
scaler=preprocessing.StandardScaler()
scaler.fit(X)
```

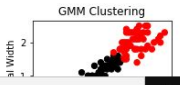
Out[7]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

In [8]:
```
xsa=scaler.transform(X)
xs=pd.DataFrame(xsa,columns=X.columns)
from sklearn.mixture import GaussianMixture
gmm=GaussianMixture(n_components=3)
gmm.fit(xs)
```

Out[8]:
```
GaussianMixture(covariance_type='full', init_params='kmeans', max_iter=100,
                means_init=None, n_components=3, n_init=1, precisions_init=None,
                random_state=None, reg_covar=1e-06, tol=0.001, verbose=0,
                verbose_interval=10, warm_start=False, weights_init=None)
```

In [9]:
```
gmm_y=gmm.predict(xs)
plt.subplot(2,2,3)
plt.scatter(X.Petal_Length,X.Petal_Width,c=colormap[gmm_y],s=40)
plt.title('GMM Clustering')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
```

Out[9]:
```
Text(0, 0.5, 'Petal Width')
```



Type here to search ENG IN 17:43 10-12-2020

A2- program no 1 | prog8(output).pdf | (12) WhatsApp | MLL/ | 8b - Jupyter Notel | 8a - Jupyter Notel | Untitled documen
localhost:8888/notebooks/MLL/8b.ipynb

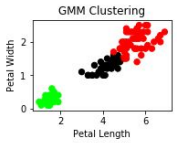
jupyter 8b Last Checkpoint: 7 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

plt.scatter(X.Petal_Length,X.Petal_Width,c=colormap[gmm_y],s=40)
plt.title('GMM Clustering')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')

Out[9]: Text(0, 0.5, 'Petal Width')



In [10]:
```
print('Observation: The GMM using EM algorithm based clustering matched the true labels more closely than the Kmeans')
plt.show()
```

```
Observation: The GMM using EM algorithm based clustering matched the true labels more closely than the Kmeans
```

In [ ]:

Type here to search ENG IN 17:43 10-12-2020