Apply EM algorithm to cluster a set of data stored in a .csv file

Aim:-

To apply EM algorithm to cluster a set of data stored in a .csv file

Program:-

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets.samples_generator import make_blobs
X, y_true = make_blobs(n_samples=100, centers = 4, cluster_std = 0.60,
random_state=0)
X = X[:,::-1]
from sklearn.mixture import GaussianMixture
gmm = GaussianMixture(n_components = 4).fit(X)
labels = gmm.predict(X)
plt.scatter(X[:,0], X[:,1], c=labels, s=40, cmap="viridis");
probs = gmm.predict_proba(X)
print(probs[:5].round(3))
size = 50 * probs.max(1)** 2
plt.scatter(X[:,0], X[:,1], c=labels, cmap="viridis", s=size);
from matplotlib.patches import Ellipse
def draw_ellipse(position, covariance, ax=None, **kwargs):
    """Draw an ellipse with a given position and covariance"""
Ax = ax or plt.gca()
if covariance.shape == (2,2):
```

```
        U, S, Vt = np.linalg.svd(covariance)
        Angle = np.degrees(np.arctan2(U[1,0], U[0,0]))
        width, height = 2 * np.sqrt(s)
    else:
        angle = 0
        width, height = 2 * np.sqrt(covariance)
    for nsig in range(1,4):
        ax.add_patch(Ellipse(position, nsig * width, nsig * height, angle,
                             **kwargs))

def plot_gmm(gmm, X, label = True, ax = None):
    ax = ax or plt.gca()
    labels = gmm.fit(X).predict(X)
    if label:
        ax.scatter(X[:,0], X[:,1], c=labels, s=40, cmap="viridis",
                   zorder=2)
    else:
        ax.scatter(X[:,0], X[:,1], s=40, zorder=2)
ax.axis("equal")
w_factor = 0.2 / gmm.weights_.max()
for pos, covar, w in zip(gmm.means_, gmm.covariances_, gmm.weights_):
    draw_ellipse(pos, covar, alpha = w * w_factor)
gmm = GaussianMixture(n_components = 4, random_state = 42)
plot_gmm(gmm, X)
gmm = GaussianMixture(n_components = 4, covariance_type = "full",
                      random_state = 42)
plot_gmm(gmm, X)
```

Result :-

Thus the program to apply EM algorithm to cluster a set of data stored in a .csv file has been executed successfully.

Output:-

```
[[1,0,0,0]
 [0,0,1,0]
 [1,0,0,0]
 [1,0,0,0]
 [1,0,0,0]]
```