Capstone Project Report

Name: **Pooja Ramdas Kadam**
Course: AI & ML (Batch - 4)

Problem Statement

Using a **gaussian mixture model**, perform a simple clustering on the given **2D Dataset**. Try to find the optimal number of clusters using python (you may use any module to implement this). Now implement the same from scratch using python and a dummy dataset generated using **scikit learn dataset** generating functions such as **make blob.**

Prerequisites

Along with Python below packages needed to be installed

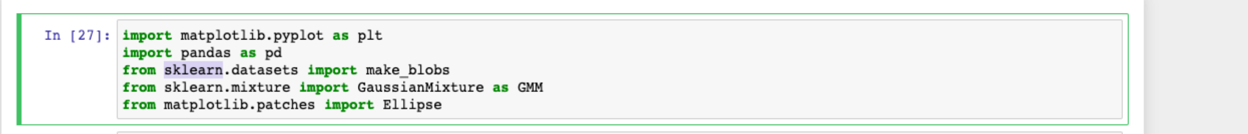
Matplotlib
Pandas
Sklearn

Dataset Used

https://cdn.analyticsvidhya.com/wp-content/uploads/2019/10/Clustering_gmm.csv

Implementation

Import required libraries and load data



```
In [27]: import matplotlib.pyplot as plt
         import pandas as pd
         from sklearn.datasets import make_blobs
         from sklearn.mixture import GaussianMixture as GMM
         from matplotlib.patches import Ellipse
```

Load data

```
In [28]: data = pd.read_csv('Clustering_gmm.csv')
```

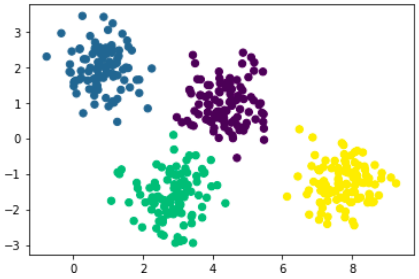
```
In [29]: data.head(10)
```

Out[29]:

	Weight	Height
0	67.062924	176.086355
1	68.804094	178.388669
2	60.930863	170.284496
3	59.733843	168.691992
4	65.431230	173.763679
5	61.577160	168.091751
6	63.341866	170.642516
7	61.041643	170.096682
8	62.633623	171.862972
9	53.407860	162.756843

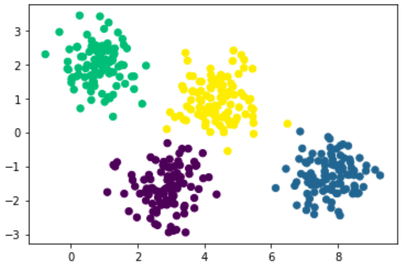
```
In [31]: X, y_true = make_blobs(n_samples=400, centers=4,cluster_std=0.60, random_state=0)
         X = X[:, ::-1]
```

```
In [32]: plt.scatter(X[:, 0], X[:, 1], c=y_true, s=40, cmap='viridis')
         plt.show()
```



Apply GMM

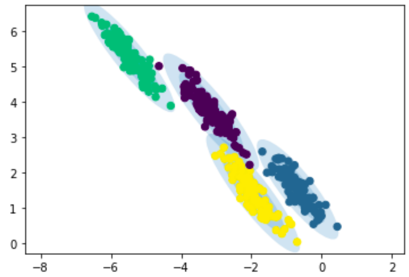
```
In [33]: gmm = GMM(n_components=4).fit(X)
         labels = gmm.predict(X)
         plt.scatter(X[:, 0], X[:, 1], c=labels, s=40, cmap='viridis')
         plt.show()
```



Apply GMM with ellipses

In [35]:

```
gmm = GMM(n_components=4, covariance_type='full', random_state=42)
rng = np.random.RandomState(13)
X_stretched = np.dot(X, rng.randn(2, 2))
plot_gmm(gmm, X_stretched)
```


6
5
4
3
2
1
0
−8
−6
−4
−2
0
2