Assignment Project Report
Gaussian Mixture Models: Bag of Words Representation

Name: Pavan Tiwari

Course: AI and 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.

  Software:
  - Python 3 (Use anaconda as your python distributor as well)

  - Tools:
    - Pandas
    - Numpy
    - Matplotlib
    - Seaborn
    - OpenCv
    - Sklearn

  - Dataset Link:
    Clustering_GMM
    https://cdn.analyticsvidhya.com/wpcontent/uploads/2019/10/Clustering_gmm.csv

- **Method Used**

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset allowing the model to learn automatically, i.e. in an unsupervised manner. The bag-of-words model is a way of representing text data when modelling text with machine learning algorithms which can be combined with GMM to get a useful model representation.

- **Implementation:**

1. Load all required libraries

Importing Libraries

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.mixture import GaussianMixture
from sklearn.cluster import KMeans
```

In [2]:
```
# Reading Dataset
data = pd.read_csv('Clustering_gmm.csv')
data.head()
```

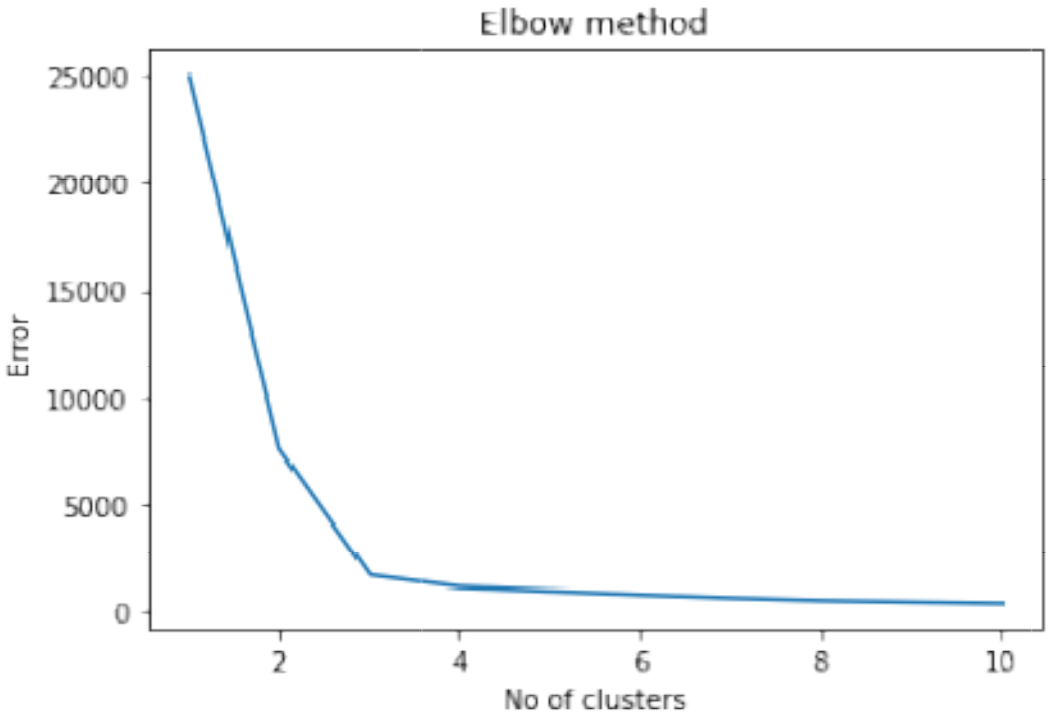
Out[2]:

	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

2. Deciding Number of Cluster

Elbow Method

:
```
Error =[]
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i).fit(data)
    kmeans.fit(data)
    Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 11), Error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
```



3. Model Building Using GMM On Clustering_GMM Dataset

Gaussian Mixture Models - Bag of Words Representation Last Checkpoint: Last Friday at 00:52

View Insert Cell Kernel Widgets Help

Run Markdown

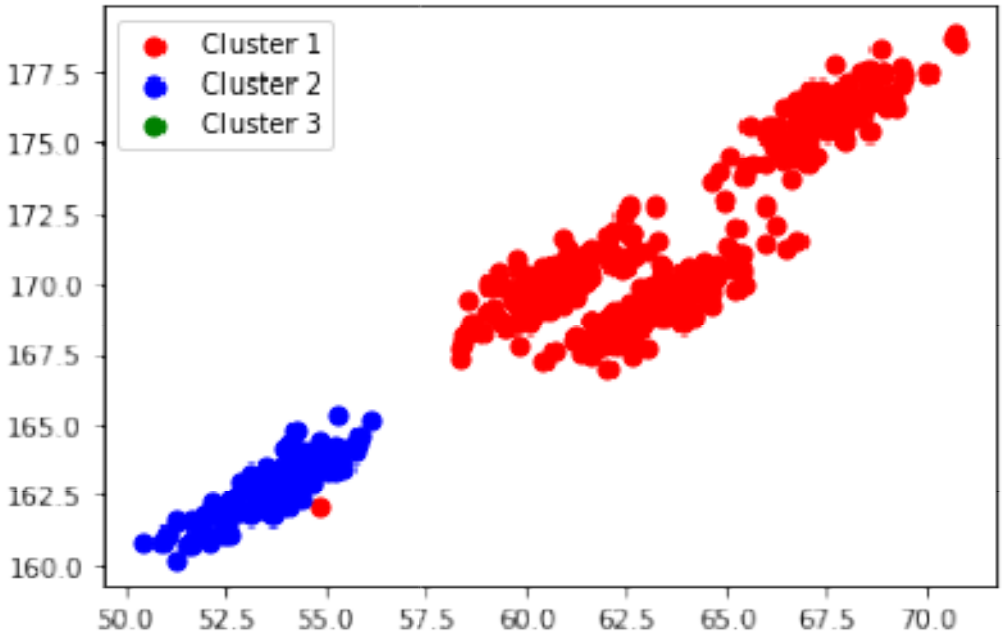
```
1 gm = GaussianMixture(n_components=2, random_state=0).fit(data)
```

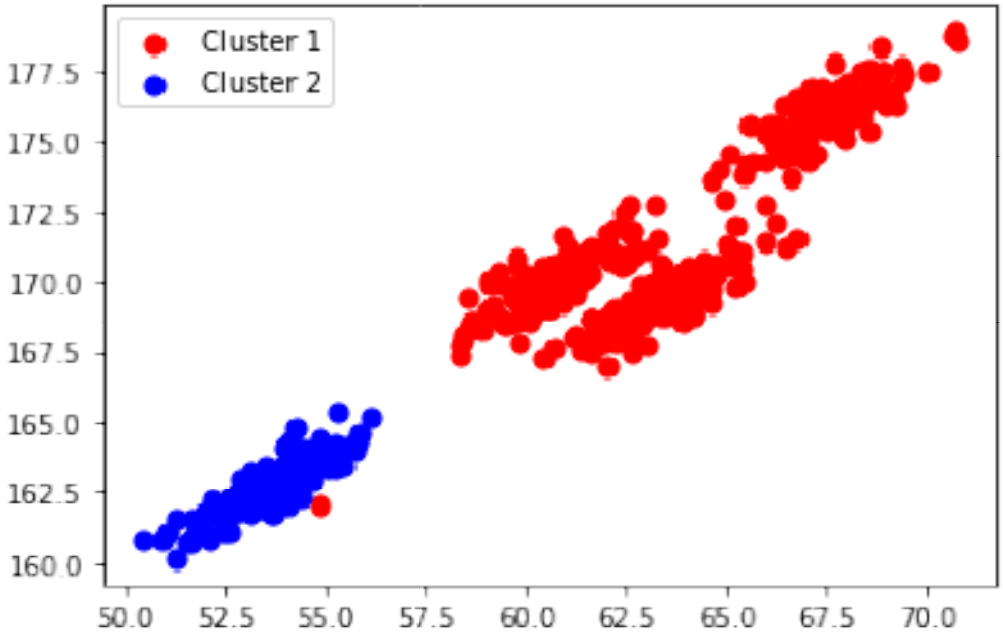
```
1 gm.means_
```

```
array([[ 63.77281821, 171.71722858],
       [ 53.57474006, 162.74626605]])
```

```
1 y = gm.predict(data)
```

```
1 x = np.array(data)
2 plt.scatter(x[y== 0, 0], x[y == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
3 plt.scatter(x[y == 1, 0], x[y == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')
4
5
6 plt.legend()
```



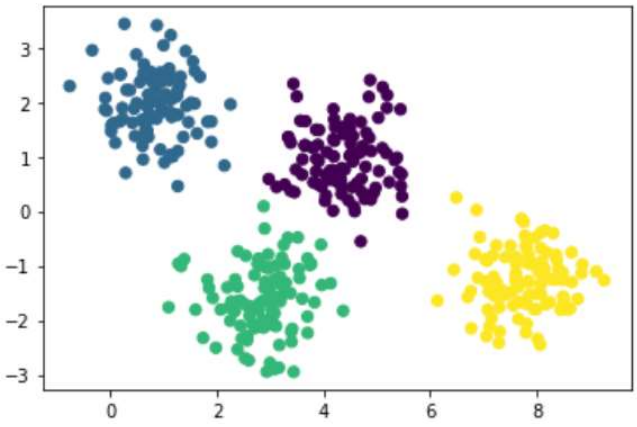


4. Importing Libraries and Making Dummy dataset

```
from sklearn.datasets import make_blobs
```

```
X, y_true = make_blobs(n_samples=400, centers=4,
                       cluster_std=0.60, random_state=0)
X = X[:, ::-1] # flip axes for better plotting
```

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



5. Model Building

[18]:

```
from sklearn.mixture import GaussianMixture as GMM
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()
```