Govardhan Seshasayee

Gaussian Mixture model: Bag of words representation

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.

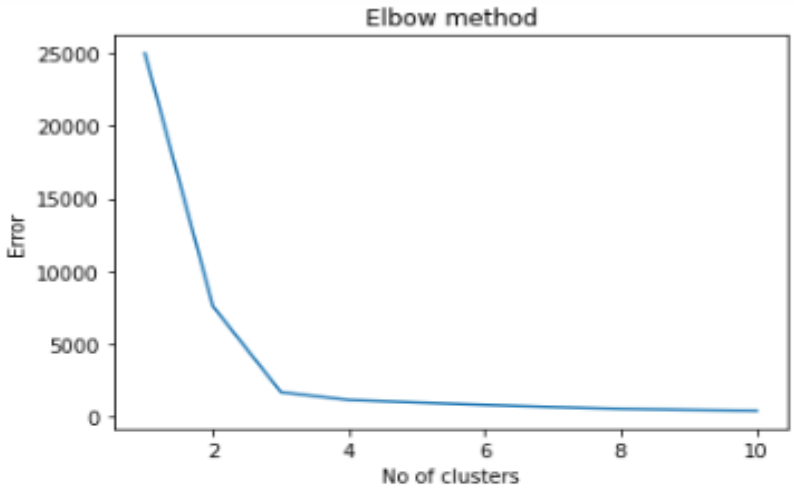
Implementation: Bag of words representation

```
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
```

```
data = pd.read_csv('Clustering_gmm.csv')
data.head()
```

	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

```
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()
```




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

```
gm.means_
```

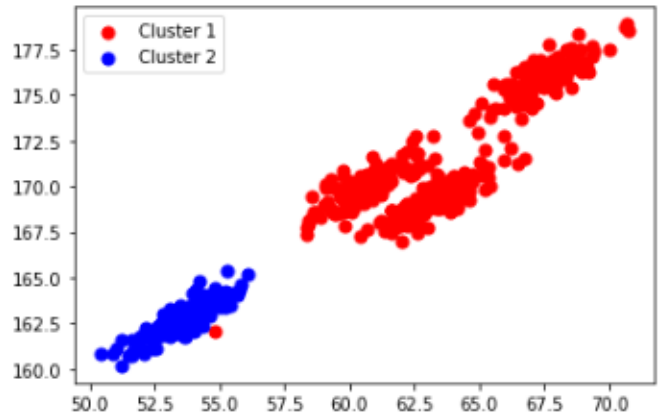
```
array([[ 63.77281821, 171.71722858],
       [ 53.57474006, 162.74626605]])
```

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

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


plt.legend()
```

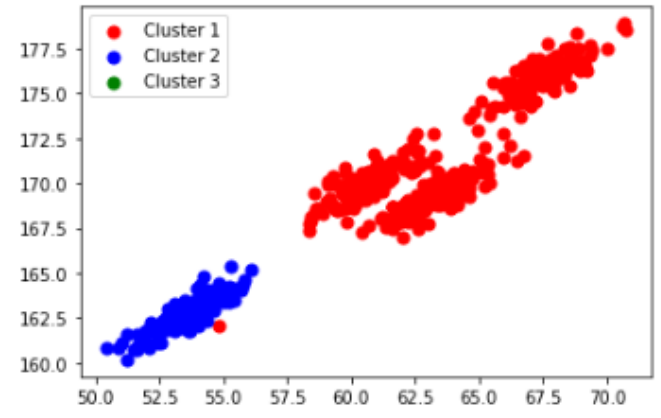
```
<matplotlib.legend.Legend at 0x235c6ce3c70>
```




```
gm = GaussianMixture(n_components=2, random_state=0).fit(data)
y1 = gm.predict(data)
x = np.array(data)
plt.scatter(x[y1== 0, 0], x[y1 == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
plt.scatter(x[y1 == 1, 0], x[y1 == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')
plt.scatter(x[y1 == 2, 0], x[y1 == 2, 1], s = 50, c = 'green', label = 'Cluster 3')

plt.legend()
```

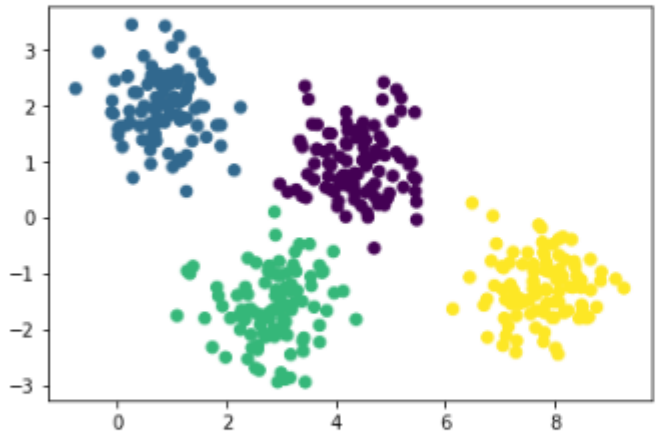
```
<matplotlib.legend.Legend at 0x235c6cf1e20>
```




```
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()
```




```
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()
```