CASE STUDY 5: MODEL SELECTION
Name: Sruthi Merlin Thomas ID:2496855T

Introduction

This case study explores different model selection strategies needed to find an optimal model for the training data set. Model Selection in Machine learning is the process of finding a suitable approach, hyperparameters or a set of features. Models are selected based on certain qualities like accuracy, ease of interpretation, performance, speed etc. This case study focusses on the Model Selection in Clustering Models particularly the **Gaussian Mixture Models.**

Clustering Algorithms

Clustering Algorithms are unsupervised form of learning algorithms where similar data points are grouped together based on their features.

1. K-Means Algorithm

One important clustering algorithm is the K-Means clustering in which the data population is divided into K Clusters each containing a centroid based on the parameter value K. The data points closest to the centroid form a cluster. Figure 1 represents a 2-dimensional example of a K- Means Cluster.

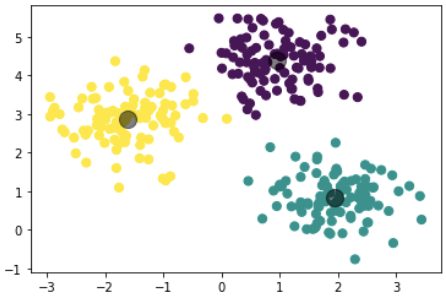


Figure 1: K-Means Clustering

The value of the parameter K i.e. the number of clusters can be determined by various methods.

1) Elbow Method:

This method finds the optimal value of K by plotting the sum of squares for each value of K. The plot like looks like an arm and the elbow point of the plot is taken as the optimal value of K. Figure 2 illustrates the Elbow point method. From this figure we can infer that the optimal value of K is 3.

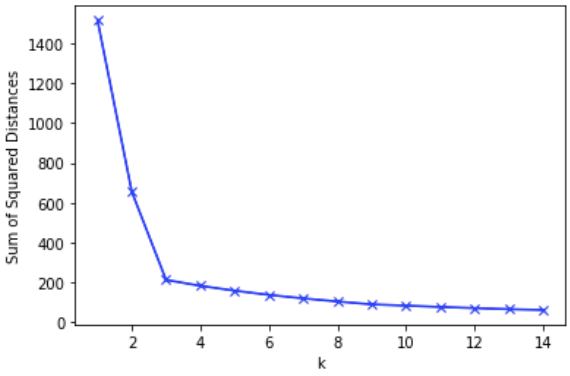


Figure 2: Elbow Method Plot

2) **Silhouette Score Method:**
Silhouette analysis uses the distance between the clusters as measure to find optimal number clusters. The figure shows the optimal value for number of clusters is 3.

```
For n_clusters = 2 The average silhouette_score is: 0.55
For n_clusters = 3 The average silhouette_score is: 0.66
For n_clusters = 4 The average silhouette_score is: 0.54
For n_clusters = 5 The average silhouette_score is: 0.44
For n_clusters = 6 The average silhouette_score is: 0.33
For n_clusters = 7 The average silhouette_score is: 0.33
```

Figure 3: Silhouette Score Analysis.

Disadvantages of K- Means Clustering Algorithm:
Although the K – means clustering is easy to implement, has a guaranteed convergence and scales easily to large data sets, it has certain disadvantages.

- It is difficult to predict the value of K.
- K- Means has trouble clustering data of clusters of different sizes.
- The final result is dependent on the initial value of k we randomly choose i.e. Sensitive to choose of initial seeds
- Curse of Dimensionality: As the number of Dimensions increases, the distance measure converges to a constant value.

2. **Gaussian Mixture Models**
Gaussian Mixture Models are another form unsupervised clustering method in which probabilistic models are used to distribute the data points to different clusters. Unlike K-Means, it uses soft clustering technique to assign data points to gaussian distribution. IIn this distribution, the clusters can be of any shape.

Input Data:
The input data used for the model selection in Gaussian Mixture Models is the digits data. In this case we generate and plot each digit as shown in the figure.

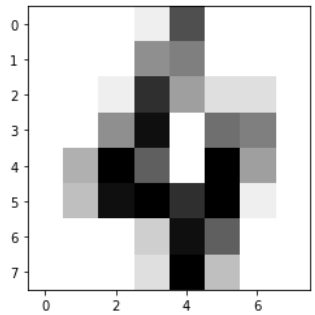


Figure 4: Digit Plot of 4

The **Principal Component analysis** is applied as visualizing the data at higher dimensions is difficult. The reduced data projected to first two principal components is used to analyze and choose parameters to select suitable Gaussian Mixture Model.

Basic Task Definition:
Implement a Gaussian Mixture Model for the reduced digits data and use different strategies to determine the number of clusters K and the covariance structure type.

The next section describes the core idea of Gaussian mixture model, different approaches used for model selection and its python implementation.

Methods

Core idea of Gaussian Mixture Model:

In Gaussian Mixture Models, datasets are modeled using Gaussian Distribution. A Gaussian Mixture model is parameterized by two values the mean and the covariance of the distribution.

Probability Density Function of a Multivariate Distribution:

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$$

where μ is the mean vector and Σ is the covariance matrix of the distribution. Hence for a data set with n features the mean vector is of length n and covariance is a matrix of n x n. Additionally, we have parameter Π which represents the density of the distribution.

Estimating the parameters:

The parameters in Gaussian Mixture Models are determined in two steps i.e the **Expectation** and the **Maximization.**

Expectation Maximization Algorithm:

This is a statistical approach in finding the suitable model parameters when the data has latent variables or in other when data is incomplete. This method uses the existing data to find the most optimal parameters for the model.

1) We start by randomly initializing the number of clusters K and randomly assigning the values of μ,Σ and Π for k clusters c1, c2,….ck.

E-step:

For each data point in the data set, we calculate the probability that it belongs to the clusters c1,c2,…ck.
This can be calculated by the formula:

$$r_{ic} = \frac{\text{Probability Xi belongs to c}}{\text{Sum of probability Xi belongs to } c_1, c_2, .. c_k} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

M-step:

1) In this step, we calculate the new density of the distribution given by the formula:

$$\Pi = \frac{\text{Number of points assigned to cluster}}{\text{Total number of points}}$$

2) The updated values of mean and covariance are calculated based on the new density of the distribution:

$$\boldsymbol{\mu} = \frac{1}{\text{Number of points assigned to cluster}} \Sigma_i \; r_{ic} x_i$$

$$\Sigma_c = \frac{1}{\text{Number of points assigned to cluster}} \Sigma_i \; r_{ic} (x_i - \boldsymbol{\mu}_c)^T (x_i - \boldsymbol{\mu}_c)$$

We then go back to the E- step to calculate the new probabilities of each data points based on the updated mean and covariance values. This process is repeated till the convergence has occurred. In other words, till the **log likelihood** is maximized.

Gaussian Mixture Model Implementation in Scikit Learn:

Scikit Learn has an inbuilt class Gaussian Mixture, that represents the Gaussian Mixture model probability distribution.

class sklearn.mixture.GaussianMixture(n_components=1, covariance_type='full', tol=0.001, reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans', weights_init=None, means_init=None, precisions_init=None, random_state=None, warm_start=False, verbose=0, verbose_interval=10)

where n_components is number of distributions and covariance_type is covariance structure type.

Covariance structures are of four types:

Full Type: In this type of covariance, individual components take up a position and shape. This type is expected to have the best performance however, it is prone to overfitting on small data sets.
Tied Type: The components have the same shape and structure.
Diagonal Type: In this type, the axis of the plot is oriented along the co-ordinates.
Spherical Type: This type is similar to diagonal type with spherical probability distribution.

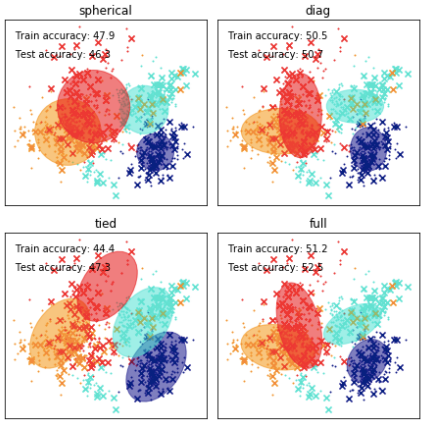


Figure 5: The covariance types on digits data

Approaches:

1) **Akaike Information Criterion**
The Akaike information criterion (AIC) is model selection approach formulated by the statistician Hirotugu Akaike that compares the quality of the statistical models.

AIC is given by the formula:

$$AIC = 2k - \ln \hat{L}$$

where k is the number of parameters and L is the maximum log likelihood

AIC suggests a quadratic model of order 2. Here log-likelihood is a measure of model fit i.e. higher the number, better the fit. AIC is an estimate of a constant plus the relative distance between the unknown true likelihood function of the data and the fitted likelihood function of the model, so that a lower AIC means a model is considered to be closer to the truth

AIC in Scikit Learn:
The class sklearn.mixture.GaussianMixture in Scikit Learn has a method to determine the AIC of an input data X **aic(self,X)** where X is the input of type array of shape(n_samples,n_dimensions)

2) **Bayesian Information Criterion**
Bayesian Information Criterion (BIC) is another model selection approach in which is closely related to the AIC criterion. BIC is an estimate of a function of the posterior probability of a model being true, under a certain Bayesian setup, so that a lower BIC means that a model is considered to be more likely to be the true model.It is given by the formula:

$$BIC = k \ln(n) - 2 \ln \hat{L}$$

where n is the number of training points, k is the number of parameters and Lis the maximum likelihood.

BIC in Scikit Learn:
The class sklearn.mixture.GaussianMixture in Scikit Learn has a method to determine the BIC of an input data X bic(self,X) where X is the input of type array of shape(n_samples,n_dimensions)

3) **Silhouette Score**
Silhouette analysis is used to find the relative distance between the resulting clusters. It is a measure of how close each data point in one cluster to the data points in the neighboring clusters. The score lies between the range [-1,1], closer the value to the +1, farther the sample from the neighboring clusters. In Gaussian Mixture Models, the silhouette score is calculated for each number of components for each of the covariance type. The higher the score, more optimal the parameter value. Silhouette score is calculated using the mean intra cluster distance and mean inter cluster distance.
It is given by the formula:

$$\boldsymbol{Silhouette\ Score} = \frac{\boldsymbol{b - a}}{\mathbf{max}\ (\boldsymbol{a, b})}$$

where a is the mean intra-cluster distance and b is the distance between a sample to the the nearest cluster that the sample is not a part of.

Silhouette Score in Scikit Learn:

Sckit Learn has an inbuilt function that returns the silhouette score for each input data set
sklearn.metrics.silhouette_score(*X*, *labels*, *metric='euclidean'*, *sample_size=None*, *random_state=None*, ***kwds*)
where X is the input data set, labels are the predicted output and metric refers to the measure of the distance used

4) **Cross Validation:**
Cross Validation can also be used to find the optimal parameter based on the Average Log-Likelihood Score. In this case study, KFold cross validation is used. The input data

is split into train set and test set on 3 folds. The number of clusters is plotted against the average log likelihood score for each covariance type.

Results

Input Data Size:

The digits data loaded of the size (1797,64). The data is reduced to lower dimensions by projecting the data onto first 2 principal components. This reduced data id of the size (1797,2).

Parameters:
The first parameter is the number of components or clusters identified as n_components in the Gaussian Mixture model. The second parameter is the Covariance Type referred as covariance_type. The n_components are assigned with a range (2,7) and the covariance_type is assigned ['full','spherical','diag','tied']. The most optimal parameters are chosen based on the different approaches.

AIC Results:

The below bar graph shows the most optimal number of clusters and the covariance type. The lower the AIC score more optimal the parameter value.

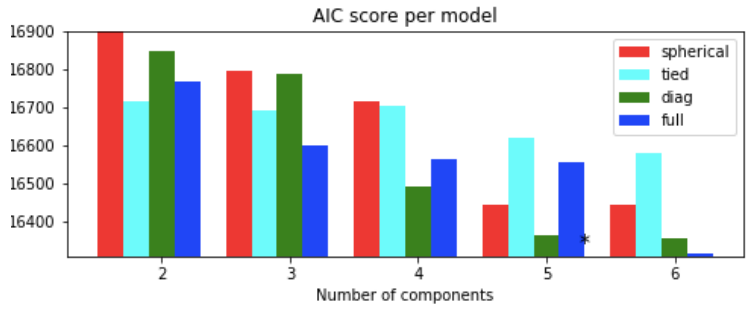


Figure 6: The AIC Score

Based on the analysis, the lowest AIC Score value is generated for 5 components or 6 component and 'Full' Covariance Type.

BIC Results:

The BIC implementation is similar to AIC. The bar graph shows the results for each parameter value. Like AIC, lower the BIC score more optimal the parameter value.

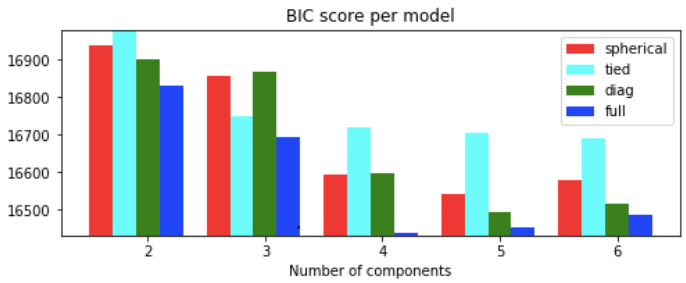


Figure 7: The BIC Score

The lowest score is generated for 4 or 5 components of covariance type 'Full'.

Silhouette Analysis Results:
The silhouette score is generated separately for each of the covariance type. Based on the analysis of the grap

1) Silhouette scores for Full Covariance Type:

```
For n_clusters in Full Covarianace Type = 2 The average silhouette_score is: 0.38
For n_clusters in Full Covarianace Type = 3 The average silhouette_score is: 0.38
For n_clusters in Full Covarianace Type = 4 The average silhouette_score is: 0.37
For n_clusters in Full Covarianace Type = 5 The average silhouette_score is: 0.36
For n_clusters in Full Covarianace Type = 6 The average silhouette_score is: 0.38
```

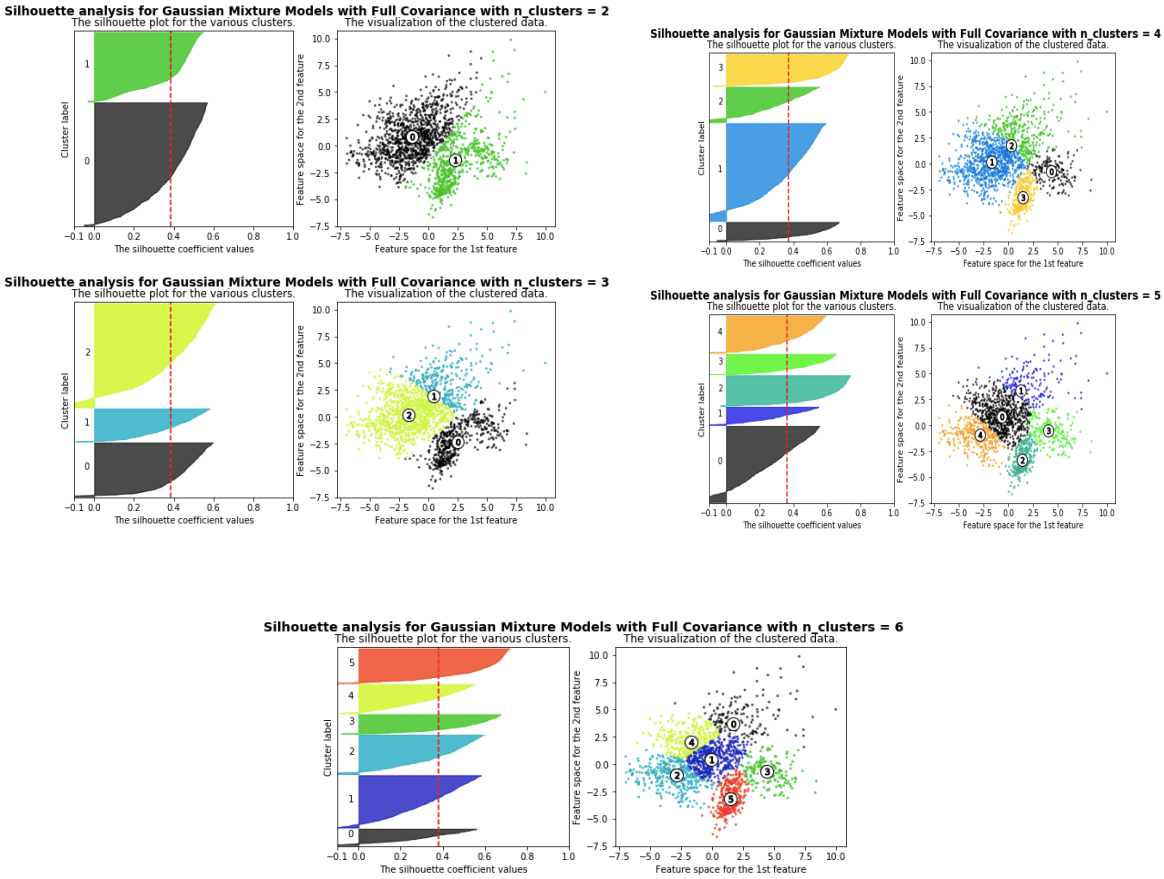


Figure 8: Silhouette Score for Full Covariance Type

2) Silhouette scores for Tied Covariance Type:

```
For n_clusters in Tied Covarianace Type = 2 The average silhouette_score is: 0.38
For n_clusters in Tied Covarianace Type = 3 The average silhouette_score is: 0.39
For n_clusters in Tied Covarianace Type = 4 The average silhouette_score is: 0.37
For n_clusters in Tied Covarianace Type = 5 The average silhouette_score is: 0.37
For n_clusters in Tied Covarianace Type = 6 The average silhouette_score is: 0.38
```

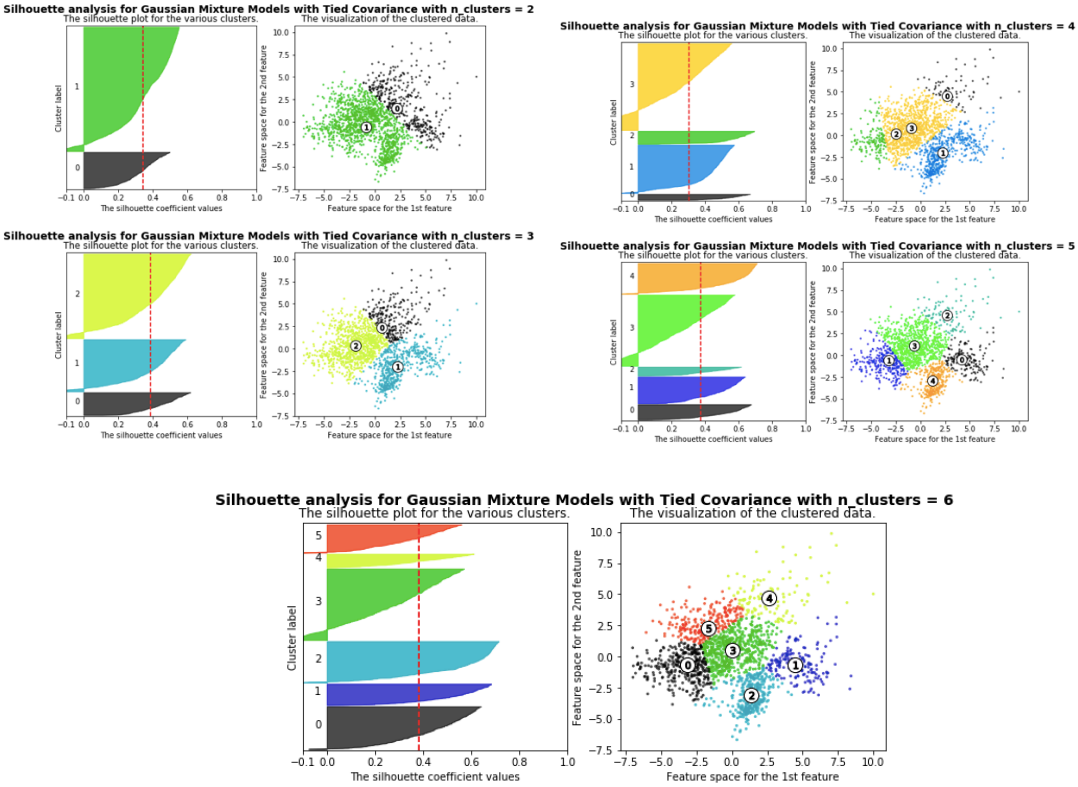
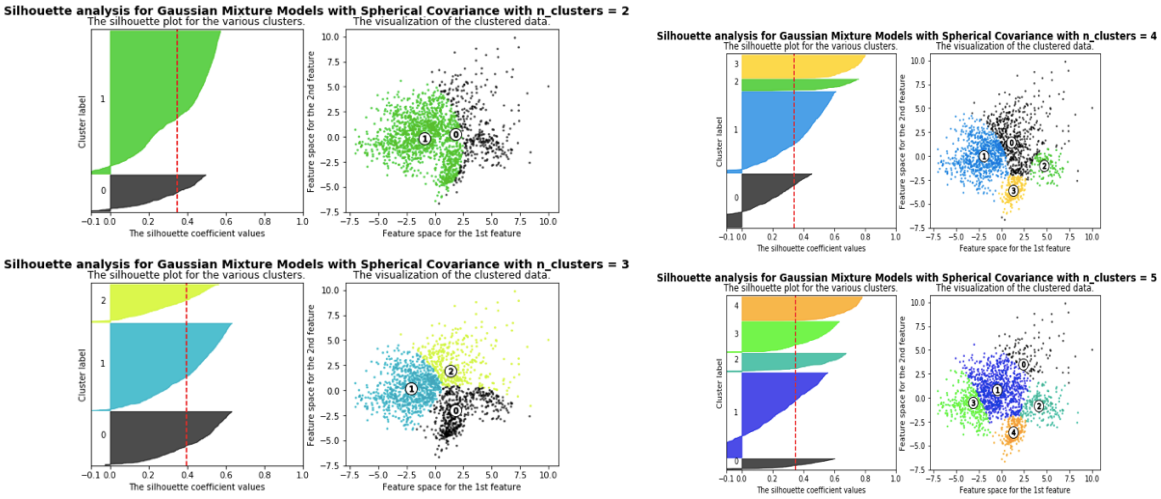


Figure 9: Silhouette Score for Tied Covariance Type

3) Silhouette scores for Spherical Covariance Type:

```
For n_clusters in Spherical Covarianace Type = 2 The average silhouette_score is: 0.36
For n_clusters in Spherical Covarianace Type = 3 The average silhouette_score is: 0.4
For n_clusters in Spherical Covarianace Type = 4 The average silhouette_score is: 0.35
For n_clusters in Spherical Covarianace Type = 5 The average silhouette_score is: 0.36
For n_clusters in Spherical Covarianace Type = 6 The average silhouette_score is: 0.29
```



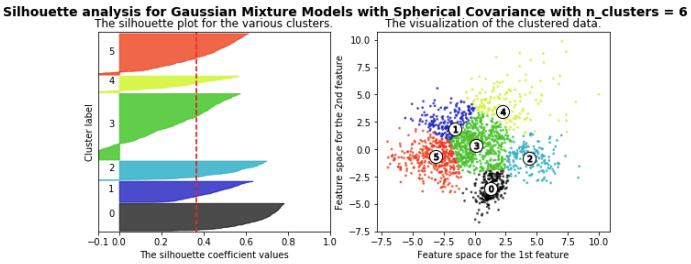


Figure 10: Silhouette Score for Spherical Covariance Type

4) Silhouette scores for Diagonal Covariance Type:

```
For n_clusters in Diagonal Covarianace Type = 2 The average silhouette_score is: 0.35
For n_clusters in Diagonal Covarianace Type = 3 The average silhouette_score is: 0.39
For n_clusters in Diagonal Covarianace Type = 4 The average silhouette_score is: 0.37
For n_clusters in Diagonal Covarianace Type = 5 The average silhouette_score is: 0.37
For n_clusters in Diagonal Covarianace Type = 6 The average silhouette_score is: 0.37
```

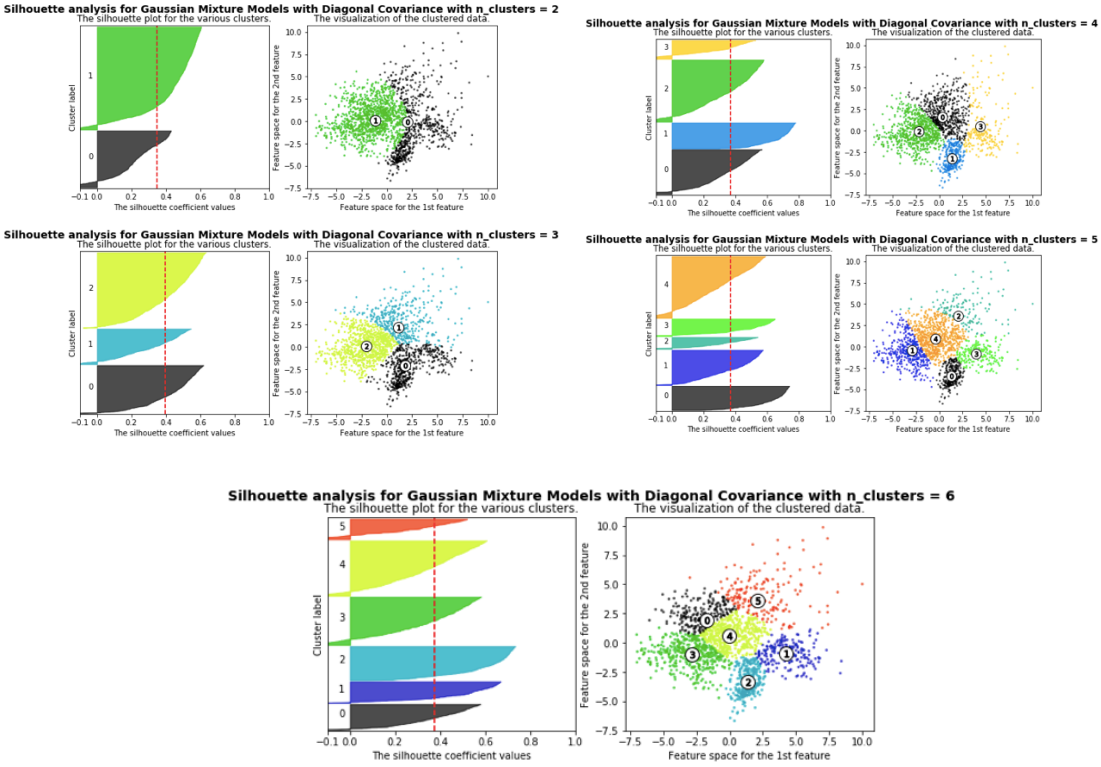


Figure 11: Silhouette Score for Diagonal Covariance Type

Cross Validation Results:

From the below plot we can infer that the Full Covariance wit around 5 or 6 clusters selects the best model.

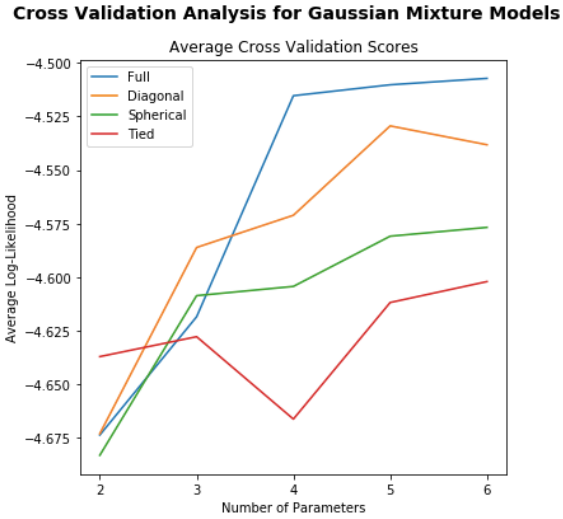


Figure 12: Cross Validation Results

Discussion

From analysis of plots of different approaches, it is observed that the AIC, BIC and Cross Validation produce almost similar results. The Silhouette Analysis produced close results for all the parameters and therefore there is no clear distinction to determine which parameters to choose.

Summary of the results:

Strategy	Optimal number of clusters	Optimal Covariance Type
AIC	5 or 6	Full
BIC	4 or 5	Full
Silhouette Analysis	3 (scores are close)	Spherical (scores are close)
Cross Validation	5 or 6	Full

From the above summary we can conclude that, the most optimal Gaussian Mixture Model can be obtained when the number of clusters is **5** and the covariance type is **'Full'.**

References:

1) https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture.aic

2) https://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_selection.html

3) https://scikit-learn.org/0.15/auto_examples/mixture/plot_gmm_classifier.html

4) https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

5) https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

6) https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

7) https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

8) http://ethen8181.github.io/machine-learning/clustering/GMM/GMM.html#Implementing-the-EM-algorithm

9) https://github.com/scikit-learn/scikit-learn/issues/10863

10) https://www.geeksforgeeks.org/gaussian-mixture-model/

11) https://en.wikipedia.org/wiki/Akaike_information_criterion

12) https://en.wikipedia.org/wiki/Bayesian_information_criterion

13) https://www.methodology.psu.edu/resources/AIC-vs-BIC/