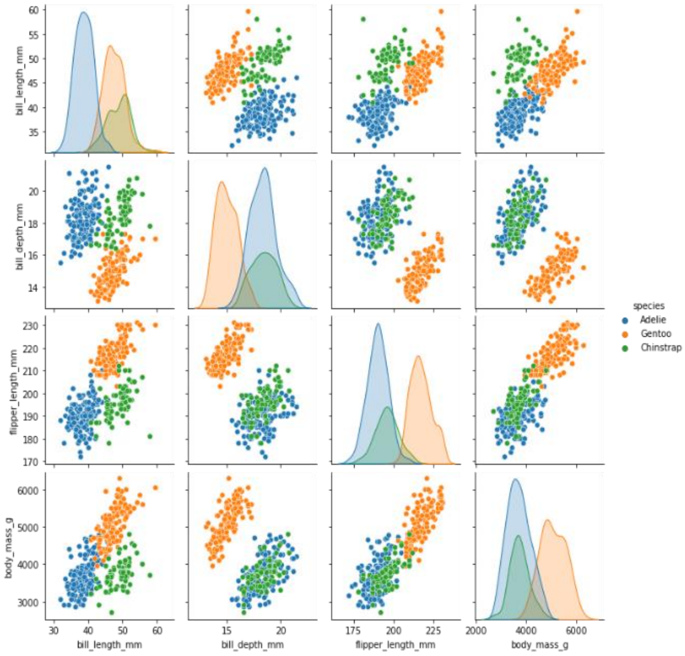
Assignment 3

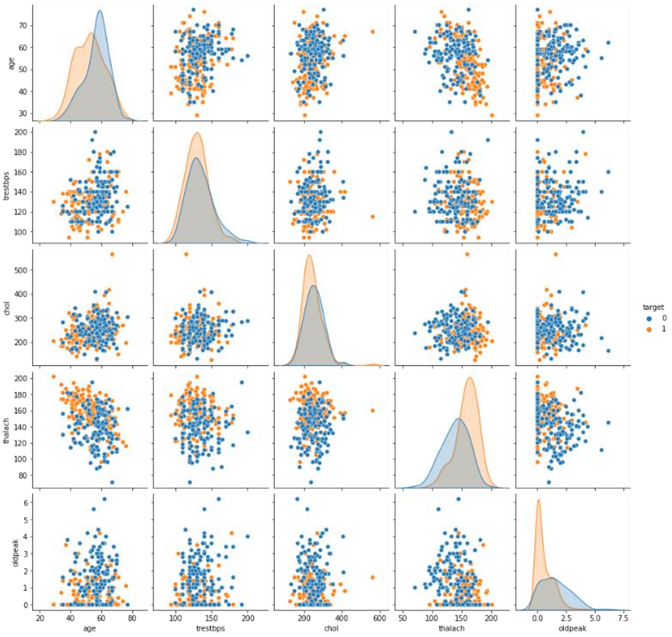
1. Datasets

In this assignment, I am using the Heart Dataset which I used in assignment 1. The Heart dataset has 2 classes (Heart Disease and Not Heart Disease). The second dataset I am using is the Penguins dataset, the Penguins dataset is kind of similar to the Iris dataset which contains features such as penguins body mass, flipper length… etc. The Penguins dataset has 3 classes or species (Adelie, Gentoo and Chinstrap). The Penguins dataset has missing values and categorial features, so it needed preprocessing before using it. The below graphs show the distribution (marginal and joint) of the features for each dataset as clusters with respect to their classes. We will need these graphs later in this report to compare them with the results we got from the clustering and dimensionality reduction algorithms.

Penguins Dataset



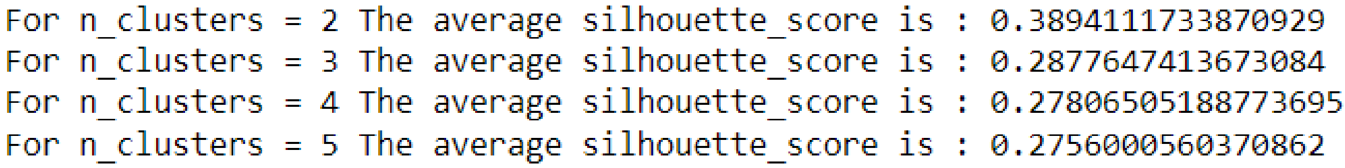
Heart Dataset



2. Clustering Algorithms

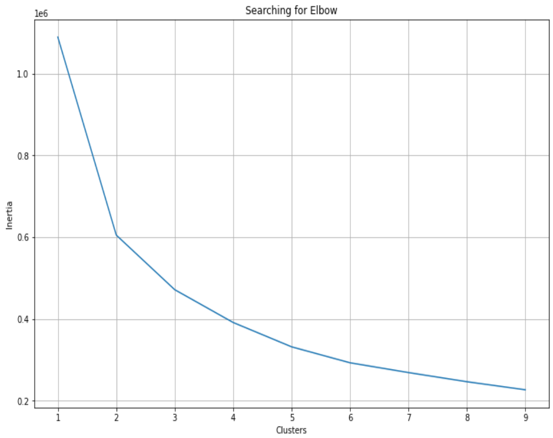
2.1.1 K-Means Clustering (Heart Dataset)

After dropping the target variable from the Heart Dataset, I used the Elbow method to figure out the optimal number of clusters for the dataset in unsupervised way. The Elbow method graph is shown below, which is pointing towards n_clusters = 2 but it isn't clear enough for me, so I decided to use the Silhouette score as another metric to confirm the n_clusters to be used. I chose range of 2 to 5 clusters. The Silhouette score also pointed to n_clusters = 2 with best score as shown below. I didn't include all the Silhouette scores graphs for the rest of the cluster numbers due to the limited space of this report but screenshot of the scores can be seen below.

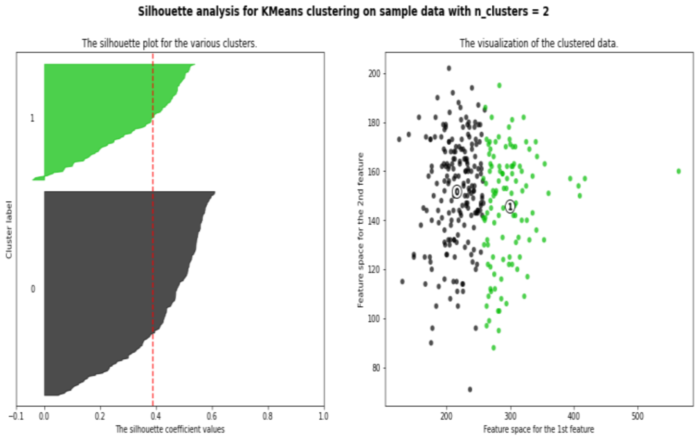


```
For n_clusters = 2 The average silhouette_score is : 0.3894111733870929
For n_clusters = 3 The average silhouette_score is : 0.2877647413673084
For n_clusters = 4 The average silhouette_score is : 0.27806505188773695
For n_clusters = 5 The average silhouette_score is : 0.2756000560370862
```

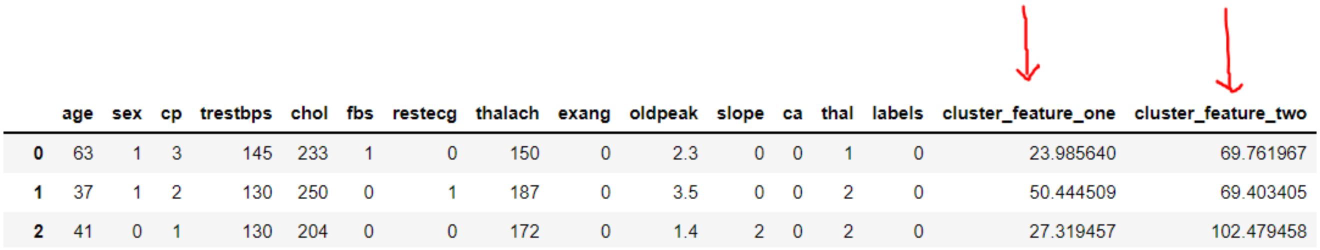
Elbow Method



Silhouette score

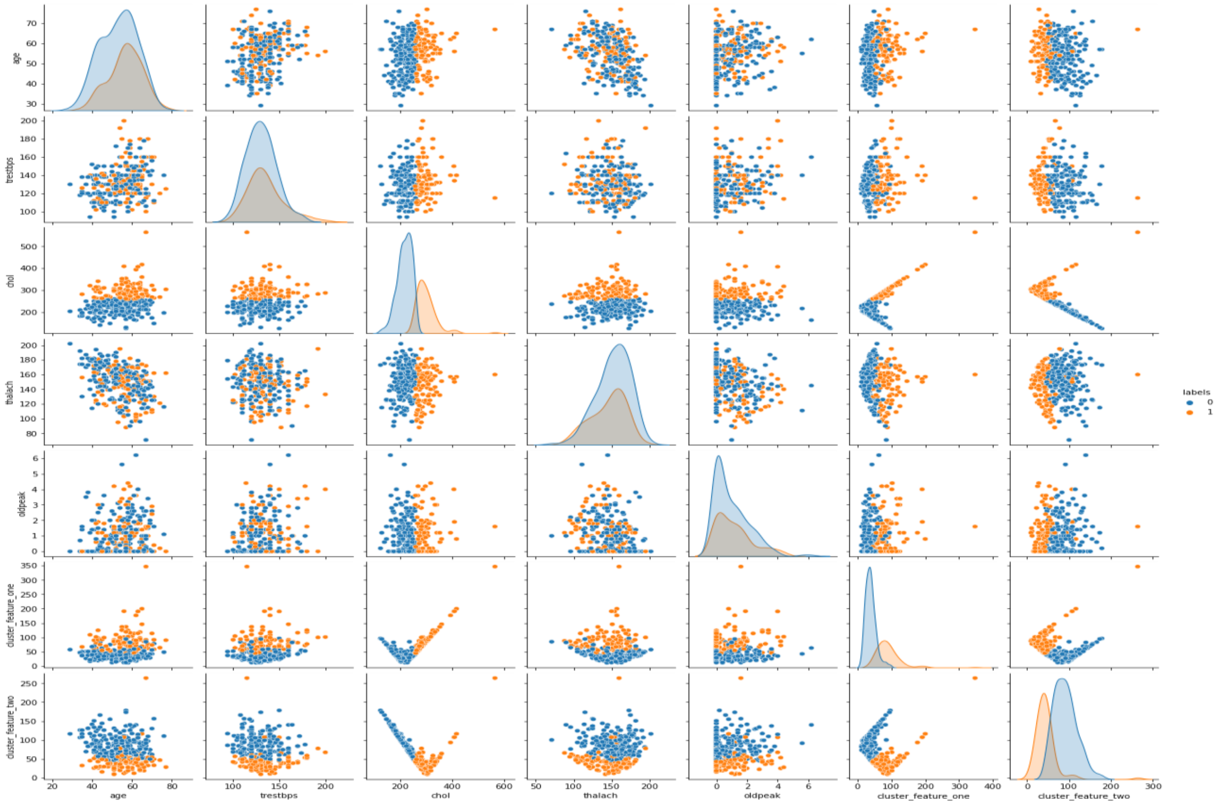


Now I am confident to use n_clusters = 2 as a parameter in KMeans clustering algorithm. By tuning the hyperparameters of the algorithm and using max_iter = 600, I was able to get new features from the KMeans clustering algorithm which are the **samples distances from the clusters centers** as shown below. These new features will play a very important role in classifying samples.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	labels	cluster_feature_one	cluster_feature_two
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	0	23.985640	69.761967
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	0	50.444509	69.403405
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	0	27.319457	102.479458

Using these new features and the generated cluster labels improved the features distributions in the clusters as shown below



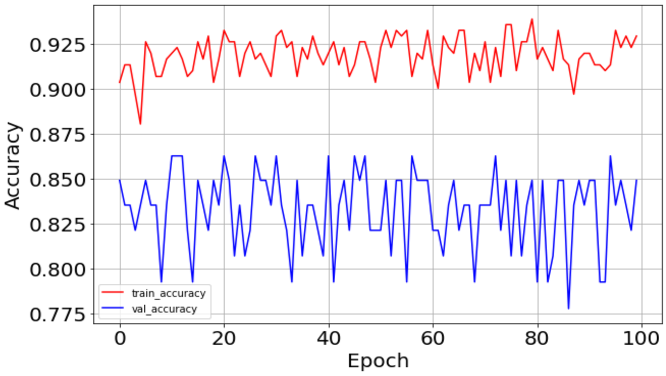
By comparing this features space distribution with the original distribution in section 1, we can confirm that using KMeans clustering features improved the feature distribution with respect to their labels.

Passing the new Heart dataset with the clustering features to the Neural Network model from Assignment 1 (Part 5 of Assignment 3)

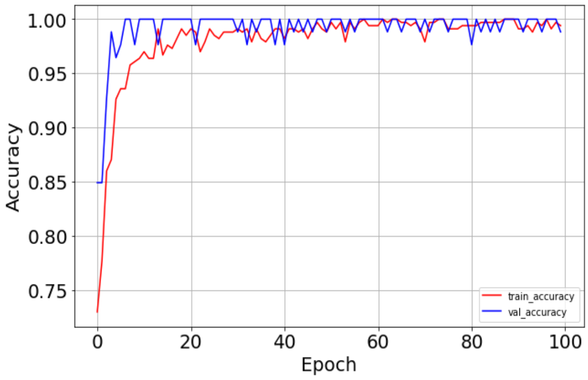
Adding the KMeans clustering features to the Heart dataset not only improved the features distributions but also improved the Neural Network test accuracy score from 85% to 97%. The learning curves are shown below for both, the original Heart dataset and the Heart dataset with KMeans clustering features added to it.

The KMeans clustering features are very powerful because it measures sample distance from the clusters centers and classify it according to the lower distance which make it achieve high accuracy scores.

Original Heart Dataset from Assignment 1

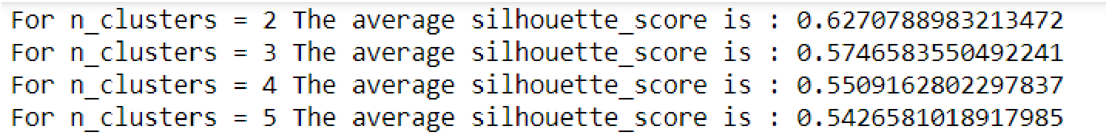


Heart Dataset with KMeans Clustering Features



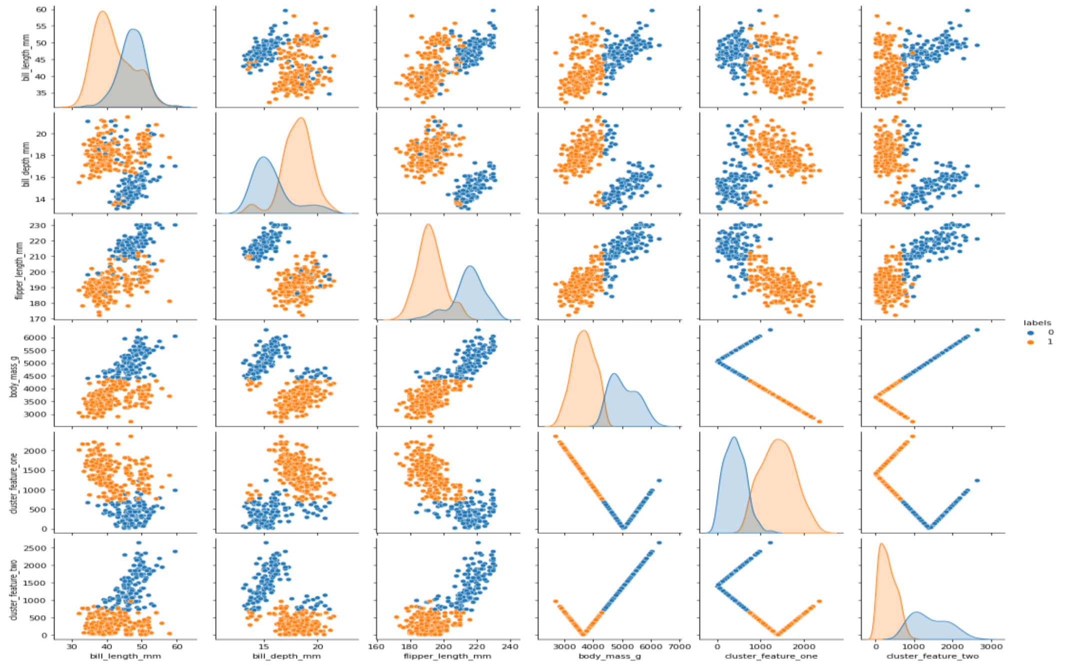
2.1.2 K-Means Clustering (Penguins Dataset)

I followed a similar approach as I did with the Heart dataset, so I am going to share below the Silhouette scores that made me decide the number of clusters I chose



```
For n_clusters = 2 The average silhouette_score is : 0.6270788983213472
For n_clusters = 3 The average silhouette_score is : 0.5746583550492241
For n_clusters = 4 The average silhouette_score is : 0.5509162802297837
For n_clusters = 5 The average silhouette_score is : 0.5426581018917985
```

n_cluster = 2 has the highest score. However, I know that the Penguins dataset has **3 classes**. **That can be explained if we go back to the original pairplot for Penguins dataset in section 1, we can see that the Chainstrap class share close/similar features with the Adelie class which can be interpreted by the KMeans clustering algorithm as one cluster because KMeans uses the distance between points as a similarity metric**. Below are the results obtained from using KMeans with 2 clusters on the Penguins dataset

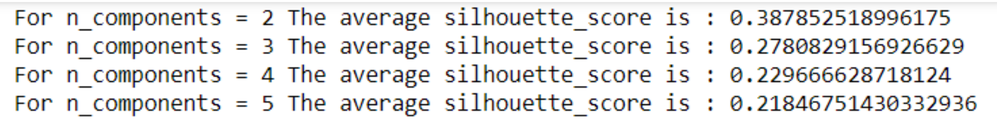


We can observe how the KMeans cluster features projected the data in different way that was originally distributed. The above plot suggests now we have 2 species, Adelie and Gentoo. The Chainstrap species are combined with the Adelie species in one cluster.

2.2.1 Expectation-Maximization (Heart Dataset)

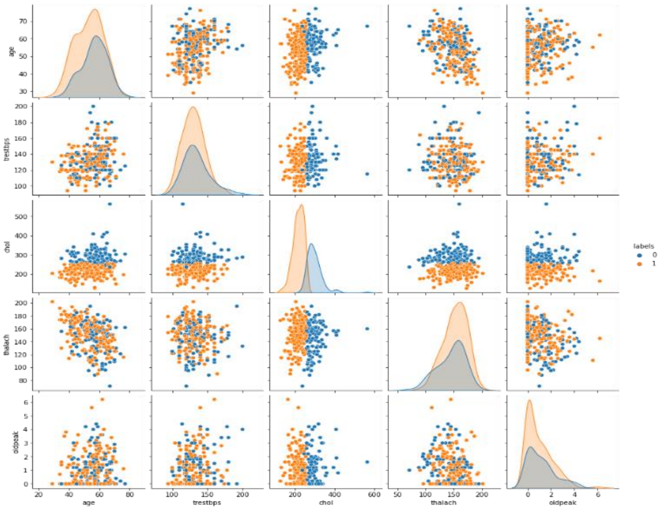
I used the Gaussian Mixture Model for the Expectation Maximization clustering. The Gaussian model would assume we have a particular number of Gaussian distributions and start calculating the probability that certain datapoints belong to a certain distribution together. The mean and variance for these Gaussian distributions are determined using Expectation-Maximization.

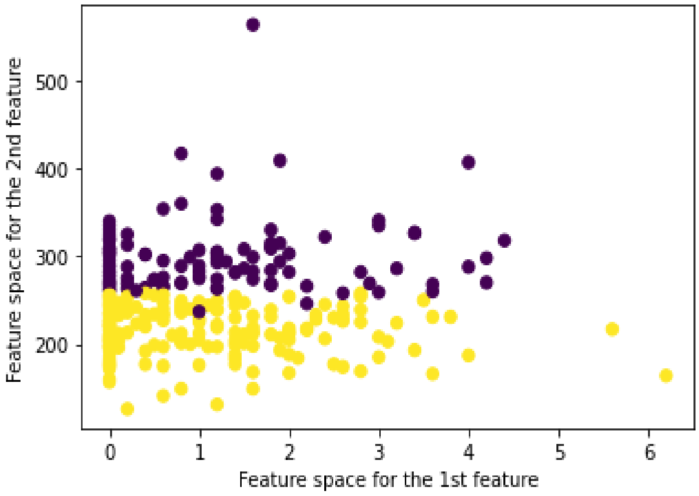
I used the Elbow method and the Silhouette score to determine the number of components to be used in the model which determine the number of clusters. Below are the Silhouette scores from using the Gaussian model on the Heart dataset.



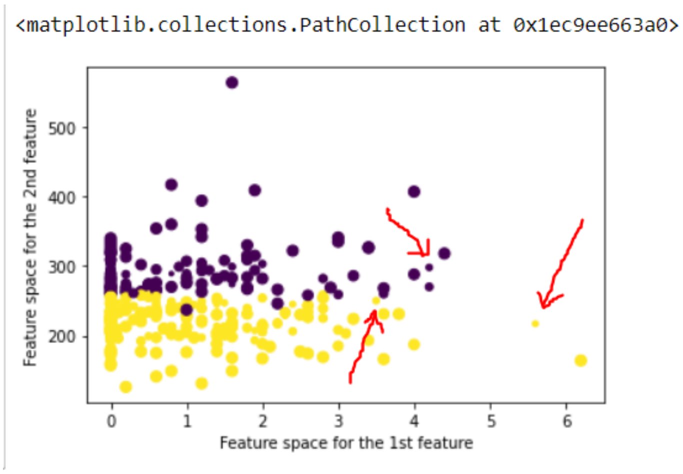
```
For n_components = 2 The average silhouette_score is : 0.387852518996175
For n_components = 3 The average silhouette_score is : 0.2780829156926629
For n_components = 4 The average silhouette_score is : 0.229666628718124
For n_components = 5 The average silhouette_score is : 0.21846751430332936
```

Using n_components = 2 gave us the below results for the original features distributions

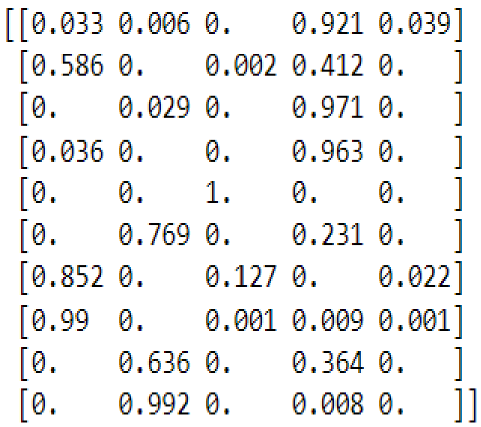




To create visualization for the probability of a certain datapoint belongs to a certain distribution, the below plot created which shows the datapoint with a smaller site represent lower probability that it belongs to this distribution, and next to it are probability scores for the first 10 samples of the dataset

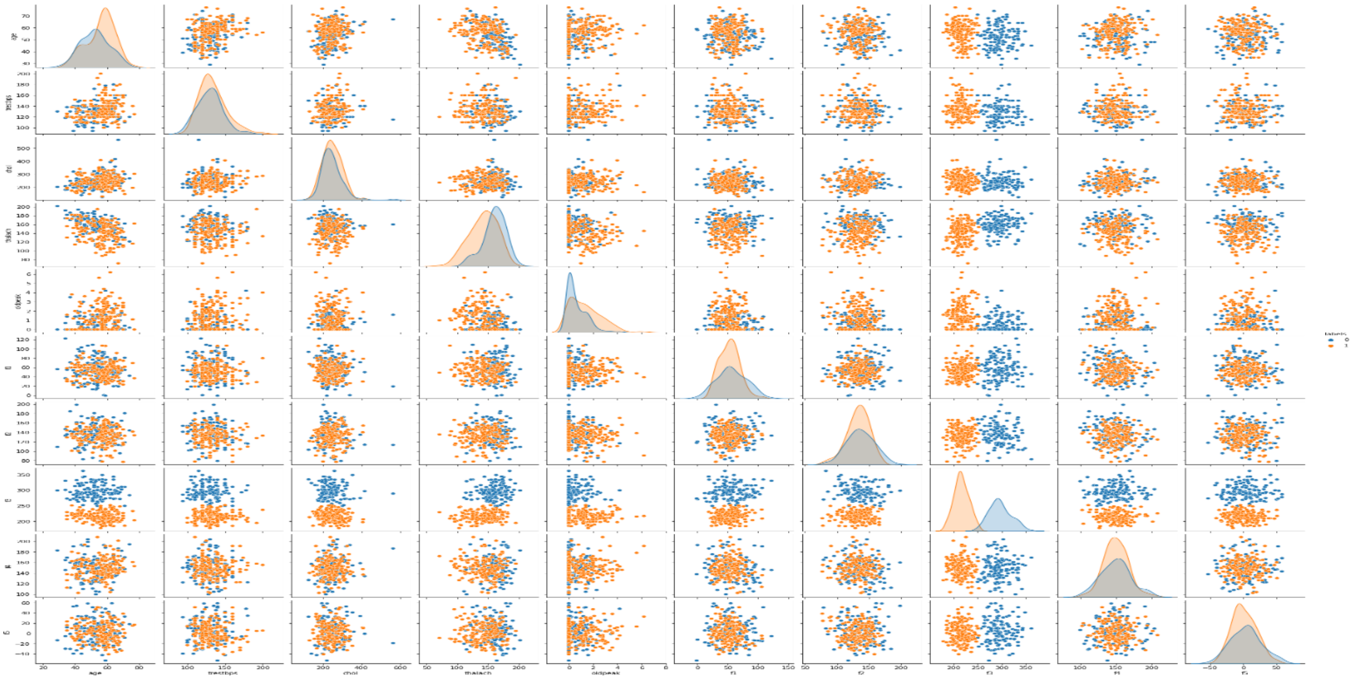


```
<matplotlib.collections.PathCollection at 0x1ec9ee663a0>
```



```
[[0.033 0.006 0.    0.921 0.039]
 [0.586 0.    0.002 0.412 0.   ]
 [0.    0.029 0.    0.971 0.   ]
 [0.036 0.    0.    0.963 0.   ]
 [0.    0.    1.    0.    0.   ]
 [0.    0.769 0.    0.231 0.   ]
 [0.852 0.    0.127 0.    0.022]
 [0.99  0.    0.001 0.009 0.001]
 [0.    0.636 0.    0.364 0.   ]
 [0.    0.992 0.    0.008 0.   ]]
```

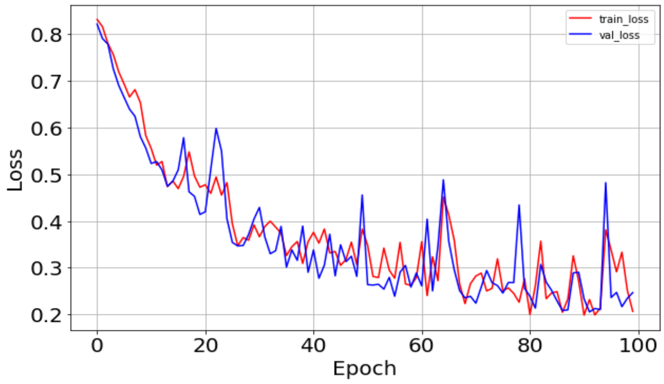
Adding the 5 new clustering features generated by the Gaussian model makes the features distribution looks like as shown below



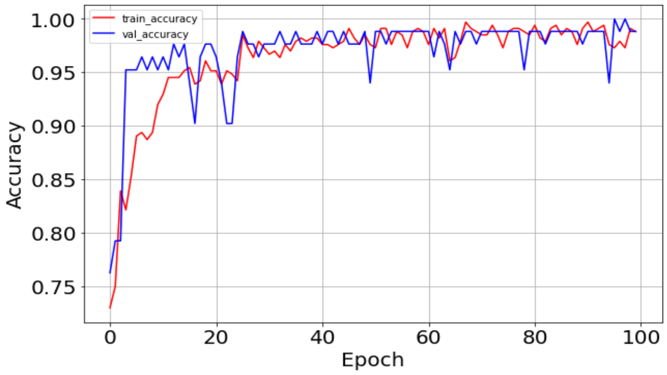
Passing the new Heart dataset with the clustering features to the Neural Network model from Assignment 1 (Part 5 of Assignment 3)

The Neural Network test accuracy score improved from to 95%. The learning curves are shown below for both and can be compared to the results from assignment 1 in section 2.1.1.

Original Heart Dataset from Assignment 1

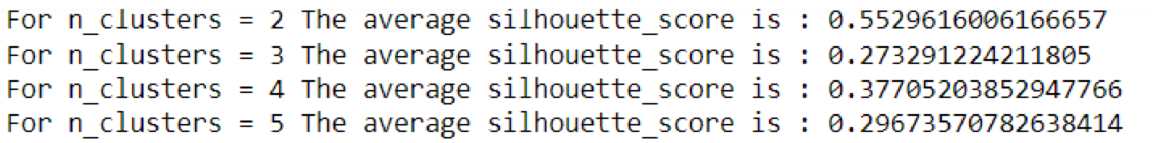


Heart Dataset with KMeans Clustering Features

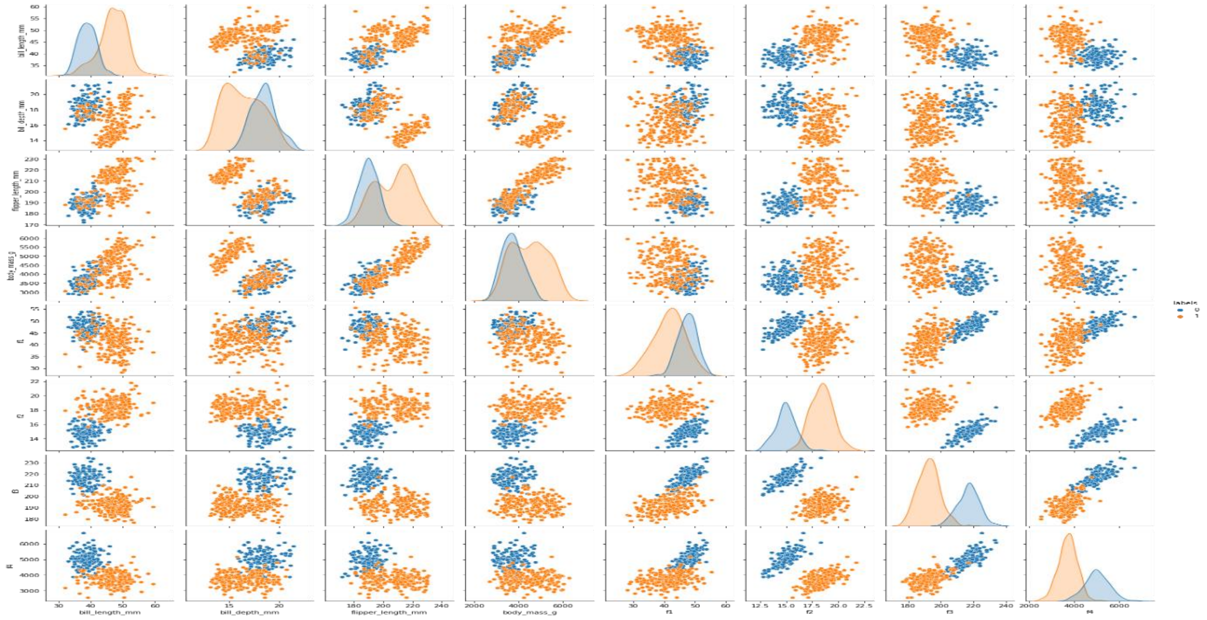


2.2.2 Expectation-Maximization (Penguins Dataset)

I repeated the same steps I did for the Heart dataset on the Penguins dataset but nothing interesting to report. The Gaussian model gave us high Silhouette scores for the 2 clusters as shown below as well as the features distribution after adding the new features generated by the Gaussian model.



```
For n_clusters = 2 The average silhouette_score is : 0.5529616006166657
For n_clusters = 3 The average silhouette_score is : 0.273291224211805
For n_clusters = 4 The average silhouette_score is : 0.37705203852947766
For n_clusters = 5 The average silhouette_score is : 0.29673570782638414
```



3. Dimensionality Reduction Algorithms

The below table to compare between different dimensionality reduction algorithms and all the plots used in these algorithms are shown after the table.

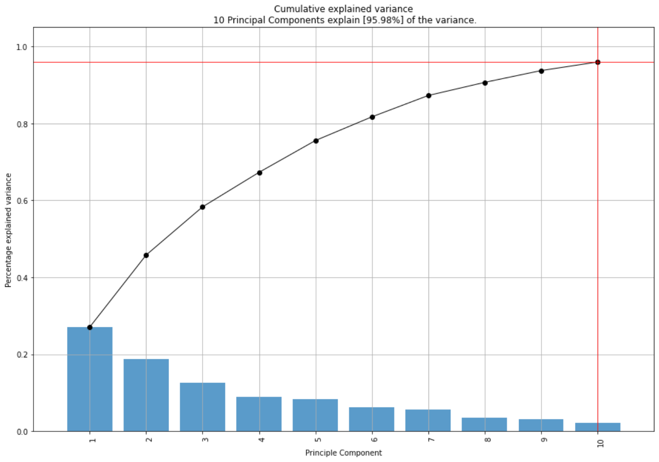
Algorithm	Method for choosing # of components	Number of components used	Number of clusters by using KMeans based on Silhouette scores	Number of clusters by using Expectation Maximization	Neural Network test Accuracy (Heart Dataset)
PCA	95% explained variance (Eigen values approach)	• 10 (Heart Dataset) • 3 (Penguins Dataset	• 2 clusters (Heart Dataset) • 2 clusters (Penguins Dataset)	• 3 clusters (Heart Dataset) • 2 clusters (Penguins Dataset)	• 99% using KMeans labels • 71% using Gaussian model labels
ICA	Maximum mean absolute Kurtosis	• 3 (Heart Dataset) • 4 (Penguins Dataset)	• 4 clusters (Heart Dataset) • 3 clusters (Penguins Dataset)	• 3 clusters (Heart Dataset) • 3 clusters (Penguins Dataset)	• 35% using KMeans labels • 67% using Gaussian model labels
RCA	Squared Distance Rate: Projected / Original (Reconstruction Error)	• 9 (Heart Dataset) • 3 (Penguins Dataset)	• 2 clusters (Heart Dataset) • 2 clusters (Penguins Dataset)	• 2 clusters (Heart Dataset) • 2 clusters (Penguins Dataset)	• 87% using KMeans labels • 98% using

					Gaussian model labels
Linear Discriminant Analysis (LDA)	Number of Classes – 1 (Supervised method)	• 1 (Heart Dataset) • 2 (Penguins Dataset)	• 2 clusters (Heart Dataset) • 3 clusters (Penguins Dataset)	• 2 clusters (Heart Dataset) • 3 clusters (Penguins Dataset)	• 99% using KMeans labels • 97% using Gaussian model labels

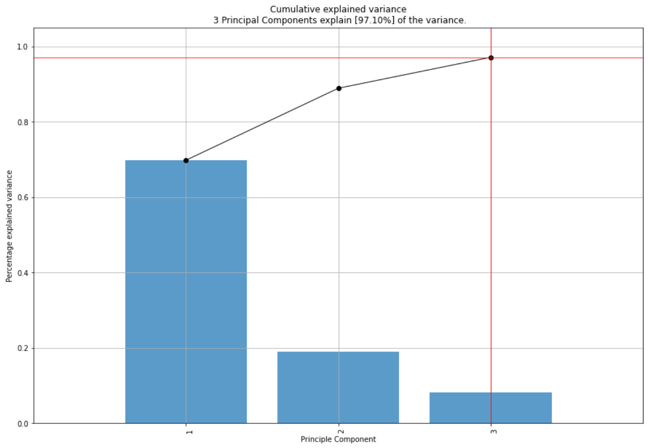
3.1 PCA

For the Heart Dataset, I used 10 components to explain 95% of the variance while 3 components were used to explain 95% of the variance in the Penguins Dataset. The bi-plot also gives us important information about which features are important for certain components. The below plots explains it all.

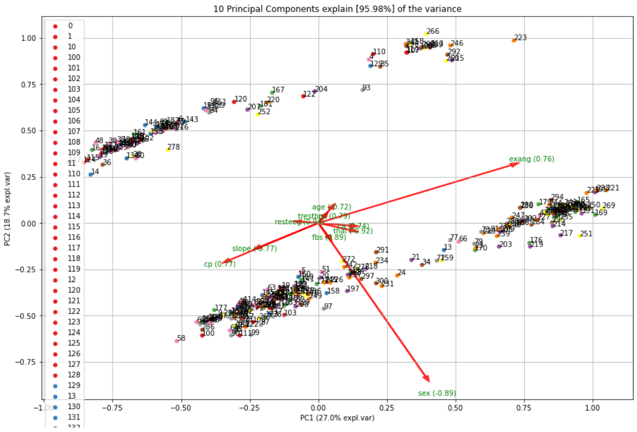
Cumulative Explained Variance (Heart Dataset)



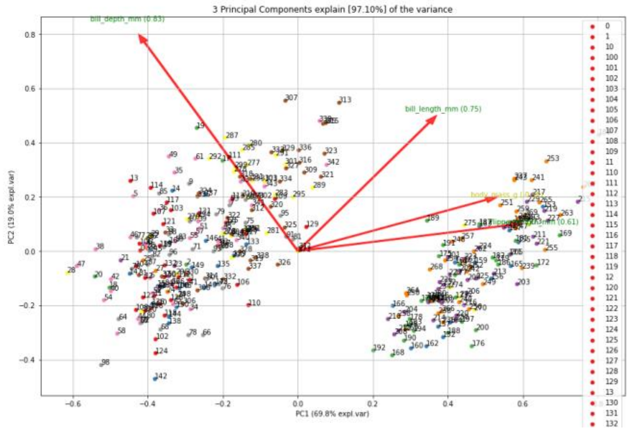
Cumulative Explained Variance (Penguins Dataset)



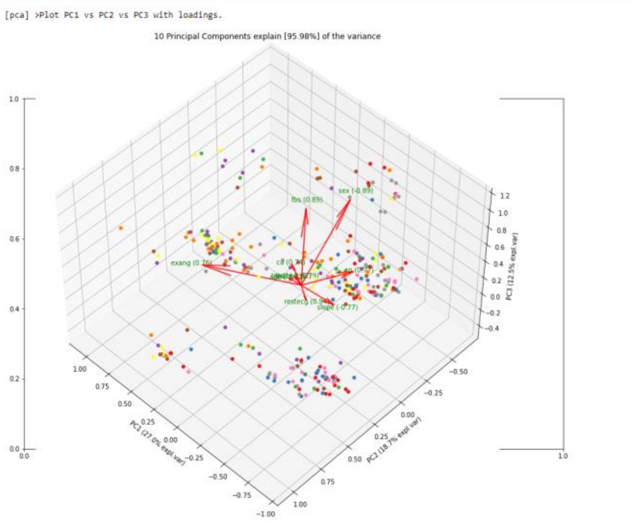
Bi_Plot (Heart Dataset)



Bi_Plot (Penguins Dataset)

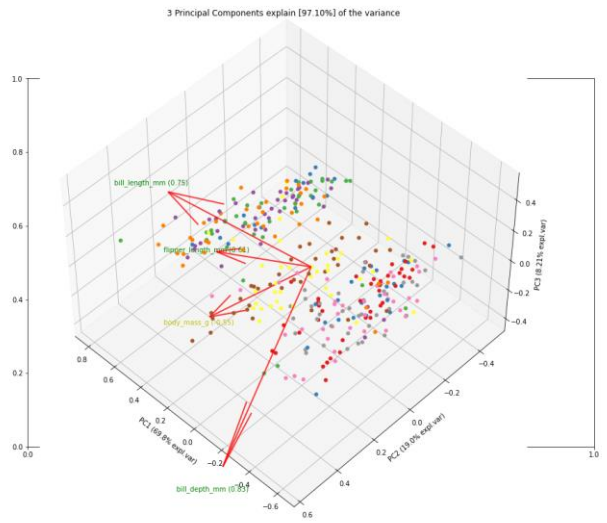


Bi_Plot_3D (Heart Dataset)

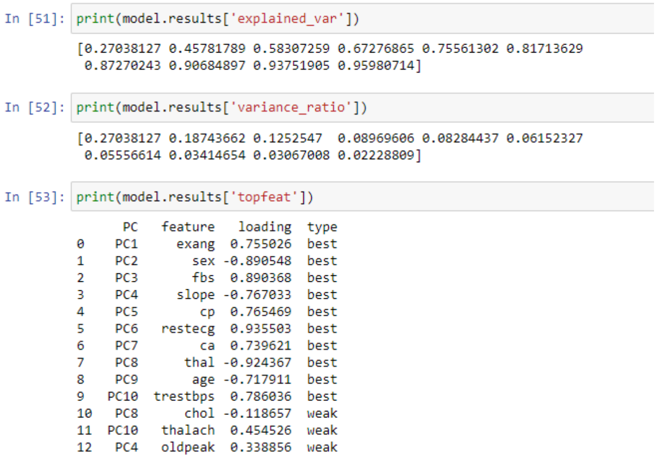


```
[pca] >Plot PC1 vs PC2 vs PC3 with loadings.
```

Bi_Plot_3D (Penguins Dataset)

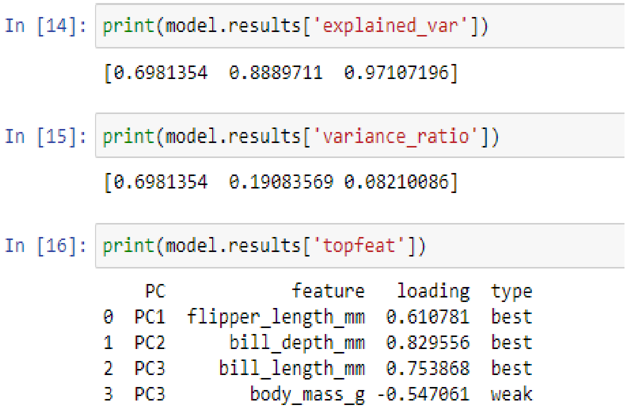


Eigen Values, Variance Ratio and Top Features (Heart Dataset)



```
In [51]: print(model.results['explained_var'])

         [0.27038127 0.45781789 0.58307259 0.67276865 0.75561302 0.81713629
          0.87270243 0.90684897 0.93751905 0.95980714]

In [52]: print(model.results['variance_ratio'])

         [0.27038127 0.18743662 0.1252547  0.08969606 0.08284437 0.06152327
          0.05556614 0.03414654 0.03067008 0.02228809]

In [53]: print(model.results['topfeat'])

              PC   feature   loading  type
         0    PC1     exang  0.755026  best
         1    PC2       sex -0.890548  best
         2    PC3       fbs  0.890368  best
         3    PC4     slope -0.767033  best
         4    PC5        cp  0.765469  best
         5    PC6   restecg  0.935503  best
         6    PC7        ca  0.739621  best
         7    PC8      thal -0.924367  best
         8    PC9       age -0.717911  best
         9   PC10  trestbps  0.786036  best
         10   PC8      chol -0.118657  weak
         11  PC10   thalach  0.454526  weak
         12   PC4   oldpeak  0.338856  weak
```

Eigen Values, Variance Ratio and Top Features (Penguins Dataset)

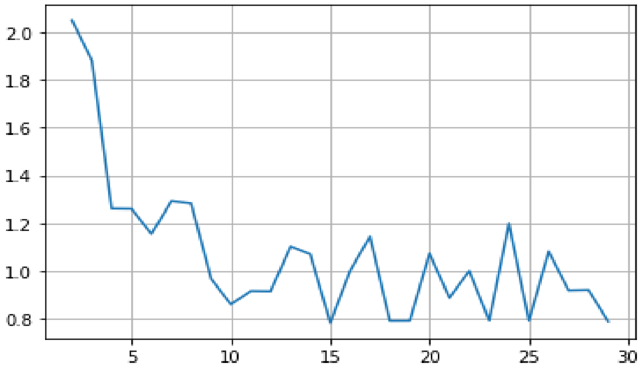


```
In [14]: print(model.results['explained_var'])

         [0.6981354  0.8889711  0.97107196]

In [15]: print(model.results['variance_ratio'])

         [0.6981354  0.19083569 0.08210086]

In [16]: print(model.results['topfeat'])

             PC            feature   loading  type
         0  PC1  flipper_length_mm  0.610781  best
         1  PC2      bill_depth_mm  0.829556  best
         2  PC3     bill_length_mm  0.753868  best
         3  PC3        body_mass_g -0.547061  weak
```

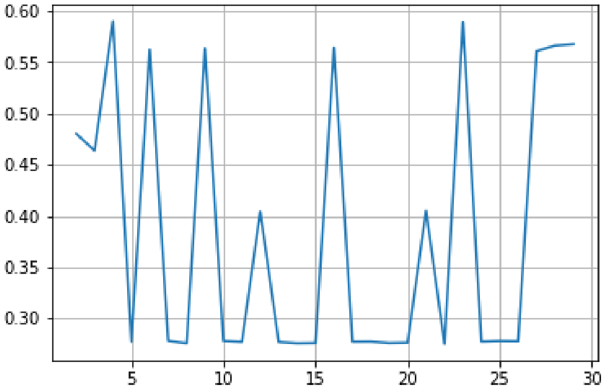
3.2 ICA

Absolute mean Kurtosis scores is used to determine the number of components to be used in ICA. The below plots show the number of components on the x-axis and the absolute mean Kurtosis score on the y-axis

Absolute Mean Kurtosis (Heart Dataset)



Absolute Mean Kurtosis (Penguins Dataset)

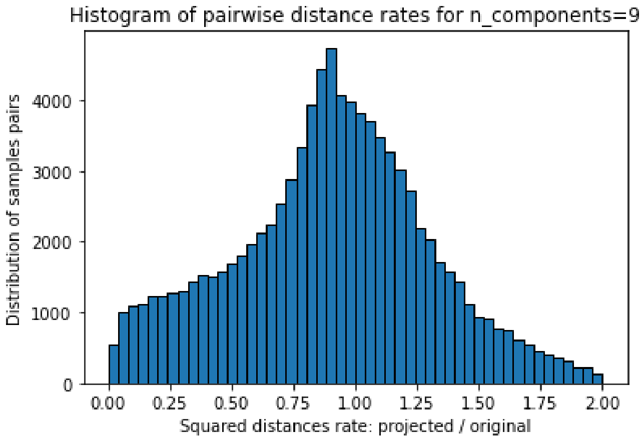


3.3 RCA

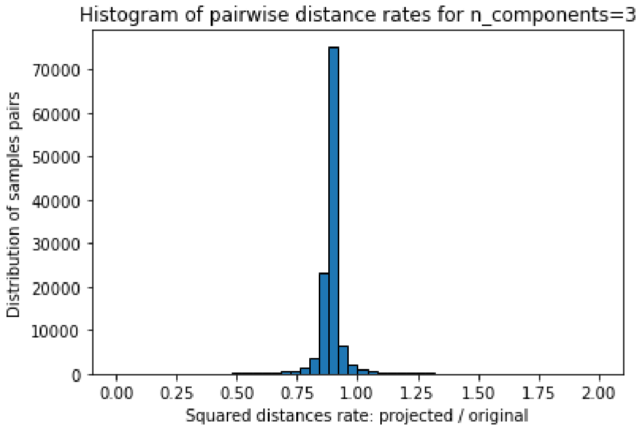
In RCA, the squared distance rate: projected/original, which is a metric for reconstruction error. I used this metric to choose the number of components to be used in the RCA model. Below is the plot of pairwise distance rates for

9 components: Mean = 0.9 and std = 0.4 for Heart dataset and 3 components: Mean = 0.89 and std = 0.07 for Penguins Dataset

Heart Dataset



Penguins Dataset



3.4 LDA

In LDA, I used a supervised approach in choosing the number of components, which is the number of classes – 1. Accordingly, I used 1 component for the Heart Dataset and 2 components for the Penguins Dataset. LDA uses Bayes' Theorem to estimate the probabilities. If the output class is (k) and the input is (x), here is how Bayes' theorem works to estimate the probability that the data belongs to each class.

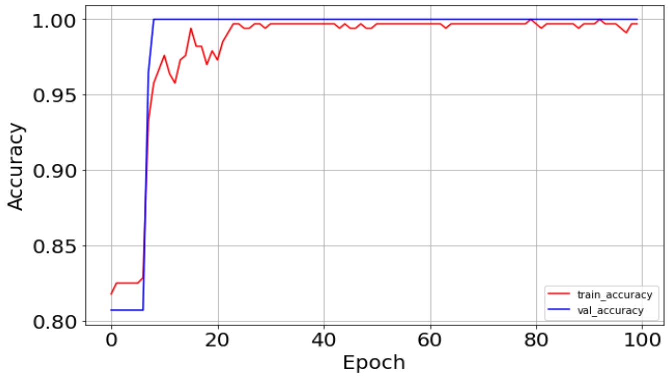
P(Y=x|X=x) = (PIk * fk(x)) / sum(PIl * fl(x)).

3.5 Learning Curves for the Neural Network after passing the dimensionality reduction algorithms to it

After I generated the features from the dimensionality reduction algorithms, I used the clustering algorithms (KMeans and EM) to generate clusters and labels for the features of each dimensionality reduction algorithms on the Heart Dataset. Then I split the output dataset into train and test datasets to use them on the Neural Network model and generated the below learning curves.

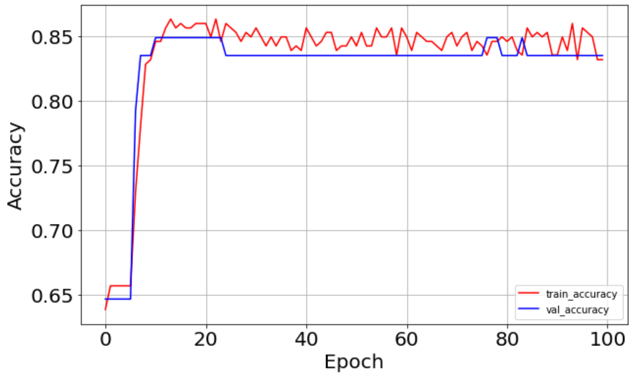
PCA (KMeans Clustering labels)

Test Accuracy = 99%



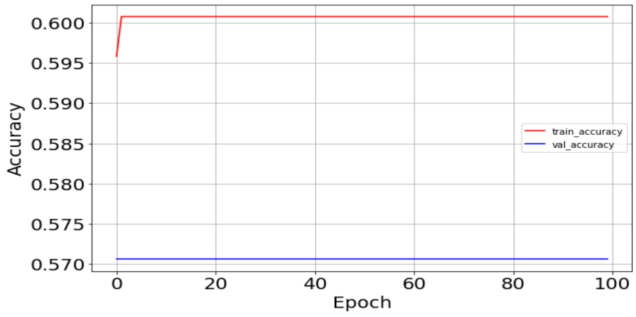
PCA (EM Clustering labels)

Test Accuracy = 71%



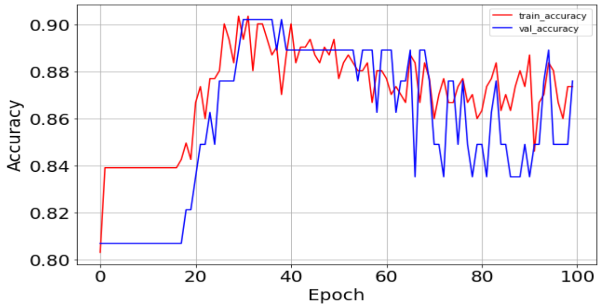
ICA (KMeans Clustering labels)

Test Accuracy = 35%



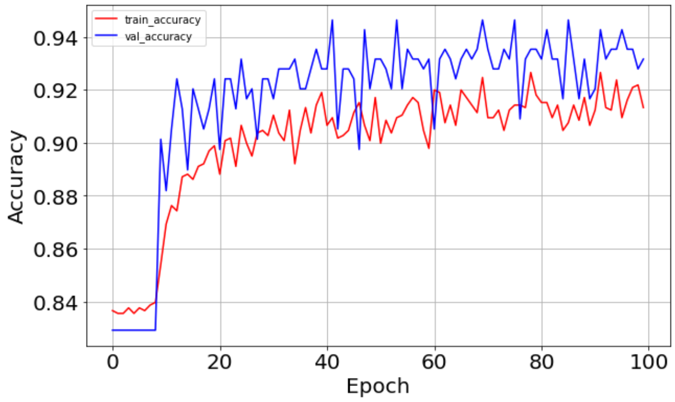
ICA (EM Clustering labels)

Test Accuracy = 67%



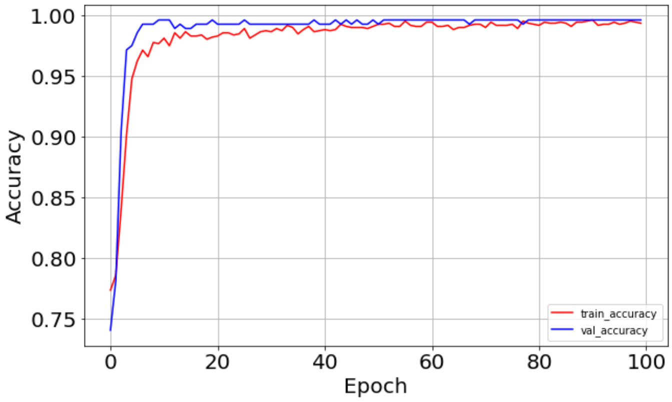
RCA (KMeans Clustering labels)

Test Accuracy = 87%



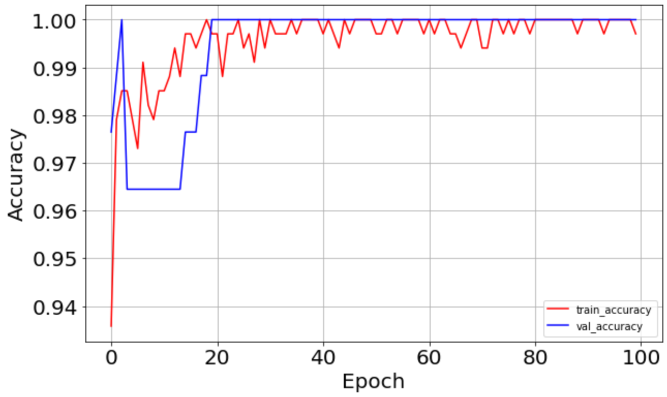
RCA (EM Clustering labels)

Test Accuracy = 98%



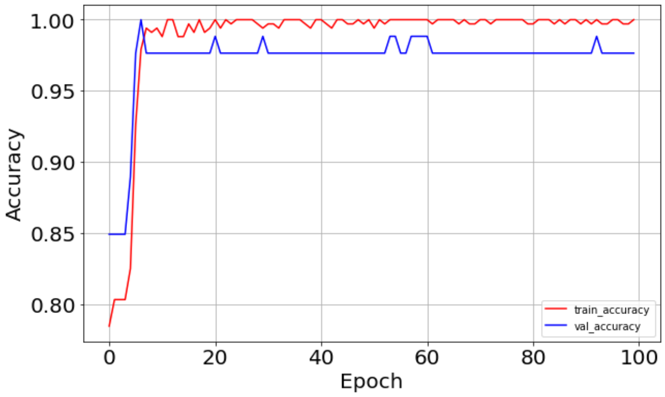
LDA (KMeans Clustering labels)

Test Accuracy = 99%



LDA (EM Clustering labels)

Test Accuracy = 97%



4. Conclusion

- In this assignment we explored unsupervised methods for exploring our datasets. We learned that clustering algorithms can combine multiple classes under one cluster if the features of these classes are close to one another, the KMeans algorithm will use distance between 2 datapoints as similarity metric while Expectation Maximization calculates the probability that certain datapoints belong to a certain distribution together. That is why in the Penguin dataset the 2 algorithms failed to find a third cluster for the Chainstrap species.
- We used clustering features from KMeans and Expectation Maximization algorithms to boost the classification scores for the Heart dataset by using the NN model from assignment 1. Features extracted from both algorithms succeeded in the improving the classification accuracy of the NN model to 97% and 95% respectively.
- We used PCA, RCA, ICA and LDA dimensionality reduction algorithms. There are many important things to point out. The PCA algorithm considered the best dimensionality reduction algorithm with respect to data explanation in terms of variance, eigen values, top features for each component. PCA provides thorough explanation and visualization of data, it has good accuracy scores when used with KMeans clustering. RCA has a very good metric for visualizing the error, which is the squared distance between the original and projected data, and it works very well with EM clustering algorithms. I didn't find ICA useful with my datasets. LDA is an excellent dimensionality reduction algorithm if working on classification problem because it reduces the number of features to number of classes – 1 and produce high classification accuracy scores. Imagine working with dataset that has 700 features and 10 classes, using LDA algorithm can reduce the number of features to 9 features instead of 700, which reduces the computational complexity.