I have downloaded all datasets from the UCI Machine Learning Repository. In addition, I've used Scikit-learn, Numpy, Pandas and other approved libraries. The source code, input and readme files can be found in: https://github.com/mnarasim/ML7641/tree/master/HW3

Introduction: I've considered two diverse data sets for modeling and analysis. Specifically, my first data set is about predicting if someone has a heart disease. The next dataset is about predicting if a customer will default on their credit card payment in future. Please note that my justification and write-up for selecting these data sets is same as in HW1 (I'm just reusing the content here).

To me both these datasets are interesting because they represent common real-life problems. Though the accuracy of the prediction is important for both the datasets, I feel like there is no room for error in predicting if someone has a heart disease or not. If someone is falsely diagnosed with a heart disease, then it can have significant mental trauma on that person. On the other hand, if a diagnosis is missed it can have fatal repercussions as well. However, in the case of the credit card dataset any error in predictions don't have serious repercussions.

Heart.csv: This dataset has 1,025 samples where each sample has 13 attributes. Though the sample size seems low, I like this dataset because in such small datasets the likelihood of having noisy data will be minimal. Thus, my model will not over fit the data. Each of the attributes have a diverse minimum and maximum range which makes the classification task interesting. For example, the age attribute varies from 29 –77 while resting blood pressure varies from 94 – 200. The data also includes categorical variables like Male and Female, though they have been converted to numbers. Also, this dataset includes almost 50% - 50% of the classifications of each type (heart disease or no heart disease). This way we can be sure that the dataset is not biased in anyway and has a good sample set of both the classification labels. Furthermore, as shown below most of the attribute values are not normally distributed which makes the dataset complex and pose a challenge to the learning algorithm.

Heart.csv	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00	1025.00
mean	54.43	0.70	0.94	131.61	246.00	0.15	0.53	149.11	0.34	1.07	1.39	0.75	2.32	0.51
std	9.07	0.46	1.03	17.52	51.59	0.36	0.53	23.01	0.47	1.18	0.62	1.03	0.62	0.50
min	29.00	0.00	0.00	94.00	126.00	0.00	0.00	71.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	48.00	0.00	0.00	120.00	211.00	0.00	0.00	132.00	0.00	0.00	1.00	0.00	2.00	0.00
50%	56.00	1.00	1.00	130.00	240.00	0.00	1.00	152.00	0.00	0.80	1.00	0.00	2.00	1.00
75%	61.00	1.00	2.00	140.00	275.00	0.00	1.00	166.00	1.00	1.80	2.00	1.00	3.00	1.00
max	77.00	1.00	3.00	200.00	564.00	1.00	2.00	202.00	1.00	6.20	2.00	4.00	3.00	1.00

age -> Null hypothesis can be rejected
sex -> Null hypothesis can be rejected
cp -> Null hypothesis can be rejected
trestbps -> Null hypothesis can be rejected
chol -> Null hypothesis can be rejected
fbs -> Null hypothesis can be rejected
restecg -> Null hypothesis can be rejected
thalach -> Null hypothesis can be rejected
exang -> Null hypothesis can be rejected
oldpeak -> Null hypothesis can be rejected
slope -> Null hypothesis can be rejected
ca -> Null hypothesis can be rejected
thal -> Null hypothesis can be rejected
ID -> Null hypothesis can be rejected
LIMIT_BAL -> Null hypothesis can be rejected
SEX -> Null hypothesis can be rejected
EDUCATION -> Null hypothesis can be rejected
MARRIAGE -> Null hypothesis is accepted
AGE -> Null hypothesis can be rejected

Credit Card.csv: This dataset has 30,000 samples where each sample has 24 attributes. As before, each of the attributes has a diverse minimum and maximum range which makes the classification task interesting. Also, this dataset includes negative values for some attributes. The data also includes categorical variables like Male and Female, though they have been converted to numbers. Unlike the previous dataset, this dataset includes 78% of the samples with a 0 (No default) classification, and 1 (will default) classification. It would be interesting to see how various learning algorithms can overcome the bias in the dataset and come up with an accurate prediction. Furthermore, as shown below none of the attributes are normally distributed which makes the dataset complex and pose a challenge to the learning algorithm.

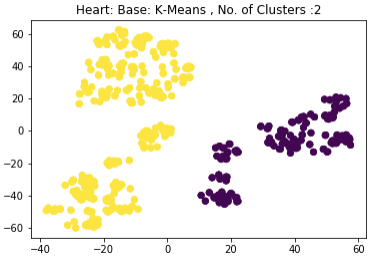
Credit Card.csv	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
count	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00
mean	15,000.50	167,484.32	1.60	1.85	1.55	35.49	(0.02)	(0.13)	(0.17)	(0.22)	(0.27)	(0.29)	51,223.33	49,179.08	47,013.15	43,262.95	40,311.40	38,871.76	5,663.58	5,921.16	5,225.68	4,826.08	4,799.39	5,215.50	0.22
std	8,660.40	129,747.66	0.49	0.79	0.52	9.22	1.12	1.20	1.20	1.17	1.13	1.15	73,635.86	71,173.77	69,349.39	64,332.86	60,797.16	59,554.11	16,563.28	23,040.87	17,606.96	15,666.16	15,278.31	17,777.47	0.42
min	1.00	10,000.00	1.00	0.00	0.00	21.00	(2.00)	(2.00)	(2.00)	(2.00)	(2.00)	(2.00)	(165,580.00)	(69,777.00)	(157,264.00)	(170,000.00)	(81,334.00)	(339,603.00)	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	7,500.75	50,000.00	1.00	1.00	1.00	28.00	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	3,558.75	2,984.75	2,666.25	2,326.75	1,763.00	1,256.00	1,000.00	833.00	390.00	296.00	252.50	117.75	0.00
50%	15,000.50	140,000.00	2.00	2.00	2.00	34.00	0.00	0.00	0.00	0.00	0.00	0.00	22,381.50	21,200.00	20,088.50	19,052.00	18,104.50	17,071.00	2,100.00	2,009.00	1,800.00	1,500.00	1,500.00	1,500.00	0.00
75%	22,500.25	240,000.00	2.00	2.00	2.00	41.00	0.00	0.00	0.00	0.00	0.00	0.00	67,091.00	64,006.25	60,164.75	54,506.00	50,190.50	49,198.25	5,006.00	5,000.00	4,505.00	4,013.25	4,031.50	4,000.00	0.00
max	30,000.00	1,000,000.00	2.00	6.00	3.00	79.00	8.00	8.00	8.00	8.00	8.00	8.00	964,511.00	983,931.00	1,664,089.00	891,586.00	927,171.00	961,664.00	873,552.00	1,684,259.00	896,040.00	621,000.00	426,529.00	528,666.00	1.00

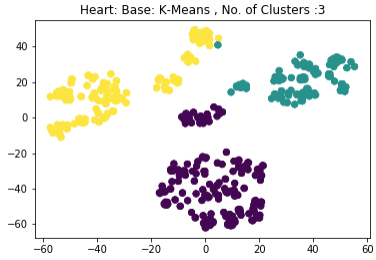
PAY_0 -> Null hypothesis can be rejected
PAY_2 -> Null hypothesis can be rejected
PAY_3 -> Null hypothesis can be rejected
PAY_4 -> Null hypothesis can be rejected
PAY_5 -> Null hypothesis can be rejected
PAY_6 -> Null hypothesis can be rejected
BILL_AMT1 -> Null hypothesis can be rejected
BILL_AMT2 -> Null hypothesis can be rejected
BILL_AMT3 -> Null hypothesis can be rejected
BILL_AMT4 -> Null hypothesis can be rejected
BILL_AMT5 -> Null hypothesis can be rejected
BILL_AMT6 -> Null hypothesis can be rejected
PAY_AMT1 -> Null hypothesis can be rejected
PAY_AMT2 -> Null hypothesis can be rejected
PAY_AMT3 -> Null hypothesis can be rejected
PAY_AMT4 -> Null hypothesis can be rejected
PAY_AMT5 -> Null hypothesis can be rejected
PAY_AMT6 -> Null hypothesis can be rejected

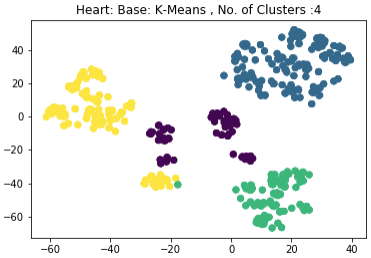
I also scaled the data using scikit-learn scaling functions to avoid any bias due to the diverse data ranges of attributes.

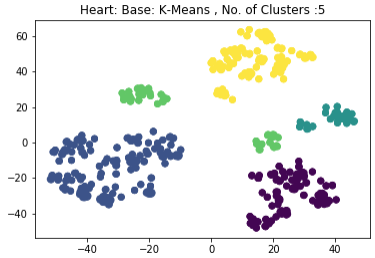
1) I ran the clustering algorithms on the two datasets and observed the below. For the K-Means algorithm I varied the number of clusters whereas for the Expected Maximization (EM) algorithm I varied the number of components. For visualization purposes, I reduced the number of components to 2 to plot the scatter plots below. If I look at the heart data set, I can see that the model predicts the data well when the number of clusters is set to 2. Though the clustering of points in the middle can be debated, since I know that the dataset is inherently used for the classification problem where the target variable is supposed to tell me if someone has heart disease or not, I can be contended with the number of clusters as 2. For increasing number of clusters, I do see that the model predicts some points as part of a cluster though the distance from that point to all other points in that cluster is large. Having said that, if I didn't know about the target variable at all, I would indeed pick the number of clusters as 6, because it tells me more about the points in the middle of the plots as shown below. The thought is that increasing the number of clusters does tell more about the patterns in the data which is not obvious if the task was just simple classification of a binary variable.

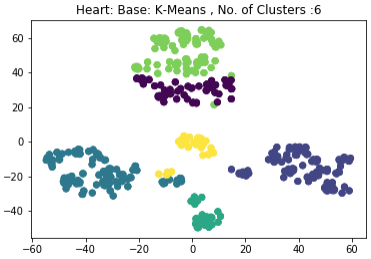
I would follow a similar reasoning for the EM algorithm as well, I would select the number of components as 2 if the primary objective was classification but if I wanted the data to show me hidden patterns beyond the simple classification I would select the number of components as 5 or 6.

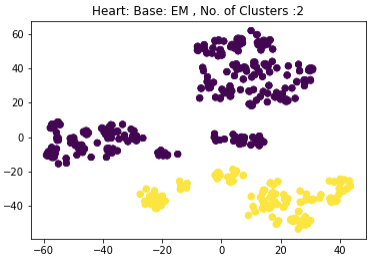


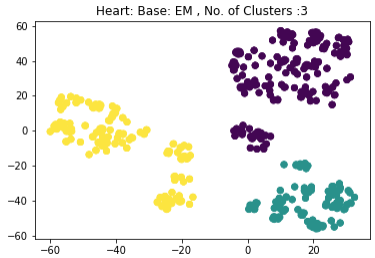


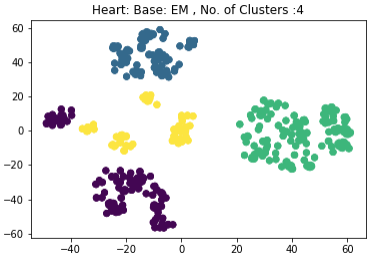


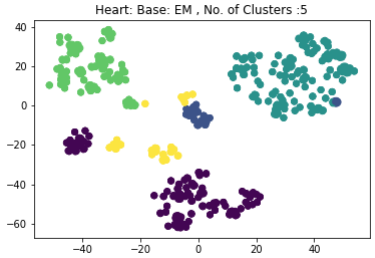


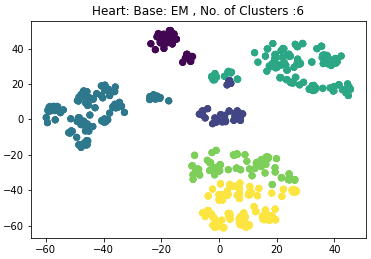




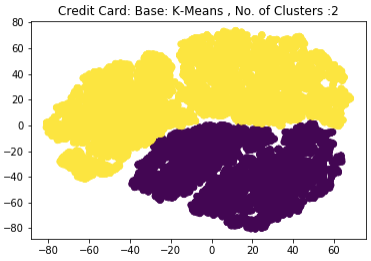


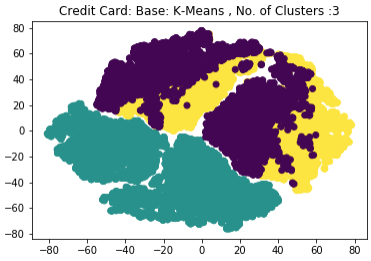


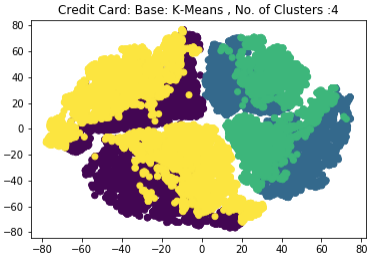


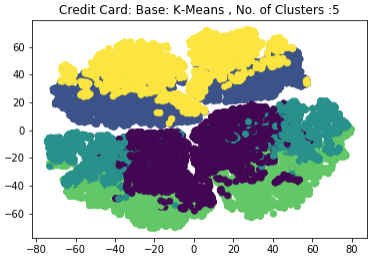


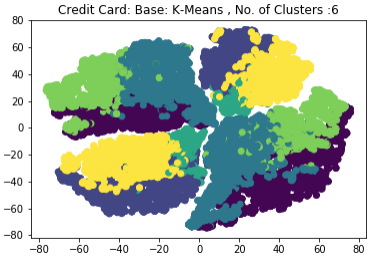
For the credit card dataset, I see a similar pattern as above. However, as shown below both the models don't behave well for higher number of clusters or components. Therefore, for this dataset I would indeed use a value of 2 both for the number of clusters and components. For K-Means, I can see that many points are incorrectly grouped as part of a cluster when the number of clusters is high. The same reasoning can be applied for EM as well.

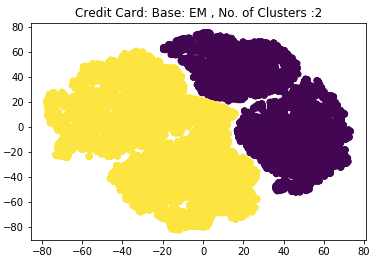


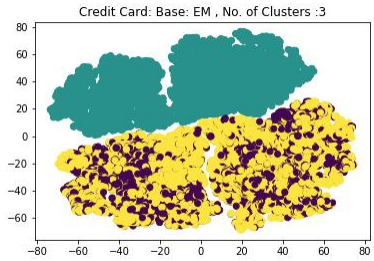


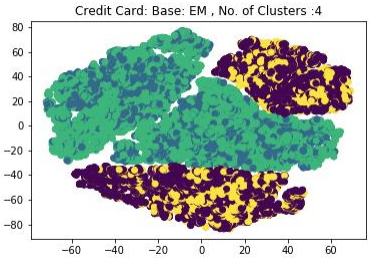


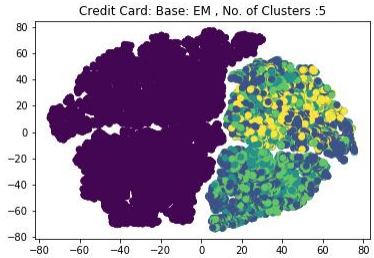


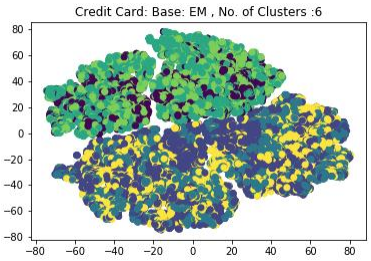










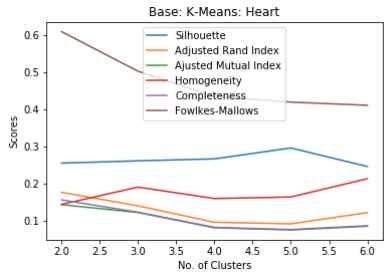


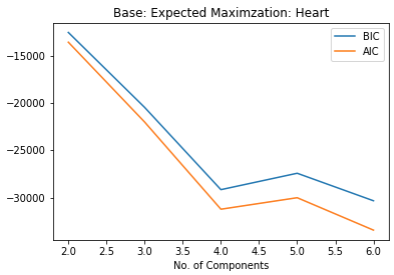
If I look at the Silhouette scores for the heart dataset, I can see that the highest value is achieved when the number of clusters is 5. However, the Silhouette score is indeed more or less flat for all cluster sizes. The Silhouette score of about 0.3 for varying cluster sizes doesn't indicate a strong fit though. However, if I look at Fowlkes-Mallows score the best value is achieved when the number of clusters is 2. This is because FM score includes False Positives and False Negatives – FM score is just the geometric mean between the Precision and Recall. Since our target label is just a binary classification variable, the number of clusters set to 2 works well.

Homogeneity score, that measures how many observations with the same class label are in the same class, increases with the number of clusters despite a slight drop at 4. In any case, a homogeneity score around 0.2 indicates that there are many clusters that contain data points there are members of different classes. Completeness that measures how many observations of a given class are assigned to the same cluster, decreases with the increase in the number of clusters. Completeness score of 0.2 also indicate that the members of the same class are in different clusters.

Adjusted Rand and Adjusted Mutual Index Scores that measure the similarity between the predicted and the true clusters are also low corroborating the points mentioned above.

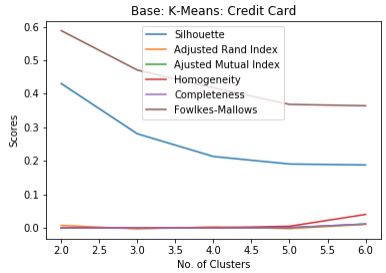
For EM, both AIC and BIC decrease with increasing number of components. Overall, these two measures indicate how fit the model is, where the lower the value the better. Based on these values the number of components as 6 would work well.

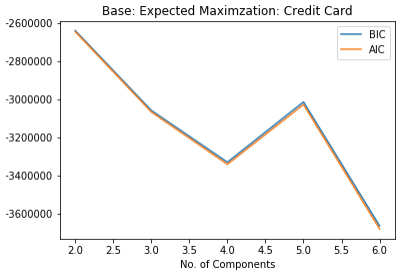




For credit card dataset both the Silhouette scores and FM scores indicate that the best number of clusters is 2. All the other metrics are consistent with the observations for the heart dataset.

For EM both AIC and BIC decrease with increase in the number of components, though there is a brief increase at 4.0 only to drop at 5.0 and beyond.

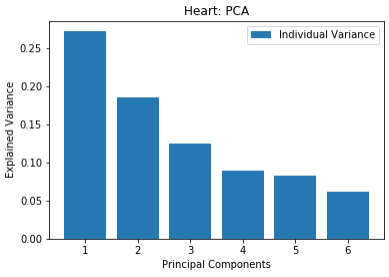


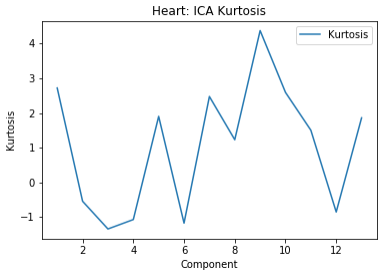


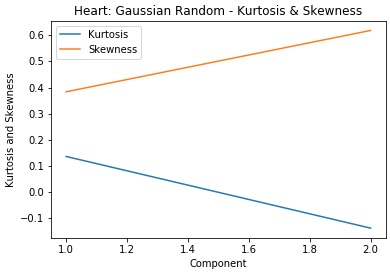
2) Next I ran the dimensionality reduction algorithms for varying number of components and selected the best number of components based on the following criteria:

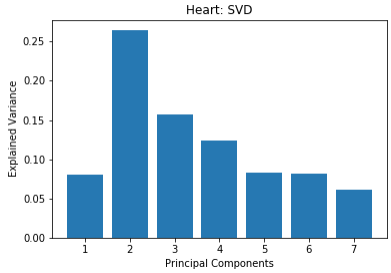
- PCA: the number of components that can explain 80% of variance in the data
- ICA: the number of components that yield the highest average Kurtosis
- Random Gaussian: the number of components that yield the highest average Kurtosis
- SVD: the number of components that can explain 80% of variance in the data

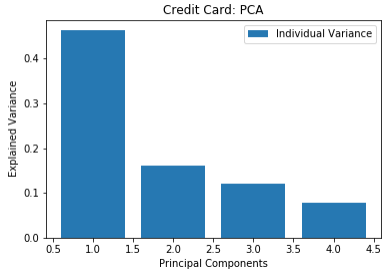
The following graphs were generated for the optimum number of components chosen based on the above criteria.

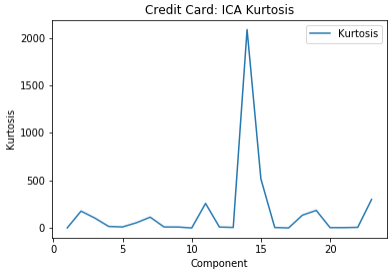


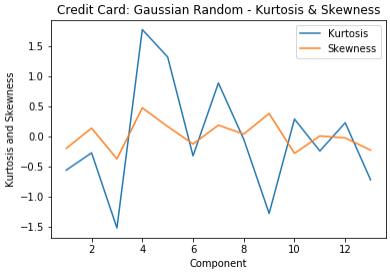


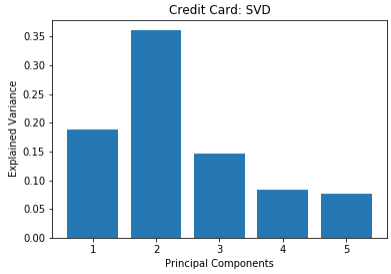






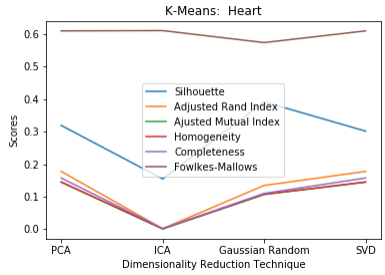


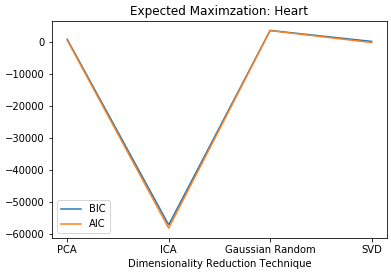


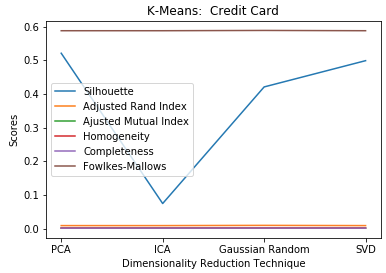


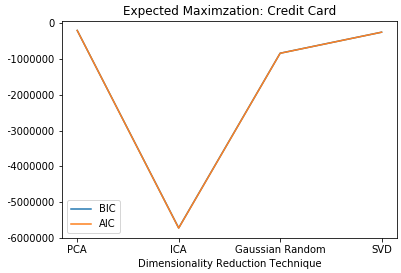
3) With the optimum number of components selected for each of the dimensionality reduction algorithm as mentioned in step 2, I ran both the clustering algorithms on each dataset.

For both the datasets, I can see that PCA performs the best on K-Means, SVD comes next and Random Gaussian comes third. However, ICA performs the best when EM is applied on the data. Though PCA and SVD produce similar results SVD requires more components than PCA to explain the same percentage of variance in the data.



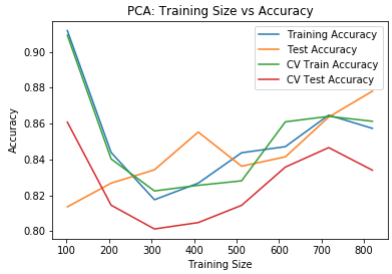


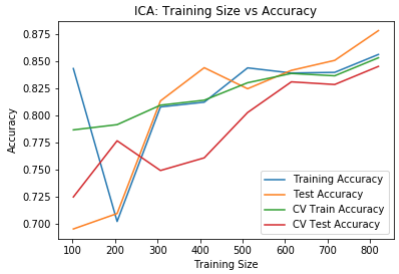


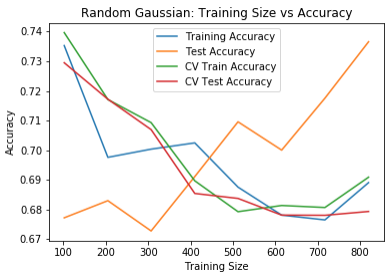


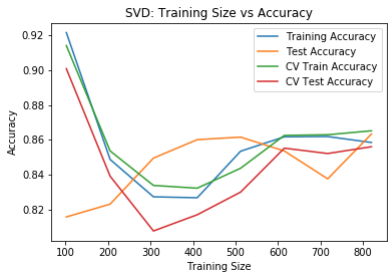
4) Next, I used the dimensionally reduced data of the heart data set and applied a neural network classifier to that data. I then varied the training, test dataset sizes and observed the corresponding accuracies. I also performed cross validation to assess the impact on the accuracies.

Overall, the accuracies are better when the data obtained after applying PCA is used. However, in this case I can see that the training accuracy is high for low training data set size and decreases until the training data size is 300, after which it climbs back up. This tells me that perhaps PCA is causing bias in my model - it oversimplifies the model at low training dataset size thus the training accuracy decreases. After a dataset size of 300, the model becomes complex enough for the accuracy to climb back up. I can see the same behaviors for the data obtained after applying SVD and to a certain extent in the data from Random Gaussian. In terms of accuracies, I would say Random Gaussian is the least accurate.

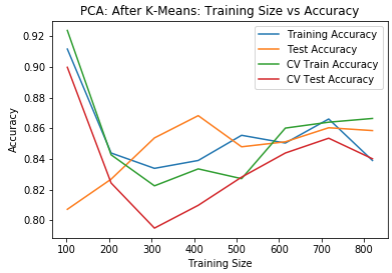


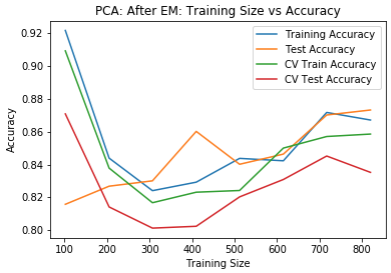


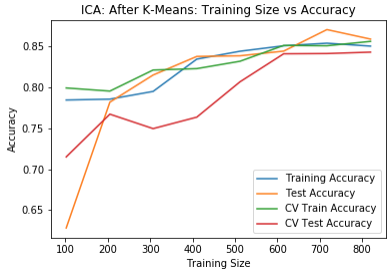


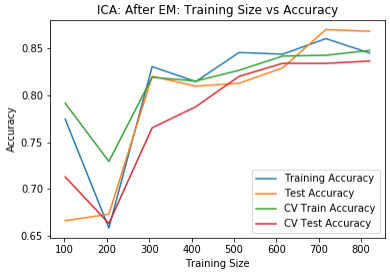


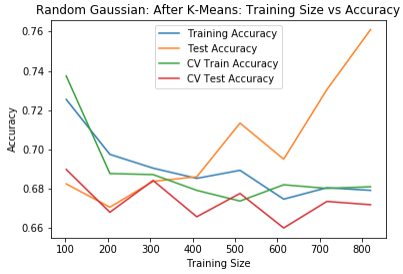
5) After running the clustering algorithms on the heart dataset, I added the cluster labels as another feature and fit a neural network learner on the resultant data. As observed before, the data derived after applying PCA seems to produce the best accuracies. The same can be said for data derived after applying SVD. The type of clustering algorithm doesn't seem to affect the accuracies much, in fact both K-Means and EM show similar pattern in the accuracies for various training/test dataset sizes. As mentioned before Random Gaussian does seem to be the least accurate.

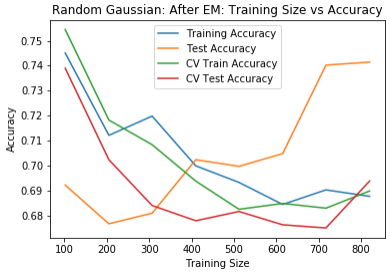

PCA: After K-Means: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size

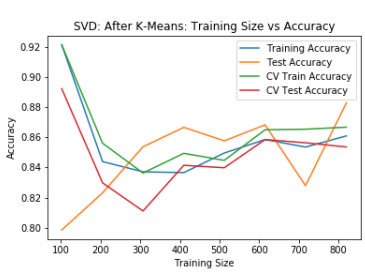

PCA: After EM: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size

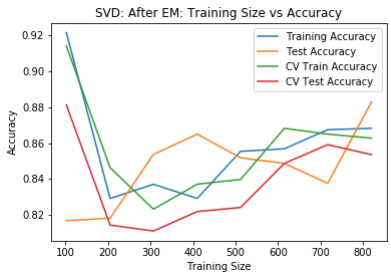

ICA: After K-Means: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size


ICA: After EM: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size


Random Gaussian: After K-Means: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size


Random Gaussian: After EM: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size


SVD: After K-Means: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size


SVD: After EM: Training Size vs Accuracy
Training Accuracy
Test Accuracy
CV Train Accuracy
CV Test Accuracy
Accuracy
Training Size