Unsupervised Learning and Dimensionality Reduction Report

Dataset 1: Credit Approval Dataset

Description of Classification problems:

The credit approval data has 10 columns including 9 independent variables and 1 target variable. It uses customers' demographic data and other credit-related data to show whether the bank will approve or reject the credit card application.

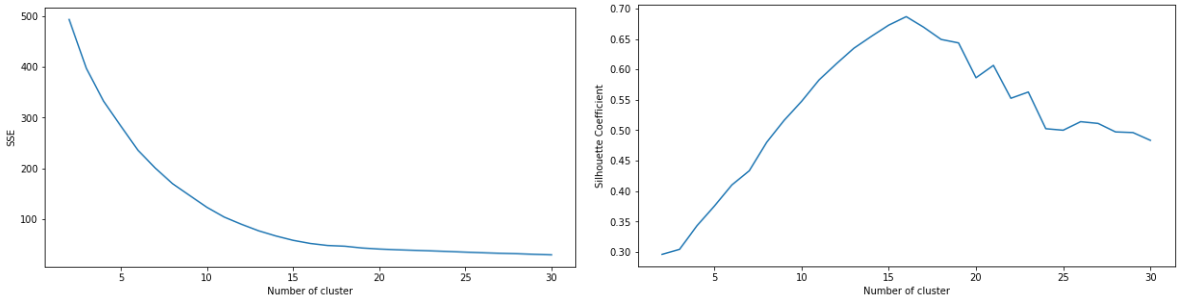
Why interesting:

It has both continuous data and categorical data in the dataset. The features, such as debt amount, years of employment, historical delinquency, and credit score, are under consideration when bank makes the decision. The dependent variable has 2 values which are "approved" or "not approved".

Clustering:

- ### K-Means

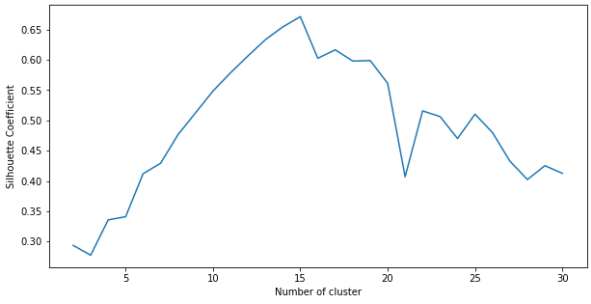
K-means clustering is trying to group the sample points into k cluster. It starts with k random cluster centers and assign the points to the closest center. Then the cluster centers will be recomputed by averaging the points within current clusters. Repeating assigning points to their closest centers until no points being reassigned.



In the left graph, we can see the SSE drops down quickly before number of clusters equal to 15. So I think the best number of clusters will range between 15 and 20. Based on the right graph above, we can see when clusters=17, the silhouette score hit the highest point of 0.68. Silhouette score is to measure how all clusters are differentiated against each other. If the silhouette score close to 1, it means the clusters are split very well.

- ### Expectation Maximization

Expectation maximization is a iterative approach to cycle expectation step and maximization step. In expectation step, we compute the probability of the point comes from different clusters. In maximization step, we need to optimize the parameters of the distribution by maximizing the log-likelihood. In this assignment, I will use Gaussian Mixture model. It assumes the data follow the normal distribution. At each step, the cluster is weighted by its probability of falling in the cluster. And the cluster center is recomputed by weighing all points.



Based on the graph above, we can see when clusters=15, the silhouette score hit the highest value of 0.66

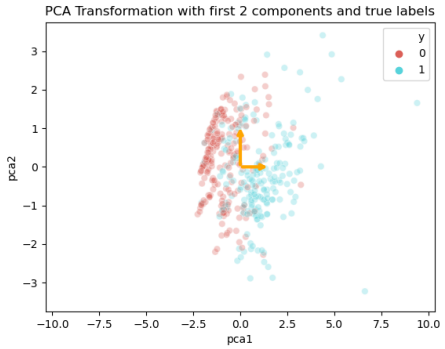
Dimensionality Reduction:

- ### PCA

PCA is commonly used to reduce the dimensionality of a dataset and keep as much information as possible. The first principal component is in the direction of maximum variance and the subsequential principal components will be in the direction of

second, third… maximum variance. Each component is orthogonal to the previous component, so the different dimensions are uncorrelated with each other. PCA is a matrix of eigenvector of the points.

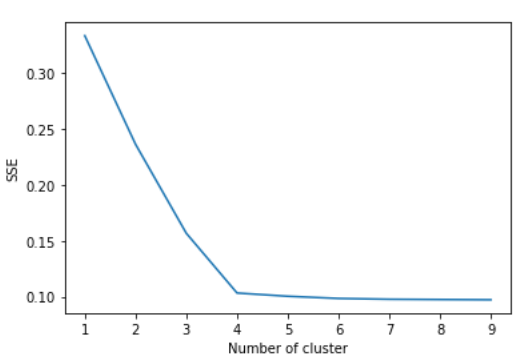
dim	explained_variance_ratio
1	0.350783
2	0.592894
3	0.791455
4	0.924812
5	0.960892
6	0.985310
7	0.994281
8	0.997375
9	1.000000

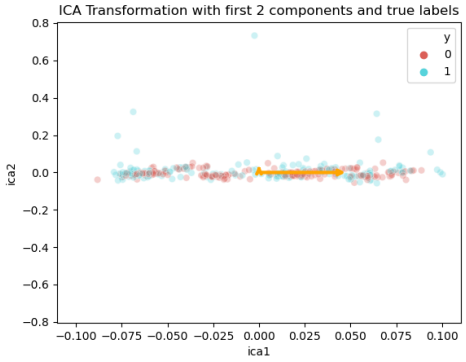


In the table above, it shows the cumulative explained variance. I think it's better to keep the first 4 components because they carry 92% information of the data. The rest 5 components can be ignored because they only cover 8% information. In the right graph, we can see the first 2 components are orthogonal to each other, meaning the information they cover are not overlapped and these 2 components are independent to each other. Even though some red and blue dots are mixed, we can see first 2 components already distinguish the red and blue dots pretty well.

- **ICA**

ICA stands for independent component analysis. Given the original features, ICA is to find the new features which are independent to each other. In other words, the new features share minimal mutual information. But ICA also wants to maximize the mutual information between old and new features.

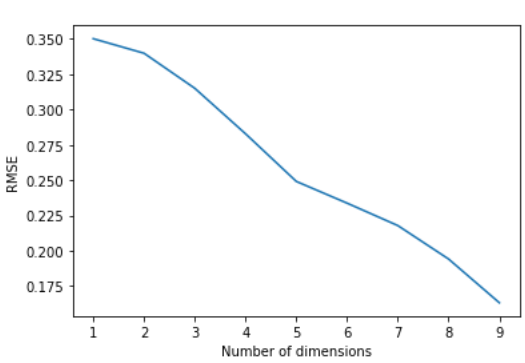


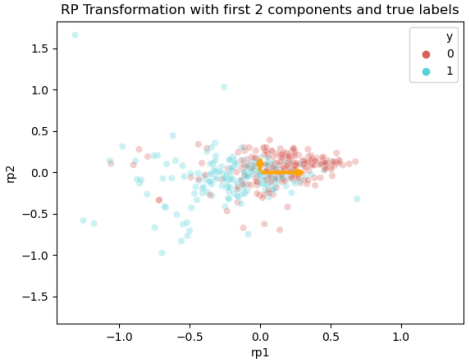


In the left graph, we can see the best number of clusters is 4 because it will result in the minimal squared error. In the left graph, it shows how true labels look like when just using first 2 ICA components.

- **RCA**

RCA stands for random component analysis. It is also called random projection. RCA generates random directions and can capture the correlations in the data. It is good to use when we want to solve the classification problem. It works very fast and have low error rate compared to other dimensionality reduction methods.



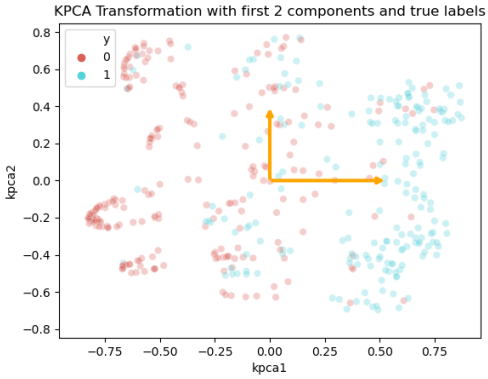


In the left graph, I can see the error has linear relationship with number of dimensions. I would select number of clusters as 5. In the right graph, we can see the first 2 components can separate the points as blue and red groups.

- **Kernel PCA**

Kernel PCA is mainly used for nonlinear dimensionality reduction. It is the extension of PCA method. By adding the kernel such as sigmoid, rbf and poly, it can project the data to a higher dimensional space and very useful when data are not linearly separated.

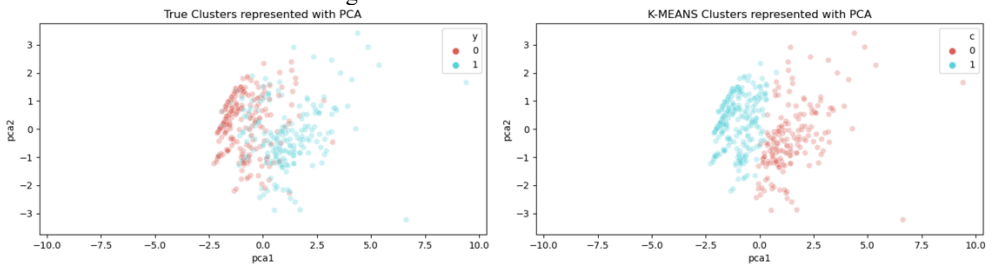
component	explained_variance
1	0.373083
2	0.630680
3	0.841996
4	0.984020
5	0.991688
6	0.996878
7	0.998785
8	0.999442
9	1.000000



In the left table, we can see the first 4 components cover 98% variance, so I think the best number of components is 4. The rest 5 components only covers the remaining 2% information. In the left graph, I can see kernel PCA does a job of separating the data points in the higher dimension and the components are independent with each other.

Clustering + Dimensionality Reduction:

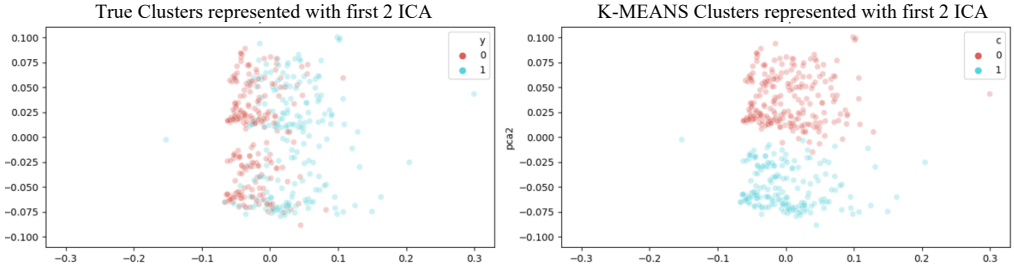
- PCA + K-means Clustering



- ICA + K-means Clustering

True Clusters represented with first 2 ICA

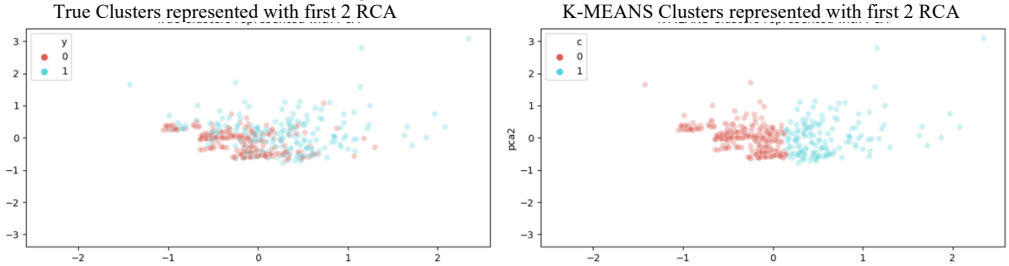
K-MEANS Clusters represented with first 2 ICA



- RCA + K-means Clustering

True Clusters represented with first 2 RCA

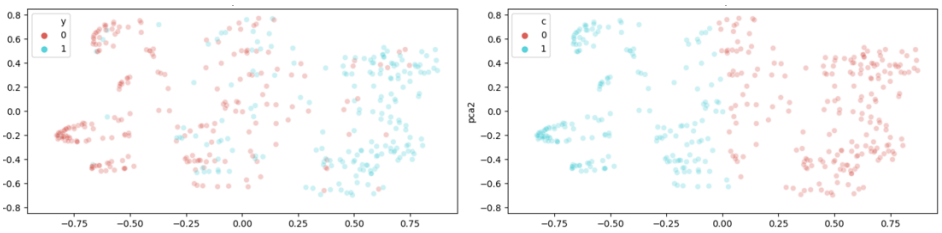
K-MEANS Clusters represented with first 2 RCA



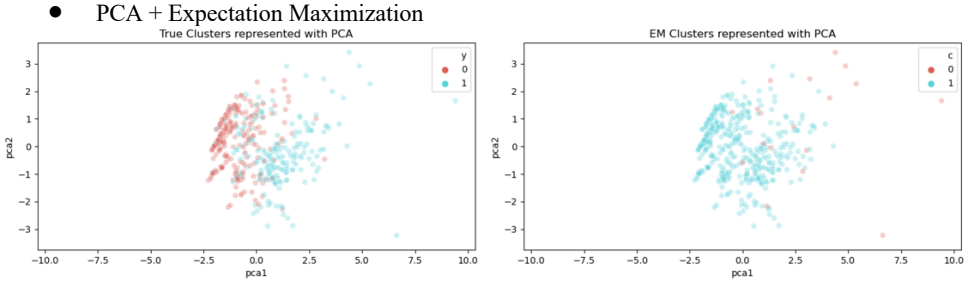
- KPCA + K-means Clustering

True Clusters represented with first 2 KPCA

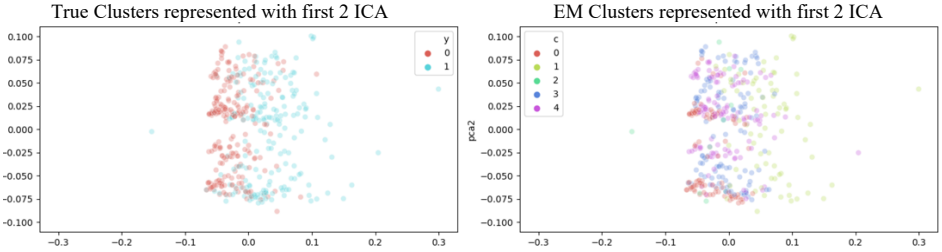
K-MEANS Clusters represented with first 2 KPCA



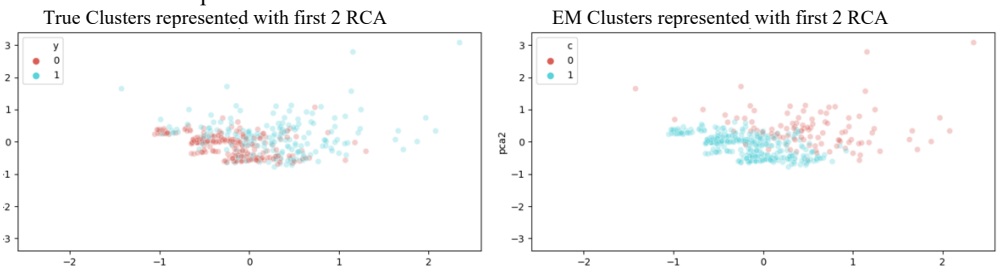
- PCA + Expectation Maximization



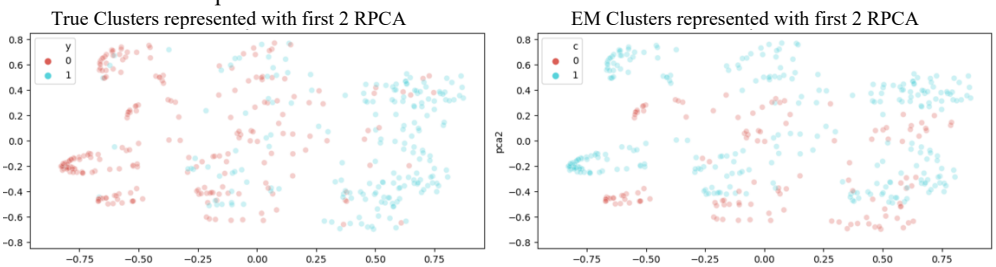
- ICA + Expectation Maximization



- RCA + Expectation Maximization



- KPCA + Expectation Maximization



For the first combination listed above, in the right graph, I can see how true labels distributed with first 2 PCA components. In the left graph, I can see k-means clustering methods is good at clustering the data points into 2 groups using first 2 components.

For the second combination, in the left graph, we can see true red and blue points are distributed to the left and right side respectively. But the combination of ICA and k-means clustering result in 2 groups distributed in the upper and lower side. I think it is because ICA make an assumption that the hidden variables are independent. So I will try to find the structure instead of using the average. So ICA is easily affected by the directionality That's why in this case it wrongly distinguish 2 groups.

Based on the last 4 combination, I can see expectation maximization is not as good as k-means in this credit approval problem. I think it is because expectation maximization uses the probability to assign the points to the closest cluster. It involves soft clustering so some points could be assigned to either clusters.

Dimensionality Reduction + Neural Network:

	Precision	Recall	F1 Score	Accuracy	Training Fitting time(seconds)
NN	0.92	0.75	0.83	0.86	0.2674
PCA + NN	0.91	0.62	0.74	0.80	0.2703
ICA + NN	0	0	0	0.55	0.2359
RCA + NN	0.96	0.48	0.64	0.76	0.2515
KPCA + NN	0.94	0.62	0.75	0.81	0.2570

Based on the table above, I think the best model is the combination of KPCA and neural network model because it has the highest accuracy rate and F1 score is also high. Even though its fitting time is the longest, the differences between the fitting time is not very large. I think the reason why KPCA is the most helpful dimensionality reduction method is because the data is not linearly separated. So the kernel of rbf will help map the data to a higher dimension and then create the components which can be used to improve the model performance. The combination of ICA and neural network performs worst. I think the reason is because ICA assume the hidden variable – the new features we transform to are independent. So, if the real hidden variables are correlated, ICA might not be helpful in the machine learning model.
I noticed that the original neural network model outperforms than all other models with dimensionality reduction. I think the reason is because the original model uses all original features, so the model uses 100% information to make the prediction. But the model after using dimensionality reduction might lose some information. For example, PCA model only uses first few components which can cover over 90% variance but not 100% variance. In this case, it makes sense that the original neural network model has the best result. But it also takes a little longer time because more dimensions are involved when training the data.

Clustering + Neural Network:

	Precision	Recall	F1 Score	Accuracy	Training Fitting time(seconds)
K-means + NN	0.88	0.73	0.80	0.83	0.2545
Expectation Maximization + NN	0.86	0.75	0.80	0.83	0.2607

According to the summary table above, I think both k-means clustering and expectation maximization perform well. They have similar performance. I can see the precision, recall, F1 score, and accuracy rate are close (sometimes even same) to each other. The fitting time is also similar.

Dataset 2: Breast Cancer Dataset

Description of Classification problems:

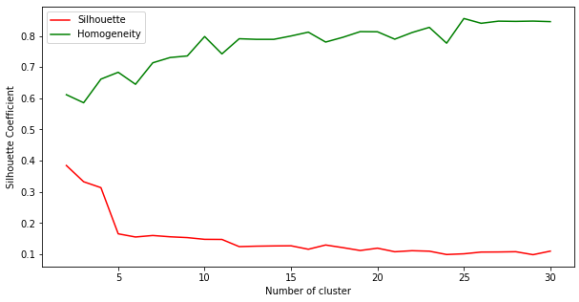
The breast cancer has 30 columns and 1 target variable. The independent variables are the description of the cell like the texture, the perimeter, the share. The target variable only contains 0 and 1 values to indicate whether it will result in cancer or not.

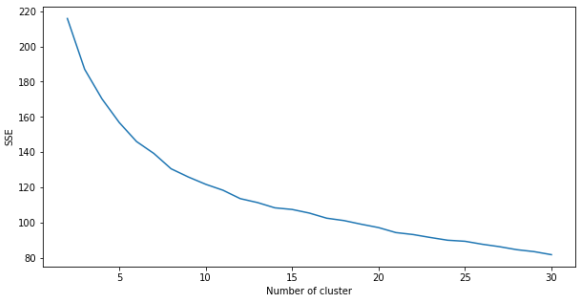
Why interesting:

It has continuous data in the dataset and different column values ranges differently. This dataset has a lot of features. If we fit the data directly into the model, we might face the curse of dimensionality problem. So, it is a good dataset to let me try various dimensionality method and clustering method.

Clustering:

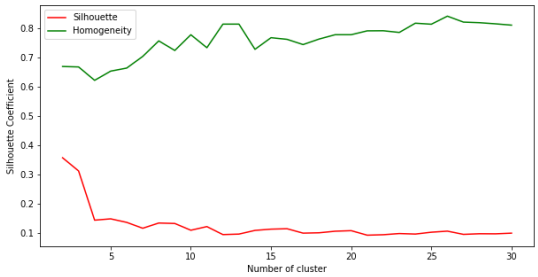
- **K-Means**





I think the best number of cluster is 4 because the line of silhouette score decreases quickly after number of clusters=4. Also, when number of cluster is 4, the error is not very high and it lies on the points where the gradient starts becoming slow according to elbow method. So I think 4 is the best.

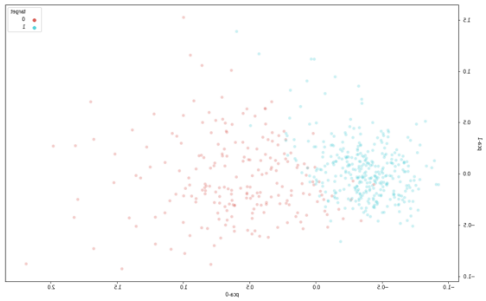
- **Expectation Maximization**

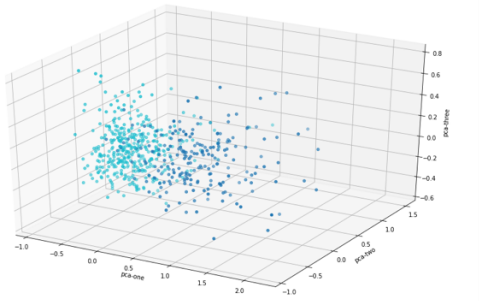


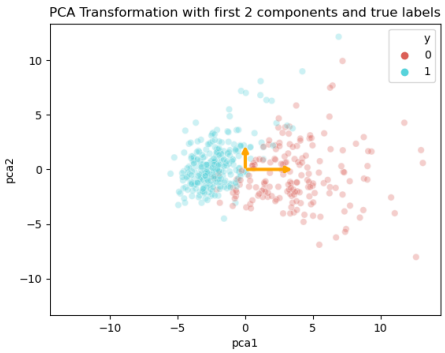
Based on the graph above, I think the best number of clusters is 2 because after that point, the silhouette score drops dramatically. Two clusters make sense to me because in this problem, we also have 2 values (yes or no) to see whether the patient will have cancer or not. So if we split the data into 2 groups, we can use it to check wether it is aligned with true labels.

Dimensionality Reduction:

- **PCA**

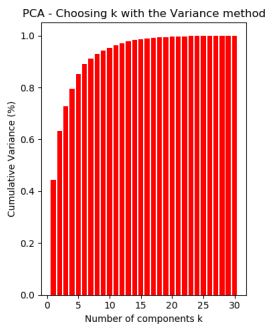


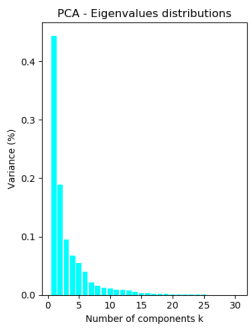




dim	cumulative_explained_variance_ratio
1	0.530977
2	0.703812
3	0.774956
4	0.839069
5	0.879930
6	0.910644
7	0.926453
8	0.938368
9	0.948252
10	0.957706
11	0.966200
12	0.973780
13	0.980346
14	0.985095
15	0.987789
16	0.990366

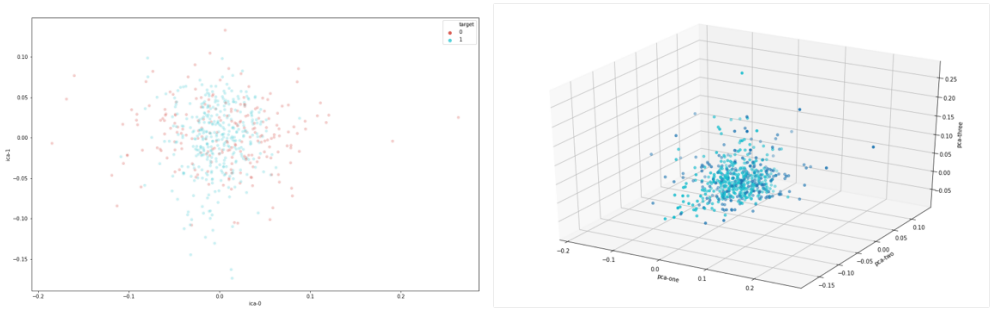
Based on the scatter plot and 3D graph, I can see the data are separated well by using the first 2 PCA components. I think using the first 13 components would be the best choice because they can explain 98% variance and reduce the number of features from 30 to 13.



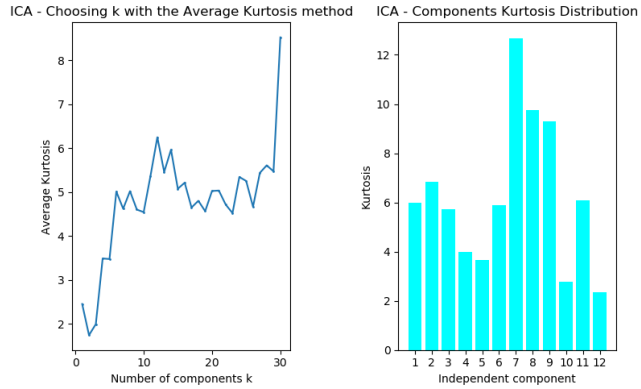


I plot the cumulative variance in the graph and it's easier to see how much variance of each component convers. Also the bar chart in the right side can show the eigenvalues distribution

- **ICA**

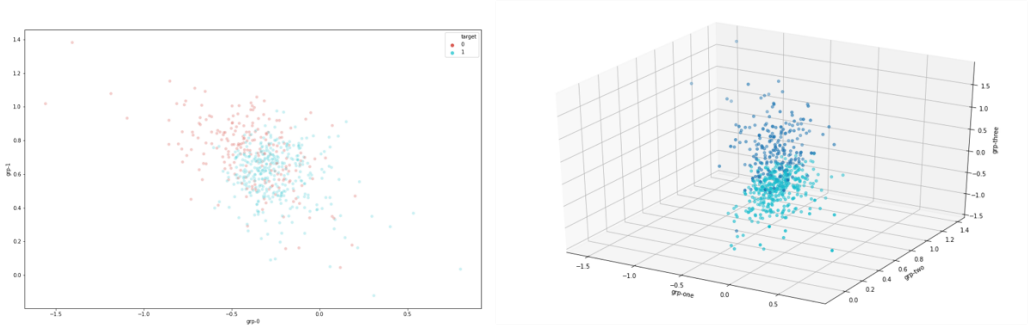


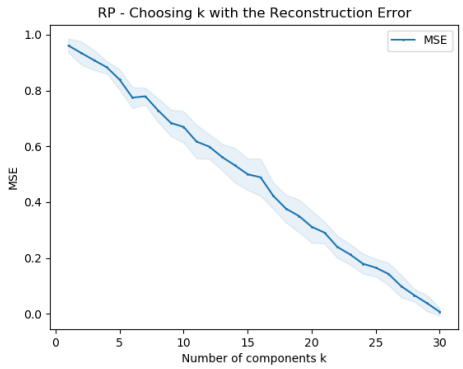
It is obvious that the data points are mixed with each other. The data cannot be easily distinguished by ICA. Since ICA reply on the assumption that hidden variables are independent, which will focus on the structure. In this dataset, I think PCA is better because PCA will focus on the average and result in a better result.



Kurtosis is used to measure the peakness and flatness of Gaussian distribution. If kurtosis values is higher than 0, it has supergaussian Otherwise, it is subgaussian. When kurtosis=0, it has normal distribution.

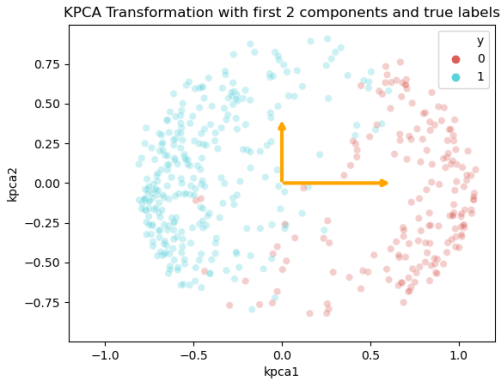
- **RCA**





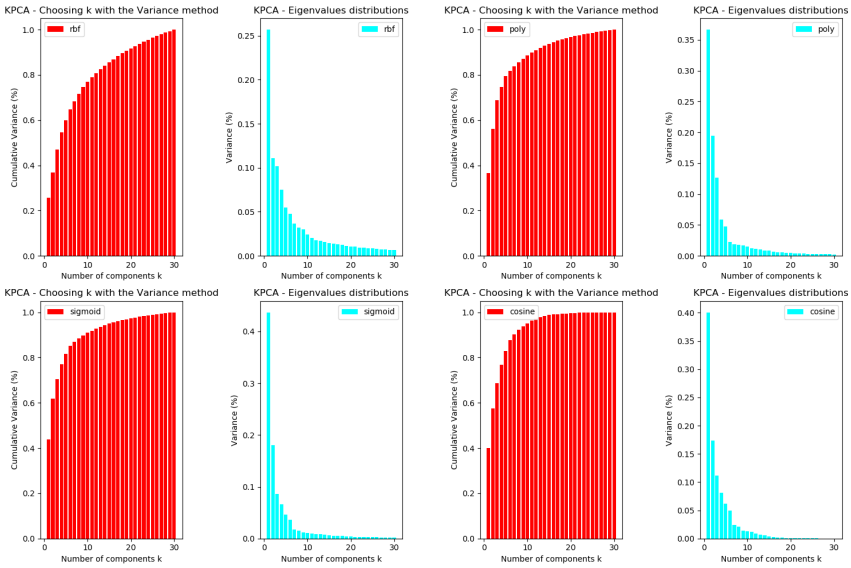
RCA does a good job at separating the data according to the graphs above. Enough random components can capture the relationship among features and it works fast in the mixture of Gaussian. Also, the number of components has linear relationship with mean squared error.

- **Kernel PCA**



component	explained_variance
1	0.536246
2	0.710797
3	0.782647
4	0.847396
5	0.888662
6	0.919682
7	0.935647
8	0.947680
9	0.957663
10	0.967211
11	0.975789
12	0.983444
13	0.990076
14	0.992473
15	0.993834
16	0.995135
17	0.996063
18	0.996827

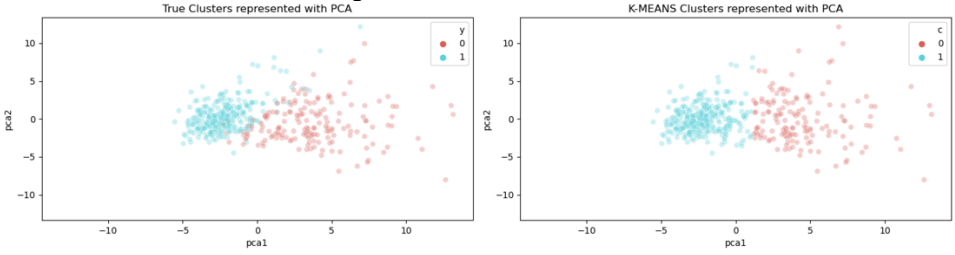
I think number of components should be 12 because 12 components can explain 98% variance and Kernel PCA does well in reducing the dimensionality when data is not linearly separately.



I tried four different kernels here including radial basis function, polynomial kernel, sigmoid kernel and cosine kernel. I can see sigmoid kernel performs best because the first few components can explain most of information (The first component contains 45 variance). The worst one is the rbf kernel. The cumulative variance increase slowly and the first component onlu contain 26% information.

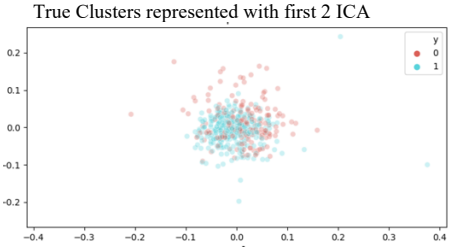
Clustering + Dimensionality Reduction:

- PCA + K-means Clustering

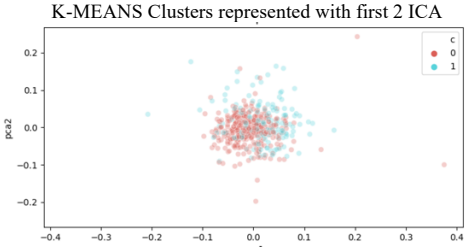


- ICA + K-means Clustering

True Clusters represented with first 2 ICA

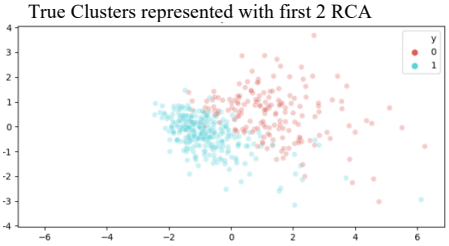


K-MEANS Clusters represented with first 2 ICA



- RCA + K-means Clustering

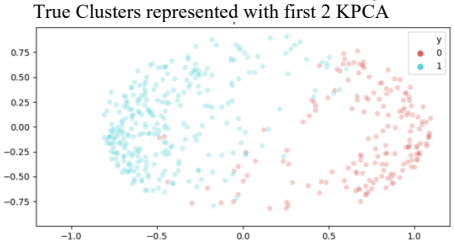
True Clusters represented with first 2 RCA



K-MEANS Clusters represented with first 2 RCA

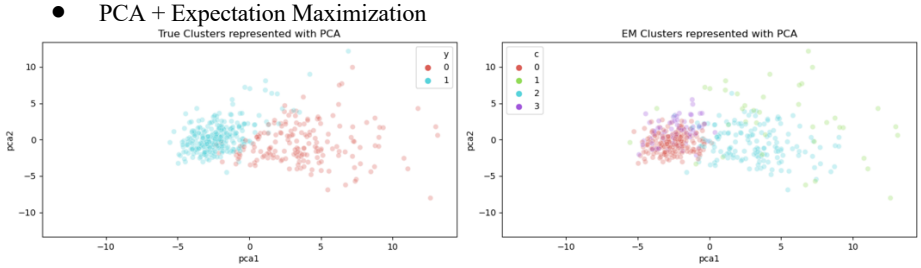
- KPCA + K-means Clustering

True Clusters represented with first 2 KPCA



K-MEANS Clusters represented with first 2 KPCA

- PCA + Expectation Maximization



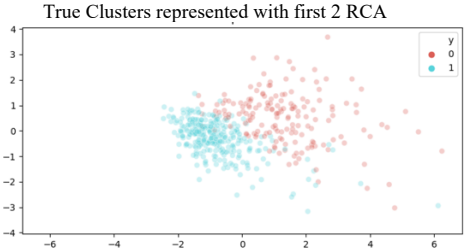
- ICA + Expectation Maximization

True Clusters represented with first 2 ICA

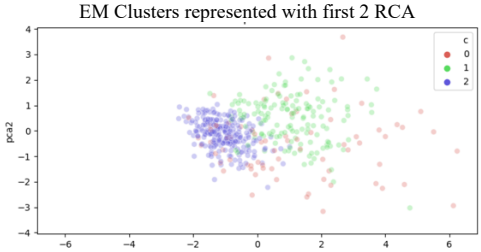
EM Clusters represented with first 2 ICA

- RCA + Expectation Maximization

True Clusters represented with first 2 RCA

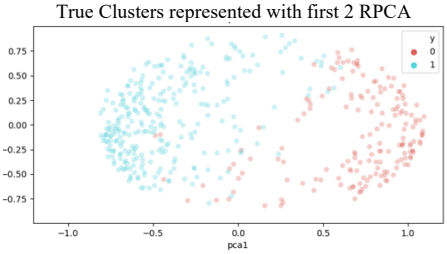


EM Clusters represented with first 2 RCA

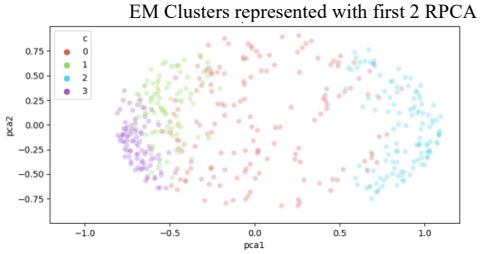


- KPCA + Expectation Maximization

True Clusters represented with first 2 RPCA



EM Clusters represented with first 2 RPCA



In the combination of dimensionality reduction and K-means clustering, the algorithms perform well in separating the data points. I can see the 2 data groups generated from the algorithms are almost aligned with the true labels represented by 2 components. But in the combination of dimensionality reduction and expectation maximization, I can see more clusters are generated in the graph. Since expectation maximization compute the probability and then adjust the cluster probability and mean, we can see the points which have low probability of being assigned to the clusters. Since we have probability, expectation maximization is doing the soft clustering. The points which stay close to the edge of the clusters is not clear to be classified. So those points will be colored in a single cluster.

Dimensionality Reduction + Neural Network:

	Precision	Recall	F1 Score	Accuracy	Training Fitting time(seconds)
NN	0.99	0.97	0.98	0.97	0.1974
PCA + NN	0.99	0.94	0.96	0.96	0.1910
ICA + NN	0.73	1	0.85	0.77	0.1822
RCA + NN	0.97	0.97	0.97	0.96	0.1924
KPCA + NN	0.99	0.94	0.96	0.96	0.1919

According to the results above, I think the best model is the combination of PCA and neural network model because it ues the shortest training time and also achieve pretty high accuracy rate, precision, recall an F1 sore. Even though the original neural network model has the highest accuracy rate, it uses 30 features and takes a little longer time to train the model. RCA and KPAC are also helpful in reducing the training time and in the meanwhile keep the high performance.

Clustering + Neural Network:

	Precision	Recall	F1 Score	Accuracy	Training Fitting time(seconds)
K-means + NN	0.99	0.94	0.96	0.96	0.1912
Expectation Maximization + NN	0.99	0.94	0.96	0.96	0.2126

According to the summary table above, I think it k-means clustering and expectation maximization contribute equally to the neural network model since they have the same performance but k-means clustering uses less training time.

Reference

[1] Brownlee, J. (2020, August 28). A Gentle Introduction to Expectation-Maximization (EM Algorithm). Machine Learning Mastery. https://machinelearningmastery.com/expectation-maximization-em-algorithm/

[2]PCA vs. random projection. (2016, September 18). Cross Validated. https://stats.stackexchange.com/questions/235632/pca-vs-random-projection