1. Introduction

This paper explores the performance of 2 different clustering algorithms:

- K-means
  Starts with k random cluster centers, associates data points with closest centers by distance, improves each center point with mean distance of data points within it. By define, it makes the assumption that clusters are convex and isotropic. Because of the randomly chosen starting centroids, it may falls in local minima.
- EM - Expectation Maximization
  Starts with k random (Gaussian) distributions, associates data points with probability/likelihood, improves the parameters of each (Gaussian) distribution with data points within it. K-means is a special case of EM with no covariance. It improves the performance on irregular shapes. It may also falls in local minima. Convergence is guaranteed but may be slower.

As well as 4 dimensionality reduction algorithms:

- PCA - Principal Component Analysis
  Feature transformation algorithm. PCA linearly transforms and decomposes a multivariate dataset into a set of orthogonal component sorted by their variance, which in total explains maximum amount of the variance of the dataset.
- ICA - Independent Component Analysis
  Feature transformation algorithm. ICA decomposes a multivariate dataset into a set of independent non-Gaussian components, maximizes mutual information between the original data and the components. The components are not required to be orthogonal, but aim to maximize independence and share minimal mutual information.
- RP - Randomized Projections
  Feature transformation algorithm. RP generates random directions and projects the dataset onto them. It works because enough random components can eventually capture correlations in the data, although the resulting feature space may be bigger than other algorithms. It tradeoff accuracy and dimension for computational efficiency. Since the pairwise distances between any two samples of the dataset are controlled, RP is a suitable approximation technique for distance based method.
- RF - Random forests
  Feature selection technique. Decision trees are essentially a filtering algorithm: For each split, pick the best available attribute by ranking their information gain. A list of features sorted by importance will be returned. Feature selection can be done afterwards. The resulting features are a subset of original features.

To measure and compare the performance of these algorithms, several experiments were performed on two datasets:

- Dataset 1 - Breast Cancer Wisconsin (Diagnostic)
  1. Clustering
  2. Dimensionality reduction
  3. Clustering on the results from step 2

4. Run Neural network learner on the results from step 2
5. Run Neural network learner on the results from step 1

- Dataset 2 - Wine Quality-red
  1. Clustering
  2. Dimensionality reduction
  3. Clustering on the results from step 2

The results of the experiments of each dataset are in section 2 and 3 respectively.

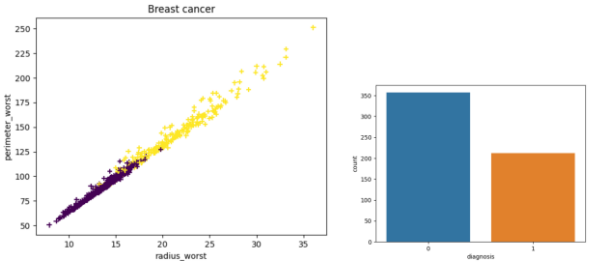
The implementations of these algorithms and evaluation metrics were pulled from the scikit-learn library.

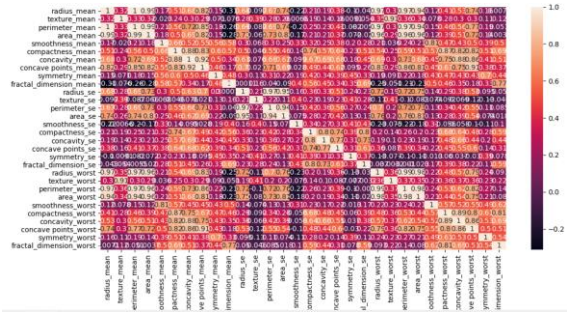
2. Dataset 1

2.1. Description

The dataset I used was Breast Cancer Wisconsin (Diagnostic) samples. There are 569 samples with 30 features. The features are computed from digitized images of breast mass fine needle aspirate (FNA), which describe characteristics of the cell nuclei present in the images. The target is diagnosis (M = malignant, B = benign). In order to make it a binary-classification problem, I turned the target class into malignant as 1, benign as 0.

I chose two dominant features, which I got from random forest learner, to visualize the dataset, as well as the class distribution and feature correlation heat map:





Intuitively, there are two clusters in this dataset. As showed in correlation heap map, there are several correlated features. We may benefit from dimensionality reduction techniques to avoid overfitting and to decrease computational expenses.

In order to train a neural network learner to evaluate the effects of clustering and dimensionality reduction, I randomly split the dataset with 0.2 of it as test set and keep the same distribution in both sets. All the clustering and dimensionality reduction algorithms were performed on training set.

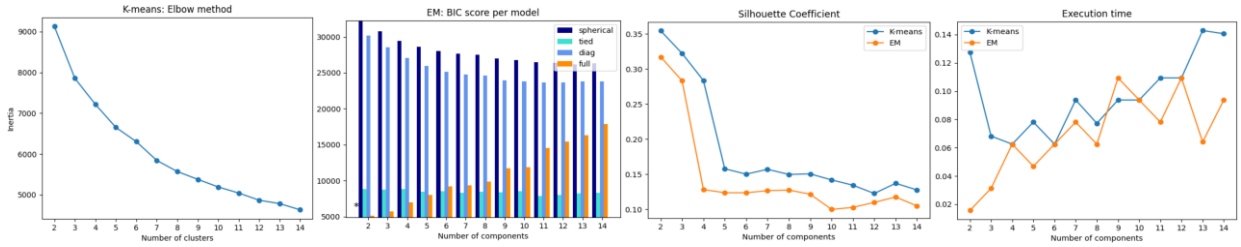
2.2. Clustering

K-means and Gaussian Mixture models, which implement the EM algorithm, are used in this experiment:

1. Standardize the samples.
   Only for K-means model. Since the variance of a feature corresponds to its influence on the clustering algorithm. Equal variance gives every feature equal chance.
2. Determine the optimal number of clusters or components.

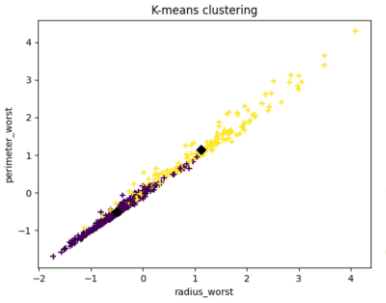
- K-means: Elbow method using inertia. Inertia measures the within-cluster sum-of-squares distance, which tells how far away the points within a cluster are. Choose a K when adding another cluster doesn't give much better modeling of the data.
- EM: Information-theoretic criteria (BIC). The model with the lowest BIC is preferred.

3. Measure and compare performance.
   - Silhouette Coefficient
     Silhouette coefficient measures how similar an object is to its own cluster as compared to other clusters. Range from -1 to 1, as -1 for incorrectly clustering, 1 for highly dense clustering and 0 for overlapping clusters. A higher Silhouette Coefficient score relates to a model with better defined clusters which are dense and well separated.
   - Execution time
   - Cross tabulation
     Cross table between clusters and class labels, which may not always works since clusters are unintuitive sometimes.

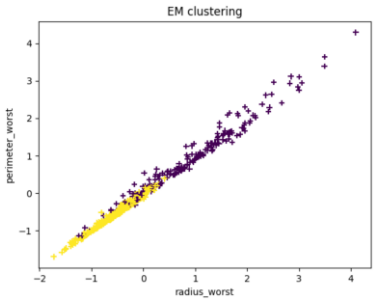


From the model selection and silhouette coefficient results above, I chose k = 2 for K-means and n = 2 with 'full' covariance_type for EM.

In this dataset, K-means performed better than EM, it likely that there are not much overlapping between classes and the shape of the dataset is "regular". While K-means didn't show much superiority over EM in execution time. If we use larger dataset or plot the average time of multiple run, we may get more information.



diagnosis	0	1
labels		
0	276	33
1	9	137



diagnosis	0	1
labels		
0	19	157
1	266	13

As showed in clustering scatter plot and cross table, both K-means and EM managed to form 2 interpretable clusters as expected.

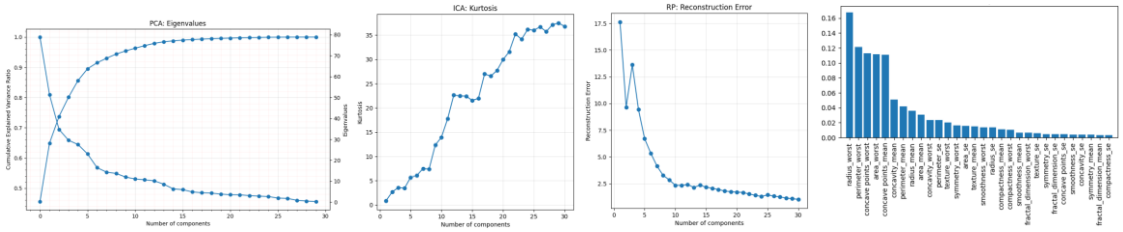
2.3. Dimensionality reduction

PCA, FastICA, Guassian Random Projection and Random Forest models are used in this experiment:

1. Standardize the samples.

Only for PCA. Since it is a variance maximizing exercise. It projects the original data onto directions which maximize the variance. Equal variance gives every feature equal chance. Random forests is a decision tree based model, by define it will not be affected by feature variances.

2. Determine the optimal number of components or features.
    - PCA: Elbow method using eigenvalues. And set the threshold of 0.85 as the desired variance ratio that is supposed to be explained by the principal components.
    - ICA: Kurtosis, which measures the degree of spikiness of a distribution and it is zero only for Gaussian distribution. Since ICA is separating non-Gaussian components so that they're independent to each other, a higher absolute value of kurtosis is preferred.
    - RP: Elbow method using reconstruction error, which measures the information loss made by projecting data points onto a lower-dimensional subspace.
    - RF: Elbow method using feature importance, set a threshold to do feature selection. Features whose importance is greater or equal are kept while the others are discarded.
3. Measure and compare performance.
    - Reconstruction error
    - Execution time



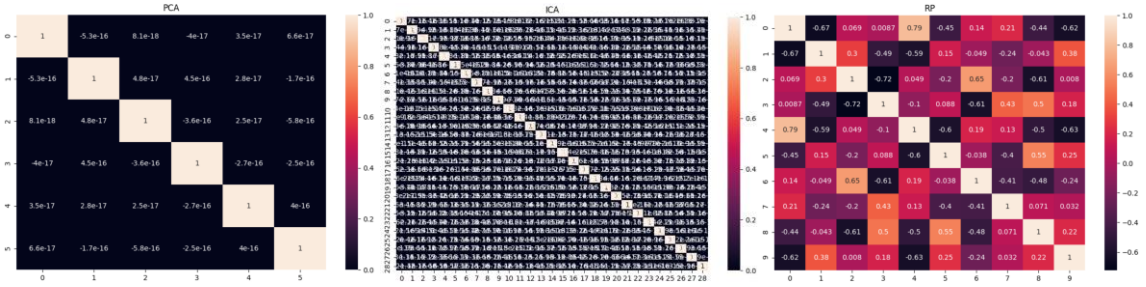
From the model selection results above, I chose n = 6 for PCA, n = 29 for ICA and n = 10 for RP. I set a threshold = 0.01 for RF to select 19 out of 30 features.

	Dimension	Reconstruction error	Execution time (ms)
PCA	6	0.105	7
ICA	29	4.33E-06	253
RP	10	2.35	1
RF	19	0.366	247

From the chart above:

- PCA performed the best with the smallest dimension as well as relatively smaller reconstruction error and shorter execution time.
- ICA found 29 non-Gaussian independent components in this 30 dimensional dataset! Although it preserved the largest amount of information, it runs too slow and the downstream estimators can not benefit much from dimensionality reduction. And independent components are not exactly the important components. So that ICA is not used for reducing dimensionality typically.
- Results for RP showed its character of trading a controlled amount of accuracy for faster processing times. Since it generates random directions, it may needs more components to capture correlations in the data compared to PCA. Considering the random nature of RP, an average results from multiple runs may better evaluate its performance.

- Feature selection using RF took much more time than others, because it has to learn and fit the dataset first. Moreover, RF uses a chain of estimators, the training time increased quadratically with complexity of the dataset. The reconstruction error is mainly depends on how many features are kept.

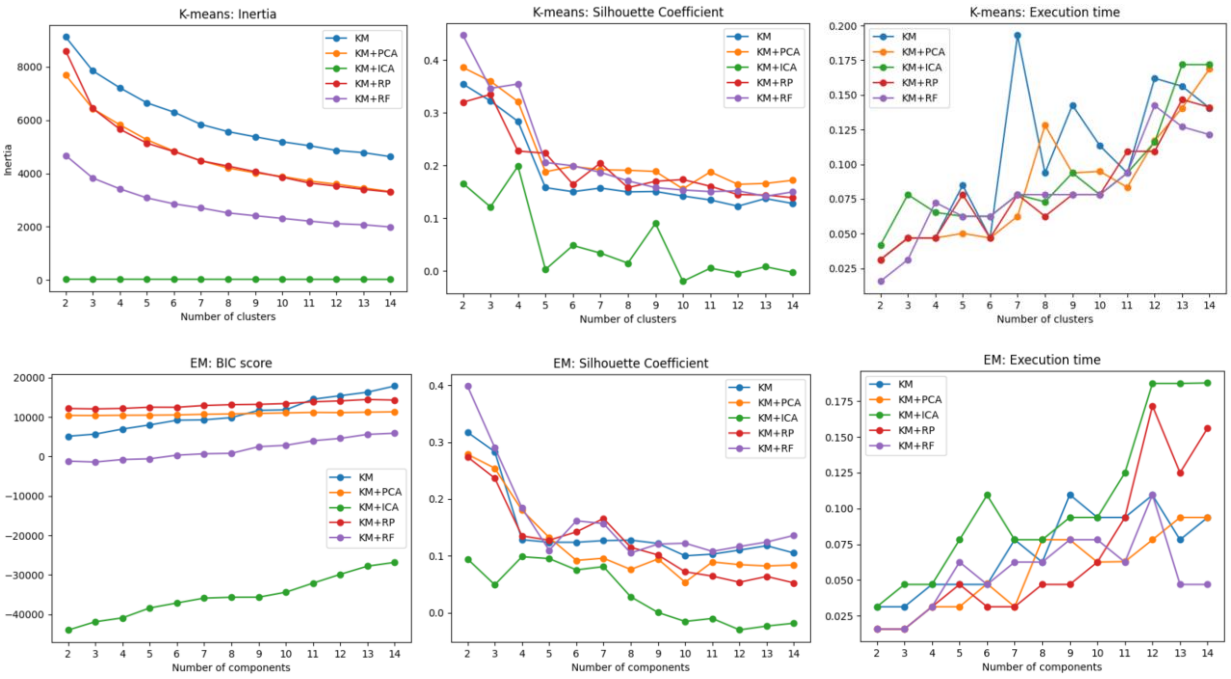


From the correlation heat map above, PCA and ICA can generate completely uncorrelated features. As for RP, since the components are randomly generated, no orthogonality or independence are guaranteed, but still better than original features.

2.4. Clustering after dimensionality reduction

I followed the same steps described in section 2.2, except:

- Replacing the original samples with outputs of dimensionality reduction.
- Treated the results of K-means and EM only as baseline.



The first row of figures are results of DR + K-means clustering, the second row are results of DR + EM clustering.

- Both KM and EM benefitted from RF with lower inertia/BIC score and higher silhouette coefficient. It may because RF kept the most interpretable information for clustering.

- As for ICA and PCA, high variance or high independence may not equivalent to high importance.
- Since RP controls the pairwise distances between any two samples of the dataset, it's supposed to be a suitable approximation technique for distance based method. Maybe an average results from multiple runs can shows it.
- Since Euclidean distances may become inflated in high-dimensional space, running a DR algorithm prior to clustering can speed up the computations. But I don't see much improvement in execution time results. It probably that the dataset is not complex enough or big enough to observe the difference.

2.5. Neural network with Dimensionality reduction and Clustering

The optimal dimensionality reduction (PCA) and clustering model (K-means) are used in this experiment:

1. Treat the outputs after dimensionality reduction or clustering as only features.
2. Standardize the inputs.
3. Tune hyper-parameters
   I used the 5-fold Grid search CV to tune the hyper-parameters and chose the combinations with the largest mean accuracy score.
4. Compare the performance with the results in Assignment 1.
   - Training/test accuracy
   - F1 and AUC score
   - Execution time

	Dimension	CV score	Train accuracy	Test accuracy	Train F1 score	Test F1 score	Train AUC score	Test AUC score	Time (s)
NN	30	0.978	0.991	0.982	0.988	0.976	0.996	0.998	0.691
DR + NN	6	0.978	0.991	0.965	0.988	0.951	0.998	0.987	0.047
Clustering + NN	2	0.934	0.938	0.93	0.918	0.907	0.985	0.973	0.251

From the chart above:

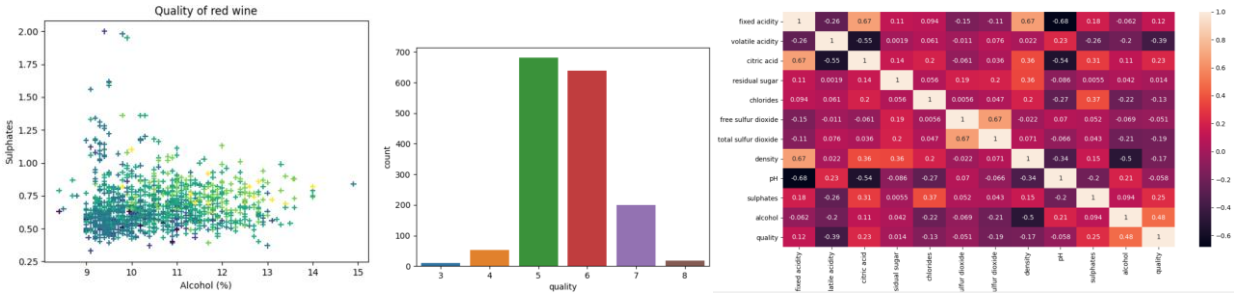
- The learning time decreased significantly after DR and clustering, especially DR. Both algorithms are proved to be capable of reducing computation and complexity of the model.
- With only 6 out of 30 features, neural network performed almost equally after DR, which shows that reasonable amount of information has been kept.
- While the performance degraded after clustering. It may because although k = 2 is the best model based on the definition of clustering, some relevant features has been discarded.
- Since the results of neural network alone is good enough, there are not much noises in this dataset. No obvious improvement are showed in avoiding overfitting.
- In practice, it's better to treat the number of clusters or components as hyper-parameters, use the grid search to choose the best model.

3. Dataset 2

3.1. Description

The dataset I used is red variant of the Portuguese "Vinho Verde" wine appraisal samples. The goal is to model wine quality based on physicochemical tests. There are 1599 samples with 11 features. The target is quality scoring between 0 and 10 (only 3 to 8 samples in this dataset).

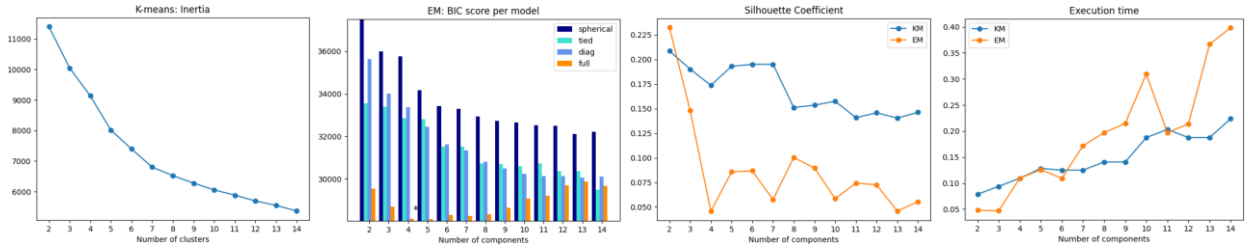
I chose two dominant features, which I got from random forest learner, to visualize the dataset, as well as the class distribution and feature correlation heat map:



Based on the overlapping output classes, we may not get intuitive clusters after clustering. From the results in correlation heat map, there isn't much room for dimensionality reduction compared to dataset 1.

The steps for the following experiments are the same as described in section 2.

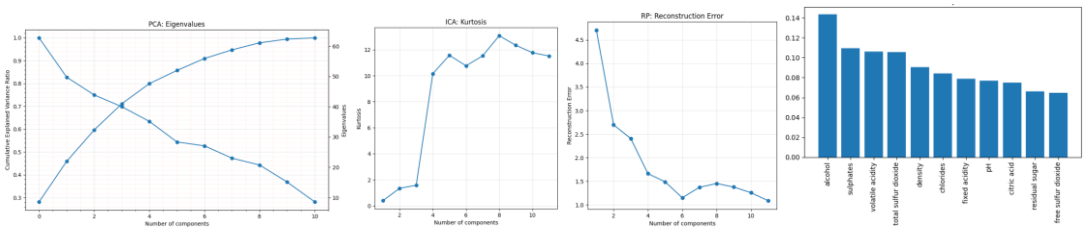
3.2. Clustering



From the model selection and silhouette coefficient results above, I chose k=7 for K-means and n=4 with 'full' covariance_type for EM.

Classes in this dataset overlap a lot. K-means still outperformed EM in silhouette coefficient. It may because that silhouette coefficient is generally higher for convex clusters than other concepts of clusters. We may use other metrics to evaluate their performance. And since the dataset is larger, EM converged slower than K-means compared to result of dataset 1.

3.3. Dimensionality reduction

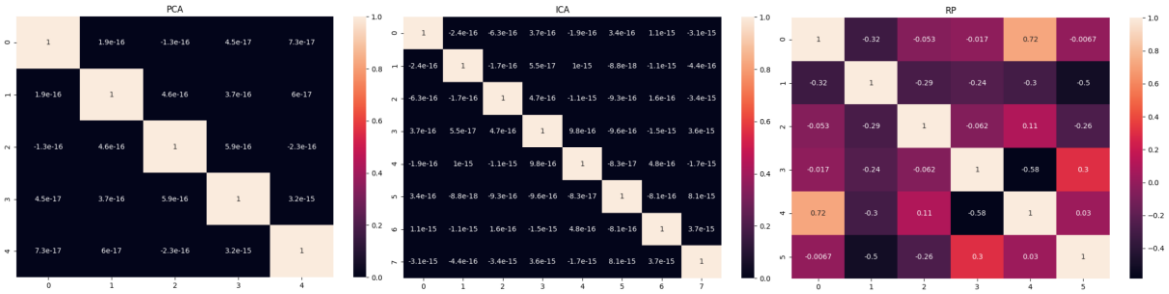


From the model selection results above, I chose n = 5 for PCA, n = 8 for ICA and n = 6 for RP. I set a threshold = 0.07 for RF to select 9 out of 11 features.

	Dimension	Reconstruction error	Time (ms)
PCA	5	0.199	16
ICA	8	5.20E-02	16
RP	6	1.15	less than 1
RF	9	0.177	531

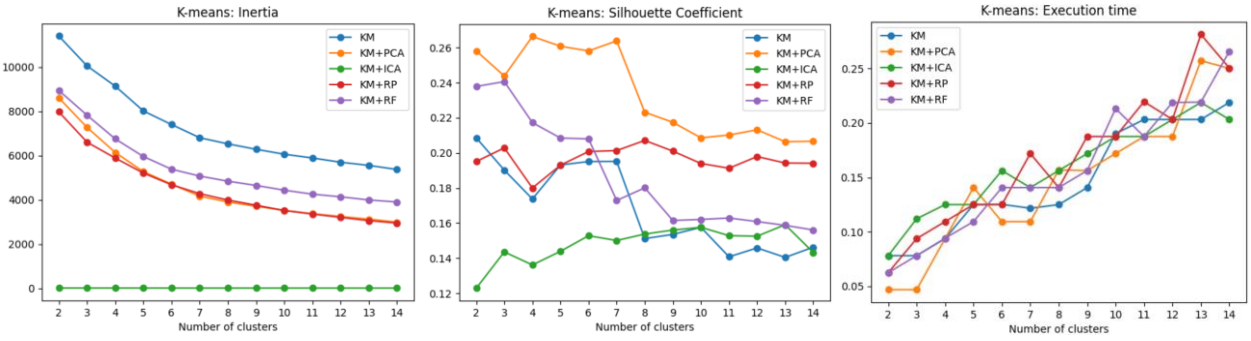
From the chart above:

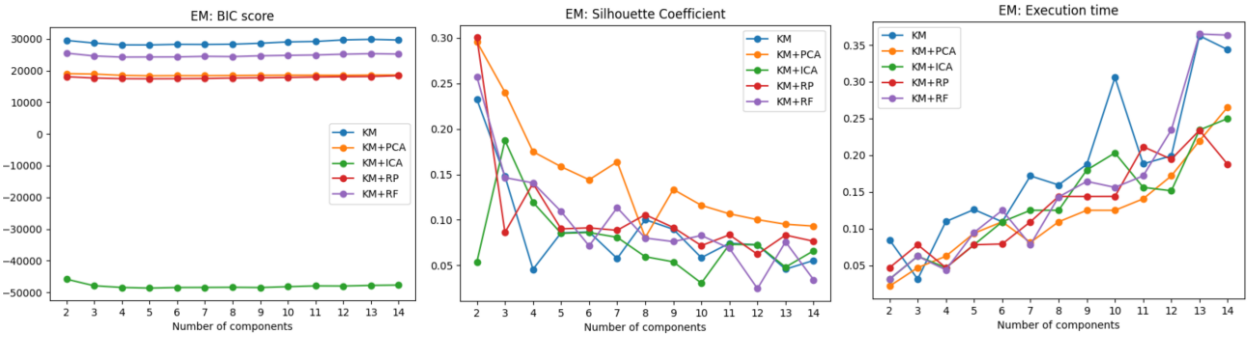
- PCA still performed the best with the smallest dimension as well as reasonable reconstruction error and execution time.
- ICA excelled PCA in reconstruction error. It may because there are higher-order correlation in this dataset and non-linear transformation are needed. It can better explains the dataset in this case.
- RP showed its advantage of computational efficiency again.
- Feature selection using RF took much more time than others. As mentioned in section 2.3, the processing time increased significantly with larger dataset compared to the results of dataset 1.



The correlation heat map before and after DR shows the same characteristics as described in section 2.3.

3.4. Clustering after dimensionality reduction





The first row of figures are results of DR + K-means clustering, the second row are results of DR + EM clustering.

- Both KM and EM benefitted from PCA and RP with lower inertia/BIC score and higher silhouette coefficient.
- Since there are not much correlation in this dataset and both features have relatively high importance, they all should be kept for RF. It may explains why RF's performance degraded.
- Results of ICA are the worst again. This probably why ICA is not used for reducing dimensionality usually.
- It took EM less time after DR, which shows the advantage of DR in reducing computation.

4. Conclusions

With clustering and dimensionality reduction techniques, downstream estimators can learn quickly and effectively with minimal datasets. They are effective ways to avoid the curse of dimension.

Reference

https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

https://scikit-learn.org/stable/modules/unsupervised_reduction.html