CS 7641 Machine Learning - Assignment 3

student: Xinru Lu - xlu320 - xlu320@gatech.edu

The repository is accessible at: https://github.com/Lorraine97/ML_2022/tree/main/project_1

1. Abstract

This assignment will focus on 2 clustering methods in unsupervised learning, 4 dimension reduction techniques, and their performance on the two datasets we studied before.

2. Introduction

2 clustering methods to be tested and compared:

- k-means clustering
- Expectation Maximization

4 dimensionality reduction algorithms:

1. PCA
2. ICA
3. Randomized Projections
4. Tree-based feature selection (my choice of feature selection algorithm)

For the above algorithms, I will first apply each of them to the two datasets, and then combine one clustering method with one dimensionality reduction algorithm (for each out of the above algorithms) to both of the datasets. Then, I will apply the 4 dimensionality reduction algorithms to one of the two datasets and train with Neural Network model. From there, I will apply the clustering algorithms first to get new data (from clustering), then train with Neural Network mode. The main purpose of the experiment is to explore and compare the performance of these methods, along with comparing with previously implemented models.

3. Dataset (Copied from project 1)

For this project, I selected the following datasets:

- Check Loan Eligibility
- Student Performance (Math only)

The loan eligibility data set is a set of processed data for modeling the eligibility check, based on features like gender, education, income, loan amount, credit history, etc. It consists of 12 columns, of which 10 are integer columns (binary) and 2 are decimal columns. It has a clear `y` column named `Loan_Status` (detailed information can be found in the dataset documentation). I have evaluated it from a ethical perspective in another course at OMSCS (AI Ethics) and it is an interesting data set to evaluate historical biases as well. Therefore, I would like to use it for the ML algorithms and see if different algorithms will induce different biases towards different groups.

The student performance data set was drawn from student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features (detailed information can be found in the dataset documentation) and it was collected by using school reports and questionnaires. I have only selected the data of the Math subject. The data was specifically modeled under binary/five-level classification and regression tasks. Interesting point about this data set is that it includes three potential `y` columns: G1, G2, G3, corresponding to 1st, 2nd and 3rd period grades. However, the G2 is dependent on G1, and G3 is dependent on both G1 and G2. I find that it could be useful for Bayesian analysis in the future. Therefore, I would like to use it for the ML course as it is versatile and easy to implement for multiple models.

4. Method and Individual Analysis

The implementation methods will be briefly introduced in the following section, but many details will be skipped since it is available on `sklearn` documentation page. For consistency, all `random_state` parameters will bet set to be 123.

Evaluation methods of the following will be performed (implementation adopted from `sklearn.metrics`):

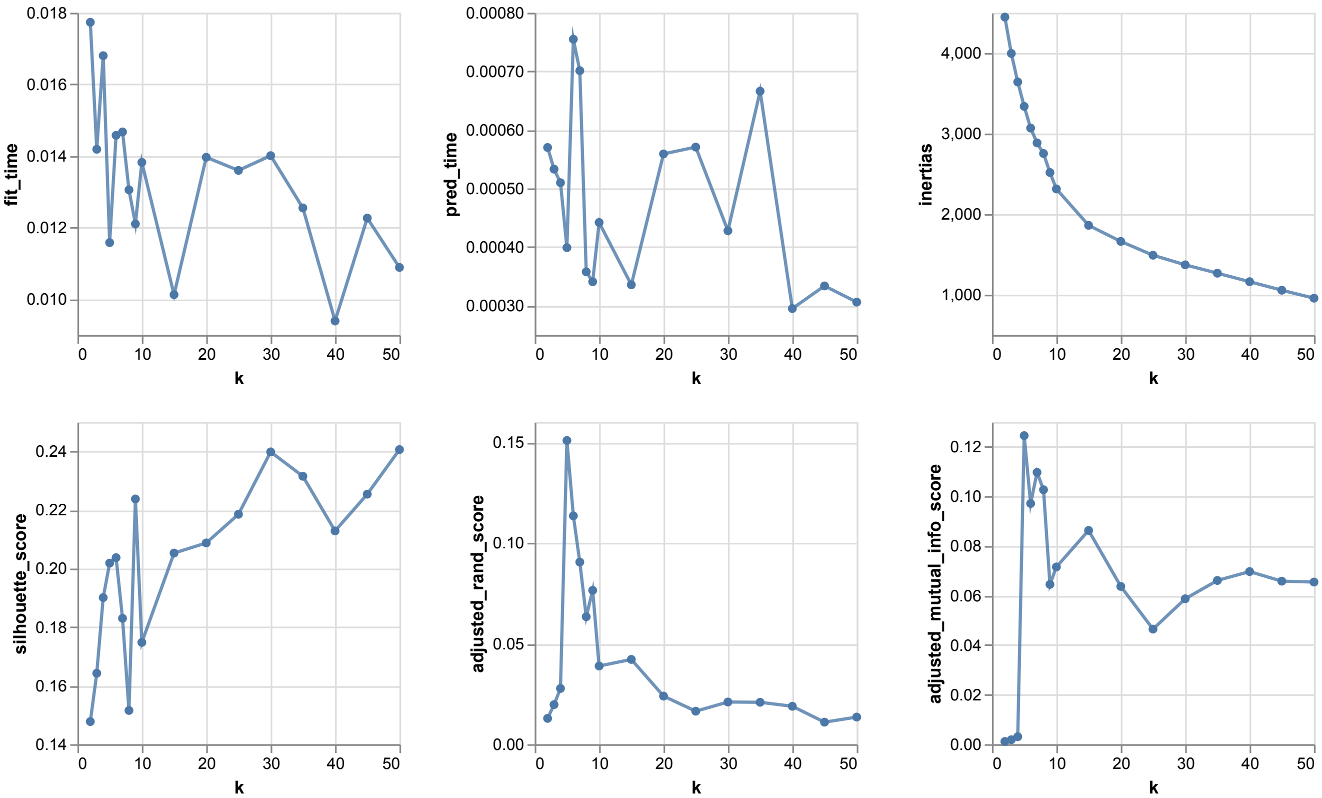
1. Silhouette Score: determine the degree of separation between clusters
2. Adjusted Rand Index: computes a similarity measure between actual labels from the dataset and the predicted labels from the clustering results
3. Mutual Information based scores: a function that measures the agreement of the two assignments, ignoring permutations

4.1 k-means Clustering

I selected the implemented algorithm from `sklearn.cluster.KMeans`, which clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares (the distance metric here). By default, it uses Lloyd's algorithm to minimize and re-center. It would only work with numeric values, therefore pre-processing for the datasets is needed.

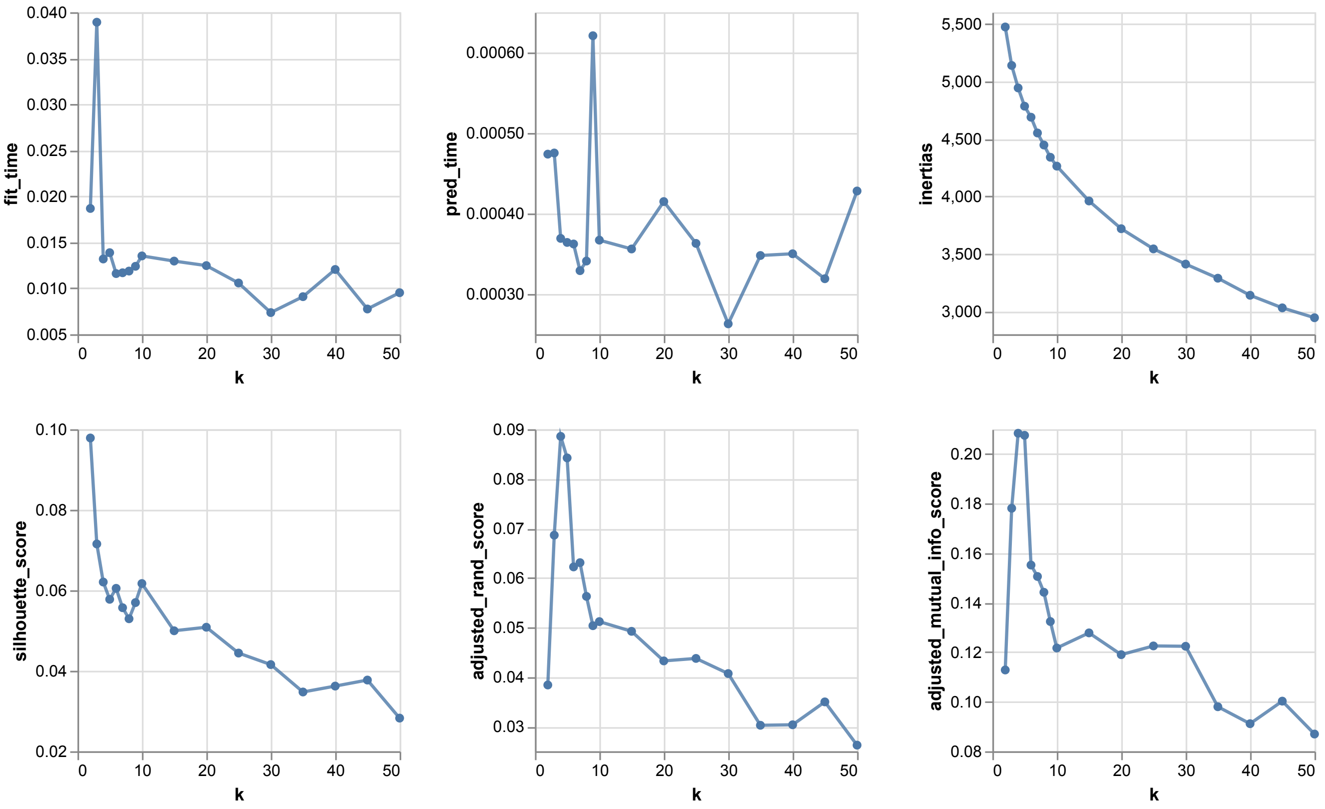
In the experiment, I aim to find the k value to maximize the adjusted rand index and mutual information based scores when comparing the cluster results with the known labels. In this way, it would help to indicate the success of clustering on the multiple dimension data sets.

Loan Dataset



From the above chart, we observe that the fitting time and prediction time for k-means model does not change much with different k values. As k increase, the inertias of the k-means model keeps decreasing. This says that as k increase, the dataset is worse clustered by K-Means. Therefore, we should choose a smaller k to best fit the dataset. From the bottom two charts, it shows that best k value to maximize the two scores of interest will be **k = 5**. From the silhouette chart, it shows that at k = 5, it reaches a local maxima. Therefore, I think the results of the different metrics generally agree with each other.

Math Dataset

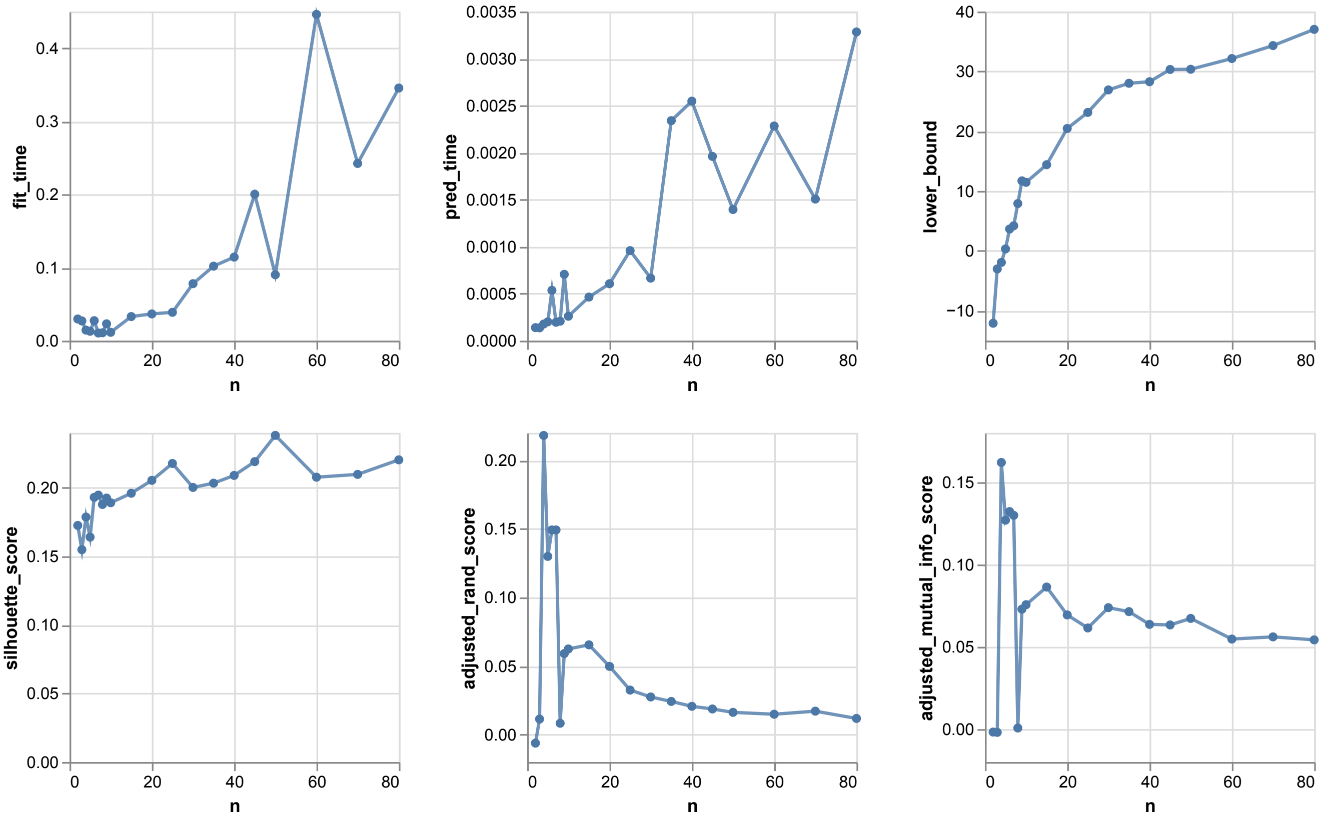


From the above chart, we observe that the fitting time and prediction time for k-means model does not change much with different k values for the math dataset either. As k increase, the inertias of the k-means model keeps decreasing. This says that as k increase, the dataset is worse clustered by K-Means. Therefore, we should choose a smaller k to best fit the dataset. From the bottom two charts, it shows that best k value to maximize the two scores of interest will be **k = 4**. From the silhouette chart, it shows that at k = 4, although it is not a maxima, it is still of high value in the entire chart. Therefore, I think the results of the different metrics generally agree with each other.

4.2 Expectation Maximization

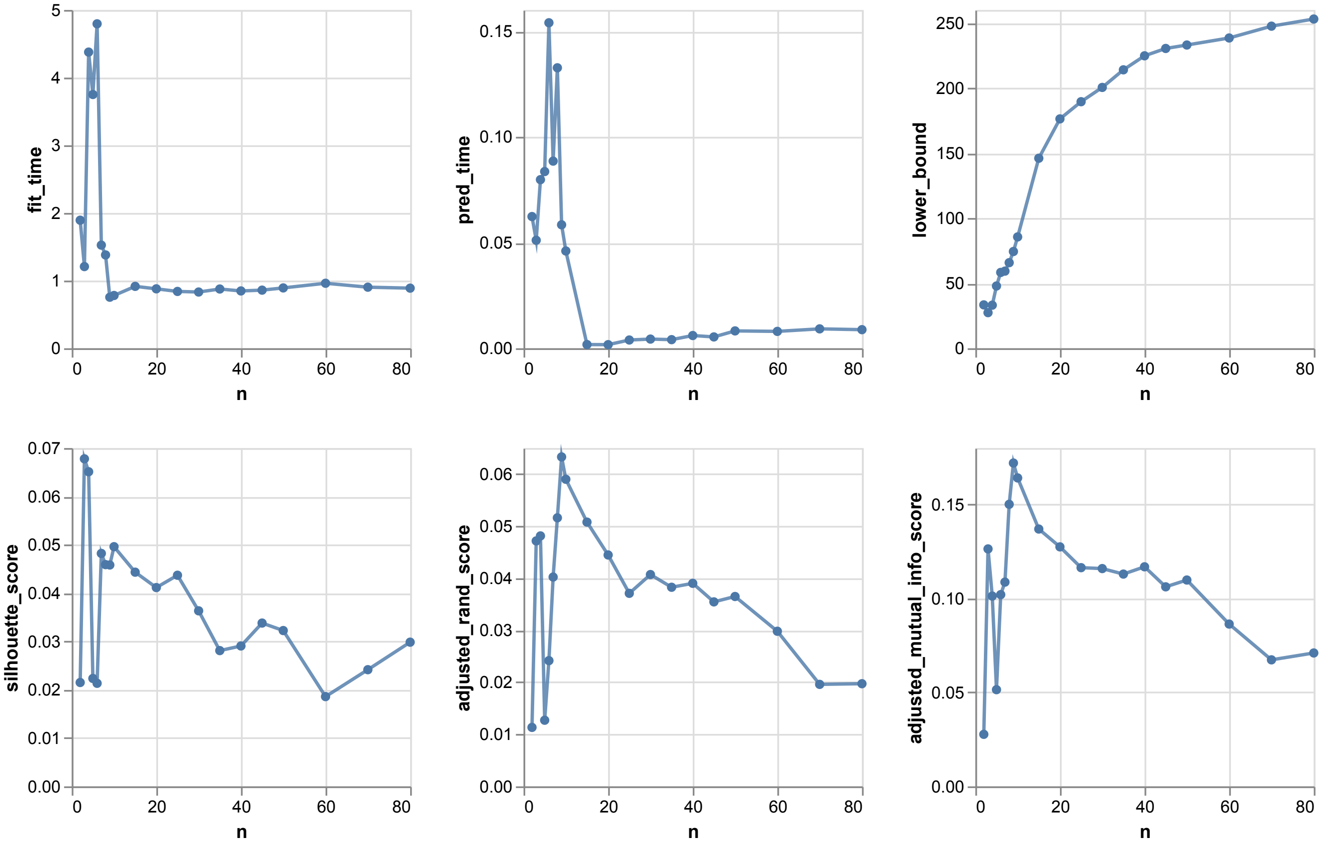
I selected the implemented algorithm from `sklearn.mixture`, which enables one to learn Gaussian Mixture Models (diagonal, spherical, tied and full covariance matrices supported), sample them, and estimate them from data. I will use the default method `covariance_type=full` so that each component has its own general covariance matrix. I plan to use `init_params=kmeans` so that it uses k-means clustering to initialize the weights, the means and the precisions. It would only work with numeric values, therefore pre-processing for the datasets is needed.

Loan Dataset



From the above chart, we observe that the fitting time and prediction time for k-means model does not change much with different k values. From the bottom two charts, it shows that best n value to maximize the two scores of interest will be **n = 4**. From the silhouette chart, it shows that at n = 4, it reaches a local maxima. Therefore, I think the results of the different metrics generally agree with each other.

Math Dataset

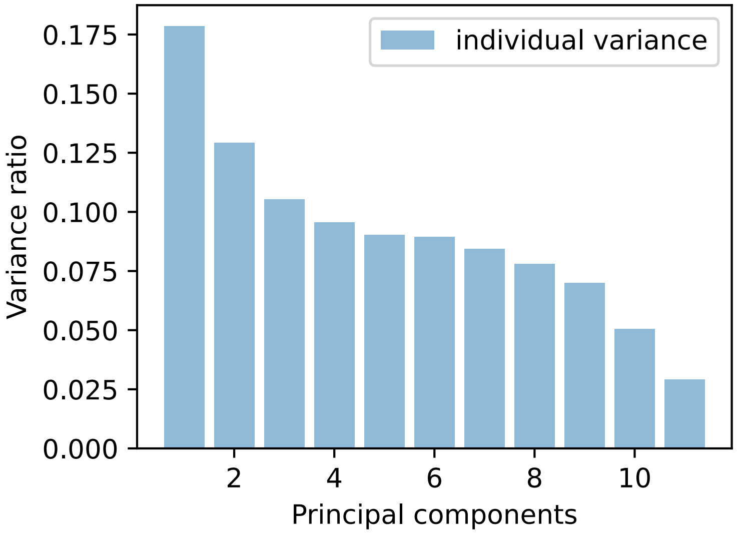


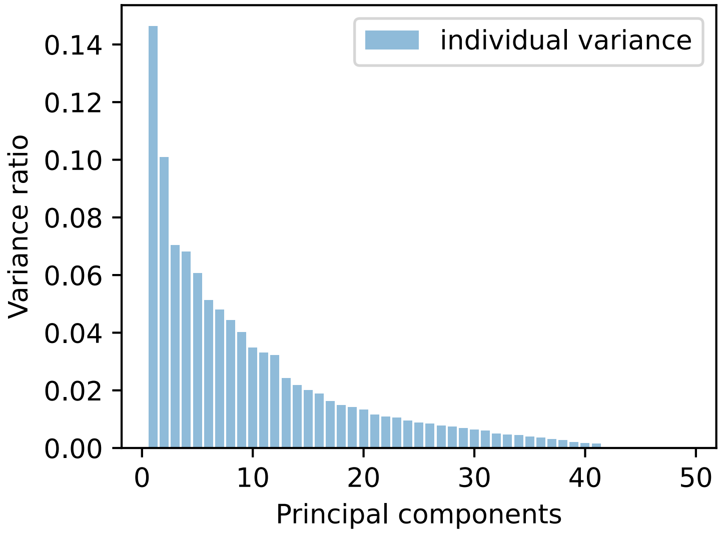
From the above chart, we observe that the best n value to maximize the two scores of interest will be **n = 9**. From the silhouette chart, it shows that at n = 9, although it is not a maxima, it is still of high value in the entire chart. Therefore, I think the results of the different metrics generally agree with each other.

4.3 PCA

I selected the implemented algorithm from `sklearn.decomposition.PCA`. It uses linear dimensionality reduction with Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD. We will see the transformed data structure and apply the transformed data to the clustering methods.

In the experiment, I grabbed the variance ratios from different dimensions of the PCA models, and achieved the following plot:





From the above chart, it is observed that for the loan dataset (on the left), with **9 features**, it can cover roughly 90% of the data variance, while for math dataset (on the right), with **30 features**, it can cover roughly 90% of the data variance. We will keep these numbers to process data for the clustering algorithms.

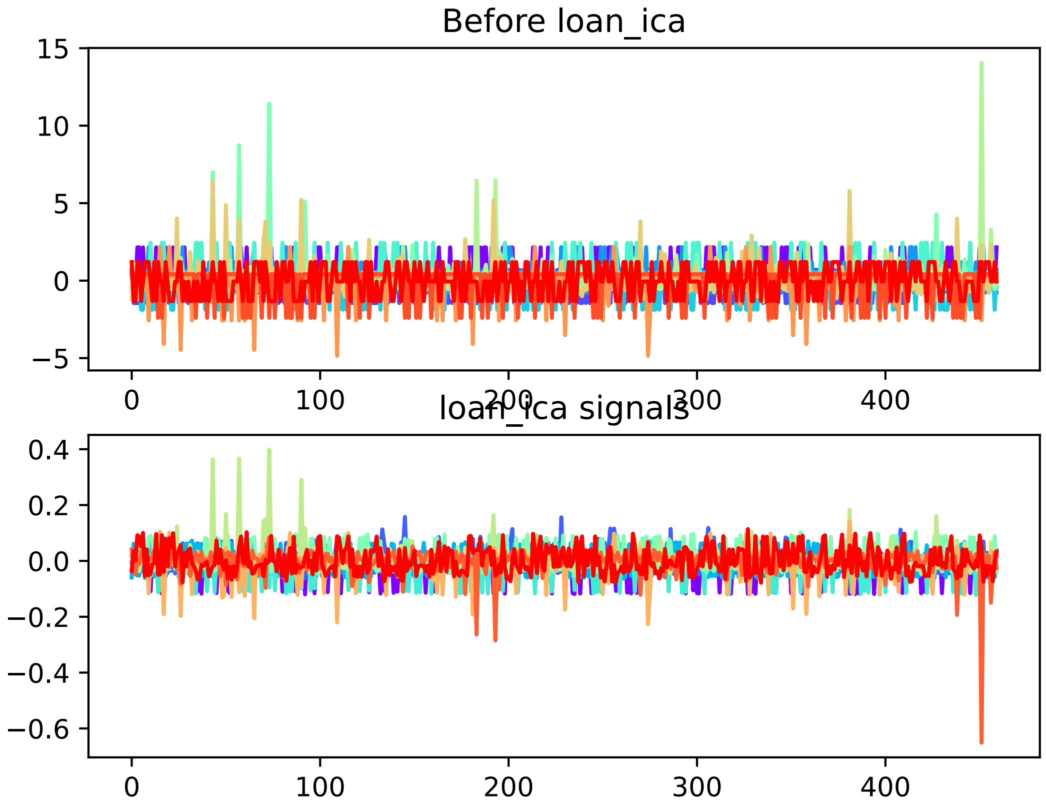
4.4 ICA

I selected the implemented algorithm from `sklearn.decomposition.FastICA`. Its purpose is to separate a multivariate signal into additive subcomponents that are maximally independent.

Since there is no good metric to measure the best number of subcomponents to separate out the signals, I will adopt from the PCA results for both datasets (9 for loan, 30 for math).

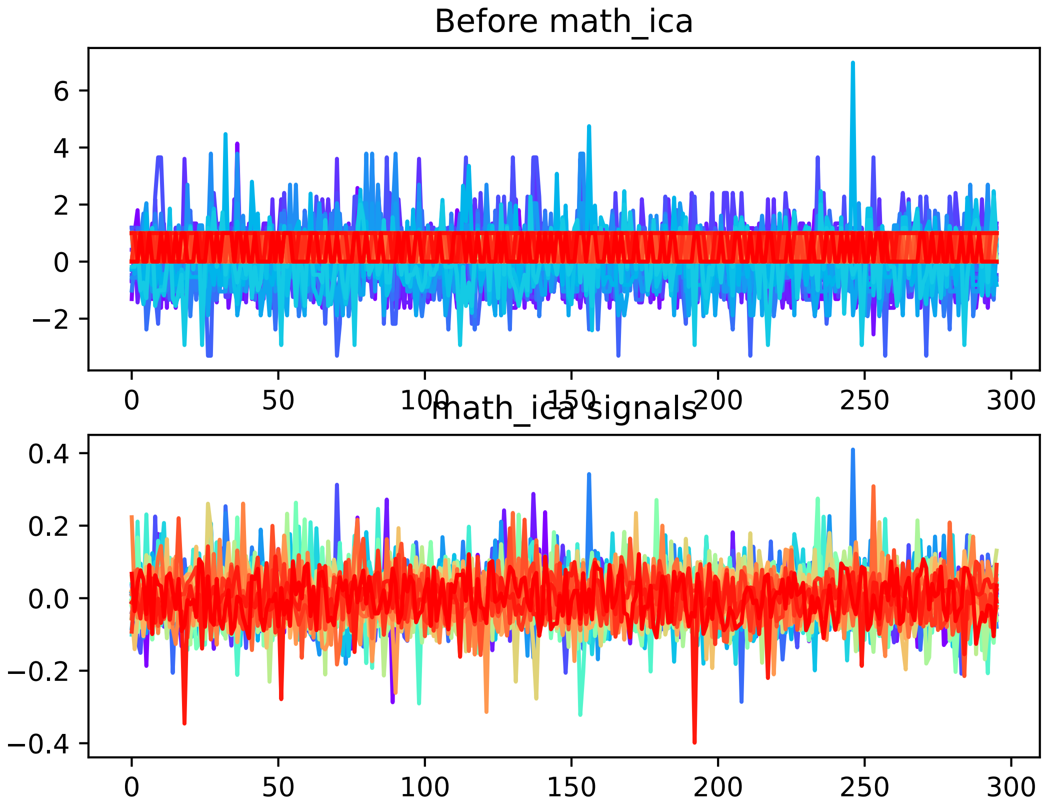
From the experiment, it is observed that the shape of the signals has changed after the ICA transformation. Here I attached the signal chart, colored by different columns of input data.

Loan Dataset



The change is not significant, but it is clear that the variance for each colored line has increased and the boundaries for the lines seem to be set free to increase variance.

Math Dataset



The change here is clear to see: the redish band in the middle, which represent binary values, are now transformed to a changing line along with other multi-class or numerical data.

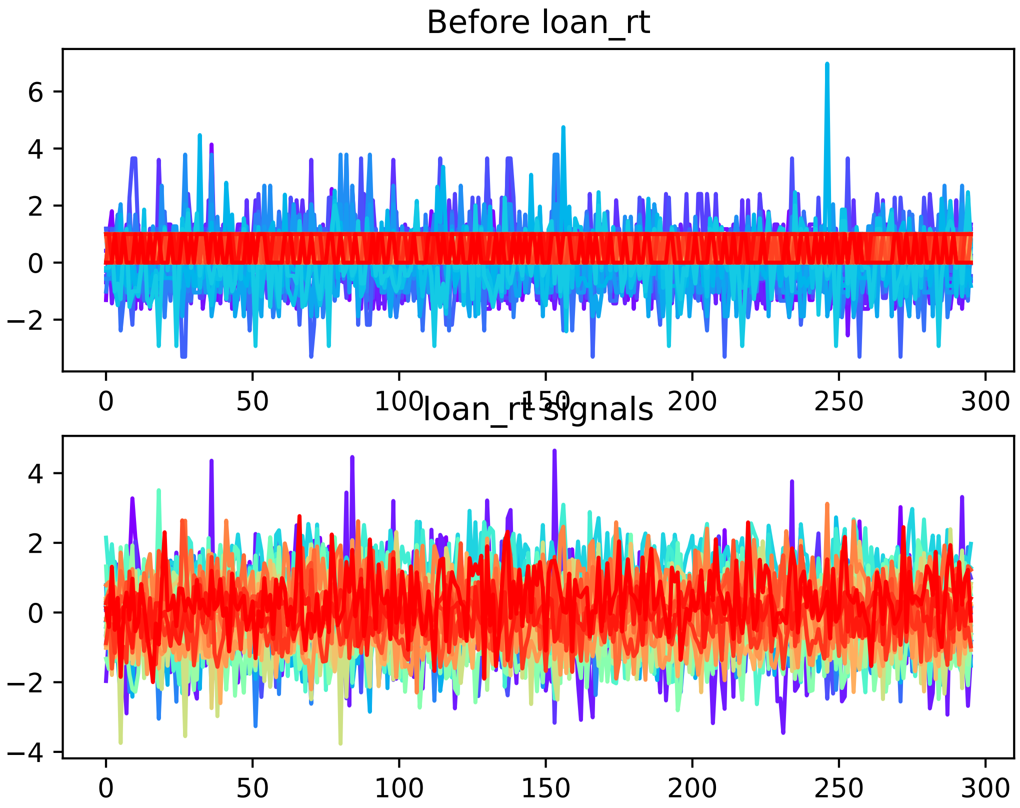
4.5 Randomized Projections

I selected the implemented algorithm from `sklearn.random_projection`. For the same reason as above, I choose to use the PCA results for both datasets (9 for loan, 30 for math) for the number of subcomponents.

From the experiment, it is observed that the shape of the signals has changed after the Randomized Projections. Here I attached the signal chart, colored by different columns of input data.

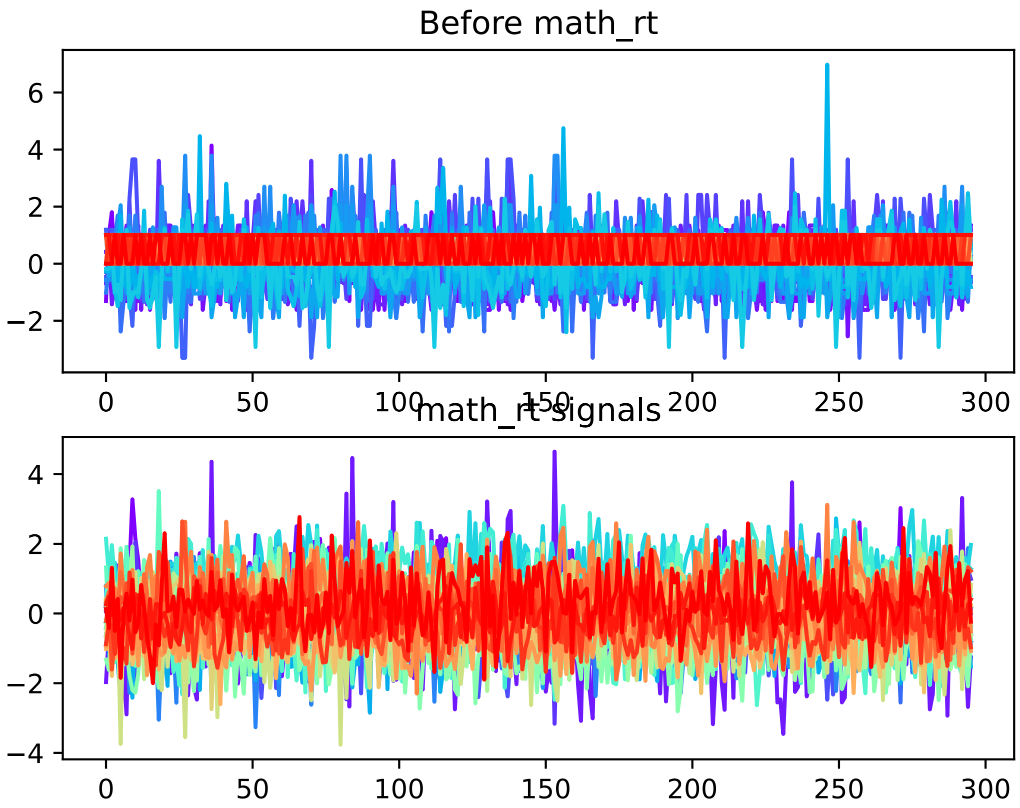
Compared to ICA, both datasets' values are more significantly different from the previous distribution. Especially for the loan dataset, the redish lines that were binary data are projected better in this model.

Loan Dataset


Before loan_rt

loan_rt signals

Math Dataset



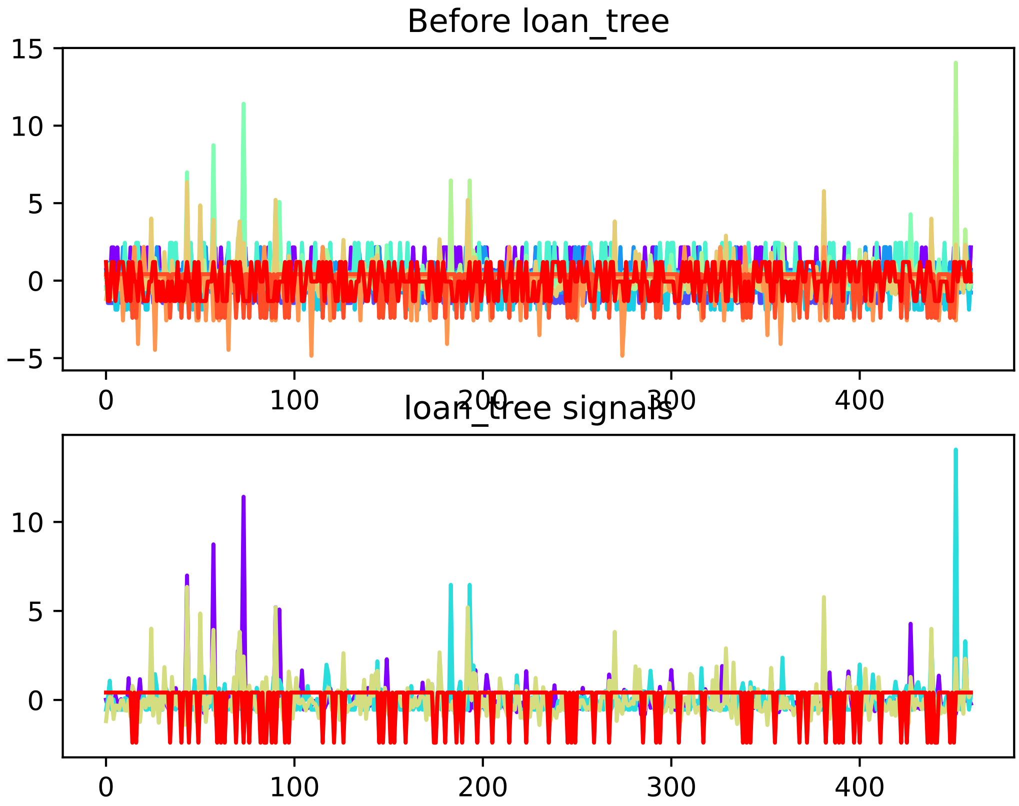
4.6 Tree-based feature selection

To implement the algorithm, I referred the documentation and used both `sklearn.ensemble.ExtraTreesClassifier` and `sklearn.feature_selection.SelectFromModel` methods.

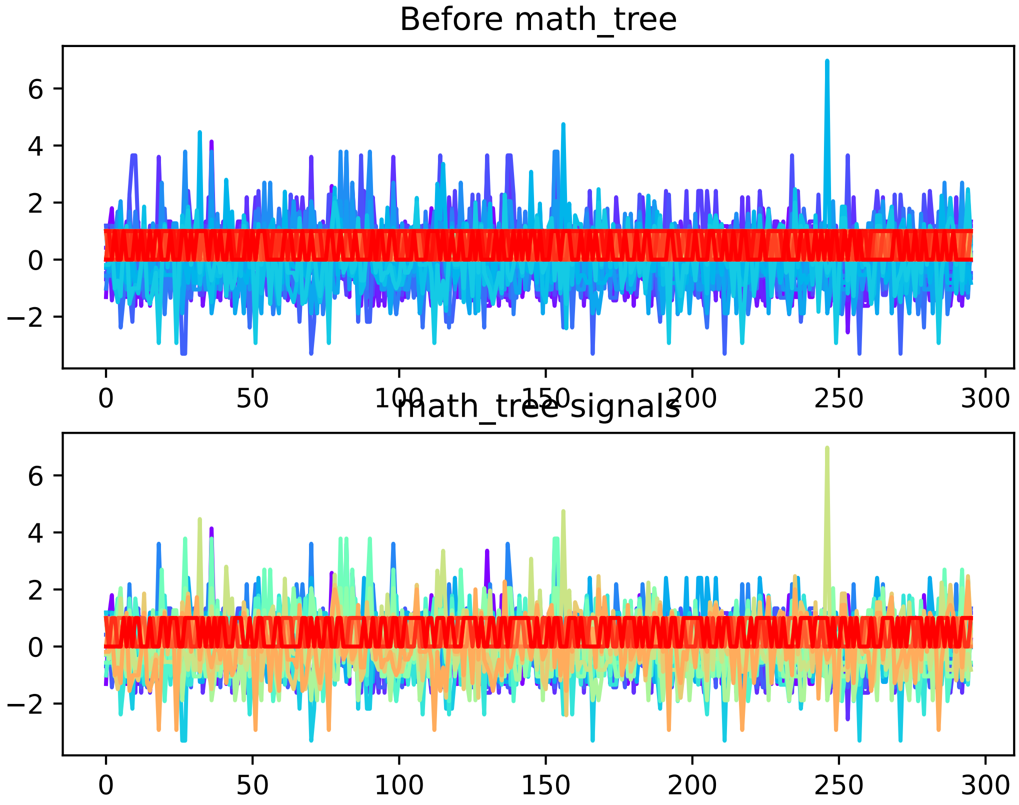
From the experiment, it is observed that the shape of the signals remained roughly the same, only layers have been removed. Here I attached the signal chart, colored by different columns of input data.

Compared to all previous transformations, here the data has been filtered, instead of transformed, based on the relevance to the results. Therefore, we observe a significant color reduction (reduction of feature lines) from the original to the resulting graph, for both of the datasets.

Loan Dataset



Math Dataset



5. Clustering on Datasets of Reduced Dimensions

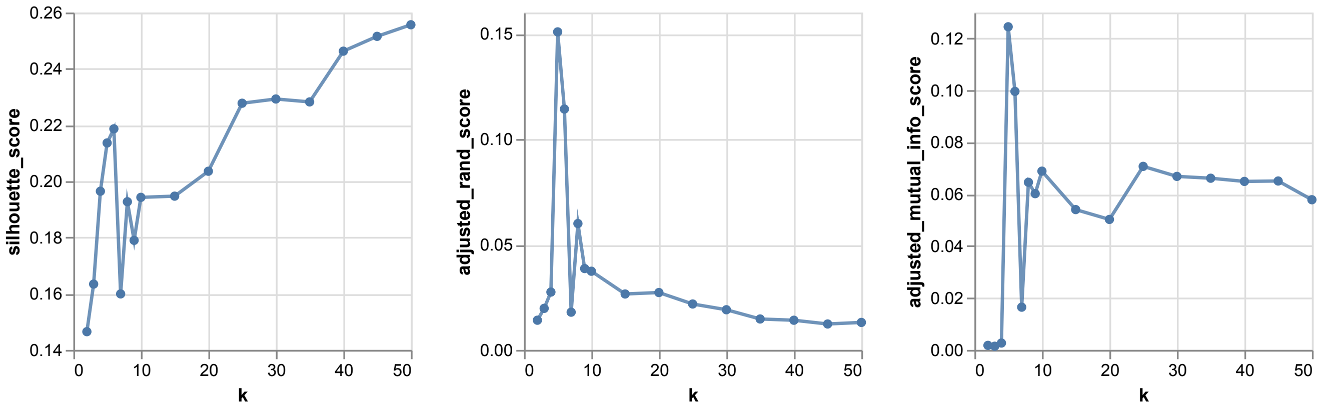
In the following section, I would show the results of the clustering algorithms on the data sets that were transformed by the above 4 algorithms. In total, 16 analysis will be conducted.

5.1. k-means

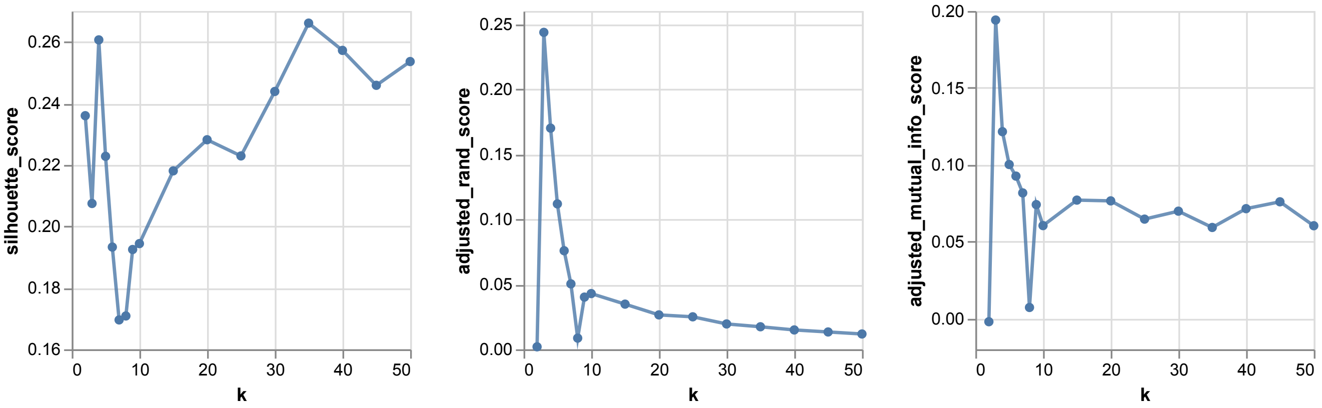
Loan Dataset

From previous k-means experiment, we learn that the best k value to maximize the clustering is: k = 5.

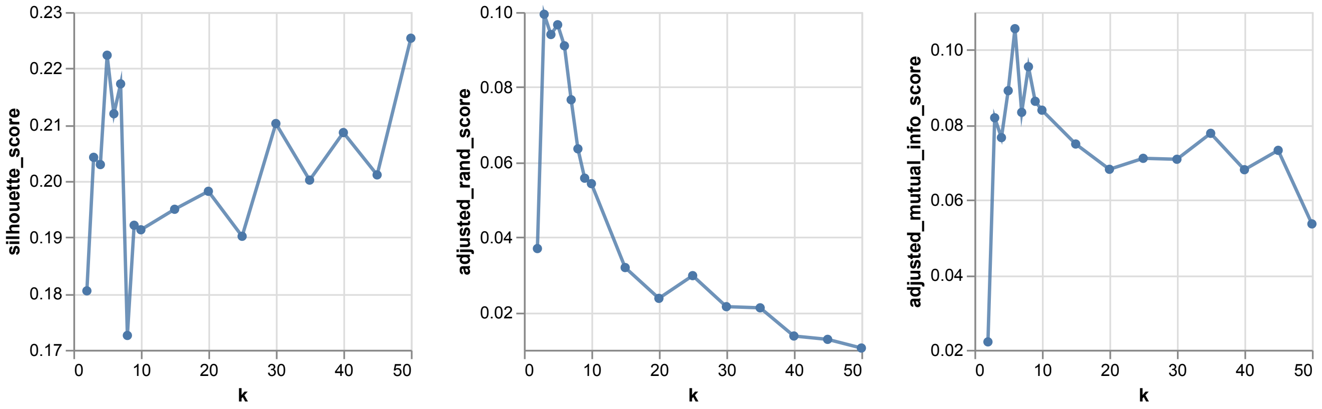
1. PCA



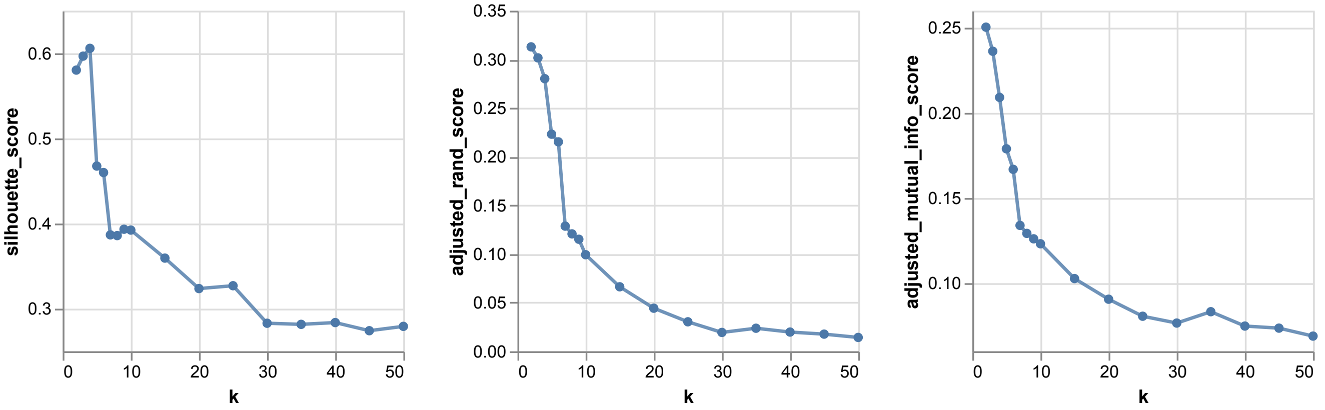
1. ICA



1. Random Projection



1. Tree Selection



From the above chart, it is observed that the optimal k values have changed:

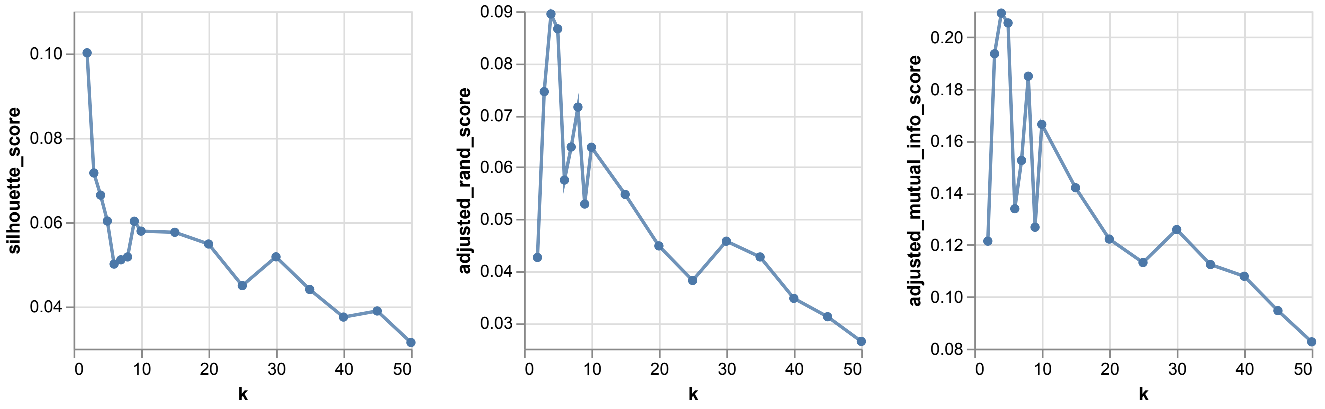
- PCA: k = 4
- ICA: k = 3
- Random Projection: k = 3
- Tree Selection: k = 2

In addition, compared to the original data and transformed data from PCA and Random Projection, it seems that ICA and Tree Selection have significantly improved the performance of the k-means model. The overall two adjusted rand index scores and mutual information scores have increased to nearly double as high, along with a much larger silhouette scores. This suggests that ICA and Tree Selection have effectively removed some noise from the original dataset that might have led to previously low performance of the k-means model. Especially with the Tree Selection algorithm, the clusters are significantly better grouped together if in the same cluster, and farther away from other groups. Overall, it seems the Tree Selection is the best transformation method here to help optimizing the k-means model for the loan dataset.

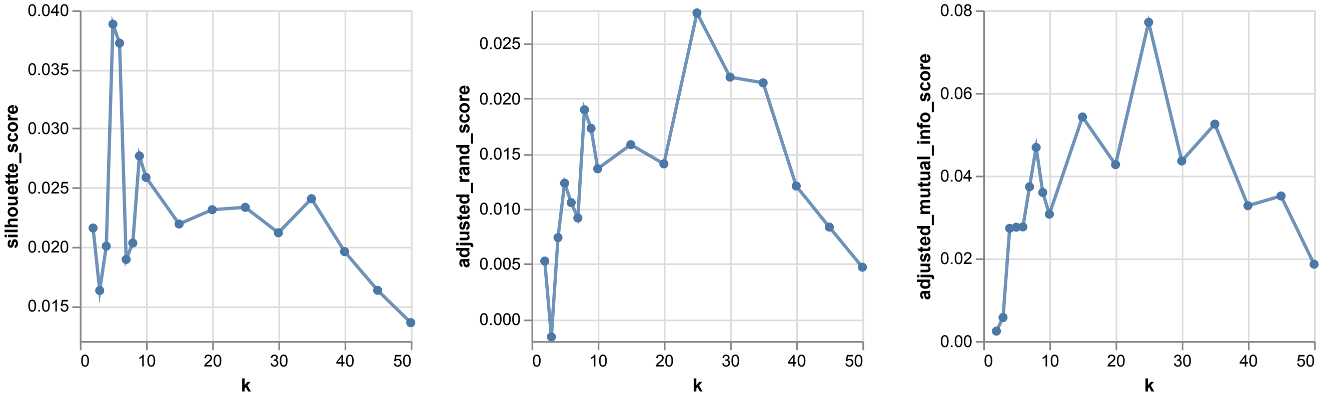
Math Dataset

From previous k-means experiment, we learn that the best k value to maximize the clustering is: k = 4.

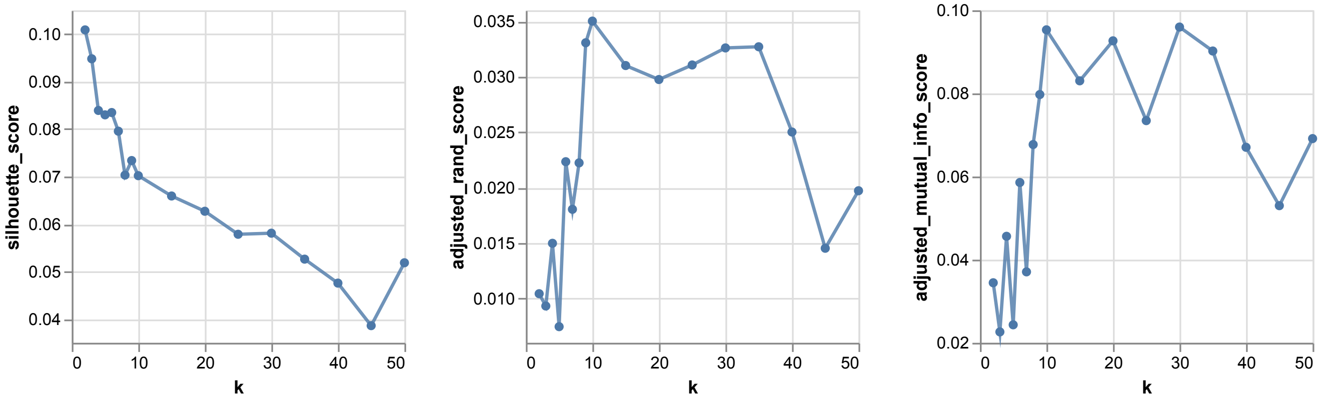
1. PCA



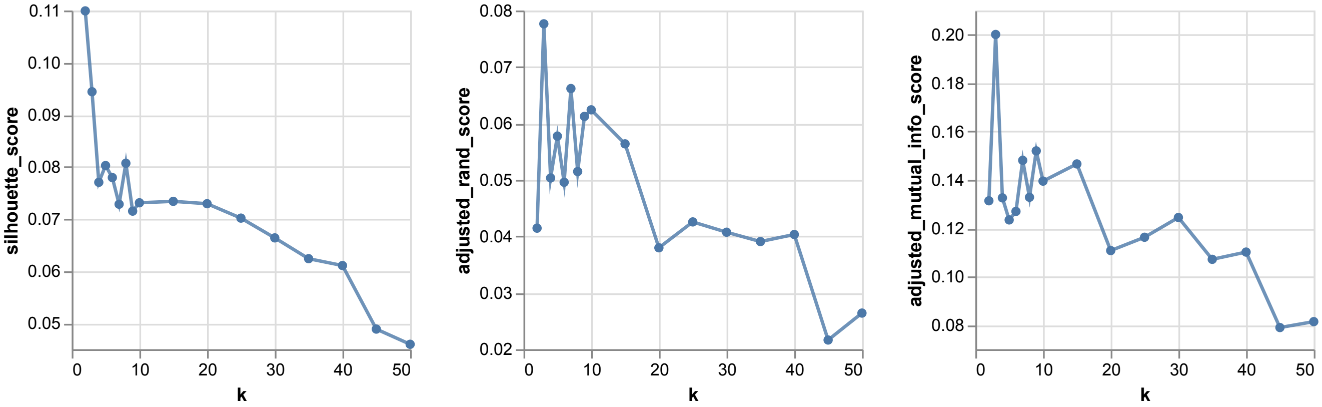
1. ICA



1. Random Projection



1. Tree Selection



From the above chart, it is observed that the optimal k values have changed:

- PCA: k = 4 (same as before)
- ICA: k = 25
- Random Projection: k = 10
- Tree Selection: k = 3

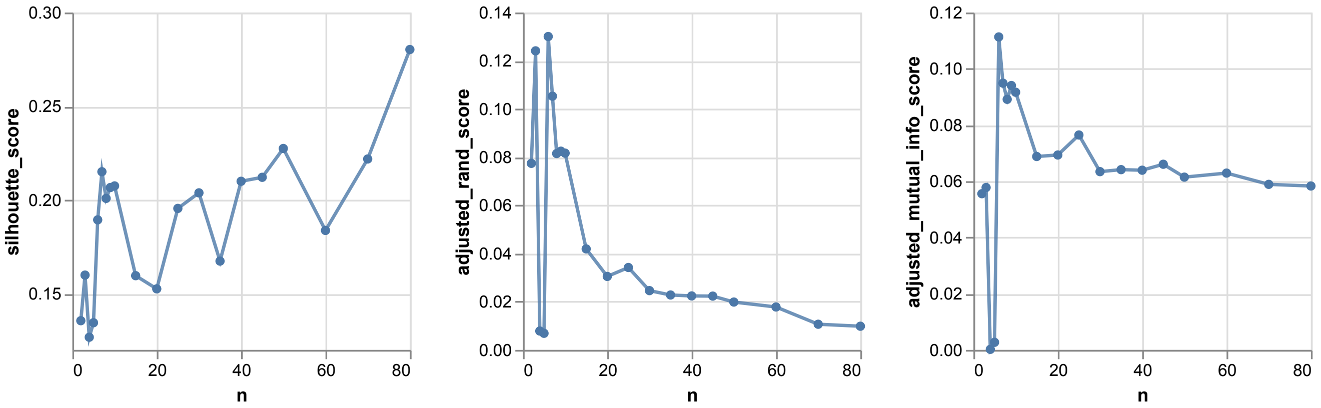
In addition, compared to the original data, it seems that PCA, Random Projection and Tree Selection have all slightly improved the silhouette scores. Among them, both Tree Selection and PCA have slightly improved the adjusted rand index scores and mutual information scores, while Random Projection has decreased scores. Overall, it seems that PCA and Tree Selection are good to transform the math dataset to optimize the k-means model.

5.2. Expectation Maximization

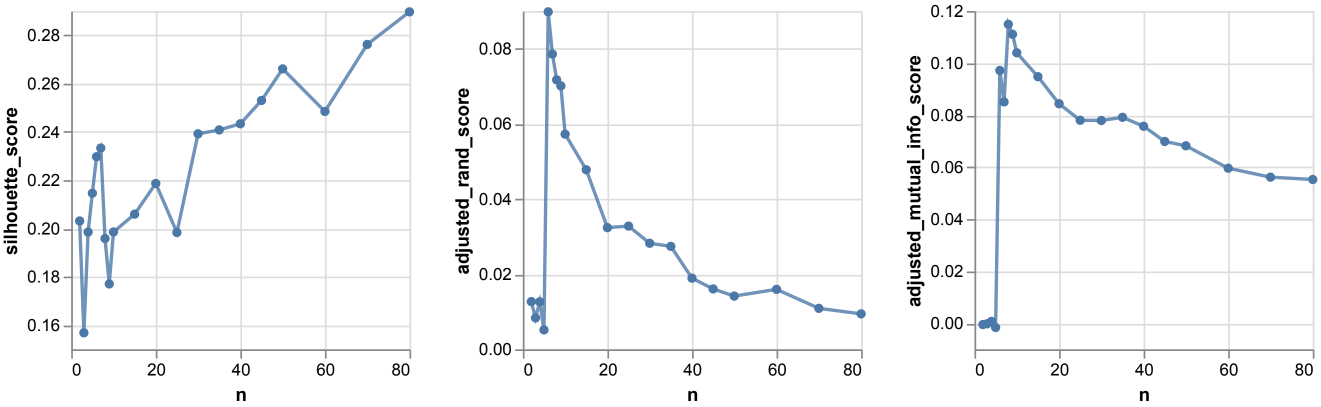
Loan Dataset

From previous EM experiment, we learn that the best n value to maximize the clustering is: n = 4.

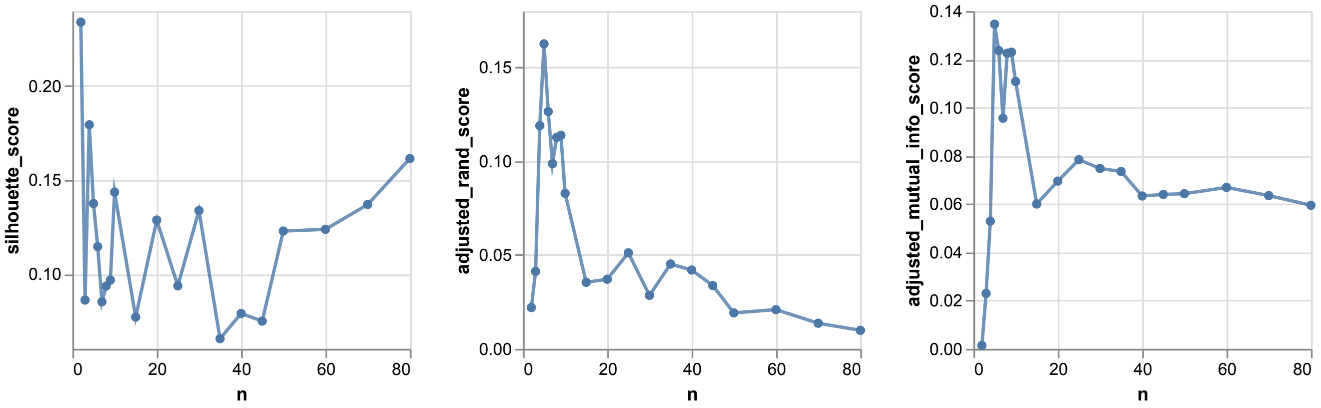
1. PCA



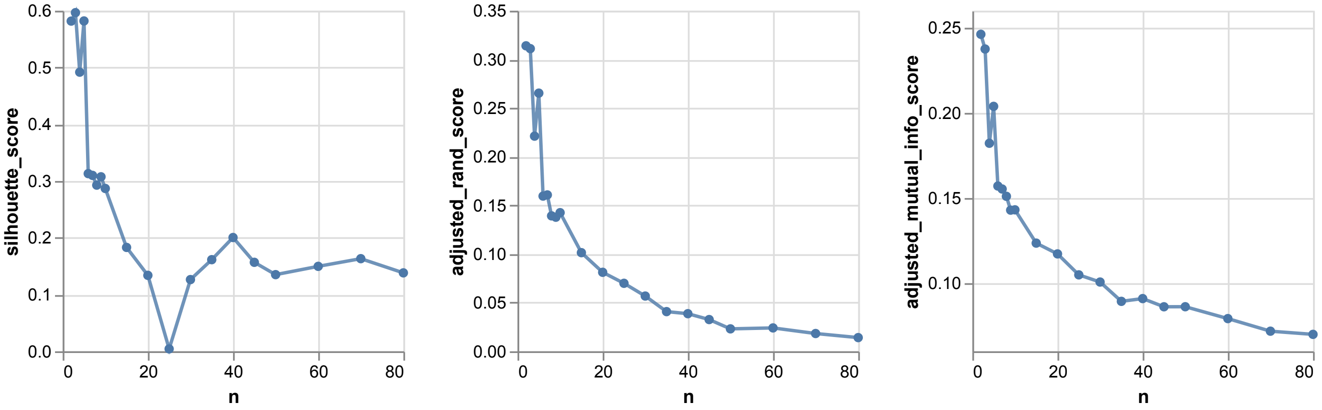
1. ICA



1. Random Projection



1. Tree Selection



From the above chart, it is observed that the optimal n values have changed:

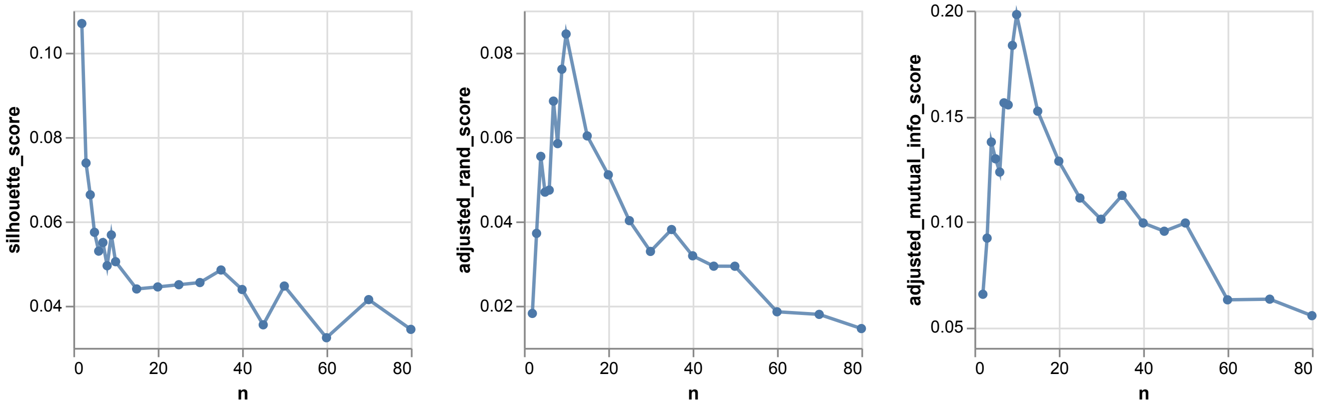
- PCA: n = 6
- ICA: n = 6
- Random Projection: n = 5
- Tree Selection: n = 2

In addition, compared to the original data, it seems that only Tree Selection has significantly improved the silhouette score, the adjusted rand index scores and mutual information scores. Under other transformations, the performance of the EM clustering algorithm is even worse than previously. The number of clusters being 2 to best cluster the data points makes a lot of sense since the y data is a binary data.

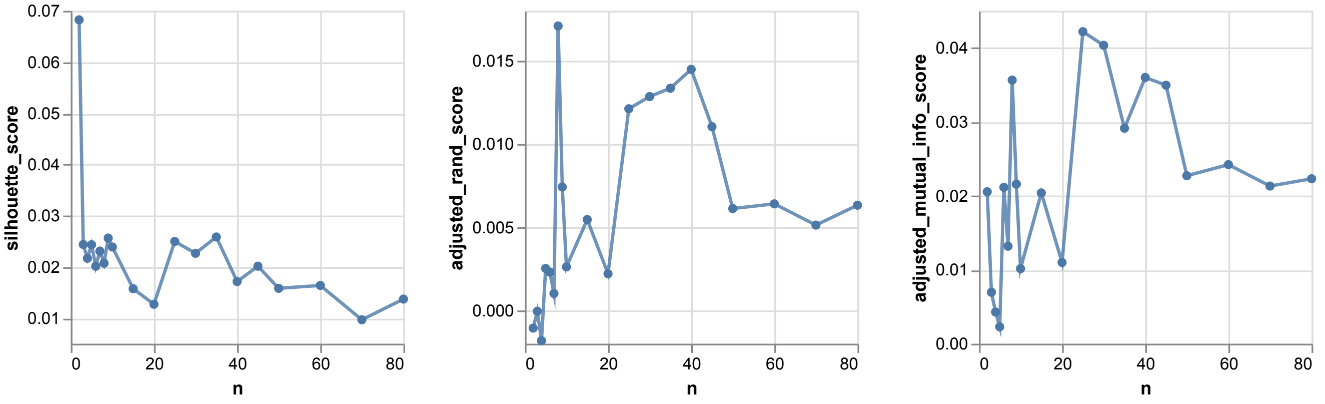
Math Dataset

From previous EM experiment, we learn that the best n value to maximize the clustering is: n = 9.

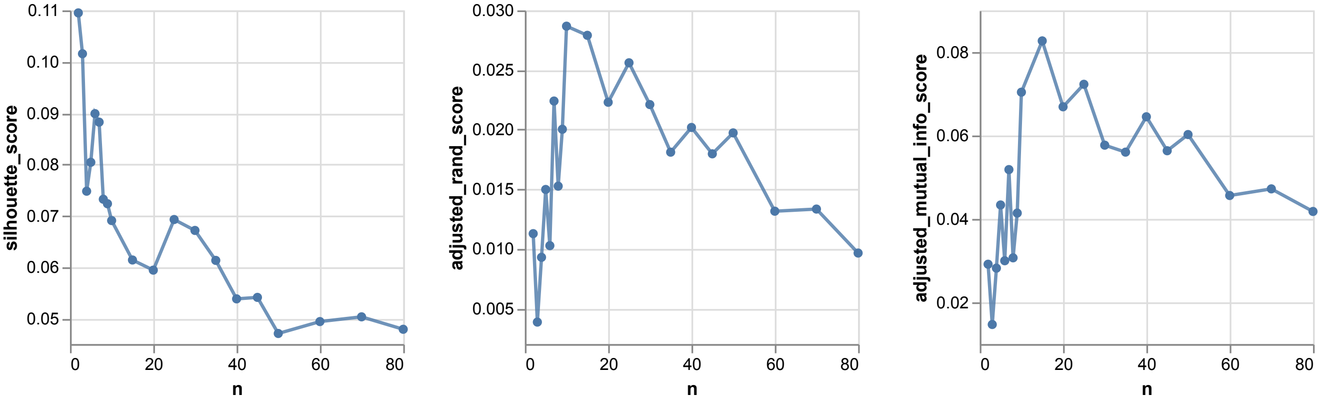
1. PCA



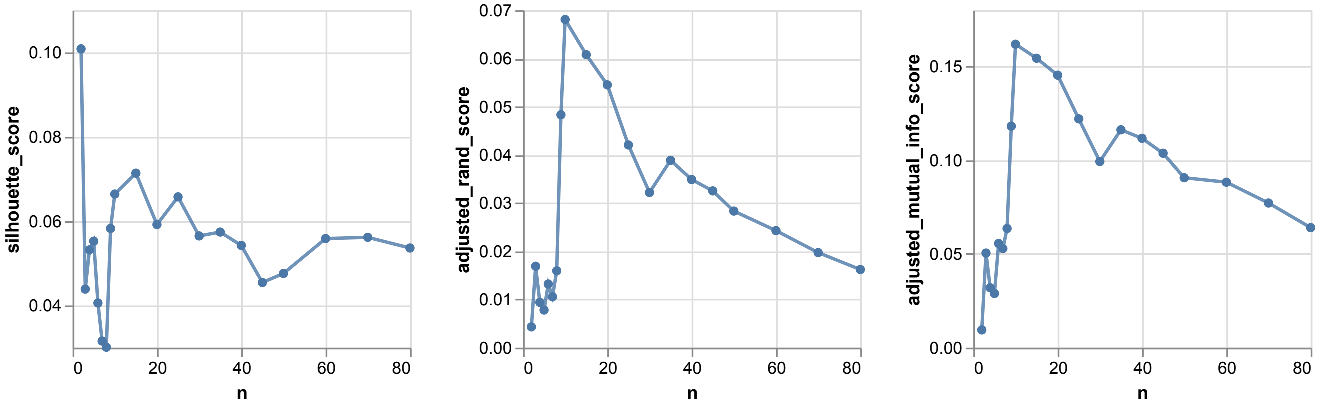
1. ICA



1. Random Projection



1. Tree Selection



From the above chart, it is observed that the optimal n values have changed:

- PCA: n = 10
- ICA: n = 7
- Random Projection: n = 10
- Tree Selection: n = 10

In addition, compared to the original data, it seems that PCA and Tree Selection have slightly improved the adjusted rand index scores and mutual information scores. Among them, only Tree Selection has slightly improved silhouette score. Under other transformations, the performance of the EM clustering algorithm is even worse than previously. The number of clusters being 10 to best cluster the data points makes a lot of sense since the y is a numerical data that falls into range of integers (0, 20).

6. Apply PCA-Transformed Loan Dataset to Neural Network Model

Here, I selected loan dataset that has been transformed with the Tree Selection method to train and test a Neural Network model from assignment 1, and got the following results for accuracy scores:

	Train Accuracy	**Test Accuracy**
Before PCA	0.696	0.714
After PCA (n = 9)	0.802	0.727
After PCA (n = 5)	0.765	0.656

From the table, we observe that feature selection here seem to improved the accuracy of the Neural Network model. This is when the variance of the features cover 90% of the dataset. I got curious and decided to use a different n = 5 to further shrink the test data. Although it seems to increase the train accuracy, it lowers the test accuracy. However, another factor that might have affected the performance is the `StandardScaler()` pipeline conducted prior to the feature selection. However, it is convincing that the

7. Neural Network with Clustered PCA-Transformed Loan Dataset

Here, I selected loan dataset that has been transformed with the PCA method to go through k-means clustering first, and then using the result form the clustering as a new column in the original dataframes, to train and test a Neural Network model from assignment 1, and got the following results for accuracy scores:

	Train Accuracy	**Test Accuracy**
Original	0.696	0.714
PCA + k-means (k = 2, n = 9)	0.804	0.734
PCA + k-means (k = 3, n = 9)	0.802	0.727

From the table, we observe similar behavior with part 6. However, when I got curious and decided to use a different k number (from optimal k = 3 to k = 3), I see an increase in the performance of the Neural Network model. Therefore, I think it proves that the clustering results has a significant impact on the Neural Network model's performance.