Unsupervised Learning and Dimensionality Reduction

Yi Zhao

Yzhao644@gatech.edu

1. Introduction

This paper discusses how to use unsupervised learning algorithms to identify extra information from dataset. Also discussed is using feature reduction techniques to select or project existing features to different dimensions to provide better representation.

In both problems, a key question to answer is how to choose the right number for clustering and feature reduction. I used elbow method with metrics like Silhouette score, Akaike information criterion, Inertia (without touching the ground truth label) and Adjusted_mutual_info_score (used to measure how much cluster is related to ground truth labels) for clustering. For feature reduction, I used Variance distribution, Kurtosis, Transformation error to find the right number to reduce the features to.

The clustering method I used are KMeans and Expectation Maximization. The feature reduction techniques are Principle Component Analysis, Independent Component Analysis, Randomized Projection, Decision Tree Feature Importance.

For the second part, I integrated the clustering label and reduced dimension data and use the newly constructed data feed a neural network to compare the performance with original dataset in terms of training speed, model accuracy and precision and recall.

The dataset I used are Adult dataset that predicts individual’s income level with census data and Wine dataset that measures wine’s quality by wine’s chemical features.

2. Dataset

The first data is obtained from UCI Adult Income Dataset. The data is extracted from 1994 Census database. The task is to predict whether a person will make more than 50K a year.

The features include many geographic information, like age, race, education, origin country etc.

What I like about the data set is there are decent amount of data points, about 30k data points. For feature wise, it has a mix of categorical data and numerical data. After combining the countries and removing the not available values and standarized, the final data has a dimension of 59.

Second data are the results of a chemical analysis of wines quality. The goal is to predict the quality of the wine by providing a score ranging from 3 to 8. The features include chemical levels of ingredients from the wine, like alkalinity of ash and alcohol level.
Since the features are all numerical, it is a great data for clustering. The total feature size is 12.

3. Clustering

3.1 KMeans and Expectation–maximization algorithm

Kmeans is using distance to decide the cluster for each data points. Starting with initial center points, it collects the close points and update to new center points. It converges when no update happens.

The nice thing about using distance is the speed, as it's fast to calculate Euclidean distance between two data points. The downside of is it's hard to decide which cluster it belongs to when distance is same to both group. Initialization would have a big effect on how the algorithm converges. Also, it suffers when the features are not scaled to unit value, as it might put more emphasis on one over another feature. Also when the cluster is not in convex shape, it's hard for the algorithm to work. So proper standardization and later feature reduction is recommended for Kmeans algorithm.

Expectation–maximization algorithm is a two-step algorithm. First to have an expectation about clusters, like mean and variance for a Gaussian distribution. Next is to compute the parameters that maximize the expected log-likelihood. Since it's using MLE to get the best parameters, the training time is longer than Kmeans but it will have a better understanding underlying distribution.

The tradeoff for slower training time is that Expectation Maximization doesn't rely on Euclidean distance and works for non-convex data set.

3.2 Metrics to consider

As mentioned earlier, the key point of the problems is how to decide number of clusters. In 3.1, I discussed the difference between EM and KMeans. There are certain metrics that measures distance well, some measure likelihood better.

Silhouette analysis is based on the distance of the data point, it is very friendly to linear based clustering such as K-Mean. As a measure strategy, its performance on density natural data might not be very ideal.

Inertia is another good metric for KMeans which is essentially a Squared Error metric checking the point's distance with the cernter.

For EM, I used Akaike information criterion as metric. AIC is calculated with number of parameters – likelihood. So the smaller, the better.

With the above metrics, I used Elbow method to determine the K, which is where the curve goes to flat.

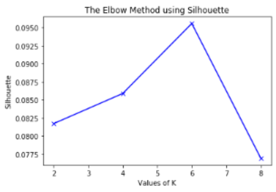
I used SKlearn's KMeans and GaussianMixture for clustering method. For EM there's another hyper-parameters to discuss, covariance-type. Different choice determines how much we care about the covariance between different clusters and feature correlation within one cluster.

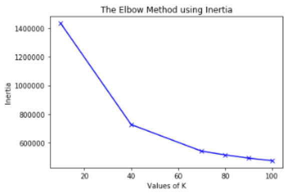
3.3 How the K is selected

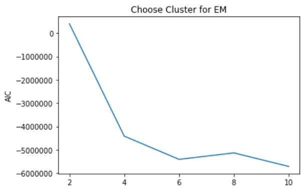
With adult data, my intuition is that in the dataset, there is a feature native_country, which could be a great candidate for clustering. So I started to find clusters smaller than 10. Also having a lower K is nice to visualize the data. For wine data, I tried from 2-20 due to there are less features.

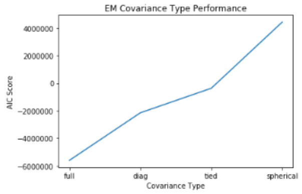
Target label is excluded when training for clustering.

As discussed in 3.3, four elbow methods are implemented to decide the best parameters, 2 for reach. Result as follow for adult data set.





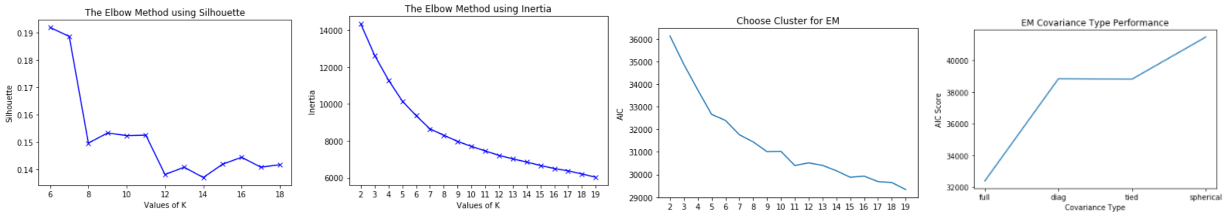


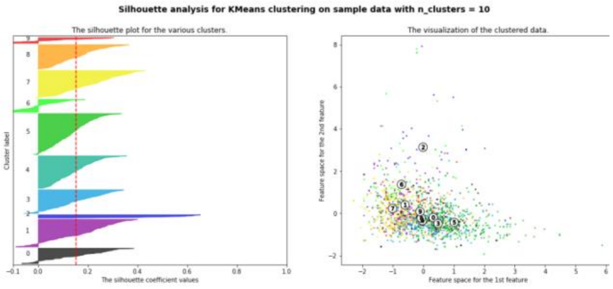


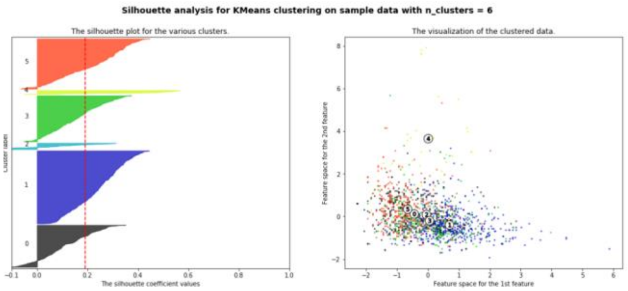
For Kmeans algorithm, Inertia always decreases as cluster number increases. With Sihouette metric, the best cluster is determined as 6.

For EM algorithm, cluster is also best at 6. I also compare variance type vs metrics. With 'full' as covirance_type, the EM performs the best. This makes sense, with many one hot encoding features and feature engineer to remove duplicate or highly correlated features, there is not much correlation between features for our dataset. So, with full covariance, the model can better measure the correlation between internal feature and outer clusters.

For wine dataset, elbow method shows a good cluster should between 6-10. With the help of visualization and Silhouette score, I decide to use 6 for both algorithm. Full covariance type also shows highest AIC score. The reason is there is not much correlation between wine's chemical feature.







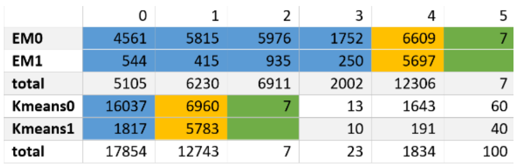
3.4 Result analysis

I use three methods for result analysis. First to measure mutual information between cluster and target label. I used adjusted_mutual_info_score to check that. My intuition for adult dataset is that the cluster doesn't have much relation to the target variable, as features describes census information, which not necessarily related to the income exactly. It makes more sense for the data to represent cluster by geographic than income itself. Plus as the best clusters are determined as 6 while target label only has two value, it's unlikely they match each other.

For wine data set, I was hope 6 clusters would somehow match to 6 points.

My mutual info scores for adult are about 0.1318 for both algorithms which is inline with my idea, and only 0.082 for wine data set. The cluster clearly is measure another latent information than the points here.

Secondly, I check overall distribution of point to each clusters, and also target label with clusters.



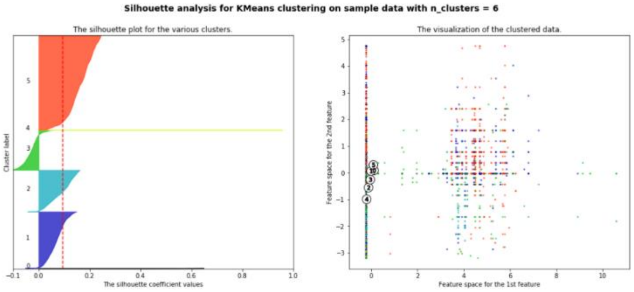
	0	1	2	3	4	5
EM0	4561	5815	5976	1752	6609	7
EM1	544	415	935	250	5697	
total	5105	6230	6911	2002	12306	7
Kmeans0	16037	6960	7	13	1643	60
Kmeans1	1817	5783		10	191	40
total	17854	12743	7	23	1834	100

The first insight is Kmeans Cluster 0 and EM cluster 0-3 has about the same distribution of the target variables. Changes are higher for the point to be labeled as 0 if they are assigned to those groups.

Kmeans cluster 1 and EM cluster 4 has about the same distribution. And they will like to be labeled as positive then other cluster, but still only 50% chance.

Also the mapping above is have the same label, meaning Kmeans make what EM 4 cluster as 1, while EM makes kmeans cluster 0 to 0-3 separately. Cluster 2 and 5 are also the same 7 individuals.

For wine data set, EM and Kmeans have similar score distribution within cluster. EM's cluster 4 and Kmeans cluster 7 have similar features. EM has a better performance on separating score 7 with 5 and 6. My explanation is that the underlying data fit EM well because it's not very convex, so Euclidean distance doesn't perform as well as distribution likelihood maximization. Due to the limit space, detail could be found in notebook named wine.



Lastly I visualize my Kmeans result. The center are all located to 0 (capital gain). This will be used to compare with next section when ICA and PCA has a better representation of data.

Wine dataset's result shows in 3.4.

4. Dimensionality Reduction

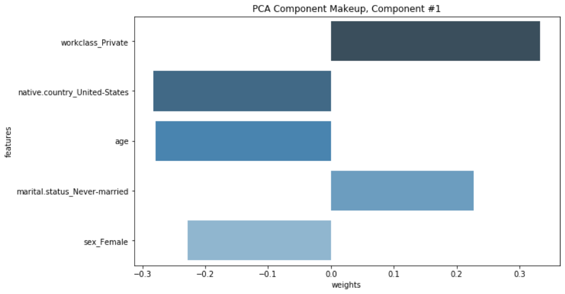
Increase dimensions has many benefits. Firstly, one hot encoding would transfer categorical data to numerical data, which is basically required for every machine learning model. Secondly, data could be only separatable in higher dimensions.

But in higher dimension, calculation could be slower; noisy feature could cause high variance; sparse matrix could cause other calculation and performance problems. This curse of dimensionality results necessity for feature reduction.

4.1 Principle Component Analysis

PCA use SVD techniques to change original data to eigen vector dimensions, so that original information could be represented with less features but as much (measured by total variance percentage) as possible.

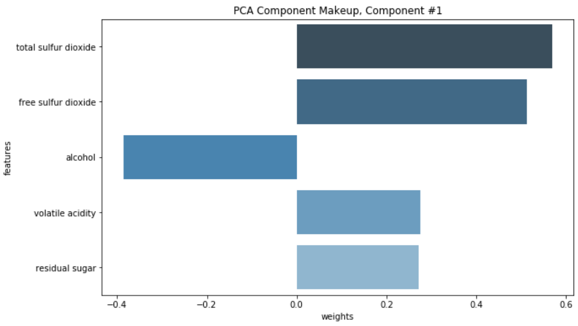
I used sklearn's PCA function and PCA.explained_variance_ratio_ to find the best number of features. In my experience, around 80% is a good representation of the original information. In our case, I need 34 features in adult dataset and 5 in wine dataset to achieve that.



One thing to notice on adult dataset is that largest eigen value only takes about 7.5% information and rest is less than 5%. This could be a sign there aren't much noise in the features and even with transform, there aren't many decisive features.

I also visualize the first and most important features.

So the most important eigen vector consists of work_class, native_country and age, which makes sense to me, as this targeting US, elder single employee that works for private institution.



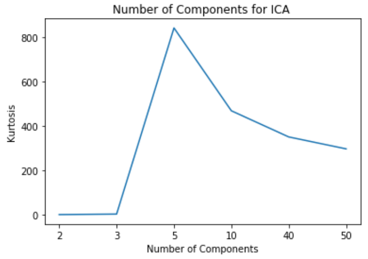
For wine data, top 4 feature covers about 70% of the variance, which means there are high insights in the first few components. Sulfree is chemical that affects wine taste and no one would enjoy sour wine. So I'd like to say they are incline with the common sense.

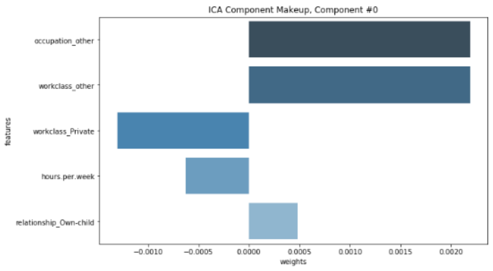
In the later tree importance, this will be revisited with more proof.

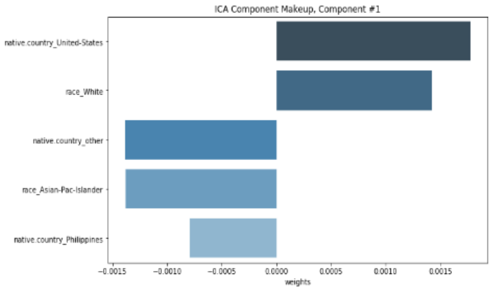
4.2 Independent Component Analysis

ICA is also using linear combination of original features to map them into a new dimension. The assumption is that the original data is sum of independently non-Gaussian distribution. So the metric to decide the number of features to keep is Kurtosis of the transformed data.

I used elbow method again to get the number as 5.

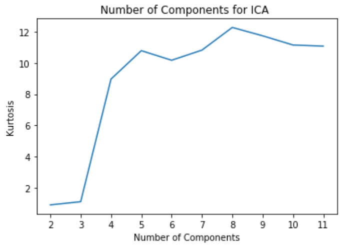


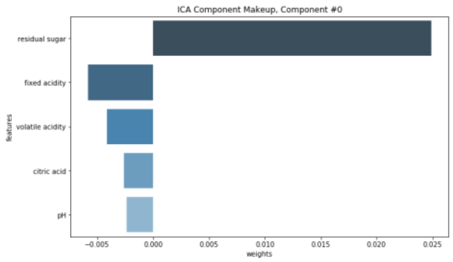




Also a visualization of first two ICA components are shown. Occupation and Work class is evident that component is talking about working. Second component is talking about geo-location of the individual.

Geo-location and working are quite independent.





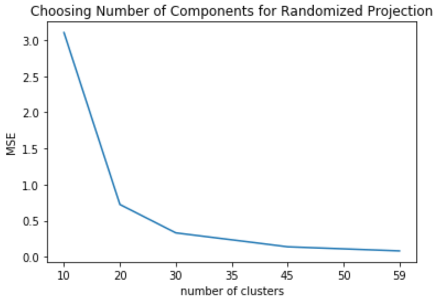
For wine data set. The best number of features is 8 according to elbow method. The ICA also projected data quite well, as the first component clearly captures the acid level of the wine.

4.3 Randomized Projections

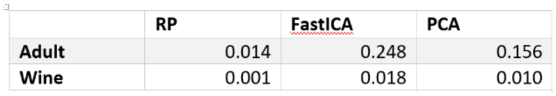
RP randomly selected features and use Johnson-Lindenstrauss lemma, to preserve the distance from higher dimensions in order to reduce dimensions.

The metric I used is the Mean Squared Error between transformed matrix and original matrix. The transformed matrix is calculated by GaussianRandomProjection.components times output of the output.

I run the RP multiple times and take the average of the errors. I chose the 45 as the elbow point and it is inline with the assumption that RP would need more features than PCA and ICA. For wine data set, the number is 6.



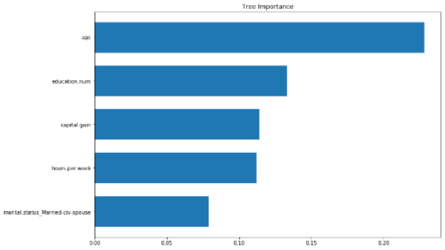
Due to the simplicity, RP is faster to train than other algorithms. I use the same components and dataset to gather the time spent. RP is at least ten times faster than other two.



	RP	FastICA	PCA
Adult	0.014	0.248	0.156
Wine	0.001	0.018	0.010

4.4 Decision Tree Feature Importance

The last feature reduction techniques I chose is using decision tree's feature importance to get the most important top features. The importance is calculated by Gini index in the scikit learn set-up. Individual node's importance is calculated first by checking the weighted impurity from parent to child node. And feature's importance is a combination of the nodes and normalized by all the features. For decision tree feature reduction, I chose 15 which covers 85% of the total importance for adult data and 7 for 71% importance for wine data. Below are the chosen features for adult data.



To compare with PCA and ICA, many features appear top from the list here also shows in the first component from PCA and ICA, like age, marital status and occupation.
Wine detail could be found in the note book.

5. Clustering after Feature Reduction

In this section, I ran the two clustering algorithms on four sets of reduced-feature data on two data sets to get 16 combinations. The idea is that with the help of feature reduction, clustering would have a better way to separate data points to different clusters. For example, in PCA or ICA setting, the data are projected to a new dimension that are either independent or orthogonal to each other, which is easy to separate.

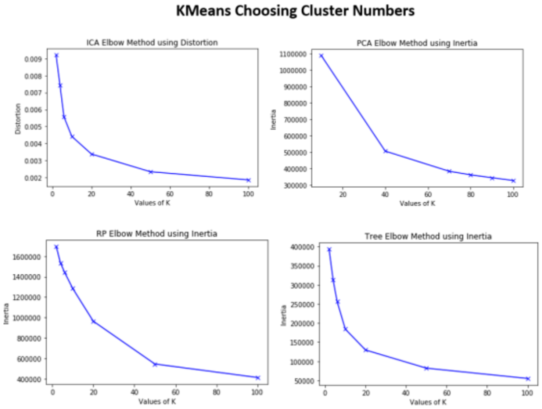
The result also confirms that assumption with higher silhouette score and more data clustered in the first few clusters.

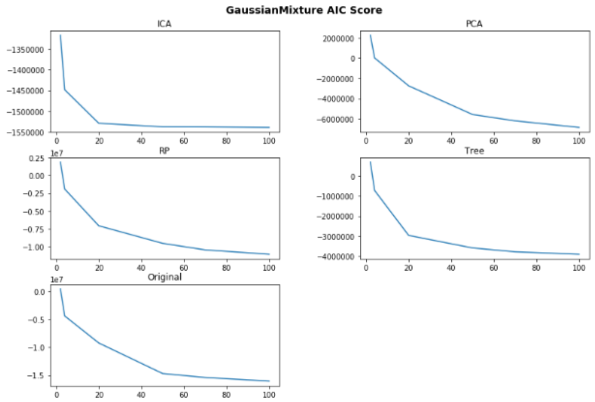
5.1 Elbow again

The first question to discuss is again to choose the right number of clusters. I am using the elbow method the same way as first section. I also expand my search from 10 to 100 for adult dataset given the feature dimension has been reduced. Wine data cluster is from 1 to 11.

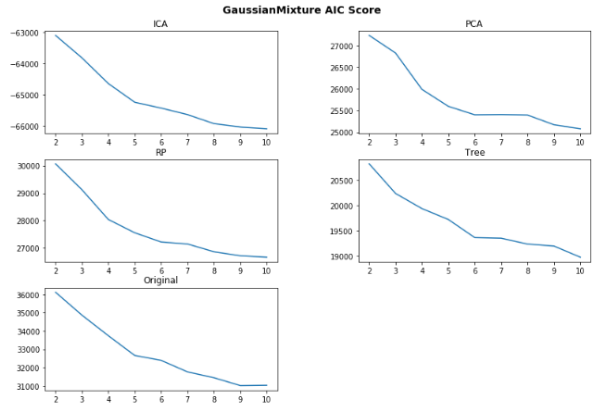
From the below elbow plots, the best cluster numbers for both EM and Kmeans are about the same. For ICA (5 dimension) and Decision Tree (15 dimension) methods, 20 clusters are needed. 50 for PCA (34) dimensions and Random projection (45). Here is my first interesting find. The number of the clusters has a positive relation with number of features. Less features leads to fewer clusters for both clustering algorithms.

This makes sense because when you move from 2D to 3D dimension, there are more ways to separate the points.





Below is wine data's elbow plot.



For wine data the cluster is 6 for all clusters.

There's not much change between original data and feature reduced data, probably due to that all features are numerical and independent to each other.

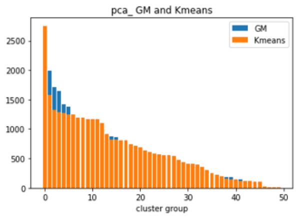
Also the findings on first clusters of reduced feature data having more datapoints than original data is not captured for wine data set.

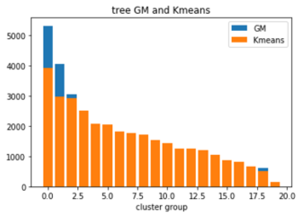
5.2 Better fitness with cluster reductions

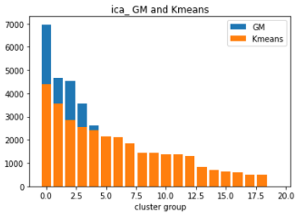
My second interesting finding is that the clusters are separating the data points better than before feature reduction. This is also one of my assumption why the feature reduction works better for clustering method.

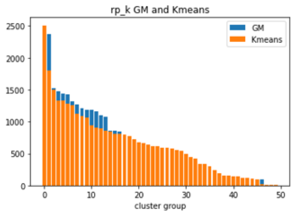
Below I plot the number of points by cluster in descending order. Note that the cluster label are different between two clusters. I only ordered by counts of points.

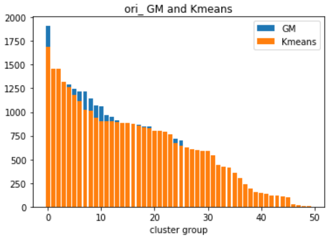
The first finding is that feature-reduced methods all have larger number of points than original data. (2500 and 3500 vs less than 2000). This means the data are more closely clustered than before. This could also be shown from the chart showing the Silhouette score in the lower right.



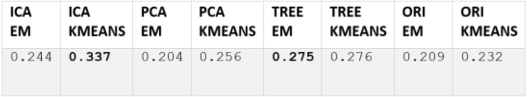








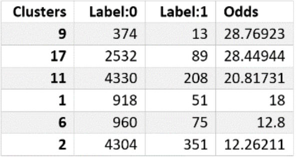
Silhouette Score after Feature Reduction



ICA EM	ICA KMEANS	PCA EM	PCA KMEANS	TREE EM	TREE KMEANS	ORI EM	ORI KMEANS
0.244	**0.337**	0.204	0.256	**0.275**	0.276	0.209	0.232

Second finding is that for ICA reduction method, EM fits better than Kmeans, since more data points are clustered in first few clusters. This make sense because ICA projects data to independent distributions that are easier for EM which tries to fit different distributions to divide clusters.

5.3 The label representation still holds

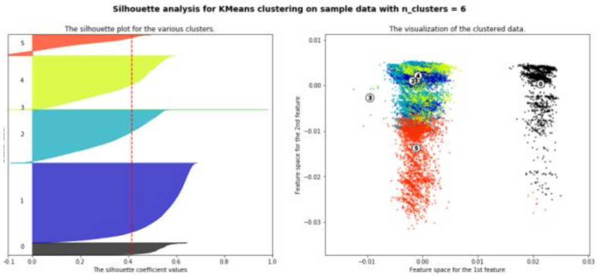


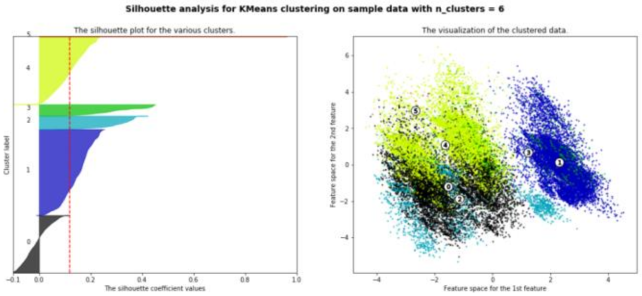
Clusters	Label:0	Label:1	Odds
9	374	13	28.76923
17	2532	89	28.44944
11	4330	208	20.81731
1	918	51	18
6	960	75	12.8
2	4304	351	12.26211

Even with more clusters, the label representation mentioned in 3.5 still holds. Just taking ICA EM clustering as an example, if a data is assigned to cluster 9, 17, 11 etc., it's likely to be assigned a 0.

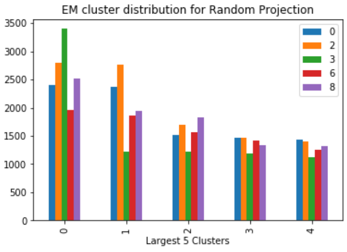
5.4 Cluster visualization for ICA and PCA

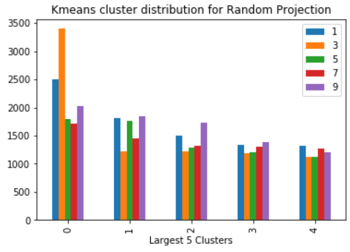
The third interesting finding is that ICA and PCA helps project data to new dimensions that are easy to visualization. Compare with the original Kmeans visualization, this is also a proof that data are clustered by after feature reduction.





5.5 Randomized Project has low cluster variation





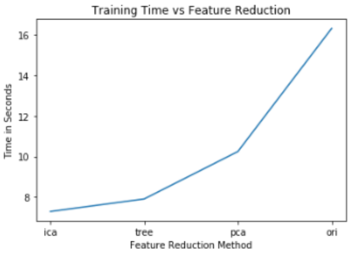
On the right is result of running RP on original data and cluster with 10 clusters with 5 different random seed. Even though the cluster numbers are different, the distribution of the points are the same. The rows belongs to same cluster are mostly assignd a same new cluster with a new RP. This means we can trust RP to transform data. The way data are clustered are not random after all. Only the ways to choose features to create new projections are random. Wine data has the same performace with low vairance on clusters. Detail could be found in wine notebook.

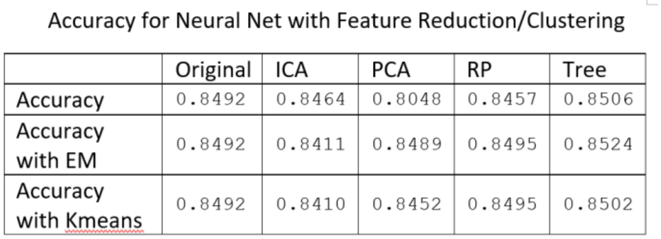
6. Neural networks with feature reduction

In this section, I run neural networks to predict target label with the same data structure as assignment 1 and compare the result. The comparison is in two ways, the speed and precision/recall, accuracy metric. I went through hyper-parameter tuning again since we have different data, but the neutral network's structure remains the same. Below result is produced with the best parameters.

6.1 Running time comparison

One big advantage of feature reduction is to improve the training speed, as there are less features to consider when training. Time comparison as below confirms that. The original data has the slowest running time where ICA runs the fastest exactly in the same order of number of features.




Accuracy for Neural Net with Feature Reduction/Clustering

	Original	ICA	PCA	RP	Tree
Accuracy	0.8492	0.8464	0.8048	0.8457	0.8506
Accuracy with EM	0.8492	0.8411	0.8489	0.8495	0.8524
Accuracy with Kmeans	0.8492	0.8410	0.8452	0.8495	0.8502

6.2 Model Accuracy and Precision and Recall

In terms of the accuracy, I was not able to get a significant improvement, especially for ICA and PCA feature reductions. My explanation is essentially ICA and PCA are essentially doing some linear transformations on features to project a data to a new dimension. This could also be achieved by neural networks, which is exactly using linear combinations of input parameters. There's not much information that ICA and PCA could bring into the model that neural networks haven't learnt. Another explanation is on the variance ration of top principal components which mentioned in section 4.1, there is not much improve for PCA as the eigen values rations are relatively small.

The improvement of tree could be due to removal of the noisy features. The result of the accuracy is shown above.

6.3 Best parameters before and after

Though I do not need to change the neural network's structure for comparison, I tried to use grid search to find the best structure. It turns out that feature reduced data's best parameter all have one layer than original data. It seems like original data would need the extra layer to match the performance with other transformed data and this layer could exactly achieve what ICA and PCA does.

7. Neural networks with cluster

In this section, I brough in the clustering label is an input parameter and retrain the neural networks to predict true label. The idea is that the cluster could bring in some latent information about underlying data to help the model to for better prediction.

7.1 Cluster choice

The cluster I used is from the original data set instead of the clustering result on individual reduced dimension data set. This is according to TA's confirmation in the Piazza. I think one can argue this is a better approach as we are trying to find a general latent information to help prediction. We want to get as much information as possible. So using original data makes sense this way.

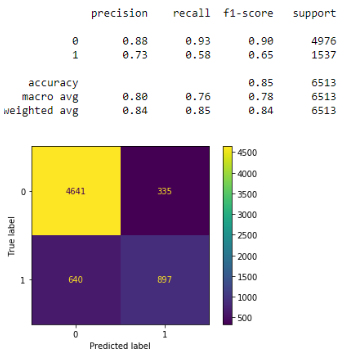
7.2 Training setup

For the training setup, I use 50 as my number cluster according to section 5.1 elbow method. I used one-hot encoding to transfer label encode to 50 extra features. Then standardize them and concatenate with reduced feature data as input data.

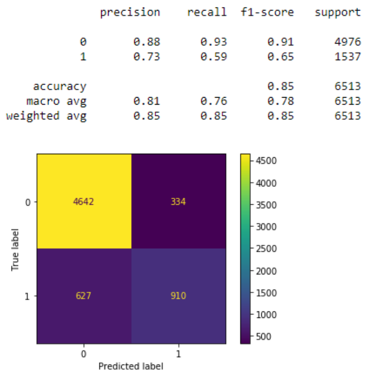
7.3 Result analysis

The accuracy metric is shown in 6.1. There is not much improvement in terms of accuracy. I will argue that the cluster information does not help a lot with the accuracy. I proved my point by calculating the adjusted_mutual_info_score for new clusters. The result, 0.06 is even lower than cluster with 6.

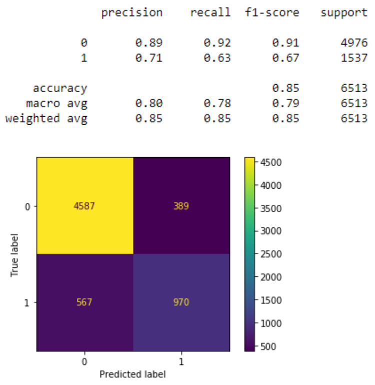
But there is some improvement on metrics for precision and recall, as shown below. The F1 score increase from 0.90 to 0.91 in both case. I explain this as the cluster provided a latent information to avoid false positive and false negative.



```
              precision    recall  f1-score   support

           0       0.88      0.93      0.90      4976
           1       0.73      0.58      0.65      1537

    accuracy                           0.85      6513
   macro avg       0.80      0.76      0.78      6513
weighted avg       0.84      0.85      0.84      6513
```



```
              precision    recall  f1-score   support

           0       0.88      0.93      0.91      4976
           1       0.73      0.59      0.65      1537

    accuracy                           0.85      6513
   macro avg       0.81      0.76      0.78      6513
weighted avg       0.85      0.85      0.85      6513
```



```
              precision    recall  f1-score   support

           0       0.89      0.92      0.91      4976
           1       0.71      0.63      0.67      1537

    accuracy                           0.85      6513
   macro avg       0.80      0.78      0.79      6513
weighted avg       0.85      0.85      0.85      6513
```

From left to right, original data metric, best with EM clustering, best with Kmeans clustering.

Summary

Clustering can provide latent information on underlying data. Feature reduction can improve training speed, sometimes improve supervised learning accuracy. EM works better if underlying data has independent distribution. Kmeans converges faster. PCA and ICA works great on data projection and hence help with clustering and visualization. RP method will project data to similar distribution and cluster will not change even random seed changes.

Reference

https://en.wikipedia.org/wiki/Akaike_information_criterion

https://www.researchgate.net/post/How_can_I_test_the_GMM_clustering_result_using_a_measure_different_to_BIC_Is_appropriate_to_use_Silhouette_beyond_to_BIC

https://www.bonappetit.com/drinks/wine/article/sulfite-free-wine