GEORGIA INSTITUTE OF TECHNOLOGY
Title: Machine Learning Homework3 – Unsupervised Learning and Dimensionality Reduction

Author(s):
Weifeng Lyu

Instructor:
Prof. Charles Isbell,
Prof. Michael Littman

TABLE OF CONTENTS

1. Introduction

1.1 Problem

This project implements six algorithms, two of them are clustering algorithms such as k-means clustering, Expectation Maximization, then implement other four dimensionality reduction algorithms such as PCA, ICA, Randomized Projections and Univariate feature selection (K best) algorithm.

1.2 Computer Configuration

Manufacturer: Dell, Model: Inspiron 7559 Signature Edition, Processor: Intel® Core™ i7-6700HQ CPU @ 2.60GHz, Installed Memory (RAM): 8.00GB (7.88 usable), System Type: 64-bit Operating System, x64-based processor

2. Dataset

2.1. Player Stats

This dataset is one of the datasets being used in Supervised Learning, which is standard long-period history small volume of data, 18000+ rows, acquired from Kaggle https://www.kaggle.com/drgilermo/nba-players-stats. For this experiment, the output is alternated from Supervised Learning, the data predicts the type of player win share level based on the stats they play in historical data. The data should use quantile_transform to pre-process the data to scale them into identifiable values since the data adheres gaussian output distribution. However, due to the expensive running time and some negative values cannot be processed, some of unimportant attributes are deduced. In addition, I pre-process this dataset with cleaning some rows and columns with null value, unselected those strongly-related attributes such as offensive win share, defensive win share. Most of attributes are selected for predicting the model. After unselecting those attributes, also win share attributes are scaled into 0 to 5, which represents 6 levels of player from bench warmer, role player, sixth man, normal starter, all-star starters, hall of fame superstar. The reason I choose this dataset is that all attributes are correlated to player win share strongly or weakly. Implementing feature selection is expected to increase the algorithm performance significantly. This dataset is also a multi-class output so clustering as unsupervised learning is expected to perform a good cluster to separate the data.

2.2 Wine Quality

I found CreditCard dataset contains many negative numbers involved and non-scalable attributes, maybe it is a bad idea to remove those attributes. For this experiment a new dataset is specifically used, which is a standard machine learning dataset with small volume, only 3919 rows with 12 columns. The output is the scaled quality of wine evaluated by different settings of wine such as pH, alcohol. The data should use MinMaxScaler to pre-process the data to scale them into identifiable values. This dataset somehow performs differently than NBA Player Stats because some good combination of data should give good wine quality, rather than most attributes in NBA Player Stats contribute to player level for greater amount. The reason I choose this dataset is that all attributes are correlated to the quality of wine with combinational impacts. Implementing feature selection algorithm is expected to increase the algorithm performance. This dataset is also a multi-class output so clustering as unsupervised learning is expected to perform a good cluster to separate the data.

3. Project Architecture

3.1. Python Implementation

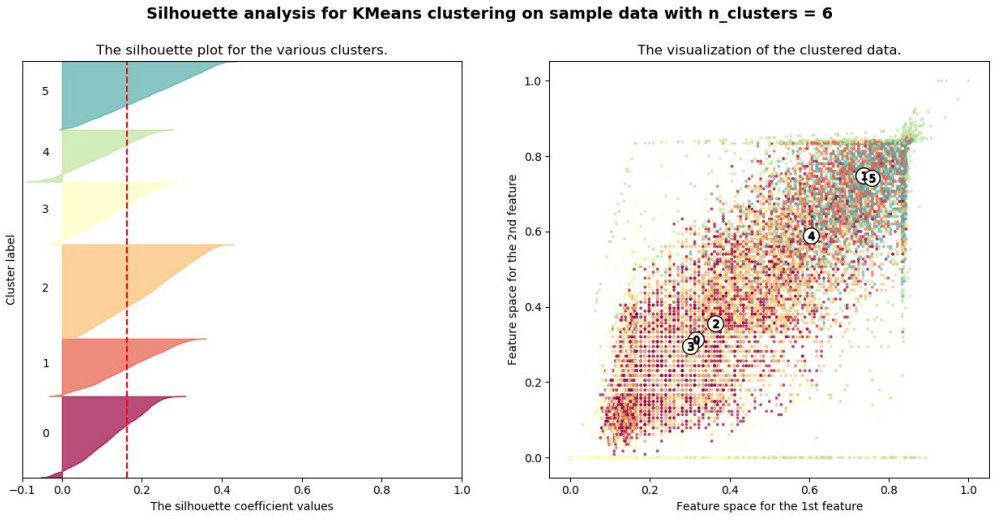
This project implements in Pycharm IDE with Python 3.6 with other toolkits such as Numpy, Pandas and machine learning framework sci-kit learn and the data are processed by Python Matplotlib to plot the figure. The workflow is designed for analyzing data visualization based on different unsupervised machine learning models, these models are as k-means clustering, Expectation Maximization, PCA, ICA, Randomized Projections and Univariate feature selection (K best) algorithm. Then we apply the clustering algorithms and the dimensionality reduction algorithms then rerun your neural network learner on the newly projected data. Two clustering methods and four feature selection algorithms are implemented for visualizing how data distributed with Python Visualization tool Matplotlib. We will also generate the Learning curves that aims to evaluate the model performance, usually denoted by the risk, cost or score versus the size of training set and test set. The learning curve indicates how much data the machine learning method might need to optimally train the model. Here are some default settings about my implementation:

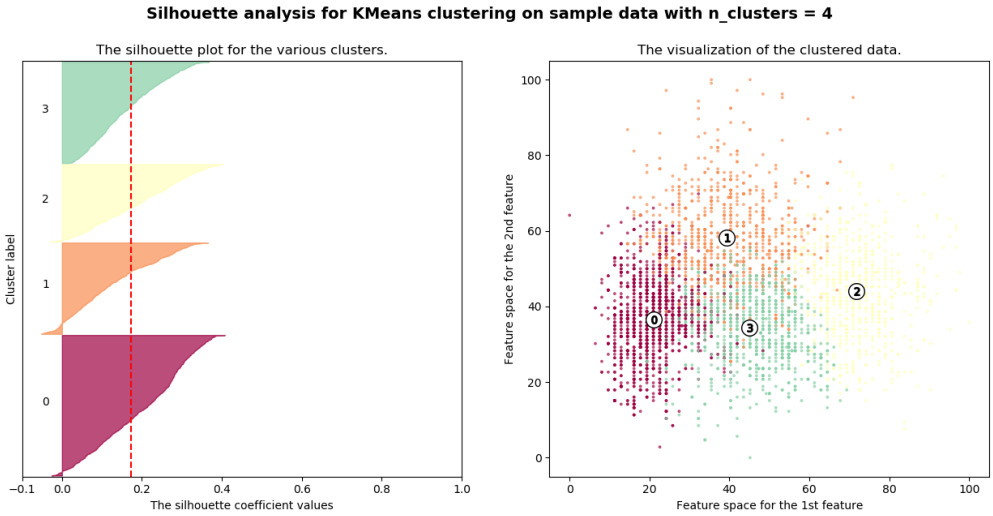
1. range_n_clusters = [2,4,6,7] for PlayerStats and range_n_clusters = [2, 4, 6, 8] for WineQuality
The number choice is selected based on the number of classes is 7 for PlayerStats Win Share level and the number of classes is 8 for Wine Quality level, it should be strictly less than the number of output classes
2. n_components=37 for PlayerStats and n_components=11 for WineQuality
The number choice is based on the number of X attributes for PlayerStats and WineQuality, which means there are 37 attributes and 11 attributes considered.
3. n_clusters=6 for both dataset
n_clusters represents the number of clusters to form as well as the number of centroids to generate.

3.2. Project Demonstration and Sample Result
PlayerStats dataset has 6 classes to define 6-level of NBA players. Therefore, K is chosen to be K=2, 4, 6. From the silhouette plot we can see how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. Since NBA PlayerStats data is usually in normal distribution, quantile_transform is arranged for normalizing all the features. The other dataset WineQuality has 8 classes to define 8-level of Wine Quality, so K is chosen to be K=2,4,6,8. The Euclidean distance used by sklearn was dominated by this large variance feature, so we determine to use MinMaxScaler to normalize all the features, which demonstrates better visualization.

(a). Clustering algorithms - K means clustering
K means clustering will pick k centers at random, for each center, it will search for the nearest points, then the number of points will increment and recompute the centers by averaging the clustered points, the process will continue until the center is unchanged and k clusters are converged. Therefore, Clustering problem is a set of objects X and inter-object distances D(X,y)=D(y,x), x and y are belonged to X, then the clustering will produce the output of partition Pd(x)=Pd(y) if x and y are in the same cluster.

n_clusters	NMI score	average silhouette_score	n_clusters (WineQuality)	NMI score	average silhouette_score
2	0.381222	0.293585	2	0.104596	0.246644
4	0.288042	0.214155	4	0.084505	0.173279
6	0.32416	0.166555	6	0.081843	0.145548
			8	0.090088	0.142966



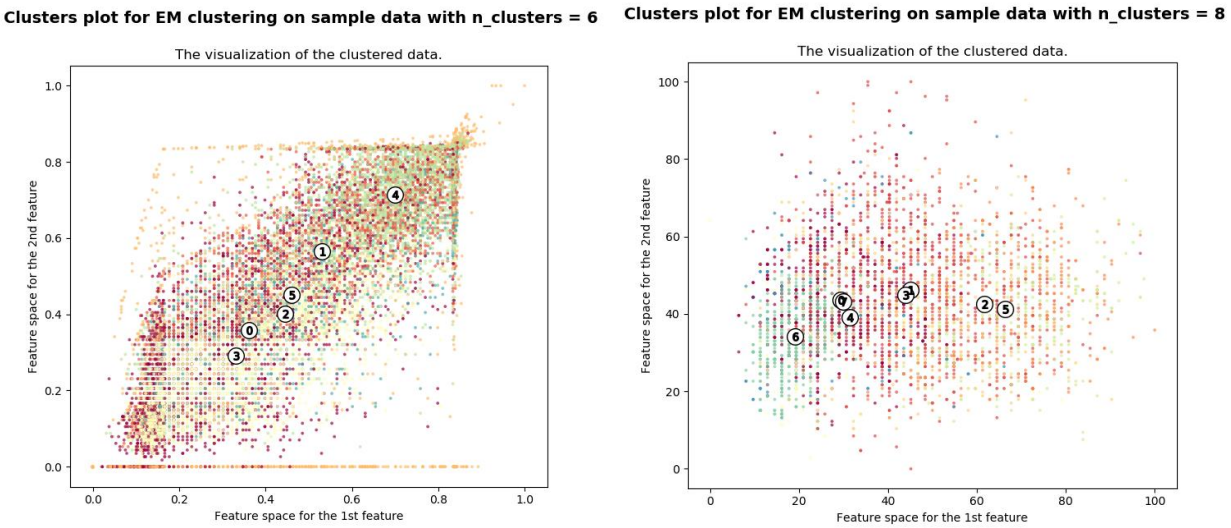


As we can see NBA PlayerStats is separated not as very well as WineQuality by different features. After comparing all the silhouette plots. The silhouette plot for PlayerStats is not clearly separated after increasing K to 6, but it provides more or less of similar thickness hence are of similar sizes can be verified from the labelled scatter plot on the right. Increasing K to much larger value cannot provide very good separation due to its data structure. For WineQuality, the classes are separated very well according to the silhouette plots, comparing K=4 and K=8, it seems that increasing K does not provide more meaningful separation, clusters [3, 4], [5, 0], [6, 7] are subgroups of clusters in the above scatter plot. All plots are available from plot folder, only selective plots are presented in this report. Both Silhouette Coefficient and Normalized Mutual Information (NMI) were calculated to test the performance of different K. As these NMI values show that our clustering results is very poor, NMI score is varied from 0.28-0.38 while K is range from 2 to 6 for PlayerStats dataset, NMI score is varied from 0.08-0.1 while K is range from 2 to 8 for WineQuality dataset. I believe it must have been caused by treating all features equally important. And because we normalize so that all of them can influence the distance a lot, some attributes that don't really distinguish between Player Win Share level, or between wine qualities introduced a lot of variance to my clustering. That's will lead me to the following sections of feature selection process. Also, the Silhouette scores were significantly lowered due to the same reason. One of interesting facts is that when K is equal to the number of player level or quality level, NMI score and Silhouette score will increase slightly.

(b). Clustering algorithms - Expectation Maximization

Expectation maximization is similar with K means clustering in algorithmic level, it will repeat through two probabilistic calculations to find maximum likelihood or maximum posterior estimates of parameters in statistical models, where the model depends on unobserved latent variables, while maximum likelihood estimation can find the "best fit" model for a set of data, expectation maximization estimates at the parameters first, accounting for the missing data, then tweaks the model to fit the guesses and the observed data. Expectation Maximization improves a parameter's estimation through this multi-step process. For Expectation Maximization clustering, we run the experiment with the same K clusters and normalized values.

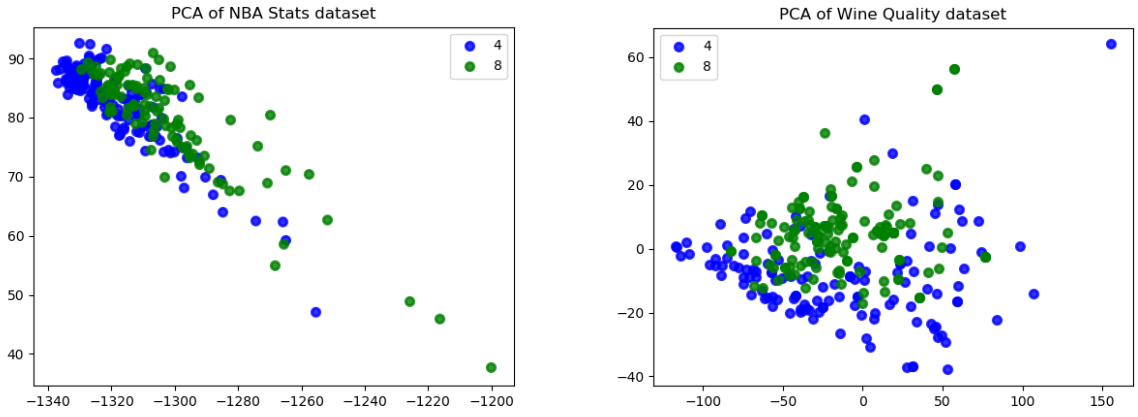
n_clusters (PlayerStats)	NMI score	n_clusters (WineQuality)	NMI score
2	0.294986	2	0.049031
4	0.226446	4	0.056479
6	0.226446	6	0.048976
		8	0.078178



We can observe the fact that WineQuality clustered data distribution is much better than PlayerStats because WineQuality's data aligns with Gaussian distribution than PlayerStats, less noisy data after data pre-processing. Another observation is that NMI score of results are very low for WineQuality, and both datasets are worse than K means method. In addition of K selection, for PlayerStats, we still max out the K but it did not divide the clusters very well, and for WineQuality, it's not so noticeable for the performance between K=4 and K=8. It would make more sense if some features are filtered. Min max normalization cannot deal with the features with extreme values. The distribution of normalized values is not ideal. Other normalization method could improve the performance better.

(c). Dimensionality Reduction Algorithms – PCA

Principal Component Analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent, let's consider n observations with p variables, then the number of distinct principal components is min (n-1, p). This transformation is defined in such a way that the first principal component has the largest possible variance, each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Therefore, data with m-columns (features) is projected into a subspace with m or fewer columns, whilst retaining the essence of the original data, it can be implemented in linear algebra.

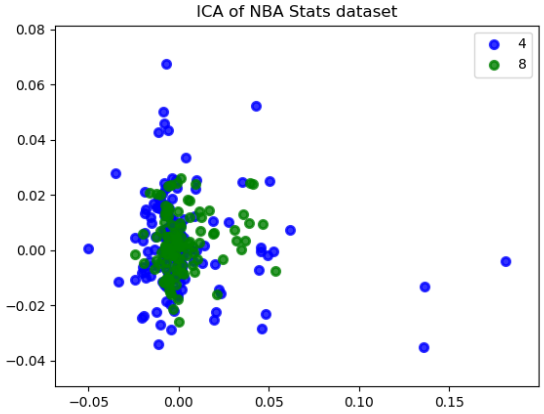


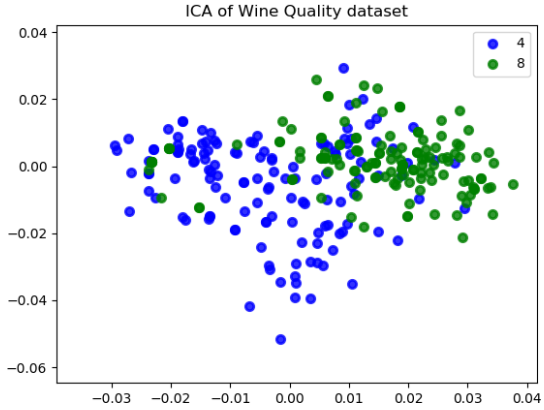
We run PCA against to quantile transformed data for PlayerStats and min max normalized data for WineQuality. The separation of points is slightly better. The magnitudes of values were changed while the variability of samples was kept the same, the percentage of variance for each component is the same. From PCA, we can obviously visualize the data better with dimensional reduction. Also the following are explained variance ratio to evaluate estimated covariance of data:

Explained variance ratio (first two components) for PlayerStats: [9.30e-01 3.26e-02 2.03e-02 6.92e-03

4.29e-03 2.70e-03 7.74e-04 6.51e-04 4.05e-04 3.19e-04 1.88e-04 1.71e-04 1.38e-04 7.96e-05 5.14e-05 3.34e-05 2.54e-05 2.02e-05 1.50e-05 1.10e-05 8.30e-06 6.21e-06 9.96e-07 4.67e-07 3.17e-07 3.04e-08 2.00e-08 1.44e-08 1.43e-08 1.26e-08 6.54e-09 8.65e-10 2.18e-10 1.97e-11 7.10e-33 7.10e-33 7.10e-33 7.10e-33]
Explained variance ratio (first two components) for WineQuality: [9.12e-01 7.77e-02 9.98e-03 5.00e-04 3.14e-04 8.40295219e-06 6.52e-06 5.28e-06 3.95e-06 1.71e-07 1.56e-10]
The amounts of variance explained are very high in these two complex datasets. Both dataset are performing well under PCA, especially NBA PlayerStats.

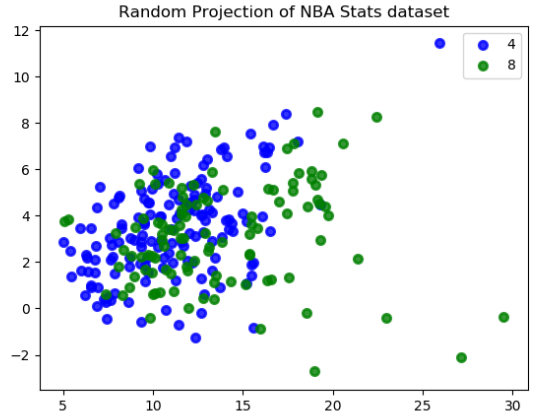
(d). Dimensionality Reduction Algorithms – ICA
Independent Component Analysis (ICA) means we have some random variable, or more commonly, a random vector that we observe, and which are assumed to be linear or nonlinear mixtures of some unknown latent variables, and the mixing system is also unknown, then aims to removes correlations and higher order dependence. The latent variables are assumed non-gaussian and mutually independent, and they are called the independent components of the observed data. The measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem.

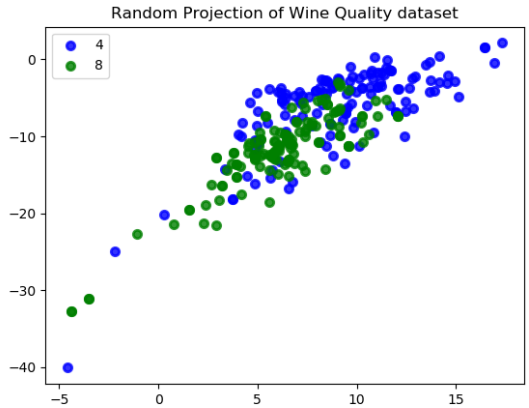




We run ICA against to quantile transformed data for PlayerStats and min max normalized data for WineQuality as well. ICA seems to separate the data better than PCA because ICA is separation of signals from several independent sources "mixed up" together. And the amounts are very low, this seems the fact that only feature transformation can't directly separate the classes, ideally the transformed values can provide better classification results if a feasible classifier applies. Here we can see ICA of NBA PlayerStats dataset performs better under ICA than WineQuality.

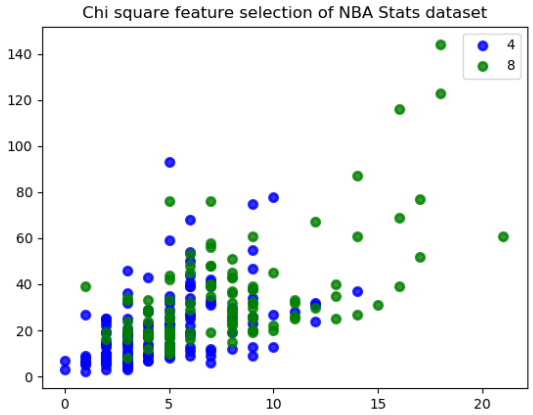
(e). Dimensionality Reduction Algorithms - Randomized Projections
Random projection is a tool for representing high-dimensional data in a low-dimensional feature space, typically for data visualization or methods that rely on fast computation of pairwise distances, like nearest neighbors searching and nonparametric clustering.

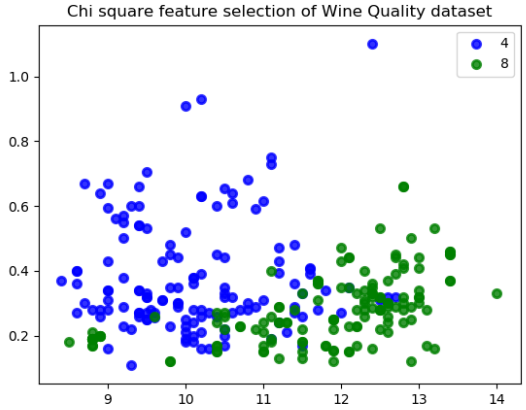




As we have a numeric dataset with n examples, each of which is represented by d features (where d is presumably relatively large, maybe on the order of hundreds or thousands). In other words, the data becomes a matrix X, with n rows and d columns. Randomized Projections performs well in high-dimensional dataset, the similarity of data vectors is preserved well under random projection.
From random projection method for feature transformation. There seems to be not much separation are visible, maybe due to the fixed random state = 10. From the plots, we can see the wine dataset performs better than PlayStats, it might be due to those attributes are more randomly scaled, rather than the fact that NBA PlayerStats are distributed much closer, many historical data and average data are mixed without preprocessing.

(f). Dimensionality Reduction Algorithms - Select K Best
K best algorithm a feature selection algorithm that scores the features using a function and then "removes all but the k highest scoring features". It removes all features whose variance doesn't meet some threshold.





The Chi Squared feature selection is chosen for running against two datasets, which provides a clear separation of the samples. And it performs better than other three feature selection algorithm. Since comparing the feature selection and transformation methods is not fair under the awareness of classes in used for feature selection. The selected and transformed features can be selected as the best K, which will produce better classification. In this method, we need to worry about overfitting problem, which is one of drawback of supervised feature selection method.

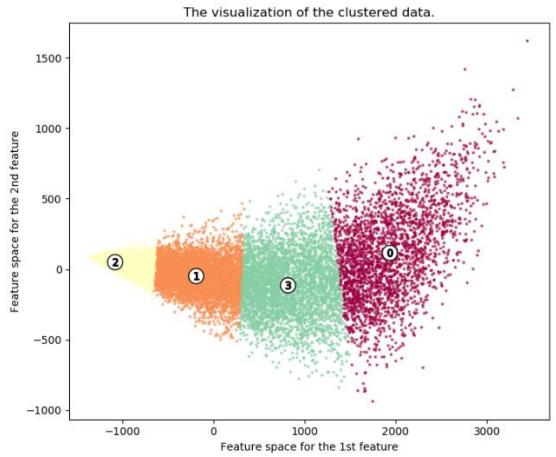
(g). Dimension Reduction

n_clusters	NMI score	average silhouette_score	n_clusters	NMI score	average silhouette_score
2	0.358616	0.593782	2	0.024817	0.511715
4	0.405528	0.477904	4	0.025186	0.37715
6	0.395672	0.41627	6	0.030188	0.313812
			8	0.029908	0.31902

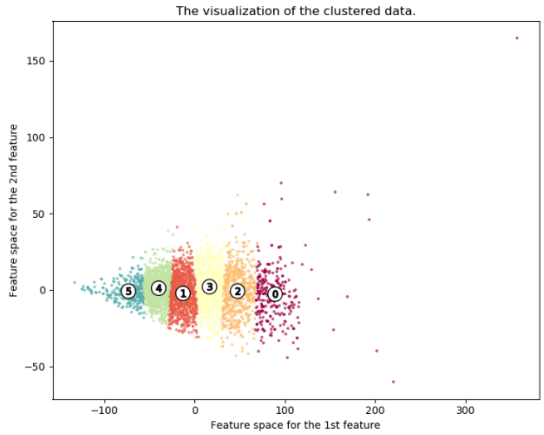
n_clusters	NMI score	n_clusters	NMI score
2	0.341201	2	0.003384
4	0.385027	4	0.058427
6	0.387346	6	0.063036
		8	0.066774

PlayerStats:

KMeans clustering using PCA feature transformation with n_clusters = 4

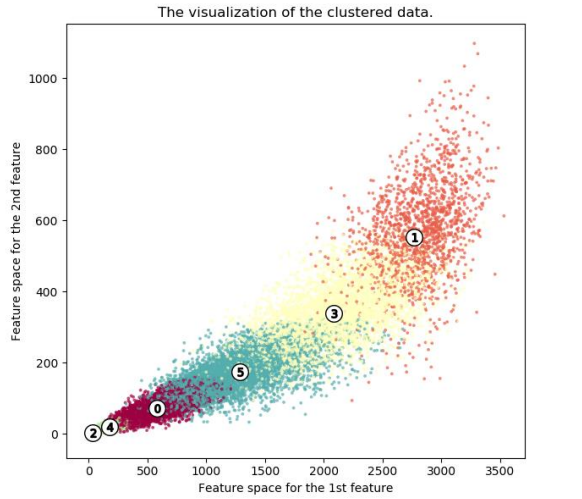


KMeans clustering using PCA feature transformation with n_clusters = 6



WineQuality:

Clusters plot for EM clustering on ICA data with n_clusters = 6

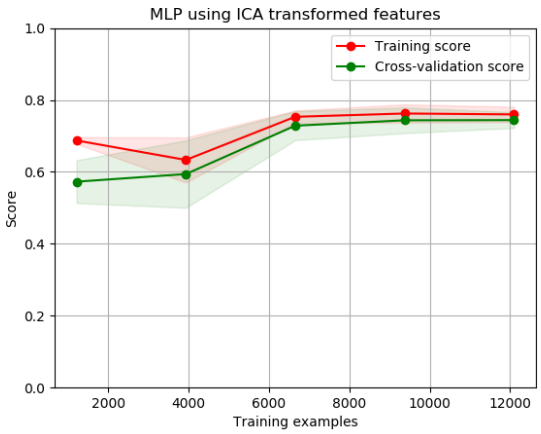


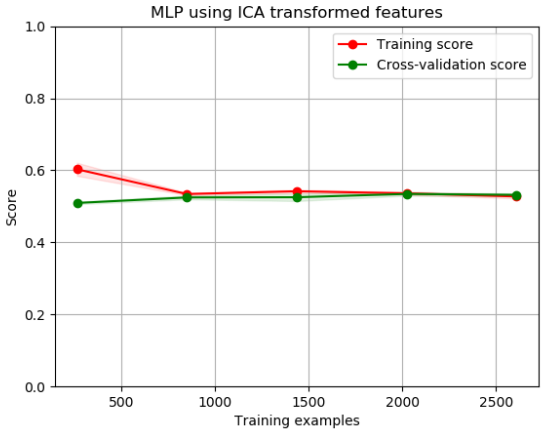
Clusters plot for EM clustering on PCA data with n_clusters = 8

Here we reproduce clustering algorithm after dimension reduction algorithm, we can see optimized results from those better scores obtained. For K means clustering, PlayerStats works best on K=4 setting and WineQuality works best on K=6 setting. For Expectation Maximization, PlayerStats works best on K=6 setting and WineQuality works best on K=8, large K. The NMI scores did improve a lot comparing to the raw feature values for PlayerStats, but there is some dropdown for WineQuality when comparing to simple min max normalized features.
From the plots we can see the separation performs much better than raw feature values. In the following figures, we only selective present the best separation for feature transformation. The best feature transformation running in K means clustering algorithm provides the best visualization is PCA for both dataset demonstrated below, the best feature transformation running in expectation maximization clustering algorithm provides the best visualization is ICA for PlayerStats, PCA for WineQuality. Other plots are available in project's plots folder.
The separation results we can see in the scatter plots can't be used as a deterministic evaluation of the clustering results. To compare the results with previous clustering, all 4 features transformation take care of the extreme values in the features and can provide a more unbiased clustering than the min max normalized result.

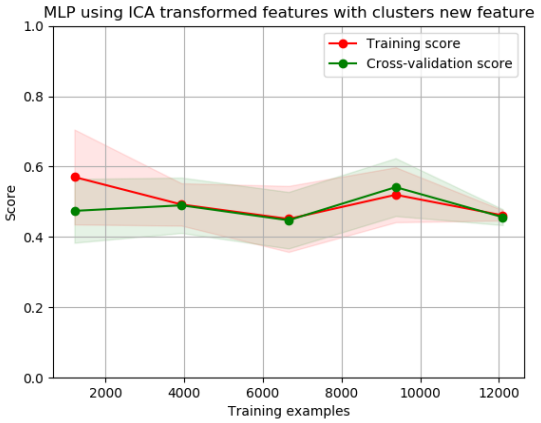
(h). Rerun your neural network learner on dimension reduced dataset
When we rerun the neural network, it's noticeable that ICA provides some very good independent components space for all the clustering methods. I think the reason why ICA perform better is due to the independence relation between the features, ICA can extract that independent information better than other ranking based methods. Using ICA feature reduction produce surprising high-performance result. Other algorithms and feature reduction method

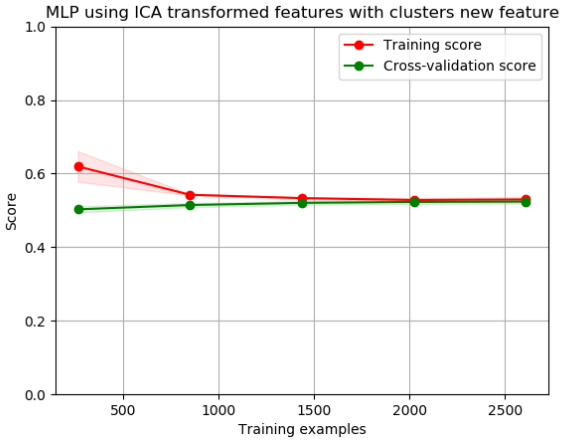
combination can achieve around 0.5-0.6 in 6-8 classes problem. Comparing with my previous ANN result of around 0.6 optimal accuracy, these results show significant improvements. My understanding is that ANN can't handle the extreme values of that one feature in my previous runs it will cause overfitting problem as well, the feature reduction is significantly good at dealing with these problems easily.
PlayerStats:





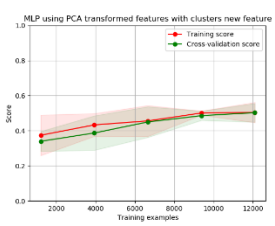
WineQuality:

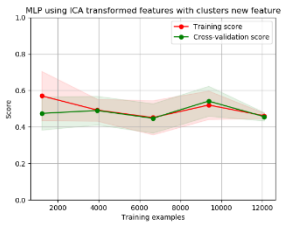


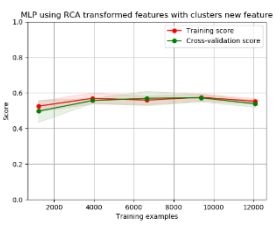


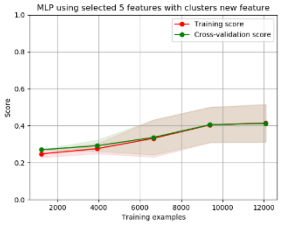
(i). Rerun your neural network learner on dimension reduced dataset with the clustering algorithms
For this experiment, we rerun the neural network learner on dimension reduced dataset, it converges a better result. K-means clustering results from each transformed data set were added to the features data frame. And the same ANN structure was used to train different models. The result of ICA was the same and converge at the same highest CV score among other feature reduction algorithms. And the overfitting situations of PCA and RCA results were not as bad as before. But the result of selected 5 features data is not as good. The bad result might be due to the added complexity of this feature. For PlayerStats, there might be many attributes involved, after extensive feature reduction it loses other important indicators, looking at select 5 features for PlayerStats, it probably lose many other good attributes since PlayerStats has 37 features, reducing 32 features will hurt the performance significantly. For WineQuality, the clustering process is imperfect, so the result is not expectedly high, it's closed to the neural network without any clustering algorithms. There could be some improvement to adjust the data preprocessing for better scale instead of min max scaler before the clustering process, it could make the training neural network more accurate for its prediction.

PlayerStats:

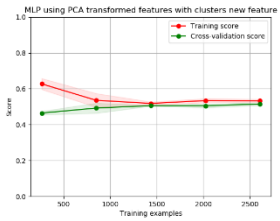


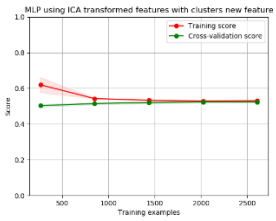


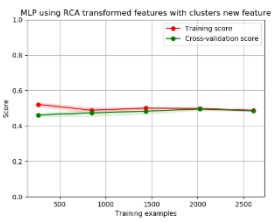


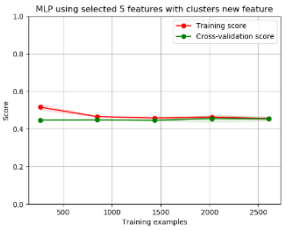


WineQuality:



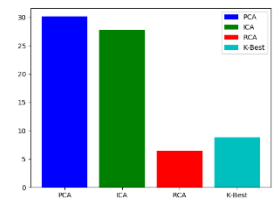


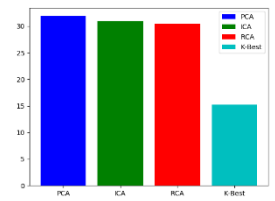


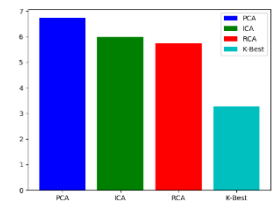


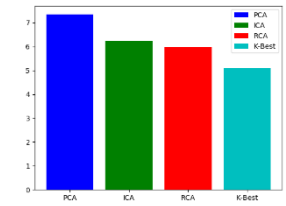
(j). Running Time Performance Analysis

Running Time for clustering + feature reduction is relatively faster than randomized optimization respect to size of dataset. It seems that the running time is linear to its size of dataset's attributes, or the coefficient of n_components. The left two (K-Means, Expectation Maximization) is running time for PlayerStats and the right two (K-Means, Expectation Maximization) is running time for WineQuality in each feature reduction algorithm. To compare their running time, PCA is normally the most expensive algorithm to compute due to calculation of the covariance matrix which is O(p^2*n). ICA computational time is similar with PCA. Random Projection computes better in K-Means but worse in Expectation Maximization, execution time of the algorithms is reduced in lower dimensions. Selecting the best 5 feature reduction is the most efficient.









5. Conclusions, Discussion and Improvement

In this assignment, there is some data-cleaning and much more data preprocessing required to be done before putting those data into algorithms. The output values are normalized to be 1 at the last few columns.

k-means clustering, Expectation Maximization, then implement other four dimensionality reduction algorithms such as PCA, ICA, Randomized Projections and Univariate feature selection (K best) algorithm are unsupervised learning algorithms. They can be utilized for very complex and structured functions, as well as be utilized for their efficiency in optimization problems. As this experiment proceeds, different types of dataset (many attributes, many rows, normally distributed dataset) performs differently in time and complexity.

In my opinion it is likely the best to evaluate the difference of dataset, run sufficient experiments according to those datasets and compute the best accuracy, then optimize the computational time. For our case, K-Best always performs faster but the accuracy is crucially low in PlayerStats so we would like to use ICA. However, WineQuality does not vary a lot in performance due to its small attributes and size of data, we should evaluate other factors of feature reduction algorithm. Feature selection is very useful method in machine learning field and it is normally applied in unsupervised learning, which is unlike supervised learning, overfitting problem will not be the drawback.

In conclusion, this assignment presents many plots and available in project plot folder, it gives me a deep understanding of how Unsupervised learning algorithm works, how feature selection algorithm optimizes the result, more importantly, it also improve my skills of manipulating large set of data, visualize data in a meaningful way in Python.