CS 7641 MACHINE LEARNING
ASSIGNMENT 3: Unsupervised Learning

NAME: NIHAR MEHTA
GT USERNAME: nmehta80

A. Description of Problems
1. PULSAR STARS
Link: https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star

This is one of the previously used datasets. This dataset contains 18000 examples of stars and the objective is to identify whether the star is a pulsar star or not. The features of the stars provided in this dataset are statistics of the integrated profile and DM-SNR curve.

Pulsars are a rare type of Neutron stars. They can produce radio emissions detectable here on Earth. They are of considerable scientific interest as probes of space-time, the inter-stellar medium, and states of matter . Pulsars produce a detectable pattern of radio emission which is periodic in nature. These patterns depend on the rotation of the pulsar. These detections are averaged over many rotations of the pulser. The first four attributes in the dataset are simple statistics obtained from the integrated pulse profile (folded profile). The remaining four variables are similarly obtained from the DM-SNR curve . These should be sufficient to categorize a star as pulsar or not.

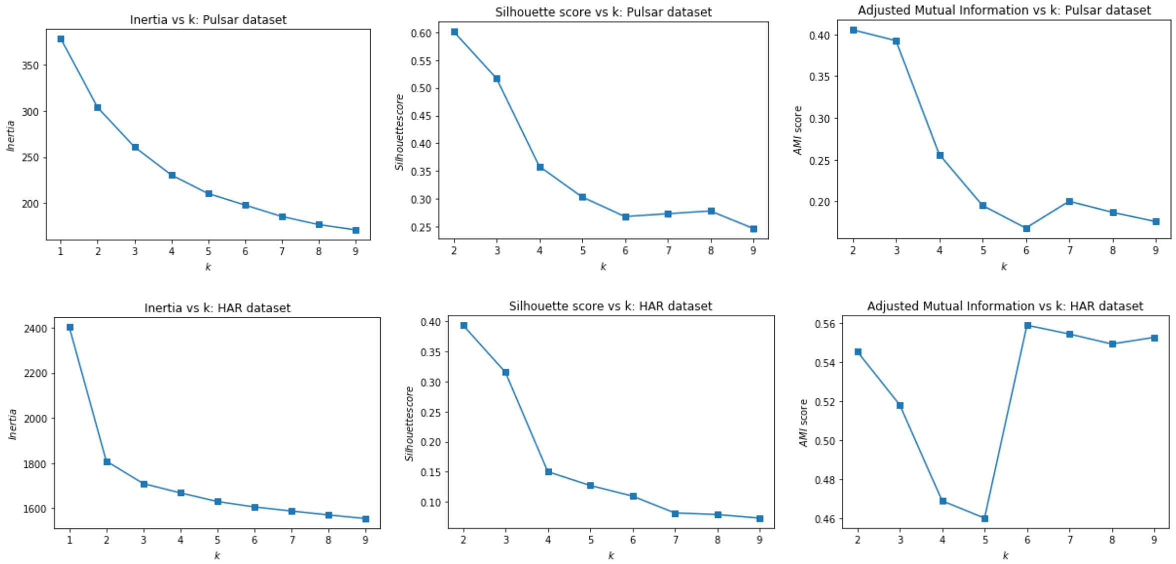
This dataset has a class imbalance since the legitimate pulsar examples are a minority positive class, and spurious examples the majority negative class. It contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators. It will be a good analysis to compare the performance of various models on such an imbalanced dataset.

2. Samsung Human Activity Recognition (HAR) Dataset
Link: https://www.kaggle.com/kashnitsky/a7-demo-unsupervised-learning/notebook
Challenge: https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones/

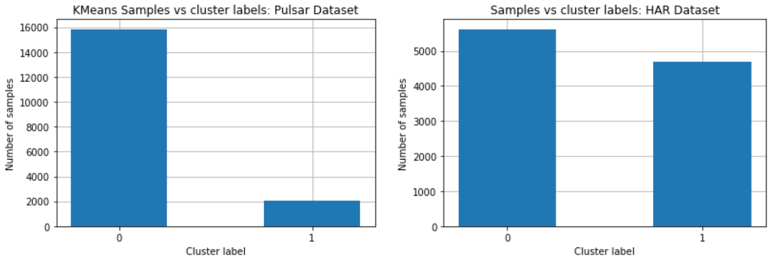
This dataset hasn't been used in previous assignments. This dataset consists of sensor recordings of 30 volunteers of various day-to-day activities while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to model the sensor readings to the activities. These activities include walking, walking upstairs, walking downstairs, sitting, standing and laying. Thus there are 6 output classes. Noise filters and sliding window sampling was performed as preprocessing of readings from accelerometer and gyroscope. From each window, a vector of 561 features was obtained by calculating time and frequency domain variables.

1. **K-Means Clustering**

K-Means clustering divides the data into k clusters by assigning each point to the cluster mean which is nearest to the point. It is an iterative algorithm which alternates computing means of clusters and reassignment of points to the clusters. It is prone to local minima and hence, as a workaround, the algorithm is repeated for numerous initializations of cluster means. Here, we have repeated it for 100 initializations which is done by taking 100 different centroid seeds. For analysis, three scores have been used: Inertia, Silhouette and adjusted mutual information. Inertia is sum of squared distances to the closest center. As k increases, we observe that the inertia decreases. This is an expected behavior because more the number of clusters, more localized the points will be and when k is equal to the number of points, each point will be its own cluster and inertia will be 0. Hence, inertia will always decrease. A good method of choosing an optimum k is the elbow method. Whenever the inertia vs k graph forms an elbow and the inertia decreases gradually after that, it is the optimal k. In our analysis, we see that for the Pulsar dataset, the major dip is at k = 2 and the dips following it are not quite high. So k = 2 is optimal number of clusters for the Pulsar dataset, which reflects the classes Pulsar and Non-Pulsar. For the HAR dataset, we find that there is a major elbow at k = 2 and then the change in inertia is very slow. This reflects the major classes: Walking (upwards, downwards and forward) and Not-Walking (Standing, sitting and laying). Silhouette score represents how close points in one cluster are close to points in the other cluster. Thus higher values of silhouette scores are better as we want maximum separation among points in different clusters. Higher the Adjusted mutual information, better is the partition between the clusters.

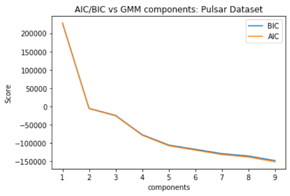


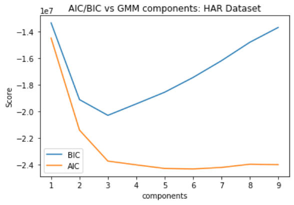
We see that the Pulsar dataset classified 2000 points in cluster 2 and 16000 points in cluster 1. Thus cluster 1 corresponds to non-pulsar and cluster 2 corresponds to pulsar. For HAR dataset, the cluster 1 represents walking and cluster 2 represents non-walking classes.

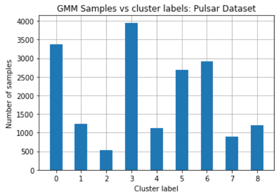


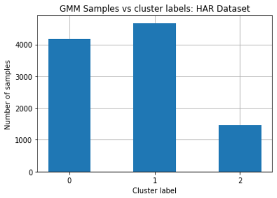
2. Expectation Maximization

Gaussian Mixtures Models (GMM) have been used for clustering the data using expectation maximization algorithm. They assume that the data comes from a distribution of combination of gaussians. The algorithm first assumes random components and then computes the probability of points lying in various clusters and repeats these steps alternatively. To choose the best number of components, Bayesian Information Criterion (BIC) or Akaike Info Criterion (AIC) are used. Lower BIC means higher likelihood and less parameters and thus prevents overfitting too. In the Pulsar dataset, the lowest BIC is for 9 components. Thus, it must be a gaussian mixture of 9 components. Similarly, for HAR dataset, it must be a GMM of 3 components where the BIC is the least.

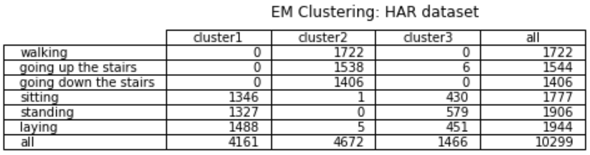








So, GMM by itself hasn't done a great job in categorizing pulsars vs non-pulsars. But for the HAR dataset, we see that cluster 2 represents walking and clusters 1 and 3 are non-walking.



EM Clustering: HAR dataset

	cluster1	cluster2	cluster3	all
walking	0	1722	0	1722
going up the stairs	0	1538	6	1544
going down the stairs	0	1406	0	1406
sitting	1346	1	430	1777
standing	1327	0	579	1906
laying	1488	5	451	1944
all	4161	4672	1466	10299

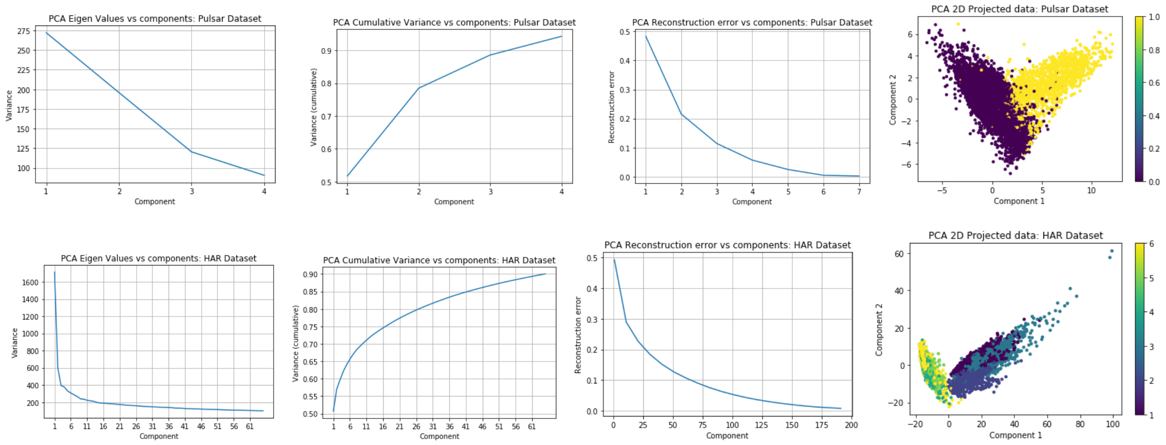
Kmeans is simple, flexible, suitable on a large dataset, efficient, easy to interpret, fast and generates spherical clusters. But number of clusters need to be chosen as it does not return an optimal set of clusters by itself. It lacks consistency and can vary with initialization. It is sensitive to scaling, can handle only numerical data and gives equal weightage to all data points. GMMs are fast. They do not bias the means of the clusters and work very well clusters of any shape. But GMMs tend to have singularities when there are insufficiently many points in a mixture and it diverges while finding the covariance matrix. Also, the algorithm tends to use all the components it has access to.

3. Principal Component Analysis: PCA

Principal Component Analysis is a dimensionality reduction algorithm that transforms a set of observations with correlated features to orthogonal uncorrelated features which are sorted by variance in the dataset. Here, we choose the optimum number of components that explain 90% variance of the dataset as the relevant components. In the Pulsar dataset, we see that only the top 4 components are enough to explain most of the data distribution. SImilarly, in Pulsar dataset, about 65 components are enough to explain the 90% variance in data. Rest 500 components contribute very less to the variance and hence can be neglected. This is how dimension of the data is reduced while retaining most of the information.

	Pulsar	HAR
No of components	8	561
Components that explain 90% variance	4	65
Variance explained by first 3 components	52,27,10	51,6,3

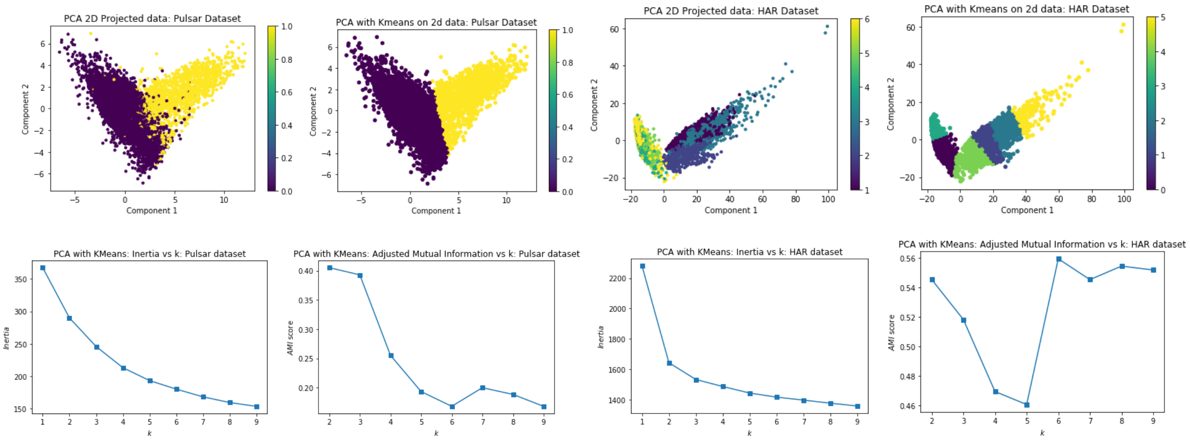
Reconstruction MSE	0.056	0.099



In the above graph, the data is projected on 2d space and is plotted as (component1, component2) and is labelled by its true label. The pulsar dataset does seem quite separable in terms of pulsar vs non-pulsar. The HAR dataset seems to be easily separable in 2 parts: walking and non-walking. These may be subdivided in three parts but just the top 2 components may not be sufficient enough to do so.

A. PCA with KMeans Clustering

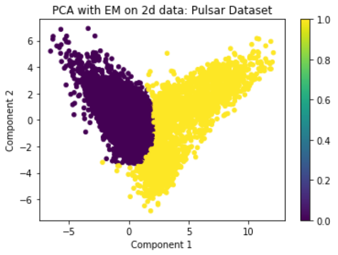
Now, KMeans Clustering has been applied to the reduced dataset by PCA and clusters have been labelled as per the KMeans algorithm. We can see that in Pulsar Dataset, they correspond very well with true labels. In case of HAR dataset too, the clustering does a good job in separating the walking vs non-walking classes. But the internal class differentiation is not as good. This is because KMeans can only handle spherical shaped clusters.

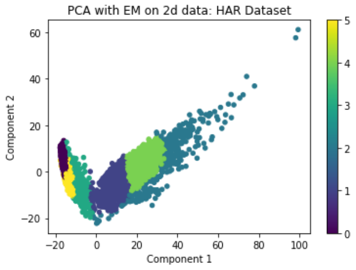


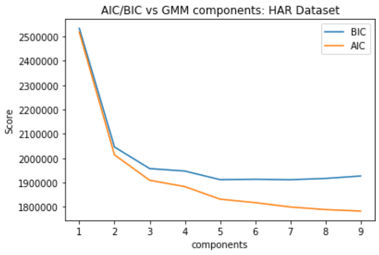
The Adjusted Mutual information graph shows that the optimal number of clusters for Pulsar dataset is at k =2 and at k =6 (lowest AMI) for the HAR dataset.

B. PCA with Expected Maximization

Here, we see that expected maximization has more false negatives as compared to kmeans in case of pulsar dataset. In case of HAR dataset, the clusters are similar to true labels.

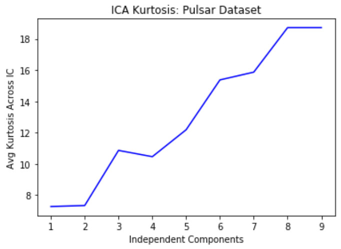


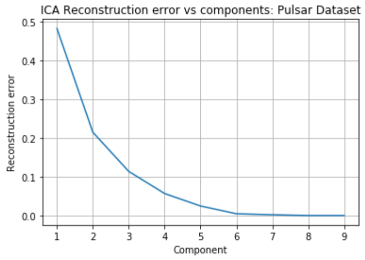


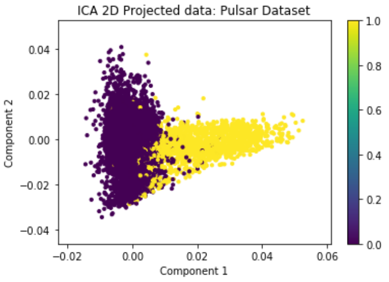


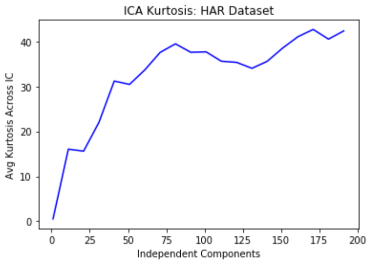
4. Independent Component Analysis: ICA

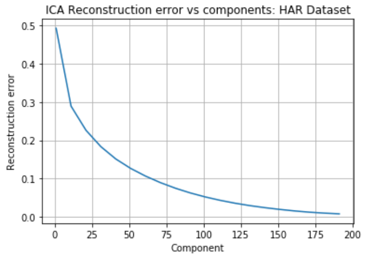
In Independent Component Analysis, the data is assumed as a mixture of non-Gaussian signals which are statistically independent. It tries to retrieve these independent subcomponents. ICA attempts to minimize mutual information and maximize non-Gaussianity. Hence, we see that the clusters of non-gaussian shape can be formed while doing the ICA analysis. Similar observations can be seen in the ICA 2D Projections as well. The clusters can be separated in two parts in both datasets. The subdivisions in HAR dataset is hard as there are more than 2 independent components. We use Fast-ICA for our analysis. Kurtosis is a measure of the non-gaussianity of the components and hence the independence of the components. So, we use it to compute the number of independent components.

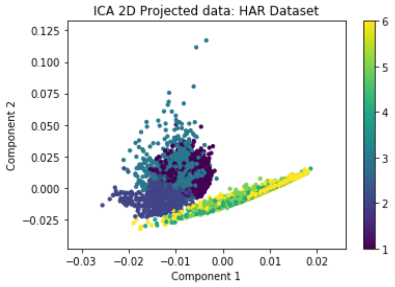










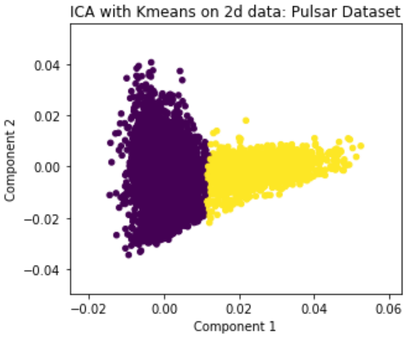


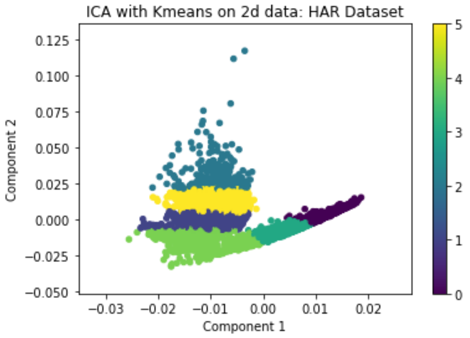
HAR: Optimal number of independent components: 171
ICA Reconstruction mean squared error for HAR dataset: 0.012447770030145184

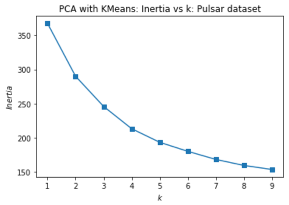
Pulsar: Optimal number of independent components: 8
ICA Reconstruction mean squared error for Pulsar dataset: 3.8870676110317267e-16

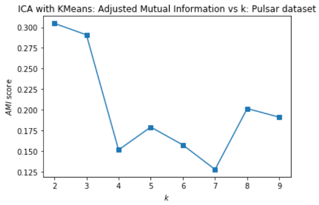
A. ICA with KMeans

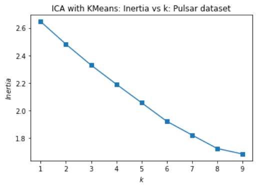
Again KMeans does an excellent job in classifying the pulsar dataset. This time, the classification is better in the HAR dataset too. But since the graphs shown here only consist of 2 components, it is difficult to have a good clustering.

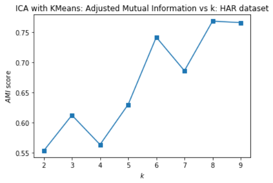

ICA with Kmeans on 2d data: Pulsar Dataset


ICA with Kmeans on 2d data: HAR Dataset

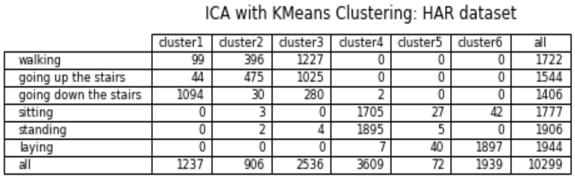

PCA with KMeans: Inertia vs k: Pulsar dataset


ICA with KMeans: Adjusted Mutual Information vs k: Pulsar dataset

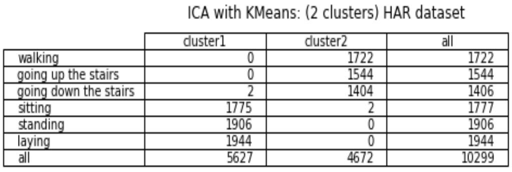

ICA with KMeans: Inertia vs k: Pulsar dataset


ICA with KMeans: Adjusted Mutual Information vs k: HAR dataset

Here we see walking upwards and walking are confused with cluster 3 and similarly, standing and sitting are confused with cluster 4. But walking and non-walking classes are separated perfectly

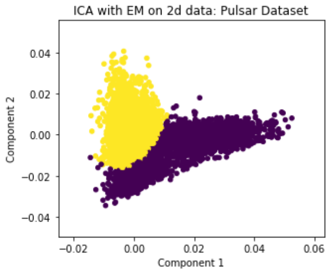

ICA with KMeans Clustering: HAR dataset

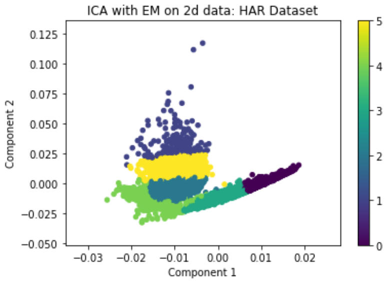
	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	all
walking	99	396	1227	0	0	0	1722
going up the stairs	44	475	1025	0	0	0	1544
going down the stairs	1094	30	280	2	0	0	1406
sitting	0	3	0	1705	27	42	1777
standing	0	2	4	1895	5	0	1906
laying	0	0	0	7	40	1897	1944
all	1237	906	2536	3609	72	1939	10299

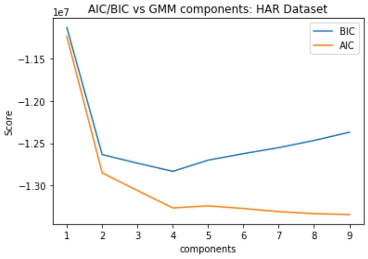

ICA with KMeans: (2 clusters) HAR dataset

	cluster1	cluster2	all
walking	0	1722	1722
going up the stairs	0	1544	1544
going down the stairs	2	1404	1406
sitting	1775	2	1777
standing	1906	0	1906
laying	1944	0	1944
all	5627	4672	10299

B. ICA with Expectation Maximization


ICA with EM on 2d data: Pulsar Dataset

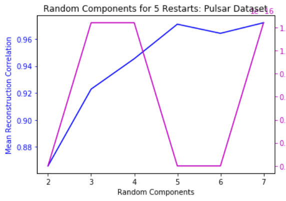

ICA with EM on 2d data: HAR Dataset

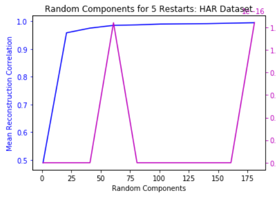

AIC/BIC vs GMM components: HAR Dataset

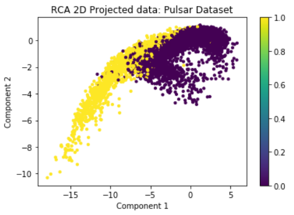
ICA does a good job along with EM algorithm. It maybe due to the fact that the independent components are already non-gaussian in nature and hence the EM algorithm finds it easy to cluster the points based on the independence in the distribution.

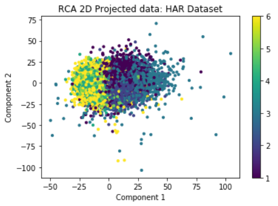
5 Random Projections: Random Component Analysis

In RCA, a random k x d matrix is used to project the data from d-dimensions to k-dimensions and thus perform dimensionality reduction in a random fashion. Here, we have used Gaussian Random projection, where the matrix is generated from a gaussian random distribution. It has orthogonality, spherical symmetry and normality. It is known to preserve the distances well but the empirical results are sparse. This is evident from the results where we see that points with same label may be far away from each other and points very close to each other may have different labels. Thus, we see that the actual clusters are not pretty much disjoint and look distributed rather than localized. This has been repeated for multiple restarts. Mean and standard deviation of the reconstruction correlation has been plotted with random components. In Pulsar dataset, best reconstruction is observed in 5 components while in HAR dataset, reconstruction is best at 171 components like ICA.

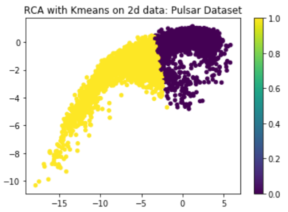

Random Components for 5 Restarts: Pulsar Dataset

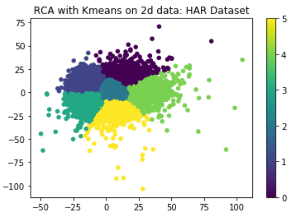

Random Components for 5 Restarts: HAR Dataset

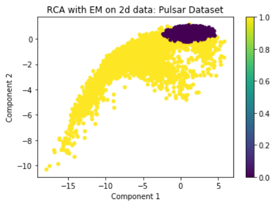

RCA 2D Projected data: Pulsar Dataset

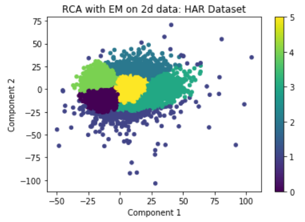

RCA 2D Projected data: HAR Dataset

A. RCA with KMeans and Expectation Maximization

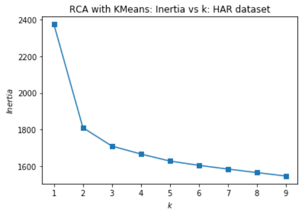

RCA with Kmeans on 2d data: Pulsar Dataset

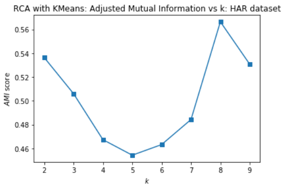

RCA with Kmeans on 2d data: HAR Dataset

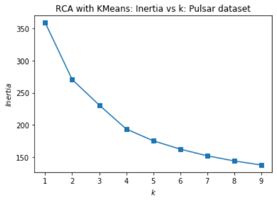

RCA with EM on 2d data: Pulsar Dataset

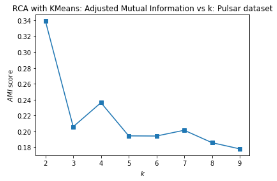

RCA with EM on 2d data: HAR Dataset

RCA performs very poorly on the HAR dataset but does a good job on the Pulsar dataset. The randomized projection results in poorly projected points in the HAR dataset.


RCA with KMeans: Inertia vs k: HAR dataset


RCA with KMeans: Adjusted Mutual Information vs k: HAR dataset

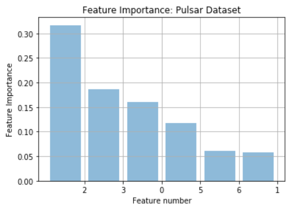

RCA with KMeans: Inertia vs k: Pulsar dataset

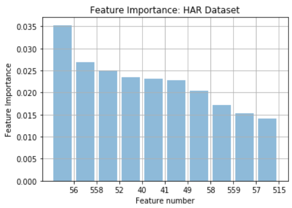

RCA with KMeans: Adjusted Mutual Information vs k: Pulsar dataset

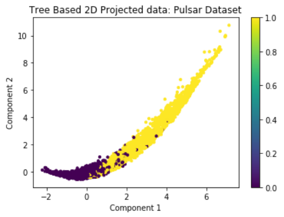
This time, both Pulsar and HAR datasets have not been clustered well. There are a lot of false negatives in the Pulsar dataset.

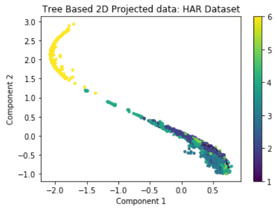
6. Tree based feature selection

Random forests can be used for feature selection. They consist of multiple trees which are built over random samplings of data points and features. These trees do not see all features or all data points at once and hence are immune to overfitting. This also ensures that the trees are uncorrelated. The importance of each feature is reflected by the information gain at each node which is the change in the entropy/purity of data points due to the split at the node. After the random forest is built, we have a list of feature importances. After this, all those features are selected which are above the mean feature importance. These are the features that carry important information which is used to distinguishing between the classes. Rest of the features do not contribute much for the classification and hence can be discarded. The feature importances are plotted.





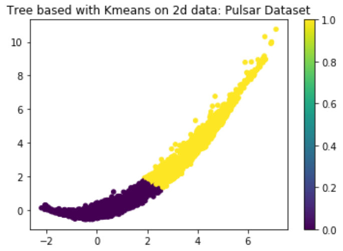


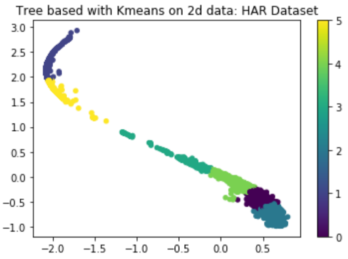


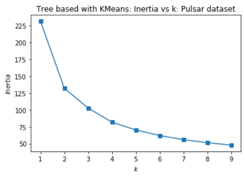
Since there is already an information about the labels that has been used to select the features, the projected data looks well separable.

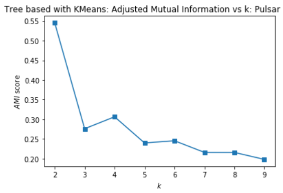
A.Tree based with KMeans

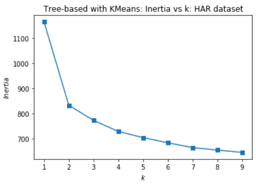
The Tree based feature selection coupled with KMeans does a very good job on Pulsar dataset. It misclassifies some classes in the HAR dataset.

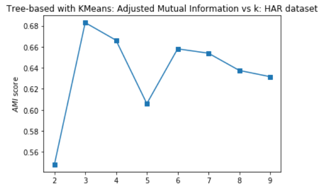




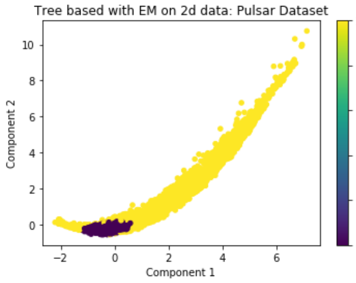


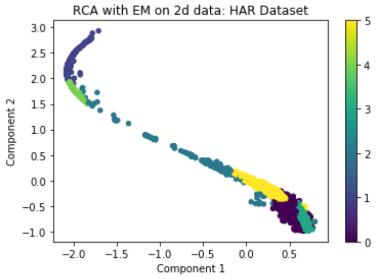


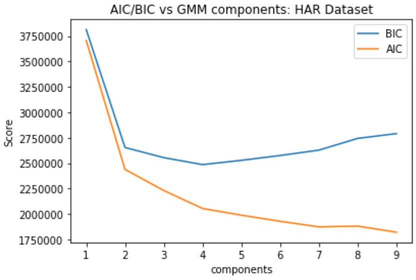




B. Tree based with EM







The clusters with Expectation Maximization algorithms in Pulsar dataset have a lot of false negatives. In the HAR dataset, they seem to cluster well and align with true labels.

Comparison of algorithms

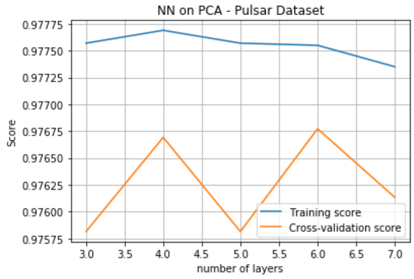
Comparison of the performance of various algorithms on the Pulsar dataset

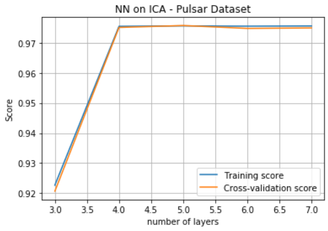
Algorithm	Accuracy	F1 score
KMeans	93.0	69.0
PCA with KMeans	93.6	69.3
ICA with KMeans	91.4	60.2
RCA with KMeans	90.9	62.6
Tree-based with KMeans	**96.0**	**73.0**

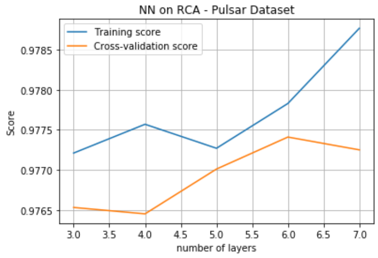
Thus we see that tree based methods outperform most of the other methods. They have relevant information about the feature importances as they have already seen the labels. Only problem with tree based methods is that correlated features are given similar importance. Also, higher preference is given to features with high cardinality.

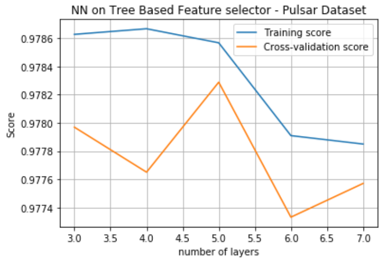
7. Neural Network

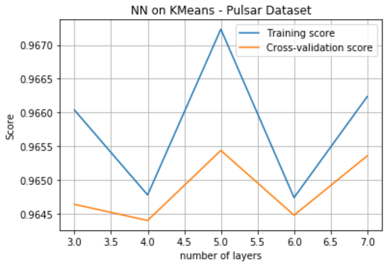
Neural networks have been trained on the datasets obtained after dimensionality reduction algorithms like PCA, ICA, RCA and tree based feature selection methods as well as on the datasets obtained after clustering methods like KMeans and EM algorithm. In each case, the number of hidden layers have been varied from 2 to 7 and the best performing network has been chosen for the analysis. All of these analysis have been done on the Pulsar dataset.

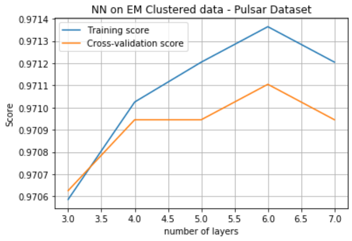












The training and cross validation scores after applying dimensionality reduction are still very good. This shows that dimensionality reduction retained considerable information.

Neural Network was run over various dataset and the performance and speeds are summarised as follows:

Dataset	AUC Score	Training time (s)	Inference time (ms)
Original Dataset	**93**	3.21	328
PCA	90	1.92	52.8
ICA	89	1.51	104
RCA	92	2.79	53.6
Tree based	92	1.39	82.3
KMeans	88	**1.32**	88.6
EM	89	1.86	**30.1**

The neural network performed the best on the original dataset, followed by Tree based,RCA, PCA, ICA and then EM and KMeans. This shows that neural network is smart enough to learn the features by itself and that there is some finite loss of information in dimensionality reduction. However, it is a big gain when we look at training times and inference times. Since the data consists of lesser features now, it is faster to train. The AUC score is still very good even after so many of the features have been reduced which means that the dimensionality reduction/clustering algorithms can prove to be very useful when handling big data. The fastest algorithm in training was KMeans while in inference was EM algorithm.

Conclusions:

1. The clustering algorithms like KMeans and EM also help in dimensionality reduction.
2. Both the clustering as well as dimensionality reduction algorithms retain information even after reducing the number of features
3. The neural network can be speeded up by using unsupervised clustering when the size of each data point is huge.
4. In PCA, the top few components retain the most information and others may be discarded
5. In ICA, kurtosis can be a good measure to decide the number of components
6. Randomized projections also perform well if tried with random restarts
7. Finally, the tree based feature selection method outperform most of the methods as they have information about the labels and know which features contribute to the information gain