

# CS7641 Assignment 3 - Eric Gregori (04/1/18)

## Datasets

### WiFi Localization

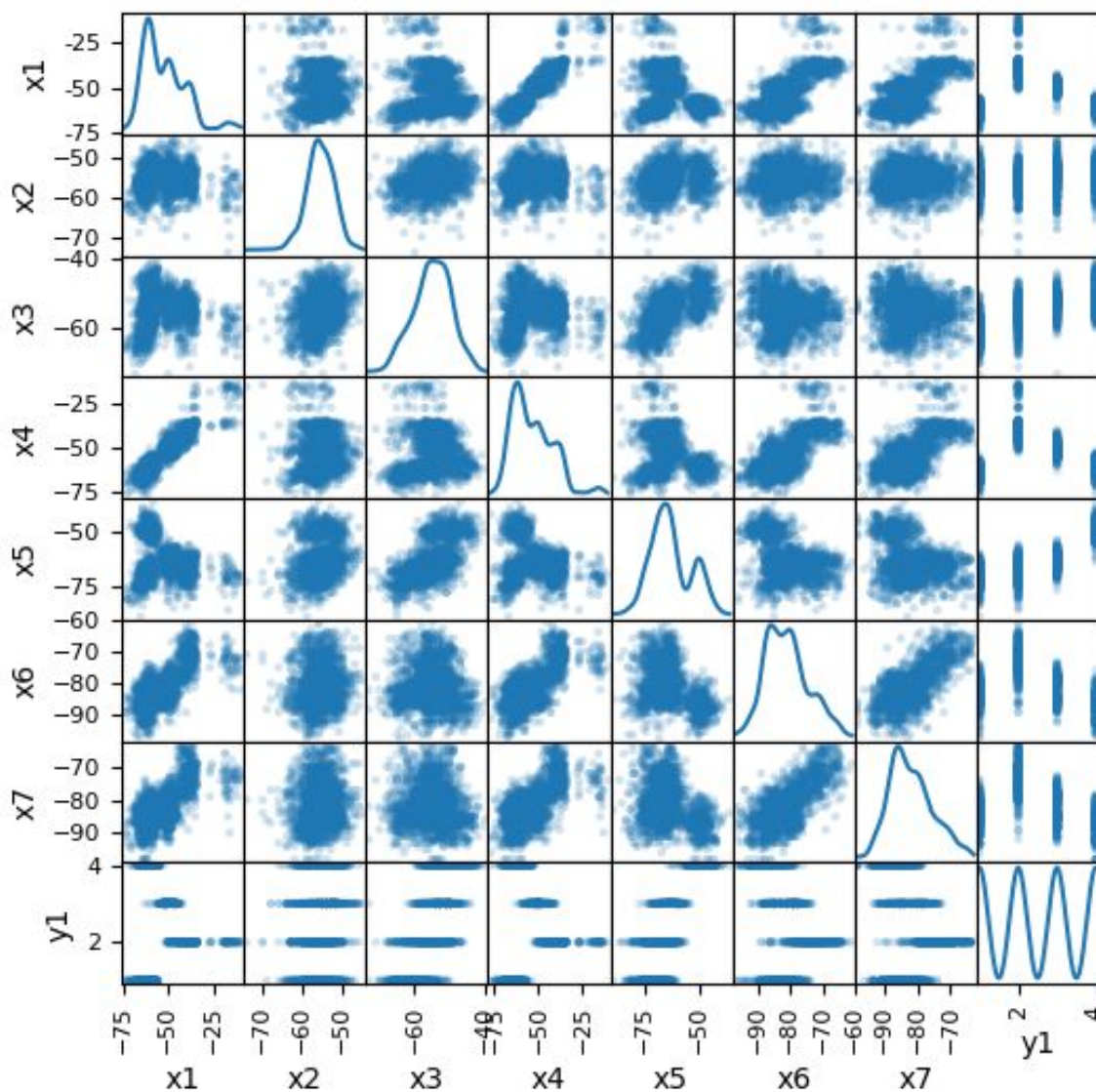
*"Collected in indoor space by observing signal strengths of seven WiFi signals visible on a smartphone. The decision variable is one of the four rooms. Each attribute is wifi signal strength observed on smartphone."*

**File:** wifi\_localization.txt    **Attributes:** 7    **Categories:** 4    **Instances:** 2000

**Attributes are balanced:** min=-98.0, max=-10.0

<http://archive.ics.uci.edu/ml/datasets/Wireless+Indoor+Localization>

wifi



**Figure 1. WIFI dataset scatter matrix**

The scatter matrix for the wifi dataset shows clear clustering (bands in y1 row/column). This dataset should cluster well.

## Letter Recognition

*"Database of character image features; try to identify the letter"*

File: letter-recognition.data

Attributes: 16

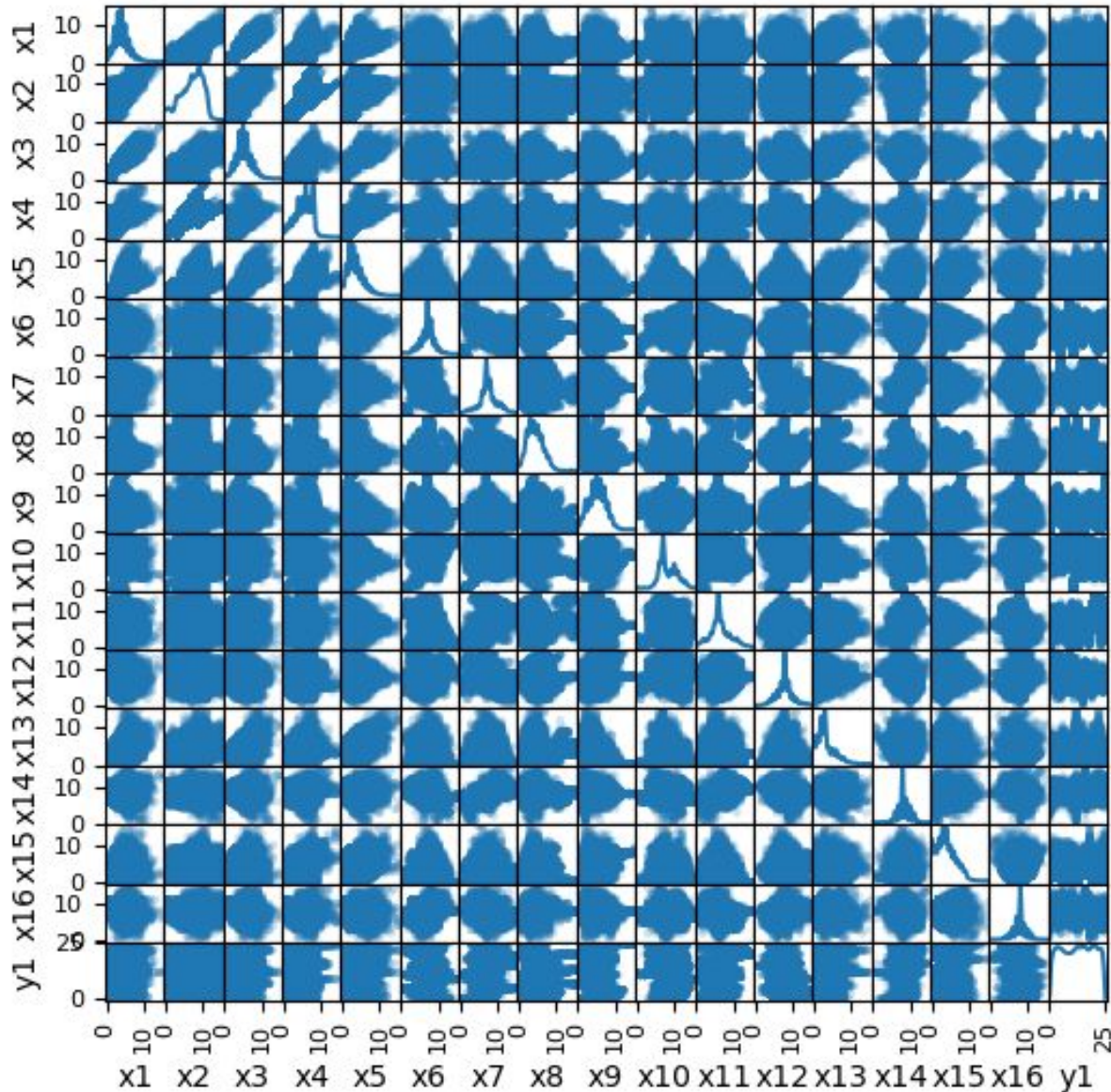
Categories: 26 (letter)

Instances: 20000

Attributes are balanced: min=0, max=15

<https://archive.ics.uci.edu/ml/datasets/letter+recognition>

letter



**Figure 2. Letter dataset scatter matrix**

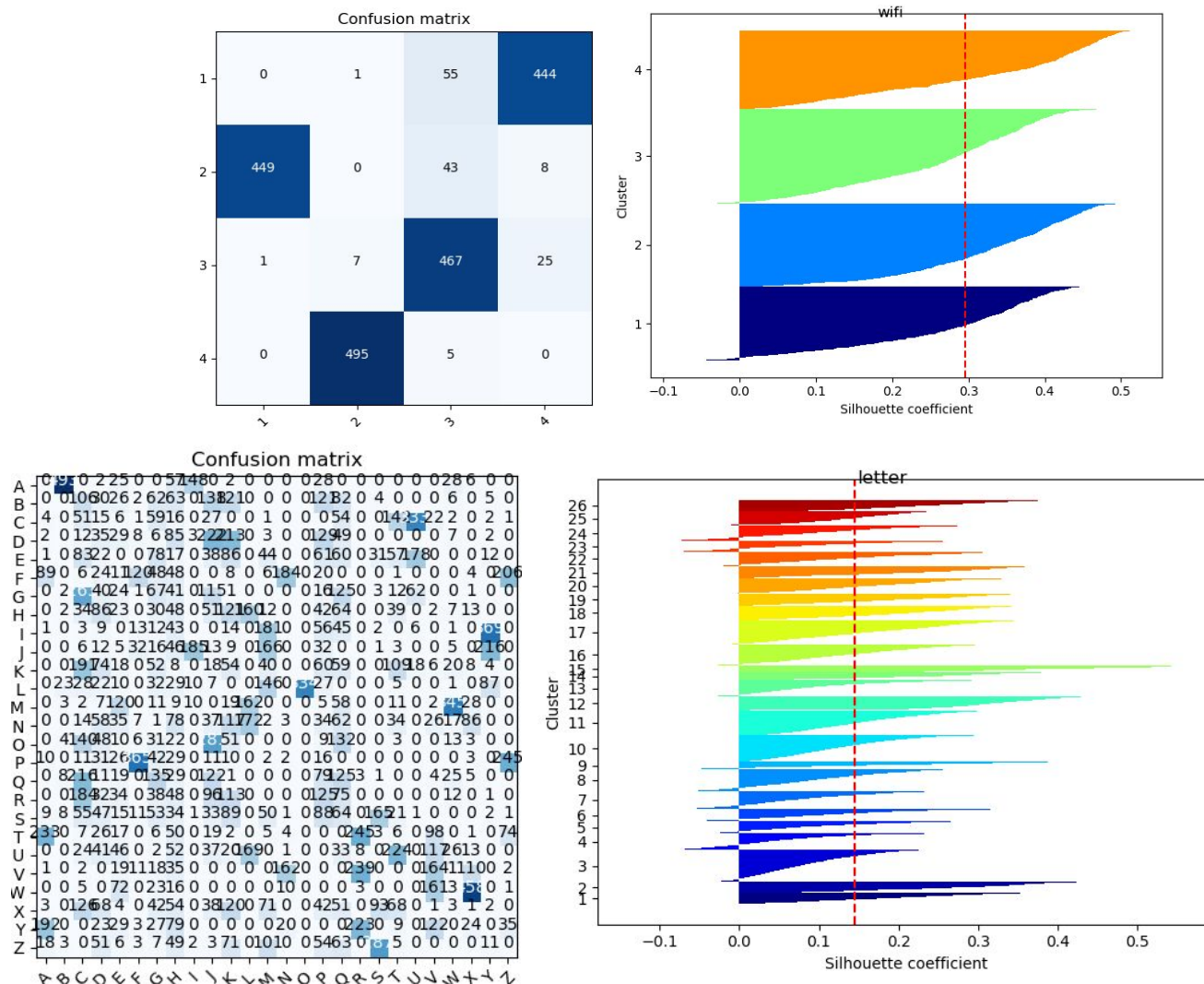
There is no obvious clustering in the letter data set (no clear bands in the y1 row/columns).

### Part 1 - Run EM and K-means on two datasets

k-means and Expectation-Maximization (EM) are unsupervised learning algorithms for clustering data into groups. K-means is discrete, while EM is continuous. EM assigns a probability that a data point is in a group. For example, K-means will assign a data point into group 2, while EM returns a 99.9% probability that the data point is in group 2.



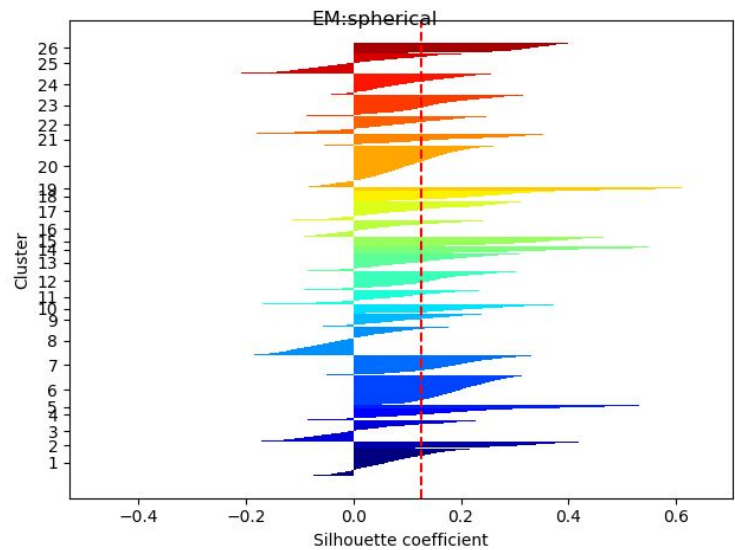
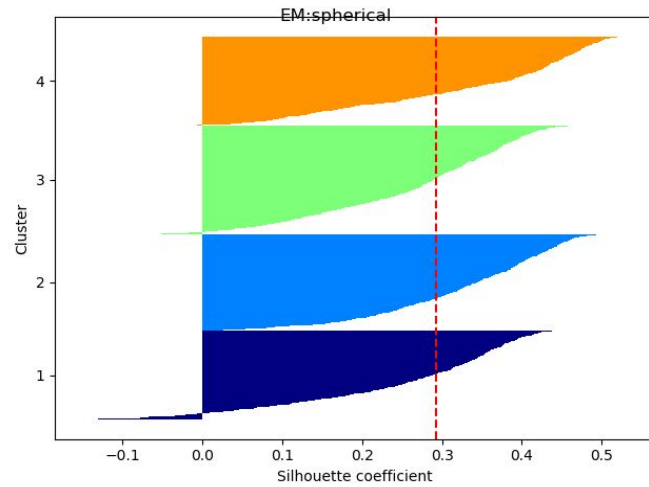
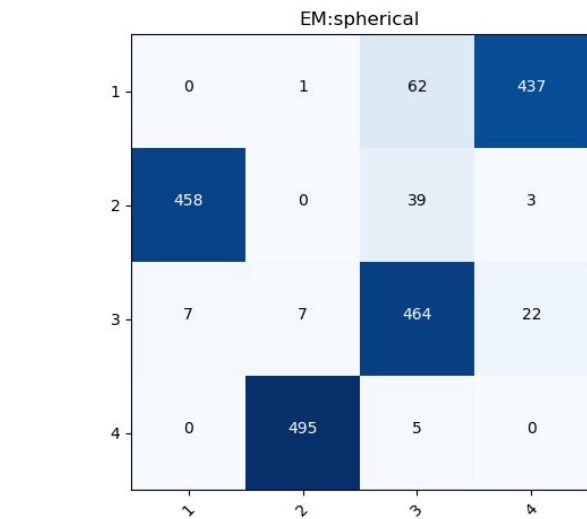
## k-Means (WiFi - top, Letter - bottom)



Running k-means on the wifi and letter dataset using the default sum of squared distance measure. The number of clusters is set to the number of categories which is known a priori. Wifi and letter datasets K-means clustering versus ground truth plotted in confusion matrices. Note the lack of a nice diagonal. Although the k-means labels do not match the ground truth labels, a distinct pattern is apparent in the plots. For the left (wifi) plot, we can see an obvious mapping: (2,1),(4,2),(3,3), and (1,4). Although kmeans did not map to the same labels, it did a good job of clustering common instances. The letter confusion matrix is more difficult to read (literally and figuratively). As predicted by the scatter matrix, the letter dataset did not cluster as well as the wifi dataset. The silhouette plots backup the confusion matrices. The wifi silhouette plot shows consistent size while the letter plot shows varying size. The letter plot shows negative silhouette coefficients indicating poor clustering. [2] One reason for the poor clustering in the letter dataset could be the distance function, another could be a limitation with the K-means algorithm in that it does not work well when there are too many cluster. [1] Gaussian Mixtures, an Expectation-Maximization based algorithm should work better on letter dataset [1].

## Expectation-Maximization(GaussianMixture) - sklearn.mixture.GaussianMixture

While k-means uses the radius of a circle to cluster data, EM uses the variance of a Gaussian distribution. [3] K\_means outputs labels and inertia for each instance, along with the coordinates for the cluster centers. The Gaussian Mixture Model outputs the probability that an instance is in a certain class. Gaussian Mixture Modeling works better than k-means on datasets with a large number of attributes. [1][3]



The gaussian mixture model clusters data in one of four ways: 'full' (each component has its own general covariance matrix), 'tied' (all components share the same general covariance matrix), 'diag' (each component has its own diagonal covariance matrix), and 'spherical' (each component has its own single variance). Only spherical (the best) is shown due to space. EM clustering works no better than K-means clustering with the letter dataset.

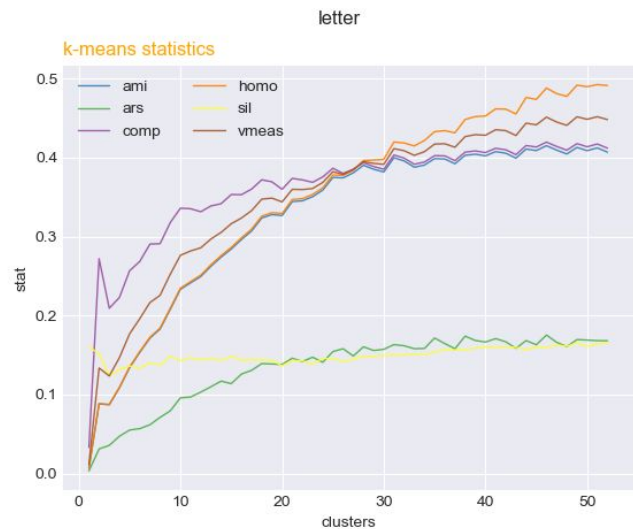
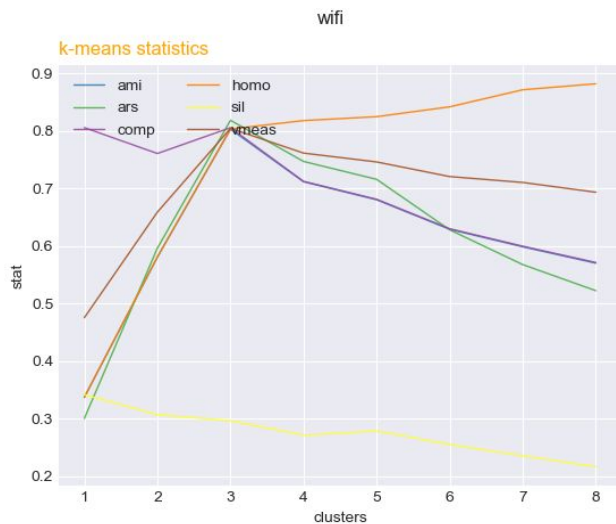
## Summary

Algorithm	time(s)	Algo specific	homo	compl	v-meas	ARI	AMI	silhouette
Kmeans (km++)	0.06	Inertia = 5508	0.803	0.805	0.804	0.818	0.803	0.294
Kmeans(random)	0.05	Inertia = 5508	0.803	0.805	0.804	0.818	0.803	0.302
EM (spherical)	0	AIC=31806, BIC=32002	0.802	0.804	0.803	0.818	0.802	0.293
EM (diag)	0	AIC=28559, BIC=28889	0.940	0.941	0.940	0.953	0.940	0.290
EM (tied)	0	AIC=28524, BIC=28855	0.893	0.896	0.894	0.899	0.893	0.296
EM (full)	0	AIC=26546, BIC=27347	0.924	0.924	0.924	0.944	0.924	0.284

Table 1. WiFi clustering results (4 clusters)

Algorithm	time(s)	Algo specific	homo	compl	v-meas	ARI	AMI	silhouette
Kmeans (km++)	5.49	Inertia = 121205	0.378	0.386	0.382	0.154	0.375	0.118
Kmeans(random)	6.46	Inertia = 121480	0.370	0.378	0.374	0.158	0.367	0.125
EM (spherical)	2.01	AIC=711352, BIC=715043	0.352	0.368	0.360	0.136	0.349	0.126
EM (diag)	2.08	AIC=565819, BIC=572592	0.332	0.362	0.347	0.108	0.329	0.043
EM (tied)	2.03	AIC=620925, BIC=625485	0.443	0.468	0.455	0.191	0.440	0.105
EM (full)	2.07	AIC=319791, BIC=351223	0.435	0.473	0.453	0.158	0.432	0.028

Table 2. Letter clustering results (26 clusters)

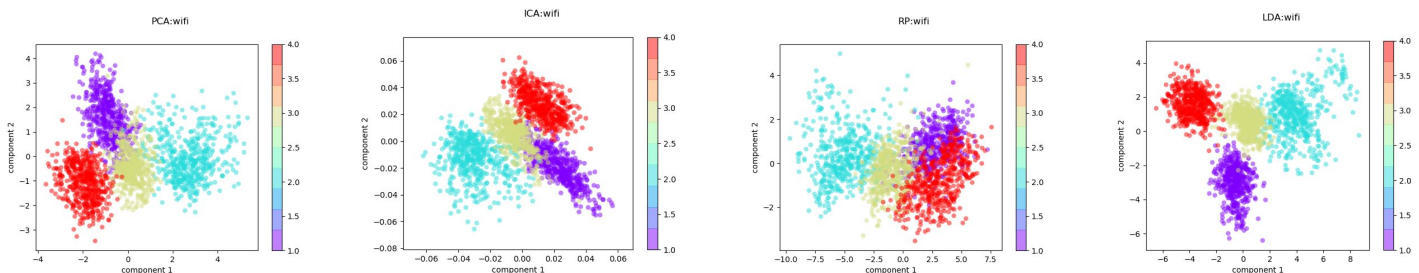


K was chosen based on peak homogeneity, completeness, v\_measure, adjusted\_rand\_score, adjusted\_mutual\_info\_score, and silhouette\_scores as plotted above for the WiFi and letter datasets. The majority of WiFi stats peak at 3 clusters. The completeness, adjusted\_rand\_score and adjusted\_mutual\_info\_score peak at 26 clusters but the homogeneity, v\_measure and silhouette\_score are not as clear.

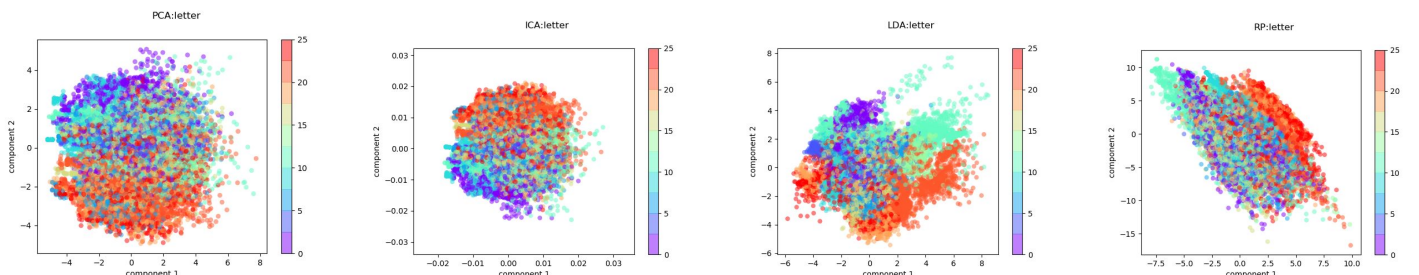
The WiFi data clustered cleanly into 3 or 4 clusters. The clusters line up with the labels. This could be predicted by the scatter plot (figure 1). The letter dataset also appears to naturally cluster around the number of labels (26). The EM algorithm is significantly faster than the k-means algorithm but provides slightly “noisier” clustering based on the silhouette plots. Spherical EM provided the best AIC, BIC, and silhouette scores. This implies that the data is clustered in a Gaussian distribution in multiple dimensions.

## Part 2 - Run PCA, ICA, RP, LDA on two datasets

*“The curse of the dimensionality (term coined by Bellman in 1961) refers to the fact that, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy (i.e. to get a reasonably low-variance estimate) grows exponentially with the number of variables.” “A related fact, responsible for the curse of the dimensionality, is the empty space phenomenon (Scott and Thompson): high-dimensional spaces are inherently sparse.” [4]*



**7 dimension WiFi dataset projected to 2 dimensions for visualization (PCA, ICA, RP, LDA)**

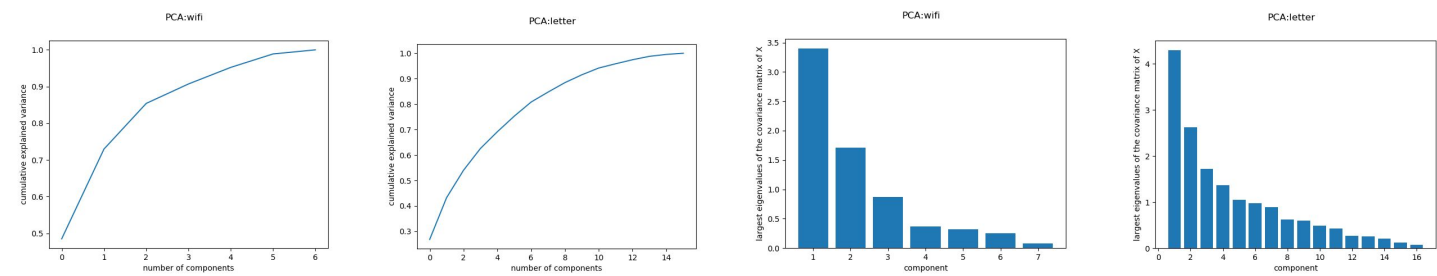


**16 dimension Letter dataset projected to 2 dimensions for visualization (PCA, ICA, RP, LDA)**

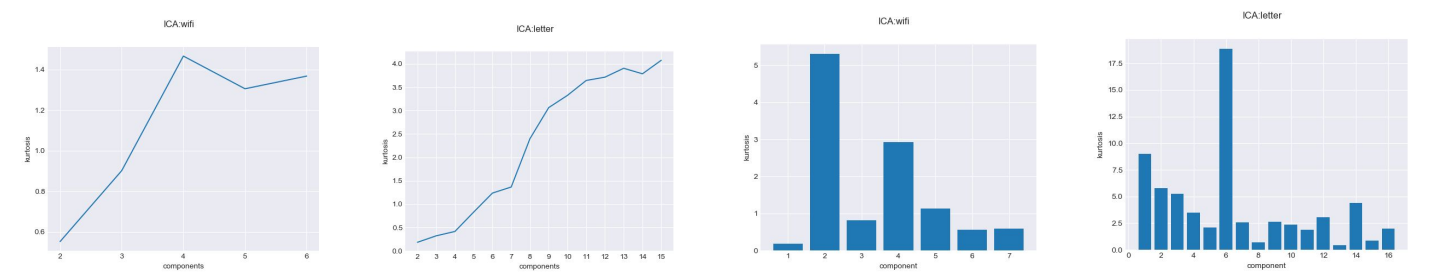
Projecting the datasets into 2 dimensions for visualization shows the spherical clusters in the WiFi dataset clearly.



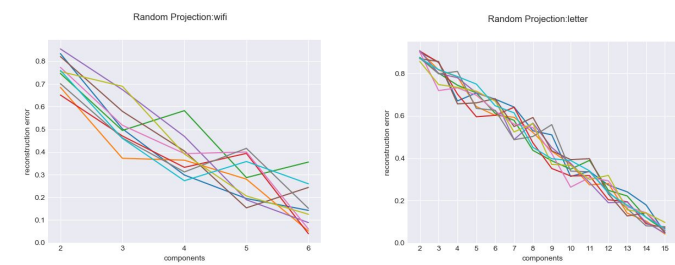
PCA (cumulative explained variance and largest eigenvalues of the covariance matrix of X)



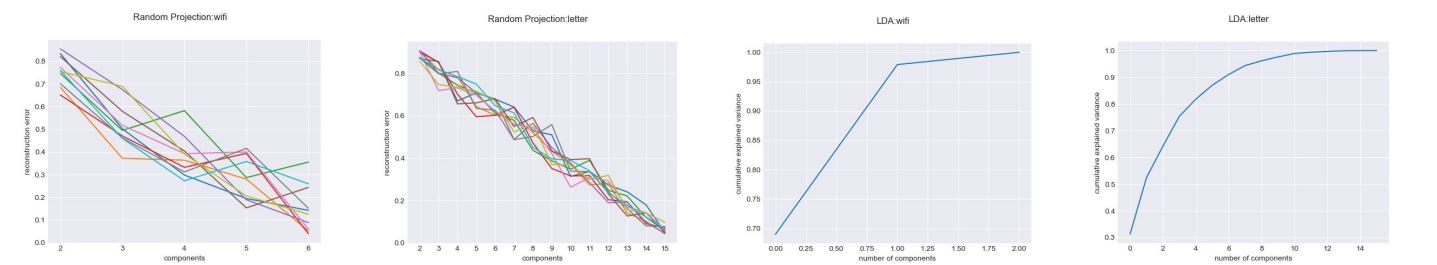
ICA (kurtosis/number of components and kurtosis distribution)



RP (reconstruction error)

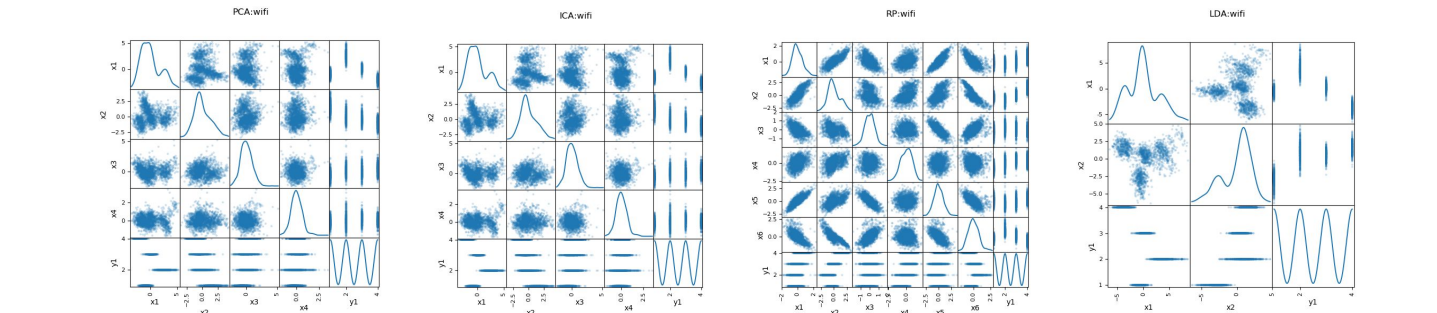


LDA (cumulative explained variance)

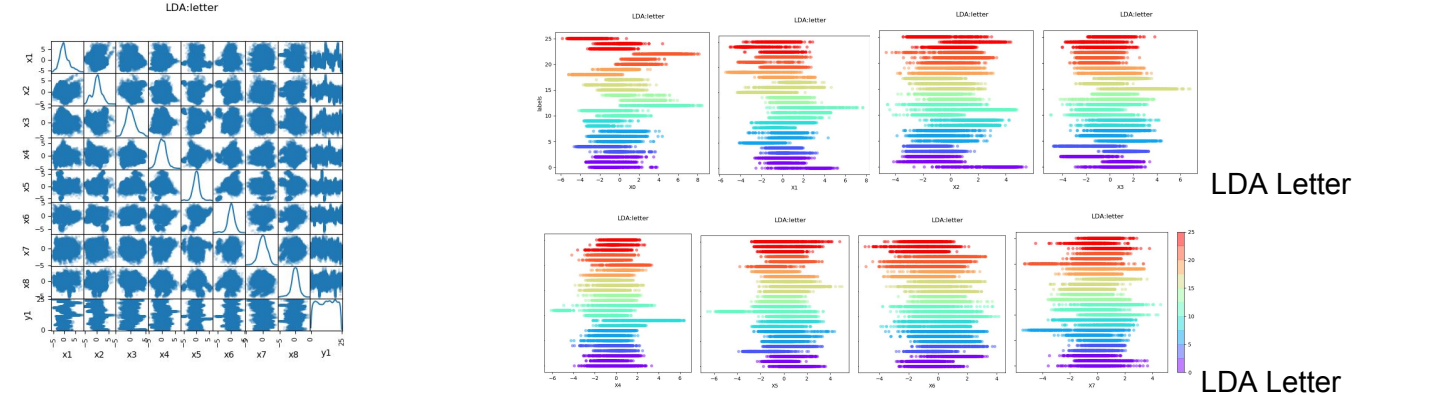


Summary

components	PCA (95% variance)	ICA (kurtosis)	RP(10% error)	LDA(95% variance)
wifi	4	4	6	2
Letter	10	13	14	8



WiFi scatter matrix

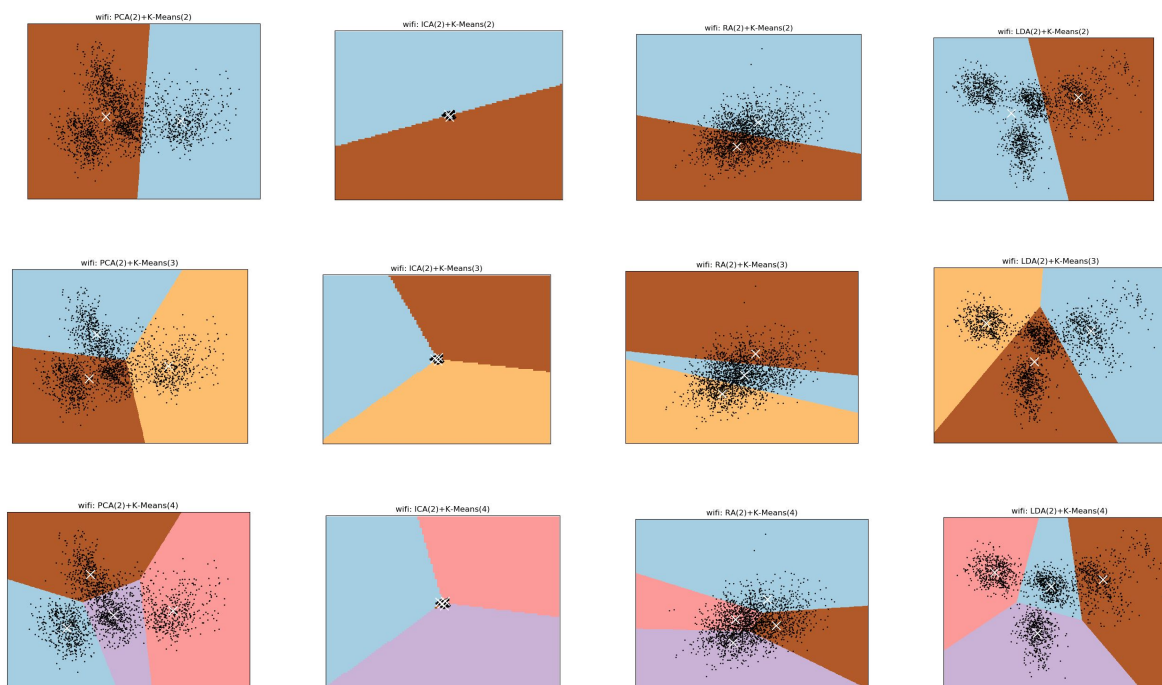


The WiFi dataset 4 clusters are visually obvious when reducing the 7 dimensions down to 2D using all 4 algorithms (PCA,ICA,RP,and LDA). Using the metrics shown, the smallest reduction for each algorithm is shown in the summary table. The diagonal is a KDE plot. Note, the distributions are less Gaussian than the original (non reduced) distributions. This would indicate the data in the remaining dimensions has been “compressed” using more of the space. The RP and LDA scatter matrix plots show some interesting patterns. Diagonal clusters and clean 2D clusters. The details of the letter scatter matrix are hard to see, so the bottom row is plotted separately. The details of the eight LDA generated attributes relative to the labels can be seen. Comparing these attributes to the bottom row in figure 2, the LDA generated attributes contain more unique information with each attribute contributing unique information to a label.

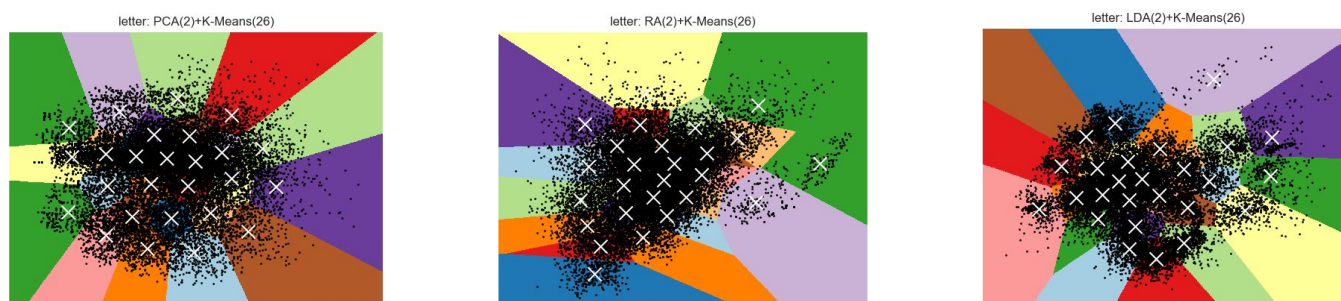
The ICA kurtosis distribution plots show at least one component with significantly higher kurtosis than the rest. Indicating that the component contains lots of unique information. Attempts were made to isolate these components and visualize them, but I ran out of room.

Finally, the RP plots show the reconstruction error from 10 runs. The variance with RP makes it difficult to determine the best number of dimensions.

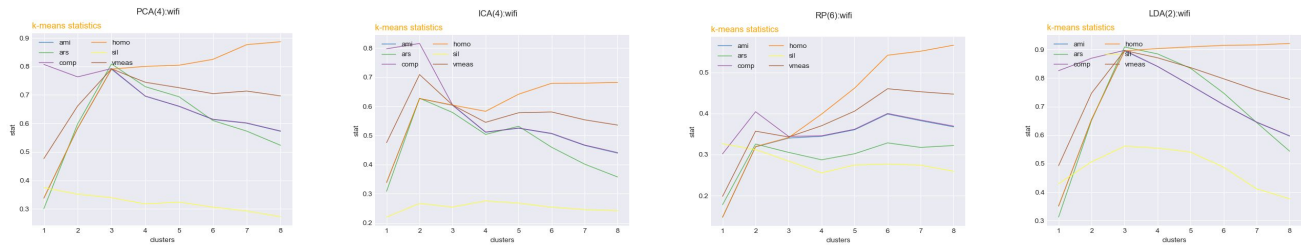
### Part 3 - Run EM and K-means on part 2 results



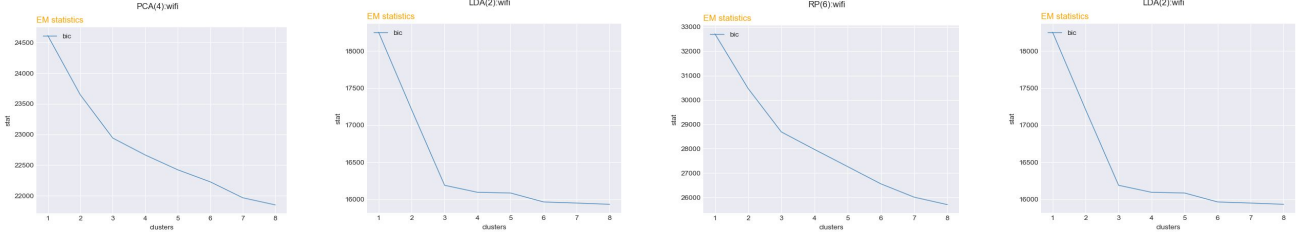
#### Visualizing the WiFi dataset by projecting to 2D then clustering using K-Means (PCA,ICA,RP,LDA)



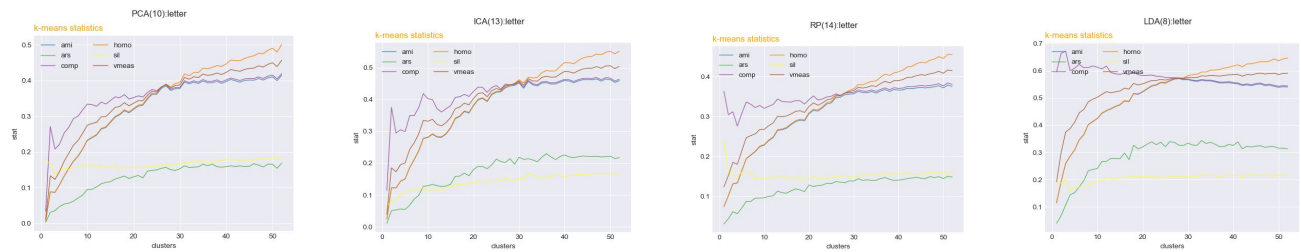
#### Visualizing the letter dataset by projecting to 2D then clustering using K-Means (PCA,ICA,RP,LDA)



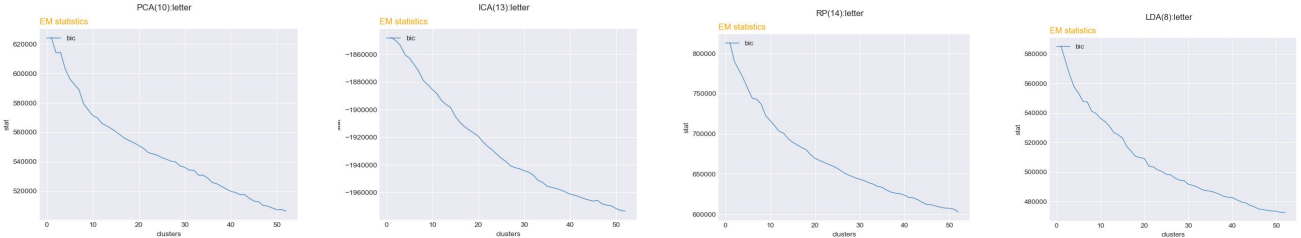
peak homogeneity, completeness, v\_measure, adjusted\_rand\_score, adjusted\_mutual\_info\_score, and silhouette\_scores



EM BIC (wifi)



peak homogeneity, completeness, v\_measure, adjusted\_rand\_score, adjusted\_mutual\_info\_score, and silhouette\_scores



EM BIC (letter)

### K-Means and EM clustering after projecting to fewer dimensions

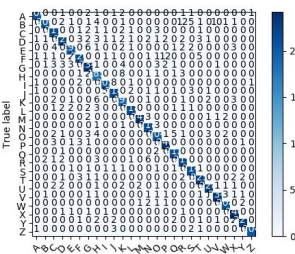
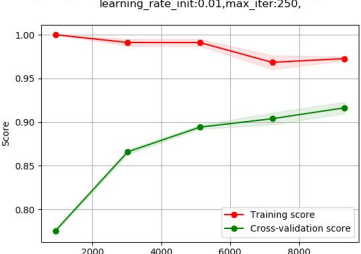
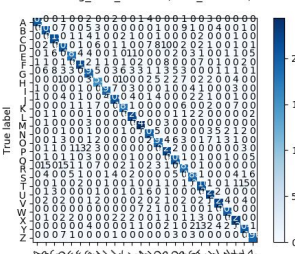

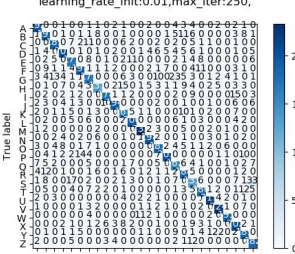
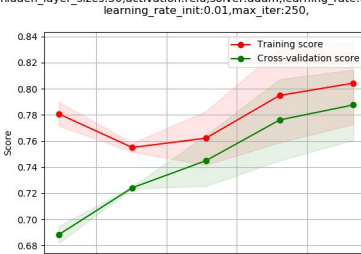
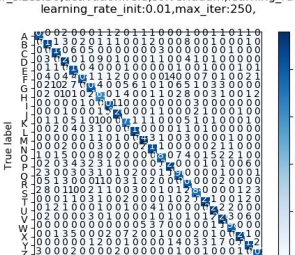

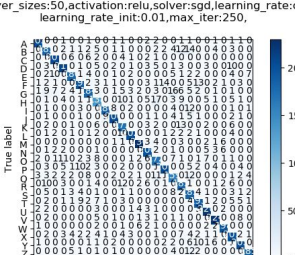
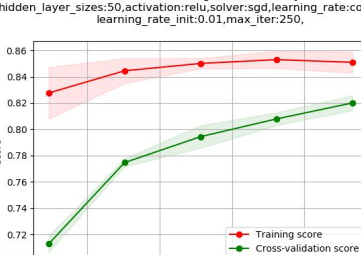
	PDA	ICA	RP	LDA
<b>K-means (WiFi)</b>	3 clusters	2 cluster	6 clusters	3 clusters
<b>EM BIC (WiFi)</b>	3 clusters	3 clusters	3 clusters	3 clusters
<b>K-means (letter)</b>	26 clusters	26 clusters	26 clusters	26 clusters
<b>EM BIC (letter)</b>	10 cluster	20 cluster	20 cluster	20 cluster

In most cases the number of clusters stayed the same. The exception are ICA and RP k-means on the wifi dataset and EM on the letter dataset. The reduced WiFi datasets are shown above. The clusters are very different for each dimension reduction algorithm. PDA and LDA create what appears to be 4 distinct clusters, with the LDA clusters being tighter. ICA and RP create blobs. The clusters are very difficult to see when visualized in the 2D space.

The k-means statistics plots for PCA and LDA reduced WiFi datasets look similar to the same plots for the full (original) dataset. This implies the clusters structure stayed relatively the same. The ICA and RP statistic plots look very unique indicating a very different cluster structure.

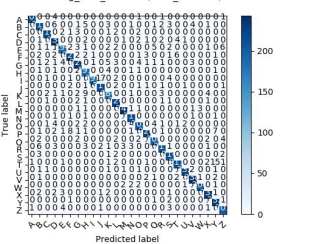
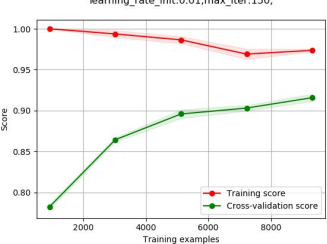
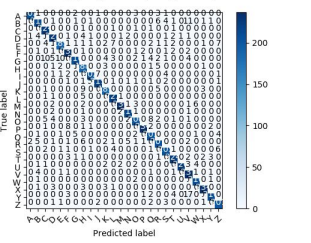
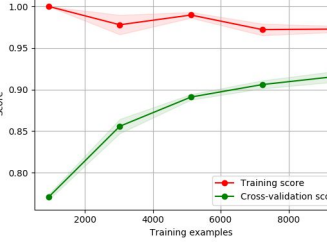
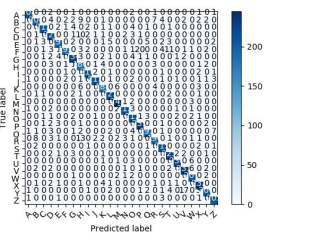
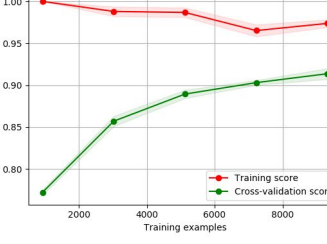


## Part 4 - Take part 2 results (one dataset) and run on ass1 NN

<p>letter:FULL Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>letter:FULL Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>Baseline plots of original letter dataset.</p> <p>High variance</p> <p>91% accuracy</p> <p>Over 250 iterations to converge (slow)</p>
<p>letter:PCA Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:150,</p> 	<p>letter:PCA Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:150,</p> 	<p>PCA reduction to 10 dimensions</p> <p>High variance</p> <p>86% accuracy</p> <p>Less than 150 iterations to converge (fast)</p>
<p>letter:ICA Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>letter:ICA Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>ICA reduction to 13 dimensions</p> <p>High bias</p> <p>Overfitting</p> <p>78% accuracy</p> <p>Over 250 iterations to converge (slow)</p>
<p>letter:RP Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>letter:RP Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>RP reduction to 14 dimensions</p> <p>High bias</p> <p>Overfitting</p> <p>87% accuracy</p> <p>Over 250 iterations to converge (slow)</p>
<p>letter:LDA Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:sgd,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>letter:LDA Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:sgd,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>LDA reduction to 8 dimensions</p> <p>High variance</p> <p>82% accuracy</p> <p>Over 250 iterations to converge (slow)</p>

The PCA learning curves look the best and trained the fastest.

## Part 5 - Take part 1 results (one dataset) and run on ass1 NN

<p>letter:FULL Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:150,</p>  <p>letter:FULL Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:150,</p> 	<p>Baseline plots of original letter dataset.</p> <p>High variance</p> <p>91% accuracy</p> <p>Over 150 iterations to converge (fast)</p>
<p>letter:kmeans Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p>  <p>letter:kmeans Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:250,</p> 	<p>K-means</p> <p>High variance</p> <p>91% accuracy</p> <p>Over 250 iterations to converge (slow)</p>
<p>letter:GM Confusion Matrix</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:200,</p>  <p>letter:GM Learning Curve accuracy</p> <p>hidden_layer_sizes:50,activation:relu,solver:adam,learning_rate:constant,learning_rate_init:0.01,max_iter:200,</p> 	<p>Gaussian Mixture</p> <p>High variance</p> <p>91% accuracy</p> <p>Over 200 iterations to converge (medium)</p>

There does not appear to be any effect when adding the additional dimension.

## References

1. <http://scikit-learn.org/stable/modules/clustering.html>
2. [http://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_kmeans\\_silhouette\\_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py](http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py)
3. [http://cseweb.ucsd.edu/~atsmith/project1\\_253.pdf](http://cseweb.ucsd.edu/~atsmith/project1_253.pdf)
4. <http://www.pca.narod.ru/DimensionReductionBrifReview.pdf>
5. Sebastian Raschka. 2015. Python Machine Learning. Packt Publishing.
6. <https://arxiv.org/ftp/arxiv/papers/1405/1405.7471.pdf>
7. <http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>
8. [http://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_cluster\\_comparison.html](http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html)
9. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
10. API design for machine learning software: experiences from the scikit-learn project, Buitinck *et al.*, 2013.

Please see readme.txt for code references.