[redacted] Unsupervised and Dimensionality Reduction

By **Sahil Gupta** [redacted]

I. Abstract

1. To understand how unsupervised learning can be applied for data description & understanding unlabeled data, *both **datasets from assignment 1 will be re-used***. Also, ***Dimensionality Reduction will be referred to as "DR" for brevity***.
2. **Scikit Learn** was used throughout the assignment because of its detailed & thoughtful documentation and powerful capabilities such as great out of the box Clustering & DR algorithms, supporting tools such learning curve, scoring mechanisms such as F1 score, silhouette score, etc . I've borrowed code directly from their website and cited it in-line with source code wherever used. I also used the book "*Introduction to Machine Learning with Python*"[1] written by authors' of Scikit Learn, Sarah Guido & Andreas Müller to employ techniques taught there.
3. **Dataset I: UCI Adult Census Dataset**[2]: The goal of this dataset is to predict whether income exceeds $50K/yr based on 1994 census data. This dataset contains ***32561 instances x 14 attributes each (multivariate) which are either categorical or integer (heterogeneous).*** Each entry contains demographic attributes about an individual such as age, sex, occupation, race, workclass, education, marital-status, relationship, etc. The final column contains labels that are only used for performance evaluation of clustering & DR algorithms during testing phase. This problem is very interesting especially in 2020 (a census year!) because one can clearly see what factors play a role in an individual's income. *This dataset is imbalanced because there are 3 times more rows with <50k USD income than >50k USD income.*
4. **Dataset II: UCI Spambase Dataset**[3]: The goal of this classification problem is to predict whether an email is spam (unsolicited) or non spam based on data collected from HP Labs postmaster & individuals who filed spam along with work & personal emails of authors. This dataset contains ***4601 instances x 57 attributes each (multivariate) which are Integer & Real only***. Each row contains either percent of words in email that match a "Word" or "Character. The final column contains whether an email was considered spam (1) or not (0) (again only used for performance evaluation since this is unsupervised learning). This dataset ***is a stark contrast from the census dataset above because it doesn't contain any categorical features & 4 times smaller in size (# of data points)***. While the number of features/dimensions is much higher, Spambase is much easier to process since it contains integers between 0 & 1 ***(homogeneous)***. This dataset is ***slightly imbalanced*** because there are 1.5 times more non spam email instances (61%) than spam (39%). This problem interests me because *email spam detection is non-trivial & spam is costly (20 billion USD/annum*[4]*. Moreover, this dataset's differing nature from the Adult Census data will show us how several clustering and DR algorithms compare.*

II. Experimental Approach & Metrics Chosen

Metrics Used

For clustering algorithms evaluation, measures like[5] ***homogeneity, completeness, V-Measure, Adjusted Rand Index "ARI", Adjusted Mutual Information "AMI" (both assess relative to a ground truth clustering) & silhouette score (doesn't require ground truth labels)*** are used. For these, ***higher scores are preferred***. Per Scikit[1], "*Silhouette score computes compactness of a cluster* (while good, it doesn't allow for complex shapes)." Homogeneity (*all clusters contain only data points which are members of a single class*), completeness (*data points that are members of a given class are elements of the same cluster) and V-Measure (harmonic means of the two*) "are based on normalized conditional entropy measures of the clustering labeling to evaluate given the knowledge of a Ground Truth class labels of the same samples". Finally, ARI *measures similarity between two clusterings* and AMI is an *adjustment of the Mutual Information (MI) score to account for chance.*

For comparing DR algorithms, ***variation captured*** will be used along with ***reconstruction loss i.e. trying to recover the original dimensions after applying dimensionality reduction on them*** with the goal of minimizing loss. For MLPClassifier evaluation, we'll use ***F1 score*** as before due to its superior measurements on imbalanced datasets.

Experimental Approach

As discussed in Assignment 1 report, feature engineering and pre-processing is performed on both datasets. Adult Census dataset required more preprocessing than Spambase. These techniques included "binning for age & hours.per.week", "minmax scaling on binned age, number of education years, etc." and "one hot encoding on categorical data such as workclass, occupation, race, sex". As a result, the ***Adult Census dataset became 30162 rows x 38 input features in final size & Spambase dataset became 4601 rows x 58 features in final size.***

Due to author's proficiency in python, matplotlib, yellowbrick & tabulate plotting libraries were used for visualizations. To ensure no bias/influence from training set seeping to out-of-sample set, a hold out set of 30% of data was used. Shuffling and stratification were used to do the split to counter the effects of imbalanced datasets: `train_test_split(x, y, test_size=0.4, shuffle=True, stratify=y, random_state=0)`.

Finally, more than 40 experimental trials were conducted to arrive upon answers and each dataset was thoroughly analyzed iteratively. You are encouraged to follow along with the code. Due to brevity, several graphs couldn't be discussed. Some experimental runs can be found with code. ***Here is the methodology used to arrive upon parameters for the algorithms that we'll discuss soon:***

Fig 1a. Process used to arrive upon ideal n_clusters (Clustering)

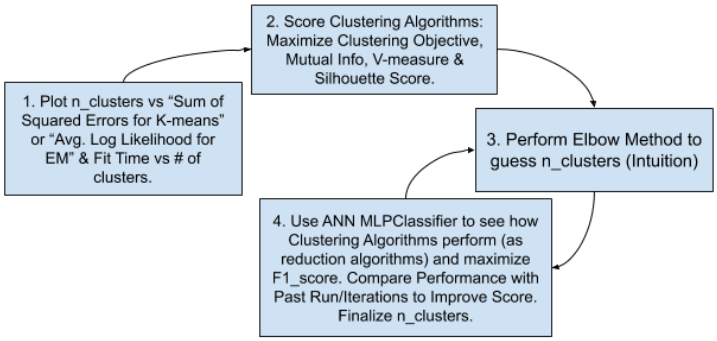
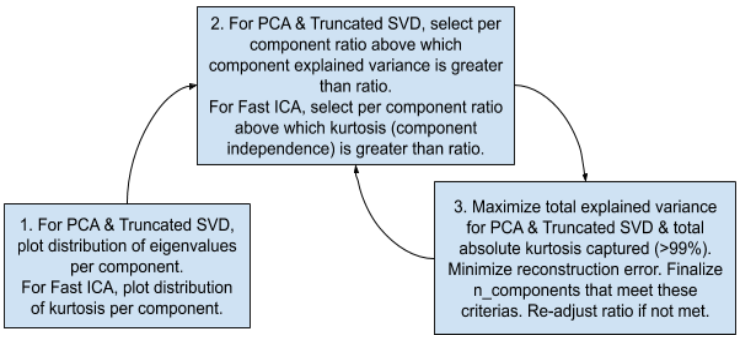


Fig 1b. Process used to find ideal n_components (for DR)



III. Clustering Algorithms: Unsupervised Learning

Elbow Method and Comparing KMeans with Expectation Maximization

1. For KMeans (KM)[6], `sklearn.cluster.KMeans` is used & for Expectation Maximization (EM)[6], `sklearn.mixture.BayesianGaussianMixture` is used. *EM mixture model uses kmeans under the hood* where it incorporates information about covariance of structure of data along with centers of latent Gaussians. Using the Elbow Method described above & with the goal of maximizing scores such as silhouette, V-measure & reducing fit times, the figures 1c, 1d, 2c, 2d show the KM n_clusters & Expectation Maximization (EM) n_components with large blue dots.
2. Clustering objectives i.e. Sum of squared distances for KM & Average Log Avg. Log Likelihood of given data for EM (a measure of goodness of fit for statistical models) vs # of clusters is used. ***We can see the asymptotic behavior of k-means (J. MacQueen[7])***. Per David Pollard[8], "a set of *n points in euclidean space partitioned into k groups that minimize sum of squared error, conditions are found to assure asymptotic normality of the vectors of means of k groups.*"
3. We can see that in ***EM, elbow is very easy to find/visualize i.e. Avg. Log Likelihood of given data increases very fast for small increase in # of clusters and then plateaus. For KM, finding elbow requires more intuition.*** *In KM, elbow seems to be* ***usually located at the intersection of fit times*** *(which increases as # clusters grows)* ***and Sum of Squared distances*** *(which decreases).*

Fig 1a. KM n_clusters vs scoring metrics (Dataset I)

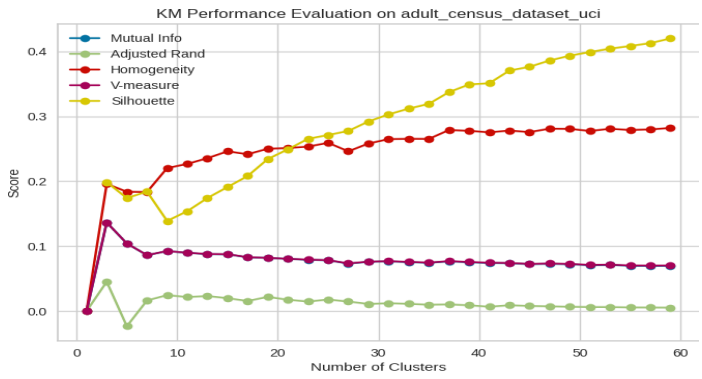


Fig 1b. EM n_components vs scoring metrics (Dataset I)

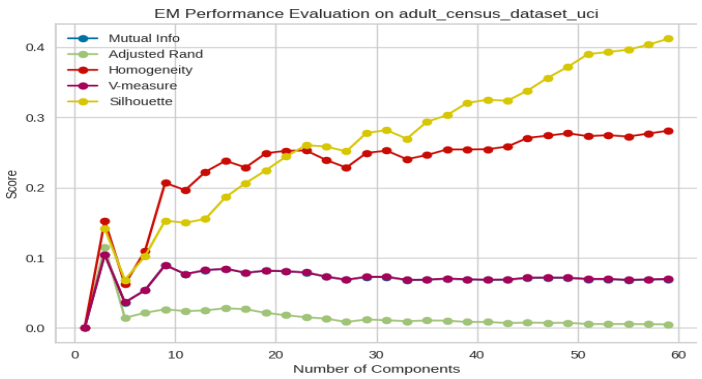


Fig 1c. KM n_clusters vs Sum of Squared Distances (Set I)

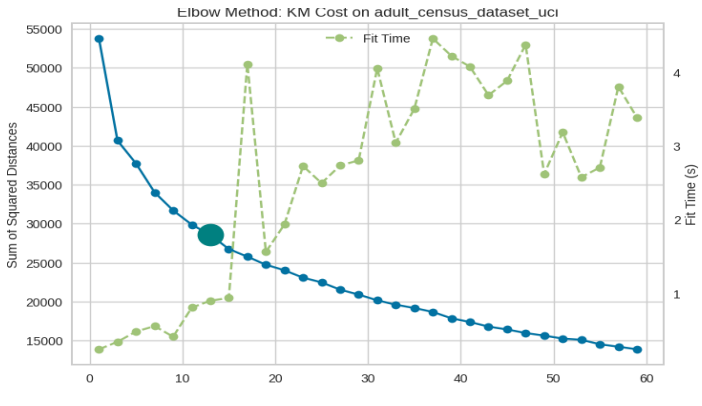
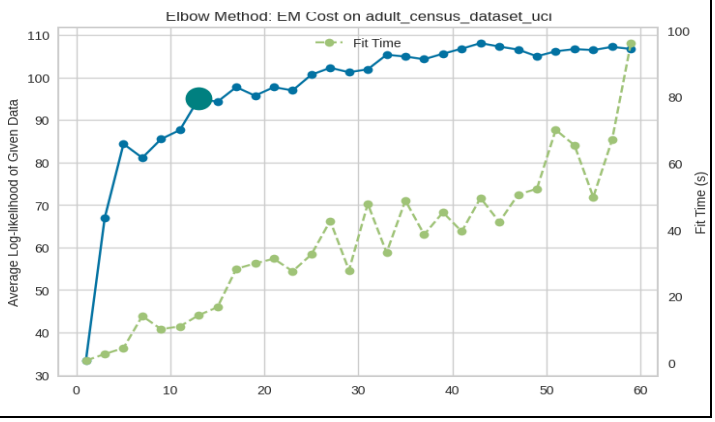


Fig 1d. EM n_clusters vs Avg. Log Likelihood (Dataset I)



Comparison Across Datasets

When comparing across datasets, some striking points emerge that are evident empirically:

4. In ***Adult Census Dataset (more heterogeneous), we can see that KM performs a litter better than EM initially as # of clusters increase*** *(figures 1a vs 1b & 2a vs 2b) but for sufficiently large # of clusters, performance is similar.* ***In comparison, EM is smoother than KM and performs better for continuous valued gaussian distributions (likely Spambase dataset).***
5. ***In both datasets, as # of clusters increase, homogeneity score increases***, likely because with more clusters, closers points get clustered/grouped together around centroids such that for k approaching number of data points, every point forms its own cluster (also discussed in lecture).
6. To understand if clusters line up with labels, all ***ground truth metrics such as homogeneity, v-measure, ARI & AMI, show higher performance on simpler & homogeneous datasets like Spambase*** vs heterogeneous imbalanced Adult Census dataset.
7. We can see that ***for more homogeneous & balanced datasets like Spambase, as # of clusters increase, silhouette score reduces*** (Fig 2a & 2b) whereas for ***less homogeneous & more imbalanced datasets like Adult Census, as the # of clusters increases, silhouette score increases***. This is likely because silhouette score doesn't use ground truth labels and is a measure of compactness. It is expected that Adult Census dataset will have tighter/closer clusters whereas Spambase will have more spread out clusters. We'll visualize clusters in section V below & see if this is indeed true.

Fig 2a. KM n_clusters vs scoring metrics (Dataset II)

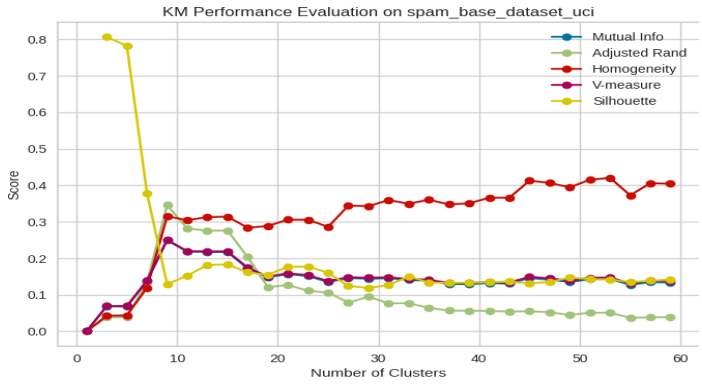


Fig 2b. EM n_components vs scoring metrics (Dataset II)

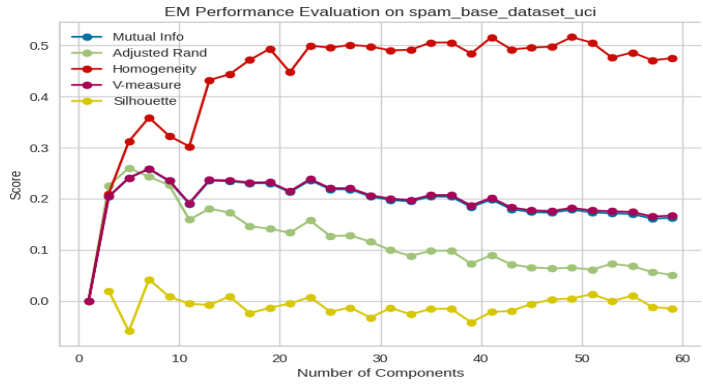


Fig 2c. KM n_clusters vs Sum of Squared Distances (Set II)

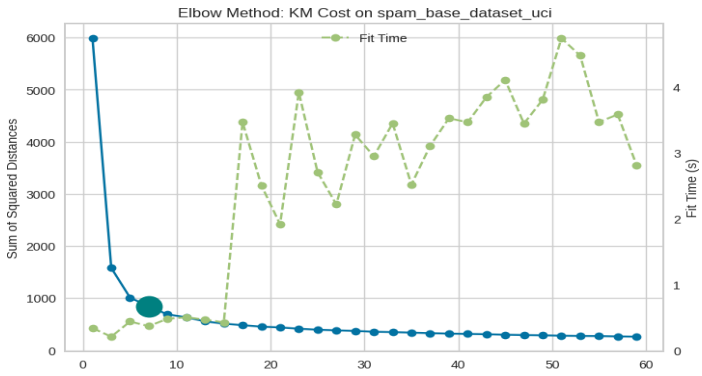


Fig 2d. EM n_clusters vs Avg. Log Likelihood (Dataset II)

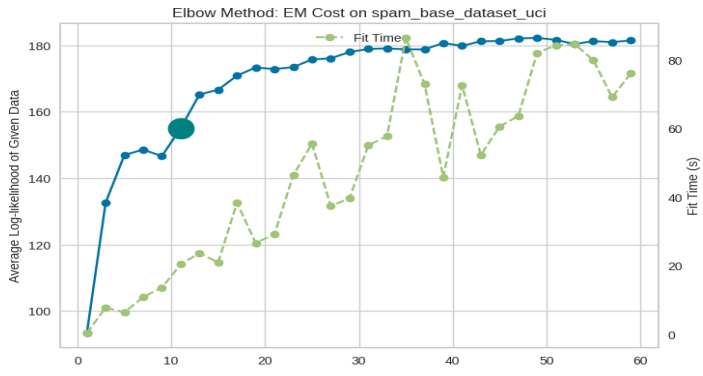


Fig 3a. KM & EM n_clusters vs Fit Time (Dataset I)

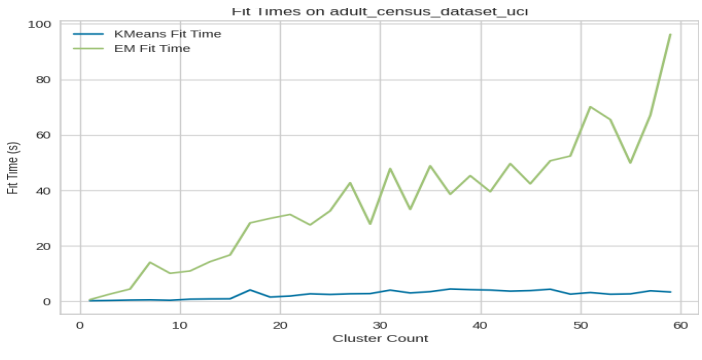
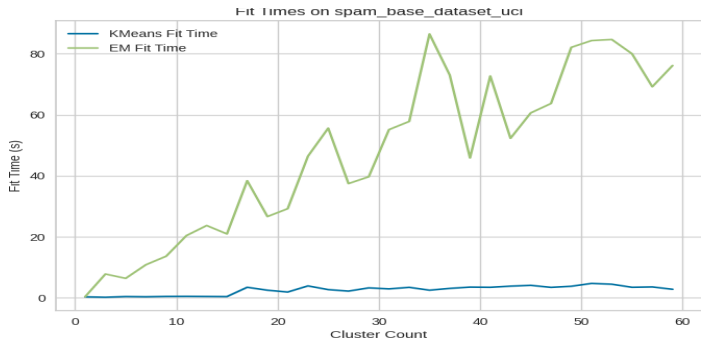


Fig 3b. KM & EM n_clusters vs Fit Time (Dataset II)



8. ***EM takes significantly longer than KM across both datasets***. KM is clearly polynomial and fast. As mentioned in lectures, this is expected because in k-means, there is a finite number of configurations but in EM, since configurations are probabilities, there is an infinite number of those and it may never converge (but practically does). It's surprising to see that ***even though Adult Census dataset is 4.3 times larger than Spambase dataset, EM actually takes longer on Spambase***. This is likely due to the curse of dimensionality because Spambase has more features 58 than Adult Census dataset 38. We can see how DR (dimensionality reduction) will help with this problem below. Notice the difference in scales for Fig 1c & 2c. ***Adult Census dataset has a larger sum of squared distances (than Spambase) because of the big variety in data samples***.

After performing the above analysis, table below shows the final clustering models implemented for both the datasets:

Fig 4a. KM n_clusters vs Sum of Squared Distances (Set II)	**Fig 4b**. EM n_clusters vs Avg. Log Likelihood (Dataset II)
`clustering_algorithms_best_adult_census_dataset = {"KMeans (11)": KMeans(n_clusters=13, n_init=3, max_iter=300, random_state=0), "EM (13)": BayesianGaussianMixture(n_components=13, n_init=3, max_iter=300, random_state=0)}`	`clustering_algorithms_best_spam_base_dataset = {"KMeans (7)": KMeans(n_clusters=7, n_init=3, max_iter=300, random_state=0), "EM (11)": BayesianGaussianMixture(n_components=11, n_init=3, max_iter=300, random_state=0)}`

IV. Dimensionality Reduction: Feature Transformation of Datasets

Principal Component Analysis (PCA)

In PCA, the dataset is rotated such that rotated features are statistically uncorrelated & maximize variance[9]. This rotation is often followed by selecting only a subset of new features, according to how important they are for explaining the data.

1. `sklearn.decomposition.PCA` was applied here. PCA Eigenvalue distributions (figure 5a & 5b) show that ***variance is more evenly distributed in the Adult Census dataset than Spambase***. In the Adult census, around 10 out of 38 features capture high variance and 20 out of 28 capture medium-low variance whereas ***in Spambase, a mere 2 out of 56 features capture almost all the variance in the dataset***. More than 40 features don't contribute to explained variance at all! This means that in Spambase, we can likely use fewer components and still maintain good reconstruction.
2. Using experimental approach discussed, following ratios are tolerated to capture >~99% total explained variance: *Adult Census eigenvalue cutoff=0.003 (99.732% variance captured) and Spambase eigenvalue cutoff=0.001* (98.774% variance captured). Hence, ***n_components=29 out of 37 & 29 out of 56 are selected respectively*** for Adult Census & Spambase.

Fig 5a. PCA Eigenvalues distribution for Dataset I.

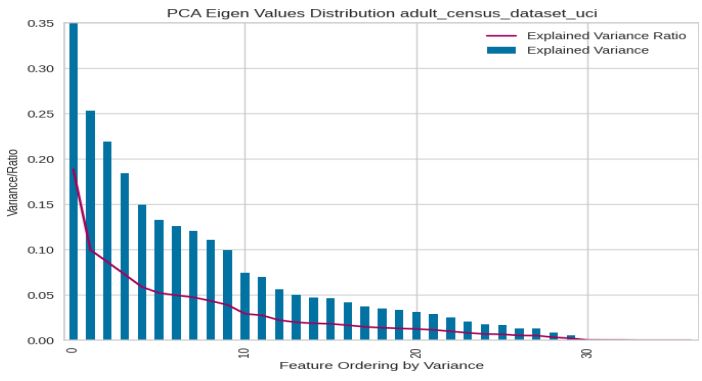


Fig 5b. PCA Eigenvalues distribution for Dataset II.

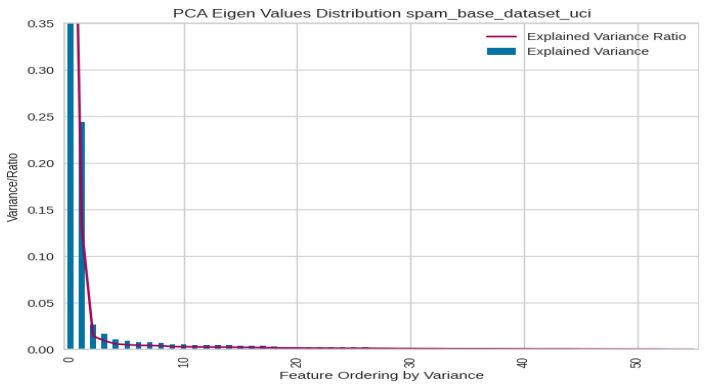


Table 6a. Reconstruction error after applying PCA on Spambase as N components is increased.

PCA is applied to both datasets but Spambase is only shown for brevity. It's clear that ***reconstruction loss (mean squared difference between original raw data and projected data by PCA) reduces as N components increase***. This occurs because higher the number of components, *the more variance is captured and hence reconstruction becomes easier. Of course, there are diminishing returns beyond certain n_components* (as discussed in point 2 above). Moreover, when n_components is equal to dimensions of original data i.e. all are kept, reconstruction error is negligible.

N Components	Loss	Mean	Std	Max	Min
5 (10%)	0.00214784	-3.95123e-19	0.0463448	0.999052	-0.342477
11 (20%)	0.00138865	1.00451e-18	0.0372647	0.99509	-0.456064
16 (30%)	0.000986923	8.68901e-19	0.0314153	0.993902	-0.434823
22 (40%)	0.000669029	2.41507e-19	0.0258656	0.988789	-0.439903
28 (50%)	0.000440473	5.58282e-19	0.0209874	0.984191	-0.426293
33 (60%)	0.000300057	-1.45766e-18	0.0173222	0.969098	-0.463259
39 (70%)	0.000173253	3.46021e-19	0.0131626	0.965886	-0.422705
44 (80%)	9.19334e-05	2.61209e-19	0.00958819	0.956621	-0.295171
50 (90%)	2.72247e-05	8.57876e-19	0.00521773	0.853841	-0.291652
56 (100%)	2.14257e-31	8.3483e-19	4.62878e-16	3.55271e-14	-1.81244e-14

Independent Component Analysis (ICA)

ICA's goal is to separate multivariate signals (e.g. cocktail party example) into additive subcomponents that are maximally independent (i.e. high kurtosis) and mutual information between original & new feature space is high/maximized[9]. From lectures, "If we take a bunch of independent variables and sum them together, that is, create a linear combination, we in fact end up with a Gaussian." Hence, ***per Central Limit Theorem, it's clear that Kurtosis is the ideal measure to rank & determine components to use***.

1. `sklearn.decomposition.FastICA` was applied here. ICA's Feature Kurtosis distributions (figure 7a & 7b) are evenly distributed across both datasets (with Spambase being more even). In Spambase, all features have kurtosis >10 whereas in the Adult census, only around 16 out of 38 have kurtosis >10. Within the Adult Census dataset, kurtosis varies from degree 0 to 3 i.e. ~3 degrees difference whereas in spambase it varies from degree 1 to 3 i.e. ~2 degrees scale difference.
2. In ICA, the cut-off is kept at abs(kurtosis) = 1.0 for Adult Census and 50.0 for Spambase. Hence, ***n_components=30 out of 37 & 44 out of 56 are selected respectively*** for Adult Census & Spambase.

Fig 7a. ICA Kurtosis distribution for Dataset I.

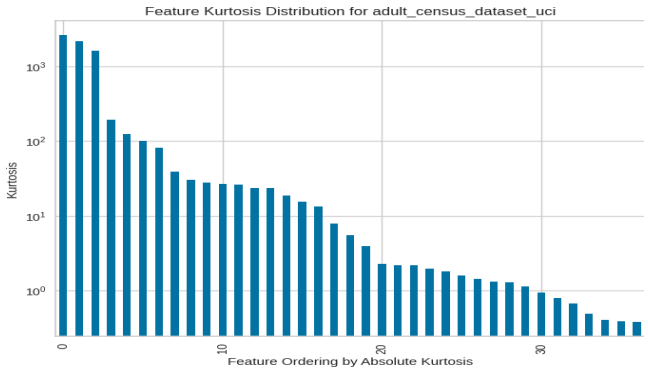


Fig 7b. ICA Kurtosis distribution for Dataset II.

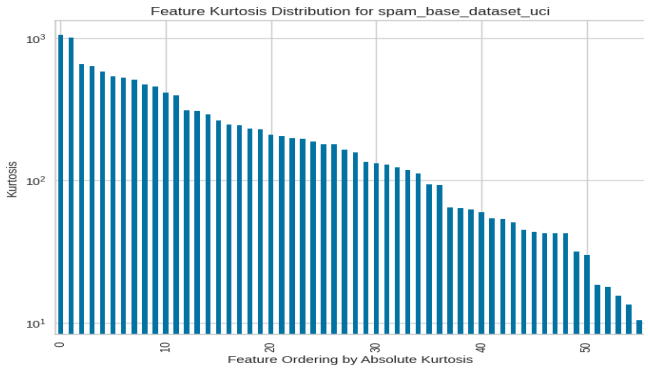


Table 8a. Reconstruction error after applying ICA on Spambase as N components is increased.

ICA is applied to both datasets but Spambase is only shown for brevity. It's clear that ***reconstruction loss (mean squared difference between original raw data and projected data by ICA) reduces as N components increase***. Though, PCA does slightly better than ICA for reconstruction. It is even more evident for the Adult Census dataset. See figure 12 below.

N Components	Loss	Mean	Std	Max	Min
5 (10%)	0.00214784	-8.10795e-19	0.0463448	0.999052	-0.342482
11 (20%)	0.00138806	-1.5023e-19	0.0372567	0.994979	-0.459051
16 (30%)	0.000986763	6.57409e-19	0.0314128	0.994195	-0.435025
22 (40%)	0.0006686	-8.99532e-19	0.0258573	0.988127	-0.437787
28 (50%)	0.000440215	3.76036e-19	0.0209813	0.984167	-0.428149
33 (60%)	0.000299958	-2.13031e-19	0.0173193	0.972333	-0.461748
39 (70%)	0.000173237	2.79526e-19	0.0131619	0.965962	-0.422644
44 (80%)	9.19334e-05	3.491e-19	0.00958819	0.956621	-0.295171
50 (90%)	2.72247e-05	1.2414e-19	0.00521773	0.853841	-0.291652
56 (100%)	2.02967e-31	2.22205e-19	4.50519e-16	9.23706e-14	-1.50713e-14

Randomized Projections (GaussianRP)

`sklearn.random_projection.GaussianRandomProjection`[9] was applied here. "RP" or Randomized Projection's goal is to reduce dimensionality by projecting original input features on a randomly generated matrix where components are drawn from distribution N(0, $1/n_{components}$). ***RP is simple but fast in comparison to above DR algorithms though at the expense of slightly poor performance***. 10 trials were performed for RP and PCA on both datasets and results are below. Pairwise metrics are calculated among trials to answer what happens when RP is run many times. It's clear that ***RP has significant variation across trials when compared to PCA but is much faster. E.g. on Spambase, RP has 220 times more std. Dev. than PCA but is 13 times faster. This speed comes at the expense of building an approximation only as seen by largest reconstruction errors in Figure 9c & 12 below.*** In RP, while an 'auto' option is available that can "auto adjust n_components according to the number of samples in the dataset & the bound given by Johnson-Lindenstrauss lemma"[9], n_components=29 is selected (treating PCA as gold standard).

Fig 9a. Fit_transform Time (s)

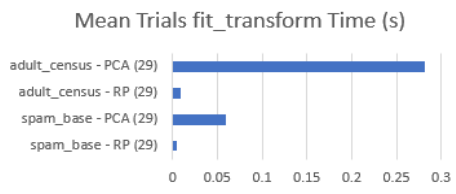


Fig 9b. Std Dev. of 10 Trial Pairs

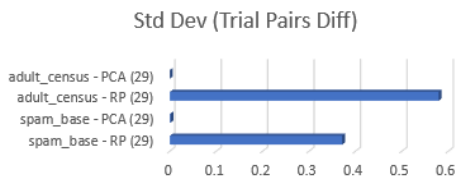


Fig 9c. Reconstruction error on Spambase.

N Components	Loss	Mean	Std	Max	Min
11 (20%)	0.230857	-0.01298	0.4803	30.6795	-27.8833
22 (40%)	0.102039	-0.0045	0.319403	17.1756	-19.5228
33 (60%)	0.053523	-0.00096	0.231348	11.2992	-13.2719
44 (80%)	0.040282	0.000905	0.200702	12.2468	-12.581
56 (100%)	0.028456	0.001793	0.168679	10.9145	-10.723

Truncated Singular Value Decomposition (TruncatedSVD)

`sklearn.decomposition.TruncatedSVD` was applied here. *Truncated SVD is very similar to PCA but the input training sample matrix X does not need to be centered and* ***works well with sparse matrices***[9]. Truncated SVD uses LSA and ***combats the effects of synonymy and polysemy*** (per lectures: polysemy is a problem with multiple things and synonymy is a problem meaning the same thing.) As seen in Fig 10a & 10b, since ***Adult Census is sparse (after pre-processing & one hot encoding), TruncatedSVD performs better than PCA*** *(notice the scale difference).* *On the other hand,* ***PCA works better than TruncatedSVD on Spambase (dense matrix) when compared using reconstruction loss (not shown).*** Similar eigenvalues distributions analysis as PCA (not shown) is conducted to finalize n_components that maximizes total explained variance. ***n_components=30 out of 37 (99.736% variance captured) & 29 out of 56 (98.766% variance captured) are selected respectively*** for Adult Census & Spambase.

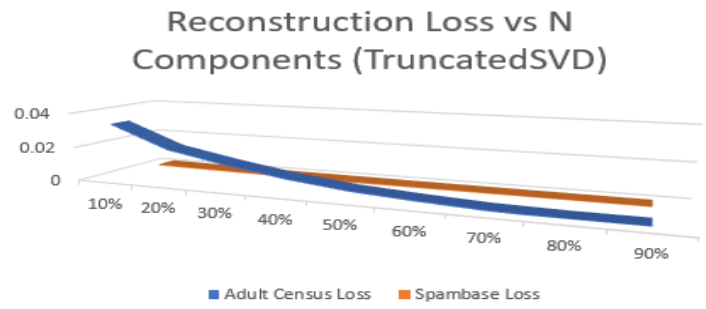
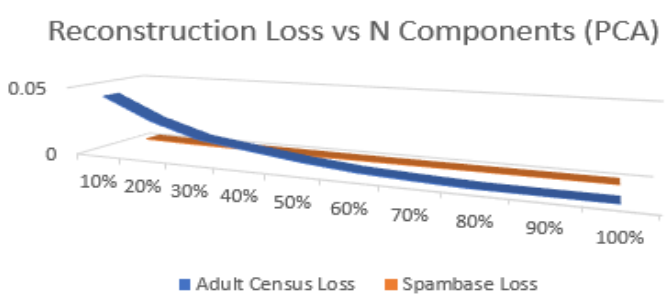
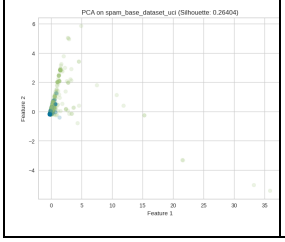
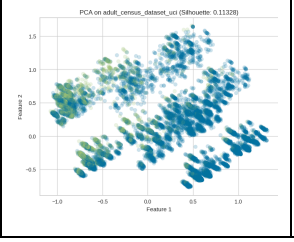
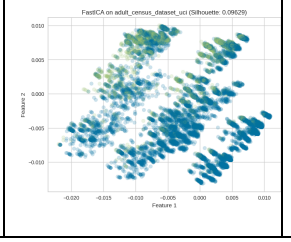
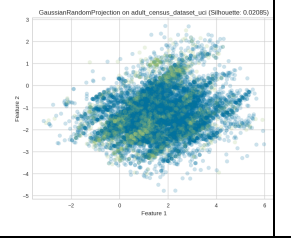
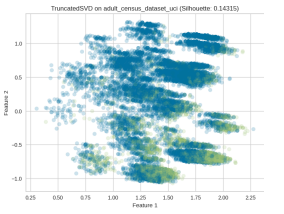
Fig 10a. Truncated SVD Reconstruction Loss for datasets	**Fig 10a**. PCA Reconstruction Loss for datasets
	

Fig 13. Out of curiosity, both datasets were projected onto 2 features after reducing them using DR algorithms and plotted using color maps. Ignore the figure's small size as their patterns are only discussed. We can see that Spambase is mostly centered around value 0 with several outliers. We'll analyze later if these outliers are Spam! In the Adult Census dataset, we can see points are scattered and no clear clusters are there for GaussianRP (likely a result of randomness). PCA, ICA & Truncated SVD have visible clusters. PCA's clusters are distributed widely. TruncatedSVD's clusters are more compact.

Spambase + PCA	Census + PCA	Census + FastICA	Census + RP	Census + TruncatedSVD
				

Comparison of All Dimensionality Reduction Algorithms

Fig 11. Reconstruction errors on Census Dataset as n_components are varied. Ranking: **PCA (best) > TruncatedSVD ~ ICA >> RP.** RP, while the fastest, has the highest reconstruction loss. TruncatedSVD & ICA excel at certain tasks as mentioned. PCA is usually a go to gold standard for DR algorithms.

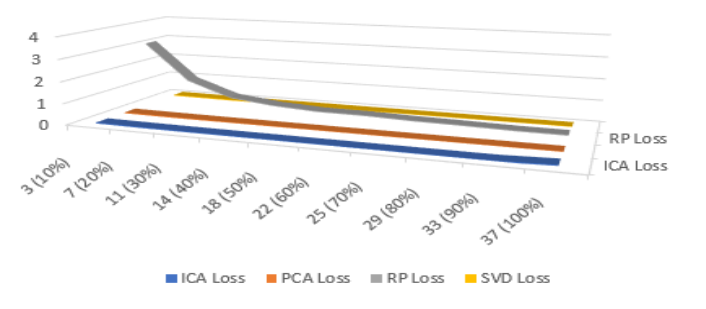
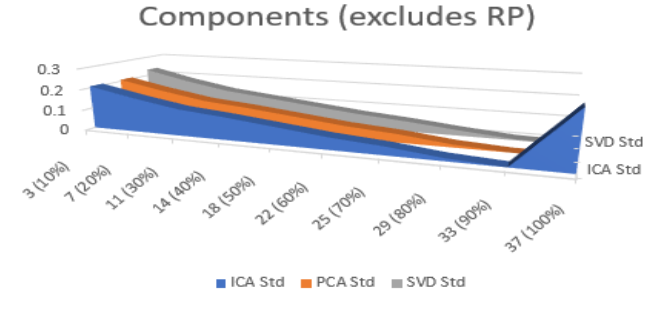
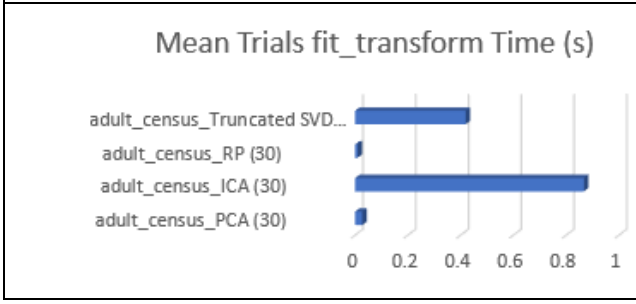
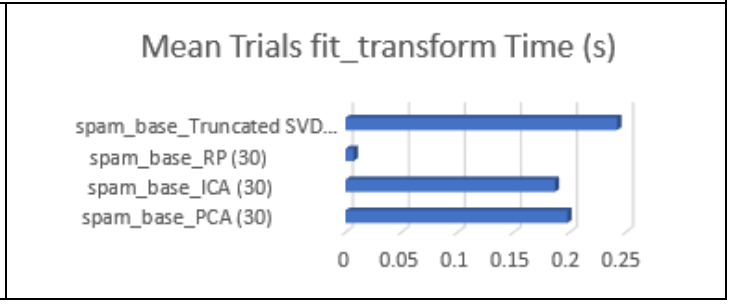
	

Fig 12. For 5 trials, fit_transform time cost ranking: **RP (best) >> PCA (best on Census) > SVD ~ ICA (better on Spambase).**





After performing the above analysis, below table shows the final models implemented for both the datasets:

Fig 14a. Adult Census Best DR Algorithms

```
dimensionality_reduction_algorithms_best_adult_census_dataset = {"PCA(29)": PCA(n_components=29, random_state=0), "FastICA(30)": FastICA(n_components=30, max_iter=1000, random_state=0), "GaussianRandomProjection(29)": GaussianRandomProjection(n_components=29, random_state=0), "TruncatedSVD(30)": TruncatedSVD(n_components=30, random_state=0)}
```

Fig 13b. Spambase Best DR Algorithms

```
dimensionality_reduction_algorithms_best_spam_base_dataset = {"PCA(29)": PCA(n_components=29, random_state=0), "FastICA(44)": FastICA(n_components=44, max_iter=1000, random_state=0), "GaussianRandomProjection(29)": GaussianRandomProjection(n_components=29, random_state=0), "TruncatedSVD(29)": TruncatedSVD(n_components=29, random_state=0)}
```

V. Clustering with Reduced/Transformed Features

Cluster & DR Visualizations

Fig 14a. KM + PCA on Spambase

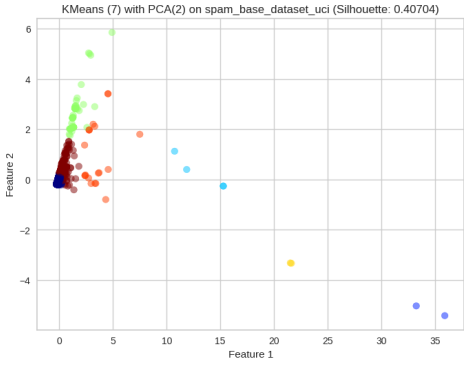


Fig 14b. EM + ICA on Census

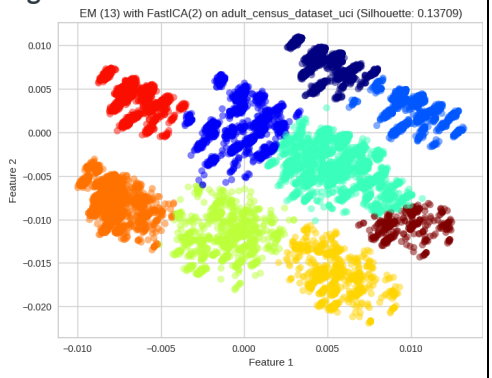


Fig 14c. KM + RP on Census

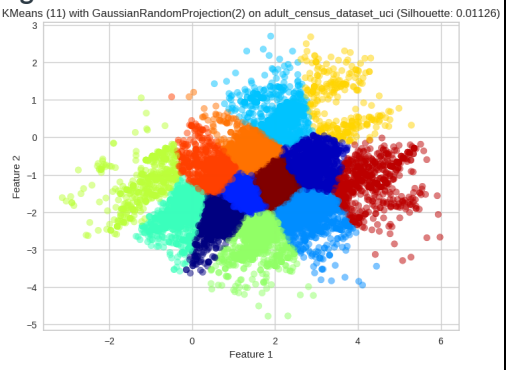


Fig 14d. EM + PCA on Census

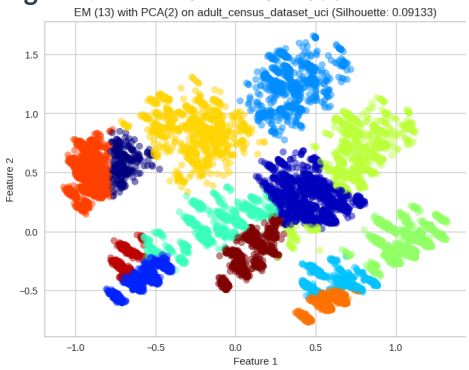


Fig 14e. KM + TruncatedSVD on Census

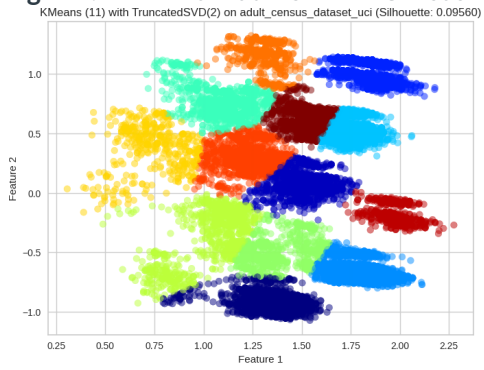
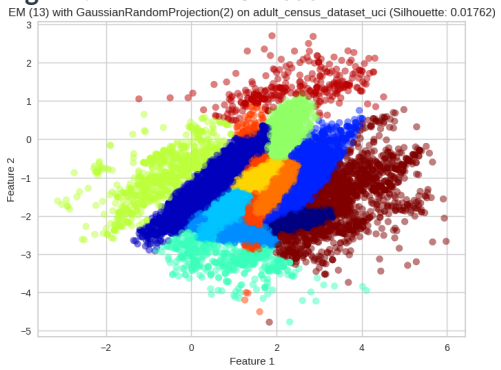


Fig 14f. EM + RP on Census



The final clustering & dimensionality reduction algorithms were run on both datasets. Also, final clustering algorithms were run on DR algorithms projected on 2 components for ease of plotting (using colormaps). Out of 16 figures, 6 are included (5 Census & 1 Spambase) above that show interesting patterns:

1. ***KMeans clusters' decision boundary follows Voronoi diagrams (Fig 14c) where clusters are broken into cells (sometimes circular) with centroids. On the other hand, EM clusters' decision boundary follows Gaussian clusters that are overlapping non circular/elliptical*** (Fig 14f). This is because EM/GMM is a generative probabilistic model for distribution of data.

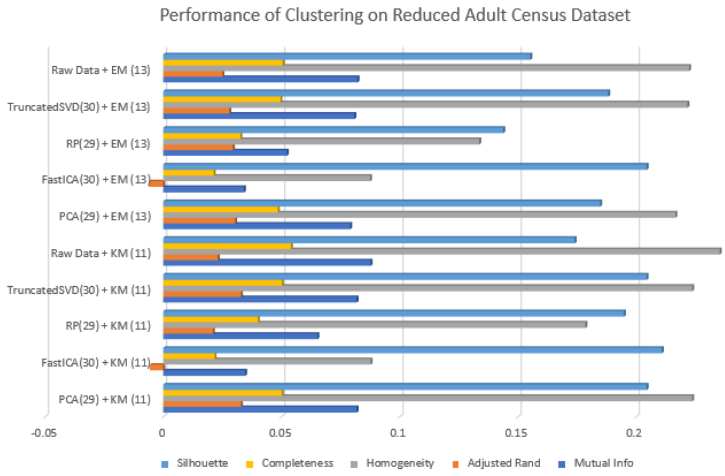
2. *Spambase dataset has some very compact clusters* (see silhouette score in Fig 14a) but also several outlier clusters/1-2 points forming their own small clusters. On the other hand, *Adult Census dataset is evenly & widely distributed (low silhouette score) with more visible big clusters.*
3. Out of curiosity and ***hunch that outliers in Spambase dataset (Fig 14a) which have exorbitantly high feature values are Spam emails, these outliers & their ground truth labels were analyzed***. Unsurprisingly, when features greater than reduced value of 2 were analyzed all 104 points that satisfy this threshold were found to be spam! When features greater than value of 1 were analyzed, 197 points had ground truth labels as spam and 3 were legitimate emails. Hence, ***it can be concluded empirically that outliers in 2D projections above are indeed Spam.***
4. *Since PCA & TruncatedSVD's goal is to maximize variance between data, hence after reduction, the dataset looks scattered* (Fig 14d & 14e). *Gaussian RP on the other hand draws from Gaussian distribution and hence is centered around mean. This results in unique clusters* as seen in Fig 14c & 14f. Finally, *ICA's goal is to maximize independence i.e. high kurtosis. We see high silhouette scores with ICA and hence dense clusters* (Fig 14b).
5. It can be empirically seen that PCA & ICA center the data while Truncated SVD & RP don't. In terms of the scale of features (see x & y-axis), we can see that ICA has smaller absolute values on both the reduced datasets (because ICA scales features). PCA didn't scale data & had a large scale on the Spambase dataset but smaller in the Census dataset.

Clustering & DR Performance Evaluation

All 16 results are compared below in terms of performance. ***Raw data without applying DR will be taken as the baseline. You'll see that clustering on raw pre-processed data usually performs best in terms of all metrics (homogeneity, silhouette, ARI, AMI) since there is no loss of information though it takes longer to fit. Of course, results from DR section performance are evident here too.***

Fig 15a. 8 Clustering + DR on Census Performance:
KM & EM have similar performance here per silhouette (light blue) score (compactness). In terms of homogeneity (grey) & completeness (yellow), KM performs slightly better than EM. **PCA is on par with baseline whereas ICA performs worst** (even has negative ARI (orange)). Generally, ***PCA > TruncatedSVD ~ RP >> ICA***.

Fig 15b. 8 Clustering + DR on Spambase Performance:
KM has higher silhouette scores vs EM. In terms of ***homogeneity & completeness, EM performs much better than KM*** here. ***ICA continues to do poorly (silhouette is negative, which implies that the sample has been assigned to the wrong cluster[5])*** here too but leads to higher homogeneity than all others. PCA, RP & Truncated SVD continue to do well on both KM & EM and sometimes even beat baseline for some metrics.



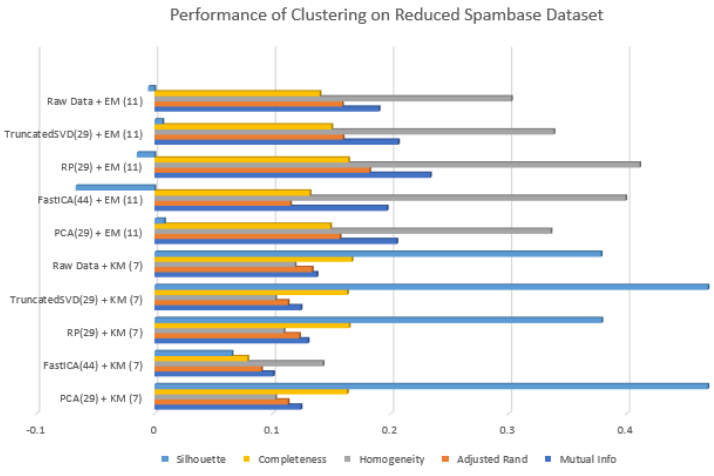


Fig 15c. Adult Census Dataset: ***Surprisingly, raw data has fastest fit time on EM & implies DR is not suitable for EM per fit time.***

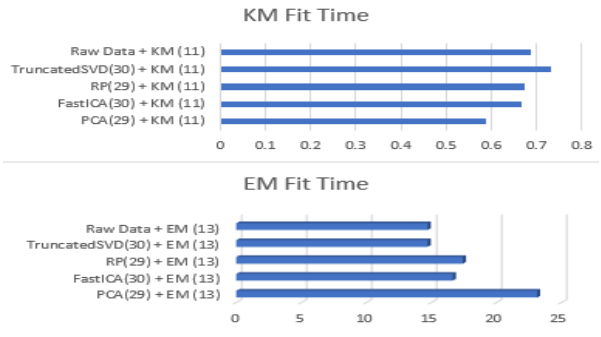
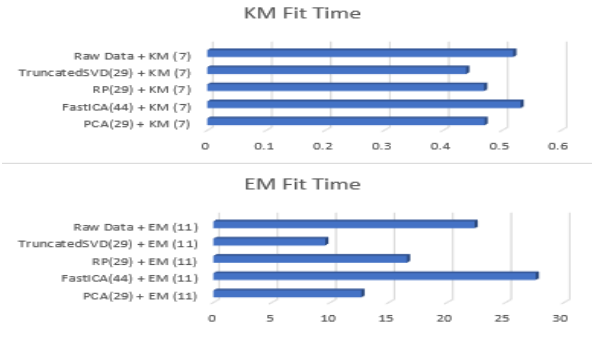


Fig 15d. Spambase: ***Though here we see raw data takes the longest on EM and using DR is useful to reduce clustering fit time.***



VI. ANN (Supervised Learning) with DR Transformed Features

Here, `sklearn.neural_network.MLPClassifier` from assignment is re-used as is without any structural changes. Since it was easy to perform this experiment on both datasets, only the most interesting dataset (Spambase) is presented here. Moreover, raw pre-processed is used as the baseline to compare how DR algorithms performed with MLPClassifier. Tuning was done using iterative Grid search & model complexity analysis similar to assignment 1 on all parameters ("activation" function, "alpha", "learning rate" & "max_iter") except hidden_layer_sizes using reduced datasets. Again, a test set of 30% was held out before any experiment was done and will now be finally used. ***Raw dataset has no information loss and always performs best in terms of F1 score accuracy***. What's surprising is that ***it's test score beat the train scores for all DR algorithms*** both in Adult Census and Spambase. Here are some insightful results:

1. *For all DR algorithms, the learning curve shape is typical*, as the training examples size increases, the test score increases and training score decreases.
2. ***ICA continues to perform poorly as a DR algorithm with very low test scores***. On the Census dataset, it performed even worse than on Spambase.
3. ***PCA does better than all (except RP & baseline) on Spambase.*** It's on par with others on the Census set.
4. TruncatedSVD does well on Census set because of its sparse nature. ***On Spambase (dense), it did poorly.***
5. It is very ***surprising to see how well in practice GaussianRP works. This may be due to it's stochastic nature that ANNs work well with***. RP did the best on both datasets and was close to baseline!

Fig 16a. DR algorithms and Spambase raw baseline datasets were each run with the MLPClassifier.

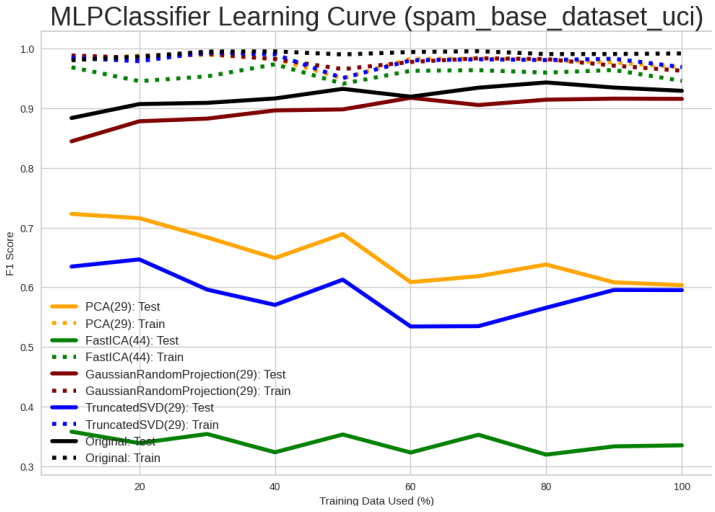


Fig 16b. DR algorithms and Adult Census raw baseline datasets were each run with the MLPClassifier.

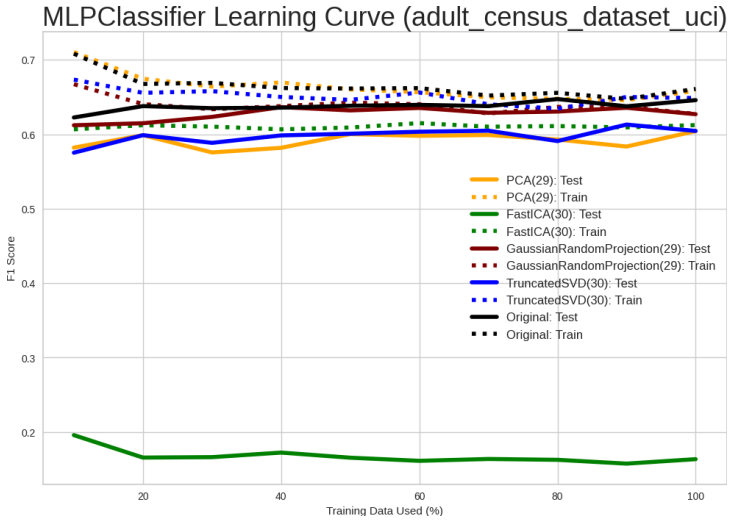


Fig 16c. In terms of Fit Time, all the DR algorithms match closely with baseline. On closer inspection, general ranking may be: TruncatedSVD ~ RP > PCA > ICA > Baseline.

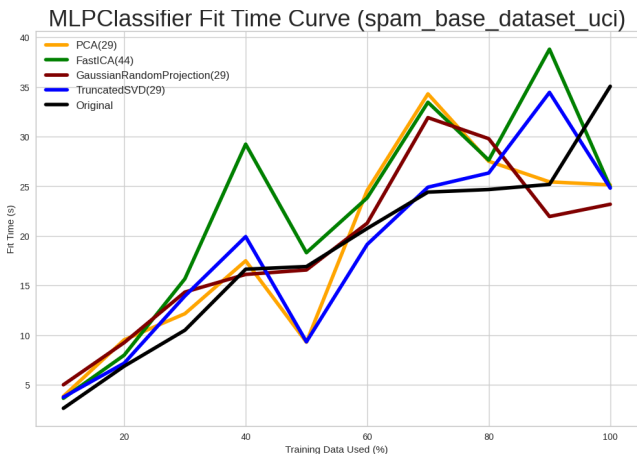
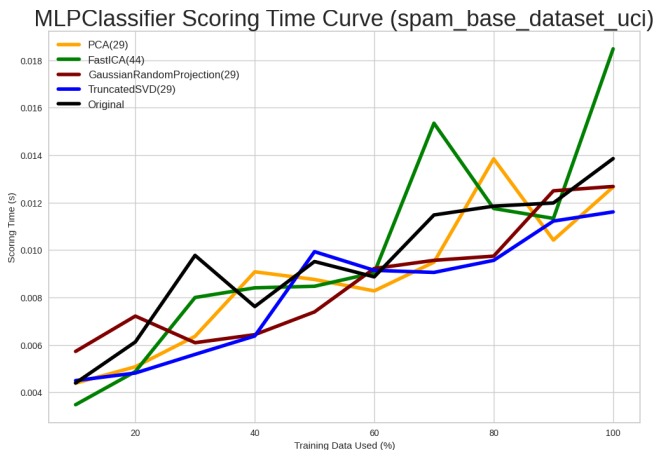


Fig 16d. In terms of predict and scoring time, all the DR algorithms also match closely with baseline.



VII. ANN (Supervised Learning) with Clustering Algorithms as DR Algorithms

Clustering algorithms will be used as dimensionality reduction algorithms. Specifically, clustering algorithms's `clustering_algorithm.predict(x_train)` output can now be used to train ANN MLPClassifier. Same as above, raw

pre-processed data is used as the baseline. Again, ***Raw dataset has no information loss and always performs best in terms of F1 score accuracy***. Here are some exciting results:

Fig 17a. (On right) Clustering algorithms and raw baseline datasets were each run with the MLPClassifier.

1. ***For EM, the learning curve is more typical for an ANN than K-means i.e.*** as training examples increase, test score increases & training score decreases.
2. ***Surprisingly, EM performs much better than K-means on both the datasets (Spambase shown)***. This may be due to EMs probabilistic model for distribution of data that ANN enjoys.
3. It is surprising to see that EM compares well with some of the DR algorithms discussed above. Specifically, *both EM & KM do better in terms of f1 score than FastICA*. ***EM comes close to PCA's performance and does better than TruncatedSVD on Spambase***!

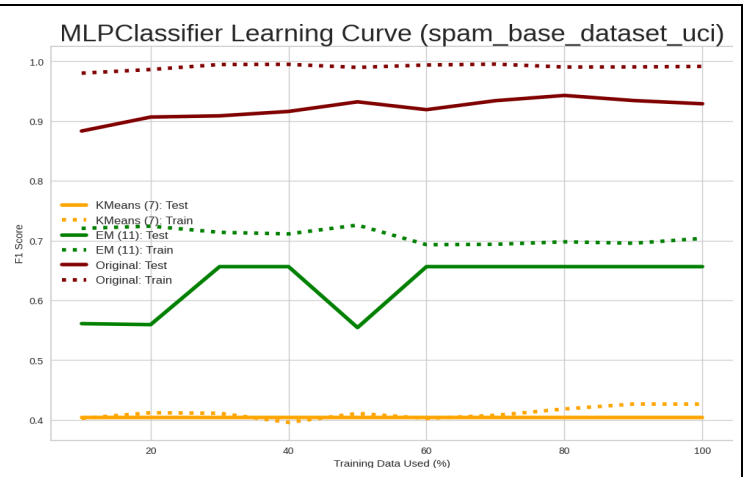
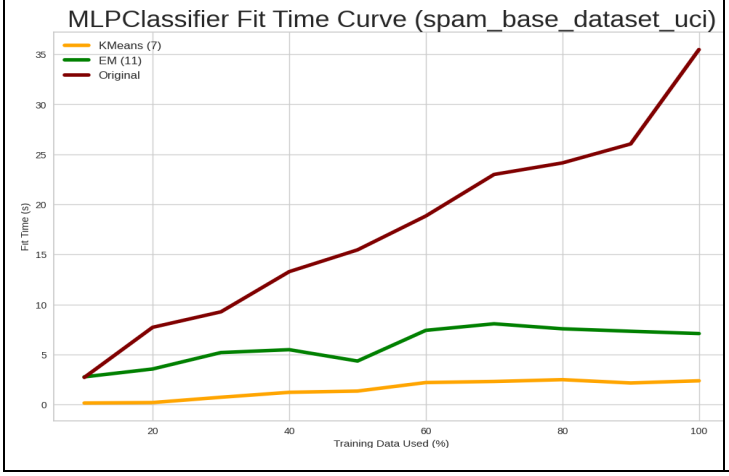
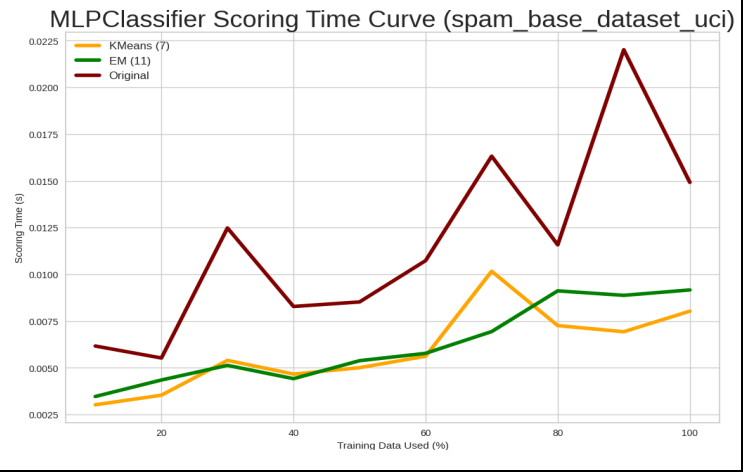


Fig 17b. In terms of Fit Time, ***KMeans clearly does the best***. Though, given it's very poor F1 score, this is not of much use. ***EM does fairly well in terms of fit time***, taking only 7s to fit the dataset. Raw set took 35 seconds to fit (worst)!

Fig 17c. In terms of predict and scoring time, a similar pattern as fit time emerges. Here, ***EM does really well***. Ranking: EM > KM > Original.





VIII. Conclusion

It's evident that trade-offs matter when it comes to selecting the right clustering or dimensionality reduction algorithm. For clustering, we saw that EM performs well but it's very slow. K-means performs well too (not as a dimensionality reduction algorithm), but it is much faster than EM. In terms of dimensionality reduction, PCA is a great choice that's efficient in terms of its ability to reconstruct data. Randomized Projections can perform really well in practice too and are fast. TruncatedSVD works well on sparse datasets but doesn't do well on dense datasets. ICA & K-means aren't an ideal choice for dimensionality reduction for the datasets discussed here. EM did surprisingly well as a dimensionality reduction algorithm.

IX. Citations

[1] Sarah Guido, Andreas C. Müller, "Introduction to Machine Learning with Python", Published by O'Reilly Media. [E-Book]
[2] Ronny Kohavi & Barry Becker, Data Mining & Visualization, Silicon Graphics, "Census Income Data Set". [URI]
[3] Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt, Hewlett-Packard Labs, "Spambase Data Set". [URI]
[4] Justin M. Rao, Microsoft Research, David H. Reiley, Google, Inc. "The Economics of Spam". [URI]
[5] Scikit Learn homogeneity_completeness_v_measure, ARI score, AMI score, silhouette_score. [URI]
[6] Scikit Learn sklearn.cluster.KMeans, sklearn.mixture.BayesianGaussianMixture. [URI]
[7] J. MacQueen, "Some Methods for Classification & Analysis of Multivariate Observations". [URI]
[8] David Pollard, "A Central Limit Theorem for k-means Clustering", The Annals of Probability. [E-journal]
[9] Scikit Learn PCA, FastICA, GaussianRandomProjection, TruncatedSVD. [URI]
[10] Scikit Learn MLPClassifier. [URI]