Dimensionality Reduction and Clustering

Yan Cheng
ycheng456@gatech.edu

I. Introduction

We explored four dimensionality reduction techniques: PCA, ICA, RP, t-SNE and two clustering algorithms in our experiments. We applied the four dimensionality reduction methods to the churn dataset and diabetes dataset from assignment 1. We applied the two clustering algorithms: K-means and Gaussian Mixture Model to the original datasets and reduced datasets. We compared the quality of the clusters we got from different combination of methods. At the end, we fit an Artificial Neural Network model to all the reduced datasets and to some new datasets with features extracted from the clusters we got from the original data.

II. Data

A. Data Description

Both two datasets we used have labels with two classes.They are preprocessed. We resampled the data to balance label distribution in the two classes. The sample size and feature types in the two datasets are shown in Table I.

TABLE I
Data Size and Feature Type

	Training	Testing	Categorical	Continuous
Churn	837	359	3	15
Diabetes	49310	21133	18	3

B. Exploratory Data Analysis

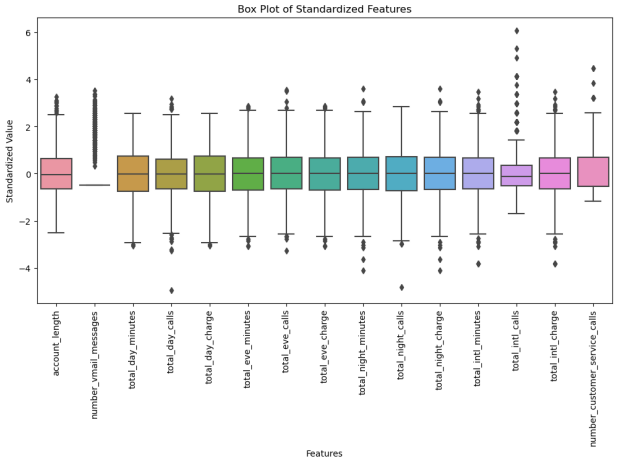

Fig. 1. Churn Data - Feature Distribution

TABLE II
Outliers and Kurtosis-Churn

Features	Outliers%	Kurtosis	VIF
account_length	0.92%	0.45	7
area_code	NA	NA	3
international_plan	NA	NA	1
voice_mail_plan	NA	NA	15
number_vmail_messages	20.4%	6.32	15
total_day_minutes	0.17%	-4.31	126356943
total_day_calls	1.17%	2.86	22
total_day_charge	0.17%	-4.3	126360359
total_eve_minutes	0.59%	-0.4	37138570
total_eve_calls	0.5%	0.63	22
total_eve_charge	0.59%	-0.41	37139398
total_night_minutes	0.67%	1.37	10576820
total_night_calls	0.25%	1.67	22
total_night_charge	0.67%	1.37	10576899
total_intl_minutes	1.25%	2.53	1008180
total_intl_calls	6.94%	10.3	4
total_intl_charge	1.25%	2.53	1008587
number_customer_service_calls	1%	4.28	2.5

The churn data is relatively clean and free of extreme deviations. From Table II, we can see that 12 of the features have kurtosis less than 3 (the kurtosis of a normal distribution) which means lighter tails and less peakedness than the normal distribution. 13 features have less than or equal to 2% of outliers. The kurtosis of total_day_minutes and total_day_charge are -4.3 which indicates an extremely platykurtic distribution. This means the distribution is much flatter than the normal distribution. Three features have kurtosis higher than 3 which means they have a leptokurtic distribution with heavy tails, a pronounced peak around the mean, and more extreme values and outliers. The percent of outliers in these three variables are 20.4%, 6.94% and 1% respectively.

The rank of the two datasets are the same as the number of features. It seems that the features in both datasets are linearly independent. However, we further investigated other multicollinearity metrics, like correlation, Variance Inflation Factor(VIF) and condition number of the input matrix. In the churn dataset, there are 8 features have astronomically high VIF values (in the millions and tens of millions). These values suggest these features are almost perfect linear combinations of each other. Features such as total_day_calls, total_eve_calls, and total_night_calls have VIF values above 20, indicating strong multicollinearity. There are four pair of features have correlation 0.9999 which means four features are perfect linear combination of another feature. One pair has correlation 0.95. The condition number of the churn data input matrix is as

TABLE III
OUTLIERS AND KURTOSIS-DIABETES

Features	Outliers%	Kurtosis
BMI	3.12%	100.63
MentHlth	16.6%	84.9
PhysHlth	15.06%	41.13

high as 143822 which indicates severe multicollinearity. There is strong redundancy among features in the churn dataset. 14 features have Shapiro-Wilk Test statistics higher than 0.88, and 12 of them are 0.99 which means most of the features appear to be normally distributed. This is very good for Gaussian Mixture Model since uses a number of Gaussian distributions to model the data.

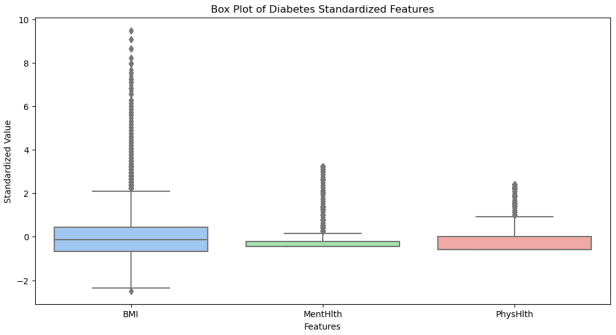

Fig. 2. Churn Data - Feature Distribution

All the three continuous features in the diabetes data have high positive kurtosis. This means these features have leptokurtic distributions with heavy tails and a sharp peak indicating a high number of outliers or extreme values.

There is only one pair of features are moderately correlated with correlation equal to 0.52. There are 7 features have VIF higher than 10. The high VIF values and the lack of strong pairwise correlations indicate complex multicollinearity involving multiple variables in the diabetes data. CholCheck, BMI, AnyHealthcare, GenHlth, Education, and Income all have high VIF values, suggesting they may be part of a complex linear relationship involving several variables. A condition number of 165.96 also indicates a significant level of multicollinearity in the diabetes dataset.Highly correlated features tend to exhibit dependencies that ICA may not be able to separate effectively, potentially resulting in components that still carry overlapping information. Preprocessing the data by PCA to handle to multicollinearity can improve ICA performance. This is automatically done in scikit learn FastICA.

C. Hypotheses

We hypothesize that K-means algorithm and GMM don't work well with the diabetes data. K-means is designed for Euclidean Distance. It minimizes the sum of squared distances from points to their cluster centroids. For categorical data, Euclidean distance is not meaningful, as there is no inherent numerical order or distance between categories. K-means computes centroids as the mean of all points in a cluster, which doesn't make sense for categorical data. Meanwhile, all the three continuous variables in the diabetes dataset have high-kurtosis distributions with heavy tails which means that there are many extreme values or outliers, which can disproportionately influence the placement of centroids. Outliers can also cause centroids to be skewed toward these extreme values, leading to clusters that don't accurately reflect the main data structure. GMM assumes that each cluster is represented by a Gaussian distribution which is not applicable for categorical data. Categorical data is typically non-Gaussian and discrete, so GMM's probability density functions cannot be applied directly. GMM models clusters based on means and covariances, which are meaningful only for continuous data. High kurtosis in all the continuous features in the diabetes dataset implies that the data has heavier tails and a sharper peak than a normal distribution, meaning the data doesn't perfectly fit GMM's Gaussian assumption. Since GMM relies on the probability density function of a Gaussian distribution, it may struggle to correctly classify points in the heavy tails of these distributions.

We also hypothesis that the dimensionality reduction methods, like PCA, ICA,RP and t-SNE don't work well with the diabetes dataset because these techniques are primarily designed for continuous, numerical data. PCA assumes continuous data, as it calculates linear combinations of features to maximize variance. It relies on operations like mean and variance, which do not have direct analogs for categorical data. ICA aims to maximize non-Gaussianity to identify statistically independent components. Like PCA, it assumes continuous data and is mathematically incompatible with purely categorical data. ICA cannot directly handle categorical data because calculating statistical independence based on numerical transformations is not feasible for categories. Random Projection (RP) does not make assumptions about the distribution of data but is generally applied to continuous numerical data.

III. DIMENSIONALITY REDUCTION

In machine learning, many algorithms suffer from the curse of dimensionality. Therefore, we face the problem of preprocessing a set of features to create a new smaller and compact feature set while retaining as much information as possible. In the following experiments, we are going to apply dimensionality reduction methods to the churn dataset and the diabetes dataset.

A. Principal Component Analysis(PCA)

PCA is a linear dimensionality reduction technique with applications in exploratory data analysis, visualization and data preprocessing [1]. It maximizes the variance of the data and minimizes L2 error. It finds directions that are mutually orthogonal.It provides a sequence of best linear approximations to the original data.

From Fig.3, we can see eigenvalues are big in the first three components in the diabetes data, beyond k=4, it starts

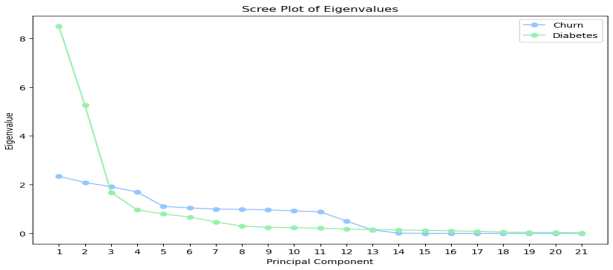

Fig. 3. Scree Plot With Eigenvalues

to level off. The first three components explained 76% of the total variance. In the churn data, the components have much smaller eigenvalues, and the eigenvalues are more equally distributed in the components. The first three components of the churn data only explained 41% of the data variance. From Fig.4, we can see that the first 11 components of both datasets explained 95% of the total variance. In both datasets, eigenvalues get close to zero beyond k=13 which means the rest of the components are redundant and they explain very little variance of the data. In the churn data, the eigenvalues of the last four components are zero this is because they are four pairs of features are perfectly linearly correlated with correlation higher than 0.99. PCA simplified the churn data very effectively. We took the first 3 components to visualize the data with true labels in the training set, and took 11 components to keep as much information as possible to fit the ANN model in supervised learning section.

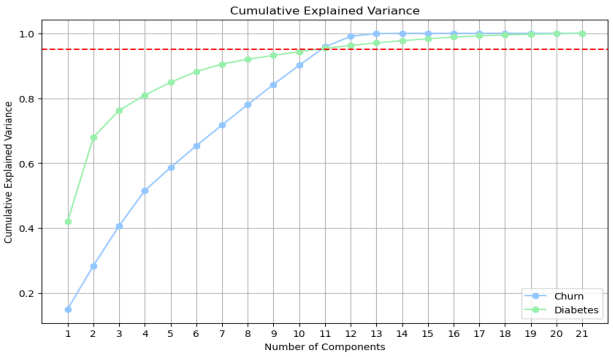

Fig. 4. Cumulative Explained Variance-PCA

There is no cluster shown in Fig.5 and 6. Since the first three principal components only explain 40% of the variance in the churn dataset, a significant amount of information is likely lost in the dimensionality reduction process. With only 40% of the variance captured, the remaining 60% could contain important information that distinguishes the classes, especially if the data is complex and requires more components to adequately represent class separation. In the diabetes data, 76% of variance is explained by the first three components.PCA attempts to capture the directions of highest variance, but in some cases, these components may not be the ones that best separate clusters or class labels. PCA is a linear method, meaning it captures linear relationships in the data. If the separation of classes in the original data space is non-linear, PCA might not capture these patterns effectively. In such cases, non-linear dimensionality reduction techniques like t-SNE might provide better visualization of separability. The lack of visible clustering in Fig.5 and 6 could also indicate that the data itself may not be inherently separable into clusters based on the true binary labels, even in the full dimensionality. Some datasets are inherently non-clusterable, especially when classes overlap significantly or are not well-defined in feature space. If the data is highly mixed between classes, then no matter how much variance is retained, separating the classes through clustering might remain challenging.

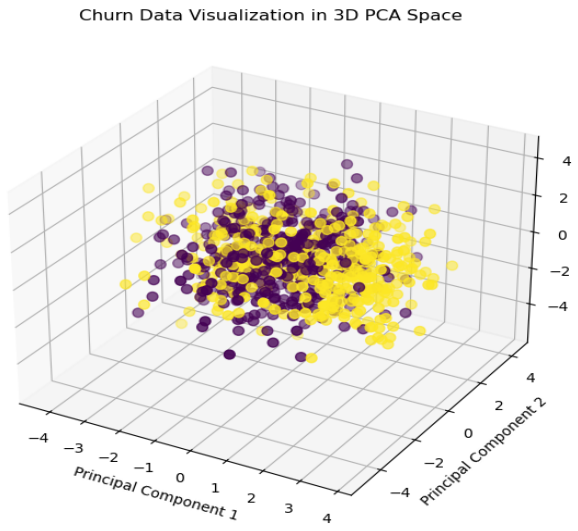

Fig. 5. 3PCA With True Labels-Churn

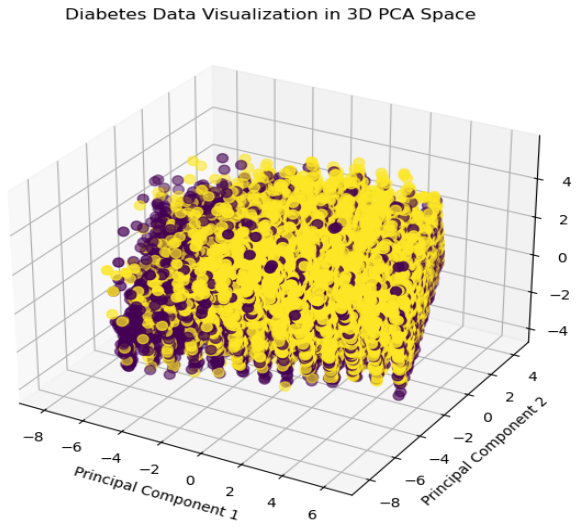

Fig. 6. 3PCA With True Labels-Churn

B. Independent Component Analysis(ICA)

ICA finds a linear transformation of the original feature space into a new feature space such that each of the individual new features are mutually independent statistically. It can be defined as: $\mathbf{x} = A\mathbf{s}$ and $\mathbf{s} = W\mathbf{x}$ ICA has several assumptions. It assumes both the mixture variables and the independent components have zero mean. It assumes the components s_i are statistically independent. It also assumes the independent components have nongaussian distributions. In our experiments, we standardized all the numeric features so they have zero mean.

Kurtosis is often used as a measure of non-Gaussianity, and since ICA aims to find components that are as independent and non-Gaussian as possible, the highest absolute mean kurtosis is a good criterion for choosing the optimal number of components. Kurtosis measures the "tailedness" or departure from Gaussianity in a distribution. A high kurtosis value (either positive or negative) indicates a distribution that is significantly different from a Gaussian distribution. Since Gaussian distributions have a kurtosis of zero, components with high absolute kurtosis values are more likely to represent non-Gaussian, independent sources. ICA works by finding a linear transformation that maximizes the statistical independence of the components. One way to achieve this is to maximize non-Gaussianity, as statistically independent sources tend to be non-Gaussian. So we computed the absolute mean value of the kurtosis for a range of numbers(less or equal to the number of original features), and then chose the number with the highest absolute mean kurtosis value.

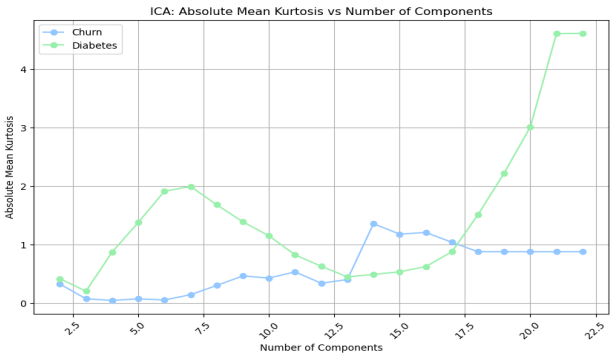

Fig. 7. Absolute Mean Kurtosis

In Fig.7, the low initial kurtosis indicate ICA struggles to separate independent signals in the two datasets. The sharp rise in kurtosis beyond k=16 in the diabetes data implies that ICA is overfitting or forcing separation on components that don't have strong independent structure. This may lead to artificially inflated kurtosis values as the algorithm fits noise or less meaningful components in an attempt to maximize separation, which isn't truly independent. Though the three continuous features have high kurtosis, ICA still doesn't work well with the diabetes data due to the presence of 18 categorical features. 21 components yield the highest kurtosis in the diabetes data which is 4.63. The 14 components yield the highest absolute mean kurtosis in the churn data which is 1.36. This is a relatively low kurtosis and it indicates the components are close to Gaussian. This suggests that the churn data doesn't have strong non-Gaussian signals for the ICA to separate effectively. This is because there are 14 features in the churn data are approximately normally distributed which violates the non-Gaussian assumption of ICA.

C. Random Projection

Random Projection generates random directions and projects data out into those directions. It provides any random linear combination of the original data. It is a simple yet powerful dimension reduction technique that uses random projection matrices to project the data into lower dimensional space. It is represented mathematically by: $\mathbf{S} = \mathbf{RX}$ where the columns of R are realizations of independent and identically distributed (i.i.d.) zero-mean normal variables, scaled to have unit length [2]. This ensures that the projection does not introduce a bias (shift) in any particular direction and that the transformation preserves distances to some extent, allowing the projection to approximately preserve the geometry of the data in the lower-dimensional space. Random projection, according to the Johnson-Lindenstrauss lemma, can approximately preserve the pairwise distances between points in $\mathbf{X}$, with high probability, even after dimensionality reduction.

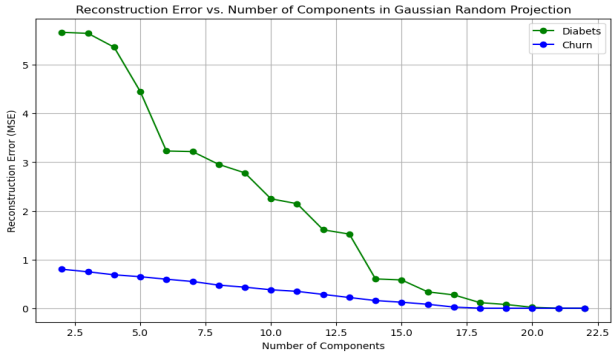

Fig. 8. Random Projection Reconstruction Error

In our experiments, we tried Gaussian Random Projection (GRP) and Sparse Random Projection(SRP), and SRP outperforms GRP. We tuned the error tolerance, but this parameter is not sensitive to tuning. We also experimented with different sparse levels, and the model performance varies for different sparse levels. 0.9 is the best sparse level we found for the churn data which means we need 90% of non-zero components to produce the optimal outcome. We generated multiple random projections by running multiple seeds to handle the random variability and averaged the metric across different projections. This make the representation of the data in the new lower space is more stable and reliable. In general, as the number of components increases, the projection captures more of the original space, reducing error until it reaches zero when the projection dimension matches the original. In our experiments, we calculated the reconstruction error for different number of components, and then use Elbow method to choose the best

number of components. From Fig.8, we can see that the reconstruction error in the diabetes data decreases as we increase the number of components. After n=14, the construction error doesn't decrease significantly if we continue to increase the number of the components. However, the reconstruction error isn't providing a clear "elbow" or plateau where error levels off in the churn data, so we tried to measure how well the pairwise distances between data points are preserved in the reduced space.

From Fig.9, we can see that the pairwise distance difference between the original data and the projected data decreases as we increase the number of components. After n=12, it doesn't decrease significantly if we continue to increase the number of the components. The pairwise distance difference is a good metric to help us choose the optimal number of components in the churn data, but it doesn't always work for a big dataset. In our case, calculating the pairwise distance difference for the diabetes data is not computationally feasible because the pairwise_distances function will need to create a large matrix of shape (49310, 49310), which consumes a significant amount of memory that raises a MemoryError.

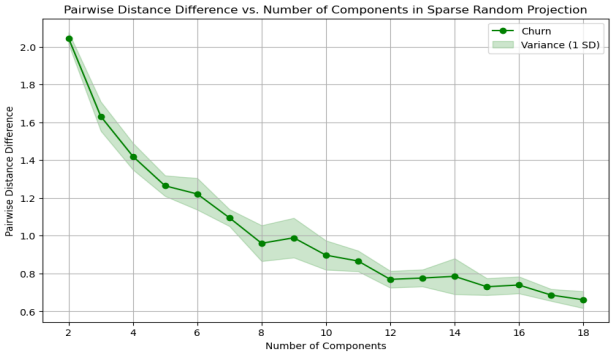

Fig. 9. Pairwise Distance Difference-Churn

D. t-Distributed Stochastic Neighbor Embedding(t-SNE)

t-SNE is a nonlinear dimensionality reduction technique for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability [3]. t-SNE creates a lower-dimensional map $\mathbf{Y}_i$ that preserves the similarity structure of the high-dimensional data. The low-dimensional similarities q_{ij} are defined using a Student's t-distribution, which helps separate clusters due to its heavy tails. t-SNE minimizes the KL divergence between high-dimensional and low-dimensional similarities, optimizing the positions of $\mathbf{Y}_i$ through gradient descent.

We tuned the perplexity to minimize the KL divergence.The perplexity is related to the variance of the Gaussian distribution that defines the probability of neighboring points. A lower perplexity gives a smaller variance (focusing on closer neighbors), while a higher perplexity increases the variance, allowing for a broader neighborhood. The best value we found is 50 which means the algorithm will look at 50 neighbors to estimate the bandwidth σ_i.

We chose three dimensions in t-SNE to visualize our churn dataset with true labels. Comparing Fig.11 and Fig.5, we can see that the data points are more scattered in Fig 11. This is because t-SNE maps local structure of the data, not global structure. Similar points are close together, and dissimilar points are far apart. There is no potential clusters or pattens shown in both plots. The two classes are just mixed together, and they are not separable.

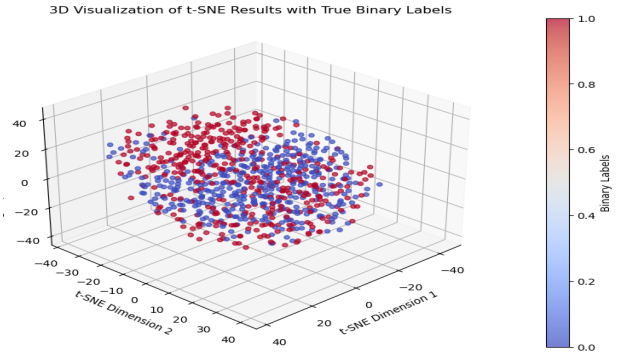

Fig. 10. Churn: True Labels in t-SNE 3 Dimensions

IV. Clustering

A. K-Means

K-means is intended for situations in which all variables are of the quantitative type, and squared Euclidean distance is chosen as the dissimilarity measure [4]. In K-means, we first specify the the desired number of clusters K, and then initialize K cluster centroids randomly, then each center "claims" its closest points defined by the Euclidean distance in the feature space, then recompute the centers by averaging the clustered points. The algorithm will repeat this process until converge. However, since the clustering process depends on the initialization of the centers, the result may represent a suboptimal local minimum. Due to this stochastic behavior in K-means, we ran the algorithm with multiple seeds and average Calinski-Harabasz Index(CHI) across the seeds.

CHI also known as the Variance Ratio Criterion (VRC). It is the ratio of the between-cluster separation (BCSS) to the within-cluster dispersion (WCSS). The higher the CHI is, the better the cluster is. So we chose the value of K that yeilds the maximum CHI in our experiments. In our experiments, we first applied K-means to the original data, and then applied it to four new datasets that are reduced by PCA, ICA, RP and t-SNE. In Table IV, we can see that optimal number of clusters for the data reduced by PCA, ICA and RP is 2. Two clusters of the data reduced by RP yields the highest CHI and lowest WCSS. Though k=4 yields the highest CHI in the data reduced by t-SNE, we calculate the quality metrics

TABLE IV
K-MEANS CLUSTERING SUMMARY:CHURN

Data	Clusters	Initial	CHI	Match Rate	WCSS
Original	3	kmeans++	79	58%	10898
PCA	2	random	76	59%	11116
ICA	2	kmeans++	66	58%	10044
RP	2	random	122	50%	9339
t-SNE	2	random	315	59%	448836

TABLE V
KMEANS: COMPACTNESS AND INTER-CLUSTER DISTANCE(CHURN)

	PCA	ICA	RP	t-SNE
Compactness-C0	3.47	3.4	3.22	22.57
Compactness- C1	3.61	3.34	3.2	21.53
Inter-Cluster Distance	2.56	2.66	2.82	29.11

TABLE VI
K-MEANS CLUSTERING SUMMARY:DIABETES

Data	Clusters	Initial	CHI	Match Rate	wcss
Original	3	kmeans++	10287	64%	90k
PCA	6	kmeans++	5407	64%	40k
ICA	3	random	2535	58%	94k
RP	2	kmeans++	24722	65%	65k
t-SNE	4	kmeans++	16805	51%	4180k

based on two clusters to compare the effects of non-linear transformation with linear transformation of the data.

PCA preserves global structure and linear relationships in the data. It does well when the important patterns in the data can be captured by linear correlations. Since PCA maximizes variance along each component, it tends to retain large-scale variations but may not capture subtle, local patterns, especially if they are nonlinear. ICA maximizes non-Gaussianity rather than variance. It preserves independence between components rather than maximizing variance. It's especially useful for data that is generated by independent sources but may not always yield good results if the data does not fit this assumption. RP preserves pairwise distances approximately but does not necessarily capture variance or independence like PCA or ICA. In our experiments, PCA and ICA yield similar compactness, inter-cluster distances, and match rates to the original data.

RP produces well-separated clusters with a relatively high CHI (122), a low WCSS(9339) and a decent inter-cluster distance (2.82), but it has the lowest match rate (50%). This suggests that while RP might improve cluster compactness and separation, it may not capture the structure that aligns well with external labels. This could be because RP does not specifically capture meaningful structure, leading to less interpretable clusters. RP may struggle to maintain clustering performance if the data relies on specific linear or nonlinear patterns, as it projects data randomly.

t-SNE stands out with the highest CHI (315) and match rate (59%), indicating that it finds well-separated clusters that align with labels. However, it has an extremely high WCSS, and low compactness, suggesting that t-SNE clusters are well-separated but not compact, which is expected as t-SNE is designed for local similarity preservation rather than compact clusters. The large inter-cluster distance with t-SNE also suggests good cluster separation, albeit with less tightness. t-SNE is a nonlinear technique designed to preserve local similarities. It places similar points (based on probability distributions) close together in the low-dimensional space while pushing dissimilar points further apart. t-SNE does not preserve distances or linear relationships and focuses on capturing local structure rather than global patterns. So it's not reliable for clustering as pre-processing.

Fig.10 shows the two clusters generated by K-means algorithm on 3 principal components from the churn data. There is some overlapping in the middle because the inter-cluster distance is 2.56, the lowest value compared with other algorithms. The compactness and WCSS is higher than other linear transformation methods like ICA and RP which means the clusters are less compact. This is why we see the points are spread out in the plot.

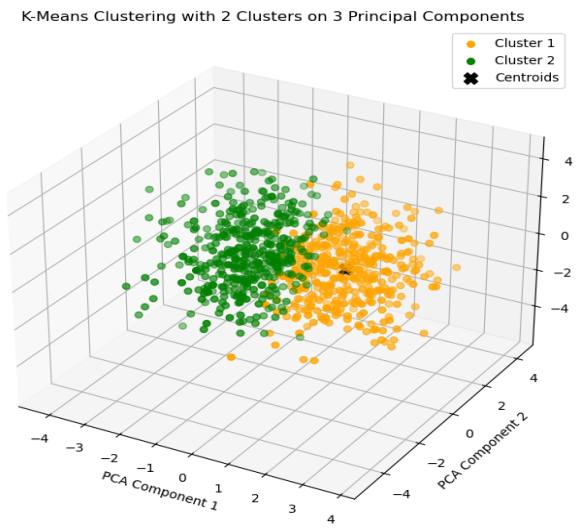

Fig. 11. 2 Kmeans Clusters on 3 Principal Components-Churn

The K-means algorithm found 3 clusters on the original data and the data reduced by ICA. It found two clusters on the data reduced by SRP. The RP-Kmeans combination has the highest CHI and good match rate(65%). The two clusters are also more compact than the clusters in the original data and ICA data. It also has highest inter-cluster distance which means the two clusters are well separated. The PCA algorithm aims to maximize the variance of the components, but there are only three continuous features in the diabetes data, so the principal components probably don't actually capture the true structure of the data. The k-means algorithm works better than what we expected on the diabetes data.

Again, the SRP-KMeans combination has the best performance. The two clusters yield the highest CHI(24722). The clusters also have lowest compactness value and highest inter-cluster distance which means they are very compact and well

TABLE VII
K-MEANS-COMPACTNESS AND INTER-CLUSTER DIST: DIAB

Data	Compactness			Inter-Cluster Distance		
	C0	C1	C2	C0-C1	C0-C2	C1-C2
Original	4.09	3.84	4.25	4.28	5.03	3.92
ICA	4.28	5.03	3.92	3.58	2	3.55
RP	3.56	3.57	NA	5.56	NA	NA

separated. t-SNE calculates pairwise distances or similarities between data points in high-dimensional space, which are typically defined using metrics like Euclidean distance. Since these metrics assume continuous values, they don't naturally apply to categorical data. Therefore, the WCSS exploded in the clusters found in the data reduced by t-SNE. However, we can still use t-SNE to visualize our data. Fig.11 shows the fours clusters in the diabetes data found by K-means in t-SNE 3 dimensions.

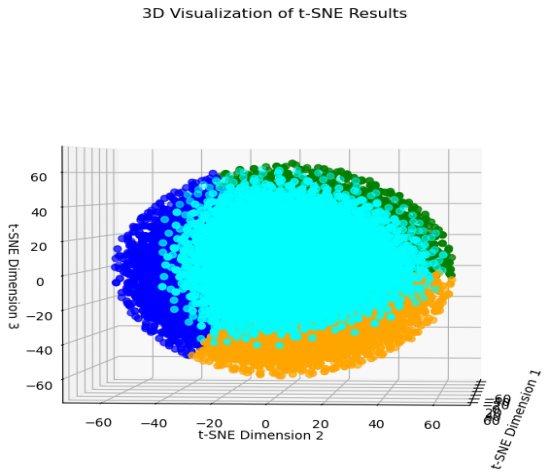

Fig. 12. 4 Kmeans Clusters in TSNE 3 Dimensions -Diabetes

B. Gaussian Mixture Model(GMM)

GMM is a mixture model that uses Gaussian as its component probability density distributions. The parameters in the GMM are fit by maximum likelihood, using Expectation Maximization algorithm. In the E-step, the algorithm construct a lower bound for the log likelihood of the parameters. In the M-step, the algorithm finds the maximum on the lower bound curve. The algorithm will repeat this process until it converges to a local optimum of the log likelihood. We set the number of initialization to 100, so the algorithm can restart 100 times to find a solution that's closer to global optimum. We chose to use Bayesian Information Criterion(BIC) to evaluate how well our model fits the data. This is because log likelihood tends to increase as we increase the number of components which might lead to overfitting and lose model generalize ability. BIC evaluates the quality of a model based on how well it fits the data while penalizing model complexity. The lower BIC is, the better is the model.

TABLE VIII
GMM CLUSTERING SUMMARY: CHURN

Data	Components	Covariance	BIC	Match Rate
Original	2	full	-2929	56%
PCA	2	diag	5822	61%
ICA	3	full	4457	61%
RP	2	full	-5188	60%
t-SNE	2	spherical	4238	59%

Table VIII shows the number of components and covariance type we chose for each dataset based on the lowest BIC values. The GMM fitted on the data reduced by SRP yields the lowest BIC(-5188) and 60% of the data points match the true labels. BIC favors simple models. It chose 2 components for most the datasets and 3 components for the data reduced by ICA. It got the the same result as K-Means. Meanwhile, the clusters we got from GMM have similar compactness with those got from k-means. They are all in the range of 3.2 to 3.7, and the inter-cluster distances all fall between 2.4 and 3.4. The compactness value of clusters got from t-SNE reduced data decreased significantly from 22.57 to 1.65. When applied to the t-SNE-transformed space, GMM can better capture the clusters' density-based patterns due to its ability to adjust the shape of clusters to fit the t-SNE-projected data, often leading to a lower compactness value.

TABLE IX
GMM-COMPACTNESS AND INTER-CLUSTER DIST: CHURN

Data	Compactness			Inter-Cluster Distance		
	C0	C1	C2	C0-C1	C0-C2	C1-C2
Original	3.62	3.75	NA	2.62	NA	NA
PCA	3.23	3.25	NA	2.62	NA	NA
ICA	3.41	3.39	3.39	3.37	2.73	2.46
RP	3.67	3.54	NA	3.22	NA	NA
t-SNE	1.65	1.65	NA	30.92	NA	NA

The GMM didn't work well on the diabetes data and the result is shown in Table X. This is because GMM assumes the data is continuous and comes from a Gaussian distribution. However, there are 18 categorical features in the diabetes data and the three continuous features have very high kurtosis values (102.09, 80.91, and 35.98, respectively), indicating heavy-tailed distributions with a high concentration of outliers. The presence of heavy-tailed distributions with extreme outliers will violate the Gaussian assumption and may lead to poor performance. High kurtosis can cause GMM to fit the data poorly, as it may try to allocate additional components to account for outliers or the tails of these distributions, resulting in overfitting or inappropriate clusters. This is why we got 16 and 21 clusters on the diabetes data. This proves our hypothesis about the performance of GMM on the diabetes data is valid. Maybe we can apply Log transformation or Box-Cox transformation to reduce skewness and kurtosis so the continuous features will be more Gaussian-like and compatible

TABLE X
GMM CLUSTERING SUMMARY: DIABETES

Data	Components	Covariance	BIC	Match Rate
Original	16	diag	-39k	65%
PCA	21	full	17k	65%
ICA	21	full	-50k	51%
RP	21	full	-52k	54%
t-SNE	6	full	27k	50%

with GMM. However, the presence of 18 categorical features still make GMM not suitable for the diabetes data.

V. ARTIFICIAL NEURAL NETWORK

After applying several dimensionality reduction techniques to the churn data, we want to know if dimensionality reduction and features extracted from clustering can help improve the performance of the ANN models. In our experiments, we fixed the architecture of the ANN and fitted the reduced datasets to get a baseline for comparison and then re-tuned the ANN models. In the PCA-ANN combination, 11 principal components were taken, and 95% of the variance in the original data was explained. The data reduced by ICA and RP reside in 14 and 12 dimensional space respectively.

TABLE XI
ANN PARAMETERS AND PERFORMANCES (CHURN)

	PCA	ICA	RP	K-means	GMM
1st hidden layer	80	90	80	100	90
2nd hidden layer	10	40	30	20	5
Learning Rate	0.008	0.001	0.015	0.006	0.008
Activation(hidden)	relu	softmax	selu	sigmoid	relu
Recall-Baseline	0.55	0.48	0.57	0.7	0.83
Recall- After Tuning	0.58	0.51	0.71	0.82	0.85
Wall Clock Time	11.4	12.54	11.8	20.5	11.5

The recall we got from fitting the ANN with the same architecture in Assignment 1 to the reduced datasets is very low. We tuned the number of neurons in each layer, learning rate and activation function to improve model performance. After we tuned the parameters, the performance improved slightly, but the model complexity increased too. In general, dimensionality reduction did not improve performance, and it even decreases the performance significantly because it discards some useful information. Important features are removed from the feature space. Linear methods like PCA, ICA and RP might fail to capture important non-linear relationships. After the linear transformation, some meaningful relationships might get distorted. All these make the learning process harder for the model and decrease the model performance. Among the three datasets, the data reduced by sparse random project has the best performance. This is probably because RP is effective in preserving approximate pairwise distances in high-dimensional data.

After we applied K-means algorithm to our churn data, we got three clusters that yields the highest CHI(79). We transformed X_train into three features, add the three features, cluster labels, distances to their assigned centroid and Silhouette score on top of the original features X_train to create kmeans_train. We trained and tuned our ANN model on kmeans_train. We extracted the same features from X_test, and then test the model performance on kmeans_test. After fitting GMM to our churn data, we got two clusters that yields the lowest BIC(-2929). We added cluster labels, log likelihood, mahalanobis distances and probability ratio on top of the original features X_train to create X_train_gmm. We trained and tuned our ANN model on X_train_gmm. We extracted the same features from X_test, and then tested the model performance on X_test_gmm. The ANN performance decreased. It seems that all these cluster labels and new features didn't help our network learn better about the data. The recall we got in assignment 1 is 0.87, but the recall we got after adding all these new features are 0.82 and 0.85. The fit time was 13.8 for the original data, and we got 20.5 and 11.5 for the new datasets. There is no significant difference in the wall clock time. Therefore, the new features extracted from clustering didn't improve the model performance and computational efficiency.

VI. CONCLUSIONS

RP-Kmeans, RP-GMM and RP-ANN outperform other methods combinations on the churn dataset. RP-Kmeans also outperform other combinations on the diabetes data. GMM and K-Means generate similar clusters on the churn data. Due to the presence of big number of categorical features and the high kurtosis in the continuous features in the diabetes data, GMM struggles to fit the data. It has to keep adding more components to approximate the density distribution of points on the tails. This proves our hypothesis about GMM will struggle on the diabetes data. Other clustering algorithms like K-Modes and K-Protypes might work better than K-Means and GMM. We didn't demonstrate it here due to the page limits. From Fig.5, 6 and 11, we can see that there is a significant overlap between the two classes in both datasets. Both dataset don't exhibit clear separability in the 3D PCA space or t-SNE space. This explains why the match rate between the cluster labels and true labels fall in the range of 50% to 65%. If the original datasets are not separable, but we forced them into two or three clusters, there certainly will have a big amount of cluster labels don't match with the true labels.

Dimensionality Reduction and new features extracted from clustering don't always improve the performance of downstream machine learning applications. These techniques should be used very carefully in practice.

REFERENCES

[1] Wikipedia
[2] I.K.Fodor, "A survey of dimension reduction techniques, " Lawrence Livermore National Laboratory, June 2002
[3] Wikipedia
[4] Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie, "The Elements of Statistical Learning ", Chapter 14.3 Cluster Analysis, Page 509
[5] Wikipedia