[redacted] Unsupervised Learning and Dimension Reduction

Abdullatif Jarkas
[redacted]
Atlanta, Georgia, USA
[redacted]

I. Introduction

In this study, we explore different unsupervised learning methods (K-Means and Expectation Maximization) on the datasets used in our previous studies, Diabetes Detection and Loan Approvals. Unlike in our previous assignments where we applied supervised learning in order to approximate a function that well describes our dataset, we are applying unsupervised learning methods here which seek to group together, or cluster, our datapoints based on the patterns in our features.

Along with unsupervised learning, we will be exploring and assessing several dimension reduction methods such as Principal Component Analysis, Independent Component Analysis, and finally Random Projection. Many times our datasets come with countless features such that many of which do not provide significant information towards our objective. These dimension reduction methods, in a sense, remove the excess fat to keep the main components that hold relative impact towards our objectives.

II. Dataset Overview

Like in previous assignments, we will be using our Diabetes and Loan Approvals dataset. So in this assignment, we will induce a brief overview:

- **Diabetes (Pima)**: A dataset of people of Pima ethnicity containing several health indicators, or features, such as Number of Pregnancies, Glucose Level, Blood Pressure, Skin Thickness, Insulin Level, BMI (Body Mass Index), Diabetes Pedigree Function, and Age. Given those datapoints, our target label is if the individual is diabetic or not (binary). Relatively small dataset with 764 samples and 8 features.
- **Loan Approvals**: The loan approvals dataset contains features such as Applicant Income, Co-applicant Income, Loan Amount, Loan Amount Term, Credit History, Education, Self Employed, Property Area, and Loan Status. The target variable here is whether the loan was approved or not, also binary. Relatively large dataset with 4269 samples and 11 features.

Both datasets are binary classification problems which are great for unsupervised learning. Even better is that both datasets have differing sample sizes (764 vs. 4269 respectively), which should offer some insight on method performance across sample sizes.

Data Pre-Processing: Our pre-processing is quite brief. The primary implication here is to exclude the target labels from our dataset as we are applying unsupervised learning and do not want to use labels as a predictors. Our Diabetes dataset is all numerical, so no type conversion was needed, however we had a few non-numerical binary categorical features in our Loans dataset that we transformed to binary integers. Finally, both dataset matrices were standard scaled to prevent any model sensitivity with our algorithms.

III. Algorithm Overview

We deployed both clustering algorithms and dimensionality reduction techniques (both exclusively and inclusively) to explore patterns within the Diabetes and Loan Approvals datasets. Below is an overview of each algorithm and its purpose.

A. Clustering Algorithms

Clustering aims to group data points into clusters based on feature similarity and relative patterns.

1) **K-Means Clustering**: K-Means segments its data on an arbitrary number of clusters. The primary goal of K-means is to minimize the variance within each cluster of datapoints. It works by initially randomly assigning a centroid and iteratively assigns new datapoints to its nearest cluster and then recalculates the centroid given the new information. It is highly cost-efficient due to its trivial calculations based on distance, which translates applicability on larger datasets. However, K-means does seem to consistently form spherical clusters, which may not be beneficial as most data are not spherically segmented.
2) **Expectation Maximization (Gaussian Mixture Model - GMM)**: Expectation Maximization (EM) takes a more probabilistic approach by clustering datapoints based on the likelihood of it belonging to a normal distribution. EM works by iteratively estimating datapoint assignment to an arbitrary number of clusters while updating its distribution. Unlike K-Means, EM is much better at capturing non-spherical relationships which include overlapping and complex structures. However,

Identify applicable funding agency here. If none, delete this.

given its calculation based on a given distribution, the computation cost significantly increases as the number of datapoints increases.

B. *Dimensionality Reduction Techniques*

As mentioned before, dimension reduction methods attempt to remove irrelevant features and retain imperative components that seem to hold most impact towards our objective. However, each algorithm approaches and solves this problem quite differently, furthermore revealing preferences and disadvantages for each method based on different contexts.

1) **Principal Component Analysis (PCA)**: PCA is a linear dimensionality reduction method that projects features into a lower dimension space. It works by maximizing the variance retained. In a sense, it merges features together by features that are highly correlated to one another. Drawbacks for PCA include its high computation expense in comparison to other methods. We will assess PCA's performance by plotting the explained variance ratio, which is a measurement of how much information is retained at different dimensions and components.
2) **Independent Component Analysis (ICA)**: ICA is also a linear dimension reduction method but unlike PCA, ICA seeks feature independence by maximizing a non-normal distribution. This method is exemplified in the ”cocktail party problem” (from lecture) where microphones capture distinct yet muffled and compounded sounds from different regions. The goal of ICA here is to filter out noise from other regions to isolate unique sounds to each microphone. Drawbacks of ICA include sensitivity to data quality and assumption that features and information are independent to each other. We'll evaluate ICA using a kurtosis plot to assess non-Gaussian (non-Normal) distribution, in other words, assess the distinctiveness of each component.
3) **Random Projection (RP)**: Random Projection is a much more computationally efficient approach which attempts to mimic PCA's final results through a randomized approach. As our dimensions increase, so does the likelihood of selecting a near orthogonal vector to apply dimension reduction on. Because this is a randomized approach, its performance is highly dependent on its random selection of matrices which can be its drawback. We will assess RP by Reconstruction Error, which measures how well we can reconstruct our original feature set from our Randomly Projected feature set.

Overall, each method carries its own unique approach, drawbacks, and attributes, and we will be assessing them on our two datasets which vary in sample size to hopefully uncover preferences based on sample size if evident.

IV. Hypotheses

1) **Experiment 1**: K-means will outperform in clustering due to its trivial and inherent objective to minimize variance within each cluster, making it segregation focused.
2) **Experiment 2**: PCA will be the best dimensionality reduction method as I believe its inherent structure of capturing the most variance will translate to capturing the most information, which I believe is superior to its counterparts grounded on independence and randomization.
3) **Experiment 3**: Given my previous two hypotheses, I believe combining both K-means and PCA will incur the most optimal results in clustering.
4) **Experiment 4**: Applying PCA to our dataset prior to our Neural Network will yield the best results as PCA is focused on retaining as much variance (information) as possible in comparison to its counterparts, leaving our Neural Network with the significant content of our information.
5) **Experiment 5**: As my previous hypotheses do state, K-means will remain the best performer in adding information to our dataset for Neural Network classification.

V. Experimentation

A. *Exp. 1: Apply Clustering Algorithms (K-Means, EM)*

Objective: To test our hypothesis that K-Means will outperform Expectation Maximization (EM), we will apply both methods to binary cluster both of our datasets that have contrasting sample sizes. Then we will assess their performance based on Silhouette score, which measures how similar a point is to its classified cluster in comparison to the opposing cluster.

1) Results and Detailed Interpretation:

Silhouette Score (Clustering Effectiveness)

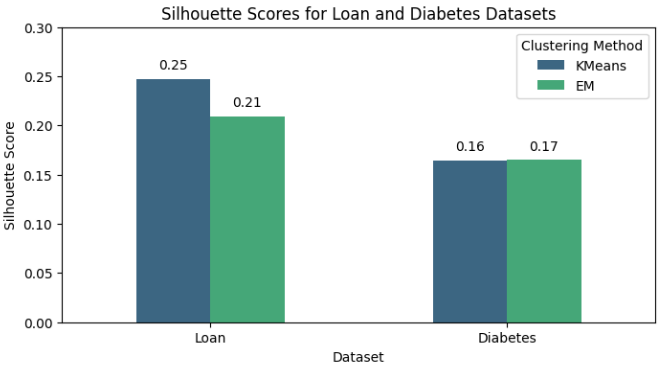

Fig. 1. Silhouette Scores for K-means and EM across both datasets

As shown in *Figure 1*, The results suggest that K-Means forms better-defined clusters compared to EM across both datasets at Silhouette Scores of 0.25 vs 0.21 for the loans dataset and 0.18 vs 0.14 for the diabetes dataset. This may be due to K-Means' being naturally dependent on Euclidean distance, which can be more cost efficient, trivial, and effective. Given that K-Means operates by forming spherical clustering, this may indicate that our dataset is best suited for spherical clustering rather than EM's more adaptive clustering style.

*2) **Summary and Hypothesis evaluation**:* Overall, K-means did indeed provide the best Silhouette score, indicating superior clustering. This is most likely evident due to K-means' nature in minimizing variance across clusters and forming distinct spherical clusters. Leaving us unable to reject our hypothesis that K-means will outperform EM generally speaking across most tasks.

*B. **Exp. 2: Apply Dimensionality Reduction (PCA, ICA, RP)***

Objective: Now, we will apply our dimension reduction methods to our datasets and then independently assess the quality of each method in terms of each method's respective attributes and metrics.

*1) **Results and Detailed Interpretation**:*

1. PCA - Explained Variance through Scree Plot

Here we will evaluate how much explained variance (main property of PCA) is captured at each component. This will indicate ideal components to select and which components amass the most information.

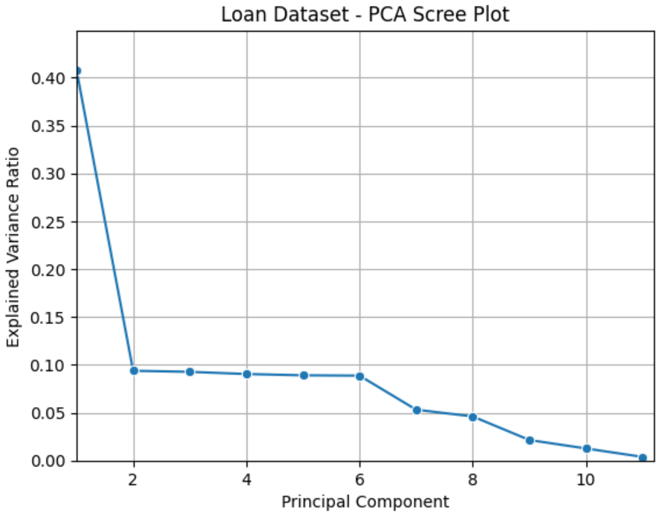

Fig. 2. Scree plot (Explained Variance) for PCA on Loan dataset

- **Loan Dataset**: Shown in *Figure 2*, The first principal component in the Loan dataset explains 40.82% of the total variance, and the next 5 components (2-6 inclusively) account for nearly another 50% of the variance with the first 6 components carrying nearly 90% of the variance and information. This finding reveals that components 1-6 capture the majority of the variance (or information) quite well and information loss is also somewhat minimal after 6 components. This indicates that the Loans dataset (a substantial size) could benefit from dimension reduction from 11 components to anything under 6 components, with the first component carrying the most information.
- **Diabetes Dataset**: As shown in Figure 3, for the Diabetes dataset, the first principal component captured 26.18% of the variance, and the cumulative variance across the top five components is about 81.1%. In comparison to the Loans dataset, PCA did carry less variance in the first component, the cumulative variance of components 1-6 carried nearly the same variance as the loans dataset at 89.6%. This implies that PCA is beneficial for the Diabetes dataset, however, because the variance is more distributed amongst the components in comparison to the Loans dataset, this indicates that more components may be needed to capture similar information to our Loans dataset with fewer components.

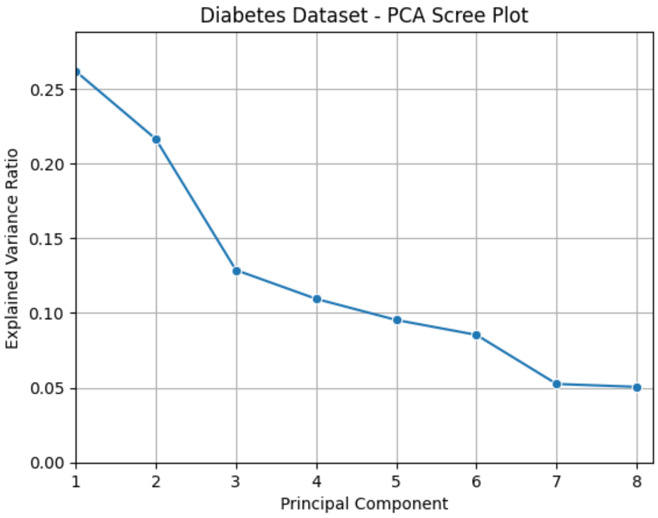

Fig. 3. Scree plot (Explained Variance) for PCA on Diabetes dataset

2. ICA (Independent Component Analysis) - Kurtosis:

Kurtosis is a measure of relative independence of components, our goal will be to attempt to find the components that maximize Kurtosis which will translate to our ICA's performance in retaing key information from our data.

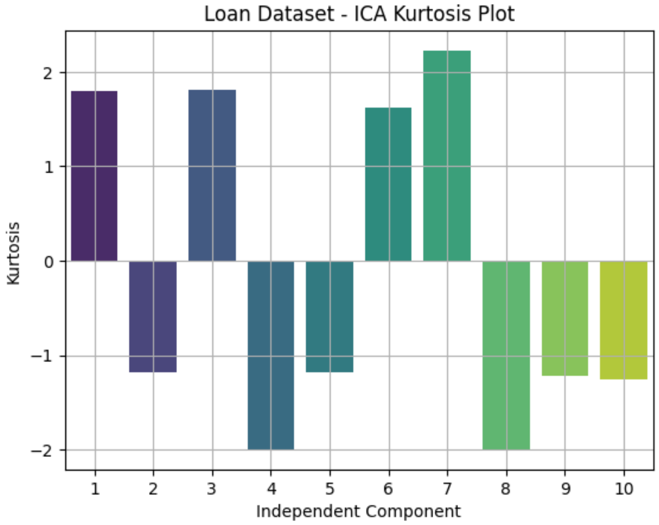

Fig. 4. Kurtosis plot for ICA on Loan dataset

- **Loan Dataset**: As shown in the *Figure 4* above, we see positive Kurtosis values at 1,3,6, and 7 components, with

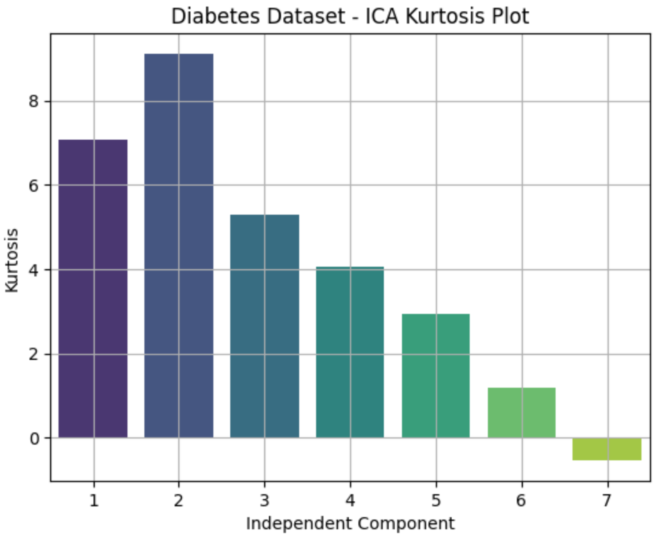

Fig. 5. Kurtosis plot for ICA on Diabetes dataset

7 components being the peak at 2.27, followed by 2 and 3 components at 1.92 kurtosis. This indicates that to maximize independence amongst our components, we can see the most benefit by reducing our dimensions from 11 components initially to 7 components. However, for data exploratory purposes, 3 components will be the most ideal as it does provide a relatively high kurtosis value while also being visually plottable as it is 3-dimensional.

- **Diabetes Dataset**: As for in *Figure 5*, we see a contrasting pattern in comparison to our Loans dataset. We see a somewhat skewed normal distribution-like shape, peaking at two components and gradually declining till 7 components. This may be an indication that our Diabetes features seem to potentially be independently segmented quite well. The majority of components result in a positive kurtosis value, with our kurtosis specifically peaking at 2 components at a kurtosis score of 8.72, which is significantly higher than our second best performing component at 6.57. This indicates that, when maximizing for independence in relevance to ICA, our Diabetes dataset will see the most optimal results when reduced to 2 components.

3. Random Projection (RP) - Reconstruction Error: Here we will be assessing Reconstruction error, which, given our Randomly Projected transformed features, measures the error in transforming back to our original feature set.

- **Loan Dataset**: When looking at the reconstruction error plot in *Figure 6* above for our Loans dataset, we can first look at the scale of our error, revealing that all components have a near 0 error with our component 1 revealing the global maximum error of 0.0045. As the components approach our original number of features, we see the reconstruction error (nearly) linearly decrease to a minimum of 0.002 at component 10. However, given that the error is insignificant, we are incentivized to approach smaller components as the error between 10 components and 1 component is only 0.0025 and the increase is only 2.2 folds.

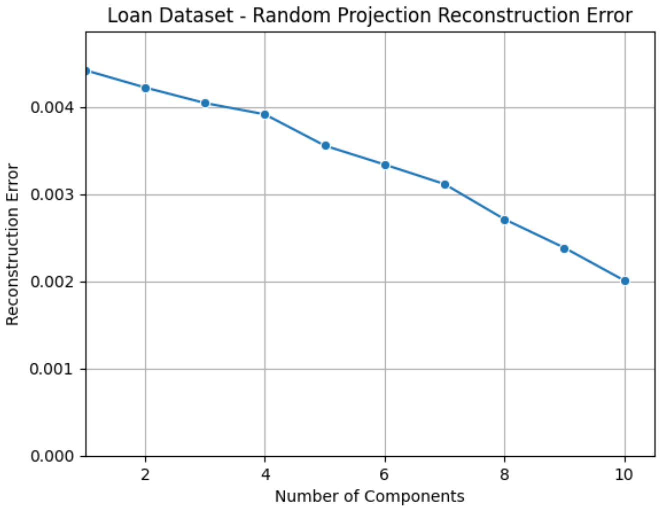

Fig. 6. Reconstruction Error plot for RP on Loan dataset

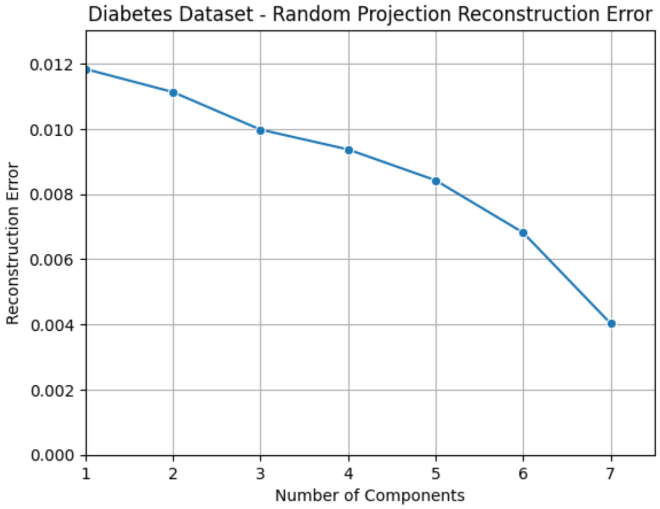

Fig. 7. Reconstruction Error plot for RP on Diabetes dataset

- **Diabetes Dataset**: Similarly to the Loans dataset, in *Figure 7*, reconstruction error approaches zero as the component count approaches the initial number of features. However, the scale of error is much larger here as the number of components decreases. The error increases exactly 3 folds from component 7 to component 1. And unlike our Loans data, the error here seems to follow a negative parabolic pattern.

*2) **Summary and Hypothesis Evaluation:*** The dimensionality reduction results highlight unique patterns in each dataset. PCA reveals that the Loan dataset's first six components capture nearly 90% of the variance, with the first component capturing 40% of the variance, supporting dimensionality reduction from 11 to 6 components with minimal loss in information. For the Diabetes dataset, the top six components also accounted for similar variance at 89.6%, but variance was more evenly spread, requiring slightly more components for sufficient capture of information. ICA showed optimal independence with five components for the Loan dataset and four for Diabetes, suggesting more distinct feature structures in the Diabetes dataset. RP maintained low reconstruction error across components for both datasets, with slightly higher error in Diabetes, potentially due to smaller sample size. Overall, with our goal being to capture as much information at lower dimension, we see preference in ICA currently, with PCA following shortly after, and RP remaining competitive. However, we cannot conclusively select one as further investigation is needed.

Overall, because each dimension reduction method is evaluated on its unique metric of performance, we are unable to compare the two. However, we can infer that for all three methods, the first 3 components carry a substantial amount of information. So we will be using 3 components per dimension reduction technique in later testing. So, we cannot conclusively reject nor fail to reject our hypothesis here.

C. Exp. 3: Clustering on Reduced Dimensions

Objective: Here we will be conducting a comparative analysis on K-means vs EM using all three dimension reduction techniques. We will also be using our original scores from experiment 1 which apply the clustering algorithms without any dimension reduction as a control for comparison. Like in experiment 1, we will be assessing based on Silhouette Score which, as a reminder, measures how well grouped our clusters are. A higher score indicates better clustering. For each Reduction Dimension method, we will be reducing to 3 components as our previous exploration saw substantial information capture in those components.

1) Results and Detailed Interpretation:

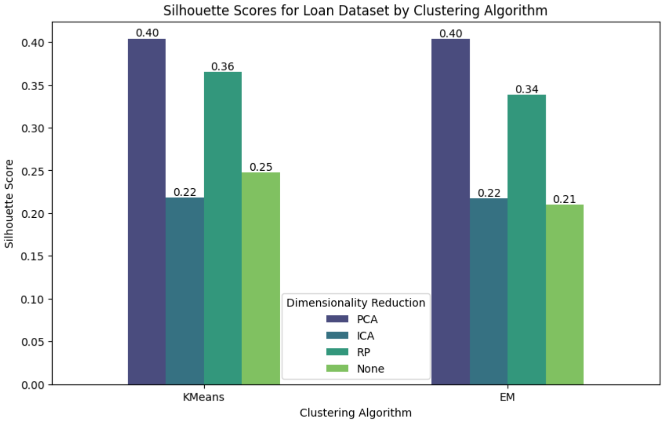

Fig. 8. Silhouette Scores per Clutering Algorithm per Dimension Reduction method (3 components) on Loan dataset

1) **Loan Dataset**: As shown above in *Figure 8* on the Loans dataset, we see that for both clustering algorithms, PCA outperformed all other dimension reduction techniques when it comes to Silhouette score, achieving 0.4 across both K-means and EM. With RP coming in a close second with 0.36 and 0.34 respectively for both clustering algorithms, and ICA and No Dimension Reduction explaining similar scores at suboptimal levels of 0.22 vs 0.25 and 0.22 vs. 0.21 respectively across the algorithms. Now comparing both clustering algorithms cumulatively, and like the results revealed in Experiment 1, K-means out-performed EM by 0.06 cumulative Silhouette scores. These findings indicate that for datasets of larger sample size (like the Loans dataset), they would benefit significantly more by PCA, and of the clustering methods, it would benefit marginally more utilizing K-means clustering.
2) **Diabetes Dataset**: Unlike the Loans dataset, we see a contrasting result in terms of which dimension reduction technique is preferable as seen in *Figure 9*. Instead of PCA being the leader here, RP significantly led for each clustering algorithm at 0.4 and 0.32 silhouette scores respectively. However, of the remaining algorithms, PCA took a humble second place in Silhouette Score achieving 0.26 for both, while ICA marginally trailing at 0.23 and 0.22 respectively, while silhouette scores from our Experiment 1 significantly underperformed the group with 0.18 and 0.14 respectively.
As revealed in our previous experiments comparing both clustering algorithms, the cumulative silhouette scores of K-means also outperformed the ones of EM by 0.13.
This indicates that maybe for smaller datasets (such as our Diabetes), RP may be preferred over other reduction techniques. Which is somewhat strange as RP's strong point is its computational efficiency, so RP's application (not necessarily performance) would be better off applied to datasets of larger sample size. However, this finding could be due to many other factors like random chance, data quality and structure, variance in data, etc. To conclusively select RP for smaller dataset sizes, we

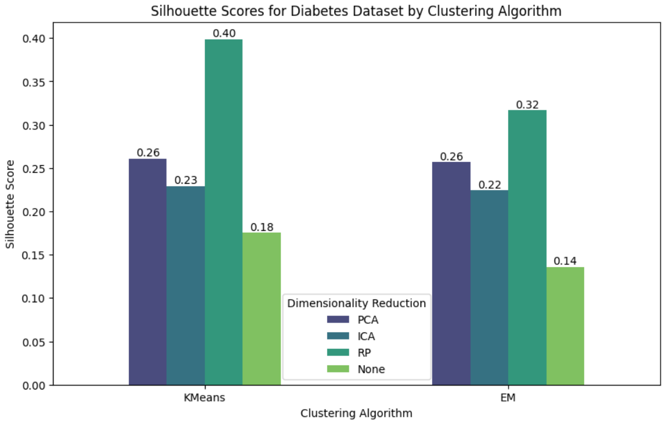

Fig. 9. Silhouette Scores per Clutering Algorithm per Dimension Reduction method (3 components) on Diabetes dataset

will need to experiment with more datasets that also share contrasting sample sizes. As for the clustering algorithms, K-means continues to show superiority over EM in Silhouette Scores, meaning K-means may be a preferred choice for binary clustering regardless of sample size.

*2) **Summary and Hypothesis Evaluation:*** Overall, the results show that PCA tends to work consistently across both K-Means and EM algorithms across both datasets. However, PCA saw its preference on the much larger Loans dataset likely due to its ability to capture the majority of variance in fewer components, which may be preferable on larger sample and feature sizes. RP also remained consistent across both clustering algorithms and datasets, however RP did see its preference with the much smaller Diabetes dataset. Likely due to the fact that its core is randomization, which may be suitable when less information or data is provided to begin with as seen in the Diabetes dataset. ICA underperformed across both datasets and algorithms, suggesting that it may not capture essential information and structure effectively of the original dataset. K-means definitively outperformed EM, while PCA and RP outperformed the rest while retaining contrasting results on our two datasets with contrasting sample sizes.

So, in regard to our hypothesis, we fail to reject our hypothesis that K-means will outperform EM, however, with RP and PCA both displaying consistent results yet inconclusive preference, we cannot prefer one over the other just yet, so we should reject our hypothesis that PCA will always outperform.

*D. **Exp. 4: Neural Network Preprocessed with Dimension Reduction***

Objective: Now, we are shifting our objective from how well our dimension reduction techniques work in clustering to how well they work in classification using our neural network from Assignment 1. We will be using our larger Loans dataset for this classification problem. We will also be assessing performance based on the following metrics: BCE Loss (Train vs. Test), Accuracy, F1 Score, and Area Under Curve, in order to find which dimension reduction technique classified our dataset the best.

*1) **PCA:*** As shown in *Figure 10* above, we see the train vs test loss (Binary Cross Entropy) over epoch iterations. As shown in the curve, a steep fall in our BCE test loss till about 200 epochs, in which, the test loss gradually decreases to 0.49 as it approaches 1000 epochs. Train loss also showed a similar pattern, however with much more variability.

Now, looking at our next plot, we see the results for PCA applied to our dataset and training a Neural Network for 1000 epochs. Our Test Accuracy quickly stabilized at 72.60% at around 150 epochs, showing a relatively high proportion of correct test classifications. Our F1 Score of 0.7869 also reached a local optimal score at around 100 epochs, meaning the model effectively identifies true positives while mitigating false positives and false negatives. The AUC of 0.8086 indicates a solid ability to distinguish between classes across various thresholds, which is a decent score, however, our classes are relatively trivial as they are binary. Observing the progression over epochs, it is clear that the model quickly converged and maintained stable performance across all metrics. However, to see if these numbers are of actual decency, we must compare with the other dimension reduction methods' results.

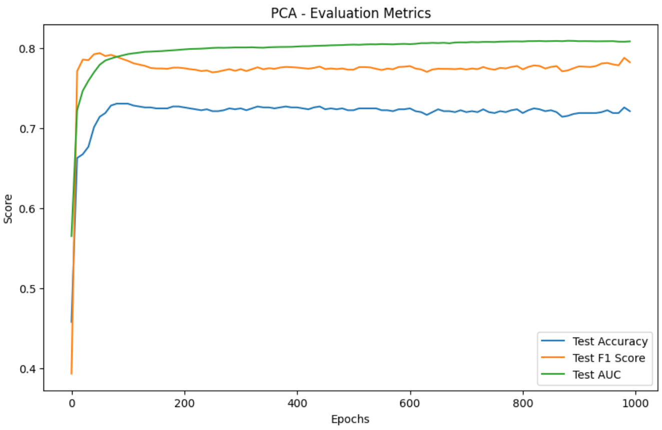

Fig. 10. PCA (3 components) pre-proccessed Neural Network Classification metrics on Loan dataset

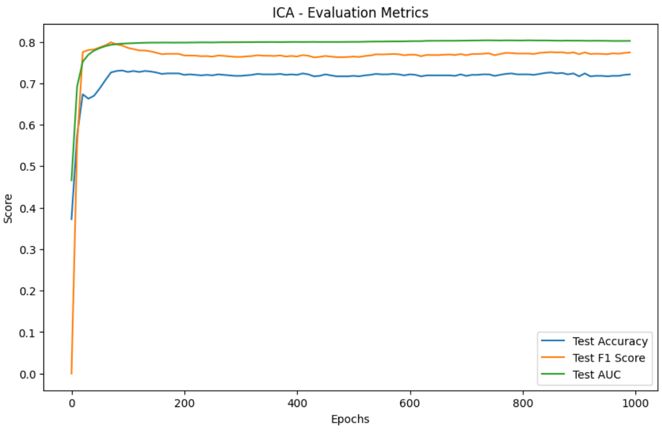

Fig. 11. ICA (3 components) pre-proccessed Neural Network Classification metrics (Test Accuracy, F1 Score, Area Under Curve) on Loan dataset

*2) **ICA:*** Like PCA, ICA visualized a steep drop in test loss at around 150 epochs, in which it gradually decreased as it approached 1000 epochs while achieving a minimum test loss of 0.504, which is only a marginal decrease from our PCA test loss at 0.49. Train loss also showed a similar pattern, however with similar variability to PCA's train loss over time.

Now looking at the ICA Metrics, we also see similar scores to that of PCA, however, they marginally underperformed with test accuracy reaching a high of 72.1%, F1 of 0.7742, and an AUC of 0.802. Overall, negligible difference between ICA and PCA in our results. Revealing that although they have different methods of dimension reduction, they did achieve nearly identical results in classification.

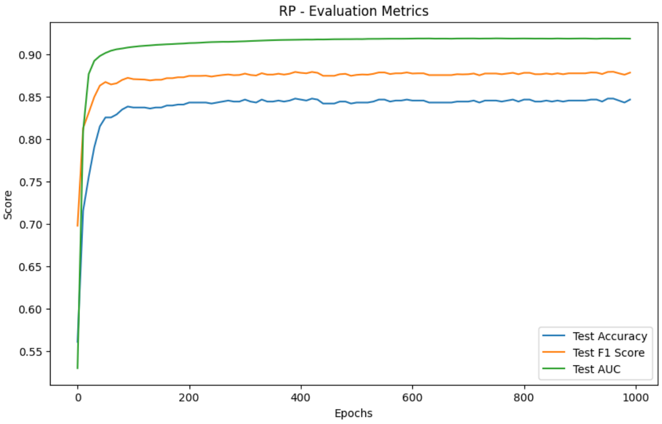

Fig. 12. RP (3 components) pre-proccessed Neural Network Classification metrics (Test Accuracy, F1 Score, Area Under Curve) on Loan dataset

*3) **RP**:* Now on to RP's train vs test loss (BCE). Unlike PCA and ICA, RP reached superior relative test loss faster and much more efficiently. We do see the same pattern visually for the test loss, however, the performance, scale, and metrics showed their superiority with RP achieving a global minimum test loss of 0.3482, which is significantly lower than that of PCA and ICA which only achieved a test loss of near 0.50 after 1000 epochs. Train loss here uniquely kept a large distance from the test loss, indicating optimal fit and predictions on test data.

Finally, on to the metrics for RP. RP outperformed both PCA and ICA in every metric and by a significant difference. RP achieved a global high test accuracy of 84.43%, an F1 Score of 0.877, and an AUC of 0.918. Indicating accurate predictions, and optimal mitigation of false positives and false negatives, efficiently distinguishing both classes. To put that in comparative perspective, RP outperformed in every metric here by around 15% when compared to PCA and ICA. This is good news as RP is the most computationally efficient of the group, but now we see that it is also the best performing of the group in classification.

*4) **Comparison with Assignment 1 Results**:* Given that our Assignment 1 resulted in 95.90% test accuracy, and a BCE test loss of 0.0921. We can see that although RP did provide optimal global results amongst its dimension reduction peers with 84.43% test accuracy and 0.3482 BCE test loss, RP still underperformed our original network results by around 10% in test accuracy and increased nearly 3 folds in test loss. This indicates that although we reduced our dimensions and retained most of the information, our neural network classifier much preferred our original, larger, and more nuanced feature set. Meaning, some substantial amount of information was left out in dimension reduction.

*5) **Summary and Hypothesis Evaluation**:* In terms of Neural Network classification, the results showed that PCA and ICA, despite using distinct approaches, resulted in near-identical performance. This suggested that they both retained similar essential data features for classification. On the other hand, RP outperformed both by achieving faster convergence, superior classification metrics, while being extremely computationally inexpensive.

This indicates RP to be the most optimal decision as it outperformed in every metric tested while remaining efficient, which rejected our hypothesis that PCA would outperform all due to its variance-based algorithm.

Overall, our findings suggested that while PCA and ICA are solid options for dimensionality reduction, RP is best-suited for this specific Loans dataset.

E. Exp. 5: Neural Network with Added Clustered Features

Objective: In this step, we concatenated the cluster labels generated from K-Means and EM clustering algorithms as additional features in our Loans dataset. By including these clusters' labels, we attempted to assess whether the appended feature and label set captured could improve the neural network's classification performance. Both K-Means and EM were used to generate clusters on the scaled data, and these labels were appended as new columns to the dataset before training the neural network.

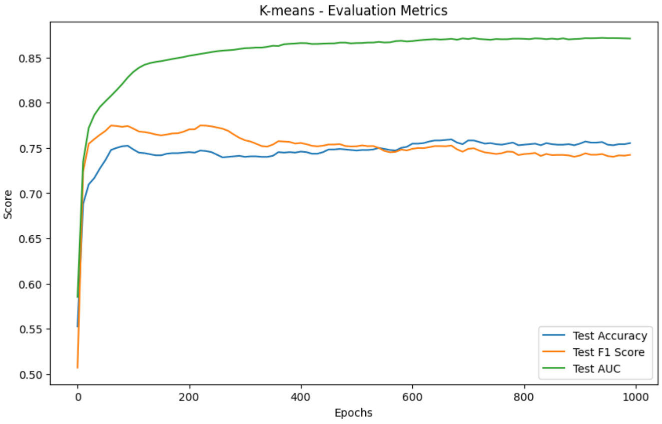

Fig. 13. K-means Appended Features Neural Network Classification metrics (Test Accuracy, F1 Score, Area Under Curve) on Loan dataset

*1) **K-Means Clustering Performance**:* With K-Means clusters appended as features, the neural network achieved a minimum BCE test loss of 0.3888, with final metrics of 75.59% accuracy, 0.7428 F1 Score, and an AUC of 0.8707. This relatively high AUC score indicates that the model was effective at distinguishing between classes with the help of K-Means clustering labels. The F1 Score of 0.7428 demonstrates a reasonable balance in mitigating false positives and false negatives, which suggests that K-Means clusters may have provided meaningful data to help the model capture relevant information without overfitting.

*2) **EM Clustering Performance**:* Now using EM clusters as additional features, our neural network achieved a minimum test loss of 0.3988, which slightly underperformed K-means by 1 point. Indicating little to no difference in test loss. However, we will now need to assess our metrics. The final metrics with EM (*Figure 14*) were a slightly lower 75.22% accuracy,

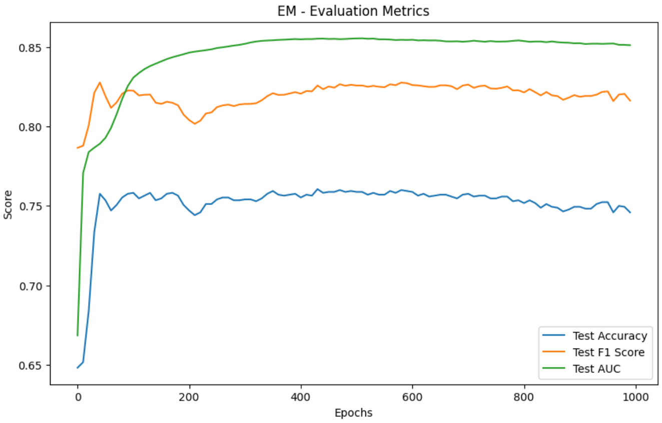

Fig. 14. EM Appended Features Neural Network Classification metrics (Test Accuracy, F1 Score, Area Under Curve) on Loan dataset

a significantly higher F1 Score of 0.8177, and a lower AUC of 0.8516. Compared to K-Means, the EM clusters resulted in a higher F1 Score, indicating that the model with EM labels performed much better at capturing true positives while mitigating false positives and false negatives. However, the AUC was reasonably lower than that achieved with K-Means, suggesting that K-Means clusters were more evident across various thresholds. Although the higher F1 Score here suggests that the EM clusters may have more complex information, this may not have improved the overall discrimination capability as indicated by AUC.

*3) **Comparison with Assignment 1 Results**:* Given that our Assignment 1 resulted in 95.90% test accuracy, and a BCE test loss of 0.0921, both of our clustering algorithms yielded similar results to each other at roughly 75% and 0.4 BCE Loss. Indicating significant underperformance and indicating that adding our clustered information offered more nuanced features to our original dataset that differed from the patterns picked up from our neural network. Overall, this experiment revealed that clustering does find points who share a relative pattern or relationship with, however, that does not translate into classification 1 to 1 as it underperformed our previous experiment substantially.

*4) **Comparison, Justification, and Hypothesis Evaluation**:* Now, comparing the two clustering methods, K-means clusters led to a slightly better AUC and lower test loss, implying that K-means is more preferable to effectively segregate classes while clustering. In contrast, EM clustering provided a higher F1 Score, suggesting that it captured the nitty-gritty patterns within our data which improved our neural network's ability to identify positive cases while mitigating false positives and false negatives.

We can make sense of our findings due to the nature and approach of each algorithm. As we previously spoke about the strengths and drawbacks of each clustering algorithm, we spoke on K-means' reliance on forming spherical clusters, which may be preferable for segregation (high AUC), however, it may not pick up the details of nuanced data, especially data points near the decision boundary as evident in the relatively low F1 Score. EM, on the other hand, with its probabilistic approach, is able to cluster a little more adaptively and may not be so deterministic when it comes to clustering, especially near the decision boundary, which again was evident in its high F1 score.

Overall, as we experienced relatively similar accuracy, we face a trade-off between segregation and capturing nuanced data, especially near the decision boundary, for both algorithms. And our decision to prefer one over the other is entirely based on our objective, so we will need to reject our hypothesis of K-means always being more preferable.

VI. Conclusion

In this report, we explored unsupervised learning clustering methods such as K-means and EM on our Diabetes and Loan Approval datasets. On top of our experimentation of clustering methods, we assessed multiple dimension reduction methods such as PCA, ICA, and RP. Through our findings, K-means consistently outperformed EM in all experiments excluding our neural network classification experiment in which K-means was better at segregation than EM, however, subpar in terms of classifying nuanced data as EM performed well there. In terms of our Dimension Reduction methods, we saw both PCA and RP outperform ICA in clustering, with PCA having preference over our larger Loans dataset while RP having preference over our smaller Diabetes dataset. However, in neural network classification, RP significantly outperformed its rivals on our Loans dataset, assuring its place as the optimal method for neural network classification on our Loans dataset.

When compared to our Assignment 1 for Neural Network classification, we saw that RP did retain most of the data's information, however, it did underperform our Neural Network classifier trained on the original dataset. Also, for our clustering algorithm adding features to our dataset to use for Neural Network classification, we see that it significantly underperformed in test accuracy and test loss compared to our Assignment 1 results.

As for our hypotheses, in regards to clustering, our hypotheses infers to be true which assumed K-means to outperform in nearly every case over EM given it's nature of minimizing cluster variance and trivial computation. However, in regards to dimension reduction, we saw mixed results between PCA and RP across different experiments, so we have to reject our hypothesis that PCA will outperform in every case due to it's ability to find impactful features and reduce feature complexity.

This reveals the overall conclusion of this study, in which unsupervised learning through clustering can be useful to find patterns and segregate between clusters, however, our findings infer that unsupervised learning is inferior to supervised learning in classification objectives. Also, while dimension reduction can be a optimal method to shrink your feature size for computation purposes, however, our findings infer that dimension reduction will likely under-perform using the original dataset for classification in many cases.