Exploring Unsupervised Learning and Dimensionality Reduction Algorithms: Weather and Housing Price Classification

Aditya Tomar
CS7641: Assignment 3
atomar45@gatech.edu

I. Introduction

Life is made up entirely of uncertainties. It has many ups and downs, as well as crests and valleys. However, as humans, it is our nature to want to make sense of the world around us and find answers to how and why specific incidents happen. With so much to be aware of, it is easy to lose track of what problem we are trying to solve. To lessen the cognitive load, data scientists and machine learning engineers report their findings and put them into datasets and design models that can take this vast information and distill it to a compressed set of variables that capture the more valuable features in a process called dimensionality reduction to help us solve real-life problems and make sense of the world. Machine learning is the application of computational techniques to predict outcomes before they happen based on what has happened before, and there are two branches of this: Supervised and Unsupervised Learning. Assignment 1 explored how we can take our observations of the past (target variable) and see if the algorithm can recognize patterns and future outcomes. This assignment explores how we can use dimensionality reduction algorithms to identify and learn patterns in unlabeled data.

This study is unique in that it aims to compare the performance of different algorithms. Specifically, it seeks to determine if applying clustering to datasets before dimensionality reduction enhances the performance of neural networks. The study also investigates whether the components created by PCA, ICA, and RP combined with KMeans and EM outperform the Neural Network Learner from Assignment 1.

To explore my hypotheses, I chose two datasets. One is the AMES Housing dataset, a classification problem containing many features like square footage, garage space, number of bathrooms, etc., factors that help determine a home's fiscal value. The second is a time series dataset that contains meteorological data from numerous European cities over a year.

II. Step 1

To determine the optimal number of clusters and components for clustering algorithms, I performed a quick EDA analysis on the housing and weather datasets to see the two essential features I could use in my analysis. After standardizing the datasets using StandardScaler from the Scikit-Learn library, I created a correlation matrix, including the target column, to show their top features. I used the average temperature and humidity characteristics from the weather data and the home's overall quality and size characteristics from the housing dataset. All these features strongly correlated with the target column from their respective datasets. I dropped those target columns to prepare for the implementation of KMeans and Expectation Maximization (EM) algorithms.

KMeans and EM are widely used clustering algorithms to analyze very noisy datasets and perform high dimensionality analysis. For my purposes, I will see how these clustering algorithms perform before any dimensional reduction. KMeans is very simple in assigning different data points to each cluster center and does well in cases with more rounder clusters. EM is a more versatile variant that can handle irregular-shaped clusters and non-normal distributions. I think I will see EM as a better candidate for the housing dataset as it has more complex distributions of features than KMeans.

A. KMeans

In both the weather and housing datasets, I created a function that could find the most optimal number of clusters for KMeans clustering. I implemented the Elbow Method and the Silhouette Score Method to compare k values derived from different approaches.

The Elbow Method is a very straightforward approach as it is the sum of the squared distances between the datapoint and the centroid of the nearest cluster. You plot the sum of all the squared distances against different numbers of clusters. The elbow is just the point where there are diminishing returns as you add more clusters.

The Silhouette Method is more robust as you are trying to see how well matched each datapoint is to its respective cluster. The higher the value, the better the data points are well-matched to their clusters; the lower the value, the more dissimilar the data points. We should find smaller values for k to maximize the silhouette score.

Not only did the SS and Elbow Method return the same value on housing and weather data, but the value of clusters k was the same, 2. Since this was originally a binary classification problem, having two clusters does make sense; it means that both datasets separate into two clusters naturally.

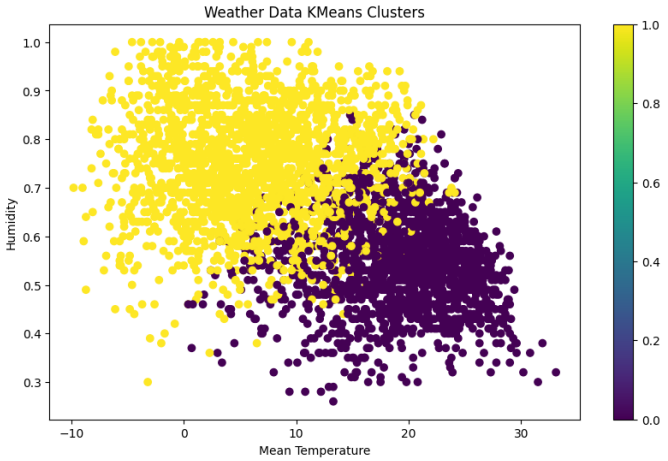

Fig. 1: Weather Data – KMeans Clusters

Fig. 2: Housing Data – KMeans Clusters

In Fig.1, we can see there is a clear separation between the classes for the mean temperature and humidity features.

In Fig.2, we see something peculiar because the data is discrete and not continuous like the features in the weather data. We know this kind of dotted grid but can see a separation between the two.

B. Expectation Maximization (EM)

I used two methods to optimize the number of components for the Gaussian Mixture Model I used for the EM method, the Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC). From the lectures, I know that the BIC uses Bayes analysis to estimate the posterior probability function as accurate. It is better in cases where we have smaller datasets, so the models are simple enough. Akaike also penalizes parameters but stresses the goodness of fit more so that it does not penalize the parameters too harshly. So, it is better to use that optimization method for larger datasets, and it helps prevent overfitting because of the focus on goodness of fit. I wanted to compare how the different scoring affects the number of components for the EM method. Sure enough, it did not change much as both returned the same values,

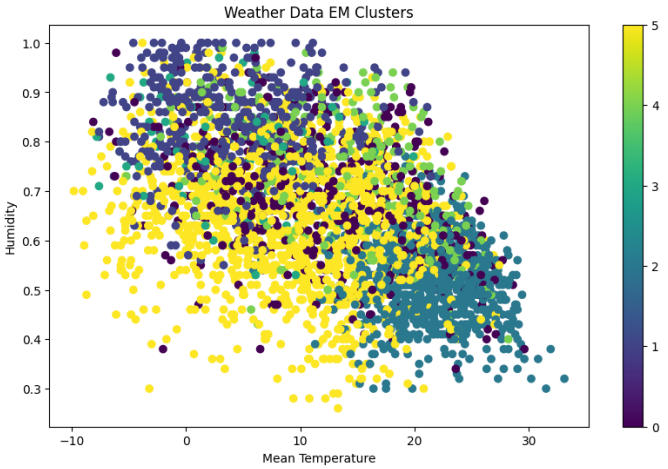

Fig. 3: Weather Data – EM Components

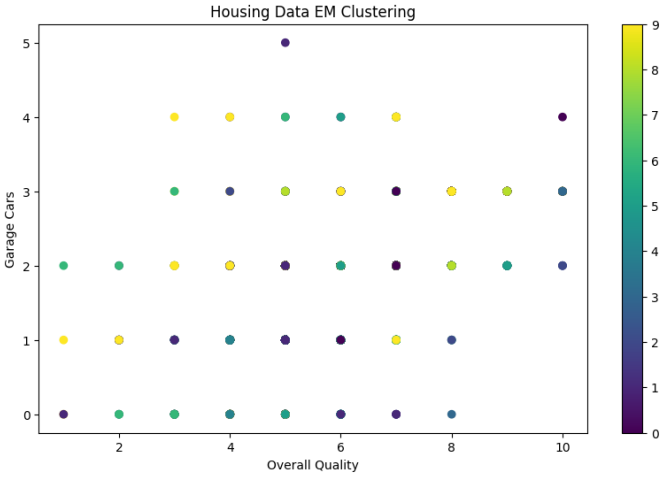

Fig. 4: Housing Data – EM Components

n=10 and n=6. In Fig.3, we can see that the weather dataset has converged to 6 parameters. In Fig.4, the housing dataset converged to 10 parameters with clustering forming a similar grid like pattern to KMeans. By the end of this step, I saw that, through different optimization methods for cluster values specialized in detecting goodness of fit and overfitting minimization, both datasets returned the same value, so the datasets were properly scaled to proceed to the next step.

III. Step 2

Dimensionality Reduction is very important for constructing unsupervised learning models for an unlabeled dataset and allows us to streamline the data into its most essential components.

A. Principal Component Analysis (PCA)

PCA is a technique where the original dataset is condensed to a set of uncorrelated components and sorted by the amount of variance they have. It aims to retain as much variance as possible in a few dimensions. Variance is important in

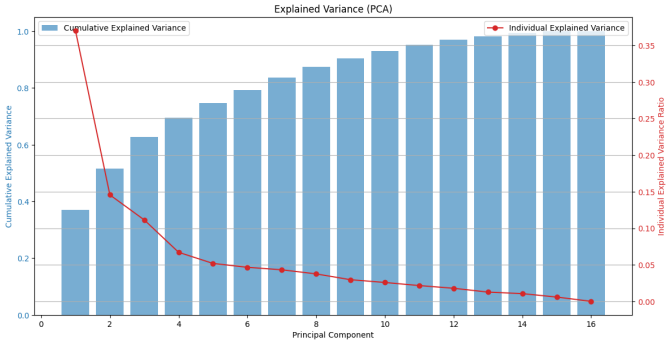

Fig. 5: Housing Data – Explained Variance (PCA)

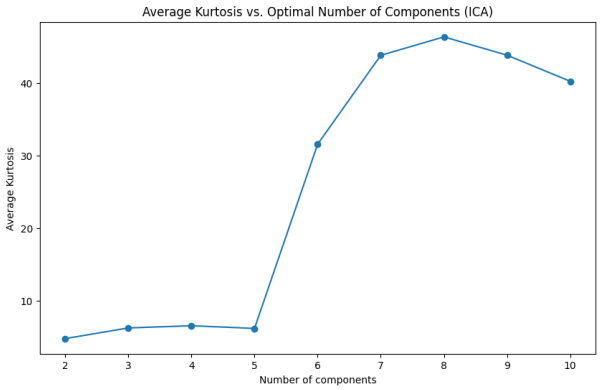

Fig. 6: Housing Data – Kurtosis (ICA)

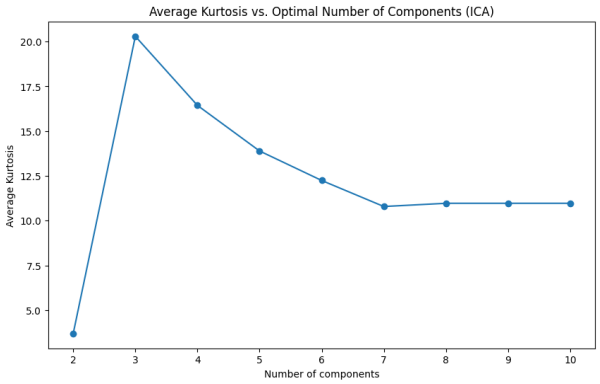

Fig. 7: Weather Data – Kurtosis (ICA)

PCA because it is the structure of the original dataset; with the dimension reduction, it becomes worthwhile if you lose too much information. To optimize to minimize the loss in variance, I used the Explained Variance method to choose several components based on when the variance exceeds a certain threshold, in this case, 95%.

In both the housing and weather datasets, the cumulative variance increased with more components, adding to the total variance and then plateaued, suggesting earlier components had more importance overall. However, I noticed a decreasing trend in the percentage of individual variance, which suggests adding more components does not add more importance. If anything, it leads to diminishing returns. Fig. 5 details the Explained Variance for the housing data.

B. Independent Component Analysis (ICA)

ICA is another alternative technique that identifies independent features—mainly used in signal processing to separate components from mixed signals. It tends to capture non-Gaussian structures in datasets, deviating from more symmetric, bell-shaped curves in Gaussian distributions. ICA should be used when you have an array of independent sources that can be easily separated. I want to use ICA to see if we can reduce the dataset to include independent components. I used the Kurtosis method to find the optimal number of independent that both the weather and housing datasets have.

Kurtosis is a measure that gives us the shape of the data distribution. About ICA, it helps us identify the features that are not normal in the data. In the graphs, I was looking for where the value of Kurtosis was the highest, meaning that the data has outliers, which makes it deviate from a bell-shaped normal distribution. Anywhere the data had lower Kurtosis, this would tell me that the data would be moving towards a normal distribution for those number of components. So, looking at Fig. 6 and Fig. 7, we see that Kurtosis was maximized when the number of components was 8 and 3, respectively, for the housing and weather datasets.

C. Random Projection (RP)

Random Projection is the third dimensionality reduction method I used that uses a random matrix to project higher dimensional data points onto a lower dimensional space. It is much faster than ICA and PCA because it can project the entire dataset to a lower dimensional area. In contrast, PCA involves calculating eigenvalues and eigenvectors and sorting them, and ICA uses more complex mathematical operations to maximize the non-Gaussianity of the dataset. RP is a straightforward operation where you multiply the matrix by some random matrix, which is less intensive computationally. I used the Reconstruction error method to optimize the number of components.

The Reconstruction Error method reconstructs the data from the lower dimensional space back to the original data using an inverse matrix. The error calculation is just the difference between the original and reconstructed data. With the weather data in Fig. 8, I was looking to see where the reconstruction error reached the minimum. However, the error was still high when I used the reconstruction error method on the housing data. The high error makes sense that the Reconstruction method performed poorly because it works better on larger datasets that need faster approximations; however, smaller datasets sacrifice accuracy for time complexity.

After looking at all three of these algorithms for dimension-

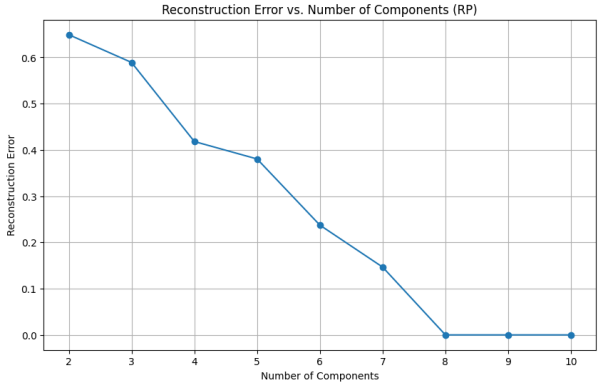

Fig. 8: Weather Data – Reconstruction Error (RP)

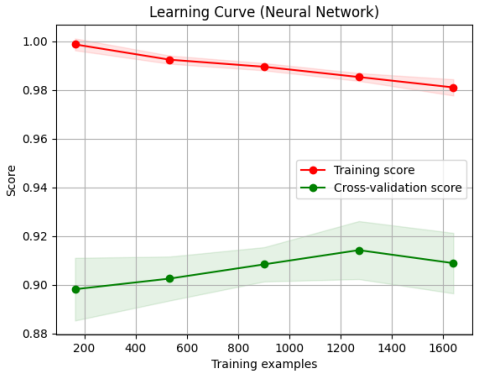

Fig. 9: Baseline Neural Network

ality reduction on my datasets, I found that some algorithms perform better than others. In particular, ICA on the housing produced many components. In the next step, I look at the algorithms' clustering performance using different criteria and find the model-reduction combination that performed the best between the two datasets.

IV. Step 3

After applying the dimensionality reduction algorithms to my two datasets, I had to consider many nuances and cases in deciding the optimal values for the number of components for each algorithm. In this section, I want to compare and rank the different algorithm and model pairs across the two datasets to see which pair had the best performance using three scoring mechanisms.

A. Silhouette Score

As discussed in Step 1, the Silhouette Method is a quite useful metric to use when trying to explore a data point's similarity with its assigned cluster and with other clusters, too. The metric is also often used to identify how well-separated the clusters are. I used the metric to assess how each model-algorithm pair handles the trade-off between compactness and cluster separation.

B. Davies-Bouldin Score

The Davies-Bouldin Score is a metric that measures the mean similarity ratio between two clusters. This metric is useful as it measures the inter-cluster scatter and intra-cluster separation, giving an insight into how well the model handled the clustering of the reduced data. To interpret the score, I am looking for lower values as they will indicate that the clusters are compact and distinct, so there will be a clear separation.

C. Calinski-Harabasz Score

The Calinski-Harbasz Score or the Variance Ratio Criterion measures the relationship between the sum of intra-cluster and inter-cluster dispersion. This metric is also very useful in ranking our pairs because it balances the compactness and separation of the clusters. A high score from this metric will indicate that the model clustered the data well.

D. Results

TABLE I: Housing Data Clustering Performance Metrics for Different Reduction Techniques

	Silhouette Score	Davies-Bouldin Score	Calinski-Harabasz Score
KMeans_PCA	0.238	1.591	994.455
EM_PCA	0.209	1.789	749.088
KMeans_ICA	0.175	1.451	324.967
EM_ICA	0.048	2.361	196.836
KMeans_RP	0.272	1.404	1230.041
EM_RP	0.174	2.018	589.094

1) Housing: Across the different reduction methods, KMeans performed much better at clustering the components just by looking at the silhouette score. Looking at the score for the housing dataset, the pair that performed the best was KMeans_RP.

TABLE II: Weather Data Clustering Performance Metrics for Different Reduction Techniques

	Silhouette Score	Davies-Bouldin Score	Calinski-Harabasz Score
KMeans_PCA	0.374	1.041	2630.676
EM_PCA	0.136	2.242	536.479
KMeans_ICA	0.386	0.843	2095.155
EM_ICA	0.205	1.495	972.762
KMeans_RP	0.446	0.842	4274.615
EM_RP	0.116	2.327	486.933

2) Weather: Across the different reduction methods, KMeans also performed the best through all metrics. Looking at the weather performance metrics chart, the pair that performed best was also KMeans_RP.

I think that assessing the performance of the different algorithms using the three metrics I mentioned: Silhouette, Davies-Bouldin, and Calinski-Harabasz scores is a much better way of analyzing how the cluster behaves when performing clustering on the reduced datasets as opposed to just showing the accuracy of the model. I understood how the algorithm handled the trade-off between the compactness and separation between the clusters by researching different methods of scoring distances between the clusters.

V. Step 4

Fig. 9 is the baseline learning curve for the NN I created in Assignment 1. We can see the accuracy peaks right under 92 percent.

TABLE III: Performance Metrics for PCA, ICA, and RP

Method	Accuracy	Training Time (s)	Prediction Time (s)	F1-Score Avg
PCA	0.9334	3.4113	0.00041	0.93
ICA	0.9471	2.6668	0.00051	0.95
RP	0.9181	2.1919	0.00032	0.92

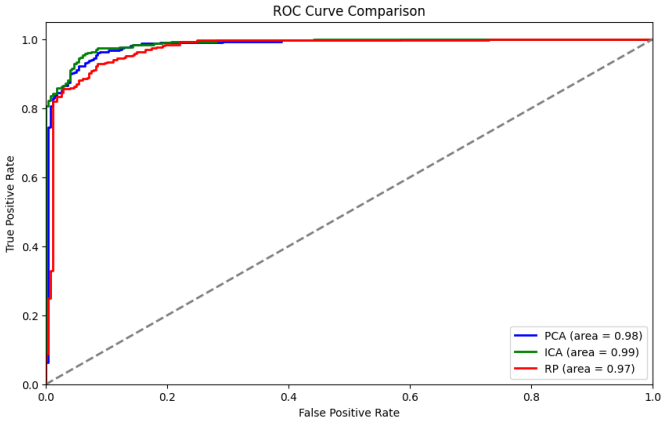

Fig. 10: ROC Curve - Neural Network on Reduced Data

Looking at the accuracy first, ICA achieved the highest accuracy, followed by PCA and RP. ICA could adequately capture the data structure for this classification task. Next, Looking at the training time, Random Projection has the fastest training time, followed by ICA and PCA. RP was more computationally efficient in projecting the data to a low dimensionality but had the lowest accuracy. The prediction time across the board was relatively fast, but RP still beat out ICA and PCA.

The ROC curve indicates that ICA had the best performance, having the most area under the curve at 0.99. This tells me there was a stellar true positive rate with few false positives. PCA and RP also performed strongly, having AUCs of 0.98 and 0.97.

, ICA stands out as the best dimensionality reduction technique for a Neural Network; it leads the pack in accuracy, F1- score, and area under the curve.

PCA performed reasonably, too, but it is the one algorithm that performs well in speed and accuracy.

RP is fastest in training and prediction, performing well and above the baseline in accuracy and F1-score.

VI. Step 5

PCA with Clusters achieved the highest accuracy, followed by ICA and RP. PCA could correctly choose meaningful principal components of the dataset. Next, Looking at the training time, Random Projection has the fastest training time, followed by ICA and PCA, similar to Step 4. It appears that RP was still more computationally efficient in projecting the data to a low dimensionality but had the lowest accuracy. The prediction time across the board was speedy; however, ICA had a much faster time than RP, which is interesting.

The ROC curve indicates that PCA had the best performance, having the most area under the curve at 0.99. This

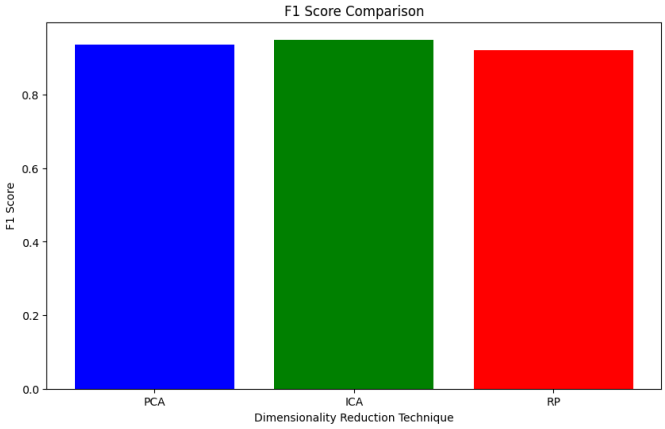

Fig. 11: F1-Score - Neural Network on Reduced Data

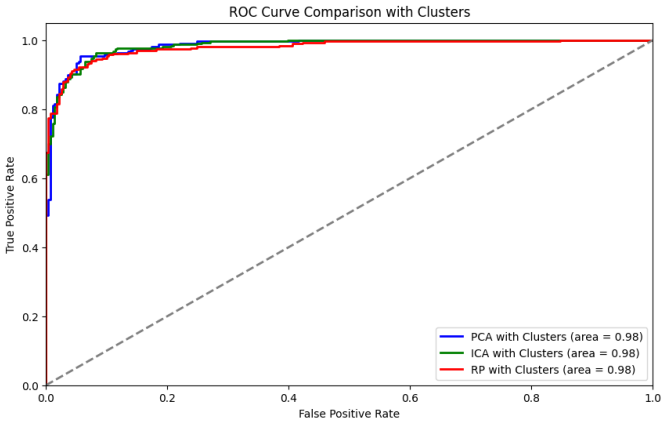

Fig. 12: ROC Curve - Neural Network on Reduced Data w/ Clusters

tells me there was a stellar true positive rate with few false positives. ICA and RP also performed strongly, having AUCs of 0.98 and 0.97.

Clearly, PCA with clusters stands out as the best dimensionality reduction technique for a Neural Network; it leads the pack in accuracy, F1- score, and area under the curve.

ICA performed reasonably, too, but it seems to be the one algorithm that performs well in speed and accuracy.

RP is fastest in training, performing well and above the baseline in accuracy and F1-score.

TABLE IV: Performance comparison of PCA, ICA, and RP with clusters.

Method	Accuracy	Training Time (s)	Prediction Time (s)	F1-Score Avg
PCA w/ Clusters	0.9454	3.6799	0.00038	0.95
ICA w/ Clusters	0.9369	3.55548	0.00029	0.94
RP w/ Clusters	0.9300	2.15453	0.00033	0.93

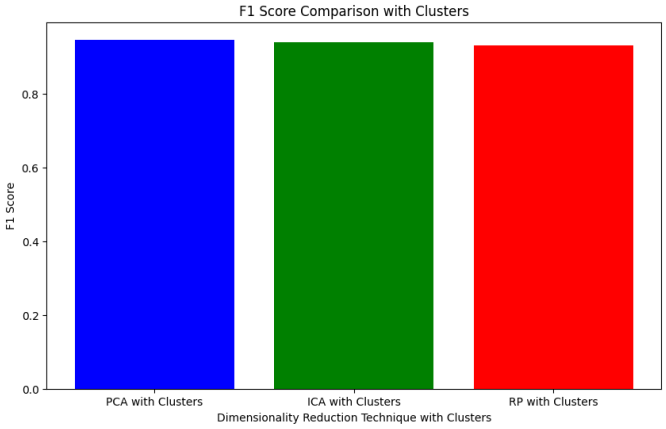

Fig. 13: F1-Score - Neural Network on Reduced Data w/ Clusters

VII. Conclusion

In this study, I extensively looked at the performance of multiple dimensionality reduction algorithms: Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Random Projection (RP). These algorithms were used on two different datasets, the ECA&D, and the Ames housing dataset. My analysis mainly focused on understanding and finding out each reduction algorithm's limitations through various metrics such as F1-score, accuracy, ROC-AUC, precision, recall, and time complexity.

In my findings, I found that applying clustering algorithms like K-Means and Expected Maximization (EM) before the dimensionality reduction method improved the Neural Network's performance considerably. I also saw that even using the components created by the dimensionality reduction algorithms performed much better than the baseline NN model.

References

[1] "A European daily high-resolution gridded meteorological data set for 1950–2021," accessed on: Jun. 9, 2024. [Online]. Available: https://zenodo.org/records/7525955

[2] "Weather Prediction Dataset," accessed on: Jun. 9, 2024. [Online]. Available: https://www.kaggle.com/datasets/thedevastator/weather-prediction