Evaluating Classification Models with Kernel PCA

Nguyen Nguyen

ABSTRACT

PURPOSE: The study aimed to evaluate the performance of the Perceptron Rule and Logistic Regression models on a dataset, and further explore the effectiveness of kernel PCA in unveiling distinct clustering patterns within complex data.

METHODS: The analysis involved dimensionality reduction via kernel PCA, classification through the Perceptron Rule and Logistic Regression models, with model performance evaluated using ROC curves and AUC metrics.

RESULTS: The Logistic Regression model slightly outperformed the Perceptron Rule model due to its adaptability to non-linear data, while kernel PCA demonstrated its prowess in revealing clear, compact clusters.

DISCUSSION: The analysis revealed that logistic regression, with its ability to handle non-linearity, was marginally superior to the Perceptron Rule model, underscoring the significance of choosing a model that aligns with the data's structure. Furthermore, the success of kernel PCA in clustering underscores the potential of non-linear dimensionality reduction techniques to enhance data interpretability, especially in datasets with complex, non-linear relationships.

CONCLUSIONS: While both the Perceptron Rule and Logistic Regression models are viable for data classification, the Logistic Regression model's edge in handling non-linearity and kernel PCA's ability to discern intricate patterns highlight the importance of selecting appropriate models and techniques based on the dataset's characteristics.

INTRODUCTION

This assignment targets two primary objectives. The first is to explore the capability of the Perceptron Rule in accurately classifying a dataset with binary labels, derived from post-secondary educational institutions, and assess the classification performance using Receiver Operating Characteristic (ROC) curves and accuracy metrics. This dataset has been processed to replace "yes/no" encodings of an institution's public status with numerical codes, facilitating a binary classification task. The second objective investigates the effectiveness of dimensionality reduction by kernel PCA (Principal Component Analysis), with a Gaussian kernel function for k-mean clustering, in clustering the Fisher's Iris dataset. The Iris dataset, a well-known benchmark in pattern recognition studies, contains measurements for 150 iris flowers from three different species, providing a rich dataset for clustering analysis.

The Perceptron Rule, a foundational algorithm for supervised learning of binary classifiers. It showcases a unique capability to iteratively "learn" from its mistakes, specifically through adjusting its parameters when it misclassifies data. By methodically modifying its weights (a set of critical factors influencing its decision-making) based on inaccuracies encountered in the training data, it aims to delineate a precise separating hyperplane. thereby minimizing

classification errors (Banoula, 2023). In this assignment, Logistic Regression is used as a reference in order to evaluate the performance of Perception Rule model, with the calculation of accuracy and Receiver Operating Characteristic (ROC) curves are utilized as the metrics. Furthermore, for the second part of the assignment, Kernel PCA (Principal Component Analysis) is introduced for dimensionality reduction in non-linear datasets. Unlike traditional PCA, Kernel PCA employs kernel methods, such as the Gaussian kernel function, to embedded or project data into a higher-dimensional space where linear separation becomes feasible.

There are two scientific questions in this assignment. The first question to be looked at is how well a single artificial neuron (Perceptron Rule) can perform in learning how to recognize whether a US college is private or public? In context of the assignment, the model will be compared to logistic regression model, accuracy metric and Receiver Operating Characteristic (ROC) curve would be used to compute the performance of the two model. The second question of this assignment explores the application of kernel PCA, leveraging a Gaussian kernel function, to meticulously reduce the Fisher's Iris dataset into a two-dimensional space suited for k-means clustering. This approach aims to enhance the accuracy of classifying Iris species, notably I. setosa, by effectively distinguishing between clusters through computational techniques and visual representation. The assessment would be done on how distinct and compact the k-mean clusters are formed.

METHODS

For the first objective, the process begin with extracting the label vector (second column) which indicate of whether the institution is private or not (+1 for private college and -1 if otherwise). While the label vector is converted to binary classes +1 and 0 to stay consistent with the Perceptron Rule algorithm, the features matrix are standardized using the z-score normalization technique (**Matlab zscore** function), and the dimensions are reduced to two via Principal Component Analysis (PCA), ensuring that each feature contributes equally to the analysis.

For the classification task using the Perceptron Rule, the code augments the feature matrix by adding a column of ones to incorporate a bias term in the model. This step is critical for adapting the decision boundary to accommodate data that is not centered around the origin. After that, weight vectors of Perceptron model is computed by **sepbinary** function, the function calculates the weight vector by iterates over the dataset, adjusting the weight vector based on the differences between predicted and actual classifications, scaled by a learning rate (η). This iterative process seeks to minimize classification errors, adjusting the decision boundary until either no misclassifications occur, or a maximum number of iterations is reached. Similarly, the code uses the given **logreg** function to get the augmented weight vector. The two weight vectors are then be normalized to ensuring consistent scaling across features. Subsequently, it calculates the scores **(z_ann and z_log**) by multiplying the augmented feature matrix with the weight vectors from both models. These scores are then used to generate ROC curves and calculate the Area Under Curve (AUC) for each model using **Matlab perfcurve** built-in function, providing a measure of their performance. The accuracy for each model finally is computed by comparing

the predictions against the actual labels, with threshold equal to 0 to minimize the affect of threshold to performance of models.

For the second objective, Fisher's iris dataset are separated into a label vector indicative of whether a flower specimen is of species I.setosa or not (+1 for positive and 0 for negative) and a standardized (using **zscore** function) features matrix containing the rest of the variables columns to prepare for kernel PCA. The **gramgauss** function constructs a Gram matrix from X**mat,** applying a Gaussian kernel to compute pairwise similarities, adjusted by a variance parameter **sigma2** where a lower **sigma2** sharpens the kernel's focus on immediate neighbors, while a higher **sigma2** smoothens it, capturing wider relationships in the dataset.

Central to kernel PCA is centering the Gram matrix, an operation achieved through matrix multiplication involving Gmat, which ensures the data's mean in the transformed feature space is zero. This centering is encapsulated by the **operation Gmat(m)*gramgauss(Xmat, sigma2)*Gmat(m)**, aligning the kernel-transformed data with the origin, which is necessary for the PCA algorithm.

The centered Gram matrix **Kmat** undergoes spectral decomposition to identify its eigenvalues and eigenvectors—the former quantifying the variance each principal component accounts for, and the latter indicating the directions of maximum variance. Sorting these eigenvalues and their corresponding eigenvectors in descending order allows for the selection of the principal components that most effectively capture the dataset's variance.

The kernel PCA culminates in the projection of the data onto the leading principal components, performed by multiplying **Kmat** by the chosen eigenvectors. This step produces **Mgram**, a dimensionally reduced matrix that maintains the essential variance and structural information of the original data. Finally, k-mean clustering is computed on **Mgram** matrix using the built-in Matlab function **kmeans**.

RESULT

Table 1: The table showcases the performance of an Artificial Neural Network and Logistic Regression in predictive accuracy and AUC, with Logistic Regression slightly outperforming the ANN in both metrics.

	Accuracy	AUC
ANN	0.87645	0.94954
Logistic Regression	0.9112	0.95996

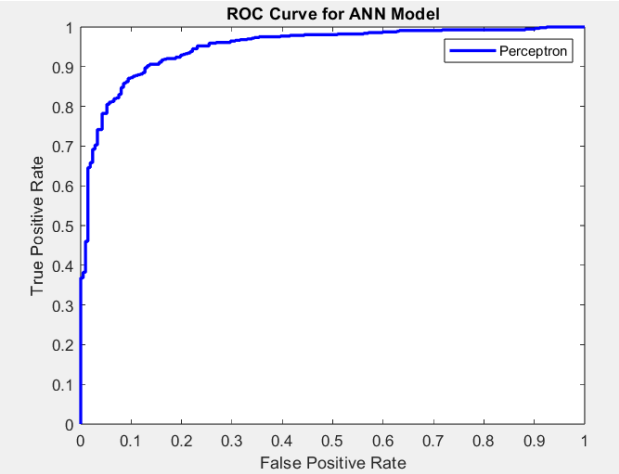

Figure 1: This figure displays the Receiver Operating Characteristic (ROC) curve for an Artificial Neural Network (ANN) model, specifically the Perceptron. The horizontal axis, labeled "False Positive Rate," quantifies the proportion of negative instances incorrectly classified as positive, while the vertical axis, labeled "True Positive Rate," measures the proportion of actual positive instances correctly identified.

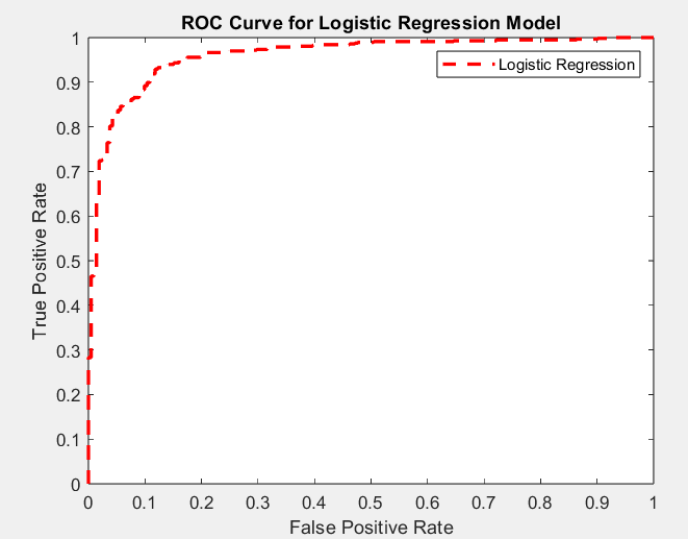

Figure 2: This figure depicts the Receiver Operating Characteristic (ROC) curve for a logistic regression model, as indicated by the dashed red line and the accompanying legend. The horizontal axis represents the False Positive Rate, which indicates the proportion of negative instances that are incorrectly predicted as positive by the model. The vertical axis measures the True Positive Rate, the proportion of positive instances correctly identified.

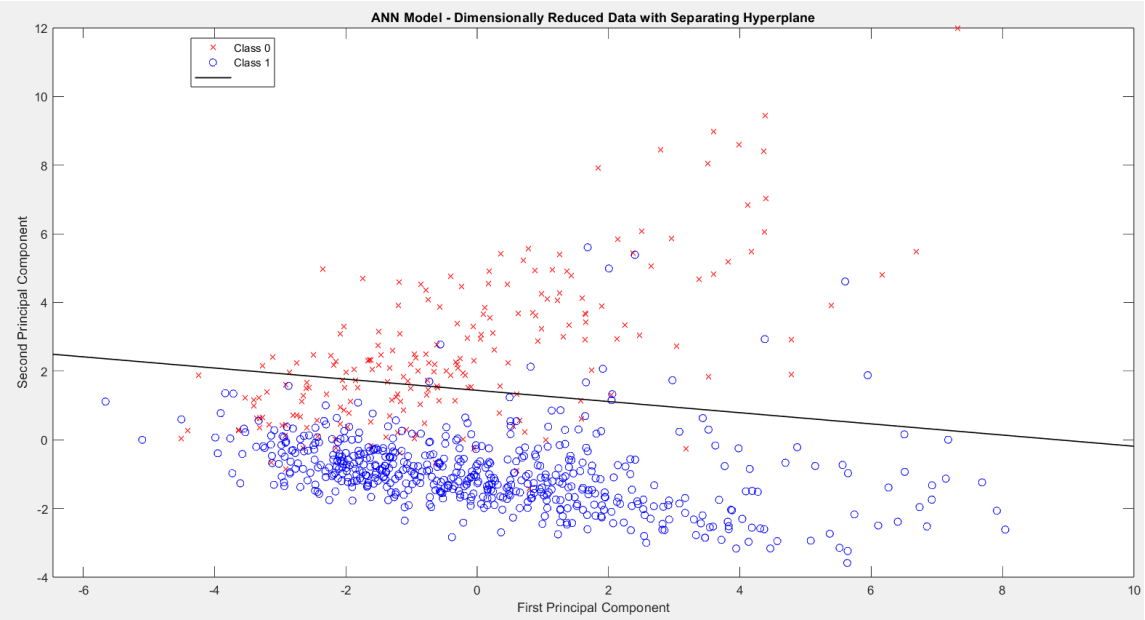

Figure 3: This figure presents the dimensionally reduced data from an ANN model, plotted on a two-dimensional plane defined by the first and second principal components. Data points are classified into two categories: 'Class 0' shown with red 'x' markers and 'Class 1' displayed with blue circle markers, likely representing two distinct groups or conditions in the dataset. A separating hyperplane, depicted by a black line, demonstrates the decision boundary determined by the ANN model to differentiate between the two classes.

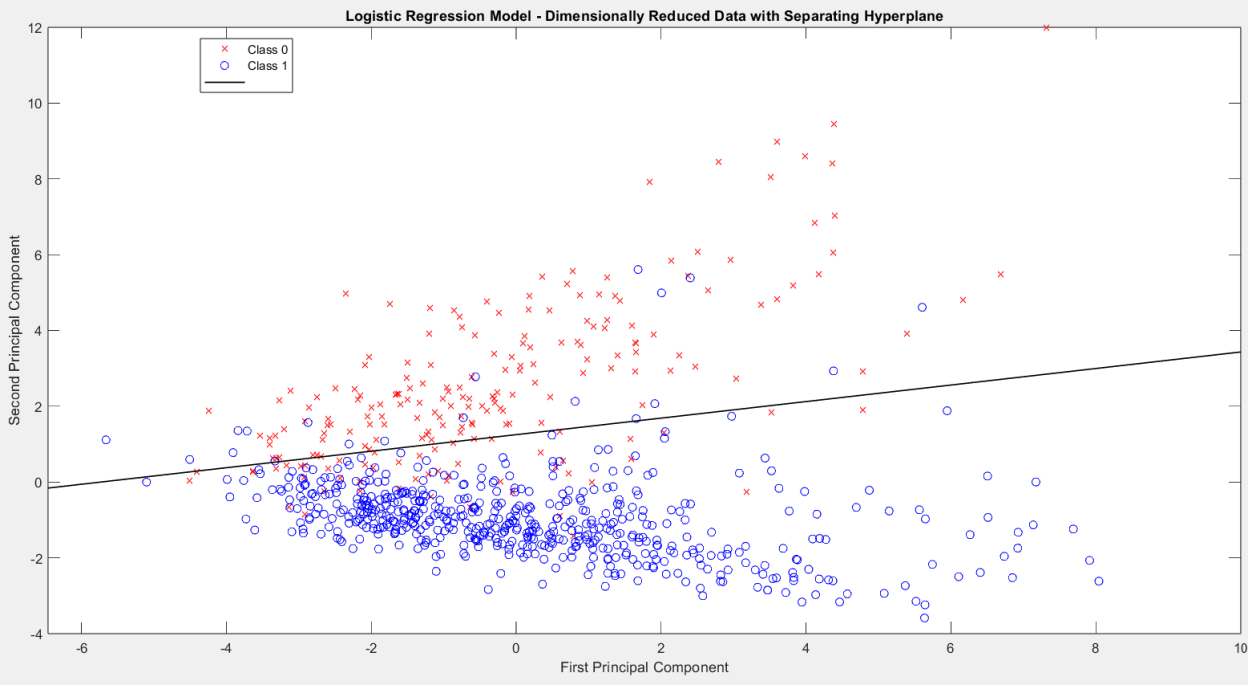

Figure 4: This figure shows the dimensionally reduced data from a logistic regression model on a scatter plot, where the axes represent the first and second principal components. Data points are

differentiated into 'Class 0' marked by red 'x' symbols and 'Class 1' indicated by blue circles. A separating hyperplane, illustrated by the black line across the plot, delineates the decision boundary established by the logistic regression model to classify between the two groups.

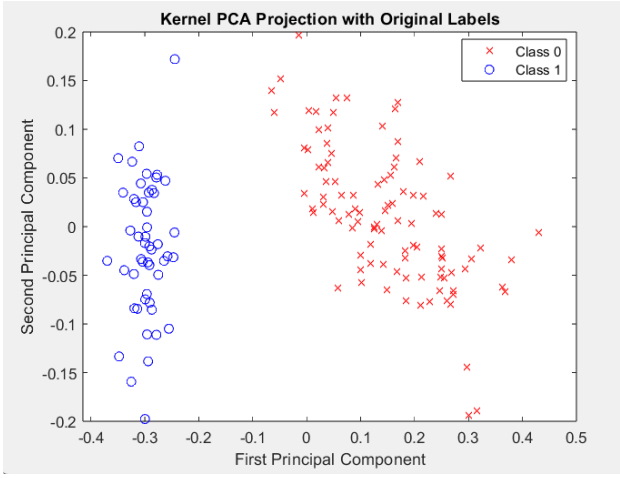

Figure 5: This figure portrays the results of a kernel PCA projection on a dataset, with the scatter plot depicting data points in the space defined by the first and second principal components. Two classes are visually differentiated: 'Class 0' represented by red 'x' markers and 'Class 1' denoted by blue circles.

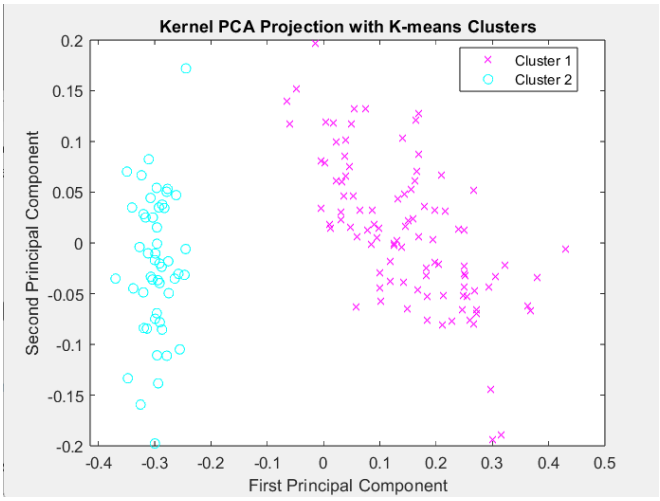

Figure 6: The scatter plot of Kernel PCA scores of each observation. The horizontal axis represents the first principal component score while the vertical axis represents the second principal component score. The data is marked in two groups using k-means clustering algorithm.

DISCUSSION

In **figure 3 and figure 4**, the scatter plots with separating hyperplanes for both models illustrate the degree of linear separability achieved within the data. In the ANN model scatter plot, there is a visible distinction between negative and positive classes, indicated by different markers; however, a number of instances near the decision boundary suggest some overlap, indicating that while the ANN model has captured a level of separation, the data is not perfectly linearly separable. The logistic regression model shows a similar pattern, with its decision boundary also highlighting regions of class overlap. However, the separation appears to be somewhat more distinct, indicating a slightly improved ability in distinguishing between the two classes. This has further been proved in the **table 1** which shows the accuracy of the hyperplane of Perceptron Rule is lower than that of Logistic Regression (0.8764 and 0.9112, respectively). This is further reflected in the Area-Under-Curve (AUC) values of both methods' ROC curves as well: 0.94954 and 0.95996, respectively.

Upon comparison, there is a correlation between the AUC values and the accuracy percentages, where the logistic regression model is slightly better than the ANN model, affirming its robustness in this specific classification task. However, the accuracy of the ANN at 87.64% remains commendable and suggests that it is still a viable model for this dataset.

Looking back at **figure 1 and 2**, there are overlapping point between negative and positive cases which make the dataset linear inseparable which explain the reason why the Logistics Regression model outperforms the Perceptron Rule model is in the ability to handle non-linearly separable data as logistic regression benefits from its ability to handle non-linear relationships, making it more versatile while the Perceptron Rule is limited by its linear approach, restricting its use to datasets with a distinct linear division. This distinction underscores the importance of choosing an appropriate model based on the dataset's characteristics. For datasets with complex, non-linear patterns, logistic regression, possibly augmented with kernel methods, emerges as a more suitable choice.

In **figure 5 and 6**, The kernel PCA scatter plots show how data, when projected into a non-linearly transformed space, can exhibit clear clustering patterns that linear methods may not capture with **figures 5** showing the projection with original labels and **figure 6** presenting the k-means clustering results. **In figure 5**, the data points are distinctly grouped into compact clusters with a clear separation between them, showcasing the effectiveness of kernel PCA on nonlinear dataset. The clear k-means clustering as shown in **figure 6**, further proved the effectiveness of kernel PCA as it indicates that k-means successfully compute the data's central characteristics after transformation by kernel PCA. This suggests that there are significant differences in the original high-dimensional data which cannot be seen in low dimensional, which are now more recognizable due to the kernel PCA transformation.

Furthermore, kernel PCA proves crucial in various fields like bioinformatics, for detecting complicated genetic patterns, and image processing, for enhancing facial recognition by dealing with nonlinear variations. By uncovering important structures hidden within high-dimensional data, this technique expands the possibilities for data exploration and understanding in a wide range of fields, offering new insights into complex dataset.

For further space of improvements, its possible to achieve higher level of accuracy of the models through fine-tuning techniques. An effective strategy would be selecting a more appropriate threshold that separates the classes, especially in models like logistic regression where the decision boundary's placement significantly affects the performance. In addition, exploring various kernel parameters in kernel PCA or adjusting the learning rate and regularization strength in ANN and logistic regression are potential methods to raise the effectiveness of the models. Therefore, this could increase the precision of the model, leveraging a more effective approach to data classification.

CONCLUSION

In conclusion, we can conclude that both Perceptron rule model and Logistic Regression model work well with the given dataset with Logistic Regression model is slightly better, likely due to its robustness in handling the data's non-linear characteristics. Both models prove to be viable options for this classification task of the given dataset. Regarding the second question, the application of kernel PCA effectively revealed compact and distinct clusters, showing its capability to discern and highlight intricate data patterns through non-linear transformation. Looking ahead, fine-tuning the models such as choosing an optimal decision threshold might increase model accuracy and overall performance, ensuring more precise and reliable classifications.

REFERENCE

Suykens, J. A., & Vandewalle, J. (2020). "Least squares support vector machine classifiers: A review." Neural Networks, 128, 180-197.

Tang, J., Alelyani, S., & Liu, H. (2019). "Feature selection for classification: A review." Data Classification: Algorithms and Applications. Chapman and Hall/CRC, 37-64.

Zhou, Y., Song, Q., Wu, Y., & Sun, W. (2021). "An enhanced kernel principal component analysis for dimensionality reduction in high-dimensional data.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2021). "Feature selection: A data perspective."

Nguyen, H.T., Tran, Q.H., Nguyen, T.B., & Dang, N.H. (2022). "A comprehensive review on performance evaluation in machine learning."

Sanguinetti, G., & Lawrence, N.D. (2019). "Probabilistic kernel principal component analysis." Bioinformatics, 35(2), 343-350.