Artificial Neuronal learning VS Logistic Regression learning and Kernel PCA

ABSTRACT

PURPOSE: To determine how well a single artificial neuron can perform in learning how to recognize whether a US college is private or public, relative to a logistic regression model, and how well we can classify Fisher's Iris data as the species I. Setosa by using a kernel PCA.

METHODS: Use of the Perceptron rule for machine learning to evaluate its efficacy as a classification algorithm relative to a logistic regression model, and the effectiveness of Kernel PCA in classifying Fisher's Iris data as the species I. Setosa and others, evaluated using its K-means counterpart.

RESULTS: The results clearly demonstrate the slightly greater effectiveness of the logistic regression model as a classification algorithm, relative to the Perceptron model as shown in Figures 1, 2, 3, and 4, through the evaluatory metric values shown in Table 1. They also demonstrate the efficacy of Kernel PCA in classifying Fisher's Iris data as the species I. Setosa and others, through Figures 5 and 6.

CONCLUSIONS: The Perceptron model, relative to the logistic regression model, is 3.97% worse at learning how to recognize whether a US college is private or public, while the data in Fisher's Iris dataset can perfectly classify the species I. Setosa, a claim which is supported by its K-means result of the dimensionality reduction.

Word Count: 210

INTRODUCTION

The purpose of this assignment is to determine how well a single artificial neuron can perform in learning how to recognize whether a US college is private or public, relative to a logistic regression model, and how well we can classify Fisher's Iris data as the species I. Setosa by using a kernel PCA.

The Perceptron rule is a simple machine learning algorithm for supervised learning used for binary classifications where the Perceptron learns a linear decision boundary to separate input observations for features into binary forms. It trains by updating estimates of the augmented vector for the hyperplane through vectorization and converging on the optimal weights. Logistic regression models are also supervised learning models used for binary classification. It trains by using a training set of labels, and weights are calculated through maximum likelihood estimation, which can handle non-linear boundaries. Kernel PCA is a dimensionality reduction algorithm which computes the principal components of a dataset by using a kernel function in a higher-dimensional feature space, which allows it to identify non-linear relationships between features and produce greater separation.

The questions being explored are how well can a single artificial neuron perform in learning how to recognize whether a US college is private or public, and how well can we classify Fisher's Iris data as the species I. setosa using Kernel PCA.

METHODS

This exploration aims to answer two interesting conceptual questions through which we evaluate the efficacy of common classification techniques. The first concerns dataset of colleges in the US and the second is the ubiquitously used Fisher Iris dataset of lengths and widths of the sepals of different species of flowers.

The first conceptual problem asks how well a single artificial neuron can perform in learning how to recognize whether a US college is private or public, relative to a logistic regression model, where we implement the Perceptron rule in a classification algorithm to produce a hyperplane separating the binary classification which is also computed by the same algorithm, and then compare its accuracy to the accuracy of the hyperplane produced by a logistic regression model which is fed the same dataset.

The second conceptual problem asks how well we can classify Fisher's Iris data as the species I. Setosa, and we evaluate this by using a kernel PCA method to reduce the data to 2D and then contrasting the results of that with a K-means clustering of the indices of the same data.

The parent function a5_20292366 is the executory function where all the data is first initialized for exploration for both collegenum and fisheriris.

Both datasets are standardized. This is particularly interesting in the case of training a perceptron since it ensures all features have the same scale and are effectively comparable, and thus separable. Doing so also minimizes the impact that possible outliers may have on the training of the Perceptron since they may skew the weights which are essential to the Perceptron's ability to learn. Standardization in the case of our datasets may help decrease the cost of training computation for the Perceptron since our sepbinary function depends on a convergence criterion to exit the loop which helps the Perceptron learn by updating estimates of the augmented vector for the hyperplane through vectorization, and standardization reduces variation across training data, making the process of converging on the optimal weights faster.

Similarly, standardization may reduce computational costs while doing kernel PCA for the second part of our code, since centering the input data around a zero mean reduces the complexity of calculations involved in constructing a Gram kernel matrix, while also making it easier to find the principal components since weights are then comparable and more easily interpreted. Standardization of input data may also help our PCA more accurate since the bias towards features with relatively larger scales is minimized.

Part (a) of the code which deals with the functions a5q1 and sepbinary seeks to answer the first conceptual problem described above. a5q1 computes the accuracy of the artificial neuron and that of the logistic regression, while also computing the thresholds at which both are maximally accurate and plotting the hyperplane-separated classification plots produced by both the Perceptron and the logistic regression model. sepbinary implements the Perceptron rule to update the current estimate of the augmented vector for the hyperplane by iterating augmented vector 10000 times and updating weights till optimal weights are achieved, which is a convolutional variation of the vectorized form of Perceptron gradient descent using a convergence criterion.

Part (b) of the code which comprises the functions a5q2 and gramgauss seeks to answer the second conceptual problem described above. a5q2 computes the kernel PCA using the gramgauss function wherein a Gram matrix for the data present in Xmat is produced using the built-in pdist2 function which calculates the squared Euclidean distance matrix and uses that to calculate the Gaussian kernel matrix. a5q2 then finds and sorts the spectral decomposition of the Gram matrix calculated previously and uses its principal component's vectors to project the data in Xmat to 2D. It then produces a K-means clustering to compare against the kernel PCA reduced data.

RESULTS

Table 1 contains the summary of the accuracies of both techniques and their respective 'Area Under Curve' (AUC) statistics derived from their specific Receiver Operating Characteristic (ROC) curve.

Figure 1 shows the ROC curve for the Perceptron's performance as a classification model at all possible thresholds, the AUC for which is shown in Table 1.

Figure 2 shows the ROC curve for the Logistic Regression Model's performance as a classification model at all possible thresholds, the AUC for which is shown in Table 1.

Figure 3 shows the PCA reduced scatterplot of colleges separated by a hyperplane produced by the Perceptron, the accuracy for which is shown in Table 1.

Figure 4 shows the PCA reduced scatterplot of colleges separated by a hyperplane produced by the Logistic Regression Model, the accuracy for which is shown in Table 1.

Figure 5 shows Fisher's Iris data processed using kernel PCA where I. setosa is indicated as blue circles and the other two species are indicated as red crosses.

Figure 6 shows Fisher's Iris data processed using kernel PCA clustered using K-means where I. setosa is indicated as cyan circles and the other two species are indicated as magenta crosses.

Table 1: Summary of accuracies and AUC statistics derived from the ROC curves of a single artificial neuron (a Perceptron) and a Logistic Regression model used to create binary classifications for US colleges.

	Accuracy	**AUC**
Perceptron	{[0.8764]}	{[0.9495]}
Logistic Reg.	{[0.9112]}	{[0.9600]}

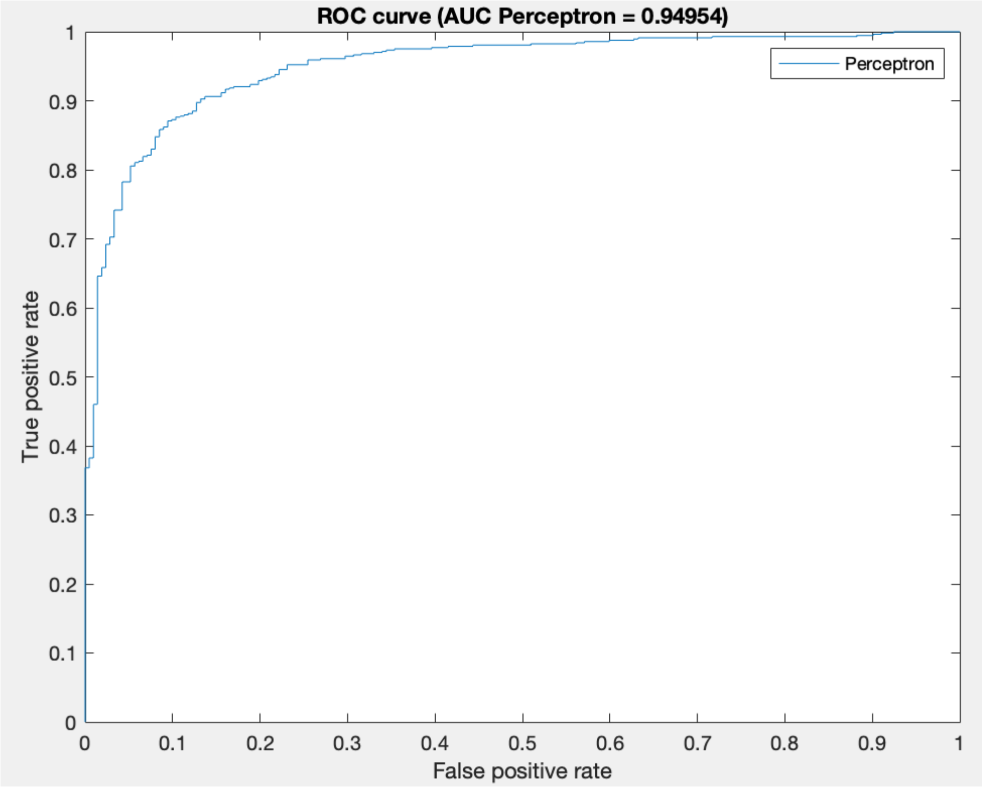

Figure 1: ROC curve for the Perceptron's performance as a classification model at all possible thresholds, the AUC for which is shown in Table 1.

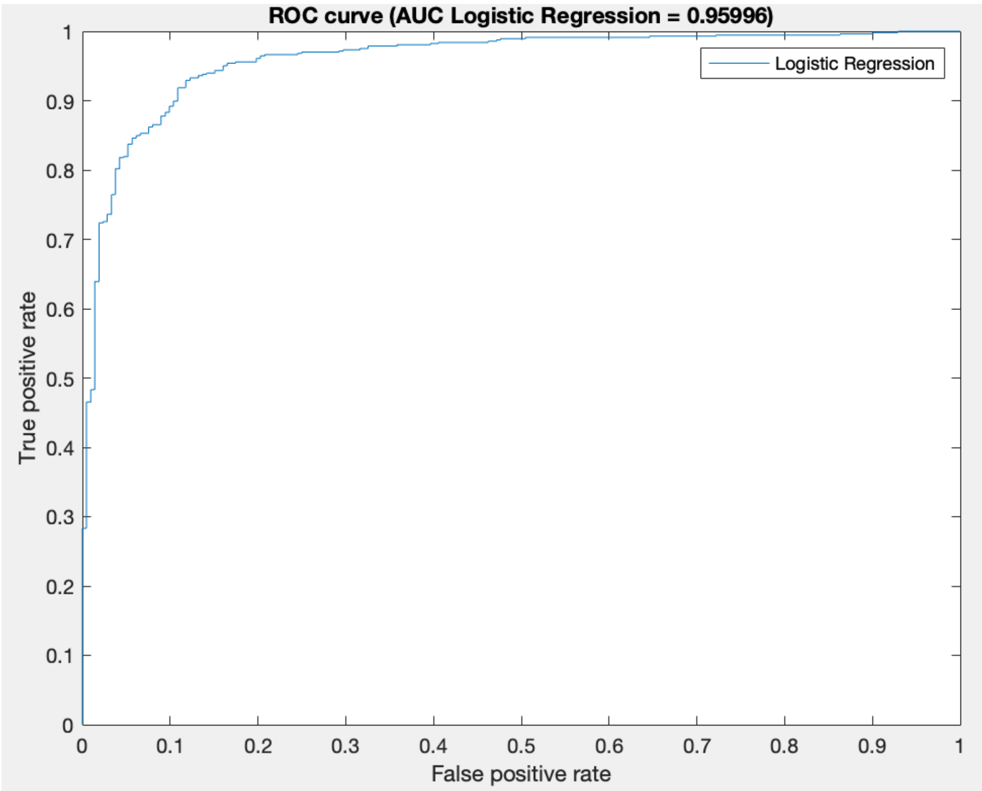

Figure 2: ROC curve for the Logistic Regression Model's performance as a classification model at all possible thresholds, the AUC for which is shown in Table 1.

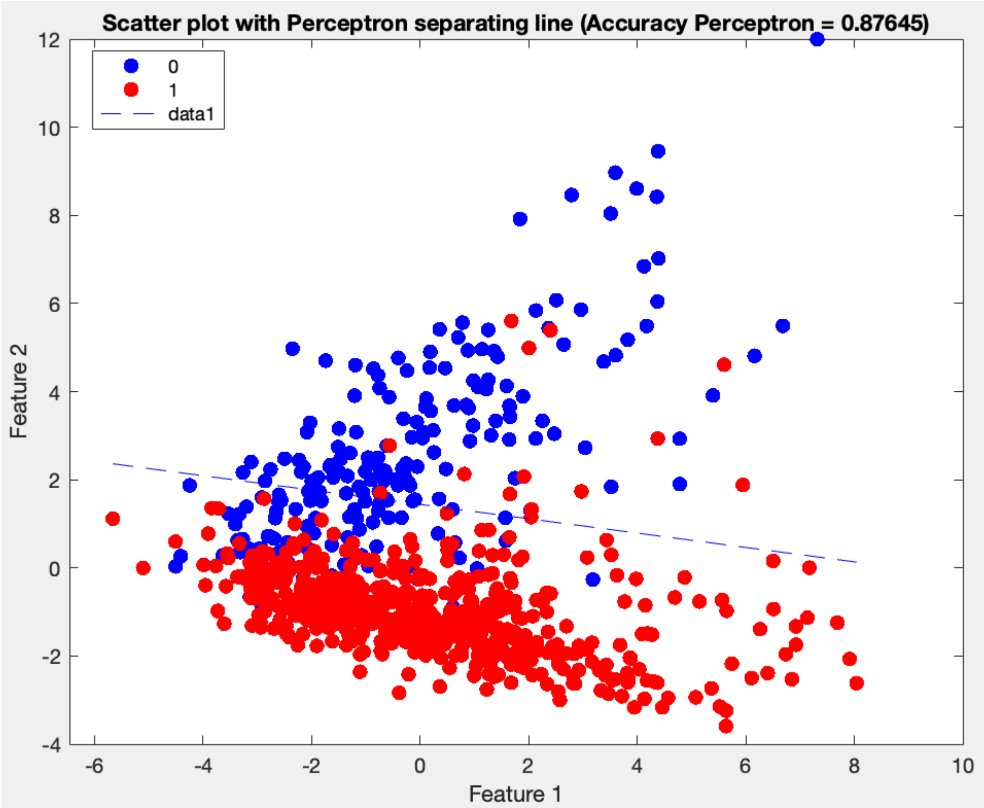

Figure 3: PCA reduced scatterplot of colleges separated by a hyperplane produced by the Perceptron, the accuracy for which is shown in Table 1.

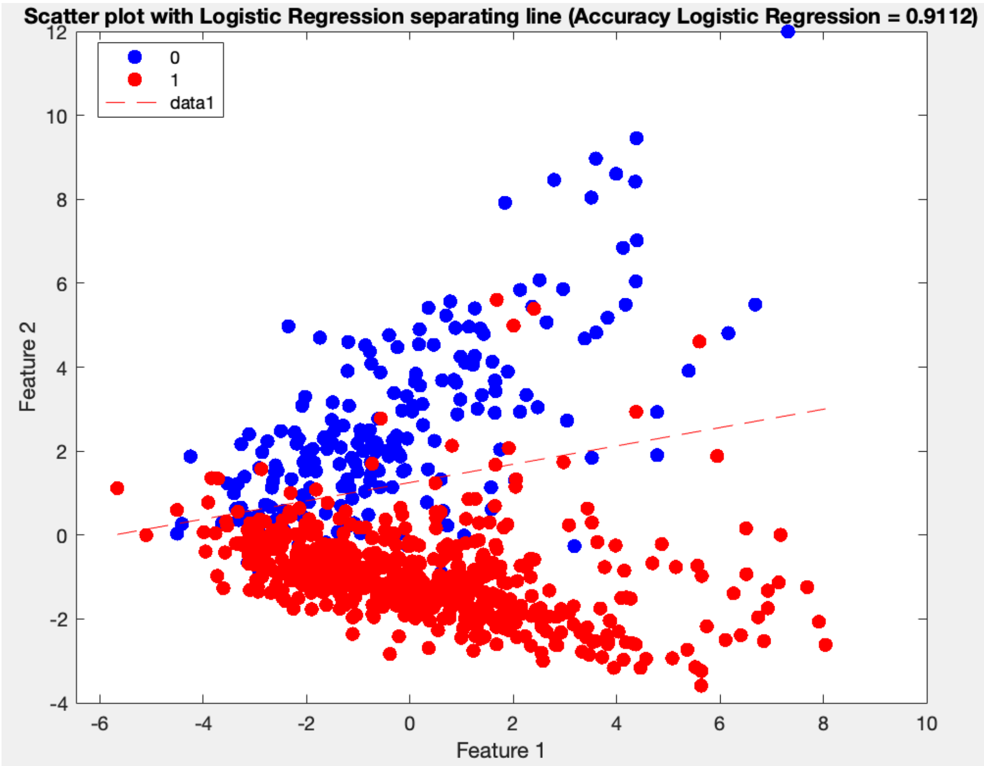

Figure 4: PCA reduced scatterplot of colleges separated by a hyperplane produced by the Logistic Regression Model, the accuracy for which is shown in Table 1.

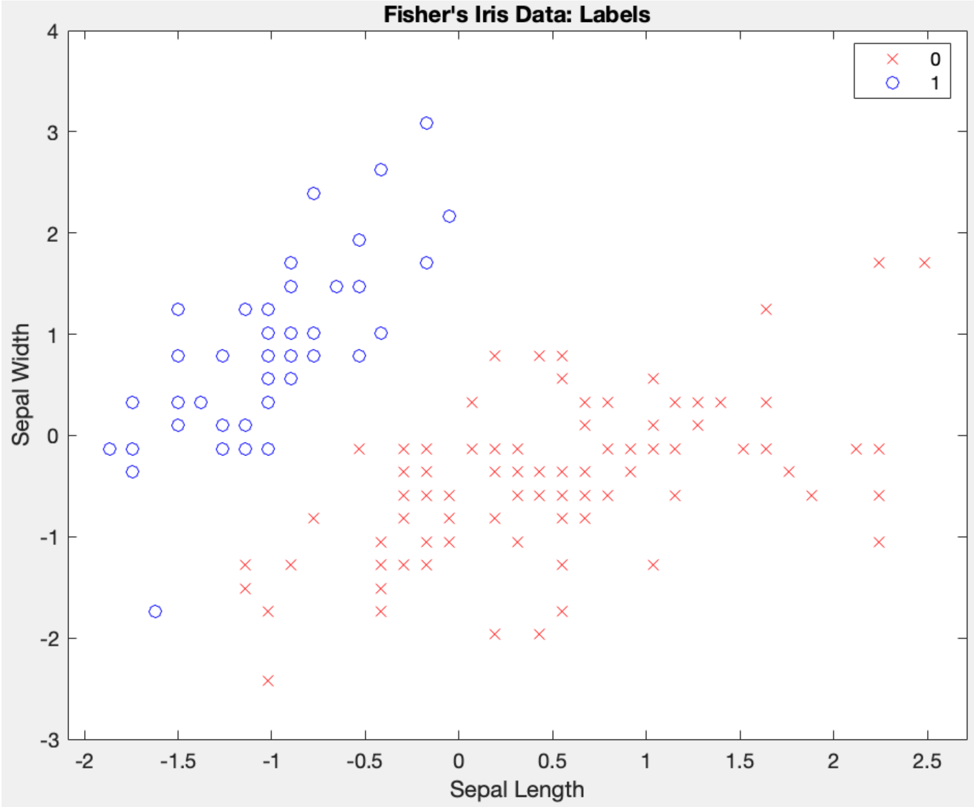

Figure 5: Fisher's Iris data processed using kernel PCA where I. setosa is indicated as blue circles and the other two species are indicated as red crosses.

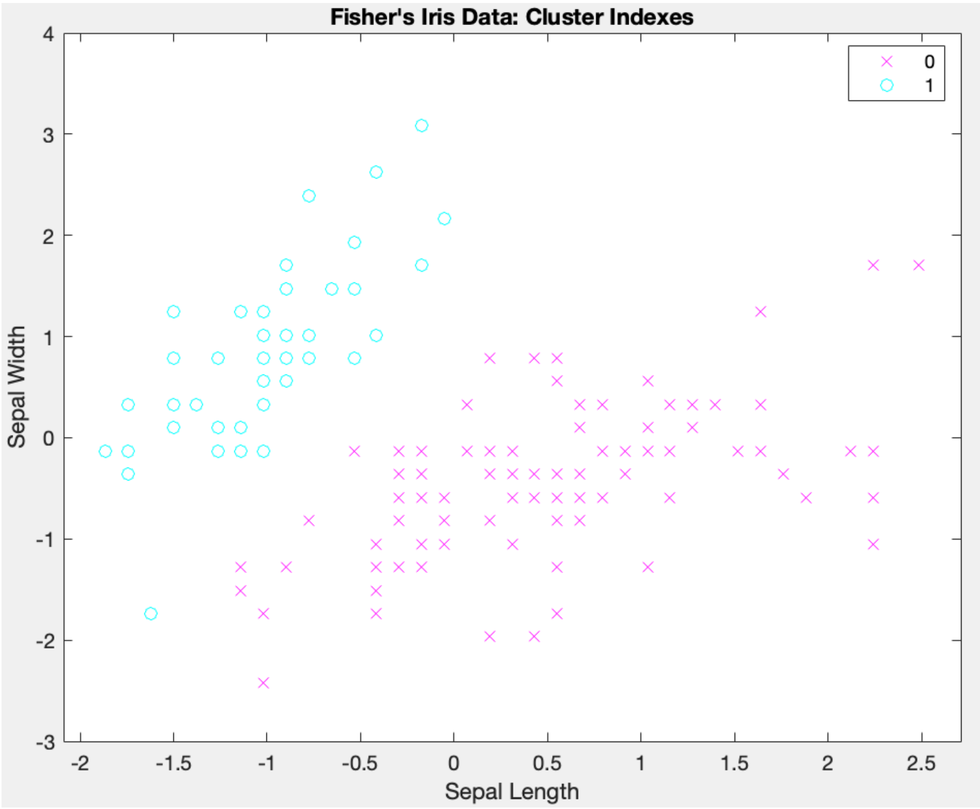

Figure 6: Fisher's Iris data processed using kernel PCA, clustered using K-means, where I. setosa is indicated as cyan circles and the other two species are indicated as magenta crosses.

DISCUSSION

Figures 1, 2, 3 and 4 are plots that shed light upon the first conceptual problem of how well a single artificial neuron can perform in learning how to recognize whether a US college is private or public, relative to a logistic regression model.

Figures 1 and 2 show the ROC curves for both models which are virtually identical. Both models have an AUC value much higher than a random guessing classification of 0.5 and rather close to a perfect classifier model with an AUC value of 1.0 where all data points would lie in the true positive and true negative sections of a confusion matrix. This metric would evaluate both the Perceptron and logistic regression models to have impressive classification performances.

The AUC value for the Perceptron, which is 0.9495, however, is lower than that of the logistic regression model, which is 0.9600, which may be due to a couple of reasons. Theoretically, logistic regression models may be better at handling datasets with a variable number of samples for each feature being processed which may lead to a more accurate setting of the weights for the learning algorithm since it treats all features differently based on their samples, whereas a Perceptron model would treat all features the same regardless of varying numbers of observations. This doesn't seem to be the case for our dataset so this may not be a contributing factor to a better AUC value for the logistic regression model.

Perceptron models use a vectorized form of gradient descent which depends on a convergence criterion in large iteration to calculate optimal weights, whereas most logistic regression models use maximum likelihood estimation which is more optimized and lower cost in terms of complexity and fits the data better than gradient descent models which tend to overfit or underfit depending upon variation in datasets. This may be a contributing factor to the higher AUC value for the logistic regression model, and the higher accuracy of 0.9112 relative to the Perceptron model's accuracy of 0.8764.

Though it doesn't seem to be the case for our data, logistic regression models are generally able to demonstrate non-linear relationships between features, compared to Perceptron models that can only learn linear separations, which may increase the AUC value for the logistic regression model in other cases.

Figures 3 and 4 which pictorially demonstrate the effectiveness of both these learning models as classification algorithms seem to back up all the discussion regarding the efficacy of our logistic regression model being higher than that of the Perceptron model since the hyperplane produced by the logistic regression clearly shows a better separation of the binary classified datapoints in red and blue in Figure 4, relative to the hyperplane produced by the Perceptron model which doesn't seem to separate the binary classified data as neatly as the other model, which again is reflected in its relatively lower accuracy of 0.8764, which is almost 3.97% worse than the logistic regression model's accuracy of 0.9112.

The optimal thresholds for both models seem to follow suit in indicating the slight superiority of the logistic regression model in the case of our dataset. Optimal threshold, in binary classification, is the threshold above which a classification algorithm classifies an observation in a dataset as positive in the most accurate manner possible. A relatively lower optimal threshold is more likely to classify an observation as positive even if the model is uncertain which results in lower precision where a higher proportion of predicted positives may be false, while also resulting in a higher proportion of true positives being true. The Perceptron model has a relatively lower optimal threshold of -6.6702 compared to the logistic regression model's threshold of -5.6617, which supports its relatively higher accuracy.

While the above argument is true, it may also be true in some cases that higher optimal thresholds may lead to a lower proportion of the true positives being correctly identified leading to lower accuracy, though this doesn't seem to be of significant involvement in our results.

Considering all points of the discussion above, we can answer the first conceptual problem by stating that the Perceptron model, relative to the logistic regression model, is 3.97% worse at learning how to recognize whether a US college is private or public, which for our results and dataset seems insignificant, but may prove to be significantly less effective for much larger ones.

Figures 5 and 6 are plots that shed light upon the second conceptual problem of how well we can classify Fisher's Iris data as the species I. Setosa by using a kernel PCA method to reduce the data to 2D and then contrasting the results of that with a K-means clustering of the indices of the same data.

Kernel PCA is a dimensionality reduction algorithm which computes the principal components of a dataset by using a kernel function in a higher-dimensional feature space, which allows it to identify non-linear relationships between features and produce greater separation. Though it is computationally expensive due to high-dimensional analysis, it is more accurate in datasets with non-linearly related features, which is the case for our Fisher's Iris dataset as regular PCA is demonstrably inaccurate at producing suitable separation when compared to its K-means counterpart which shows the natural, intuitive clustering that the human eye can distinguish instantly.

Since the classifications produced in Figures 5 and 6 are identical, we can assume that dimensionality reduction after a higher-dimensional analysis for non-linear relationships may be useful for this dataset, since it matches the clustering that the unsupervised learning K-means algorithm produced, where I. Setosa in blue and cyan circles are clearly accurately separated from the rest of the species. This higher-dimensional analysis seems invaluable since dimensionality reduction from four features to two would seem to only capture obvious linear relationships and not non-linear ones invisible at even 4D, which only speaks to the ability of the kernel function to mathematically decipher the underlying structure of the data.

Considering the points in relevance to the discussion regarding the second conceptual problem of how well we can classify Fisher's Iris data, we can say that the data in Fisher's Iris dataset can be perfectly classified as the species I. Setosa, and others, a claim which is supported by the K-means result of the dimensionality reduction.

In conclusion, the Perceptron model, relative to the logistic regression model, is 3.97% worse at learning how to recognize whether a US college is private or public, while the data in Fisher's Iris dataset can perfectly classify the species I. Setosa, a claim which is supported by its K-means result of the dimensionality reduction.