Problem Set 4 Programming: Logistic Regression, Perceptron Writeup

1) Perceptron
    a) N/A
    b) What are your trained coecients ? Do the two training procedures converge to the same solution? Why or why not? Will both classifiers have the same performance on the training set? What about on a new, held-out test set?

        The coefficients are:
        - Initial Theta = (0,0): coef = [3.202321 2.128344], mistakes = 1
        - Initial Theta = (1,0): coef = [0.911808 0.79204], mistakes = 4

        They do not converge to the same solution because they have different starting points. They are not required to converge to the same solution with different starting points because they will traverse the data differently, resulting in different coefficient values. Once the coefficients make no classification mistakes, the function will stop updating the coefficients, and there are many different coefficient solutions for linearly separable data. The perceptron algorithm does not guarantee a unique solution so both solutions are acceptable. Both classifiers will have the same (perfect) performance on the training set; however, are not guaranteed to have the same performance on a new, held-out test set.

    c) Compare the value of the mistake bound to the number of mistakes that you are getting on this data set. Explain why there might be a difference.

        gamma: 0.147979071041
        bound: 5.09745076046

        They are both under the bound. This is because we are not guaranteed to find $theta^*$ using the perceptron algorithm. $Theta^*$ is the hyperplane consistent with margins of at least $gamma^*$ so because we are not guaranteed to find this, we may find an acceptable theta with margins smaller than $gamma^*$ with fewer mistakes.

    d) Describe an adversarial procedure for selecting the order and value of labeled data points so as to increase the number of mistakes the perceptron algorithm makes before converging. Assume that the data are linearly separable.

        Before you traverse you should sort your feature vector by the L-2 norm of the feature, with the smallest magnitude first and the largest last. This will guarantee that the initial adjustments to theta are quite small, having little effect on the new prediction..

2) Perceptron vs Logistic Regression
    a) Before we start analyzing performance, let us understand the data (at a very high level). Run the perceptron algorithm (with offset) on this data. Is the training data linearly separable? How do you know? Why is this not surprising?

       The data is linearly separable, which we can tell because the accuracy that the scikit learn perceptron came up with was 1, meaning it could correctly classify every example. This is not surprising because there are 61 dimensions to 300 examples, so the ratio of examples to features is relatively low.
    b) N/A
    c)
        i) N/A
        ii) Report the mean and standard deviation of prediction accuracy for the different classifiers (up to three significant figures, computing statistics across all trials and folds).
        the perceptron mean is: 0.904. The standard deviation is: 0.051
        the dummy mean is: 0.431. The standard deviation is: 0.055
        the logistic mean is: 0.880. The standard deviation is: 0.053
        We see that the perceptron has the best performance.

        iii) For each pairwise combination of classifiers, calculate their t-test score to figure out if the differences are significant. Remember to use a two-tailed paired t-test (which can be computed using scipy.stats.ttest_rel(...)). To ease visualization, highlight any significant differences .
        the p-value for perceptron vs dummy is: 0.000000
        the p-value for perceptron vs logistic is: 0.000000
        the p-value for dummy vs logistic is: 0.000000
        the t-statistic for perceptron vs dummy is: 62.923587
        the t-statistic for perceptron vs logistic is: 6.530374
        the t-statistic for dummy vs logistic is: -60.657834
        All three p-values are significant. We can see from the t-statistics that the perceptron model is prefer to both the dummy and logistic models, and the logistic model is preferred to the dummy model.

    d) Next, let us investigate the effect of data preprocessing, in particular feature standardization, on classifier performance. Feature standardization transforms the data by removing the mean value of each feature, then scaling it by dividing non-constant features by their standard deviation. Use the preprocessing package from scikit-learn to standardize the features. Then repeat your experiments above.
        i) N/A

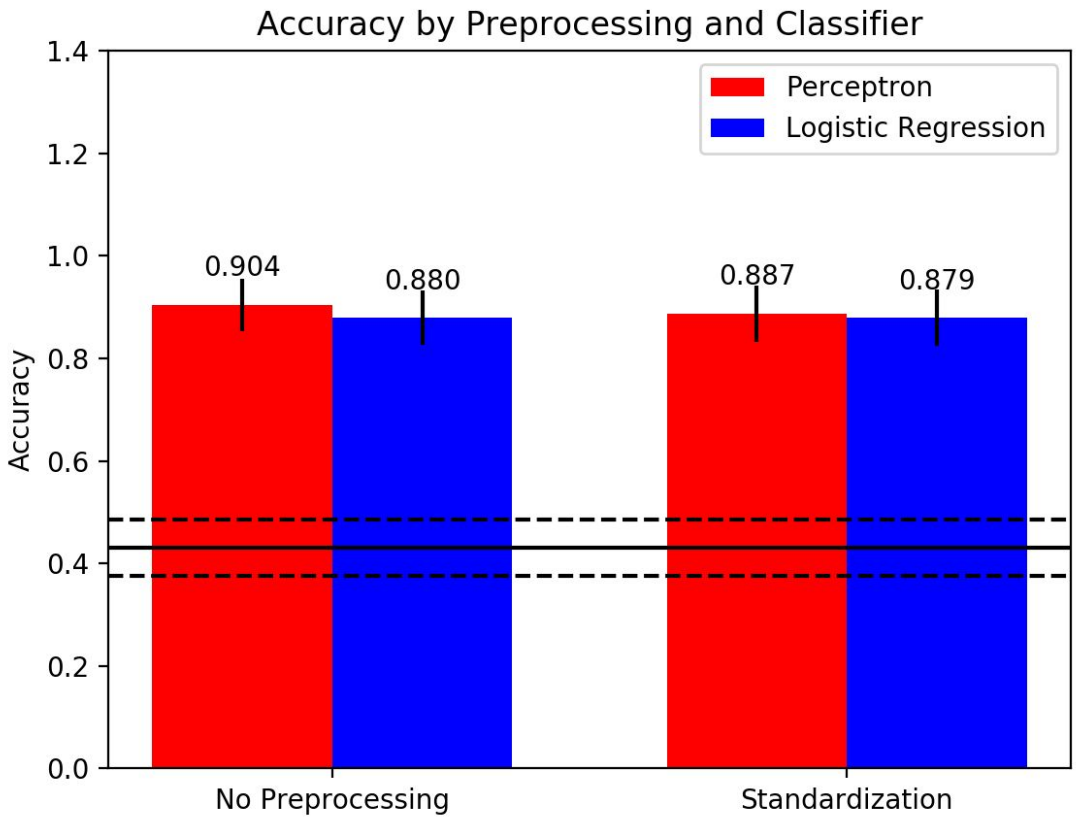
ii) Report the mean and standard deviation of prediction accuracy for the different classifiers (up to three significant figures, computing statistics across all trials and folds).
the perceptron mean_scaled is: 0.887. The standard deviation is: 0.055
the dummy mean_scaled is: 0.431. The standard deviation is: 0.055
the logistic mean_scaled is: 0.879. The standard deviation is: 0.055
We see that the perceptron is still the best model, though by not as large as a margin.

iii) For each pairwise combination of classifiers, calculate their t-test score to figure out if the differences are significant. Remember to use a two-tailed paired t-test (which can be computed using scipy.stats.ttest_rel(...)). To ease visualization, highlight any significant differences .
the p-value for perceptron vs dummy is: 0.000000
the p-value for perceptron vs logistic is: 0.021350
the p-value for dummy vs logistic is: 0.000000
the t-statistic for perceptron vs dummy is: 59.137657
the t-statistic for perceptron vs logistic is: 2.338900
the t-statistic for dummy vs logistic is: -60.371976
We see that the perceptron and logistic models are both preferred to the dummy model; however, the p-value is not significant in the comparison between the perceptron and logistic models.

e) Place graph here

i) Which data preprocessing method and classifier performs best? Were there any surprises?

The perceptron model performs the best. We found it interesting that both the perceptron and logistic models did not perform as well with preprocessing, thought the perceptron performed relatively worse.


Accuracy by Preprocessing and Classifier
Perceptron
Logistic Regression
Accuracy
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.904
0.880
0.887
0.879
No Preprocessing
Standardization