CS 5565, ECE5590CI, CS494R LAB 8-9 (SVMs and Unsupervised Learning) 95 Points

Name ___________________________________

1. View the videos at the following URLs
   https://www.youtube.com/watch?v=qhyyufR0930
   https://www.youtube.com/watch?v=L3n2VF7yKkk

   You may download the R Code for Labs and the Data Sets to use from the textbook website.
   http://www-bcf.usc.edu/~gareth/ISL/

2. (45 points total) We have seen that we can fit an SVM with a non-linear kernel in order to perform classification using a non-linear decision boundary.We will now see that we can also obtain a non-linear decision boundary by performing logistic regression using non-linear transformations of the features.

   (a) (5 points) Generate a data set with $n = 500$ and $p = 2$, such that the observations belong to two classes with a quadratic decision boundary between them. For instance, you can do this as follows:
   ```
   > x1=runif (500) -0.5
   > x2=runif (500) -0.5
   > y = 1*( x1^2-x2^2 > 0)
   ```

   (b) (5 points) Plot the observations, colored according to their class labels. Your plot should display X_1 on the x-axis, and X_2 on the $y$axis.

   (c) (5 points) Fit a logistic regression model to the data, using X_1 and X_2 as predictors.

   (d) (5 points) Apply this model to the *training data* in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the *predicted* class labels. The decision boundary should be linear.

   (e) (5 points) Now fit a logistic regression model to the data using non-linear functions of X_1 and X_2 as predictors (e.g. X_1^2, $X_1 \times X_2$, $\log(X_2)$, and so forth).

   (f) (5 points) Apply this model to the *training data* in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the *predicted* class labels. The decision boundary should be obviously non-linear. If it is not, then repeat (a)-(e) until you come up with an example in which the predicted class labels are obviously non-linear.

   (g) (5 points) Fit a support vector classifier to the data with X_1 and X_2 as predictors. Obtain a class prediction for each training observation. Plot the observations, colored according to the *predicted class labels.*

   (h) (5 points) Fit a SVM using a non-linear kernel to the data. Obtain a class prediction for each training observation. Plot the observations, colored according to the *predicted class labels.*

   (i) (5 points) Comment on your results.

3. (20 points total) At the end of Section 9.6.1, it is claimed that in the case of data that is just barely linearly separable, a support vector classifier with a small value of `cost` that misclassifies a couple of training observations may perform better on test data than one with a huge value of `cost` that does not misclassify any training observations. You will now investigate this claim.

   (a) (5 points) Generate two-class data with $p = 2$ in such a way that the classes are just barely linearly separable.

   (b) (5 points) Compute the cross-validation error rates for support vector classifiers with a range of `cost` values. How many training errors are misclassified for each value of `cost` considered, and how does this relate to the cross-validation errors obtained?

   (c) (5 points) Generate an appropriate test data set, and compute the test errors corresponding to each of the values of `cost` considered. Which value of `cost` leads to the fewest test errors, and how does this compare to the values of `cost` that yield the fewest training errors and the fewest cross-validation errors?

   (d) (5 points) Discuss your results.

4. Unsupervised Learning
   View the videos at the following URLs
   https://www.youtube.com/watch?v=lFHISDj_4EQ
   https://www.youtube.com/watch?v=YDubYJsZ9iM
   https://www.youtube.com/watch?v=4u3zvtfqb7w

   You may download the R Code for Labs and the Data Sets to use from the textbook website.
   http://www-bcf.usc.edu/~gareth/ISL/

5. (10 points total) In Section 10.2.3, a formula for calculating PVE was given in Equation 10.8. We also saw that the PVE can be obtained using the sdev output of the `prcomp()` function. On the `USArrests` data, calculate PVE in two ways:

   (a) (5 points) Using the `sdev` output of the `prcomp()` function, as was done in Section 10.2.3.

   (b) (5 points) By applying Equation 10.8 directly. That is, use the `prcomp()` function to compute the principal component loadings. Then, use those loadings in Equation 10.8 to obtain the PVE.

   These two approaches should give the same results.

   Hint: You will only obtain the same results in (a) and (b) if the same data is used in both cases. For instance, if in (a) you performed `prcomp()` *using centered and scaled variables, then you must center and scale the variables before applying Equation 10.3 in (b).*

6. (20 points total) Consider the `USArrests` data. We will now perform hierarchical clustering on the states.

   (a) (5 points) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

   (b) (5 points) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

   (c) (5 points) Hierarchically cluster the states using complete linkage and Euclidean distance, *after scaling the variables to have standard deviation one.*

   (d) (5 points) What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justification for your answer.