1. **Supervised, Semi-Supervised, and Unsupervised Learning**

   (a) Download the Blood Transfusion Service Center Data Set from: `https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center`. This data has two output classes.Use the first 20% of the positive and negative classes in the file as the test set and the rest as the training set.

   (b) **Supervised Learning:** Train an $\mathscr{L}_1$-penalized SVM to classify the data. Use 5 fold cross validation to choose the penalty parameter. Use normalized data. Report the accuracy, AUC, ROC, and confusion matrix for both training and test sets.

   (c) **Semi-Supervised Learning/ Self-training:** select 50% of the positive class along with 50% of the negative class in the training set as *labeled data* and the rest as *unlabelled data*. You can select them randomly.

      i. Train an $\mathscr{L}_1$-penalized SVM to classify the labeled data. Choose the penalty parameter using 5 fold cross validation.

      ii. Find the unlabeled data point that is the closest to the decision boundary of the SVM. Let the SVM label it (ignore its true label), and add it to the labeled data, and retrain the SVM. Continue this process until all unlabeled data are used. Test the final SVM on the test data and report the accuracy, AUC, ROC, and confusion matrix for the test set.

   (d) **Unsupervised Learning:** Run k-means algorithm on the whole training set. Ignore the labels of the data, and assume $k = 2$.

      i. Run the k-means algorithm multiple times. How do you make sure that the algorithm was not trapped in a local minimum?

      ii. Compute the centers of the two clusters and find the closest 30 data points to each center. Read the true labels of those 30 data points and take a majority poll within them. The majority poll becomes the label predicted by k-means for the members of each cluster. Then compare the labels provided by k-means with the true labels of the training data and report accuracy and the confusion matrix.[1]

      iii. Classify test data based on their proximity to the centers of the clusters. Report accuracy and confusion matrix for the test data.

   (e) Extra Practice: The more principled way of doing the above steps is to run a Monte-Carlo simulation: Repeat the supervised, unsupervised, and semi-supervised learning procedures 50 or 100 times, for randomly selected train and test data(make sure you use 20% of both the positve and negative classes, because this is an imbalanced data set). Then compare the *average* accuracies that you obtain from each algorithm.

   (f) One expects that supervised learning on the full data set works better than semi-supervised learning with half of the data set labeled.One expects that unsuper-

[1]Here we are using k-means as a classifier. The closest 30 data points to each center are labeled by *experts*, so as to use k-means for classification. Obviously, this is a naïve approach.

vised learning underperforms in such situations. Compare the results you obtained by those methods.

2. **K-Means Clustering on a Multi-Class and Multi-Label Data Set**

   (a) Use k-means clustering on Anuran Calls (MFCCs) Data Set of Homework 4. Choose k automatically based on one of the methods provided in the slides (CH or Gap Statistics or scree plots) or any other method you know.

   (b) In each cluster, determine which family is the majority by reading the true labels. Repeat for genus and species.

   (c) Now for each cluster you have a majority label triplet (family, genus, species). Calculate the average Hamming distance between the true labels and the labels assigned by clusters.

   (d) Extra Practice: Again, a more principled way of doing the above experiment is by Monte-Carlo Simulation. Perform it!

3. ISLR 10.7.2

4. Extra Practice: The rest of problems in 10.7.