1. **Active Learning Using Support Vector Machines**

   (a) Download the banknote authentication Data Set from: `https://archive.ics.uci.edu/ml/datasets/banknote+authentication`. Choose 472 data points randomly as the test set. This is a binary classification problem.

   (b) Repeat each of the following two procedures 50 times. You will have 50 errors for 90 SVMs per each procedure.

      i. Train a SVM with a pool of 10 randomly selected data points from the training set using linear kernel and $\mathscr{L}_1$ penalty. Select the penalty parameter using 10-fold cross validation.[1] Repeat this process by adding 10 other randomly selected data points to the pool, until you use all the 900 points. Do NOT replace the samples back into the training set at each step. Calculate the test error for each SVM. You will have 90 SVMs that were trained using 10, 20, 30, ... , 900 data points and their 90 test errors. You have implemented *passive learning*.

      ii. Train a SVM with a pool of 10 randomly selected data points from the training set using linear kernel and $\mathscr{L}_1$ penalty. Select the parameters of the SVM with 10-fold cross validation. Choose the 10 closest data points in the training set to the hyperplane of the SVM[2] and add them to the pool. Do not replace the samples back into the training set. Train a new SVM using the pool.Repeat this process until all training data is used. You will have 90 SVMs that were trained using 10, 20, 30,..., 900 data points and their 90 test errors. You have implemented *active learning*.

   (c) Average the test errors for the incrementally trained 90 SVMs in 1(b)i and1(b)ii. By doing so, you are performing a Monte Carlo simulation. Plot average test error versus number of training instances for both active and passive learners on the same figure and report your conclusions.

2. **Multi-class and Multi-Label Classification Using Support Vector Machines**

   (a) Download the Anuran Calls (MFCCs) Data Set from: `https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29#`. Choose 70% of the data randomly as the training set.

[1] How to choose parameter ranges for SVMs? One can use wide ranges for the parameters and a fine grid (e.g. 1000 points) for cross validation; however,this method may be computationally expensive. An alternative way is to train the SVM with very large and very small parameters on the whole training data and find very large and very small parameters for which the training accuracy is not below a threshold (e.g., 70%). Then one can select a fixed number of parameters (e.g., 20) between those points for cross validation. For the penalty parameter, usually one has to consider increments in $\log(\lambda)$. For example, if one found that the accuracy of a support vector machine will not be below 70% for $\lambda = 10^{-3}$ and $\lambda = 10^{6}$, one has to choose $\log(\lambda) \in \{-3, -2, \ldots, 4, 5, 6\}$. For the Gaussian Kernel parameter, one usually chooses linear increments,e.g. $\sigma \in \{.1, .2, \ldots, 2\}$. When both σ and λ are to be chosen using cross-validation, combinations of very small and very large λ's and σ's that keep the accuracy above a threshold (e.g.70%) can be used to determine the ranges for σ and λ. Please note that these are very rough rules of thumb, not general procedures.

[2] You may use the result from linear algebra about the distance of a point from a hyperplane.

(b) Each instance has three labels: Families, Genus, and Species. Each of the labels has multiple classes. We wish to solve a multi-class and multi-label problem. One of the most important approaches to multi-class classification is to train a classifier for each label. We first try this approach:

i. Research exact match and hamming score/ loss methods for evaluating multi-label classification and use them in evaluating the classifiers in this problem.

ii. Train a SVM for each of the labels, using Gaussian kernels and one versus all classifiers. Determine the weight of the SVM penalty and the width of the Gaussian Kernel using 10 fold cross validation. You are welcome to try to solve the problem with both normalized and raw attributes and report the results.

iii. Repeat 2(b)ii with $\mathscr{L}_1$-penalized SVMs. Remember to normalize the attributes.

iv. Repeat 2(b)iii by using SMOTE or any other method you know to remedy class imbalance. Report your conclusions about the classifiers you trained.

v. Extra Practice: Study the Classifier Chain method and apply it to the above problem.

vi. Extra Practice: Research how confusion matrices, precision, recall, ROC, and AUC are defined for multi-label classification and compute them for the classifiers you trained in above.

3. **K-Means Clustering on a Multi-Class and Multi-Label Data Set** Monte-Carlo Simulation: Perform the following procedures 50 times, and report the average and standard deviation of the 50 Hamming Distances that you calculate.

(a) Use k-means clustering on the whole Anuran Calls (MFCCs) Data Set (do not split the data into train and test, as we are not performing supervised learning in this exercise). 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 (score) between the true labels and the labels assigned by clusters.

4. ISLR 10.7.2

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