STA 4364 HW 5

Submission Format: Please submit your homework as 1) a HTML or pdf document, and 2) also submit the source file in either R Markdown or Jupyter notebook format (at most one of each type of file).

Problems can be done in Python or R. ISL = Introduction to Statistical Learning textbook.

Problem 1 Fire debris analysis interprets gas chromatography – mass spectrometry data to determine if a fire debris sample collected at a fire scene contains an ignitable liquid or not. All fire debris contains the remains of substrate materials that decompose in the fire; however, some may also contain the residue of an ignitable liquid. The volatile chemical compounds contained in the fire debris are collected and subsequently analyzed by gas chromatography-mass spectrometry generating a total ion chromatogram (TIC). A TIC provides the relative concentration represented as intensity of the separated chemical compounds throughout the time of analysis. The separated chemical compounds are seen as peaks in the chromatogram at different retention indices, which are related to the time required for each compound to transverse the gas chromatographs column. The data set is generated by National Center for Forensic Science at UCF, and is in the file `2023-08-18-IS-tic.csv`. It contains the intensity of the chemical compounds at each retention index that ranges from 1 to 2800. In addition, the file `2023-08-18-MixDat.csv` contains the classification designation, labelled class of the fire debris sample, as containing an ignitable liquid (IL) or not containing an ignitable liquid (SUB).

(a) Load the data (can use the pandas function `read_table` with the arguments `sep=','` and `header=None`). Split the data into a training and test set. Scale and center the columns using the mean and standard deviation of each column from the *training set* (make sure you use the same scaling on the test set that is used on the training set).

(b) Learn the following models to classify the training data:

- **Logistic Regression**: Can import `LogisticRegression` from `sklearn.linear_model`.
- **LDA**: Can import `LinearDiscriminantAnalysis` from `sklearn.discriminant_analysis`.
- **KNN Classifier**: Need to choose the number of neighbors k.
- **Linear SVM**: Need to choose the margin penalty C as a hyperparameter.
- **Gaussian (Radial) SVM**: Need to choose the margin penalty C and the radius width γ.

To tune hyperparameters for each model, you can either use cross-validation or hand-tune by examining the model performance for reasonable values of the hyper-parameters.

(c) Apply your models to the test set. Report the accuracy, visualize an ROC curve, and report the AUC for each model.

Problem 2 In this problem you will compare the performance of a variety of classifiers that you have learned about so far. The data is in the file `magic04.data` and the column names are in the file `magic04.names`. The last column is a categorical response with values `g` or `h`, and the rest of the columns are numerical features. You can read more about the dataset here.

(a) Load the data (can use the pandas function `read_table` with the arguments `sep=','` and `header=None`). Split the data into a training and test set. Scale and center the columns using the mean and standard deviation of each column from the *training set* (make sure you use the same scaling on the test set that is used on the training set).

(b) Learn the following models to classify the training data:

- **Logistic Regression**: Can import `LogisticRegression` from `sklearn.linear_model`.
- **LDA**: Can import `LinearDiscriminantAnalysis` from `sklearn.discriminant_analysis`.

- **KNN Classifier**: Need to choose the number of neighbors k.
- **Linear SVM**: Need to choose the margin penalty C as a hyperparameter.
- **Gaussian (Radial) SVM**: Need to choose the margin penalty C and the radius width γ.

To tune hyperparameters for each model, you can either use cross-validation or hand-tune by examining the model performance for reasonable values of the hyper-parameters.

(c) Apply your models to the test set. Report the accuracy, visualize an ROC curve, and report the AUC for each model. For Logistic Regression, report the most meaningful predictors.