Fall 2018 DS 4400

Homework 2

Due date: Friday, October 19 2018

Instructions

Please make sure your name appears in all the files and source code you submit. Submit a PDF file named ”%LASTNAME%_HW2.pdf” in Gradescope that will include the following:

- Link to Jupyter notebook with Python or R code. You can store the code in an online service (such as Google drive, Dropbox, or private github).
- Link to a simple README file with instructions on how to run your code.
- Answers to the questions. You can use Latex or Word to generate the PDF.

Course policy on collaboration and cheating:

- You may discuss the concepts with your classmates, but write up the answers entirely on your own.
- You cannot share your code with your classmates.
- You cannot use code from the Internet for your assignment.

Dataset: The dataset for this assignment is the SPAMBASE dataset from the UCI repository and it is available at: `https://archive.ics.uci.edu/ml/datasets/spambase`

The first 57 columns are features counting word frequencies (see documentation at `https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names`). The last column indicates 1 for the SPAM class and 0 for the HAM class.

Problem 1 [Logistic regression] - 25 points

Use an existing package of your choice to train and test a logistic regression model.

(a) Split the original data into 75% for training and 25% for testing. Choose the training set at random and ensure that the ratio of SPAM examples in the training set is close to the ratio of 39.4% SPAM examples in the entire dataset.

Train a logistic regression model on the training set and output the following **on the testing set**:

1. **Confusion matrix**

2. **True Positives**, **False Positives**, **True Negatives**, **False Negatives**
3. **Accuracy**, **error**
4. **Precision**, **Recall**, **F1 score**

(b) Print the coefficients of the features in the model. Which features contribute mostly to the prediction? Which ones are positively correlated and which ones are negatively correlated with the SPAM class?

(c) Vary the decision threshold $T \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and report for each value the model accuracy. Plot a graph of the accuracy of the model as a function of the threshold.

Problem 2 [Comparing classifiers] - 25 points

In this problem, you will use existing packages of your choice for training and testing various classifiers, and then compare them. You will use the same SPAMBASE dataset.

Split the original data into 75% for training and 25% for testing (chosen at random). Train the following classifiers using the training data:

1. Logistic regression
2. LDA
3. kNN
4. Decision tree

(a) Experiment with different values of k for kNN and report 2 metrics: **accuracy** and **error**. Choose the value of k that gives the highest accuracy.

(b) Print the **accuracy** and **error** metrics for all 4 classifiers. Which model is performing best? Which one is performing worst? Write down some observations.

(c) Generate a graph that includes 4 ROC curves (one for each of the 4 classifiers). Compute the Area Under the Curve (AUC) metric for all 4 classifiers.

Problem 3 [kNN] 25 points

In this problem, you will implement your own kNN (k Nearest Neighbors) model, using the Euclidian distance between points as a distance metric. You will also compare your model with the one trained with the kNN package of your choice (the same one you used in Problem 2).

Split the original data into 75% for training and 25% for testing (chosen at random).

(a) Implement a function that computes the Euclidian Distance between 2 points with d features. If $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$, the Euclidian Distance between points x and y is $\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$.

(b) Write the implementation for the kNN classifier. Given the value of k and the training set, you should implement a **test** function that produces a label for a new point x in the testing set.

(c) Pick several values of k (the same ones you picked in Problem 2) and print the **accuracy** and **error** metrics **on the test set** using your implementation of the kNN classifier.

(d) Compare the results obtained by your implementation with those obtained in Problem 2 with the package. Are the results similar or different? If there are differences, explain why.

Problem 4 [Cross validation] 25 points

In this problem, you will implement your own k-fold cross-validation algorithm and apply it to two linear classifiers (Logistic Regression and LDA).

(a) Implement k-fold cross-validation (CV) for training a model. The CV algorithm consists of the following steps:

   (a) Divide the entire data into k partitions of equal size.

   (b) Run k experiments. In each experiment $i \in \{1, \ldots, k\}$, train on $k - 1$ partitions and test on the validation set (partition i).

   (c) Record the validation error for each experiment.

   (d) Compute and print the **average validation error** across all k experiments.

(b) Run the CV experiment for logistic regression and LDA for $k \in \{5, 10, 20\}$. You can use a package for training the logistic regression and LDA models. Print for each model the average validation error for each value of k.

(c) Which model performs better? Compare the results.