CIVE 7150 - Data-Driven Decision Support for Civil and Environmental Engineering

Homework 3

Due date: March 8, 2024

Instructions

- Submit a PDF writeup named "LASTNAME_HW3.pdf" on Canvas. Please answer all the questions in the PDF. If a problem asks for graphs or other results, please extract the results from the code and include them in the report. The PDF report should have complete answers to all the questions.
- Please use Jupyter notebooks in Python and include comments in the code if needed.

Dataset: The dataset for this assignment is the SPAMBASE dataset from the UCI repository 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 on the SPAMBASE dataset.

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

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 $T \in \{0.25, 0.5, 0.75, 0.9\}$ and report for each value the model accuracy, precision, and recall. Comment on how these metrics vary with the choice of threshold.

Problem 2 [Gradient Descent for Logistic regression] - 25 points

Use your implementation of Gradient Descent from Homework 2 and adapt it for logistic regression. Take 3 values of the learning rate and report the cross-entropy loss objective after 10, 50, and 100 iterations. At 100 iterations, report the accuracy, precision, recall, and F1 score for the 3 learning rates, and compare with the metrics given by the package on the training and testing sets.

Problem 3 [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.

You can use the same training and testing data as in Problem 1. Train the following classifiers using the training data:

1. Logistic regression
2. LDA
3. kNN

(a) Use cross-validation to select the k hyper-parameter for kNN. Show the **accuracy**, **error**, **precision**, and **recall** metrics on the validation dataset for multiple values of k. Select the value of k that minimizes the average cross validation error.

(b) Print the **accuracy**, **error**, **precision**, and **recall** metrics for all 3 classifiers on both training and testing data. Which model is performing best? Which one is performing worst? Write down some observations.

(c) Generate a graph that includes ROC curve for the logistic regression classifier on the testing set. Compute the Area Under the Curve (AUC) metric. You can use a package for this.

(d) Write code to plot a ROC curve without a package for logistic regression. Vary the prediction threshold $T \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, and compute false positive and true positive rates for each threshold. Plot a ROC curve for these points, and compare it with the ROC curve generated with the package in part (c). What are the differences and what changes can you make to get the two ROC curves to become more similar?

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). Use the SPAMBASE dataset for this problem.

(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 $k \in \{5, 10\}$. 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.