CS 412: Spring'21
Introduction To Data Mining

Assignment 4

(Due Tuesday, May 4, 11:59 pm)

General Instructions

- Feel free to talk to other students of the class while doing the homework. We are more concerned that you learn how to solve the problems than that you demonstrate that you solved it entirely on your own. You should, however, write down your solution yourself. Please try to keep the solution brief and clear.
- Please use Slack first if you have questions about the homework. You can also use CampusWire, send us e-mails, and/or come to our office hours. If you are sending us emails with questions on the homework, please cc all of us (Arindam, Carl, Jialong, and Qi) for faster response.
- The homework is due at 11:59 pm on the due date. We will be using **Gradescope** (details in the next section) for collecting the homework assignments. Please submit your code as homework4.py via Gradescope (`https://www.gradescope.com/`). Please do NOT email a hard copy of your code. Contact the TAs if you are having technical difficulties in submitting the assignment. We will NOT accept late submissions!
- All programming will be in python 3.6.9. A stub submission file is on Compass.
- You cannot use external libraries for k-fold cross-validation. You can use external libraries for classifiers. scikit-learn and numpy are available.

Programming Assignment Instructions

- The homework will be graded using Gradescope. You should be automatically added to this course. You will be able to submit your code as many times as you want. For this homework, you will be required to test two different splitting strategies for classification. Each will contribute half of the possible points on this problem.

Assignment. The assignment will focus on developing your own code for k-fold cross-validation and random train-test split validation. Your code will be evaluated using five standard classification models applied to a multi-class classification dataset.

Dataset: We will be using the following dataset for the assignment.

`Digits`: The `Digits` dataset comes prepackaged with `scikit-learn` (sklearn.datasets.load_digits). The dataset has 1797 points, 64 features, and 10 classes corresponding to ten numbers $0, 1, \ldots, 9$. The dataset was (likely) created from the following dataset:
`https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`

Classification Methods. We will consider five classification methods from `scikit-learn`:

- Linear support vector classifier: `LinearSVC`,
- Support vector classifier: `SVC`,
- Logistic Regression: `LogisticRegression`,
- Random Forest Classifier: `RandomForestClassifier`, and
- Gradient Boosting Classifier: `XGBClassifier`.

Use the following parameters for these methods:

- `LinearSVC: max_iter=2000`
- `SVC: gamma='scale', C=10`
- `LogisticRegression: penalty='l2', solver='lbfgs', multi_class='multinomial', max_iter=5000`
- `RandomForestClassifier: max_depth=20, random_state=0, n_estimators=500`
- `XGBClassifier: default parameters`

1. (50 points) Develop code for `get_splits`(n, k) (30 points), which returns a list of folds, and `my_cross_val`(`method`,X,$\mathbf{y}$,k) (20 points), which performs k-fold cross-validation on $(X, \mathbf{y})$ using `method`, and returns the error rate in each fold. `my_cross_val`, should return the error rates in each fold for the five methods: `LinearSVC`, `SVC`, `LogisticRegression`, `RandomForestClassifier`, and `XGBClassifier` applied to the `Digits` dataset.

   You will have to submit **code** for these methods:

   (a) **Code**: You will have to submit code for `get_splits`(n, k) and `my_cross_val`(`method`,X,$\mathbf{y}$,k).
   The **input parameters** are: (1) `method`, which specifies the (class) name of one of the five classification methods under consideration, (2) X,$\mathbf{y}$ which is data for the classification problem, (3) k, the number of folds for cross-validation, (4) n, which is the length of data to split.
   `get_splits`(n, k) has the following **output**: (1) a list containing k lists, arrays, or sets. There will be elements from 0 to $n-1$. Each of these sublists should contain roughly $\frac{1}{k}$ of these elements. The sublists should be disjoint and roughly the same size. For example,

`get_splits`(4, 2) might return [[0,2], [1,3]]. You can use this method in `my_cross_val` to get your splits. You should make sure to randomize your splits as well.
`my_cross_val`(`method`,X,$\mathbf{y}$,k) has the following **output**: (1) the test set error rates for each of the k folds. The error should be measured as $\frac{\texttt{wrong}}{\texttt{total}}$. The mean and standard deviation of your errors should fall within three standard deviations of the solution.

2. (50 points) Develop code for `my_train_test`(`method`,X,$\mathbf{y}$,π,k), which performs random splits on the data $(X, \mathbf{y})$ so that $\pi \in [0, 1]$ fraction of the data is used for training using `method`, rest is used for testing, and the process is repeated k times, after which the code returns the error rate for each such train-test split. Your `my_train_test` will be tested with $\pi = 0.75$ and $k = 10$ on the five methods: `LinearSVC`, `SVC`, `LogisticRegression`, `RandomForestClassifier`, and `XGBClassifier` applied to the `Digits` dataset.

   (a) **Code**: You will have to submit code for `my_train_test`(`method`,X,$\mathbf{y}$,π,k).
   This main file has **input**: (1) `method`, which specifies the (class) name of one of the five classification methods under consideration, (2) X,$\mathbf{y}$ which is data for the classification problem, (3) π, the fraction of data chosen randomly to be used for training, (4) k, the number of times the train-test split will be repeated.
   It has **output**: (1) A list of the test set error rates for each of the k splits. Error should be calculated as above. The grader will compare the mean and standard deviation of your list with the solution; it must be within three standard deviations.