Week 6 Project

ACADEMIC HONESTY

As usual, the standard honour code and academic honesty policy applies. We will be using automated **plagiarism detection** software to ensure that only original work is given credit. Submissions isomorphic to (1) those that exist anywhere online, (2) those submitted by your classmates, or (3) those submitted by students in prior semesters, will be detected and considered plagiarism.

INSTRUCTIONS

Assume you are given labeled data $(x_1, y_1), \ldots, (x_N, y_N)$, where $x \in \mathbb{R}^d$ and $y \in \{1, \ldots, K\}$. In this assignment, you will implement a *K*-class Bayes classifier. In the specific classifier that you will implement, assume the following generative model: For the *i*-th data point, assume that

$$y_i \overset{iid}{\sim} Discrete(\pi), \quad x_i | y_i \sim Normal(\mu_{y_i}, \Sigma_{y_i}), \quad i = 1, \ldots, N.$$

For this model, you will need to derive the maximum likelihood updates for the class prior probability vector $\hat{\pi}$ and the class-specific Gaussian parameters $(\hat{\mu}_k, \hat{\Sigma}_k)$ for each class $k = 1, \ldots, K$, where $\hat{\cdot}$ indicates "maximum likelihood estimate". While you will not turn in these derivations, you will need to implement them in your code, as well as the prediction for a new point y_0 given x_0 and these estimates:

$$Prob\left(y_0 = y | x_0, \hat{\pi}, (\hat{\mu}_1, \hat{\Sigma}_1), \ldots, (\hat{\mu}_K, \hat{\Sigma}_K)\right)$$

More details about the inputs we provide and the expected outputs are given below.

Sample starter code to read the inputs and write the outputs: Download hw2_classification.py

---

WHAT YOU NEED TO SUBMIT

You can use either **Python (3.6.4)** or Octave coding languages to complete this assignment. Octave is a free version of Matlab. Your Matlab code should be able to directly run in Octave, but you should not assume that advanced built-in functions will be available to you in Octave. Unfortunately we will not be supporting other languages in this course.

.

Depending on which language you use, we will execute your program using one of the following two commands.

.

Either

```
$ python3 hw2_classification.py X_train.csv y_train.csv
X_test.csv
```

Or

```
$ octave -q hw2_classification.m X_train.csv y_train.csv
X_test.csv
```

.

You must name your file as indicated above for your chosen language. If both files are present, we will only run your Python code. We will create and input the csv data files to your code.

.

The csv files that we will input into your code are formatted as follows:.

1. **X_train.csv:** A comma separated file containing the covariates. Each *row* corresponds to a single vector x_i.
2. **y_train.csv:** A file containing the classes. Each row has a single number and the *i*-th row of this file combined with the *i*-th row of "X_train.csv" constitutes the labeled pair (y_i, x_i). There are 10 classes having index values 0,1,2,...,9.
3. **X_test.csv:** This file follows exactly the same format as "X_train.csv". No class

file is given for the testing data.

WHAT YOUR PROGRAM OUTPUTS

When executed, you should have your code write the output to the file listed below. It is required that you follow the formatting instructions given below.

.

`probs_test.csv`: This is a comma separated file containing the posterior probabilities of the label of each row in "X_test.csv". Since there are 10 classes, the *i*-th row of this file should contain 10 numbers, where the *j*-th number is the probability that the *i*-th testing point belongs to class *j-1* (since classes are indexed 0 to 9 here).

.

Note on Correctness

Please note that for both of these problems, there is one and only one correct solution. Therefore, we will grade your output based on how close your results are to the correct answer. We strongly suggest that you test out your code on your own computer before submitting. The UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) has a good selection of datasets for classification.

USE OF VOCAREUM

This assignment uses Vocareum for submission and grading. Vocareum comes equipped with an editing environment that you may use to do your development work. You are **NOT** required to use the editor. In particular, you are free to choose your favorite editor / IDE to do your development work on. When you are done with your work, you can simply upload your files onto Vocareum for submission and grading.

However, your assignments will be graded on the platform, so you **MUST** make sure that your code passes at least the submission test cases. In particular, do not use third-party libraries and packages. We do not guarantee that they will work on the platform, even if they work on your personal computer. For the purposes of this project, everything that comes with the standard Python or Matlab libraries should be more than sufficient.

To check the formatting of your submission, select to have your code submitted, but not graded. We will output the results of the formatting check to SubmissionReport.txt. We can guarantee a very low grade if you do not pass this submission test. Once your outputs satisfy the formatting requirements and you are confident in your code, select to have it graded.

- **You will have unlimited opportunities to submit your code for grading**
- **You can test your code in the terminal without submitting it**
- **You will get graded once you click "Submit"**
- **The assignment due date is December 8th, 23:30 UTC. Assignments submitted before October 27th, 23:30 UTC are eligible for bonus points (we count grades on your latest submission). Due to edX policy, all assignment grades are capped at 100%.**

Verified Track Access

Graded assessments are available to Verified Track learners.