Homework 4

Due: Thursday, February 28, 2019 at 12:00pm **(Noon)**

Programming Assignment

Introduction

In this assignment, you'll implement Naive Bayes and use this algorithm to classify handwritten digits. The textbook section relevant to this assignment is 24.2 on page 347.

Stencil Code & Data

You can find the stencil code for this assignment on the course website. We have provided the following two files:

- `main.py` is the entry point of your program which will read in the data, run the classifiers and print the results.
- `models.py` contains the `NaiveBayes` model which you will be implementing.

You should *not* modify any code in the `main.py`. All the functions you need to fill in reside in `models.py`, marked by `TODO`s. You can see a full description of them in the section below.

To feed the data into the program successfully, please do *not* rename the data files and also make sure all the data files are in a directory named `data` located in the same directory as `main.py`. To run the program, run `python main.py` in a terminal.

MNIST Dataset

You will be using the well-known MNIST Dataset, which includes 60,000 examples for training and 10,000 for testing. All images are size normalized to fit in a 20×20 pixel box and this box is centered in a 28×28 image based on the center of mass of its pixels. The full description of this dataset is available at `http://yann.lecun.com/exdb/mnist/`. There are four files:

- `train-images-idx3-ubyte.gz`: images of 60,000 training examples,
- `train-labels-idx1-ubyte.gz`: labels of 60,000 training examples,
- `t10k-images-idx3-ubyte.gz`: images of 10,000 testing examples,
- `t10k-labels-idx1-ubyte.gz`: labels of 10,000 testing examples.

On a department machine, you can copy the files to your working directory by running the command

```
cp /course/cs1420/pub/hw4/* <DEST DIRECTORY>
```

where `<DEST DIRECTORY>` is the directory where you would like to copy the four data files. If you are working locally, you will need to use the `scp` command to copy the files to your computer over `ssh`:

```
scp <login>@ssh.cs.brown.edu:/course/cs1420/pub/hw4/* <DEST DIRECTORY>
```

Data Format

The original feature values in this dataset are integers ranging from 0 to 255 (encoding grey scale values), but we have thresholded those features to make them binary (1 for values higher than 99, 0 otherwise). The labels are the same as the original dataset, which are integers in the range $[0, 9]$.

As always, we have written all the preprocessing code for you. The final dataset is represented by a `namedtuple` with two fields:

- `data.inputs` is a $m \times 784$ `NumPy` array that contains the binary features of the m examples.
- `data.labels` is a m dimensional `NumPy` array that contains the labels of the m examples.

You can find more infomation on `namedtuple` **here**

The Assignment

In `models.py`, there are three functions you will implement. They are:

- `NaiveBayes`:
  - **train()** uses maximum likelihood estimation to learn the parameters. Because all the features are binary values, you should use the Bernoulli distribution (as described in lecture) for the features.
  - **predict()** predicts the labels using the inputs of test data.
  - **accuracy()** computes the percentage of the correctly predicted labels over a dataset.

Note: You are not allowed to use any off-the-shelf packages that have already implemented these models, such as scikit-learn. We're asking you to implement them yourself.

Project Report

Guiding Questions

- Report the training and testing error of the Naive Bayes classifier. Discuss the results in a short paragraph.

Grading Breakdown

We expect the accuracy that `NaiveBayes` reaches should be above 84%. As always, you will primarily be graded on the correctness of your code and not based on whether it does or does not achieve the accuracy target.

The grading breakdown for the assignment is as follows:

Naive Bayes	90%
Report	10%
Total	100%

Handing in

Programming Assignment

To hand in the programming component of this assignment, first ensure that your code runs on *Python 3* using our course `virtualenv`. You can activate the `virtualenv` on a department machine by running the following command in a Terminal:

```
source /course/cs1420/cs142_env/bin/activate
```

Once the `virtualenv` is activated, run your program and ensure that there are no errors. We will be using this `virtualenv` to grade all programming assignments in this course so we recommend testing your code on a department machine each time before you hand in. Note that handing in code that does not run may result in a significant loss of credit.

To hand in the coding portion of the assignment, run `cs142_handin hw4` from the directory containing all of your source code and your report in a file named `report.pdf`.

Anonymous Grading

You need to be graded anonymously, so do not write your name anywhere on your handin. Instead, you should use the course ID that you generated when filling out the collaboration policy form. If you do not have a course ID, you should email the HTAs as soon as possible.

Obligatory Note on Academic Integrity

Plagiarism—don't do it.

As outlined in the Brown Academic Code, attempting to pass off another's work as your own can result in failing the assignment, failing this course, or even dismissal or expulsion from Brown. More than that, you will be missing out on the goal of your education, which is the cultivation of your own mind, thoughts, and abilities. Please review this course's collaboration policy and, if you have any questions, please contact a member of the course staff.