Homework 8

Due: Thursday, March 18, 2021 at 12:00pm **(Noon)**

Programming Assignment

Introduction

In this assignment, you'll implement Naive Bayes and use this algorithm to classify the credit rating (good or bad) of a set of individuals. The textbook section relevant to this assignment is 24.2 on page 347.

Stencil Code & Data

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. Note that pre-processing has been done for you; feel free to examine the code for what exactly was done.
- `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 not to move either the data file or `models.py` from the stencil. To run the program, run `python main.py` in a terminal.

You can find the stencil code for this assignment in the course directory. On a department machine, you can copy the files to your working directory by running the command

```
cp /course/cs1420/pub/hw08/* <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/hw08/* <DEST DIRECTORY>
```

German Credit Dataset

You will be using the commonly-used German Credit dataset, which includes 1000 total examples. The prediction task is to decide whether someone's credit is good (1) or bad (0). A full list of attributes can be found **here**; note that this includes sensitive attributes like sex, age, and personal status. The specific file we are using comes from **Friedler et.al., 2019**. This data is in the file `german_numerical-binsensitive.csv`.

Data Format

The original feature values in this dataset are mixed—some categorical, some numerical. We have written all the preprocessing code for you, transforming numerical attributes into categories and encoding all attributes as binary features. After preprocessing, there are a total of 69 attributes which take on either 1 or 0. We also balance the dataset such that there are 350 examples with good credit and 300 examples with bad credit. **`credit = 1` corresponds to "good" credit, and `credit = 0` corresponds to "bad" credit.**

The Assignment

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

- `NaiveBayes`:
    - **train()** uses maximum likelihood estimation to learn the parameters (attribute distributions and priors distribution). Because all the features are binary values, you should use the Bernoulli distribution (as described in lecture) for the features. Remember to add Laplace smoothing as you calculate the distributions.
    - **predict()** predicts the labels using the inputs of test data. You should return 1-D numpy array.
    - **accuracy()** computes the percentage of the correctly predicted labels over a dataset.

Note that there is also a **print_fairness()** method implemented for you in `NaiveBayes`. You should not change this method. Additionally, you are not allowed to use any off-the-shelf packages that have already implemented Naive Bayes, such as scikit-learn; we're asking you to implement it yourself.

Project Report

Guiding Questions

- Report the training and testing accuracy of the Naive Bayes classifier. (A correct implementation should have testing accuracy above 70%.) (1 point)
- What strong assumption about the features/attributes of the data does Naive Bayes make? Comment on this assumption in the context of credit scores. (3 points)
- This dataset was originally structured as follows:

Month	Credit Amount	Number of credits	...	Credit
6	1169	2	...	1
48	5951	1	...	2
12	2096	1	...	1
9	2134	3	...	1

  For each of the above attributes, describe what transformations to the original dataset would need to occur for it to be usable in a Bernoulli Naive Bayes model. *(hint: every attribute must take on the value of 0 or 1)* (5 points)

- Restate the definition of Disparate Impact from lecture (also included in code comments); make sure to notate what each variable (e.g. S) represents. Why might this be a useful measure of model performance? What are some limitations of this measure? (5 points)
- A different way to think about fairness is based on the errors the model makes. We define the false positive rate (FPR) as $P(\hat{Y} = 1|Y = 0)$, and the false negative rate (FNR) as $P(\hat{Y} = 0|Y = 1)$. Suppose we calculate FPR and FNR for each group. In words, what does the false positive rate and false negative rate represent in the context of credit ratings? What are the implications if one group's FPR is much higher than the other's? What are the implications if one group's FNR is much higher than the other's? (6 points)

Grading Breakdown

We expect the accuracy that NaiveBayes reaches should be above 70%, on both training and testing. Note that your results will fluctuate each time you run the program, as there is some stochasticity in the preprocessing of the data. 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	60%
Report	40%
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 hw08` from the directory containing all of your source code.

Report

Please upload your report on Gradescope, rather than turning it in with your code.

Anonymous Grading

You need to be graded anonymously, so do not write your name anywhere on your handin.

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.