

CSE 575: Statistical Machine Learning

Density Estimation and Classification Project

Purpose

In this project, we will systematically implement and examine the three major categories of Machine Learning techniques of this course, including supervised learning, unsupervised learning, and deep learning. This project will be submitted and graded through Ed Lessons. Please follow the links in your course to access Ed Lessons and complete this project.

Objectives

Learners will be able to:

- Understand and implement supervised learning, unsupervised learning, and deep learning techniques in the context of density estimation and classification.
- Extract relevant features from the training dataset and estimate the parameters for a 2-D normal distribution for each digit.
- Utilize the estimated distributions to perform Naïve Bayes classification on the testing dataset.
- Implement the fundamental learning algorithm Naïve Bayes
- Report the classification accuracy for digits "0" and "1" in the testing set.

Technology Requirements

The specific algorithmic tasks you need to perform for this part of the project include:

1. Extracting the features and then estimating the parameters for the 2-D normal distribution for each digit, using the training data. Note: You will have two distributions, one for each digit.
2. Use the estimated distributions for doing Naïve Bayes classification on the testing data. Report the classification accuracy for both "0" and "1" in the testing set.

Algorithms:

- MLE Density Estimation, Naïve Bayes classification

Resources:

- You may go to the original MNIST dataset (available here <http://yann.lecun.com/exdb/mnist/>) or you can download the dataset file from the PDF (MNIST DATABASE.pdf -- located in Project Overview page of course) to extract the images for digit 0 and digit 1, to form the dataset for this project.

Workspace:

- Any Python programming environment
- Ed Lesson

Software:

- Python environment

Language(s):

- Python

Project Description

This project involves implementing supervised, unsupervised, and deep learning techniques for density estimation and classification. The project focuses on a subset of the MNIST dataset containing images of digits "0" and "1". The project involves four tasks: feature extraction, parameter calculation, implementation of Naïve Bayes classifiers, and prediction of labels for the test data using the classifiers. Finally, calculating the accuracy of the predictions.

Directions

Accessing Ed Lessons

You will complete and submit your work through Ed Lessons. Follow the directions to correctly access the provided workspace:

1. Go to the Canvas Assignment, "**Submission: Density Estimation and Classification Project**".
2. Click the "**Load Submission...in new window**" button.

3. Once in Ed Lesson, select the assignment titled "**Density Estimation and Classification Project**".
4. In the code challenge, first review the directions and resources provided in the description.
5. When ready, start working in the notebook titled "**project1.ipynb**".

Preparation

Access the link to your workspace through your Canvas course. You will be in the 'Project1' Jupyter notebook through Ed Lesson. As you run the code, you will load the trainset and testset for digit0 and digit1 respectively (Please read the code and you will understand). Both trainset and testset are sub-dataset from the MNIST dataset. The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. We use only a part of images for digit "0" and digit "1" in this question.

Therefore, we have the following statistics for the given dataset:

- Number of samples in the training set: "0": 5000 ;"1": 5000.
- Number of samples in the testing set: "0": 980; "1": 1135

We assume that the prior probabilities are the same ($P(Y=0) = P(Y=1) = 0.5$), although you may have noticed that these two digits have different numbers of samples in testing sets.

In the existing code, myID is a 4-digit string and please change this string to the last 4-digit of your own studentID; train0 is your trainset for digit0; train1 is your trainset for digit1; test0 is your testset for digit0; and test1 is your testset for digit1. They are all Numpy Arrays. You can also convert them into python arrays if you like.

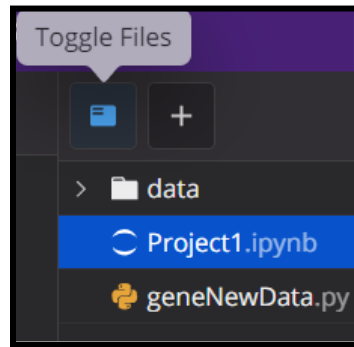
Other than the string named 'myID', **please do not** change any existing code and just write your own logic with the existing code.

You may go to the original MNIST dataset (available here <http://yann.lecun.com/exdb/mnist/>) to extract the images for digit 0 and digit 1, to form the dataset for this project. To ease your effort, we have also extracted the necessary images, and store them in ".mat" files. You may use the following piece of code to read the dataset:

- import scipy.io
- Numpyfile= scipy.io.loadmat('matlabfile.mat')

Files for you to download: “**CSE 575_Project 1 Mat Files**” (attached in the “Project Overview and Resources” page in the course)

The Ed notebook files and data can be downloaded by selecting the “**Toggle Files**” icon in the workspace (first option in the right corner).



Programming

For your own code logic, you have 4 tasks to do:

Task 1:

You need to first extract features from your original trainset in order to convert the original data arrays to 2-Dimensional data points.

You are required to extract the following two features for each image:

- **Feature1:** The average brightness of each image (average all pixel brightness values within a whole image array)
- **Feature2:** The standard deviation of the brightness of each image (standard deviation of all pixel brightness values within a whole image array)

We assume that these two features are independent and that each image is drawn from a normal distribution.

Task 2:

You need to calculate all the parameters for the two-class naive bayes classifiers respectively, based upon the 2-D data points you generated in Task 1 (In total, you should have 8 parameters).

- (No.1) Mean of feature1 for digit0

- (No.2) Variance of feature1 for digit0
- (No.3) Mean of feature2 for digit0
- (No.4) Variance of feature2 for digit0
- (No.5) Mean of feature1 for digit1
- (No.6) Variance of feature1 for digit1
- (No.7) Mean of feature2 for digit1
- (No.8) Variance of feature2 for digit1

Task 3:

Since you get the NB classifiers' parameters from Task 2, you need to implement their calculation formula according to their Mathematical Expressions. Then you use your implemented classifiers to classify/predict all the unknown labels of newly coming data points (your test data points converted from your original testset for both digit0 and digit1). Thus, in this task, you need to work with the testset for digit0 and digit1 (2 Numpy Arrays: test0 and test1 mentioned above) and you need to predict all the labels of them.

Note: Remember to first convert your original 2 test data arrays (test0 and test1) into 2-D data points as exactly the same way you did in Task 1.

Task 4:

In Task 3 you successfully predicted the labels for all the test data, now you need to calculate the accuracy of your predictions for testset for both digit0 and digit1 respectively.

Preparing the Deliverables

Results Submission & Output:

Submitting your work through Ed Lessons will create your results submission. As the result from your Notebook of Project 1, you should have your ASUId(string), 8 components for computed parameters and 2 components for accuracy. The order of these 11 components should be a list and look like the following:

```
[ 'ASUId', Mean_of_feature1_for_digit0, Variance_of_feature1_for_digit0,
Mean_of_feature2_for_digit0, Variance_of_feature2_for_digit0 ,
```

Mean_of_feature1_for_digit1, Variance_of_feature1_for_digit1,

Mean_of_feature2_for_digit1, Variance_of_feature2_for_digit1,

Accuracy_for_digit0testset, Accuracy_for_digit1testset]

Report Submission

Draft a report to go with your Results Submission. The report must contain:

- Your full name and student ID number on the first page in the upper left corner
- A detailed description of your observations and analysis of the project

The report must also follow the required format:

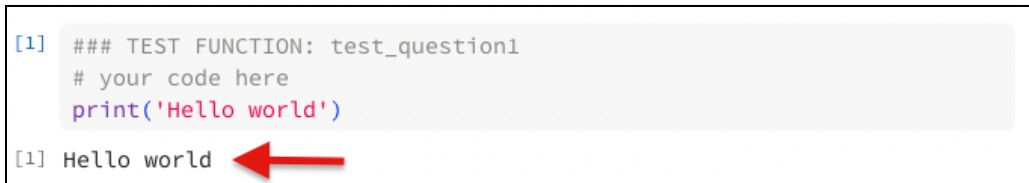
- A maximum font size of 12pt
- A maximum length of two (2) pages (8x11 or A4 paper).
- Saved as a PDF (.pdf) file type

Submission Directions for Project Deliverables

Result Submission

This assignment will be auto-graded. You must complete and submit your work through Ed Lesson's code challenges to receive credit for the course:

1. In order for your answers to be correctly registered in the system, you must place the code for your answers in the cell indicated for each question.
 - a. You should submit the assignment with the output of the code in the cell's display area. The display area should contain only your answer to the question with no extraneous information, or else the answer may not be picked up correctly.
 - b. Each cell that is going to be graded has a set of comment lines (ex: `### TEST FUNCTION: test_question1`) at the beginning of the cell. **This line is extremely important and must not be modified or removed.**
2. After completing the notebook, run each code cell individually or click **"Run All"** at the top to print the outputs.



```
[1] ### TEST FUNCTION: test_question1
    # your code here
    print('Hello world')

[1] Hello world
```

A red arrow points to the output 'Hello world'.

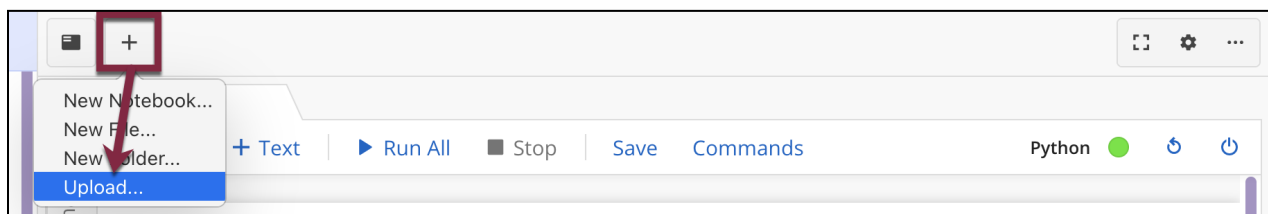
3. When you are ready to submit your completed work, click on **“Test”** at the bottom right of the screen.
4. You will know you have successfully completed the assignment when feedback appears for each test case with a score.
5. If needed: to resubmit the assignment in Ed Lesson
 - a. Edit your work in the notebook
 - b. Run the code cells again
 - c. Click **“Test”** at the bottom of the screen

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Report Submission

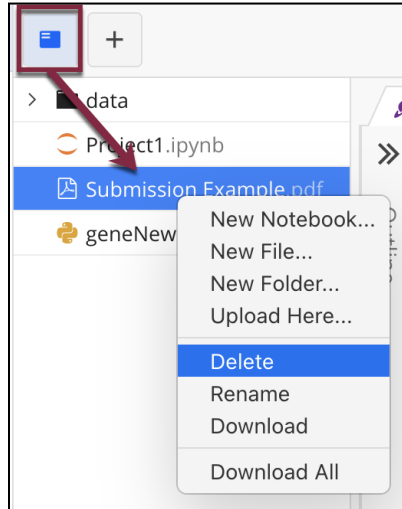
Your report will be manually graded by the course team. You must submit your report in the designated code challenge workspace where you also submitted your results submission.

1. Click the **Plus (+)** icon in the upper left corner of the notebook workspace (second icon from the left)
2. Select **“Upload”**
3. Locate and select your report submission from your device (PDF file only)



4. Your file will appear in a left-pane menu that appears next to the notebook workspace
5. Click **“Submit”** in the upper right corner to submit your completed project.
6. If needed: to resubmit the report in Ed Lesson

- Click the “**Toggle Files**” icon in the upper left corner of the notebook (first icon from the left)
- Locate and right-click on your previous report submission file
- Click “**Delete**” to remove it from your attempt and then repeat the upload directions from Step 2



Your latest report submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Evaluation

For both component combined, there are one hundred (100) points available for this project.

Result Submissions

The results submission is worth 60 points. The results are auto-graded and will be evaluated on:

- 10 points - Mean and variance of feature1 for digit0
- 10 points - Mean and variance of feature2 for digit0
- 10 points - Mean and variance of feature1 for digit1
- 10 points - Mean and variance of feature2 for digit1
- 10 points - Predicting new labels for digit0testset and calculating the accuracy.
- 10 points - Predicting new labels for digit1testset and calculating the accuracy.

Note: The **acceptable** range for parameters is $[x-0.2, x+0.2]$; The **acceptable** range for accuracy is $[x-0.005, x+0.005]$. It means that if one of your float-number answers falls into its corresponding range, your answer will be graded as correct. No, otherwise.

Report Submission

The report submission is worth forty (40) points. The report will be manually graded using a rubric and evaluated on:

- 10 points - Analysis is present
- 20 points - Correct solution with no errors and documentation
- 10 points - Successful run with no errors