

Rules:

You NEED access to a computer with Internet access and Python installed. You NEED all Python packages used in the course exercises (NumPy, Matplotlib).

You are allowed to use Internet and all course materials. You are NOT allowed to communicate with anyone in any form during the exam.

You MUST join the course Slack channel during the exam and you can send private messages to the course instructor.

All code and text MUST be written by yourself. Copying substantial parts from any source is PLAGIARISM.

Online test has multiple deadlines. If you MISS one, then the rest of them are not evaluated. DO NOT re-submit files for the previous deadlines as it resets their time tags and your test is over.

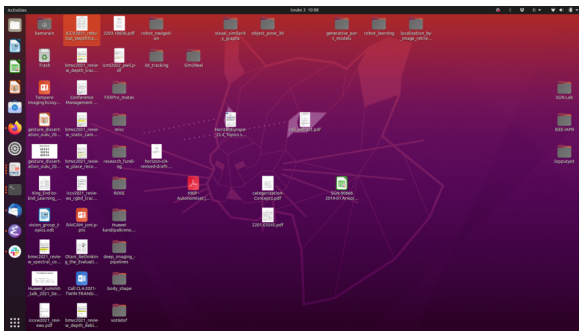
1. Basics (Opts) [Return before 10am25]

Take two screenshots of your *full* desktop:

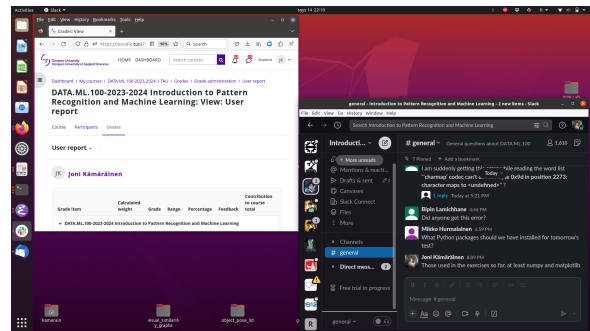
- An empty desktop where all applications are minimized (see Fig. 1(a))
- Open your grading page in Moodle DATA.ML.100 (your name must be visible) and the course Slack (latest message must be visible) (see Fig. 1(b))

Submit the following items before the deadline:

- Screenshot 1 as PNG file: surname_desktop1.png
- Screenshot 2 as PNG file: surname_desktop2.png



(a) Empty desktop



(b) Moodle and Slack open

Figure 1: Examples of the screenshots that should be returned

2. Data histograms (2pts) [Return before 10am45]

Download the training samples file

- X_train.txt

and the target values file

- y_train.txt

You can use `numpy.loadtxt()` function to load the data files.

Your training data contains measurements of the male and female height (1st column) and weight (2nd column). The target (ground-truth) vector contains the class labels (0: male, 1: female).

Plot the height histograms of the both classes into the same histogram plot. Make sure the histograms are clearly distinguishable from each other.

Show class labels in the histogram plot (legend) and name the y- and x-axes correctly.

All code should be in a single Python program which can be run as:

```
(dataml100) Joni$ python kamarainen_histogram.py
(dataml100) Joni$
```

Submitted items:

- Full desktop screenshot of running your code and printing/plotting outputs (+Moodle grading view +histogram plot windows):
`surname_histogram_screenshot.png`
- Python source code: `surname_histogram.py`

3. Bayes classifier [Multiple deadlines]

In addition to the training data, download also the test files

- `X_test.txt`
- `y_test.txt`

We use only the first column that is the person height. The class labels are 0 for male and 1 for female.

You write a single Python file that is extended for each new deadline.

(a) Baseline classifier (3pts) [Return before 11am00]

Assign all test samples either the male (0) or female (1) label and print the correct classification percentage for the both case.

(b) Gaussian pdf (10pts) [Return before 11am20]

Write a Python function for the Gaussian probability density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (1)$$

Use the function to plot the class-specific likelihoods for all training male and female samples (black circles for male and red circles for female). Use male distribution for the male samples and female for the female samples. X-axis is height and y-axis is the computed likelihood.

Note: NumPy functions `mean()` and `std()` can be used.

(c) Bayes classifier (10pts) [Return before 11am45]

Using the Gaussian pdf compute the bayes posterior probabilities for all test samples, classify them according to the probabilities and print the correct classification percentage.

Submitted items:

- Full desktop screenshot of running your code and printing/plotting outputs (+Moodle grading view +results output):
surname_baseline_screenshot.png
- Full desktop screenshot of running your code and printing/plotting outputs (+Moodle grading view +plot window):
surname_gaussian_screenshot.png
- Full desktop screenshot of running your code and printing/plotting outputs (+Moodle grading view +results output):
surname_bayes_screenshot.png
- Python source code: surname_all.py