Machine Learning (HWS24)
Assignment 1: Naive Bayes

The archive provided to you contains this assignment description, a datasets in binary format, as well as Python code fragments for you to complete. Comments and documentation in the code provide further information. Please note the following:

- It **suffices to fill out the "holes" that are marked in the code fragments** provided to you, but feel free to modify the code to your liking. The aim of the assignments is to familiarize you with NumPy, so use NumPy over pure Python and try to keep your implementation efficient.
- Please **adhere to the following guidelines** in all the assignments. If you do not follow those guidelines, we may grade your solution as a FAIL. Provide a single ZIP archive with name `ml24-a0<assignment number>-<your ILIAS login>.zip`. The archive needs to contain:
  - A **single PDF report** that contains answers to the tasks specified in the assignment, including helpful figures and a high-level description of your approach. **Do not simply convert your Jupyter notebook to a PDF!** Write a separate document, stay focused and brief. You report **must not exceed 8 pages, excluding references and appendix**. You must use the seminar template from this link for your report (including the final signature page).
  - All **the code that you created** and used in its original format.
  - A **PDF document that renders your Jupyter notebook with all figures.** (If you don't use Jupyter, then you obviously do not need to provide this.)
- **A high quality report is required to achieve an EXCELLENT grade.** Such a report is self-explanatory (i.e. do not refer to your code except for implementation-only tasks), follows good scientific practice (e.g. when using images, tables or citations), does not include hand-written notes, and does not exceed 8 pages. In addition, label all figures (and refer to figure labels in your write-up), include references if you used additional sources or material, and use the tasks numbers of the assignments as your section and subsection numbers.
- You **may work on this assignment in pairs**—i.e, with one (and only one) additional student—and then hand in a pair submission. To do so:
  - The PDF report and notebook must clearly report then **name and ILIAS login** of **both students** right at the beginning.
  - **Both students must submit** the same assignment on ILIAS separately.
  - You may change whether or not you submit in a pair and with whom from assignment to assignment.

  To be clear, if only one student of the pair submits the assignment, then only this student will receive the grade.
- Hand-in your solution via ILIAS until the date specified there. This is a hard deadline.

Report. The following list of tasks are pure implementation tasks:

Task 1, Task 2, Task 5a

You do *not* need to discuss these tasks in your report.

MNIST Dataset

We will use a preprocessed variant of the MNIST digits dataset in this assignment. The task is to classify hand-written digits. There is one class for each digit (i.e., classes $0, 1, 2, ..., 9$). The features represent a scanned image (28×28 pixels, values in $\{0, 1, \ldots, 255\}$). The dataset contains both training data ($\approx$ 6000 images per class) and test data ($\approx$ 1000 images per class). See http://yann.lecun.com/exdb/mnist/ for more information.

1 Training

Provide a function `nb_train` that trains a Naive Bayes classifier for categorical data using a symmetric Dirichlet prior and MAP parameter estimates. A description of the parameters and expected result can be found in the Python file. For example, you may assume that all features take values in $\{0, 1, \ldots, K-1\}$, where K is the number of possible values.

Hints. First, try out your function with a uniform Dirichlet prior ($\alpha = 1$) on the small example provided by us. Then try it with add-one smoothing ($\alpha = 2$). If both results are correct, train your model on the MNIST digits dataset. You may need to work with a sample of the training data (the source code to do this is provided). Can you get your implementation to train the model on the full dataset in a reasonable time of, say, a few seconds? If not, don't worry and continue with a sample of the data.

2 Prediction

Provide a function `nb_predict` that takes your model and a set of examples, and outputs the most likely label for each example as well as the log probability (confidence) of that label. A description of the parameters and expected result can be found in the Python file.

Hints. Again, start with our small example, check that your implementation is correct, then try it on the digits dataset.

3 Experiments on MNIST digits data

a) Train your model with $\alpha = 2$ on the MNIST training dataset, then predict the labels of the MNIST test data using your model. What is accuracy ($= 1 -$ misclassification rate) of your model?

b) Plot some test digits for each predicted class label (code provided). Can you spot errors? Then plot some misclassified test digits for each predicted class label (code provided). Finally, compute the confusion matrix (https://en.wikipedia.org/wiki/Confusion_matrix, code provided). Discuss the errors the model makes.

Note. In your report, document your experiments (i.e., training process, metrics used)

4 Model selection (optional)

Use cross-validation to find a suitable value of the hyperparameter α (of the symmetric Dirichlet prior). Also plot the accuracy (as estimated via cross-validation) as a function of α. Discuss.

Note. Some hints to get you started with running cross-validation are given in the Python file.

5 Generating data

a) Implement a function `nb_generate` that generates digits for a given class label. A description of the parameters and expected result can be found in the Python file.

b) Generate some digits of each class for your trained model and plot (code provided). Interpret the result. Repeat data generation for different models by varying the hyperparameter α. How does α influence the results? Discuss.

6 Missing Data

Consider a classification problem with D discrete features, each taking values in $\{0, \ldots, K-1\}$, and C classes $0, \ldots, C-1$. For the MNIST dataset, we had $D = 784$, $K = 256$, and $C = 10$.

We make the Naive Bayes assumption and are given distributions $p(y)$ and $p(x_j|y)$ for $1 \leq j \leq D$. Using solely these distributions, give an **as simple as possible** formula for each of the following distributions and state why/when it can be useful

a) $p(y|x_{1:D})$

b) $p(y|x_{1:D'})$ for $1 \leq D' < D$.

c) $p(x_{D'+1:D}|x_{1:D'})$ for $1 \leq D' < D$.

Optional: Use the Naive Bayes model that you created in the previous tasks to sample from the above distributions. Experiment with some example images and discuss the result.