Machine Learning Project, 2023/2024

Margarida Silveira, Catarina Barata and Jorge Marques

1 Introduction

The Machine Learning project should be done in groups of 2 students, using the Python programming language.

The project is split into 2 parts (regression and image classification) and each part comprises 2 questions with deliverables for evaluation.

The programming language used in the project is Python because it has powerful libraries for building Machine Learning applications and it is a widespread language in industry. The first problem session in the Machine Learning course is devoted to the basics of programming in Python.

2 Student Evaluation

Student evaluation will take into account:

- statistical performance of the four algorithms developed by the group, evaluated on an independent data set (25%);
- final report (maximum length of 10 pages, font size 12 pt, including cover and table of contents) describing the methodologies adopted by the group, including figures and statistical evaluation and discussion of the results. Participation in the lab sessions will also be taken into account (75%).

Each group should work independently. Consultation with other people and exchange of ideas or software are not allowed and may invalidate the work. The use of tools such as ChatGPT or CoPilot is not allowed. Each group member will be assessed individually, and both students must actively solve the project and be acquainted with the work. Instructors will only answer clarification questions, but all the exercises need to be fully solved by the students.

Each group must submit the output of the proposed algorithms using an independent data set (test set) for each of the four questions until the end of the deadlines, as well as the Python code used to solve them. The outputs will be compared with the ground truth by the teaching team and the results (a leaderboard with the scores achieved by each group) will be published on the Fenix web page. Attendance to the laboratory sessions is mandatory and submissions from groups who fail to do so **will not be evaluated**.

3 Datasets and Project Submissions

The training and test data for each of the project questions will be made available by the teaching team, through the course web page on Fenix. For each question, the students will have access to a training set (feature vectors and real outputs) and a test set (just the feature vectors).

All data will be stored in *numpy* (.npy) format.

The students must implement and train their machine learning approaches using the training set. There are no restrictions regarding the number of machine learning models that the students

can research and try but the minimum is two models to be discussed in the report. However, in each of the project questions they **must pick only one model** to apply to the test set and perform the submission.

Project submissions should be made through Fenix, in the appropriate section. For each question, the students must submit a **zip** file containing: i) the output of their model of choice on the test set; and ii) the Python code. The predictions must respect the same format as that of the output within the training set.

The assessment of the performance on the test set will be carried out by the teaching team using appropriate statistical metrics. The scores achieved by each group will be made available on the course webpage.

4 Part 1 - Regression with Synthetic Data

The first part is devoted to regression analysis using synthetic data.

4.1 First Problem - Linear Regression

The first problem is a basic linear regression problem.

Consider a training set with $n = 15$ examples

$$\mathcal{T}_{train} = \left\{ (x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \right\},$$

where each example comprises a feature vector, $x^{(i)} \in \mathbb{R}^{10}$, with 10 features, and an outcome $y^{(i)} \in \mathbb{R}$, for $i = 1, \ldots, n$.

We wish to train a linear predictor

$$\hat{y} = f(x) = [1 \; x^T]\beta.$$

You may minimize the Sum of Squared Errors (SSE) criterion or adopt a regularization technique such as Ridge or Lasso. Try to see what makes more sense for the specific data you are considering.

To evaluate the performance, the estimated predictor should be applied to an independent set of data (test set) with $n' = 1000$ examples

$$\mathcal{T}_{test} = \left\{ (x^{(1)}, y^{(1)}), \ldots, (x^{(n')}, y^{(n')}) \right\},$$

provided on the Fenix web page. The students should predict the outcome $\hat{y}^{(i)}, i = 1, \ldots, n'$ for each test example and send this information to the teaching team through the Fenix platform. The comparison between the predictions $\hat{y}^{(i)}$ and the true values $y^{(i)}$ will be done by the teaching team since the test outcomes will not be given to the students. The metric used by the teaching team to evaluate the submissions will be the **SSE**.

4.2 Second Problem - Linear Regression with Two Models

The second problem is similar to the previous one but the available data is generated by two linear models. This means that some examples are generated by one probabilistic model and the other

examples are generated by another model. You don't know which examples are generated by the first or second models, but there is not much disproportion.

You should devise a method to estimate the target values for both types of instances, using the training data. Of course, you may apply the same method adopted in the previous problem (where data was generated by a single model) but this will lead to bad prediction results because you are trying to apply the same prediction model to both types of examples.

In fact, the training set is given by

$$\mathcal{T}_{train} = \left\{ (x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \right\},$$

where $n = 100$ and $x^{(i)} \in \mathbb{R}^4$. As stated earlier, we do not know which examples are associated with the first or second model.

When it comes to evaluating the model, you should provide two predictions for each test example, obtained with model 1 and model 2: $y^{(i)}_{M1}, y^{(i)}_{M2}$ and send this information to the teaching team through the Fenix platform, using two columns. The teaching team knows which model is active for each test example and will consider two hypotheses (H1: first predictor corresponds to model 1 and H2: first predictor corresponds to model 2). The best hypothesis will be chosen.

The metric used by the teaching team to evaluate the submissions will be the **SSE**.

4.3 Suggestions

- try to implement the linear predictor using vectors, matrices and algebraic operations available in the *numpy* package;
- check the linear regression examples available in the scikit-learn[1] package;
- visualize the outcomes and prediction errors of the developed models.

5 Part 2 - Image Analysis

The second part is devoted to the analysis of 2D medical images. We will use real data from MEDMNIST (https://medmnist.com/) but with some adaptations. There are two classification tasks.

5.1 First Task

The first classification task is a binary one, where we want to create a model that predicts whether a dermoscopy image is from a melanoma or a nevu. For this task the label is either 0 (nevu) or 1 (melanoma), as illustrated in Figure 1. The images are 28x28 pixels and have 3 color channels (RGB), thus each input has 2352 elements (28x28x3). Note that the dataset is imbalanced since there is a big difference in the number of melanomas and nevus images in our training dataset. The metric used by the teaching team to evaluate the submissions for this task will be **Balanced Accuracy**, which is defined as the average of precision and recall obtained in each class.

The test set will contain samples from two distinct sources. Part of the test set is from the same hospital as the training data and part is from a different hospital. The evaluation score will

[1] https://scikit-learn.org/stable/

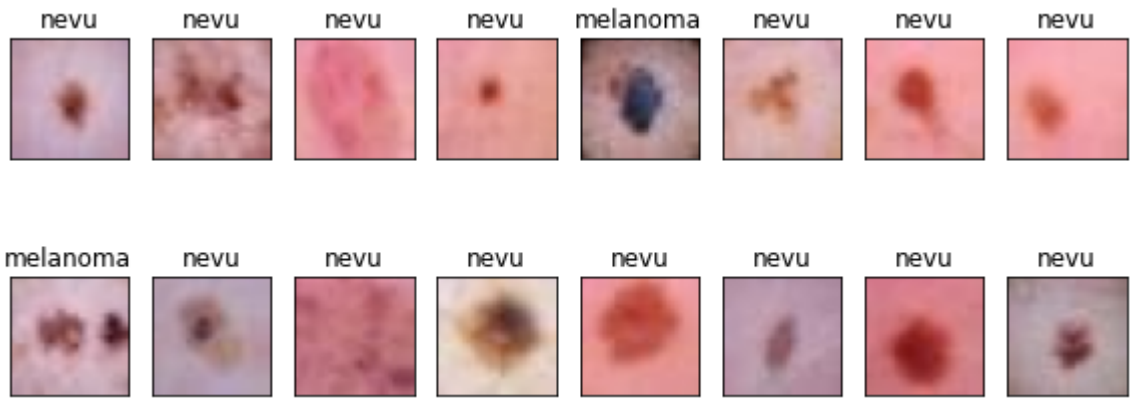

Figure 1: Dermoscopy images used for the first classification task.

be computed only for the first part but the scores obtained for the second part will also be given, so they can be analyzed and discussed in the report.

5.1.1 Suggestions

- investigate which are the most suitable classifiers for image tasks;
- investigate ways to deal with imbalance in classification tasks;
- compute and analyze different performance metrics.

5.2 Second Task

The second task also consists of the classification of 2D medical images but this time the images have six possible classes and come from two distinct datasets: dermoscopy and blood cell microscopy. The class labels are the following, 0 (nevu), 1 (melanoma), 2 (vascular lesions), 3 (granulocytes), 4 (basophils), and 5 (lymphocytes), where the first three classes come from the dermoscopy dataset and the latter from the blood cell one. These are illustrated in Figure 2. The classifier only needs to predict the class label, not the dataset label. As in the previous task, the image matrices are 28x28x3 but they are given as vectors.

Note that this dataset is also imbalanced, since there is a significant difference in the number of training patterns from the six different classes.

The metric used by the teaching team to evaluate the submissions for this task will be **Balanced Accuracy**.

5.2.1 Suggestions

- use the information about the dataset label if you think it can be helpful;
- compute different metrics and compare the performance for the different classes.

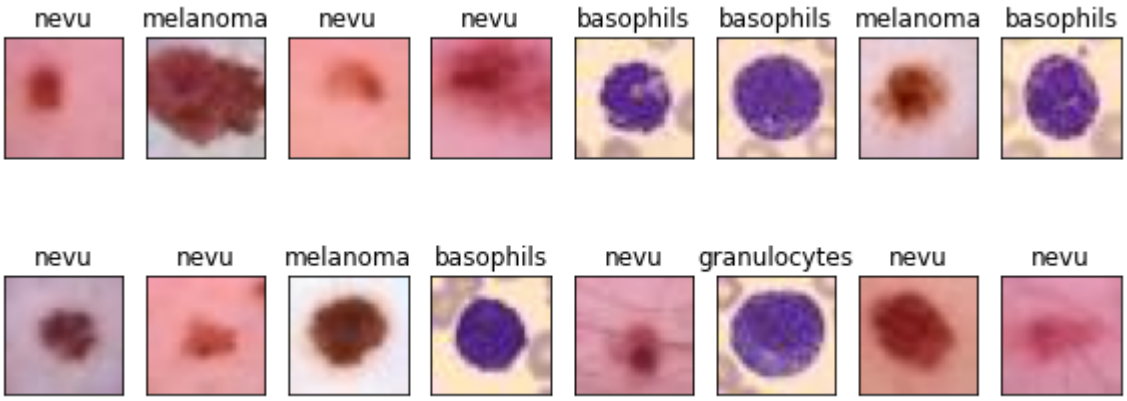

Figure 2: Dermoscopy and blood cell images used for the second classification task.