Machine Learning Project, 2022/2023

Prepared by Prof. Margarida Silveira, Prof. Catarina Barata and Prof. 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 analysis) 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.

Each group should work alone. Consultation of other people, as well as exchange of ideas or software, are not allowed and may invalidate the work. Instructors will only answer clarification questions, but all the exercises need to be fully solved by the students.

2 Student Evaluation

Student evaluation will take into account:

- interaction with the teacher during the laboratory sessions - this is an individual assessment of each group member, thus both students must actively solve the project and be acquainted with the work;
- statistical performance of the algorithms developed by the group, evaluated on an independent data set;
- final report (maximum length of 10 pages, font size 12 pt) describing the methodologies adopted by the group, including figures and statistical evaluation.

Each group should 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 webpage 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. However, in each of the project questions they **must pick only one** 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 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 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

4.1 First Problem - Basic Linear Regression

The first problem is a basic linear regression problem.

Consider a training set with $n = 100$ 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

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

using the sum of squared errors (SSE) criterion, computed in the training set.

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 in Fenix web page. The students should predict the outcome for each test example $\hat{y}^{(i)}, i = 1, \ldots, n'$ 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 the be the SSE.

4.2 Second Problem - Linear Regression with Outliers

The second problem is similar to the previous one but it has a major difference: some of the training examples (about 20%) are *outliers*. This means that these examples are not generated by the same probabilistic model used to generate the other data and the predictor will fail to predict the outcome in such examples.

The goal of this problem is to devise a method to estimate a predictor in the presence of outliers in the training data. Of course, you may apply the same method adopted in the previous problem (where there were no outliers) but this will probably 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

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

as before, but we do not know which examples are *good* and which are *not*.

When it comes to evaluating the model, we assume that the test set, $\mathcal{T}_{test}$, **does not have any outliers**. This means that in the testing phase we know which examples are good and which are bad, and we are only interested in characterizing the performance of the predictor in the inliers.

The students should predict the outcome for each test example $\hat{y}^{(i)}, i = 1, \ldots, n'$ and send this information to the teaching team through the Fenix platform. The metric used by the teaching team to evaluate the submissions will the be SSE.

4.3 Suggestions

- read the slides (linear regression Chapters 1,2,3);
- 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 butterfly images. Our goal is to classify and segment two types of patterns that develop in the wings of these butterflies: the spots and the eyespots. Spots are patterns that develop a single spot of color. Eyespots are patterns that develop spots and rings of color. Both types are illustrated in Figure 1.

The detection of the two types of patterns in the butterfly images is outside the scope of this work. This was done automatically for a large number of butterflies of different species using a well known object detection method known as YOLO [1]. After detection, the patterns bounding boxes have been resized to a common size of 30x30 pixels.

5.1 First Task

The first classification task is a binary one, where we want to create a model that predicts the type of pattern that has been detected. For this task the label is either 0 (spot) or 1 (eyespot). Since the images are in color, the patterns have 3 color channels (RGB), thus each input is 30x30x3. Note that the dataset is imbalanced, since there is a big difference in the number of spots and eyespots in our dataset. The metric used by the teaching team to evaluate the submissions for this task will be the F_1 score.

5.1.1 Suggestions

- investigate which are the most suitable classifiers for image tasks;
- investigate ways to deal with imbalance in classification tasks.

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

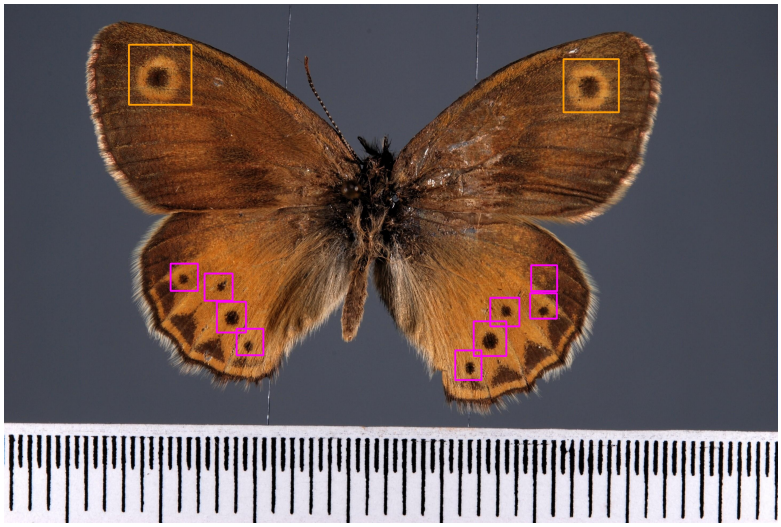

Figure 1: Image of a butterfly where eyespots are shown in orange and spots are shown in pink.

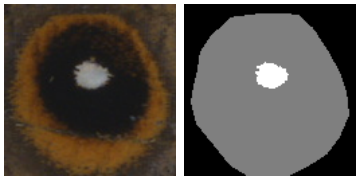

Figure 2: An eyespot image and its ground truth segmentation mask.

5.2 Second Task

The second task consists in the segmentation of a particular type of eyespot that develops in butterflies of the bicyclus anynana species. These eyespots have a white center, surrounded by a black ring and then a golden ring. Our goal is to segment three distinct areas: 1) the white ring, 2) the black and gold ring combined, and 3) the background:

This task can be regarded as a pixel classification task. In this case we want to classify each pixel in a 30x30 RGB eyespot image using as features the set of pixels in a 5x5 neighborhood surrounding it. The classification is a multiclass problem for which the label is an integer from 0 to 2, denoting background, rings, and white center, respectively. For this task, the segmentation masks used for training have been created by manually drawing the three areas and assigning the desired label to all the pixels in the same region, as illustrated in Figure 2.

Note that this dataset is also imbalanced, since there is a big difference in the number of pixels in the three areas of the eyespots.

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

5.2.1 Suggestions

- do not use thresholding or circle detection methods, and instead use the classification methods learned in class;
- check functions **extract_patches_2d** and **reconstruct_from_patches_2d** from the skimage toolbox.

References

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788