University of Macedonia

Department of Applied Informatics

7Thesemester | Academic Year: 2022 – 2023

Machine Learning

1theWork – Regression problems

You are asked to write code in Python in which:

1st part: state approximation assuming noise in the measurements and knowledge of the parametric model (knowledge of equation form, unknown values of the coefficients in the terms).

1. You will generate 150 random real numbers that follow a uniform distribution. Make sure the numbers are in ascending order. All numbers must belong to the interval [-4, 4].
2. You will define the following function: $= {}_{1}\cdot \frac{1}{\exp(\ )} + {}_{2}\cdot \sin(\ )$, named function myCustFunc().
   The function myCustFunc(x, l1, l2) must accept as an argument an arrangement of numbers like that of question 1 and the values for the parameters λ1and l2. It will return an array of the same size as the input.
3. You will create one (1) new array of size 150 x 1, giving values for l1,l2and the vector x obtained from question 1.
4. Introduce noise to the output data that follows a distribution of your choice [*except* (0,1)]. Describe the allocation you chose, in the report you will submit

5. Create a plot showing the [x_input, y_output] [x_input, y_noisy] data you created in questions 1, 3 and 4. The outputs of the functions should be differentiated: use different colors and symbols. Don't forget to include a title, axis names, and a caption so we know which function generated the data.

6. You will fit the parameters to the data**with noise**, i.e. [x_input, y_noisy], of myCustFunc, using scipy.optimize.curve_fit.
7. You will define a new function, named poly4thDegree(x, a, b, c, d, e): a polynomial 4u degree. Caution: the function should be defined in such a way that it can be used by curve_fit() to estimate the parameters.
8. Using the set of values you have available, [x_input, y_noisy], approximate the parameters of the polynomial.

9. You will make 2 different plots showing the predicted values (y_predicted) when using myCustFunc (query 2) and poly4thDegree (query 7).
    a. At 1Thethe values of myCustFunc with the parameters obtained from query 6 will be shown.
    b. In 2Theyou will do exactly the same for the poly4thDegree function.
10. You will calculate the statistical errors mean absolute error and root mean squared error, for each of the following combinations:
    a. original_values - noisy_values
    b. noisy_values - myCustFunc (parameters from query 6)
    c. noisy_values - poly4thDegree (parameters from query 8)

    The above errors should appear on the screen along with a sentence explaining what the numbers just printed refer to.

    How good an approximation does the polynomial give you for the given problem?

2nd part: state approximation based only on the measurements you have taken (you assume there is noise in the measurements, you have no knowledge of the model generating the data).

1. Use the 150 values, *with noise*, that you generated in 1Thepart.
2. Divide the available data, [x_input, y_noisy], into three separate subsets: train, validation, and test.
3. Train three different regressors (eg kNN, SVR, etc.) of your choice on the train set.

4. Evaluate the performance of the regressors on the test set and create appropriate graphs to show the results.
5. Repeat steps 2 to 4, but this time normalize the input values to the interval [0,1]

You will collect all the results and submit them in a report format. Information on the structure of the report can be found in the "Instructions" section.

Instructions:

A. Assignments are in groups of up to four (4) people. Each person can submit work to only one group at a time.

B. Assignments should be uploaded to eClass in a zip (not rar) file by the deadline. No extension will be granted.**Caution**: Each group assignment will be submitted by only one group member (you choose who/who).

C. Each assignment must be accompanied by:

- One and only one file.py will contain the answers to the queries
- One**report**inpdf with the following information: o
    Cover: 1 page, includes the details of the students in the group, course name, date, department and other relevant details.
    - Summary table of contents, images, and other graphics that cite in the report.
    - Introduction section: 1 page, describe the problem (*without* copying exactly the speech of the exercise)
    - Methods applied: from 2 to 10 pages, describe the methods you used and list the relevant results. Make sure it's clear which question you're referring to.
    - Conclusions: 1 page, based on the results what do you recommend, which model performs better, what could be done to further improve the performance.
    - The report will contain graphical representations of all kinds and tables of evaluation of the results that must be accompanied (each) by at least one paragraph with commentary.

Make sure that:

- The code must necessarily be accompanied by appropriate comments.
- An editorial and spelling check has been done on the report you will submit.
- The sentences should be understandable and short in length.
- Images to***do not***have arisen fromprint screen. If the program creates an image, save it normally (jpg or png) before using it.
- The graphs should include names on the axes and a legend. Purpose is to understand what it shows, at a glance.
- If something is not specified, you have the right to make any implementation that suits you. Make sure you can explain exactly what you did in the code.
- The libraries you will use *must* be able to be installed via it pip.

- The code *must* run onGoogle Colab.

<u>Delivery Due Date:</u> **December 12, 2022**. No extension will be granted.