MINOR SMART EMBEDDED

MACHINE LEARNING MINI PROJECT

THE INDIVIDUAL RESEARCH PROJECT

In this Machine Learning course of Smart Embedded you have been introduced to several different classification algorithms. As you have seen each of these methods is easy to program with an API in Python. Now it is time to compare them on a dataset of your own choice. It a known fact that the best algorithm to apply to a dataset depends very much on the particular data set you are analysing.

STEP 1 – FIND A DATA SET AND RESEARCH QUESTION

Start by looking for a labelled dataset with pictures on which you can apply supervised learning. As a guideline opt for a dataset that has at least 500 pictures. There is a lot of stuff out there on the Internet. Try to find something that has your interest. Be orginal.

Make your interest specific with a concrete research question or a hypothesis that can be answered/tested with a numeric analysis. Write down a strategy (algorithms) on how you think you could answer the question with ML.

Upload your proposal under Canvas Assignment “dataset proposal + plan”. Your teacher will assess the proposal (e.g. to check if it is not too difficult/easy) and give a go/no-go.

STEP 2 – DESCRIBE THE DATA

Make a good description of the data and the problem that you are going to solve with supervised machine learning. That is good practice anyway, and it helps you to set-up the data analysis.

Include at least the following items in your description:

- Name of the dataset.
- Link (URL) to the dataset.
- Domain of the dataset (e.g. medical, customer & sales, science, social study, online click behaviour, gaming, finance & economy, etc.)

If necessary, clean the data and describe your strategy for cleaning.

STEP 3 – EXPLORE THE DATA

Investigate and explore the data.

- Plot multiple graphs (boxplots, scatterplots) and
- draw initial conclusions from this (a.k.a. initial hypothesis).

STEP 4 – DO THE ANALYSIS

First split the data into a test and train set. Hold out the test set for checking the accuracy at the end (e.g. 20-30% of the number of samples). Ideally, you find optimal parameter settings for two (or more) classification algorithms. Use cross-validation (K=5 is a typical value) to 'prove' that you have prevented overfitting. Below you find some specific indications for the parameter variations per algorithm that we want you to do.

Compare achieved accuracies on the test set (and precision or recall if applicable in your case).

Tip 1: Use a fixed random state when instantiating classifier objects or calling functions.
That makes the analysis reproduceable and therefore easier to measure improvements.

Tip 2: Consider a "grid search" to find the optimal hyper parameters. Show the results in plots.

STEP 5 – COMPARE ALGORITHMS / APPROACHES

Compare the performance of multiple classification algorithms on the object detection problem.
An example is shown in table 1 with so-called ensemble learning.

	SVM	MLP	Random Forest	Majority Vote	Best Algo (= MLP)	Ensemble Learning	Class Label
1	1	2	2	2		2	1
2	1	3	1	1		1	1
3	2	2	2	2		2	2
4	3	2	1	-	2	2	2
5	1	1	1	1		1	1
6	2	2	2	2		2	2
7	1	3	2	-	3	3	3
8	…	…	…	…	…	…	…

Table 1: Ensemble learning for a 3-class problem. Note that for the first 7 samples ensemble learning has only 1 misclassification, SVM and MLP both 2, and random forest 3.

Note that in this fictitious example ensemble learning leads to an overall improvement in accuracy.

STEP 6 – REPORT YOUR FINDINGS

Report your findings in Jupyter notebook format.

Apply the principle of reproducible research (provide explanation, code, and data).

This is best practice in data science / engineering. Comment on the results and the used algorithms. Post your Jupyter Notebook in the ML Canvas course under Assignment “ML portfolio assessment”.

The report will be the basis for the oral assessment of ML at the end of the course.

Tip: Check which of the learning goals of ML is covered in your project. Compensate the missing goals by expanding the project (or with other work in notebook format).