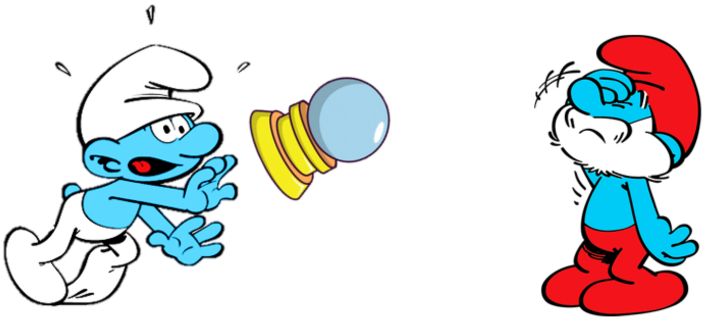
LELEC2870 - Machine Learning Project:
Heart failure on the rise in the Smurf society

Academic year 2023-2024

Introduction

In the once tranquil world of Smurfs, death was an unfamiliar concept. However, significant changes in their society have cast a shadow over their formerly flawless paradise, introducing them to cardiovascular diseases they had never encountered. To aid in diagnosing and treating these newfound health issues, Papa Smurf used an extraordinary crystal ball. He manipulated this enigmatic device in collaboration with Doctor Smurf to predict the chances of heart failure for a fellow Smurf within the next 10 years of his life. While the crystal ball seemed to give accurate predictions, they did not fully understand the root causes behind these new heart problems. Regardless, calamity struck when Clumsy Smurf inadvertently shattered the magical oracle. In their quest to recover a life-saving prediction tool, they have turned to their most trusted human friends, you. They hope to harness your kind of magic: machine learning.



Fortunately for you (and for them), Doctor Smurf has meticulously gathered data on his peers. Indeed, for every Smurf he assessed, he documented information regarding his lifestyle, nutrition habits, blood test results and some other basic health indicators. Also, using the latest smurf medical imaging technology, he managed to obtain 28 by 28 pixel heart scans! Of course he also took careful note of Papa Smurf's crystal ball predictions. Your aim is to take all this available information and train a machine learning model that produces predictions similar to those from the now shattered oracle. At the same time, you will try to formulate hypotheses on the causes of all these heart failures. This document provides guidelines to help you start this project, as well as a description of the data and details about the expected outcomes. Please take the time to read it carefully; a lot of little blue creatures are counting on you (no pressure ☺).

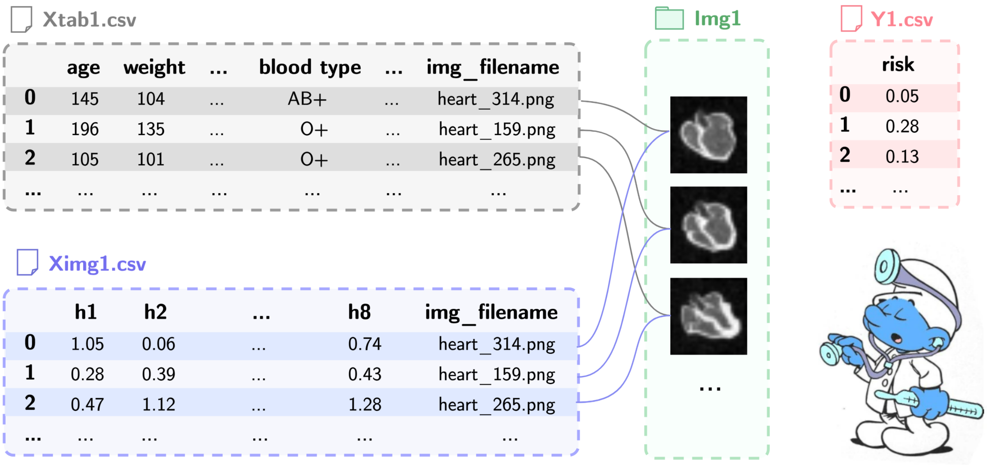
Dataset

To complete your task, you will need to work with several data files. These are available on Moodle. The file `Xtab1.csv` contains medical data stored as a table. Each line/row corresponds to a Smurf and each column to a measured attribute/variable/feature. A description of each variable is given bellow:

age	Age (can be well over 100 for Smurfs)
blood pressure	Systolic blood pressure (in mmHg)
blood type	Blood type (antigens and rhesus)
cholesterol	Level of LDL cholesterol (”bad cholesterol”) in blood (in mg/dL)
hemoglobin	Level of hemoglobin in blood (in g/dL)
physical activity	If the Smurf practices a physical activity on a regular basis (yes - no)
sarsaparilla	Consumption of sarsaparilla leaves (very low - low - moderate - high - very high)
smurfberry liquor	Consumption of smurfberry liquor (very low - low - moderate - high - very high)
smurfin donuts	Consumption of smurfin donuts (very low - low - moderate - high - very high)
temperature	Body temperature at the time of the visit by Doctor Smurf (in °C)
testosterone	Level of testosterone in blood (in ng/dL)
weight	Body mass of Smurf (in grams)

The risk of developping a heart failure within the next ten years is the target variable; it is stored in the `Y1.csv` file. The indices match those of `Xtab1.csv`.

The last element of each line in `Xtab1.csv` is the name of the image file that contains the heart scan. These images are stored in the folder `Img1`. For convenience, we provide you with ready-to-use image embeddings (i.e., low-dimensional vector representations of the images). These are stored in the `Ximg1.csv` file; each element of an embedding vector can be considered as an additional numerical feature. The image embeddings were obtained by training a convolutional neural network and extracting the output values from its last (fully connected) hidden layer. If you want to perform this feature extraction step on your own (for example, in order to improve it), some code is provided in `feature_extraction.ipynb` to help you out. Feel free to modify it in any way you like.



Xtab1.csv

	age	weight	...	blood type	...	img_filename
0	145	104	...	AB+	...	heart_314.png
1	196	135	...	O+	...	heart_159.png
2	105	101	...	O+	...	heart_265.png
...	...	...	...	...	...	...

Ximg1.csv

	h1	h2	...	h8	img_filename
0	1.05	0.06	...	0.74	heart_314.png
1	0.28	0.39	...	0.43	heart_159.png
2	0.47	1.12	...	1.28	heart_265.png
...	...	...	...	...	...

Y1.csv

	risk
0	0.05
1	0.28
2	0.13
...	...

Figure 1: Labeled dataset overview (Values in the illustrations do not correspond to the ones in the files).

The files `Xtab1.csv`, `Ximg1.csv` and `Y1.csv` form the labeled dataset on wich you will train your models and estimate the generalization performance. An overview of the labeled data is provided in Figure 1. You will find additional data in the files `Xtab2.csv` and `Ximg2.csv` (and folder `Img2`) for which you do not have the labels (but we do ☺). This second dataset will be used for the evaluation of your best model.

Instructions

You can work on the project individually or by groups of 2. First, you will have to write python code to complete the proposed regression task: the prediction of heart failure risk. Then, we ask you to write a report in which you summarize your work and explain your results. The goal is not really to find the absolute best regression model, but more to build a robust and sound machine learning pipeline. This pipeline will consist of three key components: data engineering, feature selection, and model selection. Note that while we present each of these components separately, they are not necessarily independent (e.g., feature selection can be part of model selection, data engineering and feature selection may overlap, etc).

Data engineering

Data engineering is the foundation of any successful machine learning project. Your task here is to prepare the dataset for analysis. This includes removing the features that are clearly useless for prediction, encoding the categorical variables such that they can be processed by your regression models, transforming numerical variables if needed (normalization, standardization, etc), handling missing data and outliers if any, etc. In this particular project, you will also have to think about how you combine the features extracted from the heart scans with other features.

Moreover, if you feel adventurous, you can try to modify the code from `feature_extraction.ipynb` to produce better image embeddings (i.e., that will lead to better regression performances). You may for example try to change the dimensionality of the embeddings, fine-tune the hyper-parameters of the given convolutional neural network, modify its architecture, or even try out something completely different. Note that this part is to be thought of as a bonus. You will not loose any point if you use the provided image embeddings as given.

Feature selection

This step involves choosing the most relevant and informative features from the dataset to improve the performance of the regression models. The goal is to eliminate irrelevant or redundant features, which can lead to overfitting and decreased model interpretability. There exists various methods for feature selection, including filter, wrappers and embedded methods. There is no best solution so do no hesitate to experiment with different approaches and/or tools from other courses you might have followed (e.g., statistical tests, ANOVA, etc).

We also ask you to engage in some critical thinking regarding your feature selection outcomes. Consider the following questions: What are the most crucial features? How do they relate to the target variable? Does this make sense within the context of our fictional scenario? Can you formulate hypotheses concerning the reasons behind the increasing occurrence of heart failure in the Smurf Society?

Model selection

We expect you to, at least, implement a linear regression, a K-Nearest Neighbors regressor, a MLP and one other non-linear method (you do not need to implement models from scratch, use dedicated libraries such as scikit-learn). For each model, define an appropriate set of hyper-parameters and try to fine-tune them. This can require a lot of computational resources so explore the hyper-parameter space according to the time available and your laptop configuration.

Remember that careful data splitting is crucial for model selection. Clearly define how you partition your data into training, validation, and test sets. Use cross-validation whenever it is computationally feasible and clearly explain your validation procedure. Based on your results, determine what your best model is and compute an estimate of its generalization performance.

Deliverable

We now detail what you are expected to deliver by the end of this project.

Prediction

Once your model is properly selected and validated, you are asked to produce predictions on the data from `Xtab2.csv` and `Ximg2.csv` for which we have kept secret the corresponding targets. This prediction vector should be uploaded on Moodle in a csv file named `Y2_pred.csv` that contains one prediction per line and no header (no quotation marks around your numbers either). Check that your format is correct by opening it with a text editor and compare it to `Y1.csv`. At the tail end of this file you will add an additional number which is the estimated performance of your model on the unseen data (your file should thus have 401 lines with pure floating point numbers! No more no less!). We will use **RMSE** as the evaluation criterion. **If you transform the target at some point, do not forget to apply the inverse transform before estimating the generalization performance and/or before making your predictions.**

Report

You will produce a report documenting your technical choices and experimental results. We do not need a course on the methods you use. We are more interested in what you did and why. Be concise and go straight to the point! Try to illustrate your results with graphics (with legends and labeled axes) and comment them. Be critical about what you observe and try to give a possible justification of the obtained results. Summarize your observations in a conclusion. A strict **maximum of 6 pages** (fontsize 11 or larger) will be observed. All your figures and computation need to be reproducible by us running your implementation code on the provided data.

Code

Also on Moodle, you should submit a compressed folder containing all your python scripts (notebooks, utils.py, etc). Theses should be runnable and contain at least what you discussed in your report. There is no size limit, but these files should be structured, commented, and clear enough so that information can be easily found without deciphering everything! If you used any packages that were not used during the practical sessions, or a different version of those, don't forget to mention it in the beginning of your file.

Schedule

- As soon as possible: Register your group (maximum two people) on Moodle
- Thursday 9/11 at 8h30: Q/A session #1
- Thursday 7/12 at 8h30: Q/A session #2
- Thursday 21/12 at 23h55: final deadline where you submit your work as 3 separate files (a csv file for your predictions, a pdf for your report, and a compressed folder for all your scripts)

You have about 7 weeks to complete this project. Do not wait until the last minute to start, and take advantage of the Q/A sessions for asking your questions and receiving feedback. We also encourage you to discuss about the project with other groups. We do not want to see plagiarism, but we certainly value exchange of ideas and experiences. Remember to cite all your sources.

Evaluation

The project will account for half of the points in this course (10/20). We have provided you with the evaluation grid in Table 1, which will serve as the basis for grading. The aim of this evaluation grid is to provide a general assessment of the quality of your work (A is better than D, NA means ”Not applicable”). Please, do not regard it as a rigid list of points to be addressed: although most are indeed important, you do not necessarily have to address every single one of them. It is not exhaustive either: you can tackle other questions than those listed in the grid if you find them interesting. Use this tool wisely to evaluate yourself. You can get an idea of the weighting of each criterion in the global evaluation with the ⋆ symbols : more ⋆'s means a greater weight. Note that performance of your model will only account for a small part of your grade. Also, we really insist on the quality of the report; be concise, clear, justify your choices and interpret your results! Embrace this mantra: a good project with a bad report is a bad project!

CONTENT	
Data engineering (⋆⋆)	
Treatment of categorical variables	**A - B - C - D - NA**
How are categorical variables encoded? Relevance of encoding strategies?	
Feature transformation	**A - B - C - D - NA**
Normalization and/or Standardization? Other transformations? *Are the transformations correctly applied to the training/validation and the test sets?*	
Image embeddings	**A - B - C - D - NA**
How are image embeddings used? *Enhancement of CNN architecture for producing better embeddings?*	
Others	**A - B - C - D - NA**
Handling of missing data? Outliers? Dimensionality reduction? etc	
Feature selection (⋆⋆⋆)	
Selection method	**A - B - C - D - NA**
How are features selected? According to which criterion? *Use of filters methods? Of wrappers? Thresholds?* *Is the employed feature selection model-agnostic/model-dependent?*	
Feature importance analysis	**A - B - C - D - NA**
What are the least/most important features/group of features? Interpretation?	
Model selection (⋆⋆⋆)	
Hyper-parameter tuning	**A - B - C - D - NA**
Which hyper-parameters are tuned for each model? Relevance of tuned hyper-parameters? *What search method is employed? How is the search space defined?* *What are the best hyper-parameters for each model?* *Impact on performance? Cross-validation? Robustness of results?*	
Model comparison	**A - B - C - D - NA**
Are the models adequately compared to one another? Strong/weak points of each model? *How is the best model selected? What is the best model? Interpretations?* *Cross-validation? Robustness of results?*	
Generalization error estimation	**A - B - C - D - NA**
How is the generalization error estimated? Is it estimated without bias?	
Performance (⋆)	
Test set error	**A - B - C - D - NA**
How does the model perform on the test data (X2)?	
Estimation error	**A - B - C - D - NA**
Is the generalization error estimate close to the error measured on the test data (X2)?	
FORM	
Report (⋆⋆⋆)	
Quality of writing	**A - B - C - D - NA**
Spelling, layout, correct use of language, ability to summarize, compliance with length criteria	
Consistency of writing	**A - B - C - D - NA**
Clear and precise explanations, coherent structure, appropriate scientific/technical vocabulary	
Figures, tables and diagrams	**A - B - C - D - NA**
Lisibility, clarity, choice, relevance	
Code (⋆⋆)	
Quality of the code	**A - B - C - D - NA**
Readability, structure	

Table 1: Evaluation Criteria.