Statistical Learning

Due whenever you want before June 3rd

Homework-02

(A) Data and Goal

What and where

For this homework you will work with a reduced version of the dataset I prepared for the **2022 edition of the SL-Hacka**. Although not huge, if you are working with `R`, I strongly suggest to have a look at the `data.table` package and in particular to the function `fread()` to import the data.

The Task: Run Baby Run

Heart rate zones, or HR zones, are a way to monitor how hard you're training. There are typically 5+1 heart rate zones (`Zone-0`, `Zone-1`,...,`Zone-5`) based on the intensity of training with regard to the individual *maximum heart rate*.

Heart rate zones are closely linked to your *aerobic* and *anaerobic* thresholds. Understanding this can really help when considering heart rate zones exercise, especially your heart rate zones for running or heart rate zone training for other fitness goals.

The following HR zones chart shows the level of intensity as a percentage of Maximum Heart Rate used in each one.

Zone	Intensity	Percentage of HRmax
Zone 0	Resting	-
Zone 1	Very light	50–60%
Zone 2	Light	60–70%
Zone 3	Moderate	70–80%
Zone 4	Hard	80–90%
Zone 5	Maximum	90–100%

Data Collection

Modern sports watches contain many sensors to monitor `heart rate`, `cadence`, `altitude`, etc. The readings are typically saved every once per second.

The data are coming from FIT files collected on many runs during 2021 and 2022. The runs were made in various environments, i.e. hilly and flat. Also, various efforts, e.g. long and slow runs and interval training, are included. The data collection was made using an Apple Watch accompanied by a Polar OH1 sensor.

Your Task: Get in the "`Zone`"

I will provide you with data relative to 1 minute long running efforts in order to predict the associated heart-rate zone.

Evaluation Metric

The evaluation metric for this competition is **Micro F1-Score**. The F1 score, commonly used in information retrieval, measures accuracy using the statistics **precision** (`p`). and **recall** (`r`).

- **Precision** is the ratio of true positives `tp` to all predicted positives `tp` + `fp`.
- **Recall** is the ratio of true positives `tp` to all actual positives `tp` + `fn`.

The F1 score is given by:

$$\text{F1} = 2\frac{\text{p}\cdot\text{r}}{\text{p}+\text{r}} \quad \text{where} \quad \text{p} = \frac{\text{tp}}{\text{tp}+\text{fp}}, \quad \text{r} = \frac{\text{tp}}{\text{tp}+\text{fn}}$$

The F1 metric weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

Submission format

For each **id** in the test set, you must predict the corresponding *hear-rate zone*: `Z0`, `Z1`,...,`Z5` in a **one-hot-encoding** representation (in `R` you can use `dummyVars()` from `caret`, whereas in `Python` you can use `OneHotEncoder` from `sklearn`) The file should be in .csv format, must contain a header and have the following format:

id	Z0	Z1	Z2	Z3	Z4
1	0	1	0	0	0
6	0	0	0	0	0
7	0	0	1	0	0
10	0	1	0	0	0
13	0	0	0	0	1

Data Description | Feature list

- Each **row** of the train/test datasets corresponds to 1 minute/60 seconds of running.
- The `train` dataset has columns/features (... possibly too many...) that can be broken down in the following way:
  - `id`: simply the row index (ignore for modeling)
  - `y`: the target variable to predict (only available in training, of course), with values: `Z0`, `Z1`,...,`Z5`
  - `from_start`: elapsed time from the beginning of the activity (in seconds)
  - `month`: month of the year
  - `day`: period of the day
  - From column `6` to `35`: **speed** in *meters per seconds* (one measurement every 2 seconds)
  - From column `36` to `65`: **cadence** in *revolutions per seconds* (one measurement every 2 seconds)
  - From column `66` to `95`: **altitude** in *meters over sea-level* (one measurement every 2 seconds)
- Please notice that he **cadence** fields in a FIT file represent RPMs. For cycling 1 RPM equals one full rotation of the cranks. For running 1 RPM represents a step from each leg.

⇝ Your job (A) ⇜

To achieve a `F1` of 0.76 (in test), the *2022-Champs* essentially worked on a two steps procedure with a nonlinear dimensionality reduction followed by a tree-based classifier (ensemble), let's see what you can do...

1. First of all, randomly pick $m = 10$ observations from the `training set` and put them aside, that is, don't use them in training/validation ⇝ you will use them in Part (B) of this homework.
2. Similarly to HW01, enter our (private) Kaggle competition, create a notebook, share it with me, and then start competing to show me how good you are. Each submission must be completely reproducible by running its associated *version* of you code in that notebook (try **not** to share multiple notebooks!)
3. You are free to use any method you like although, for Part (B), it would be better if it provides as output a rough estimate of the class probabilities $\Pr(Y = j \mid \boldsymbol{X} = \boldsymbol{x})$. Can you beat 0.76? Keep in mind that the *2022-Champs* had only 48 hours...
4. As for HW01, I expect you to write (and also upload on Moodle) a well commented working code that covers the entire pipeline with all the due explanations behind your choices: from data loading/pre-processing and feature engineering/dim-reduction to model fit and prediction on `test`.

(B) Predicting with Confidence

Introduction

In statistics we usually provide confidence sets in addition to point estimates, is there a similar notion for **predictions**? The answer is yes: we provide *prediction sets* or *set-valued predictions*. Given data $\mathsf{D}_n = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ we construct a set-valued function C_n, depending on D_n such that $\Pr\left(Y_{n+1} \in C_n(\boldsymbol{X}_{n+1})\right) \geqslant 1 - \alpha$.

Conformal prediction is a powerful, model agnostic idea due to Vovk, Gammerman and Shafer (2005). The statistical theory was developed in Lei, Robins and Wasserman (2013), Lei, G'Sell, Rinaldo, Tibshirani and Wasserman (2017), Sadinle, Lei and Wasserman (2018). More recent variations on this idea can be found in Angelopoulos et al. (2021), Bates et al. (2021) and Romano, Sesia and Candès (2020).

⇝ Your job (B) ⇜

First of all, watch the introductory presentation that a student of mine gave last year:

» Juan on Conformal Prediction | SL2023, passcode: 9x4jf+id «

You can also read (at the very least) the first 7 pages of Angelopoulos and Bates (2022) review paper. Then...

1. Starting from the (best) model used in Part (A), implement the *Adaptive Prediction Sets* (`APS`) algorithm.
2. Apply `APS` to the $m = 10$ observations you set aside in Part (A) from the `training set` and check if your intervals cover the actual response. Provide a suitable visualization of the results and comment.
3. Then, randomly pick $m = 100$ observations from the `test set` and build their predictive sets (no ground thruth here). Provide a suitable visualization of the results and comment.