# Guide to replicate SurvLIMEpy experiments

Version 0.0.5

***

February 14, 2023

# 1 Introduction

This document is created to help the reviewers of the Journal of Statistical Software run the code of the experiments of the manuscript "SurvLIMEpy: A Python package implementing SurvLIME".

All the material is split into two folders: *analyses* and *experiments*. The first folder contains just an script that generates "Table 1", "Table 2" and "Table 3" of our manuscript. Section 2 is devoted to explain how to run this script.

Regarding the *experiments* folder, it contains all the scripts needed to reproduce the experiments presented in Section "4. Experiments". These scripts allow to generate all the data needed to obtain the figures in Section "4. Experiments". Additionally, they also generate the figures. Sections 3 to 8 are devoted to explain how to run those scripts.

Note that all the submitted material is available in our *GitHub* repository[1], as it is explained in the manuscript.

# 2 Reproducing the tables

We have created the tables using the R language, since tidyverse package is suitable to perform analyses in an easy way. The script is inside the folder *analyses* and it is called *bivariate_analyses.R*. Before starting the analyses, *tidyverse* package is installed. After that, the script analyses all the datasets and it produces all the tables. Note that to have a successful execution, the datasets need to be in the same folder as the scripts, as it is currently submitted. A single execution of the script produces all the tables.

# 3 Preliminary for experiments

From this section onwards, we assume that the reviewer is located inside the folder *experiments*. All the scripts are coded using Python language. Before running the experiments, we recommend to create an isolated environment, since some packages are required to be installed. Once the environment is created, it is needed to install all the packaged present in the file *requirements.txt*. It can be done by running the following command in the terminal:

The code is spread in 7 different scripts which are all combined in the *all_experiments_script.py* script:

[1] https://github.com/imatge-upc/SurvLIME-experiments

```
pip install -r requirements.txt
```

- all_experiments_script.py
- experiment_1_montecarlo.py
- experiment_1_2_montecarlo.py
- experiment_real_dataset_ml.py
- deepsurv_rds.py
- make_plots_script.py
- hyperparams.py

We recommend to use this *all_experiments_script.py* script if the reviewers want to execute any experiment of the manuscript. Throughout the following sections, it is explained how to use this script.

The Manuscript Preparation Section inside the Information for Authors of this journal specifies that experiments should be able to run within an hour. Our experiments take longer than 1 hour to run as, at its core, we are solving optimisation problems multiple times.

In our manuscript, we repeat some experiments 100 times. However, in the submitted script, this parameter has been set to 10 in order to shorten the execution time. Therefore, the results obtained will not be exactly the same as the one submitted in the paper. Nonetheless, we will provide details about what to do in order to obtain the same results. If so, the experiments will last more than 1 hour.

In any case, we also provide the data we have obtained to generate the manuscript so that the reviewers can either inspect it or use it.

# 4 Experiments of Section “4.1. Simulated data”

In case the reviewers want to obtain similar results as the ones provided in Section “4.1. Simulated data”, execute the following command in the terminal:

```
python all_experiments_script.py --exp simulated
```

In case the reviewers want to obtain exactly the same results as the ones submitted in the manuscript, the following code needs to be executed in the terminal:

```
python all_experiments_script.py --exp simulated --repetitions 100
```

The directory *computed_weights_csv/exp2* is created and the data is stored in that directory. There, multiple csv files are stored:

- center_cluster_1.csv: data needed in order to create Figure 3 left.
- center_cluster_2.csv: data needed in order to create Figure 3 right.
- exp_2_cluster_1_mean.csv: data needed in order to create Figure 4 top left.
- exp_2_cluster_1_min.csv: data needed in order to create Figure 4 top middle.
- exp_2_cluster_1_max.csv: data needed in order to create Figure 4 top right.
- exp_2_cluster_2_mean.csv: data needed in order to create Figure 4 bottom left.
- exp_2_cluster_2_min.csv: data needed in order to create Figure 4 bottom middle.
- exp_2_cluster_2_max.csv: data needed in order to create Figure 4 bottom right.

If the reviewers want more details about the experiments related to this section, please inspect scripts *all_experiments_script.py*, *experiment_1_montecarlo.py* and script and *experiment_1_2_montecarlo.py*.

Once all the data is stored, it is used to generate the figures related to this section. The script that manages this part is *make_plots_script.py*. It is not needed to execute it since the script *all_experiments_script.py* is in charge of executing it. Before storing the figures, a directory called *figures* is created. Afterwards, all the figures are stored. Concretely, the following figures are created:

- center_cluster_1.png: Figure 3 left.
- center_cluster_2.png: Figure 3 right.
- simulated_exp_cluster1_mean_values.png: Figure 4 top left.
- simulated_exp_cluster1_min_values.png: Figure 4 top middle.
- simulated_exp_cluster1_max_values.png: Figure 4 top right.

- simulated_exp_cluster2 _mean_values.png: Figure 4 bottom left.
- simulated_exp_cluster2_min_values.png: Figure 4 bottom middle.
- simulated_exp_cluster2_max_values.png: Figure 4 bottom right.

## 5 Experiments of Section “4.2. Real data”

In case the reviewers want to obtain similar results as the ones provided in Section “4.2: Real data”, they must execute the following command in the terminal:

```
python all_experiments_script.py --exp real
```

In case the reviewers want to obtain exactly the same results as the ones reported in the manuscript, the following command should be executed on the terminal:

```
python all_experiments_script.py --exp real --repetitions 100
```

Note that the execution time of the previous command takes longer than 1 hour. After executing it, all the models are trained using the same train/test used for the manuscript. Additionally, the set of hyperparamenters are provided to the models. This set is the same as the one used for the manuscript. In order to obtain more details about the set of hyperparameters, please inspect the script *hyperparams.py*

Once the models are trained, the data needed to produce the results are obtained. Concretely, the data obtained is:

- cox_exp_udca_surv_weights.csv: data needed in order to create Figure 5 top.
- rsf_exp_udca_surv_weights.csv: data needed in order to create Figure 5 bottom left.
- xgb_exp_udca_surv_weights.csv: data needed in order to create Figure 5 bottom right.
- cox_exp_lung_surv_weights.csv: data needed in order to create Figure 6 top.
- rsf_exp_lung_surv_weights.csv: data needed in order to create Figure 6 bottom left.

- xgb_exp_lung_surv_weights.csv: data needed in order to create Figure 6 bottom right.
- cox_exp_veteran_surv_weights.csv: data needed in order to create Figure 7 top.
- rsf_exp_veteran_surv_weights.csv: data needed in order to create Figure 7 bottom left.
- xgb_exp_veteran_surv_weights.csv: data needed in order to create Figure 7 bottom right.

All these files are store in the folder *computed_weights_csv/exp_real_datasets*. If this folder does not exist, it is created before storing the files. After storing the data, the figures are produced and stored in the folder *figures*. Concretely, the set of figures created are:

- cox_udca.png: Figure 5 top.
- rsf_udca.png: Figure 5 bottom left.
- xgb_udca.png: Figure 5 bottom right.
- cox_lung.png: Figure 6 top.
- rsf_lung.png: Figure 6 bottom left.
- xgb_lung.png: Figure 6 bottom right.
- cox_veterans.png: Figure 7 top.
- rsf_veterans.png: Figure 7 bottom left.
- xgb_veterans.png: Figure 7 bottom right.

It is not needed to execute the whole script if just a subset of figures is needed. For instance, if just the experiments related to UDCA are needed, the following code should be execute on the terminal: The code executed is just

```
python all_experiments_script.py --exp real --detaset udca
```

the one needed in order to produce the data and the figures related to UDCA dataset. If 100 repetitions are needed, include this option in the terminal. In order to replicate the results for the other two datasets, the following code must be execute:

In case the reviewers want to obtain more details about the experiments of Section “4.2. Real data”, please inspect the scripts *all_experiments_script.py* and *experiment_real_datasets_ml.py*.

```
python all_experiments_script.py --exp real --detaset lung
python all_experiments_script.py --exp real --detaset veterans
```

## 6 Experiments of Section “4.3. Simulated data and deep learning models”

In case the reviewers want to obtain similar results as the ones provided in Section “4.3 Simulated data and deep learning models”, they must execute the following command in the terminal:

```
python all_experiments_script.py --exp dl
```

In case the reviewers want to obtain exactly the same results as the ones reported in the manuscript, the following command should be executed on the terminal:

```
python all_experiments_script.py --exp dl --repetitions 100
```

After executing this command, the directory *computed_weights_csv/exp_deepsurv_rds* is created. There, the following csv files is stored:

- exp_deepsurv_rds_surv_weights.csv: data needed in order to create Figure 8.

Once all the data is stored, it is used to generate the figure related to this section. The script that manages this part is *make_plots_script.py*. It is not needed to execute it since the script *all_experiments_script.py* is in charge of executing it. Before storing the figures, a directory called *figures* is created. Afterwards, the figures is stored. Concretely, the following figure is created:

- deepsurv_rds.png: Figure 8.

In case the reviewers need more details, please inspect scripts *all_experiments_script.py* as well as *deepsurv_rds.py*

## 7 Executing all experiments at the same time

The script *all_experiments_script.py* can also be execute to generate all the data and all the figures explained in Sections 4 to 6 using a single command.

```
python all_experiments_script.py --exp all
```

In case the reviewers want to obtain similar results as the ones provided in our manuscript, the following command must be executed:

In case the reviewers want to obtain exactly the same results as the ones reported in the manuscript, the following command should be executed on the terminal:

```
python all_experiments_script.py --exp all --repetitions 100
```

This will create all the data as well as all the figures described in Sections 4 to 6. Note that it will take much longer than 1 hour.

## 8 Providing our data

Apart from the scripts, we are also providing the data we have obtained to generate the manuscript. It is stored in the folder zzz. This data allows to create all the figures presented in our manuscript. The following command produces all of them:

```
python all_experiments_script.py --exp only_plot
```

Before running the script, the folder zzz must be renamed to zzz. The code is in charge of creating a folder called *figures* and storing there all the figures. Note that the figures stored in the folder *figures* are explained in Sections 4 to 6.