TASK 2

MEDICAL EVENTS PREDICTION

You are in the group Upsanamattomar consisting of ocariaga (ocariaga@student.ethz.ch (mailto://[u'ocariaga@student.ethz.ch'])), uchakrabo (uchakrabo@student.ethz.ch (mailto://[u'uchakrabo@student.ethz.ch'])) and wmatt (wmatt@student.ethz.ch (mailto://[u'wmatt@student.ethz.ch'])).

1. READ THE TASK DESCRIPTION

2. SUBMIT SOLUTIONS

3. HAND IN FINAL SOLUTION

1. TASK DESCRIPTION

INTRODUCTION

Patients in hospitals are often continuously monitored by the medical personnel, who collect data about the patients' demographics, vital signs and lab test results. In this task, you will explore how to forecast future occurrence of medical events such as sepsis, future orders of medical tests, as well as evolution of key vital signs of patients in the remainder of their stay, based on data available from their first 12 recorded hours of stay. If the predictions are accurate, they could potentially help in the resource planning and workflow management in the hospitals.

In this task you will face the typical challenges of working with real medical data: missing features and imbalanced classification, where we are predicting rarely-occuring events.

DATASET DESCRIPTION

Download handout (/static/task2_s92hdjw2.zip)

The handout you download contains the following files:

- **train_features.csv** training set features
- **train_labels.csv** training set labels
- **test_features.csv** test set features
- **sample.zip** a sample submission file in the correct format, compressed
- **score_submission.py** code to illustrate how the server calculates the score for a submission

The data of each patient is identified through a unique patient id (*pid*). In train_features.csv, medical data for each patient is arranged in 12 consecutive rows representing the information about the first 12 recorded hours of their hospital stay. The column *Time* provides the time after hospital admission

(as an offset in hours) at which the vital sign / medical test measurements were observed. The column *Age* contains the age of the patient, and it is unique for a single pid. The columns *Heartrate, SpO2, ABPs, ABPm, ABPd, RRate, Temp* are vital signs. and the remaining columns without a 'LABEL_' prefix are medical test observations. All of these columns can contain missing values, which are indicated by 'nan'.

In this task, we include patients that are known to have at least 24 hours of recorded stay in the hospital, and we provide you with the data about the first 12 recorded hours of their stay. Your task is to predict how the patient's state evolves and which tests will be ordered by the clinicians during the remainder of their hospital stay.

The labels that should be predicted can be found in train_labels.csv and have column names prefixed by 'LABEL_'. These labels can be categorized into 3 subtasks:

- **Subtask 1**: Predict whether medical tests are ordered by a clinician in the remainder of the hospital stay: 0 means that there will be no further tests of this kind ordered, 1 means that at least one of a test of that kind will be ordered. In the submission file, you are asked to submit predictions in the interval [0, 1], i.e., the predictions are not restricted to binary. 0.0 indicates you are certain this test will not be ordered, 1.0 indicates you are sure it will be ordered. The corresponding columns containing the binary groundtruth in train_labels.csv are: *LABEL_BaseExcess, LABEL_Fibrinogen, LABEL_AST, LABEL_Alkalinephos, LABEL_Bilirubin_total, LABEL_Lactate, LABEL_TroponinI, LABEL_SaO2, LABEL_Bilirubin_direct, LABEL_EtCO2*.
- **Subtask 2**: Predict whether sepsis will occur in the remaining stay: 0 means that no sepsis will occur, 1 otherwise. Similar to Subtask 1, you are asked to produce predictions in the interval [0, 1] for this task. The corresponding column containing the binary groundtruth in train_labels.csv is *LABEL_Sepsis*.
- **Subtask 3**: Predict future mean values of key vital signs. The corresponding columns containing the real-valued groundtruth in train_labels.csv are *LABEL_RRate, LABEL_ABPm, LABEL_SpO2, LABEL_Heartrate*.

Tip: for subtasks 1 and 2, you will need to produce predictions in the interval [0, 1]. The evaluation metric on these tasks is the Area Under the Receiver Operating Characteristic Curve (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html#sklearn.metrics.roc_auc_score), which takes into account the ranking of your predictions and not their actual values. So it is important not to restrict yourself to binary classification, but produce real-valued predictions in the interval [0, 1]. How can you achieve this with an SVM? In the lecture, you have seen that the SVM prediction for binary classification is $\mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i y_i k(x, x_i)\right)$. In order to produce real-valued predictions in the interval [0, 1] with SVM, you can replace the sign function by the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$.

Note: while train_features.csv contains 12 entries for a patient (corresponding to the first 12 recorded hours of their stay), train_labels.csv contains 1 entry per patient, since these refer to the events during the remaining hours of the patient's stay, as described above.

The structure of test_features.csv is the same as the structure of train_features.csv. For each patient, you will need to make predictions for subtasks 1-3. The submission format should be identical to the format of sample.csv compressed in sample.zip, which is provided in the handout.

Both train_features.csv and test_features.csv contain missing values ('nan' entries). An important part of this project is how to deal with such missing data (known as 'data imputation' in the ML literature) and feature engineering (what features can you extract from measurements taken in consecutive hours, etc.)

Important:

- While train_features.csv and test_features.csv contain multiple entries for a single patient (identified by the pid), your submission should contain a single line associated with each pid, since you are predicting vital signs / events / medical test orders for a patient for the period between the 13th recorded hour and the end of their hospital stay. Please see sample.zip for reference.
- To keep the server load small, there is an 1 MB total upload limit for the submission. For this, please submit your predictions in a **zip** archive. To reduce the size further, we recommend to

limit the decimal places you write for floats. An example of how to write a pandas dataframe directly to a zip, with floats up to 3 decimal digits:

```
import pandas as pd

# suppose df is a pandas dataframe containing the result
df.to_csv('prediction.zip', index=False, float_format='%.3f', compression='zip')
```

EVALUATION

For subtasks 1 and 2 we calculate the Areas Under the Receiver Operating Characteristic Curve, aka ROC_AUC scores (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html#sklearn.metrics.roc_auc_score). For the regression subtask 3, we measure the R^2 score (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html), threshold it below at 0 and normalize it to the range [0.5, 1]. Your final score is the average of these 3 scores. For your convenience, in the handout we provide you with the `get_score` function in score_submission.py, which shows how to calculate this score in Python.

GRADING

We provide you with **one test set** for which you have to compute predictions. We have partitioned this test set into two parts (of the same size) and use it to compute a *public* and a *private* score for each submission. You only receive feedback about your performance on the public part in the form of the public score, while the private leaderboard remains secret. The purpose of this division is to prevent overfitting to the public score. Your model should generalize well to the private part of the test set.

When handing in the task, you need to select which of your submissions will get graded and provide a short description of your approach. This has to be done **individually by each member** of the team. We will then compare your selected submission to three baselines (easy, medium and hard). Your final grade depends on the public score and the private score (weighted equally), on your submitted code and on a properly-written description of your approach. The following **non-binding** guidance provides you with an idea on what is expected to obtain a certain grade: If you hand in a properly-written description, your source code is runnable and reproduces your predictions, and your submission performs better than the easy baseline, you may expect a grade exceeding a 4. If it further beats the medium baseline, you may expect that the grade will exceed a 5. If in addition your submission performs equal to or better than the hard baseline, you may expect a 6. If you do not hand in a properly-written description of your approach, you may obtain zero points regardless of how well your submission performs.

⚠ Make sure that you properly hand in the task, otherwise you may obtain zero points for this task.

FREQUENTLY ASKED QUESTIONS

WHICH PROGRAMMING LANGUAGE AM I SUPPOSED TO USE? WHAT TOOLS AM I ALLOWED TO USE?

You are free to choose any programming language and use any software library. However, **we strongly encourage you to use Python**. You can use publicly available code, but you should specify the source as a comment in your code.

IN WHAT FORMAT SHOULD I SUBMIT THE CODE?

You can submit it as a single file (main.py, etc.; you can compress multiple files into a .zip) having max. size of 1 MB. If you submit a zip, please make sure to name your main file as *main.py* (possibly with other extension corresponding to your chosen programming language).

WILL YOU CHECK / RUN MY CODE?

We will check your code and compare it with other submissions. We also reserve the right to run your code. Please make sure that your code is runnable and your predictions are reproducible (fix the random seeds, etc.). Provide a readme if necessary (e.g., for installing additional libraries).

SHOULD I INCLUDE THE DATA IN THE SUBMISSION?

No. You can assume the data will be available under the path that you specify in your code.

CAN YOU HELP ME SOLVE THE TASK? CAN YOU GIVE ME A HINT?

As the tasks are a graded part of the class, **we cannot help you solve them**. However, feel free to ask general questions about the course material during or after the exercise sessions.

CAN YOU GIVE ME A DEADLINE EXTENSION?

⚠ We do not grant any deadline extensions!

CAN I POST ON PIAZZA AS SOON AS HAVE A QUESTION?

This is highly discouraged. Remember that collaboration with other teams is prohibited. Instead,

- Read the details of the task thoroughly.
- Review the frequently asked questions.
- If there is another team that solved the task, spend more time thinking.
- Discuss it with your team-mates.

WHEN WILL I RECEIVE THE PRIVATE SCORES? AND THE PROJECT GRADES?

We will publish the private scores, and corresponding grades before the exam the latest.