Data Science Challenge – Phase I

The aim of this challenge is to predict risk of death (mortality) in patients admitted to intensive care units (ICU) within hospitals.

The data consists of 5990 patient records where each **patient record** has the following **variables**:

1. ID: a unique identifier for each patient
2. Age
3. 6 Vitals: see table 1
4. 25 Labs: see table 2
5. Timestamps: measurement time relative to first measurement for patient
6. ICU flag: indicates whether a patient is in ICU or not at a given time
7. Mortality label: indicates whether a patient survived or died (the label or outcome variable)

A **timestamp** shows when a **vital or lab measurement** is made relative to the first measurement. There can be one or more measurements at each timestamp. The first timestamp for each patient is always 0 (zero). Subsequent timestamps show time elapsed from this first timestamp in seconds. Regular intervals between measurement timestamps *cannot* be assumed. The length-of-stay of each patient is different and so the number of measurements (hence timestamps) are different in each patient record.

Vitals and Labs are **time-series variables** associated with timestamps. Each vital or lab variable may be measured multiple times (at different timestamps) for a patient. Not all variables are measured at all timestamps. Some variables may not be measured at all in a patient record.

At each timestamp the **ICU flag** indicates whether the patient is inside an ICU or outside (in the hospital). Every patient in this dataset visits the ICU exactly once during the hospital stay. Thus the period when the patient is in the ICU is indicated by a series of consecutive timestamps where the ICU flag is set to 1. Patient records have measurements both inside as well as outside the ICU.

Each patient has a single age value. Age values for all patients are given.

The **mortality indicator**, a single label per patient, indicates whether the patient survived or died at the end of the hospital stay. No patient in this dataset dies outside the ICU during the hospital stay. Time of death is not provided. For patients that die, the time of death is after the last measurement recorded.

	Vital	Unit
1	Systolic Blood Pressure	mmHg
2	Diastolic Blood Pressure	mmHg
3	Respiration Rate	bpm
4	Heart Rate	bpm
5	Oxygen Saturation	%
6	Temperature	Celsius

Table 1: Vitals

	Lab investigations	Unit
1	Arterial blood Ph	
2	Partial Pressure of Carbon dioxide (PaCO2)	mmHg
3	Partial Pressure of Oxygen (PaO2)	mmHg
4	Sodium	mmol/L
5	Potassium	mmol/L
6	Bicarbonate	mmol/L
7	Blood Urea Nitrogen	mg/dL
8	Serum Creatinine	mg/dL
9	WBC Count	x 10^3/µL
10	Hematocrit	%
11	Platelet Count	x 10^3/µL
12	Bilirubin	mg/dL
13	Urine Output	ml
14	LDL Cholesterol	mg/dL
15	Lactic Acid	mmol/L
16	Troponin I	ng/ml
17	Troponin T	ng/ml
18	Random Blood Glucose	mg/dL
19	Fasting Blood Glucose	mg/dL
20	Fraction of Inspired Oxygen (FiO2)	%
21	Albumin	g/dl
22	Alkaline Phosphatase	IU/L
23	Alanine	IU/L
24	HDL Cholesterol	mg/dL
25	Magnesium	mg/dL

Table 2: Lab investigations (labs)

The hyperlinks point to websites giving more information about the investigation. These are for your information only.

Patient records are divided into 3 sets: train, validation and test sets. Train set contains 3594 patient records. Validation and test sets each contain 1198 patient records. Only the train and validation sets are available to participants. The test set will not be disclosed.

Data format

The train set consists of 4 files:

1) id_time_vitals_train.csv
2) id_time_labs_train.csv
3) id_age_train.csv
4) id_label_train.csv

File id_time_vitals_train.csv contains timestamped vitals measurements of all the train patients along with the ICU flag for each timestamp. The column headers in the file are:

ID, TIME, V1,…,V6, ICU

File id_time_labs_train.csv contains timestamped labs measurements of all train patients. The column headers in the file are:

ID, TIME, L1,…, L25

Each row contains measurements made for a patient at a given timestamp. Measurements that are not made at a timestamp has the value ‘NA’. The first timestamp for a patient is always 0. Subsequent timestamps show time elapsed from this first timestamp in seconds.

V1 – V6 indicate the six vital measurements and L1 – L25 indicate 25 lab measurements (given in tables 1 and 2).

Some lines in the file may contain only 'NA's after the ID and timestamp. This could happen in the vitals file if there are only lab measurements at that timestamp and no vitals are measured. Similarly it could happen in the labs file it there are only vitals measured at that timestamp. Also note that the values may not look like real world values due to the noise added intentionally. E.g. Systolic Blood Pressure values are integers in real world, but may be float values in the dataset.

The ICU variable can take two values: 0 indicates the patient is not in ICU at the timestamp and 1 indicates the patient is in ICU at the timestamp.

File id_age_train.csv contains one line per patient, each line containing ID and age. Column headers are: ID, AGE

File id_label_train.csv contains one line per patient, each line containing ID and mortality label. Label 0 indicates survival and label 1 indicates death. Column headers are: ID, LABEL

Validation and test sets

The **validation set** consists of three files: id_time_vitals_val.csv, id_time_labs_val.csv, id_age_val.csv. The format is the same as that of the corresponding train files (with different patient IDs and data, of course!).

The labels for the validation set will **not** be provided. A script will compute the performance metrics and show the scores on the leaderboard.

The test set will not be provided. It consists of four files: id_time_vitals_test.csv, id_time_labs_test.csv, id_age_test.csv and id_label_test.csv following the same formats as those of train and validation sets. The finalists of this round will be selected based on the performance of their models on this test dataset.

Prediction

Prediction must be made for each patient **only when the patient is in ICU**. The prediction must be done in an **online manner**, that is, at a given timestamp the model can use any of the past data to make a prediction for that timestamp. Predictions are to be made at every measurement timestamp while the patient is in ICU.

Thus, **each patient has a sequence of predictions**. Each prediction is a label 0 or 1 and the number of predictions for each patient is not more than the number of rows in the data, for the given patient, where ICU flag == 1.

Evaluation

We obtain a **final prediction** per patient as follows: If the sequence of predictions for the patients contains only zeros, then the final prediction is 0, otherwise 1.

Prediction time is only defined for patients whose final prediction is 1. It is the difference between the last timestamp (for the patient) and first timestamp with a prediction of 1.

We obtain a patient-wise classification table as follows:

	Mortality label 1	Mortality label 0
Final Prediction 1	True Positives (TP)	False Positives (FP)
Final Prediction 0	False Negatives (FN)	True Negatives (TN)

Sensitivity = TP/(TP+FN), Specificity = TN/(TN+FP)

Our evaluation metrics are:

1. Sensitivity at 0.99 (or higher) specificity
2. Median Prediction Time (over all TP patients in validation/test set)

The metrics are chosen to measure the ability of the model to identify high risk patients at 1% false positive rate and to identify high risk patients as early as possible. Participants should use suitable strategies (like setting sample weights/misclassification costs based on the class during training) to achieve 0.99 or higher specificity. The median prediction time will be used to break ties for entries with same sensitivity and specificity.

The leaderboard will show **sensitivity, specificity, accuracy** and **prediction time** measured over the validation set using our code **compute_performance.py**. The top 10 finalists will be chosen based on sensitivity and prediction time on the test set.

Deliverable

The submitted code should run from the command line taking three arguments which are filenames containing vitals, labs and age of patients (following the formats of the validation set files) respectively. Code for training must be present in the file. Additional documentation describing the entire approach taken will be required from the top 10 finalists.

Example: if the submitted code is run_model.py, it should run on the command line as follows:

$python run_model.py id_time_vitals_test.csv id_time_labs_test.csv id_age_test.csv

The output should be a file with name **output.csv** where each line contains patient ID, prediction time and final prediction. The timestamps must be present in id_time_vitals_test.csv and the icu label must be 1 at these timestamps. **Output files that do not follow these rules will not be evaluated and these entries will be rejected**.

See **sample_output.csv** for a sample output. The code **compute_performance.py** shows how the output file will be used to compute the final performance metrics using the file output.csv generated from the submitted code.

Programming Language

Only open-source languages are allowed, that can run on Linux (Ubuntu) operating systems. Python is the preferred choice. Proprietary software, like MATLAB, are not allowed.

Disclaimer: The dataset has been simulated from real patients' data. No measurement in the dataset belongs to any real patient. Any match with a real patient's measurement is purely coincidental.

The objective of this competition is similar to a competition held by Physionet in 2012: http://physionet.org/challenge/2012/. The data is not the same although many of the measurement variables used are same. Participants are encouraged to see the approaches used by winners of the challenge that are described in papers here: http://physionet.org/challenge/2012/papers/.
