# Pytorch_EHR: Building Recurrent Neural Network based Predictive Models using Electronic Health Records

## Use Case: COVID-19 Patient's Risk for PASC

*A HANDS-ON TUTORIAL*

*BROUGHT YOU BY DEGUI ZHI, LAILA RASMY, ZIQIAN XIE*

*ICHI 2023*

# Learning Objectives

Understanding the theories behind EHR predictive modeling using deep learning.

Learn the basic tools of deep learning to convert theory to practice.

Understand the basics of proper cohort definition.

Practice data preparation and preprocessing

Practice RNN model training and evaluation for binary classification and survival prediction

Learn different techniques used for hyperparameter tuning.

Learn how to present model predictions as well as explanations using attribution mechanism.

# Agenda

- Introduction of the EHR predictive modeling: theory and practice
- EHR data preparation
- RNN-based model training and evaluation
- Explainability of model predictions

# Section 1:
# EHR predictive modeling: Introduction and theory

# Learning Objectives

Understanding the theories behind EHR predictive modeling using deep learning.

Learn the basic tools of deep learning to convert theory to practice.

Understand the basics of proper cohort definition.

Practice data preparation and preprocessing

Practice RNN model training and evaluation for binary classification and survival prediction

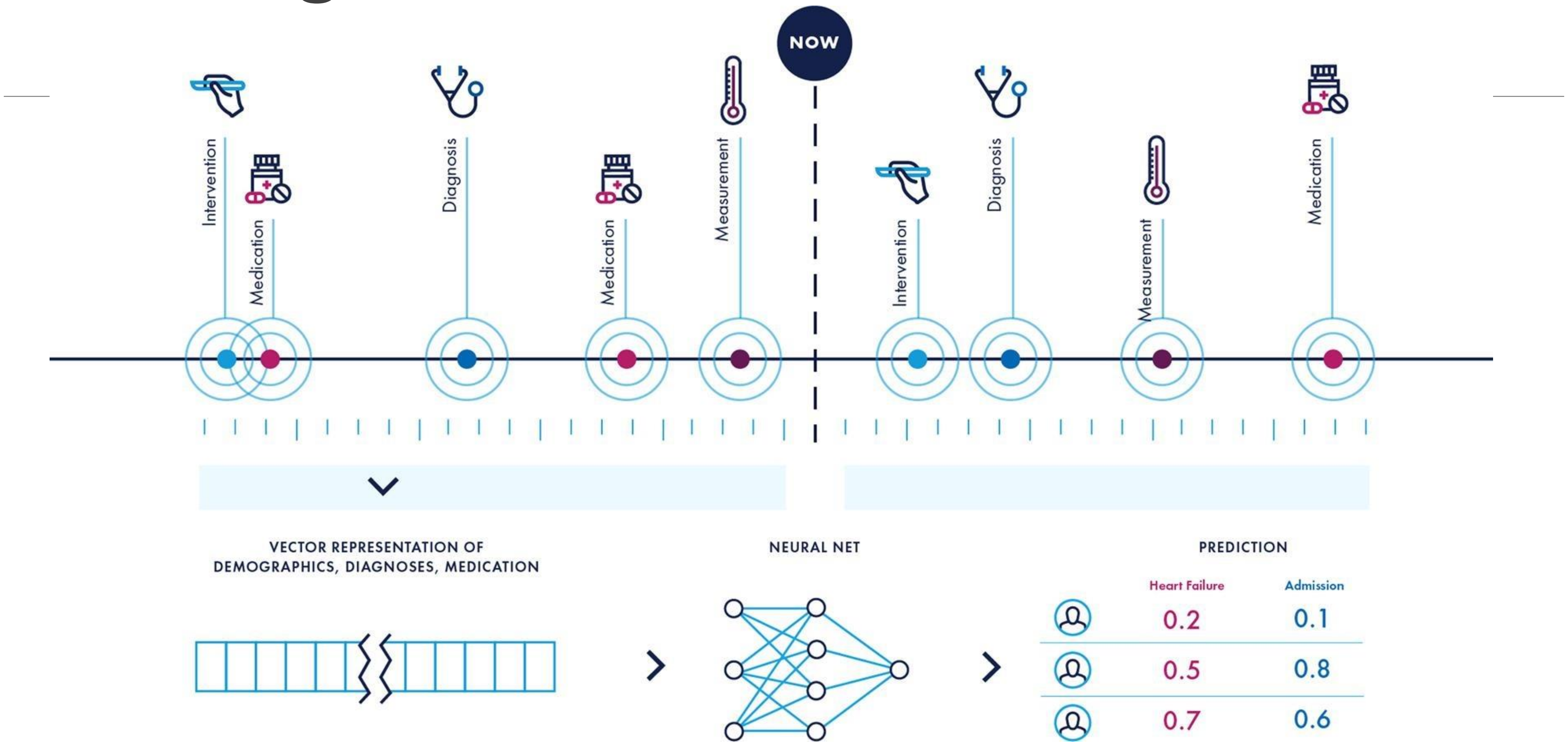
Learn different techniques used for hyperparameter tuning.

Learn how to present model predictions as well as explanations using attribution mechanism.

# Introduction:

## Deep learning for EHR predictive modeling

# Deep Learning for EHR Predictive Modeling






Solaris et al, J Biomed Informatics, 2020

# Flexible architecture of neural nets allows modeling complex dependency structures in EHR data.

Data Volume

Data Quality

Temporality

Multi-Modality

Knowledge



https://academic.oup.com/bib/article/19/6/1236/3800524

# Deep Learning starting to achieve SotA

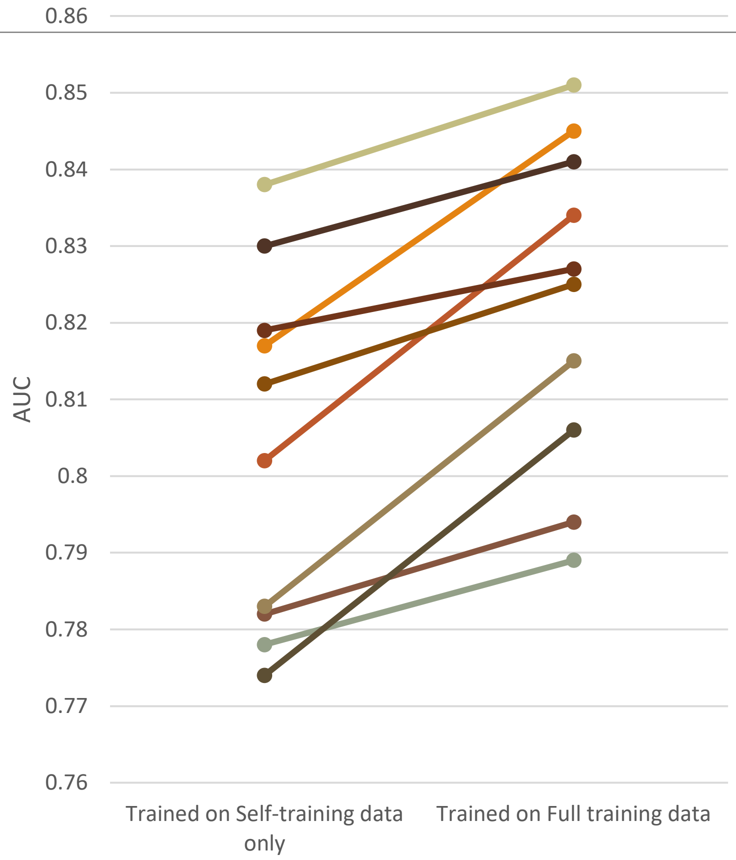
| Model | Heart Failure | Readmission |
|---|---|---|
| GRU | 84.8 | 75.5 |
| LSTM | 83.9 | 73.8 |
| Vanilla-RNN | 83.3 | 63.9 |
| D-GRU | 83.3 | 73.5 |
| D-LSTM | 83.3 | 72.8 |
| D-RNN | 83.2 | 70.9 |
| Bi-GRU | 84.5 | 74.4 |
| Bi-LSTM | 84.4 | 75.2 |
| Bi-RNN | 83.1 | 74.1 |
| T-LSTM | 82.4 | 72.1 |
| QRNN | 83.2 | 71.5 |
| RETAIN | 83.8 | 70.1 |
| LR | 79.0 | 67.0 |
| RF | 78.8 | 73.6 |

Rasmy et al, MedInfo, 2018 https://arxiv.org/abs/2110.00998

# Pooling data improves performance

RNN-based RETAIN model
Predicting Heart Failure Risks for 10 largest hospitals in Cerner Health Facts 2016

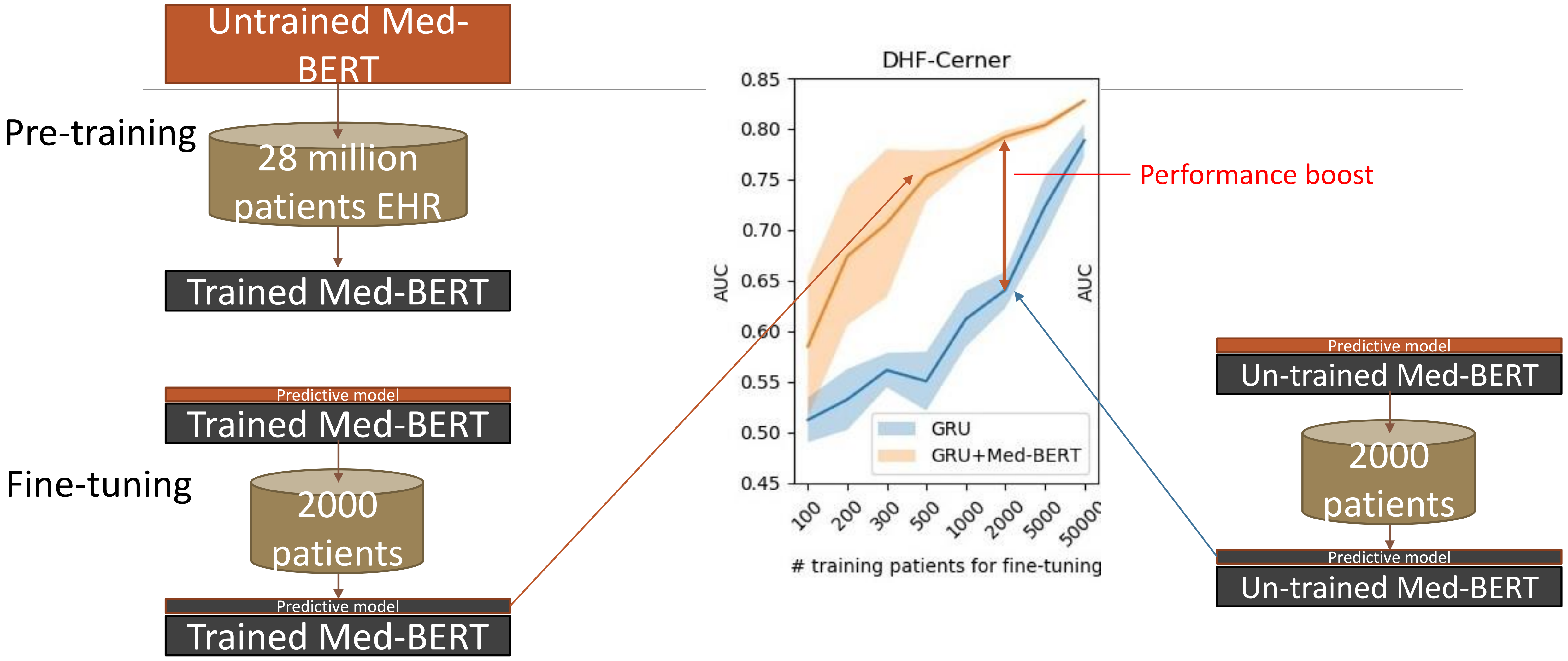
Each hospital has 2,000-6,500 patients
Full data set has 1.3 million patients



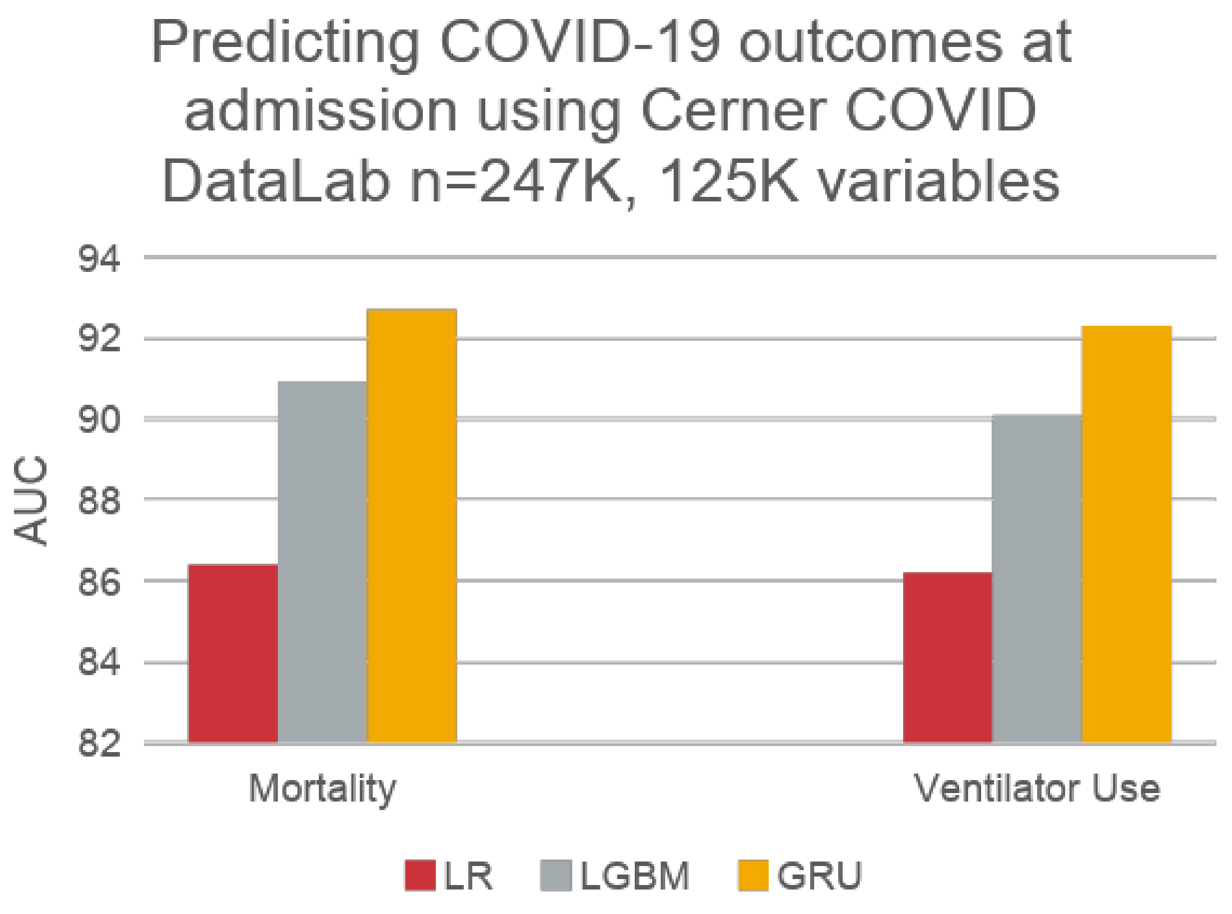


Rasmy et al, J Biomedical Informatics, 2018

# Pre-trained models boost performance




Rasmy et al, npj Digital Medicine, 2021

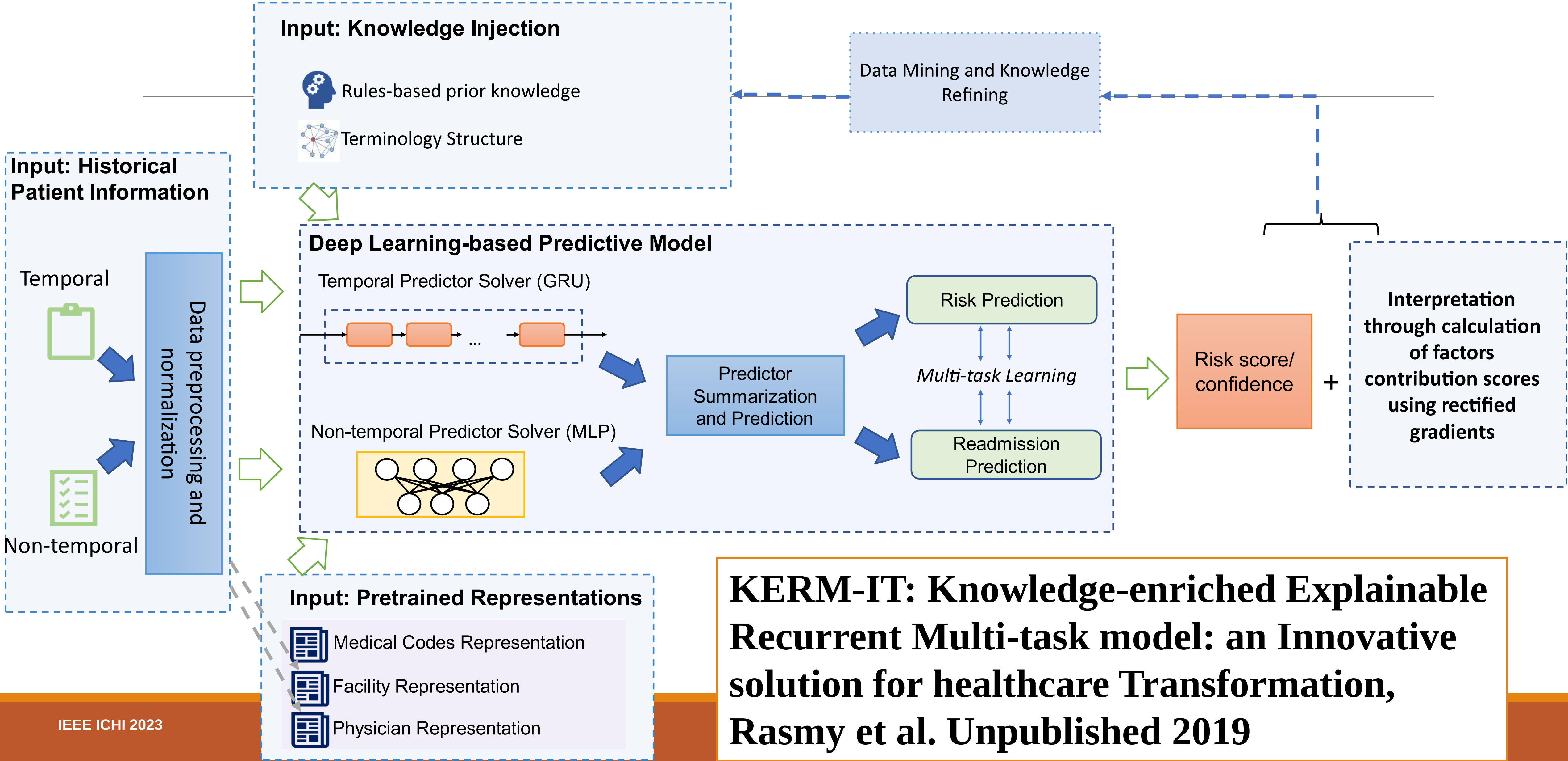



# Our most recent example



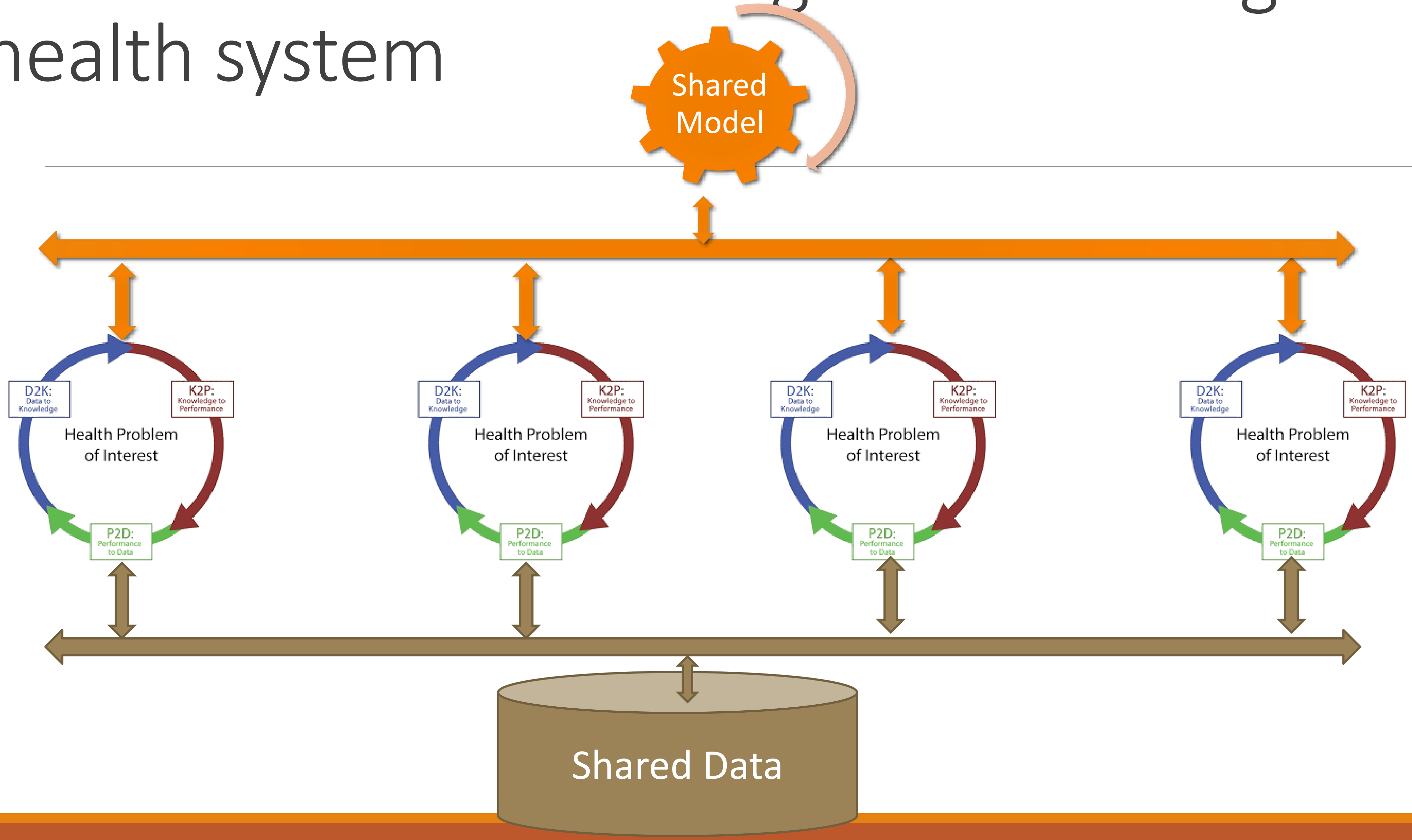
Rasmy et al, Lancet Digital Health, 2022

# Promise: An integrated DL system



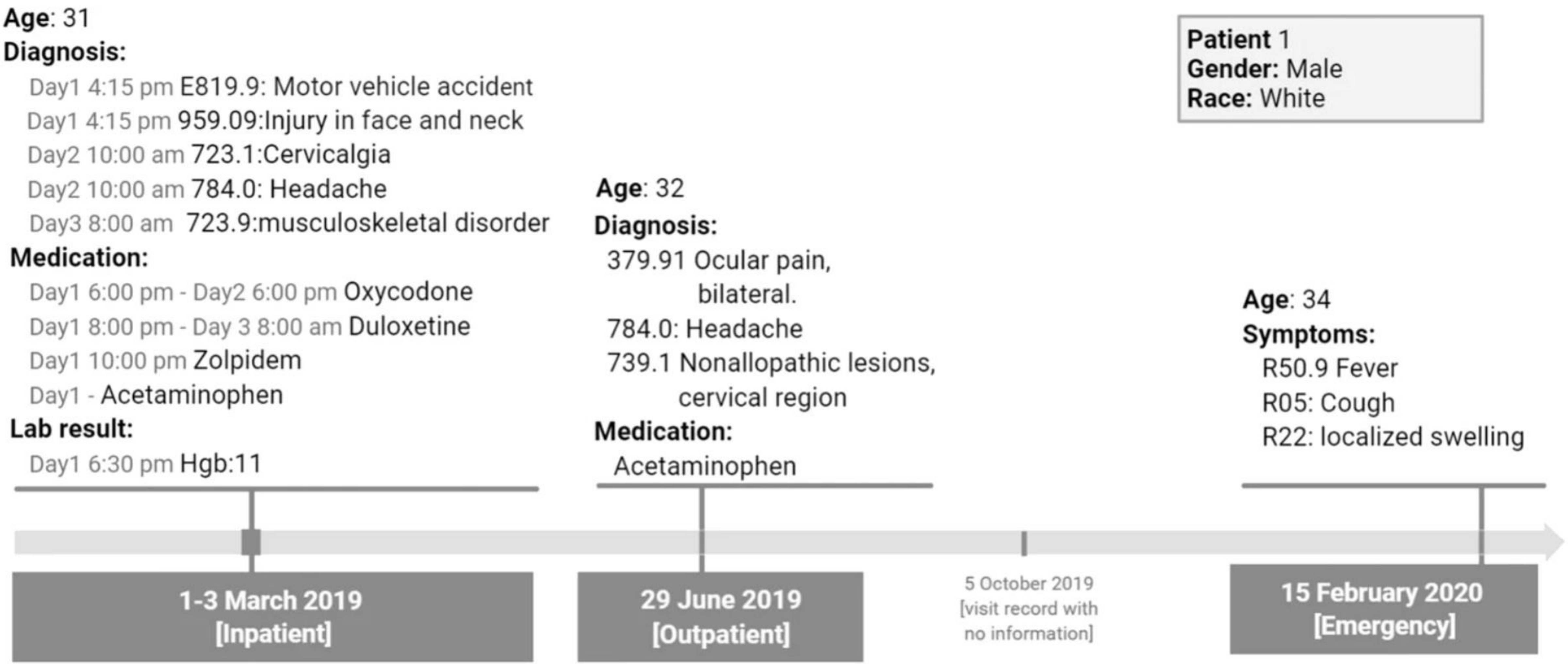

**KERM-IT: Knowledge-enriched Explainable Recurrent Multi-task model: an Innovative solution for healthcare Transformation, Rasmy et al. Unpublished 2019**

# Future: Towards an integrated learning health system



# Theory:

## RNN for structured EHR




# Electronic Health Records (EHR)

*One of the richest (and messiest) sources of patient information*

Rasmy et al, npj Digital Medicine, 2021

# EHR data vs NLP data

| Criteria | Natural language | EHR |
|---|---|---|
| Token granularity | word | code |
| Syntactic: Hierarchical structure | Document – paragraph-sentence – phrase - word | Patient – visit – code (of different categories) |
| Syntactic: Sequential order | Simple and clear. | Codes may with time stamp, but the codes within a visit may be unordered |
| Semantic | Dependency relations are clear to average human. | Dependency unclear |
| Time interval | Regular | irregular |
| Data completeness | Relatively complete. | Usually incomplete, may contain errors. |
| Sequence length | Within a relatively narrow range. | More variable |



Rasmy et al, npj Digital Medicine, 2021

# EHR predictive modeling: Input data modality

## EPISODIC

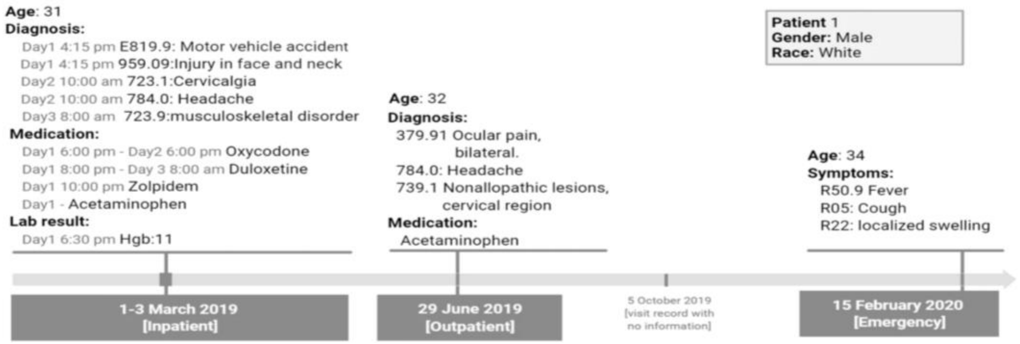
Longer time span, chronical conditions

Patient is a sequence of visits

Time interval irregular between visits

Each visit has a number of codes

Each codes are categorical variable

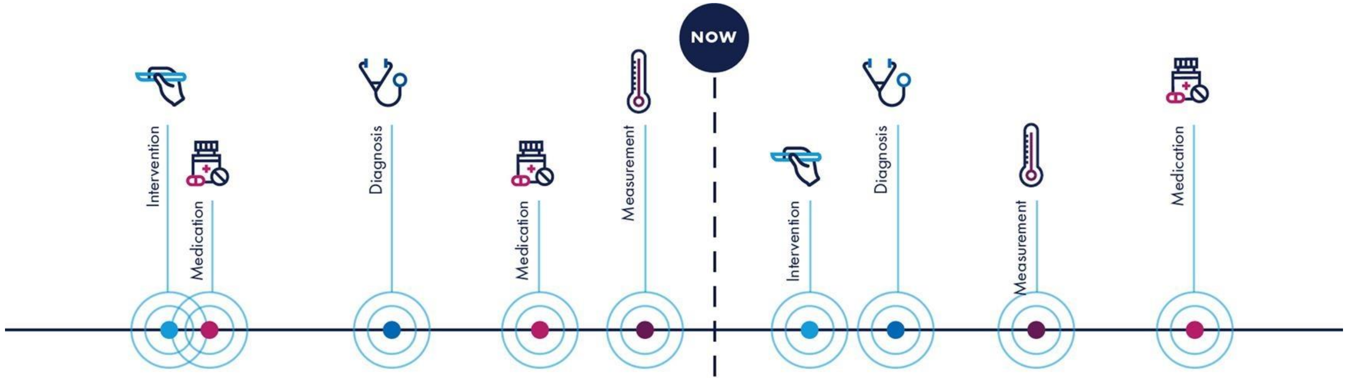



## CONTINUOUS MONITORING

Shorter time span, acute intense care, usually a single visit

Patient has observations at continuous times

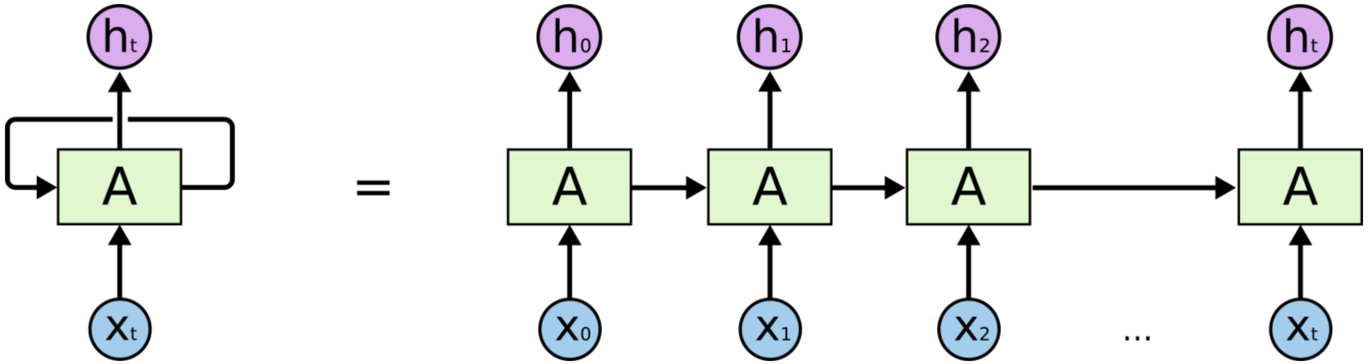
One measure per variable per window



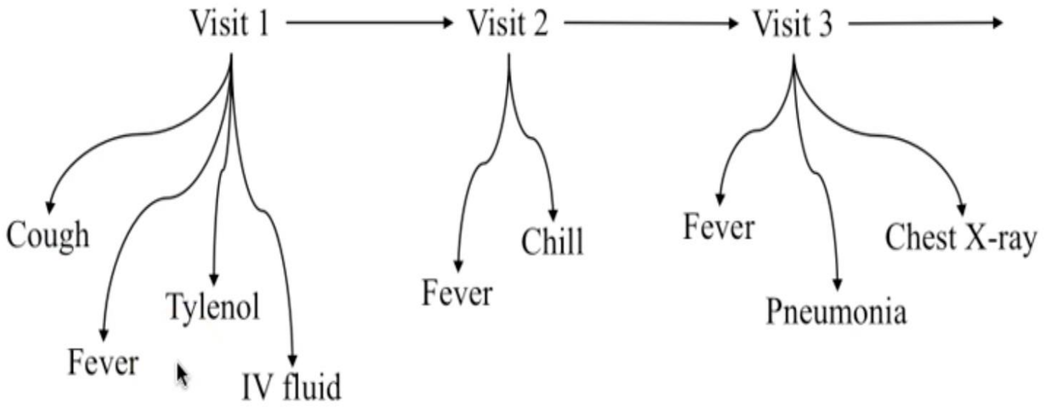
# EHR predictive modeling: Output data modalities

- Binary outcomes
- Survival (Binary outcome with a time horizon)
- Continuous variables (e.g., biomarkers)
- Drug concentration monitoring
- Multiple structured outcomes (e.g., length of stay)

# Recurrent Neural Network - RNN




**An unrolled recurrent neural network.**

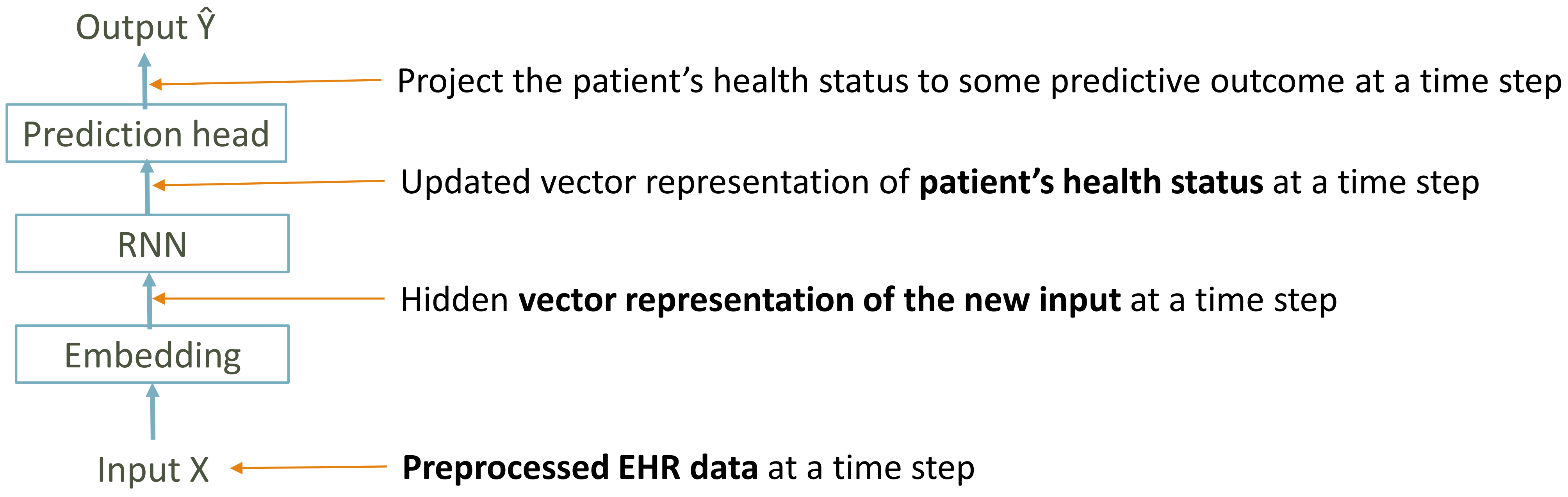



*https://www.youtube.com/watch?v=co3lTOSgFlA&feature=youtu.be*



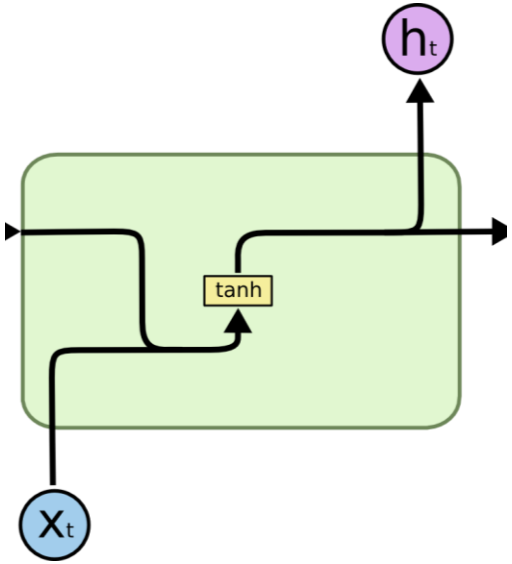
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

# The RNN Framework



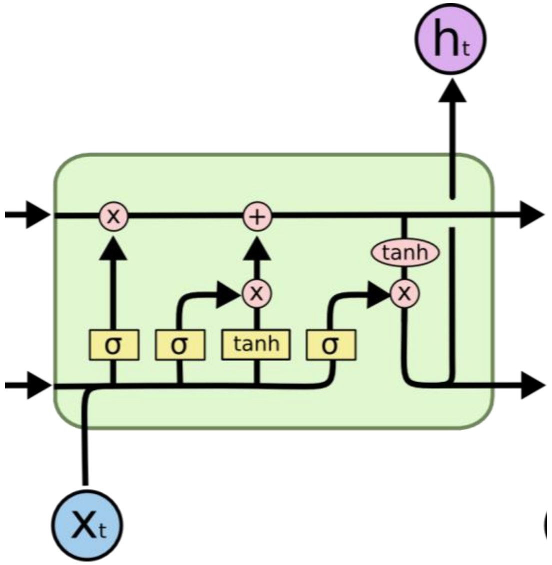
Vanilla

RNN




Long Short Term Memory

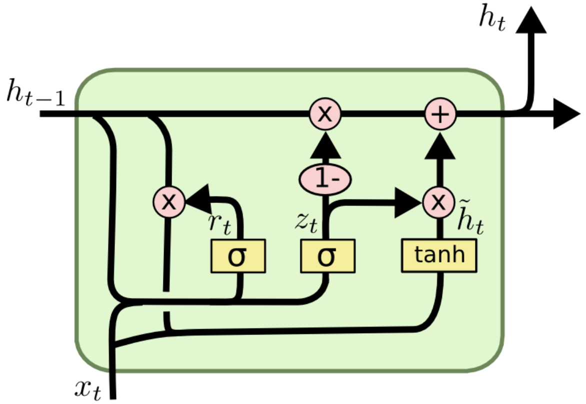
*(Hochreiter & Schmidhuber, 1997)*




Better memory for long sequences

Gated Recurrent Unit

*(Chung et al., 2014)*

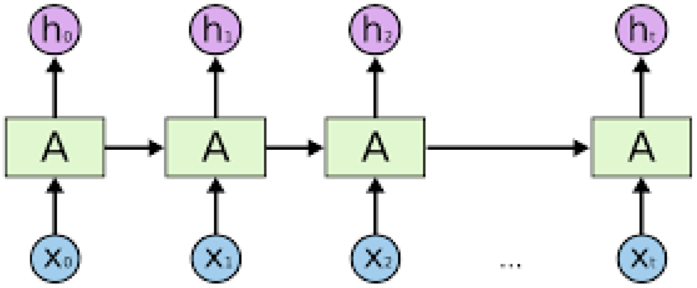



Computational efficient

# Baseline RNN cells

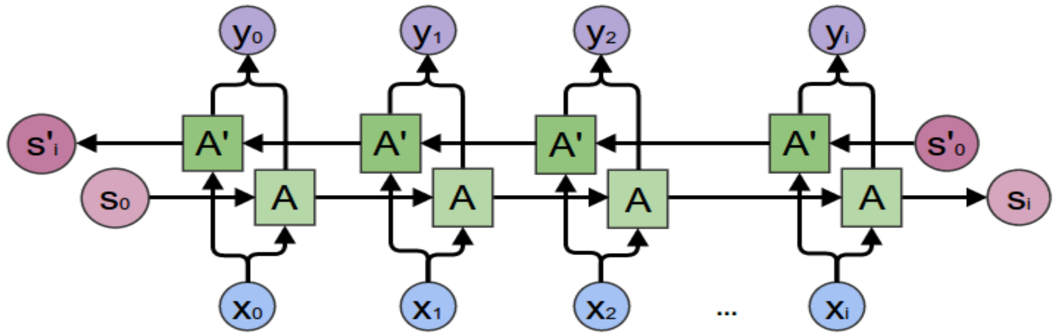
*LSTM and GRU. Images from Colah's blog*
*http://colah.github.io/posts/2015-08-Understanding-LSTMs*

Basic Unidirectional RNN




Bidirectional RNN

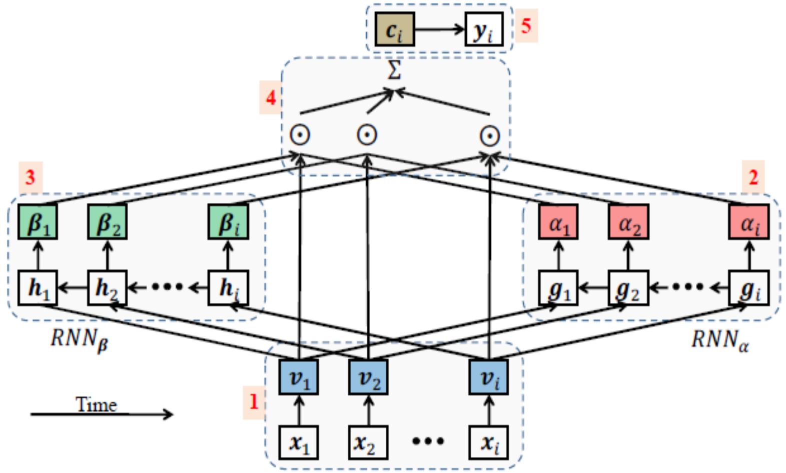
*(Schuster & Paliwal, 1997)*




Better representation of the context and eliminate ambiguity.

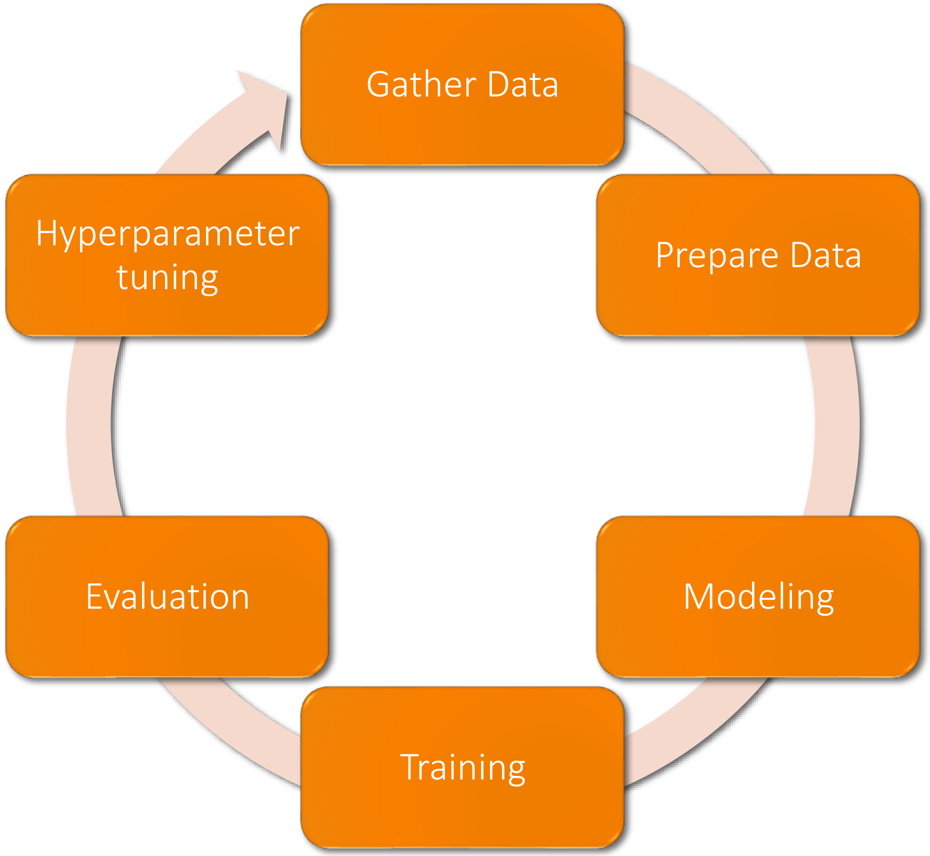
RETAIN model architecture

*(Choi et al., 2016)*




# more RNN structures

# Hyperparameter tuning




- Grid Search implemented in `GridSampler`
- Random Search implemented in `RandomSampler`
- Tree-structured Parzen Estimator algorithm implemented in `TPESampler`
- CMA-ES based algorithm implemented in `CmaEsSampler`
- Algorithm to enable partial fixed parameters implemented in `PartialFixedSampler`
- Nondominated Sorting Genetic Algorithm II implemented in `NSGAIISampler`
- A Quasi Monte Carlo sampling algorithm implemented in `QMCSampler`

# Explanation by Attribution score

*lens into the inner working of the models*

# Learning Objectives

Understanding the theories behind EHR predictive modeling using deep learning.

Learn the basic tools of deep learning to convert theory to practice.

Understand the basics of proper cohort definition.

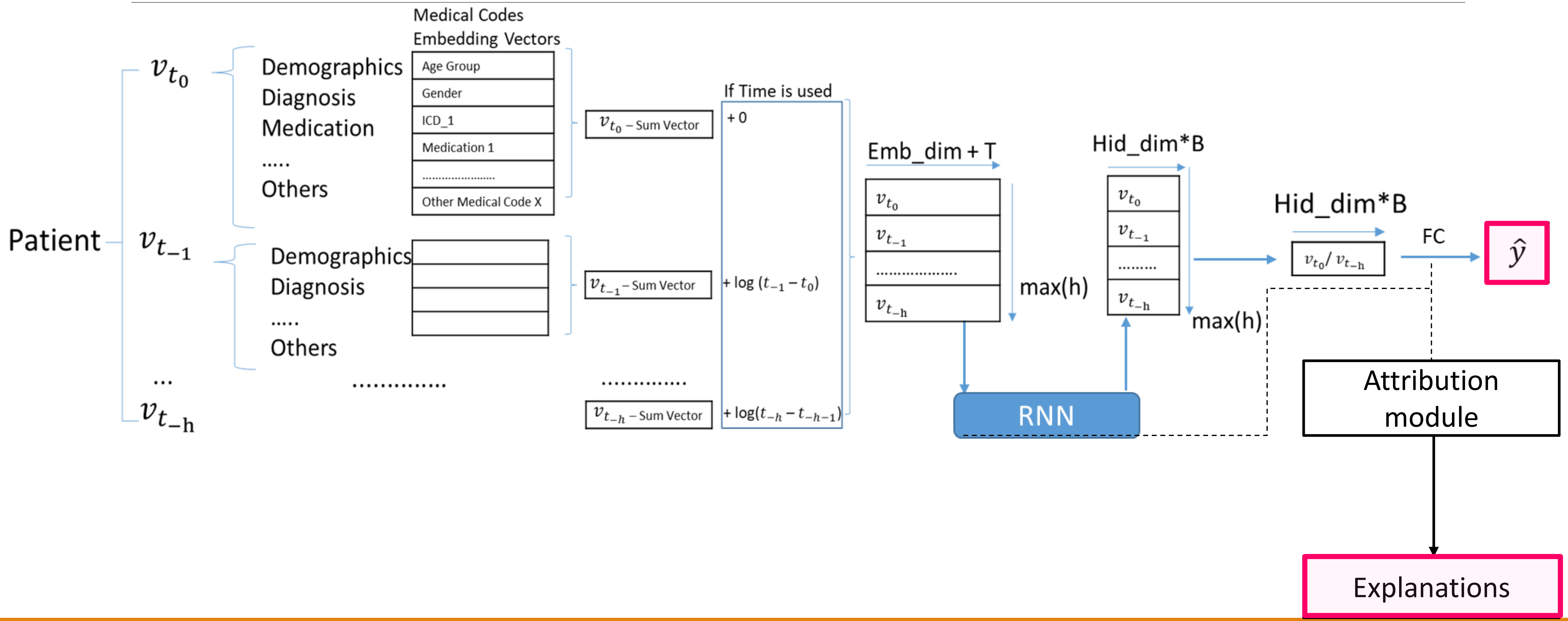
Practice data preparation and preprocessing

Practice RNN model training and evaluation for binary classification and survival prediction

Learn different techniques used for hyperparameter tuning.

Learn how to present model predictions as well as explanations using attribution mechanism.

# Explainable Pytorch_EHR



# Attribution mechanism

Assign a value to each of the input features

The value represent the contribution of the feature to the output of the model

Common methods includes LIME [1], SHAP [2], LRP [3], IG [4], etc.

[1] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.
[2] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Proceedings of the 31st international conference on neural information processing systems*. 2017.
[3] Montavon, Grégoire, et al. "Layer-wise relevance propagation: an overview." *Explainable AI: interpreting, explaining and visualizing deep learning* (2019): 193-209.
[4] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." *International Conference on Machine Learning*. PMLR, 2017.

# Integrated Gradient

Posthoc method

Computation doesn’t require modifying network structure

Has some good theoretical properties

# Integrated Gradient

$$f(X) - f(X_{baseline}) = \int_{\mathrm{X}_{baseline}}^{X} \sum \frac{\partial f}{\partial x_i} dx_i$$

Multivariate calculus

For each variable, the attribution is $\int \frac{\partial f}{\partial x} dx$

Can be calculated using simple Riemann sum

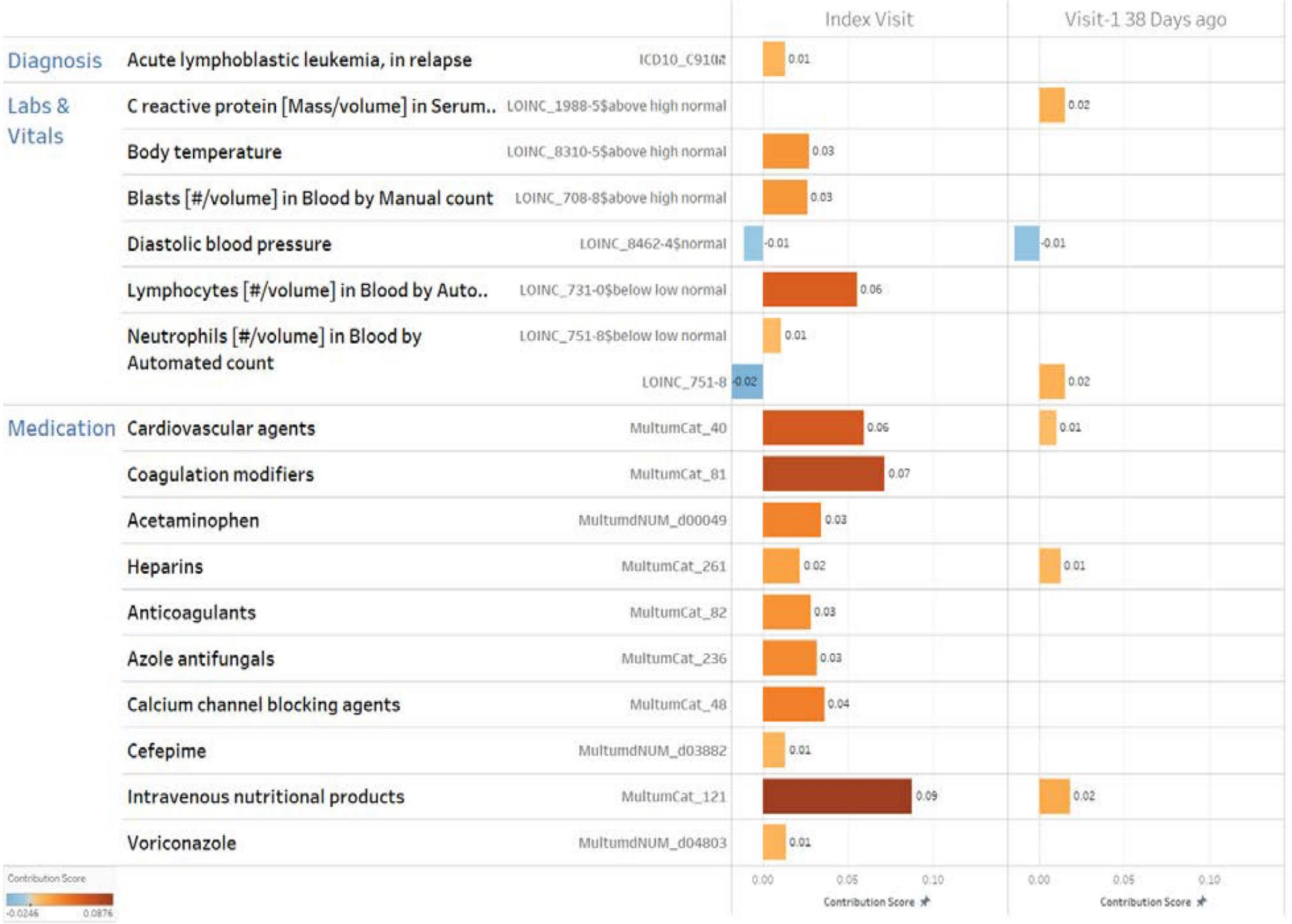
# Pytorch implementation

Make sure the feature tensor that you want to calculate attribute score on is a leaf node (detach if necessary)

Set requires_grad = True

Gradually increase the feature from baseline level to current level, accumulate the gradient.

# Example




Rasmy, Laila, et al. "Recurrent neural network models (CovRNN) for predicting outcomes of patients with COVID-19 on admission to hospital: model development and validation using electronic health record data." *The Lancet Digital Health* (2022).

# Pytorch_EHR for EHR predictive modeling

# Major components of EHR predictive modeling

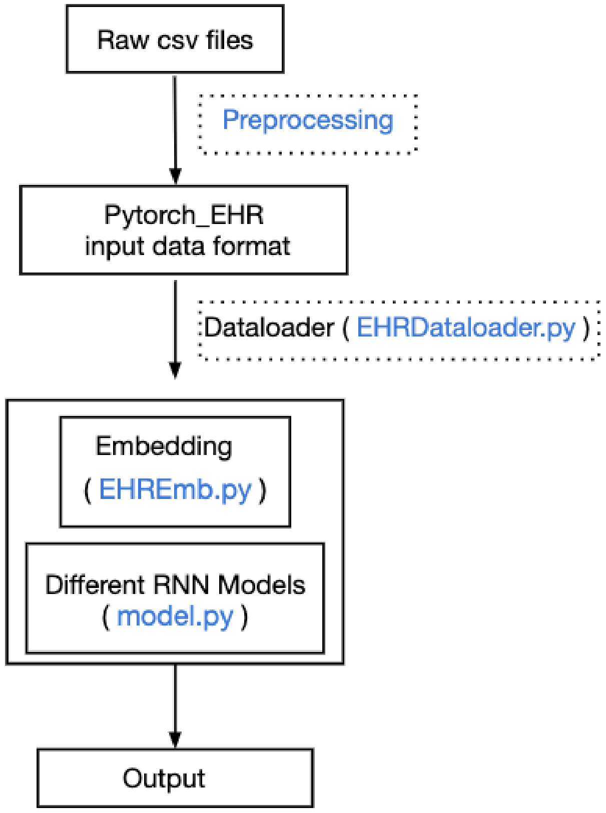
DATA PREPARATION

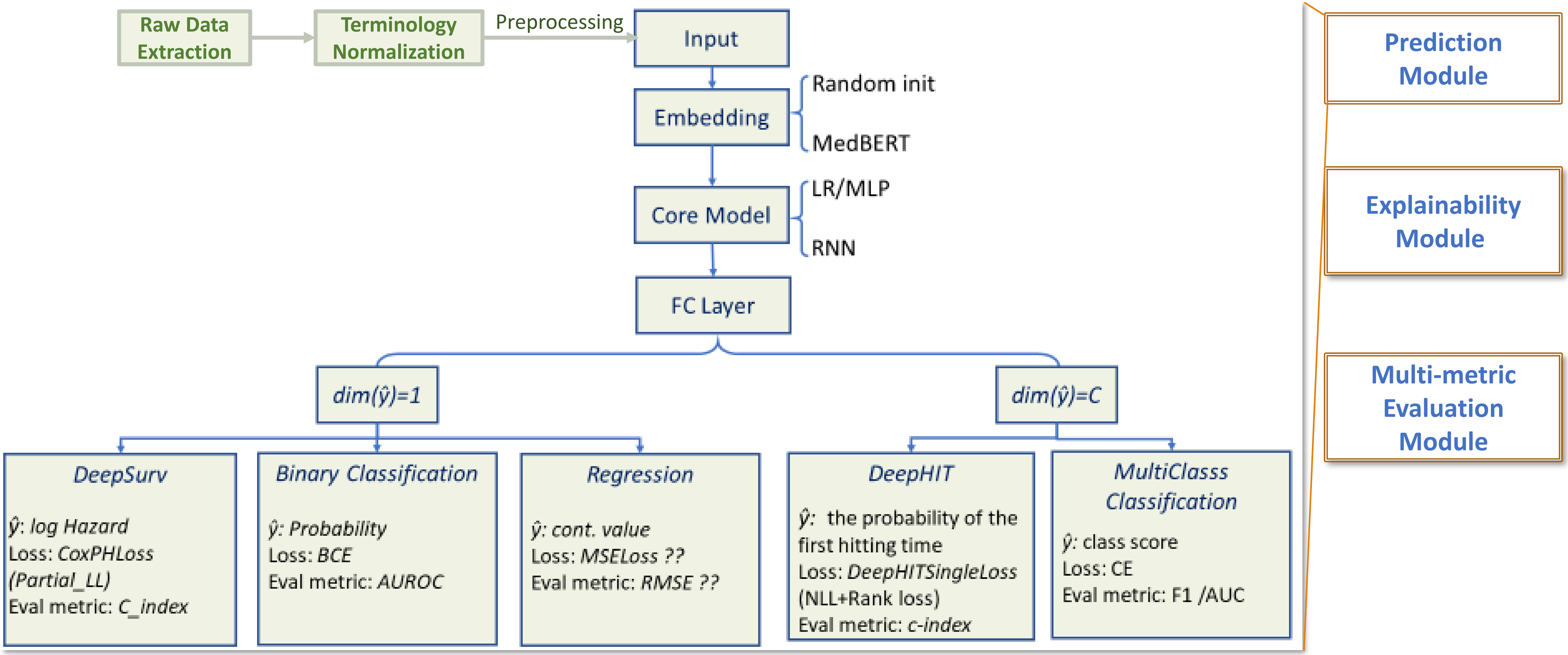
PREDICTION ENGINE

INTERPRETATION

# PyTorch_EHR: source codes based on PyTorch to analyze EHR

- Lower the bar of entering this field for researchers
- Provide efficient data loading
- Enable experimenting mix and match of components
- Deliver competitive performance






# Pytorch_EHR (v.3) Framework

# Section 2: Data Preparation

# Learning Objectives

Understanding the theories behind EHR predictive modeling using deep learning.

Learn the basic tools of deep learning to convert theory to practice.

Understand the basics of proper cohort definition.

Practice data preparation and preprocessing

Practice RNN model training and evaluation for binary classification and survival prediction

Learn different techniques used for hyperparameter tuning.

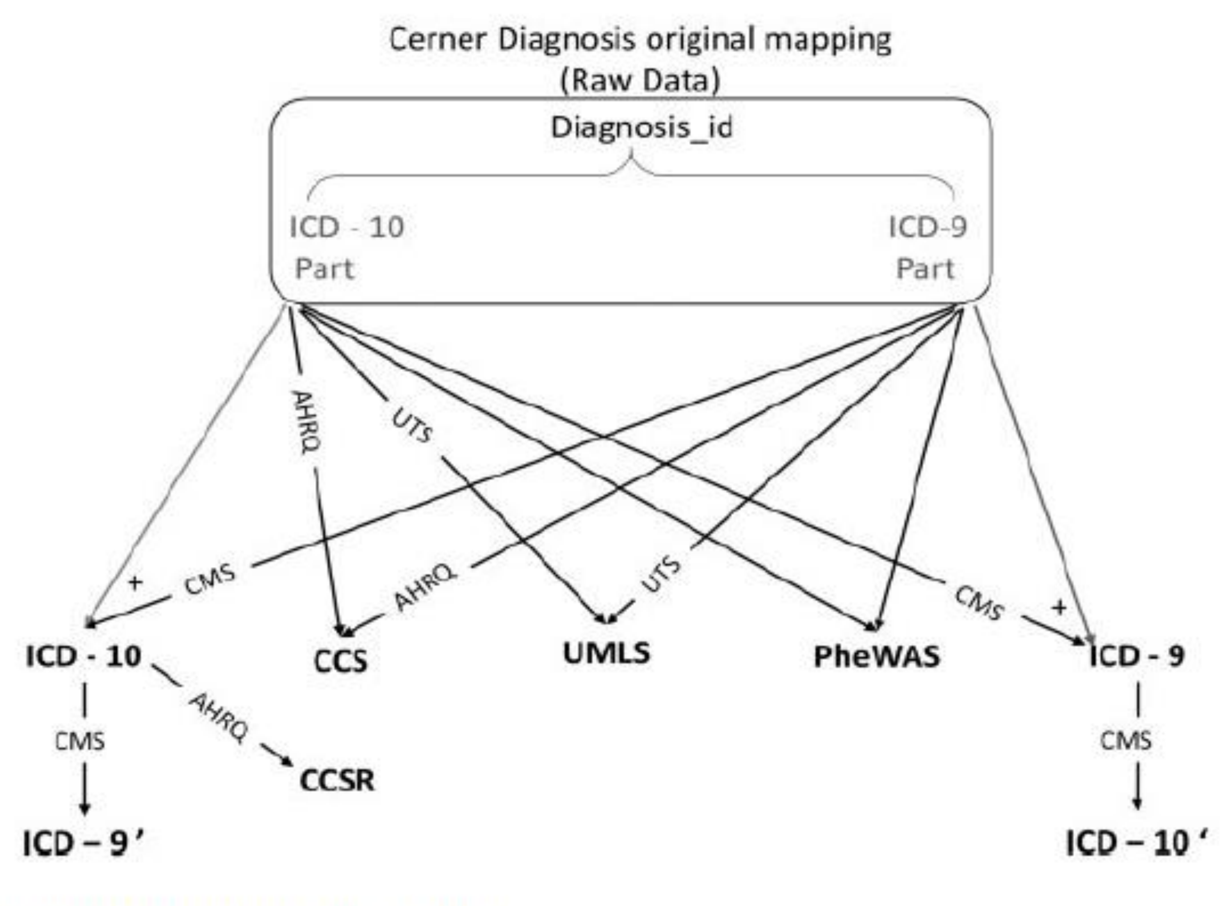
Learn how to present model predictions as well as explanations using attribution mechanism.

# Cohort Definition

Steps toward proper cohort definition:

1. Understand the clinical problem
2. Engage stakeholders (clinicians / users)
3. Clearly define your outcome based on how the model is intended to be used
   - What to predict
   - When to predict
4. Understand the data
   - strength
   - Limitations
5. Decide on your inclusion / exclusion criteria
   - Basic data cleaning

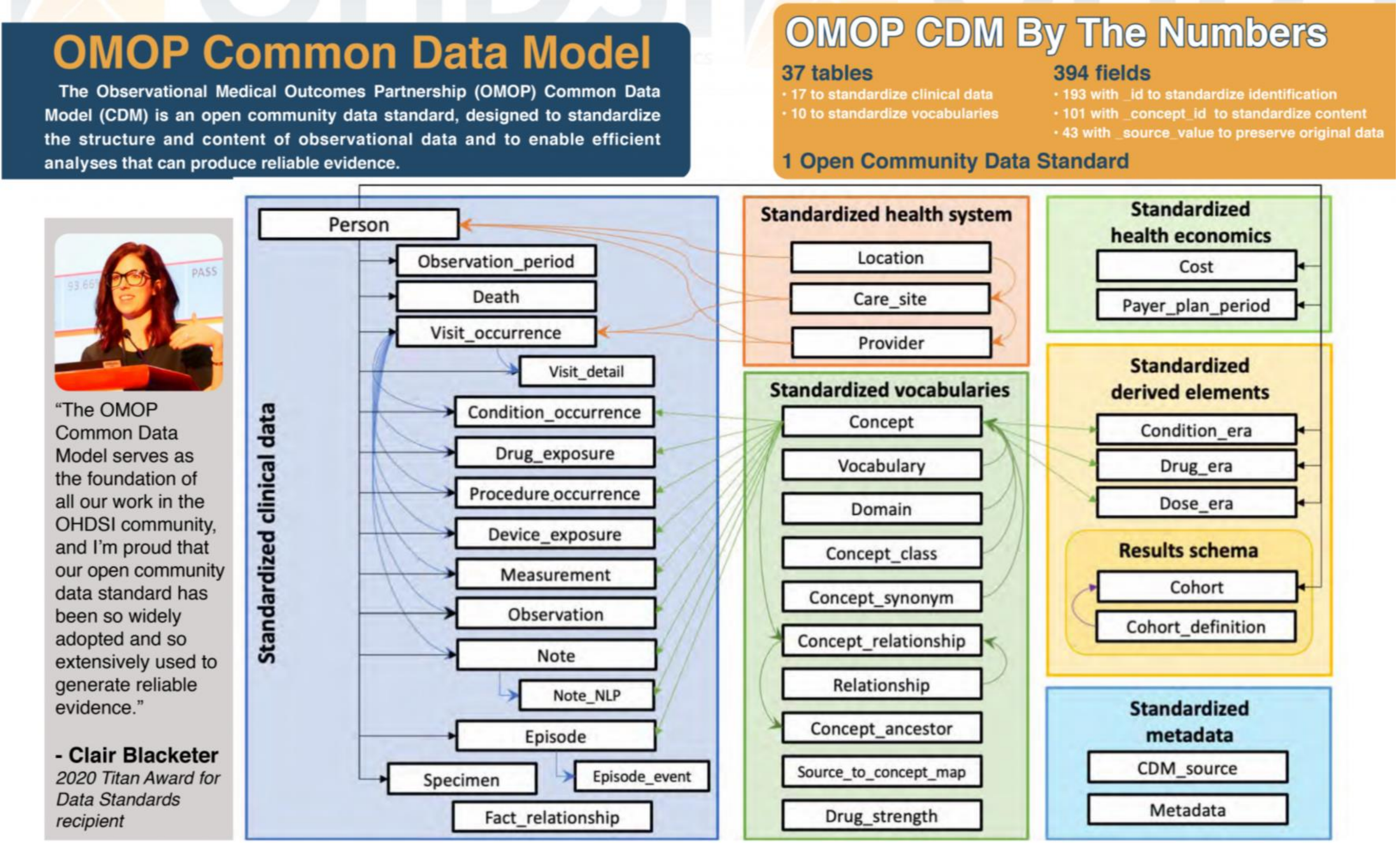
Our model can consume all data, so no need for further feature selection




| Diagnosis terminology | Diabetes heart failure cohort (DHF) Number of unique codes | LR | RNN | Pancreatic cancer cohort (PC) Number of unique codes | LR | RNN |
|---|---|---|---|---|---|---|
| **Raw data (ICD -9 +ICD-10)** | 26,427 | **80.61** | **85.48** (0.10) | 13,071 | **80.30** | **81.43** (0.37) |
| **CCS-single level** | 284 | 78.07 | 82.96 (0.15) | 253 | 77.23 | 79.03 (0.36) |
| **CCSR** | 538 | 78.87 | 84.17 (0.21) | 538 | 77.92 | 79.63 (0.34) |
| **ICD-9** | 11,187 | 80.12 | 85.20 (0.13) | 7,055 | **79.15** | 80.78 (0.32) |
| **ICD-10** | 22,893 | 79.78 | 84.35 (0.20) | 13,620 | 78.95 | 79.27 (0.44) |
| **PheWAS** | 1,820 | **80.71** | **85.87** (0.10) | 1,715 | 78.82 | **81.15** (0.31) |
| **UMLS CUI** | 29,491 | **81.15** | **85.55** (0.06) | 14,551 | **80.53** | **82.24** (0.29) |

# Terminology Normalization

**https://github.com/ZhiGroup/terminology_representation**

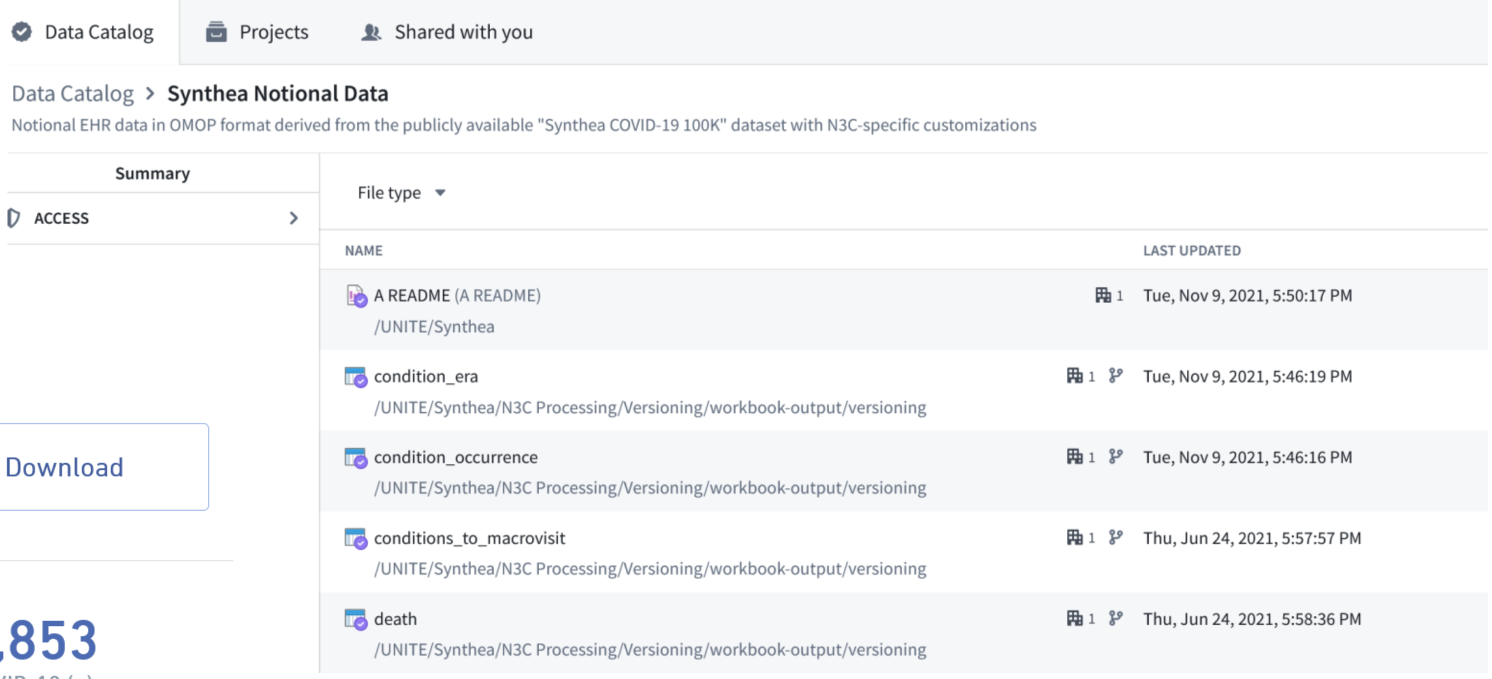


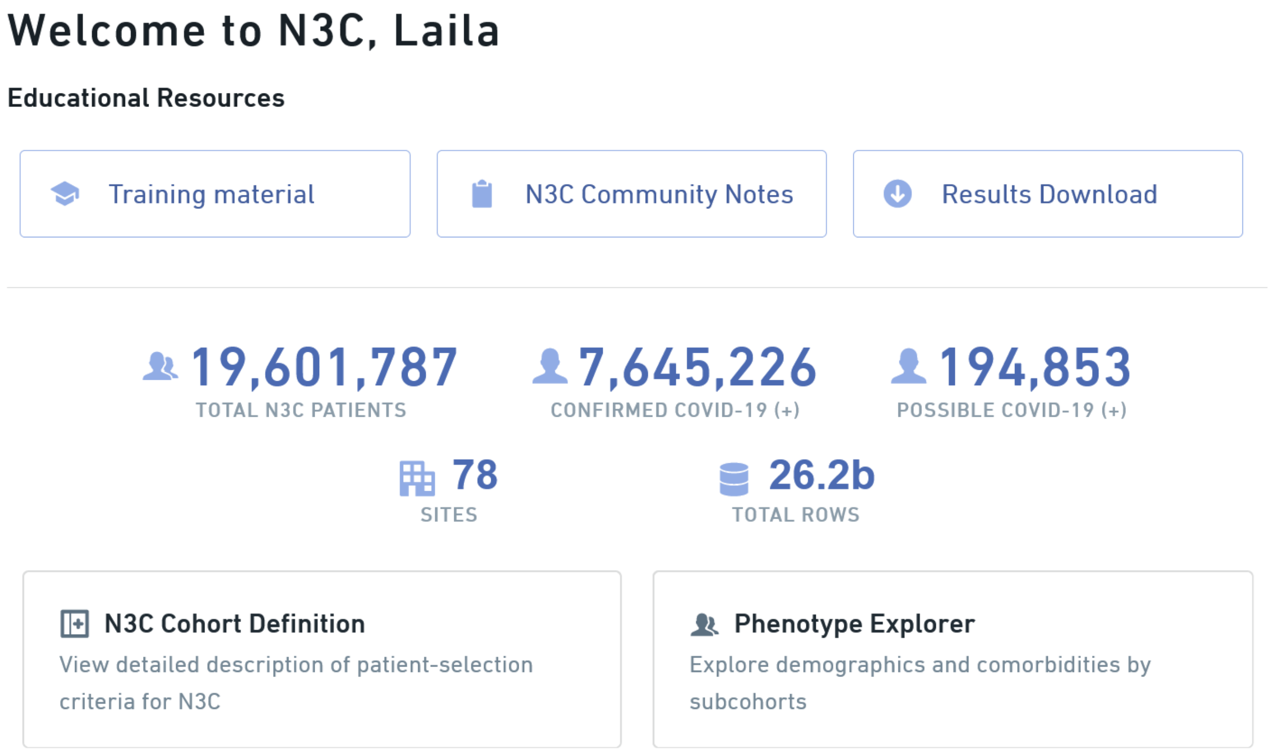

http://ohdsi.github.io/CommonDataModel/cdm54.html

https://athena.ohdsi.org/vocabulary/list

# OMOP Standard Concepts

https://www.ohdsi.org/data-standardization/




# N3C Data

https://unite.nih.gov/workspace/compass/data-catalog

# Let's Practice Now

https://github.com/ZhiGroup/pytorch_ehr/tree/ICHI_2023

1. Cohort definition and data extraction from the EHR database
2. Data reformatting to be efficiently consumed by Pytorch_ehr

# Section 3 : Model Training & Evaluation

# Learning Objectives

Understanding the theories behind EHR predictive modeling using deep learning.

Learn the basic tools of deep learning to convert theory to practice.

Understand the basics of proper cohort definition.

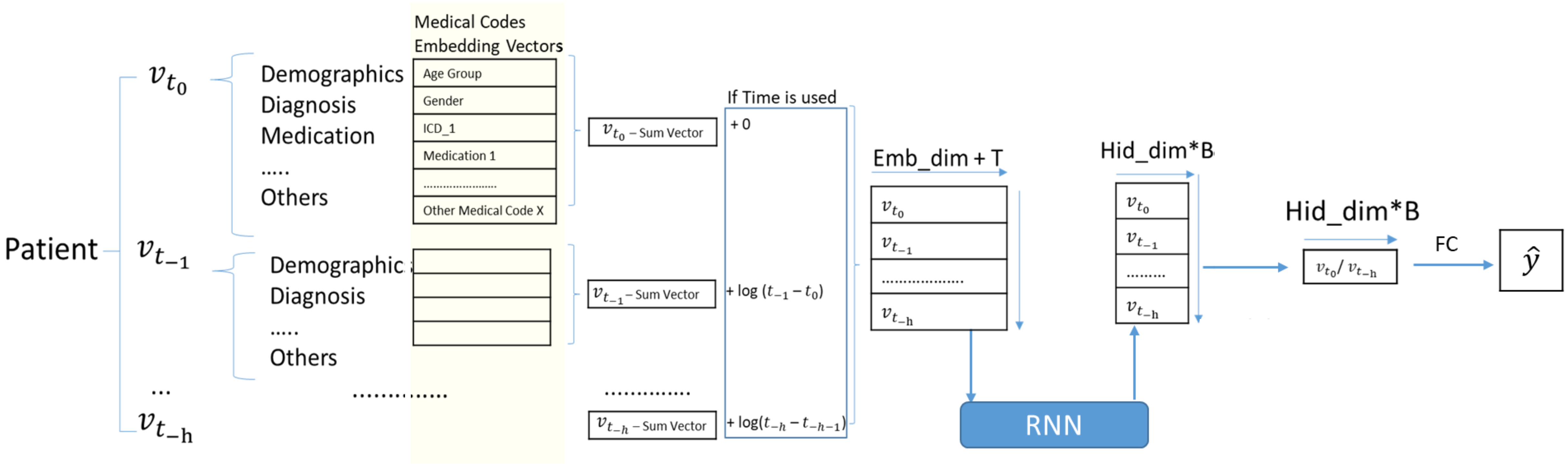
Practice data preparation and preprocessing

Practice RNN model training and evaluation for binary classification and survival prediction

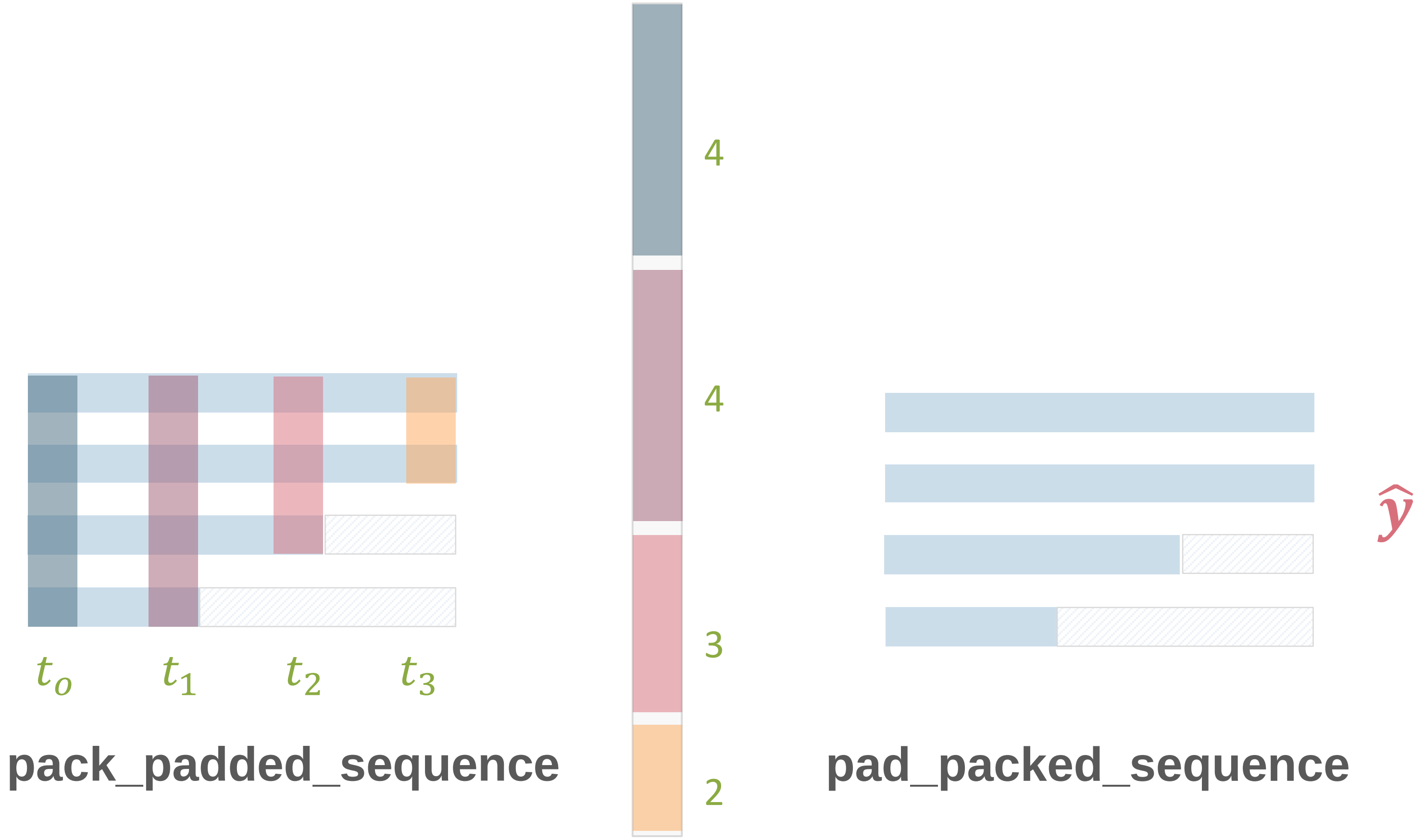
Learn different techniques used for hyperparameter tuning.

Learn how to present model predictions as well as explanations using attribution mechanism.

# Pytorch_EHR: under the hood



# Packed Sequence

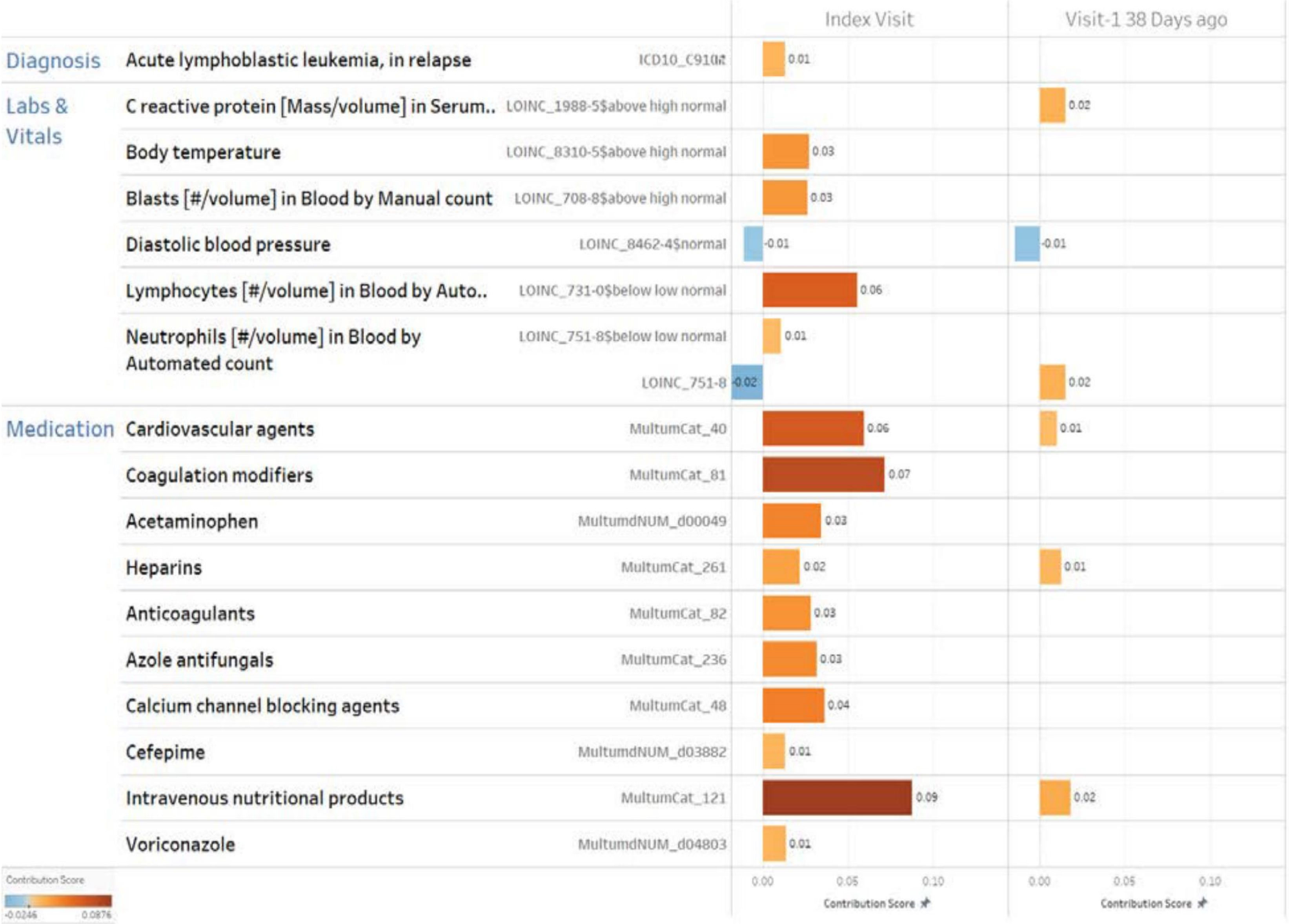


# Let's Practice Now

1. Model training for binary classification
2. Model training survival prediction
3. Hyperparameter tuning
4. Model evaluation

Back to the colab

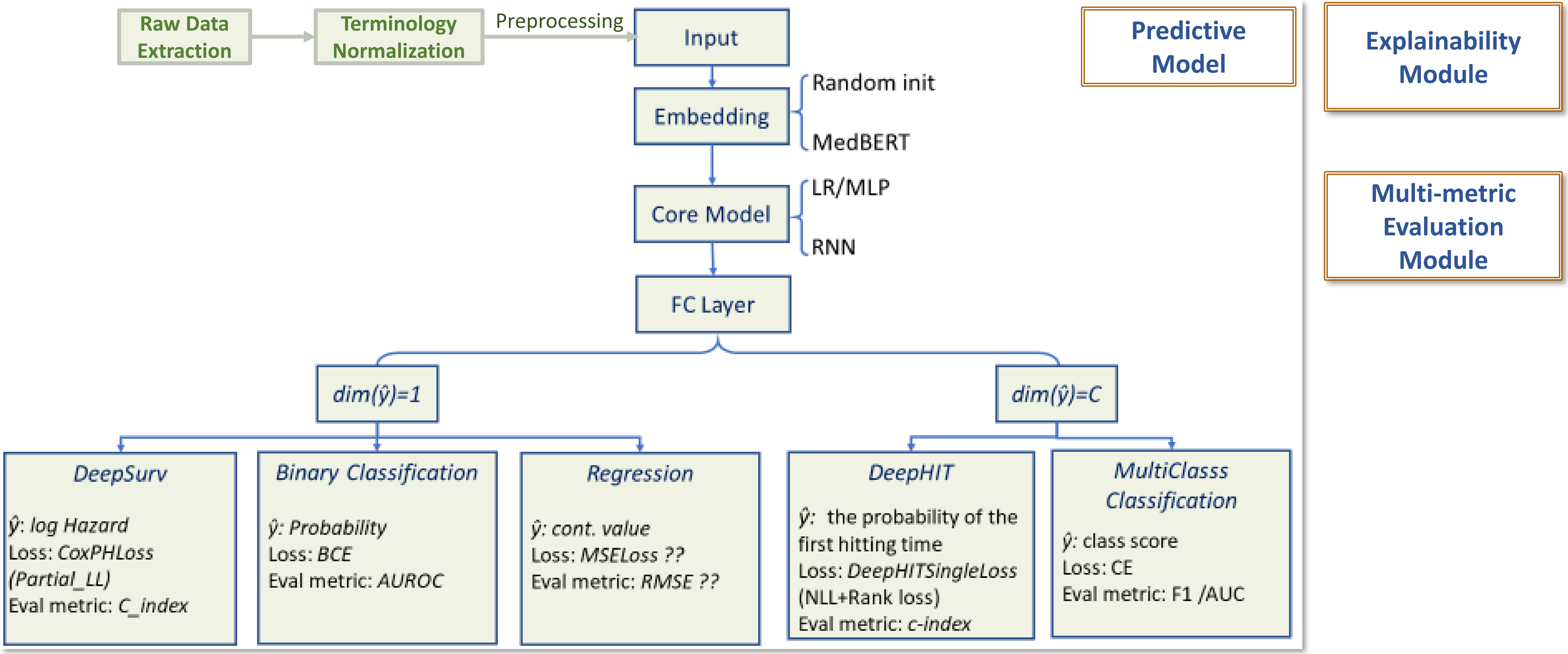
# Contribution Scores Visulaization Example




Rasmy, Laila, et al. "Recurrent neural network models (CovRNN) for predicting outcomes of patients with COVID-19 on admission to hospital: model development and validation using electronic health record data." *The Lancet Digital Health* (2022).

# Thank you!

HTTPS://GITHUB.COM/ZHIGROUP/PYTORCH_EHR/




# Pytorch_EHR v.3 Framework