UNIVERSITÄT BONN

SMART DATA ANALYTICS
FROM DATA TO KNOWLEDGE

Fraunhofer
IAIS

Lab Semantic Data Web Technologies

Topic: Distillation of DNN Networks into Gradient Boost tree model

Md. Emtazul Haque, Jelena Trajkovic, Anam Siddiqi

Supervisor: Firas Kassawat

Final presentation

Motivation

Interpretable Deep Models for ICU Outcome Prediction

Zhengping Che, Sanjay Purushotham, Robinder Khemani, Yan Liu

- Decision trees - successfully employed in the health care domain, due to ease of interpretation.
- Problem - they can overfit easily.
- Question - How to learn interpretable models from well trained deep network models?
- Approach proposed in the paper, and our guideline: employ mimic learning to learn an interpretable model.
- Goal: develop data driven solution, with performance comparable to deep learning models, easy interpretable by healthcare professionals.

Mimic Method Training Pipeline 1

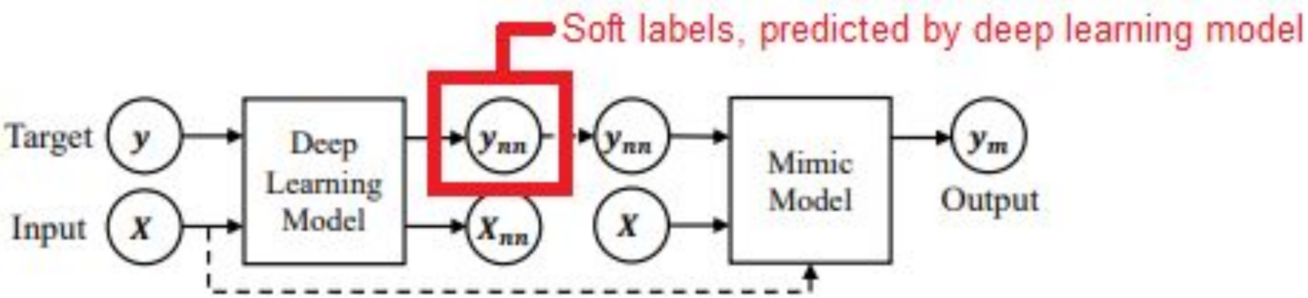
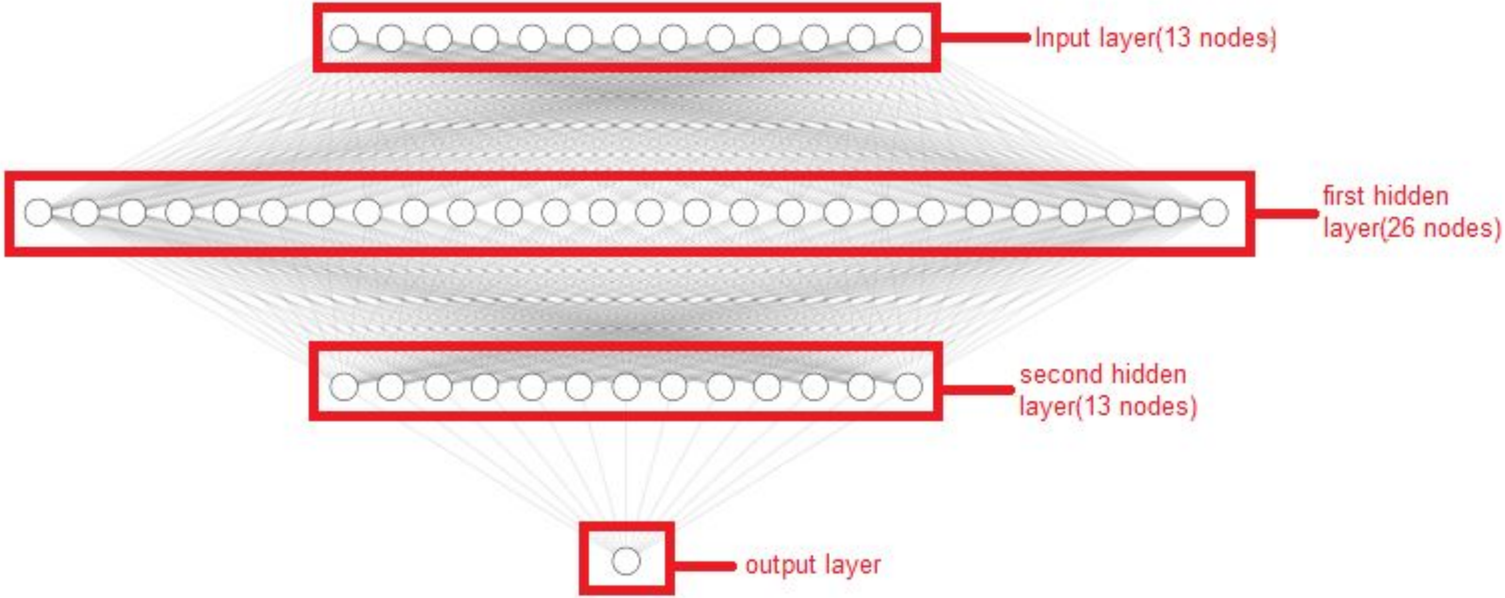


Figure 2: Illustration of mimic method training pipeline 1.

Deep Learning Model Implementation



Dataset

We have used a heart disease dataset for the prediction of presence of heart disease. The dataset contains 1035 samples, and it is well balanced.

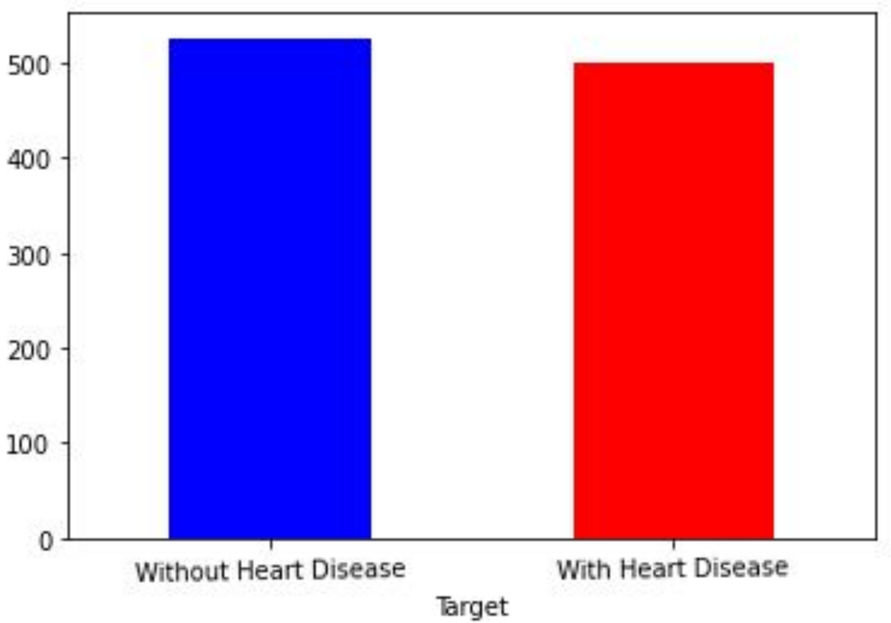
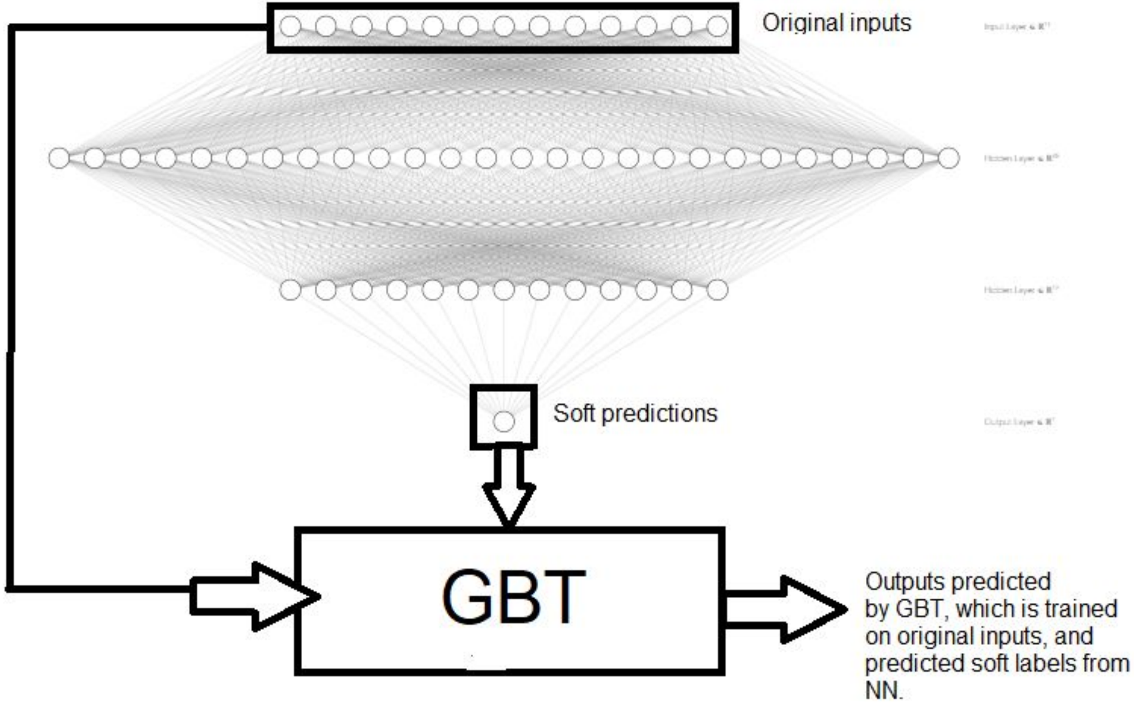
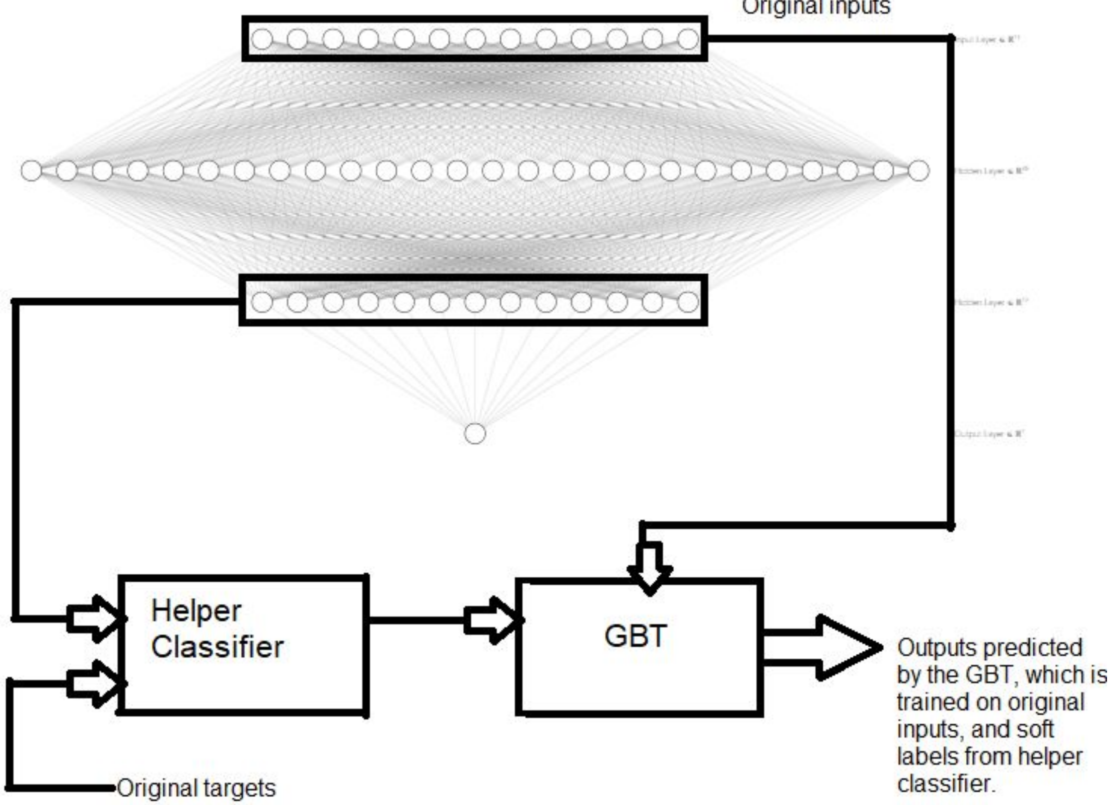


Illustration of Pipeline 1



Outputs predicted by GBT, which is trained on original inputs, and predicted soft labels from NN.

Illustration of Pipeline 2



Outputs predicted by the GBT, which is trained on original inputs, and soft labels from helper classifier.

Gradient Boosting Tree(GBT)

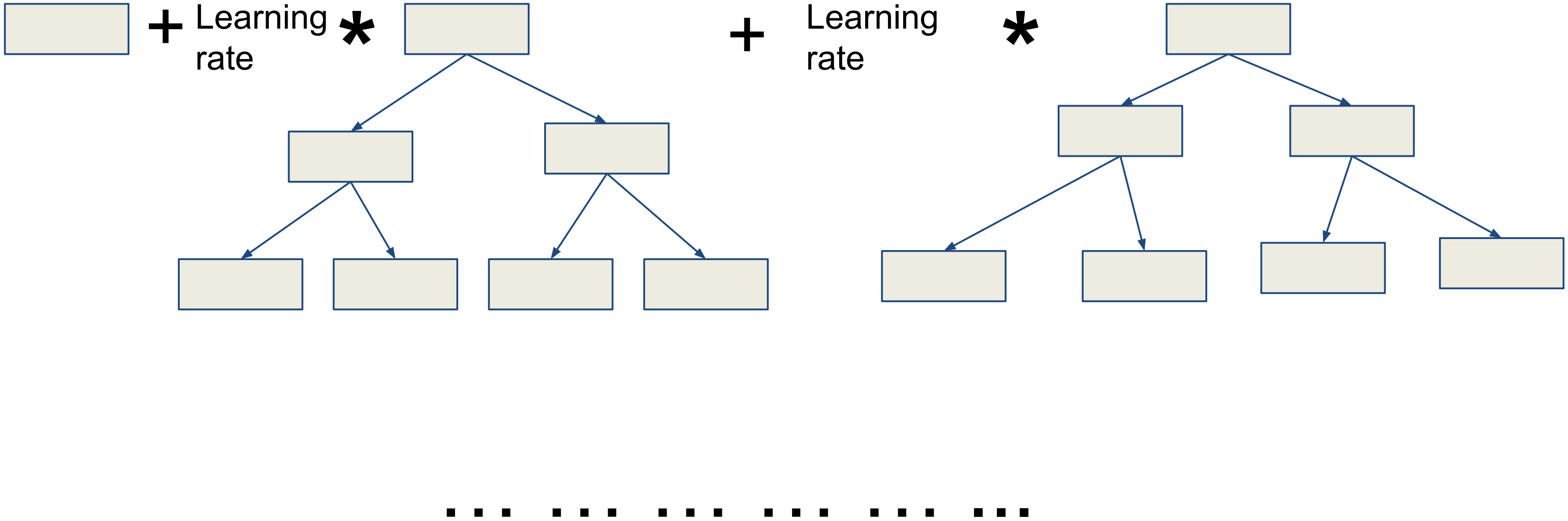
- Gradient boosting is a machine learning technique for regression and classification.

- Produces a strong prediction model as a collection of weak prediction models (typically decision trees).

- It builds the model in a stage-wise fashion and it generalizes them by optimizing an arbitrary differentiable loss function.

- It has several parameters that can be calibrated according to problem domain
  - Learning rate
  - Maximum tree depth
  - Maximum number of trees

Gradient Boosting Tree(GBT)



+ Learning rate *

+ Learning rate *

... ...

Fig: Illustration of Gradient Boosting Tree

GBT Implementation

- **“scikit-learn”** provides an implementation of GBT classifier and regressor in python.

- **XGBoost** provides an implementation of Extreme Gradient Boosting on “scikit-learn” API in python.

GBT Parameters

- Configure GBT.
  - Learning rate = 0.1
  - Maximum tree depth = 3
  - Maximum number of trees = 100
  - Objective function = softprob

GBT Training in Pipeline 1

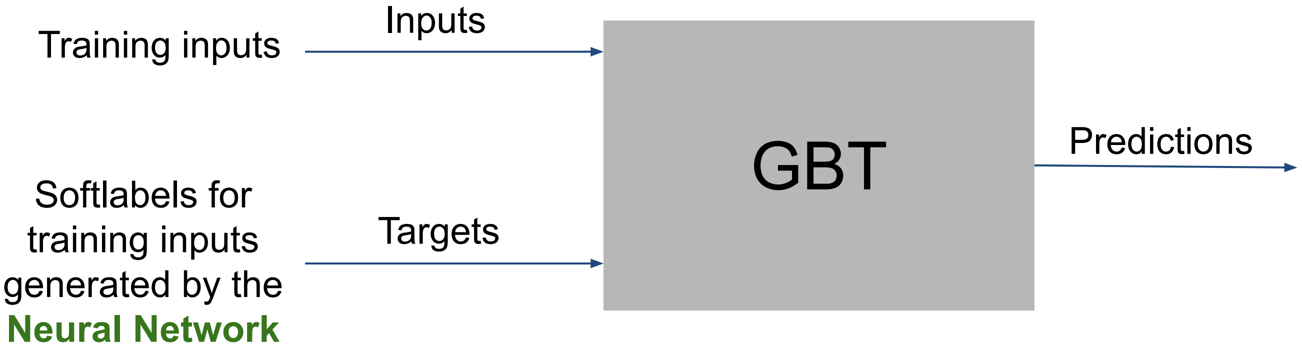
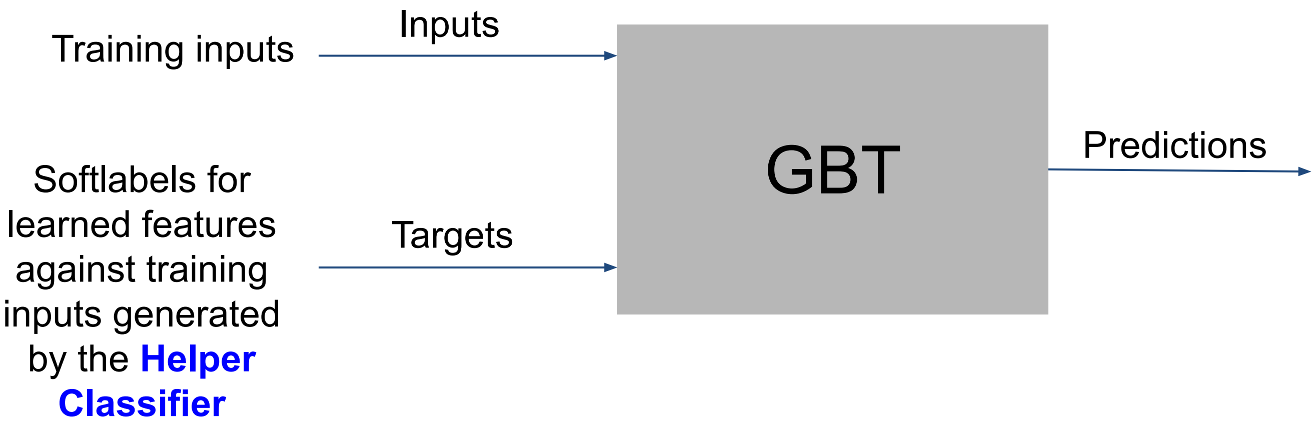

Training inputs
Inputs
Softlabels for training inputs generated by the **Neural Network**
Targets
GBT
Predictions

Fig: GBT training in pipeline 1

GBT Training in Pipeline 2



Inputs

Training inputs

GBT

Predictions

Targets

Softlabels for learned features against training inputs generated by the **Helper Classifier**

Fig: GBT training in pipeline 2

Visualization of GBT

- ❖ The XGBoost Python API provides a function **plot_tree()** for plotting decision trees
- ❖ Dataset for Cardiovascular patients have been used

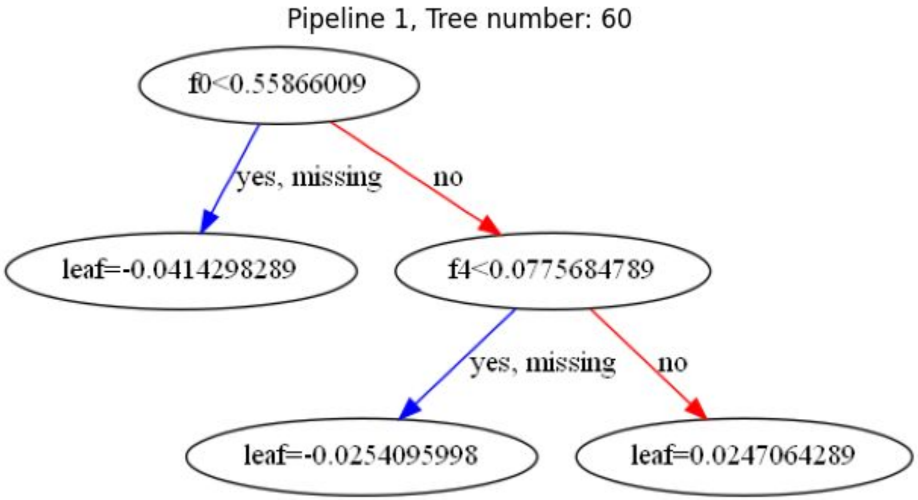


Fig : XGBoost Plot of Single Decision Tree

Visualization of GBT

❖ Some of the decisions trees which have been plotted.

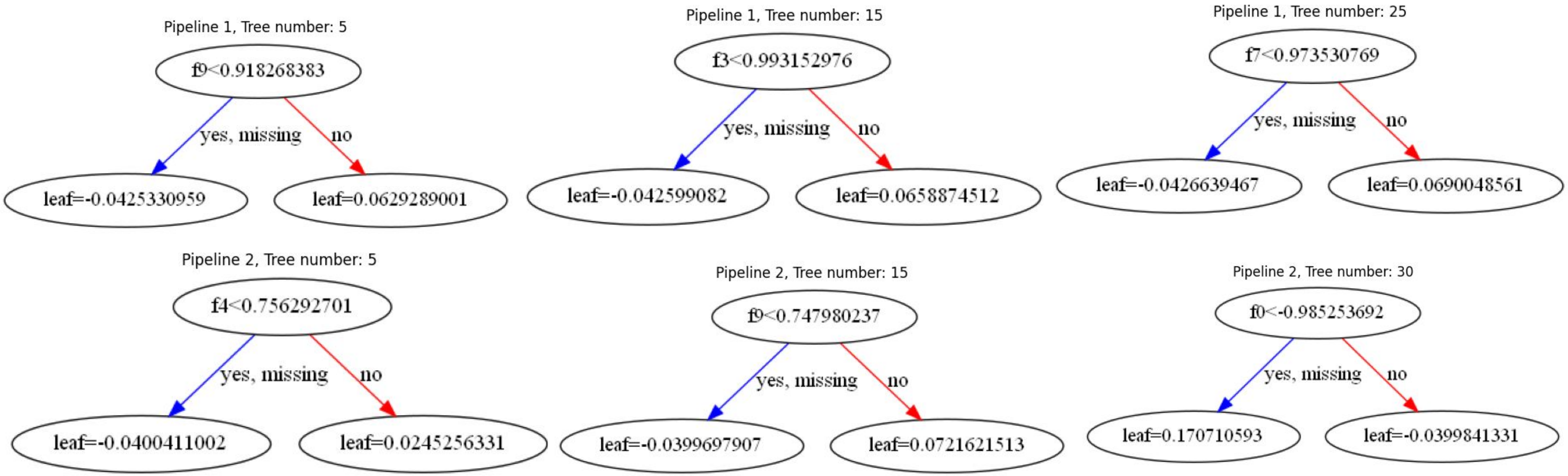
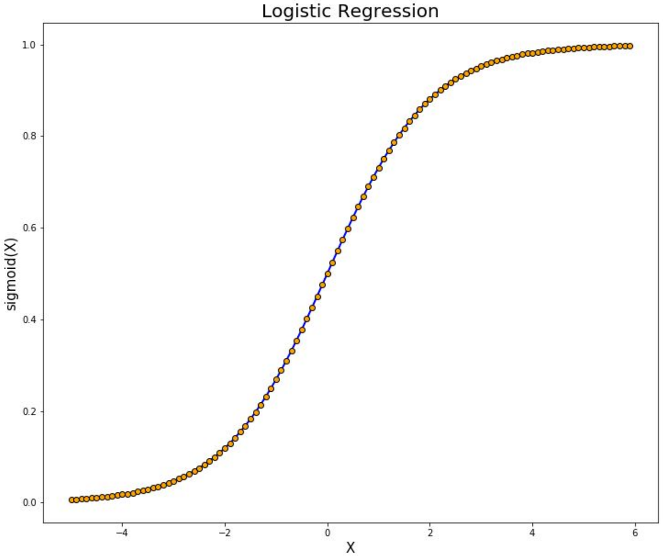


Fig : XGBoost Plot of Decision Trees plotted from code

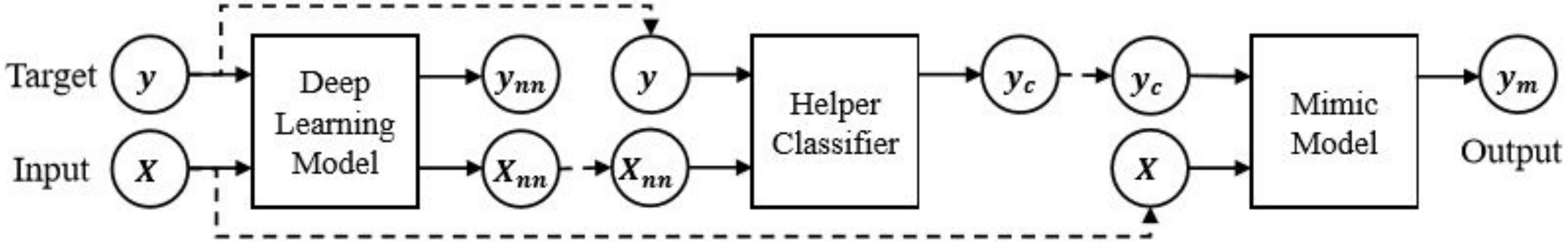
Helper Classifier

- Logistic Regression Classifier is used as a helper classifier in Pipeline 2
- Dataset for Cardiac patients have been used
- Input: Output from neural network



https://towardsdatascience.com/logistic-regression-explained-9ee73cede081

Helper Classifier



In the second pipeline, instead of taking soft labels from DNN, we are taking the learned features, extracted from the highest hidden layer. We then feed learned DNN features into the Helper Classifier, to predict the original label. We are taking soft predictions from the classifier. We train GBT model with original input and obtained soft labels.

In both pipelines we train GBT model with soft labels. We will see later, that we will obtain different accuracy.

Result Analysis

OUTPUT:

Model	**Accuracy**
Neural Network Accuracy	99.026
GBT(with helper classifier) Accuracy	94.481
GBT(only soft labels) Accuracy	96.429
GBT(only hard labels) Accuracy	95.455

Summary

✓

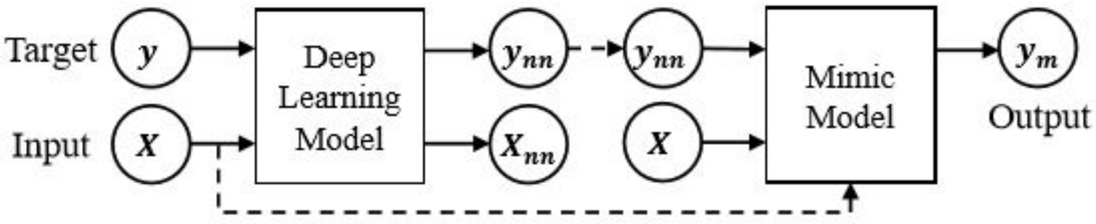


Fig: Illustration of mimic method training pipeline 1 *(Interpretable Deep Models for ICU Outcome Prediction)*

✓

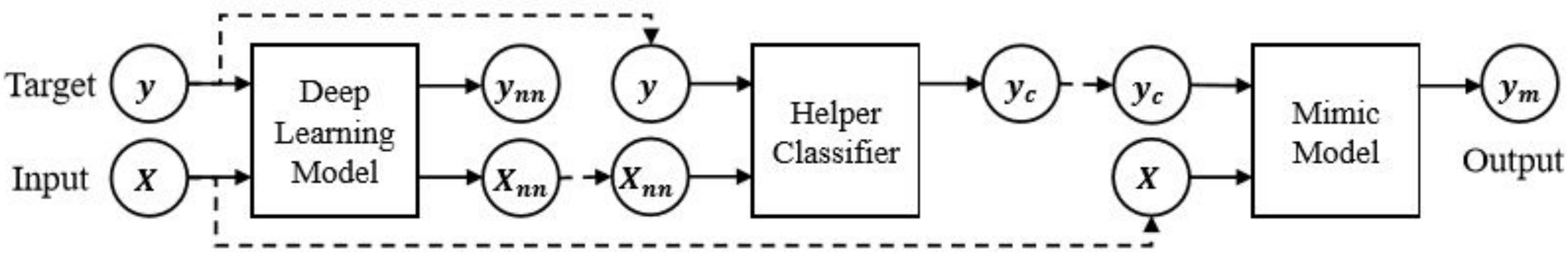


Fig: Illustration of mimic method training pipeline 2 *(Interpretable Deep Models for ICU Outcome Prediction)*

Code Documentation

The PDF file consists of the following:

- ❖ Abstract
- ❖ Steps to run the code in IDE
- ❖ Function definitions generated using PyDoc
- ❖ Decision Trees visualization using XGBoost

It contains all the details about the tasks implemented from the paper

References

- ➢ https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

- ➢ https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html

- ➢ https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn

Thank You!