The Clinician Engineer Hub

Development of an Explainable Machine Learning Model for Predicting 10-Year Risk of Death in NHANES I Participants: A Clinician Engineer's Guide to Synergizing Clinical Expertise, Engineering Innovation, and Data Science

UNIVERSITY OF CAMBRIDGE

Yosra Magdi Mekki (1,2), Neel Sharma (2)

Qatar University (1)
Clinician Engineer Hub, University of Cambridge (2)

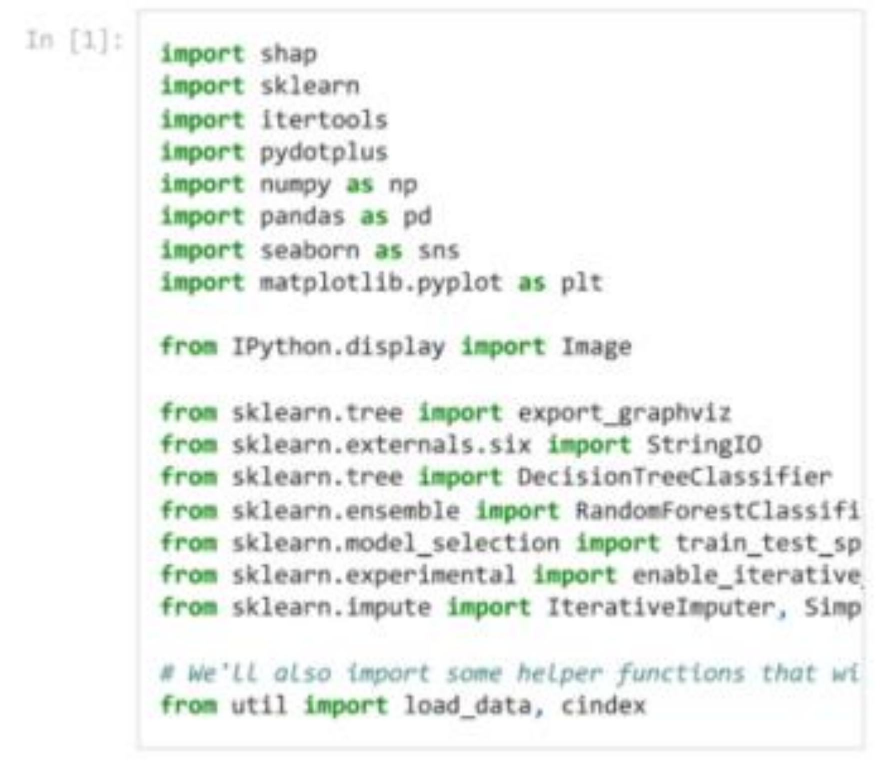
كلية الطب College of Medicine
QATAR UNIVERSITY جامعة قطر
Member of HEALTH الصحة عضو في

SCAN ME

Introduction

This is a summarized walk-through case study by the Clinician Engineer Hub, University of Cambridge, aimed at inspiring doctors and engineers interested in using tree-based models to predict the 10-year risk of death using the NHANES I epidemiology dataset. The study explores the application of decision trees, random forests, and gradient boosting in developing and evaluating risk prediction models. The dataset contains demographic and clinical information of hospital patients, along with the outcome of whether or not they died within 10 years. The dataset is split into a development set and a test set, with the dev set further divided into a training and validation set to train and tune the models. The performance of these models is evaluated using various metrics, such as accuracy, sensitivity, specificity, and the area under the receiver operating characteristic curve (AUC-ROC). This case study is part of a series of studies that demonstrate the potential of tree-based models and the NHANES dataset in improving patient outcomes.

Import Packages



```
In [1]:
import shap
import sklearn
import itertools
import pydotplus
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from IPython.display import Image

from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifi
from sklearn.model_selection import train_test_sp
from sklearn.experimental import enable_iterative
from sklearn.impute import IterativeImputer, Simp

# We'll also import some helper functions that wi
from util import load_data, cindex
```

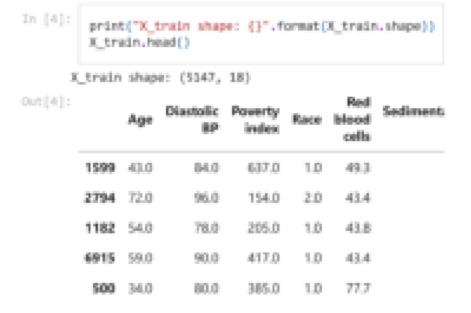
This code imports the packages shap, sklearn, itertools, pydotplus, IPython.display, numpy, pandas, seaborn, and matplotlib.pyplot, and sets up the matplotlib plotting environment to work within Jupyter notebooks.

Exploring Dataset and Dealing with Missing Data

The NHANES I Epidemiologic Follow-up Study (NHEFS) is a national longitudinal study that was jointly initiated by the National Center for Health Statistics and the National Institute on Aging in collaboration with other agencies of the Public Health Service. A recent study published in BMC Cardiovascular Disorders in 2021 used the NHANES I epidemiology dataset to develop prediction models for chronic obstructive pulmonary disease (COPD) risk. The study included a total of 3,226 COPD patients retrieved from NHANES 2007-2012, which were divided into training and testing sets

The study utilized multivariable logistic regression and random forest analyses for the prediction models. The predictive performance of the models was assessed using receiver operating characteristic (ROC) curves, area under the curves (AUC), and internal validation.

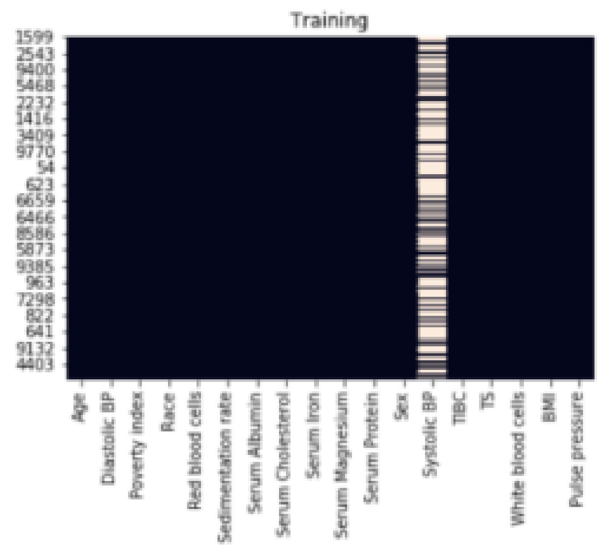
The NHANES I epidemiology dataset, containing various features of hospital patients and their outcomes, is loaded in the next cell. The dataset is divided into a development set (dev set) for developing risk models, and a test set for testing models. The dev set is further split into a training and validation set with a 75/25 split, respectively, to train and tune the models, using a set ra ndom state for reproducibility. Use "load_data(10)" to load the dataset.



```
In [4]:
print("X_train shape: {}".format(X_train.shape))
X_train.head()

X_train shape: (5147, 18)

Out[4]:
```

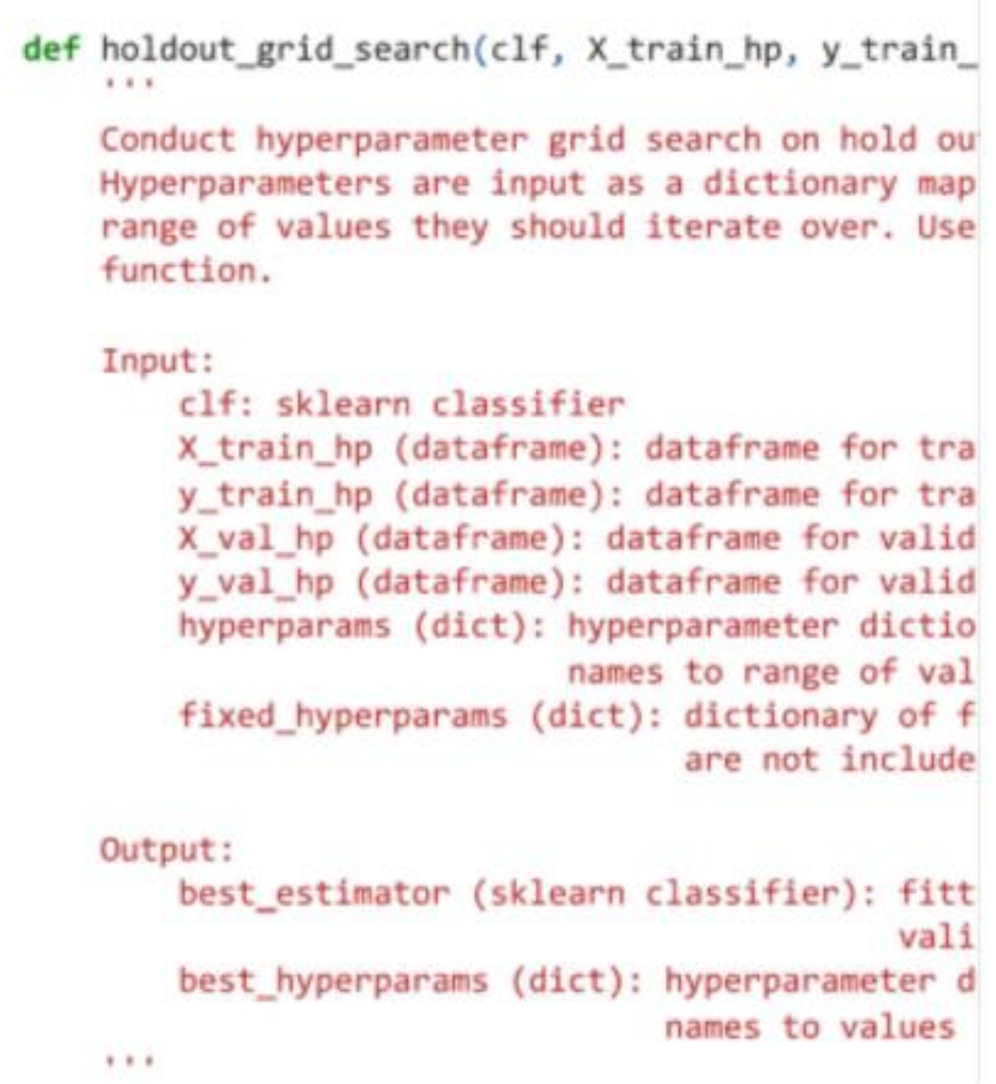
	Age	Diastolic BP	Poverty index	Race	Red blood cells	Sediment
1599	43.0	84.0	637.0	1.0	49.3	
2794	72.0	96.0	154.0	2.0	43.4	
1182	54.0	78.0	205.0	1.0	43.8	
6915	59.0	90.0	417.0	1.0	43.4	
500	34.0	80.0	385.0	1.0	77.7	

The training set size is (5147, 18) and a sample of the data is shown in the table. The target variable is whether or not the target died within 10 years, which can be seen by running the next cell.



Training Dataset. For each feature, represented as a column, values that are present are shown in black, and missing values are set in a light color.

Random Forests

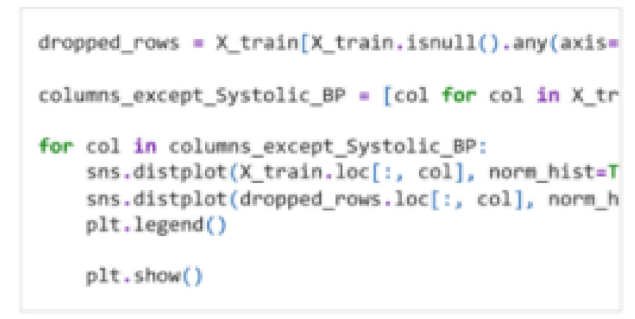


```
def holdout_grid_search(clf, X_train_hp, y_train_
    '''
    Conduct hyperparameter grid search on hold ou
    Hyperparameters are input as a dictionary map
    range of values they should iterate over. Use
    function.

    Input:
        clf: sklearn classifier
        X_train_hp (dataframe): dataframe for tra
        y_train_hp (dataframe): dataframe for tra
        X_val_hp (dataframe): dataframe for valid
        y_val_hp (dataframe): dataframe for valid
        hyperparams (dict): hyperparameter dictio
                            names to range of val
        fixed_hyperparams (dict): dictionary of f
                                  are not include

    Output:
        best_estimator (sklearn classifier): fitt
                                             vali
        best_hyperparams (dict): hyperparameter d
                                 names to values
    '''
```

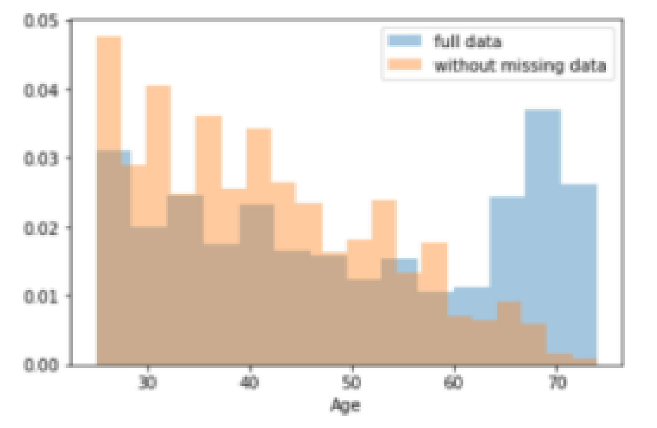
The task is to find a subgroup of at least 250 cases from the test data on which the model has a C-Index of less than 0.69 using Random Forests. A mask should be defined using a feature and a threshold to define a subset with poor performance. The bad_subset function is defined, which uses the mask to select a large subset with poor performance. The imputation approach is used to replace the missing values with substituted values based on the other available values using mean substitution. Hyperparameter grid search is used to find the best-performing random forest model, where n_estimators, max_depth, and min_samples_leaf parameters are defined and optimized. The target performance for the test C-Index is set to at least 0.74 or higher. In summary, a single decision tree is prone to overfitting, so a random forest can be used to combine predictions from many trees to create a robust classifier. We can use scikit-learn to build a random forest with default hyperparameters, which performs better than a single decision tree but still overfits. To find a better model, we need to optimize the hyperparameters for both good predictive performance and minimized overfitting.

Imputation

The random forest model has been built and optimized, but there was still a decrease in the test C-Index due to the loss of data caused by missing values for systolic blood pressure. To address this, the missing values can be imputed. Before proceeding with imputation, the data is explored to determine if the missing values are missing at random or not.To impute missing values in features, the multivariate feature imputation strategy using scikit-learn's IterativeImputer class is suggested. This method trains a regression model for each feature with missing values, using all other features to predict observed values and infer the missing ones. Since one iteration may not be enough, multiple iterations can be performed. The IterativeImputer class can be used for this purpose.



```
dropped_rows = X_train[X_train.isnull().any(axis=

columns_except_Systolic_BP = [col for col in X_tr

for col in columns_except_Systolic_BP:
    sns.distplot(X_train.loc[:, col], norm_hist=T
    sns.distplot(dropped_rows.loc[:, col], norm_h
    plt.legend()

    plt.show()
```



Plot histograms of dropped rows and the entire dataset's covariates (excluding systolic blood pressure) to check for trends.

Important- Acknowledgements

This poster showcases a case study of a machine learning project that predicts 10-year risk of death in individuals from the NHANES I epidemiology study. As a TA for AI in Medicine elective at Qatar University and Industry Lead at Clinician Engineers Hub, I present this project as a useful example for doctors and medical students to understand the intersection of medicine and engineering. It is also relevant for those interested in specializing in machine learning. The project is based on the Machine Learning Theory Lectures by Andrew Ng and Pranav Rajpurkar.

Yosra Magdi

سدرة للطب Sidra Medicine

Welcome to our tech-focused conference! Our poster presentation features an innovative AR experience that you can access by scanning the image below. See the concepts come to life and get a deeper understanding of the topic at hand.

Don't forget to turn up the volume on your device for an explanation. During the networking breaks, feel free to connect with the presenters to discuss the topic further.

Thanks for joining us!