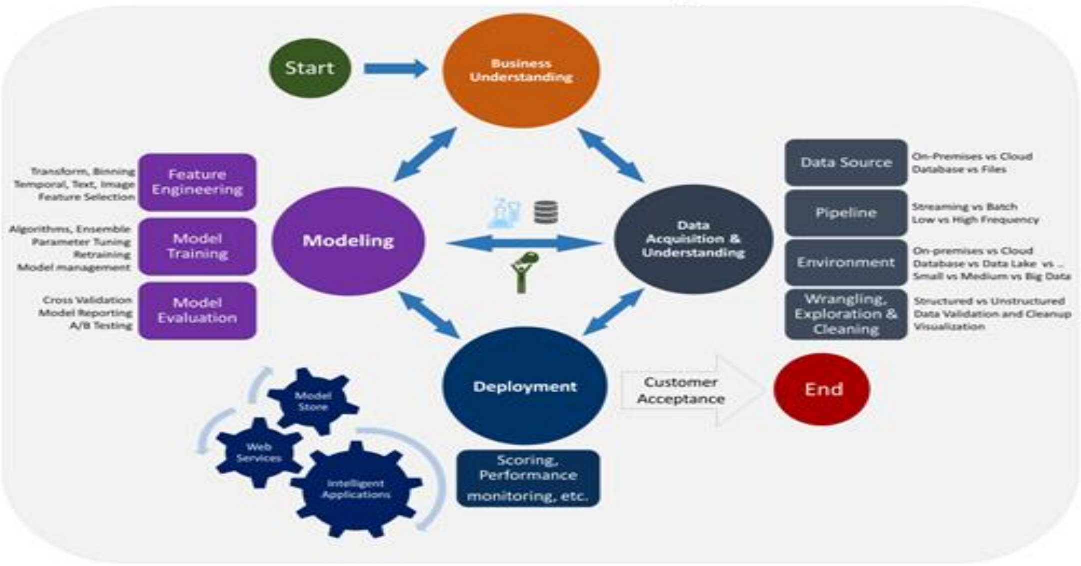
MACHINE LEARNING FRAMEWORK FOR HEART FAILURE PROGNOSTICS

Catherine Prashanthini Andra Thomas Balraj
Illinois Institute of Technology
candrathomasbalraj@hawk.iit.edu

PROBLEM DESCRIPTION

The healthcare industry generates and collects huge amounts of healthcare data from Electronic Health Records (EHR), In-Hospital Records (IHR), and vitals of the patient. The healthcare sector is “data rich” but “knowledge poor”. This is because not all the data are utilized for discovering patterns in occurrence of diseases and for effective decision making. These can also be used to analyze the quality of care provided by the hospitals.

The human and financial costs of cardiovascular disease are enormous. Heart disease is the leading cause of death for men and women all over the world. The key to cardiovascular disease management is to evaluate large scores of datasets, compare and mine for information that can be used to predict, prevent, manage, and treat chronic diseases such as heart attacks. Healthcare analytics in heart failure has the potential to reduce costs of treatment, predict future heart failure and improve the quality of life in general.



SOURCE: http://./microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

In this project, the possibility of heart failure is predicted using vitals of the patient and data from their health records. It is hard to diagnose heart problems with just observation and early detection of heart failure can prevent death rates. The features from Framingham Heart Study are used to predict Coronary Heart Disease (CHD). The attributes collected are fed into a machine learning algorithm for classification, which provides reliable results for diagnosis. It is deployed using Python Flask to obtain and display data in real time.

DESCRIPTION OF DATASET

The Framingham Heart Study is a long-term, on-going cardiovascular cohort study on residents of the city of Framingham, Massachusetts. The study began in 1948 with 5,209 adult subjects from Framingham and is now on its third generation of participants (having a total of 14,428 participants). Phenotypic traits collected in the FHS cohorts over 59 years of follow up relevant to Centre research include: physiological indices, disease risk factors and biomarkers, ages at disease onsets, behavioral and life history characteristics, selected markers of aging and stress resistance and longevity.

The dataset used for this project is from Kaggle and contains 14 attributes that are related to cardiovascular diseases. In this 10-year cardiovascular risk of the person is estimated.

- **Sex**: Male or Female
- **Age**: age of the patient
- **Current Smoker**: whether the patient is a current smoker or not
- **Cigs Per Day**: number of cigarettes that the person smokes on average in a day.
- **BPMeds**: whether the patient was on blood pressure medication or not.
- **PrevalentStroke:** whether the patient previously had a stroke or not.
- **PrevalentHyp:** whether the patient was hypertensive.
- **Diabetes:** whether the patient has diabetes or not.
- **TotChol:** total cholesterol level.
- **SysBP:** systolic blood pressure.
- **diaBP:** diastolic blood pressure.
- **BMI:** Body Mass Index

- **HeartRate:** heart rate of the patient.
- **Glucose:** glucose level
- **TenYearCHD:** 10-year risk of coronary heart disease.

Education feature is removed for predictive analytics as it does not have any impact directly on heart risk.

DATA CLEANING

The Dataset has missing values that contribute about 12% of the entire dataset.

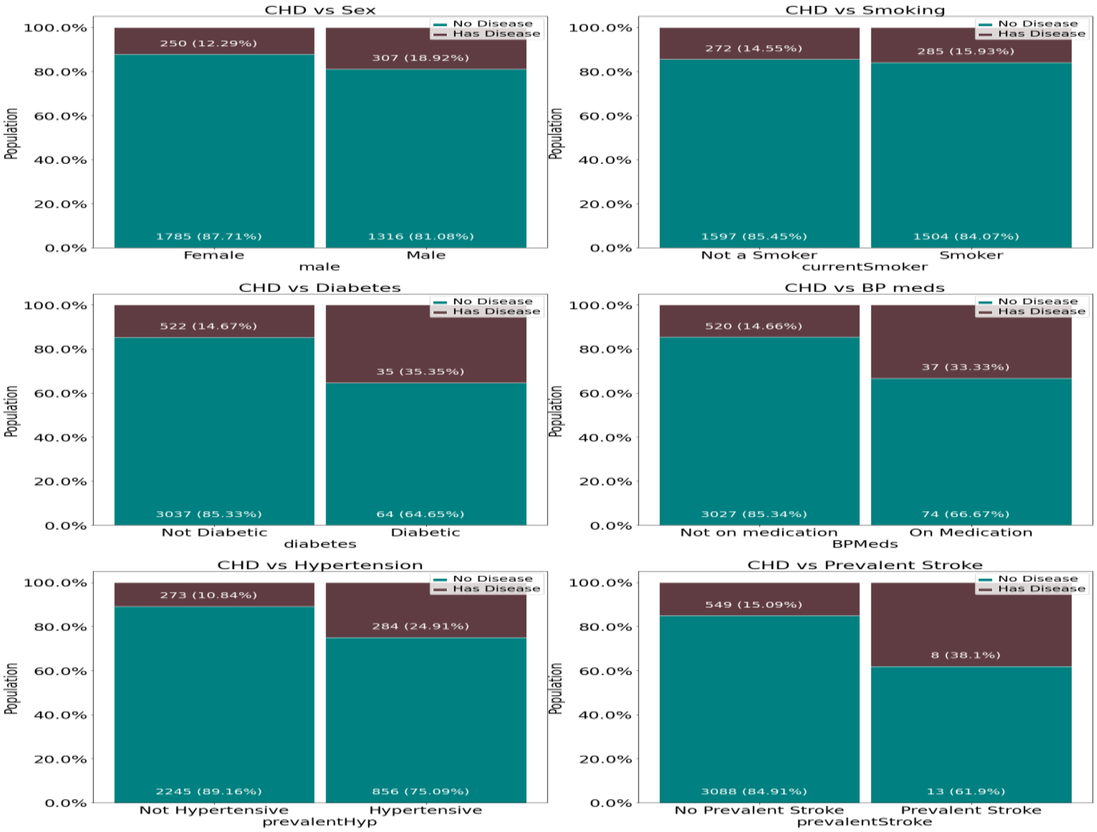
Thus, the rows with the missing values can be removed from the dataset so that the dataset is clean and ready for modelling.

DATA EXPLORATION

Exploratory data analysis is performed on the dataset to get insights from the data. From the visualizations, it is observed that:

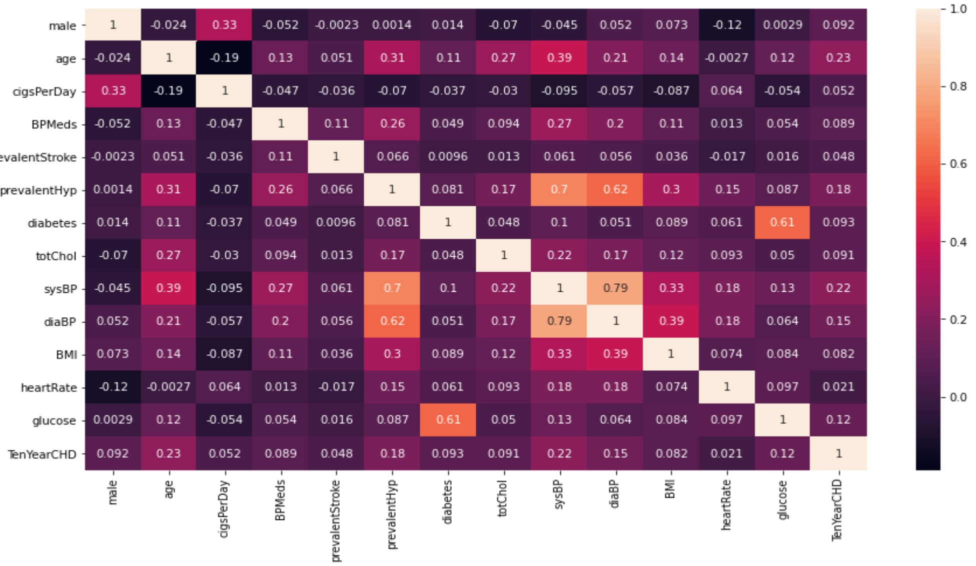
- AgeGroup 50-70 years are more likely to get CHD.
- Males are more prone to the disease than the females.
- The percentage of people who have CHD is slightly high in smokers than nonsmokers and those who smoke more than 10 cigarettes per day are more prone to high risk of CHD.
- A larger percentage of the people who have CHD are on blood pressure medication.
- The percentage of people who have CHD is higher among the diabetic, and those with prevalent hypertension as compared to those who don’t have similar morbidities.
- People who have had prevalent strokes before are more likely to get CHD.

- Since currentSmoker variable is similar to the cigsPerDay variable as cigsPerDay=0 is considered as currentSmoker=0. Thus currentSmoker field can be removed.



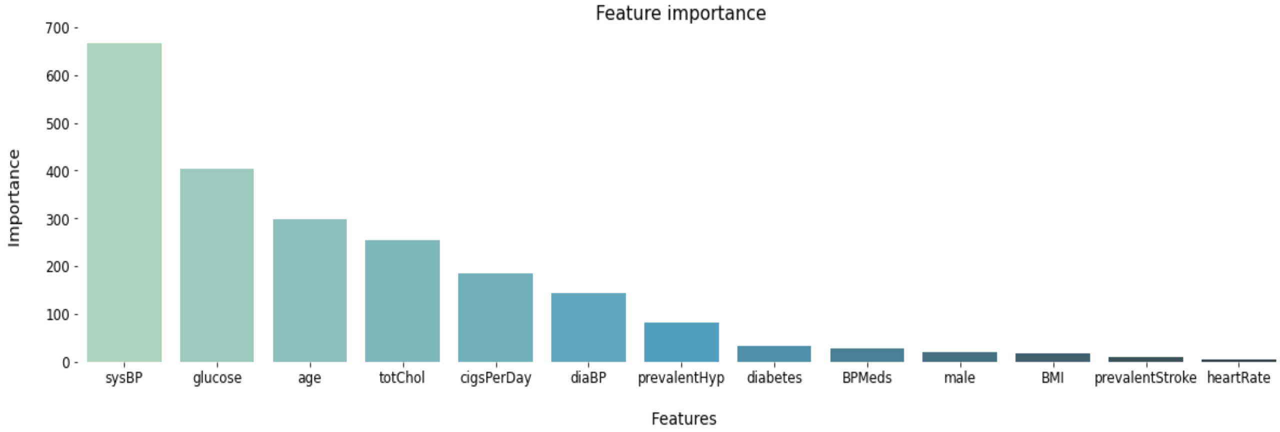
FEATURE ENGINEERING

Correlation map for the 13 features is shown below.



None of the feature has correlation greater than 0.5 with the TenyearCHD. Age and systoclic BP have the highest correlation.

Feature importance is calculated using SelectKBest class from sklearn to extract the required number of best features.

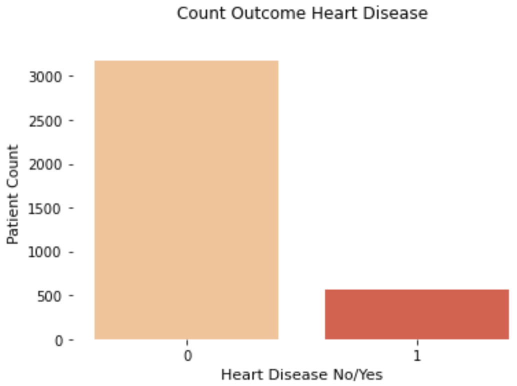


The systolic BP has the highest importance of all the features followed by glucose level and age of the patients.

The top ten features are selected for further analysis and modelling.

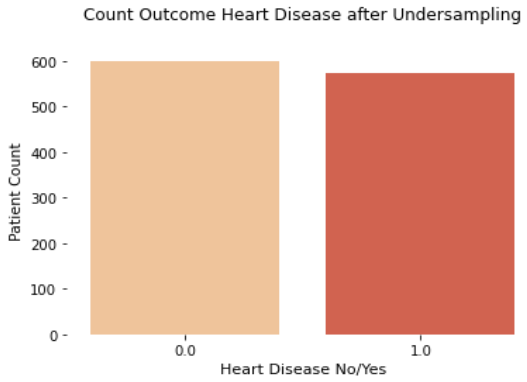
RESAMPLING IMBALANCED DATASET

From the above graph it can be seen that there is an imbalance in the dataset with the proportion of class 0 and class1 as 5.56 : 1. Thus the classifier will be biased to the negative cases and have high accuracy but poor precision and recall.



Undersampling Technique

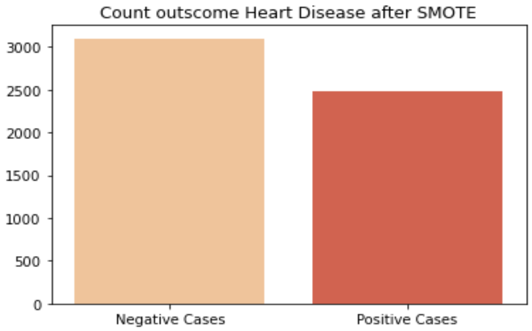
With the highly imbalanced dataset, the classifier always predicts the major class as it dominates in the dataset. Undersampling aims to decrease the number of instances from the overrepresented class in the data set thus balancing the major and minor classes in the dataset.



Synthetic Minority Oversampling Technique (SMOTE)

This procedure can be used to create as many synthetic examples for the minority class as are required to balance the class distribution. The number of majority class is first trimmed using random undersampling and then SMOTE is used to oversample the minority class.

After applying SMOTE, the new dataset is much more balanced as shown in the graph below: the new ratio between negative and positive cases is 1:1.2 up from 1:5.57.



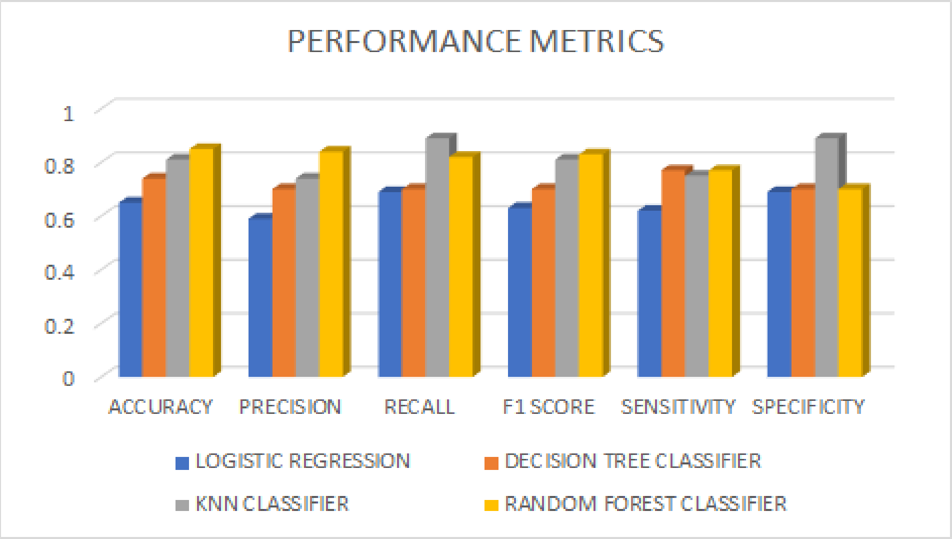
The SMOTE resampling technique is selected and used in this project.

MODELS

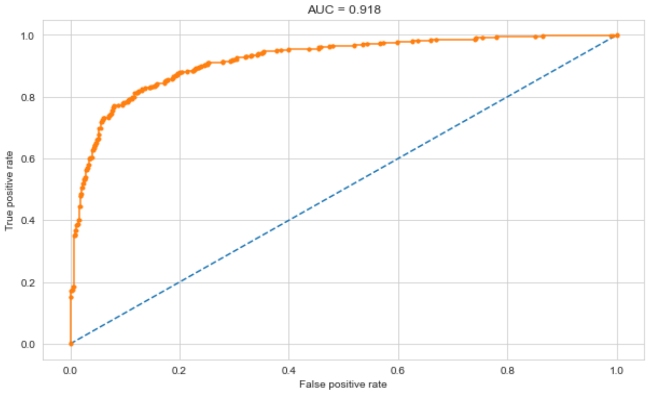
The following supervised ML algorithms were used to train the model:

1. Logistic regression
2. Decision Tree Classifier
3. KNearest Neighbor Classifier
4. Random Forest Classifier

Various performance metrics such as accuracy, precision, recall, f1 score, sensitivity, specificity and the AUC scores were calculated to find the best model.



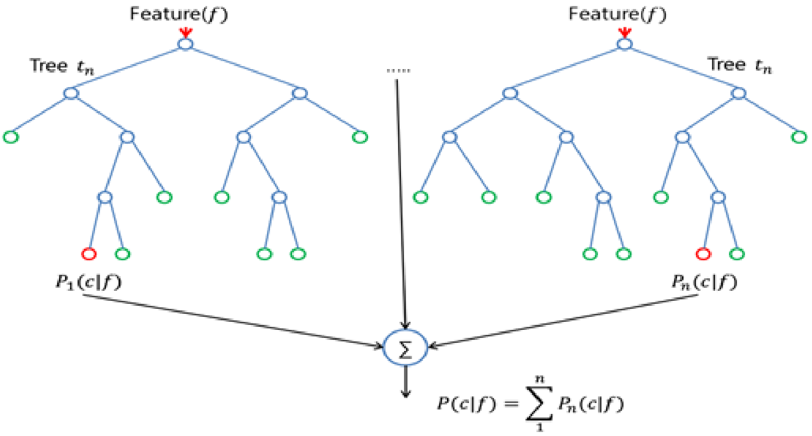
The Random Forest Classifier which is an ensemble algorithm has the best performance compared to the other algorithms as shown in above graph. The ROC curve for the Random Forest is shown below with AUC score of 0.92.



ROC curve for RF algorithm

RANDOM FOREST CLASSIFIER

Random forest is an ensemble tool which takes a subset of observations and a subset of variables to build decision trees. It builds multiple such decision trees and amalgamates them together to get a more accurate and stable prediction. If there are N numbers of cases in the training set, then a sample of N is taken at random with replacement and this sample will be the training set for growing the tree. Out of M input variables, m variables are selected at random and the best split on these m is used to split the node and this m value is kept constant during the forest growing. Each tree is grown to the largest extent possible. The forest chooses the classification having the most votes (over all the trees in the forest). Random forest is generally seen as a black box which takes in input and gives out predictions, without knowing much about what calculations are going on the back end.



SOURCE: ***https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/***

WORK YET TO BE DONE

1. Tune the parameters of the Random forest model to check its effect on the performance of the model.
2. Pickle the model. Python flask framework will be used for real time user interface.
3. Create a front-end using HTML and CSS.
4. Deploy the model.