Capstone Project

Supervised ML-Classification
Cardiovascular Risk Prediction

Keshav Sharma, Arvind Krishna, Jayesh Panchal, Sahil Ahuja
Data science students
Cohort- Boston, Alma Better

Abstract:

Cardiovascular diseases, also called CVDs, are the leading cause of death globally, causing an estimated 17.9 million deaths each year.

CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. More than four out of five CVD deaths are due to heart attacks and strokes, and one third of these deaths occur prematurely in people under 70 years of age.

The most important behavioural risk factors of heart disease and stroke are **unhealthy diet, physical inactivity, tobacco use** and **harmful use of alcohol**.

The effects of behavioural risk factors may show up in individuals as raised blood pressure, raised blood glucose, raised blood lipids, and overweight and obesity.

Introduction:

What is machine learning?

Machine learning (ML) is a subset of artificial intelligence (AI) which allows applications to become more accurate in predicting outcomes without being explicitly programmed to do so.
Machine learning algorithms use historical data as input to predict new output values.
Classification is the supervised machine learning technique which is used to predict the discrete values.
Using this technique we are going to predict the Cardiovascular Risk based on the available data.

Objective

The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD) based on their health statistics and information about their tobacco usage.

1. About the data

Data provided are in unformatted manner, corrupted data, and duplicate data and also sometimes it is irrelevant. For doing the analysis on the data the data needs to be in correct format.

The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The dataset provides the patient's information and health stats. It includes over 4,000 records and 15 attributes.

Data Description and Attributes:-

- Id: Patient identification number.

Demographic:

- Sex: male or female("M" or "F")
- Age: Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

Behavioural

- is smoking: whether or not the patient is a current smoker ("YES" or "NO")
- Cigs Per Day: the number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

Medical (history)

- BP Meds: whether or not the patient was on blood pressure medication (Nominal)
- Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)
- Prevalent Hyp: whether or not the patient was hypertensive (Nominal)
- Diabetes: whether or not the patient had diabetes (Nominal)

Medical (current)

- Tot Chol: total cholesterol level (Continuous)
- Sys BP: systolic blood pressure (Continuous)
- Dia BP: diastolic blood pressure (Continuous)
- BMI: Body Mass Index (Continuous)
- Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
- Glucose: glucose level (Continuous)

Dependent variable (desired target)

- **10-year risk of coronary heart disease CHD(binary: “1”, means “Yes”, “0” means “No”)**

2. Exploratory Data Analysis and Feature Engineering

We found out so many of null values available in the data, especially in columns like glucose, education, BPMeds, and tot cholestrol the numbers are very high.
Now the main issue with these null values is that they can't be estimated

from other data entries. The dataset we are working on is from a medical domain, that said, the entries in this data are person specific and the values vary among different individuals. Its a rare chance the two individuals share same health stats, hence the most logical option that we have to deal with such values is removing the rows with any null value.

We could have tried imputing them using some advance techniques like *KNN Imputer*, but they couldn't be that accurate because it'll use other entries to estimate the nulls, hence the values would depend on the values present among other rows, which as discussed earlier isn't a ideal approach for such dataset. Exploratory data analysis or commonly known as EDA helps to explore data, and possibly formulate hypotheses that might cause new data collection and experiments. EDA build a robust understanding of the data, issues associated with either the info or process. It's a scientific approach to get the story of the data. It focuses more narrowly on checking assumptions required for model fitting and hypothesis testing. It also helps while handling missing values and making transformations of variables as needed. Also we'd deal with outliers in this section. Furthermore, we'll try to extract or convert some of the attributes using some feature engineering.

All machine learning algorithms use some input data to create outputs. This input data comprise features, which are usually in the form of structured columns. Algorithms require features with some specific characteristics to work properly. Here, the need for feature engineering arises. Feature engineering mainly have two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.
- As we can see all the null values have been removed, this surely cost us some valuable data loss but given the options, there was no better choice but to remove those rows having nulls.
- Also the patient ID doesn't contribute their health stats, and it wouldn’t be of much help to the model either. Hence, we can drop the "ID" column too.

3. Independent and target variable

- **Independent variable are as follow**
  age', 'education', 'sex', 'is smoking', 'cigsPerDay', 'BPMeds', 'prevalent Stroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartrate', 'glucose'
- **TARGET VARIABLE:**
  TenYearCHD signifies if the person has a risk of heart disease or not. It’s a binary attribute (binary: “1”, means “Yes”, “0” means “No”) resembling the diagnosis results for patients. We can use this attribute to see how many patients have a risk of CHD.

4. Data distributions

Following are the categorical features in this dataset:

['education', 'sex', 'is_smoking', 'BPMeds', 'prevalentStroke', 'prevalentHyp', 'diabetes', 'TenYearCHD']

Following are the numeric features in this dataset:

['age', 'cigsPerDay', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose']

Categorical Features are mostly binary type in our case, hence the values majorly are either 0 and 1 (some exceptions excluded). As for the numeric features, all have a different set of range and continuity of values.
We can observe that most of the distributions are right skewed for numeric features. **totChol** (total cholesterol) and **BMI** have roughly similar distributions, which depicts a linear dependency. **Glucose** have a highly right skewed distribution, this might lead to some biasness. It shows Glucose has a lotof outliers. Though it is usually a good practice to deal with such outliers, however in this case we can't do much about it. The data is taken through medical survey and the values are patient's health stats, hence those values are absolute and we cant manipulate them by any means. We could have used techniques like Square root transformation, Log tranformation,etc., to convert the column for getting a better distribution, but since we can't manipulate a medical statistics of a person, it is suggestive to go with the actual values.

We can see a lot of outliers in columns like, **Totchol**, **SysBP**, **DiaBP**, **BMI**, **Glucose**, etc. As stated before we can't manipulate data in such way that we change the original patient stats, neither we can entirely drop those entries with outliers. This will lead to huge amount of data loss, We would lose meaningful data in order to achieve this. The best solution to this could only be, to drop the rows with such outliers with minimal data loss.

Dropping rows with borderline outliers, We'll try to be considerate and only drop values that do not make any sense or unlikely to occur.

5. New Derived Feature

DiaBp and SysBP

From the above Heatmap, We can see both of these columns are heavily correleted, there's some relationship we can establish with these two features further.

Also Elevation of systolic blood pressure predicts the risk of cardiovascular disease better than increases in diastolic blood pressure. Although associated with more variability in measurement, systolic blood pressure is easier to determine and allows more appropriate risk stratification than diastolic blood pressure.

We can combine these two features using the following formula:

MAP = (Systolic Blood Pressure + 2 x Diastolic Blood Pressure) / 3

Here, MAP signifies Mean Arterial Pressure

6. Model Development

Now its time to implement the Machine Learning models and check the accuracy of each model to point out the best one out of all. In this project we are implementing 8 machine learning algorithms to predict the target variable and also we'll apply optimization techniques to get the best resulting accuracy.

Prerequisites

Now that the Dataset is cleaned and we have added all the neccessary features along with some conversions of categorical features. Its time to split the data into training and testing sets.

Note:- These training and testing data are going to be same for all the model we'll build such that all of the models are evaluated on a same set of parameters.

We can clearly see, the class are imbalanced and it'd result the model to be more biased towards '0' class (people with no Risk of CHD). We need to find a way to train a model in such a way that it can take some risks and give more of '1' class results.

The reason behind that is we have a make a model that can predict a risk of CHD. If, based on patient's health stats, there's even small chance that a person could have a risk of heart disease, the model should be able to predict the risk.

Surely this would lead our model to attain a lower accuracy value since its taking more risks for 'at risk' patients, but in a real world scenario this model is more useful because its highly likely for such model to be able to predict if a patient is at risk.

If we would have used a normal approach here, it might give us a better accuracy thanks to the class imbalancy, but in real world such model isn't a ideal solution, its expected that it'd mostly fail to predict if the patient isat risk, which takes away the sole purpose of the project.

Hence, to deal with this imbalancies we are using SMOTETomek on the training set. This is a type of data augmentation for the minority class and is referred to as the **Synthetic Minority Oversampling Technique**, or **SMOTE** for short. The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class.

Models

Following models have been used for predictions:-

- Logistic Regression Classifier
- K-Nearest Neighbors(KNN Classifier)
- Naive Bayes Classifier
- Support Vector Machine(SVM Classifier)
- XGB Classifier
- Decision Tree Classifier
- Random Forest Classifier
- Neural Networks Classification

7. Conclusion

- We've noticed that XBG Classifier is the stand out performer among all models with an f1-score of 0.828.
- In case of Logistic regression, We were able to see the maximum f1-score of 0.656, also in case of K-Nearest Neighbors, the f1-score extends upto 0.742.
- Naive Bayes Classifier showed a balanced result amongst the model we have implemented, it has a f1-score of 0.781, which is neutral with regards to our observations across various models. But in case of SVM(Support Vector Machines) Classifier, the f1-score lies around 0.64, which also happens to be the lowest score among all models we've implemented.
- Out of the tree-based algorithms, the Random Forest Classifier was providing an optimal solution towards achieving our Objective. We were able to achieve an f1-score of 0.826 for the test split. We also noticed that in the case of Decision-tree Classifier, we were able to achieve an f1-score of 0.768 for the test split.
- We have also implemented a experimental neural network, however the results were non-conclusive. However the NN approach seems promising.

8. References-

- **Python Pandas Documentation**

  https://pandas.pydata.org/pandas-docs/stable
- **Python MatPlotLib Documentation**
  https://matplotlib.org/stable/index.html
- **Python Seaborn Documentation**
  https://seaborn.pydata.org
- **Python SkLearn Documentation**
  https://scikit-learn.org/stable
- **Keras API Official Documentation:**
  https://keras.io/api/