Cardiovascular Risk Prediction

Vidit Ghelani,
Mahesh Lakum
Data Science Trainees,
AlmaBetter, Bangalore

Abstract:

In this modern world, there are a lot many cases of Cardiovascular disease. There are a lot many factors causing this. As a research project we have gathered a dataset of the various parameters of patients and health candidates. The aim of this exercise is to study the impact of each parameter and build a model to correctly predict if a patient is at the risk of this disease in the near future. We will implement Machine Learning Algorithms for this task. The outcome is a dichotomous variable, hence this is a classification problem.

Keywords: EDA (Exploratory Data Analysis), Classification, Model Training, Machine Learning Algorithms.

1. Problem Statement

A dataset of patients' information of over 4,000 records and 15 attributes is available from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD).

2. Introduction

The medical companies check the present status of certain vitals and the medical history of the client before they provide a treatment. This helps them help the patient in a better way. However, they can still not predict the risk of cardiovascular disease from the gathered data. While looking at this from the insurance companies' point of it, it could be very useful if they could predict this risk beforehand. For this they can gather certain vital information of the client and based on it they can predict this risk factor using Machine Learning Algorithms. This can also help the doctor treat and alarm the patient well in advance. And a lot of patients can be saved from the actual event occurring. Our aim is to achieve this.

3. Understanding the features

The raw data contains various features mentioned in columns. These features can be divided into demographics, behavioral, and medical history. They are as below:

Demographic:

1. Sex: male or female("M" or "F")
2. Age: Age of the patient
3. Education: Level of education. ("1", "2", "3" and "4".)

Behavioral:

4. is_smoking: whether the patient is a smoker ("YES" or "NO")
5. Cigs Per Day: number of cigarettes smoked on average in one day.

Medical(history):

6. BP Meds: whether the patient was on blood pressure medication
7. Prevalent Stroke: whether the patient had previously had a stroke
8. Prevalent Hyp: whether the patient was hypertensive
9. Diabetes: whether the patient had diabetes

Medical(current):

10. Tot Chol: total cholesterol level (Continuous)
11. Sys BP: systolic blood pressure (Continuous)

12. Dia BP: diastolic blood pressure (Continuous)
13. BMI: Body Mass Index (Continuous)
14. Heart Rate: heart rate (Continuous)
15. Glucose: glucose level (Continuous)
16. TenYearCHD: 10-year risk of coronary heart disease CHD

4. Approaching the Problem

The entire process to analyze the data, find out some useful conclusion was done in four steps:

4.1 Pre- Processing
4.2 Performing exploratory data analysis (EDA)
4.3 Training the Models
4.4 Observation and conclusion

4.1 Pre- Processing

4.1.0. View the data

4.1.0.1. Understand the Features

4.1.1. Clean the data

We will create a new dataframe and deal with the null values by replacing it with appropriate values or may drop them. This way we do not make any changes to the raw data.

1. Remove/ replace data for missing values
2. Remove duplicate values (here none)
3. Change column names (if needed)
4. Add derived features (sysBP, diaBP Pulse Pressure)
5. Drop the redundant features

Summary about the data:

- The shape of data is (3390,16). Which means it has 16 features and 3390 rows.
- Data types in the data are: object, float and integer. Here, the data types are already relevant. So, we need not change it.
- There are no duplicate values in the data.
- The missing values amount to 510 rows.
- In the dependent variable, the strength of the minority class or the Risk class is 511 out of 3390.
- This dependent variable is categorical in nature.
- Certain data corresponds Dealing with irrelevant values. 22
- Looking at the percentage of classes, we can say that there is a huge class imbalance. We will need to deal with this before training the machine learning models.
- Label Encoding: Converting the data of 'sex' & 'is_smoking' feature from categorical to numerical where, M = 1, F = 0 and YES = 1, NO = 0, respectively.

4.2 Performing exploratory data analysis (EDA)

4.2.1 Univariate Analysis:

4.2.1.1 Observations from the plot of Categorical Features:

1. Majority of the people in the data belong to the 1st class of education which amounts to about 1500. As the class count increases, the number of people reduces.
2. Number of females in the data is more than men by nearly 500 counts.
3. The data contains nearly same amount of non-smokers and smokers.
4. Very few people are on BP medication. A bove 3000 people are free from those medicines.
5. Majority of the people have not suffered a stroke previously.
6. About 1/3rd people have hypertension. This data amounts to more than 1000 counts.
7. Very few people have diabetes. Over 3000 people are free from diabetes.

4.2.1.2 Observations from the plot of Numerical Features:

1. In the data, age ranges from 35 years to 70 years, of which majority people belong to the group of 40 - 45. Overall the graph is mostly normally distributed.
2. The data contains majority of the values

of non smoker as zero cigarettes has the highest count. Other notable peaks are of 20 and 10 cigarettes per day.
3. The total Cholesterol follows a near nor mal distribution which ranges from 100 to 400 units, with peak nearing 250 units.
4. The Heart rate ranges from 35 to 110 with a peak at 75.
5. Systolic BP appears to be slightly left skewed with range 100 to 200 units.
6. Diastolic BP appears to be near normal distribution with range 60 to 120 units.
7. The BMI has a range of 16 to 40 with peak at 25. It follows normal distribution.
8. Glucose is left skewed graph with peak at about 75. It spans from 50 to 125.

4.2.2 Bivariate Analysis:

As cholesterol is low the systolic BP also remains low. Systolic BP and diastolic BP have a positive relationship. Most of the cases, less cholesterol means less glucose. Diastolic BP, systolic BP and BMI have a slight positive relationship.

4.2.3 Checking correlation

1. Variables with high correlation (i.e. above 50%):
   - "is_smoking" - "cigsPerDay": This is the obvious correlation as the first one is a categorical variable while the second is a numerical one.
   - "prevalentHyp" - "sysBP": Hypertension is related to Systolic Blood Pressure.
   - "prevalentHyp" - "diaBP": Hypertension is related to Diastolic Blood Pressure.
   - "diabetes" - "glucose": This is the obvious correlation as the first one is a categorical variable while the second is a numerical one.
   - "sysBP" - "diaBP": These two are highly correlated as both of them are blood pressures.
2. Systolic and Diastolic Blood Pressure does influence hypertension and BMI.
3. Systolic BP and age have a positive correlation.
4. Variables such as age, prevalent hypertension, systolic BP, diastolic BP, and glucose have lower but positive correlation with the dependent variable. Hence, we can say that it has an influence on the risk of heart disease.
5. Education is the only variable negatively correlated with the dependent variable.

4.2.4 Feature Engineering:

Here, we will add derived variables and drop the highly correlated variables.

1. "sysBP" and "diaBP" are highly correlated. According to the article these two are linearly related variable hence we can convert them to a single variable "pulsePressure" and drop these two individual variables.
2. We will drop the 'is_smoking' variable, as it is categorical, it gives less information compared to the numerical variable "cigsPerDay".
3. "sex" is a categorical variable. So, we will create its dummy variable for the ease of analysis and We will drop this variable. Here, the dummy variables will be "sex_0" corresponding to 'Female' and "sex_1" corresponding to 'Male'.

4.2.4 Dealing with Class Imbalance:

There is a huge class imbalance in the dependent variable (target variable). Hence, we used 'SMOTE' (Synthetic Minority Oversampling Technique) to balance that for better model training. This is an oversampling technique which will balance the classes of dependent variables. The original data consisted of 3390 rows. After oversampling the number of rows becomes 5758, hence total 2368 new rows were added.

4.2.4 Normalizing the data:

Looking at the count plot earlier for each independent feature, we observed that the

majority of the features were skewed and not in the same range. Hence we need to Normalize these features. We use MinMaxScaler to achieve the normalized data.

4.3 Training the Models

Here, first we split the data for training and testing. We split the data into 80 : 20 ratio. Next we train the model using the train data set and then test its performance on the test data. We undertake this process for various models. Based on that we will select the best suited model for our prediction.
The models used here are:

- Linear regression
- Logistic Regression
- Random Forest
- XGBoost Classifier
- KNN Classifier
- Support Vector Machine (SVM)

4.4 Identify the best model and conclude

Here, we find the accuracy, precision, recall, F1 Score and AUC ROC Score for each model. This is done for all the six models and based on this we identify the best suited model. From this we can say that:

- RandomForest model out performs all the models when it comes to training. Also, the value of all the metrics are fairly high while testing the model. However, if we compare the train vs. test metrics, we see that somewhere, there is a slight occurrence of over fitting. We may consider hyperparameter tuning on this to improve the test results.
- The second best model appears to be the XGBoost Classifier. Presently it appears to have optimal fit as the test and train metrics have the same value. Hence, it is highly preferable to perform better after hyper parameter tuning.
- Logistic regression and SVC/ SVM are performing well, however, accuracy and recall for these are quite lower. Hence, we may not work upon improving these models.
- KNN does appear to have fine train metrics however, it does not quite really perform well for the test dataset. Hence, it also might have some degree of over fitting. So we will not consider it for further improvement.

	Model	Train Accuracy	Test Accuracy
0	Logistic Regression	0.739036	0.719618
1	Random Forest	1.000000	0.908854
2	XGB Classifier	0.874946	0.858507
3	KNN	0.877117	0.812500
4	SVC	0.762918	0.746528

4.4.1 Confusion Matrix & ROC Curve:

From the Confusion Matrix we can say that The performance of a model becomes critical to evaluate in case of classification problems when the minority classes are present in the data. Hence, we need to focus on the amount of False Negatives generated by the model. Looking at the count of False Negatives, the best models in the decreasing order of performances are Random Forest, KNN, XGB.

From the ROC AUC curve we see that Random Forest has a better performance compared to all the other models. The next one to follow Random Forest is XGB Classifier.

4.4.2 Hyperparameter Tuning:

From the model building section we can understand the best models are Random Forest and XGB Classifier. Between these two, Random Forest is overfitting to some extent compared to other models. XGB Classifier is the second best performer, hence this will be chosen for hyperparameter tuning.
Here, we have used GridSearchCV, with hyperparameters as:

- Number of estimators: 300 & 350
- Max depth: 7, 8 & 9
- Learning Rate: 0.01 & 0.001

After applying this we find the best parameters as: 350, 9 and 0.01
As we can see the hyperparameter tuning has improved the model from the base XGBoost Classifier especially for recall, the parameter we are focusing on.
However, this still doesn't give the model explainability. For this we need to check the feature importance.

4.4.3 Feature Importance:

1. Gender, education and age are the main features which help us classify the risk of CHD.
2. Looking at the features related to medical condition or medical history, we could say that they have about 10% influence on the Risk of CHD; which is quite less or insignificant.
3. Cigarettes per day and Blood Pressure (difference of SysBP and DisBP) influences the CHD variable by about 30%.

5. Challenges Faced:

Few of the challenges we faced were:

1. Need to plot a lot of graphs to analyze the data.
2. Feature selection and feature engineering.
3. Hyperparameter selection for optimizing the model.

6. Conclusion:

From this exercise we can conclude that:

1. Keeping the accuracy and performance of Confusion Matrix, performance metrics value (Accuracy, Precision, Recall) and ROC Curve we could say that XGBoost model performed the best as it did not overfit the data as much as done by the Random Forest model.
2. We did hyperparameter tuning on XGB to improve its performance. The overall model performance was not improved, however, the Recall value did improve. Which is an indication that we succeeded in reducing the False Negative counts.
3. Later on we use SHAP method to know the feature importance. We could deduce that gender, education and age appear to be the major influencers. Compared to other medical features, cigarettes per day and Blood Pressure played a significant role with about 30% influence on the dependent variable. Here, it is a bit confusing to relate education with health. However, if enough data is available we could deduce if the students of certain educational backgrounds do fall victim of this. To get a better understanding of the dependence of this feature on the Risk factor, we must consult a domain expert and gather more data. This could be another research in itself.
4. We could further improve the model performance by further fine tuning the hyperparameters where we could seek help from an expert with excellent domain knowledge.

References:

1. GeekforGeeks
2. Stackoverflow
3. Python guide
4. Matplotlib guide
5. Seaborn guide
6. Scikit learn guide
7. Github
8. Kaggle