## **Capstone Project:-**

## **Cardiovascular Risk Prediction**

# **Lalit Ahirrao**

# **AniketGajmal**

# **Rushikesh Pawar**

# **Prasad Ghegade**

# **Samarth Gangurde**

## **Cardiovascular risk :-**

Heart and blood vessel disease (also called heart disease) includes numerous problems, many of which are related to a process called atherosclerosis Atherosclerosis is a condition that develops when a substance called plaque builds up in the walls of the arteries. This buildup narrows the arteries, making it harder for blood to flow through. If a blood clot forms, it can block the blood flow. This can cause a heart attack or stroke.

## **Facts Regarding Cardiovascular :-**

Cardiovascular diseases are the leading cause of death globally.

An estimated 17.9 million people died from CVDs in 2019, representing 32% of all global deaths. Of these deaths, 85% were due to heart attack and stroke.Over three quarters of CVD deaths take place in low- and middle-income countries.Out of the 17 million premature deaths (under the age of 70) due to non communicable diseases in 2019, 38% were caused by CVDs. It is important to detect cardiovascular disease as early as possible so that management counseling and medicines can begin.

#### 

#### **Reason behind the project :-**

Cardiovascular disease is a significant and ever-growing problem in all over nations. It is of pressing interest as developing countries experience a change in lifestyle which introduces novel risk factors for cardiovascular disease. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications. This research intends to pinpoint the most relevant/risk factors of heart disease as well as predict the overall risk

### **Dataset Resources :-**

The dataset is 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). The dataset provides the patients’ information. It includes over 4,000 records and 15 attributes.

**Variables :-**

Each feature is a potential risk factor. There are both Demographic, behavioral and medical risk factors.

* Demographic

1. sex:- male or female
2. age:- age of the patient

* Behavioral

1. Is\_smoking( do person smoke currently)
2. cigsPerDay( number of cigarettes person prefers, even half a cigarette)

* Medical( history)

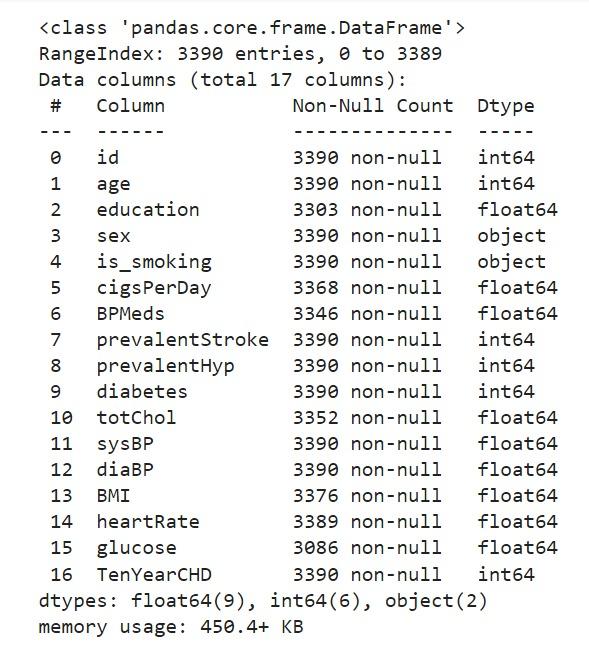
1. BPMeds: whether or not the patient was on blood pressure medication
2. prevalentStroke: whether or not the patient had previously had a stroke
3. prevalentHyp: whether or not the patient was hypertensive
4. diabetes: whether or not the patient had diabetes

* Medical(current)

1. totChol: total cholesterol level
2. sysBP: systolic blood pressure
3. diaBP: diastolic blood pressure
4. BMI: Body Mass Index
5. heartRate: heart rate
6. glucose: glucose level

* Predict variable

1. 10 year risk of coronary heart disease TenYearCHD ( 1 means 'Yes', 0 means 'NO')



### **Preprocessing :-**

### **Handling Missing values :-**

Why do we need to handle missing values?

The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.that's why we check missing values first

### **Handling Duplicate values :-**

Why is it important to remove duplicate records from my data?

"Duplication" just means that you have repeated data in your dataset. This could be due to things like data entry errors or data collection methods. by removing duplication in our data set, Time and money are saved by not sending identical communications multiple times to the same person.

**HANDLING OUTLIERS :-**

Outlier treatment is another important

step in data preprocessing (one may prefer doing missing value treatment once outliers are treated, especially if using mean imputation as outliers can skew the data). Outlier as such is an observation that lies on an abnormal distance from other values or any observation far away from the mass of data or the overall pattern. Outliers can be mild and extreme with the extreme being away from the source by a great deal. Also, an outlier can be looked for in each variable (Univariate Outlier) or can be looked for in relation to other variables (Bivariate Outlier)

**EXPLORATORY DATA ANALYSIS :-**

EDA means trying to understand the given data much better, so that we can make some sense out of it. Using univariate frequency analysis was conducted to describe key characteristics of each feature including, minimum and maximum value, average, standard deviation and others. It was also used to produce a value distribution and identify missing values, and outliers.

EDA is a process of examining the available dataset to discover patterns, spot anomalies, test hypotheses, and check assumptions using statistical measures. In this chapter, we are going to discuss the steps involved in performing top notch exploratory data analysis

In statistics, A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing tasked in Python uses data visualization to draw meaningful patterns and insights

**UNIVARIATE ANALYSIS:**

If we analyze data over a single variable/column from a dataset, it is known as Univariate Analysis. Univariate analysis looks at one feature at a time. When we analyze a feature independently, we are usually mostly interested in the distribution of its values and ignore other features in the dataset

Univariate analysis is the simplest form of analyzing data. It means that our data has only one type of variable and that we perform analysis over it. The main purpose of univariate analysis is to take data, summarize that data, and find patterns among the values. It doesn't deal with causes or relationships between the values. Several techniques that describe the patterns found in univariate data include central tendency (that is the mean, mode, and median) and dispersion (that is, the range, variance, maximum and minimum quartiles (including the interquartile range), and standard deviation).

**BIVARIATE ANALYSIS :-**

If we analyze data by taking two variables/columns into consideration from a dataset, it is known as Bivariate Analysis.

**Numeric-Numeric Analysis :-**

Analyzing the two numeric variables from a dataset is known as numeric-numeric analysis. We can analyze it in three different ways.

* Scatter Plot
* Pair Plot
* Correlation Matrix

·**Numeric - Categorical Analysis:**

Analyzing the one numeric variable and one categorical variable from a dataset is known as numeric-categorical analysis. We analyze those mainly using mean, median, and box plots.

**MULTIVARIATE ANALYSIS:**

Multivariate analysis is the analysis of three or more variables. This allows us to look at correlations (that is, how one variable changes with respect to another) and attempt to make predictions for future behavior more accurately than with bivariate analysis.

One common way of plotting multivariate data is to make a matrix scatter plot, known as a pair plot. A matrix plot or pair plot shows each pair of variables plotted against each other. The pair plot allows us to see both the distribution of single variables and the relationships between two variables

### 

### **OneHotEncoding :**

A one hot encoding is a representation of categorical variables as binary vectors.This first requires that the categorical values be mapped to integer values.Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.A one hot encoding allows the representation of categorical data to be more expressive.Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.We could use an integer encoding directly, rescaled where needed. This may work for problems where there is a natural ordinal relationship between the categories, and in turn the integer values, such as labels for temperature ‘cold’, warm’, and ‘hot’.There may be problems when there is no ordinal relationship and allowing the representation to lean on any such relationship might be damaging to learning to solve the problem. An example might be the labels ‘dog’ and ‘cat. In these cases, we would like to give the network more expressive power to learn a probability-like number for each possible label value. This can help in both making the problem easier for the network to model. When a one hot encoding is used for the output variable, it may offer a more nuanced set of predictions than a single label.

## **SMOTE**

In our data set we have Imbalanced Data Distribution in our dependent variable, it generally happens when observations in one of the classes are much higher i.e not defaulter or lower than the other classes i.e defaulter.As Machine Learning algorithms tend to increase accuracy by reducing the error, they do not consider the class distribution.

Standard ML techniques such as Decision Tree and Logistic Regression have a bias towards the majority class, and they tend to ignore the minority class. They tend only to predict the majority class, hence, having major misclassification of the minority class in comparison with the majority class. In more technical words, if we have imbalanced data distribution in our dataset then our model becomes more prone to the case when the minority class has a negligible or very lesser recall. SMOTE (Synthetic Minority Oversampling Technique) Oversampling is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them.

**ALGORITHMS WE USED:**

## KN Neighbors Classifier

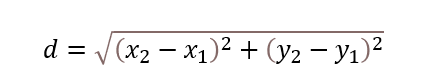
## Random Forests Classifier

## Logistic Regression

## **KNNeighborsClassifier**

The K-Nearest Neighbors algorithm is a supervised machine learning algorithm for labeling an unknown data point given existing labeled data.The nearness of points is typically determined by using distance algorithms such as the Euclidean distance ,Manhattan Distance formula based on parameters of the data.

Euclidean Distance is the most commonly used distance formula. To find the Euclidean distance between two points, we first calculate the squared distance between each dimension. If we add up all of these squared differences and take the square root, we’ve computed the Euclidean distance.



Manhattan Distance is extremely similar to Euclidean distance. Rather than summing the squared difference between each dimension, we instead sum the absolute value of the difference between each dimension. It’s called Manhattan distance​

image-1.png

K-nearest neighbors select the K number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of ‘K’ training data and which class holds the highest probability will be selected.

## **Random Forests Classifier**

The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone. When using Random Forest for classification, each tree gives a classification or a “vote.” The forest chooses the classification with the majority of the “votes.” When using Random Forest for regression, the forest picks the average of the outputs of all trees. In this classification algorithm, we will use cardiovascular patients datasets to train and test the model. We will build a model to classify whether a person shows cardiovascular risk or not.

**References :-**

* [SMOTE | Towards Data Science](https://towardsdatascience.com/smote-fdce2f605729)
* [sklearn.neighbors.KNeighborsClassifier — scikit-learn 1.0.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html)
* [sklearn.ensemble.RandomForestClassifier — scikit-learn 1.0.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)