**Summary on**

**Capstone Project - 3**

**Sachin Pandey**

**Data Science Trainee**

**Alma Better, Bangalore**

**Abstract:**

In today’s world health problems has become very common because of the not having any proper routine in our daily life. In all these types of diseases CHD (Cardio Vascular Disease) is very common, according to internet sources : Coronary heart disease is now the leading cause of death worldwide. **An estimated 3.8 million men and 3.4 million women die each year from CHD**. In developed countries heart disease is the leading cause of death in men and women. In Europe CHD accounts for an estimated 1.95 million deaths each year. So, it has become very important to detect these type of disease in order to rectify it with the help of proper medication.

In order to rectify these kind of problems we have tried to develop a model by which we can detect whether a person is going to get infected by CHD in the next 10 years. In this we have used algorithms like Logistic regression, KNN (K Nearest Neighbour) , Decision Tree Classifier, XGBoost and Random Forest Classifier.

**1.Problem Statement**

### We are provided with Cardiovascular risk prediction dataset and we have to predict that a person will get affected with CHD in next 10 years or not. In this dataset there are various columns like glucose\_level, heartbeat\_rate, BMI, age etc. on which we have to perform various operations.

**2. Introduction**

In today’s world we can see that heart disease is becoming very common. More than 18 million people get affected with Cardiovascular heart disease out of which around 3.6 lakh people died. The objective of this project is to build a model that predicts whether a person will get CHD (Cardio Vascular Heart disease) in next 10 years.

**3. Steps Involved**

Discussion of Cardiovascular\_risk\_prediction dataset will involve various steps such as:

* Loading the data into DataFrame.
* Data cleaning and performing EDA.
* Resampling the data.
* Dividing the data in train and test split.
* Hyperparameter tuning.
* Model Training.
* Model Testing.
* Model adjustment on various Evaluation metrices.
* Questions that can be asked from the dataset.
* Conclusion.

**Data Preparation**

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

**Gathering data:**

This step is about getting to know the data and understanding what has to be done before the data becomes useful in a particular context. This can be done by reading the CSV file and doing initial statistical analysis.

Though the dataset may seem to have the correct datatypes for each column, we need to check it. Inconsistent datatypes will create issues while dealing with problems.

**Cleanse and validate data:**

This step is crucial for removing faulty data and filling in gaps. Important tasks here includes:

* Removing extraneous data.
* Filling in missing values.
* Conforming data to a standardized pattern.

Dataset may contain duplicate values for particular application, so we have to delete the duplicate data in the dataset to make dataset more efficient to work on.

**Handling Null values:**

If our dataset contains null values which might tend to disturb our accuracy hence, we dropped them at the beginning of our project in order to get a better result.

**Detecting and removing the outlier:**

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal.

We can find outlier of the dataset using Boxplot and Scatterplot. Using Boxplot we have found the outlier of the dataset and then removed it from the dataframe.

# Exploratory Analysis and Visualization:

Exploratory data visualizations (EDVs) are the type of visualizations we assemble when we do not have a clue about what information lies within our dataset.

Like in this dataset we can find:

* We draw a relationship between BMI and Age of a person.
* We can draw a relationship between CHD and Sex of a person.

And many more things can be visualised using the graphs and charts.

**Dividing the data in train and test split:**

The dataset then will be divided into two parts i.e. Train and Test data. Training data will be used to train the model on the other hand Test data will be used in evaluation of a model.

**Algorithms used:**

1. **Logistic Regression:**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



1. **KNN (K Nearest Neighbour ):**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



1. **Decision Tree Classifier:**

Decision Tree is a supervised learning method used in data mining for classification and regression methods. It is a tree that helps us in decision-making purposes. The decision tree creates classification or regression models as a tree structure. It separates a data set into smaller subsets, and at the same time, the decision tree is steadily developed. The final tree is a tree with the decision nodes and leaf nodes. A decision node has at least two branches. The leaf nodes show a classification or decision. We can't accomplish more split on leaf nodes-The uppermost decision node in a tree that relates to the best predictor called the root node. Decision trees can deal with both categorical and numerical data.

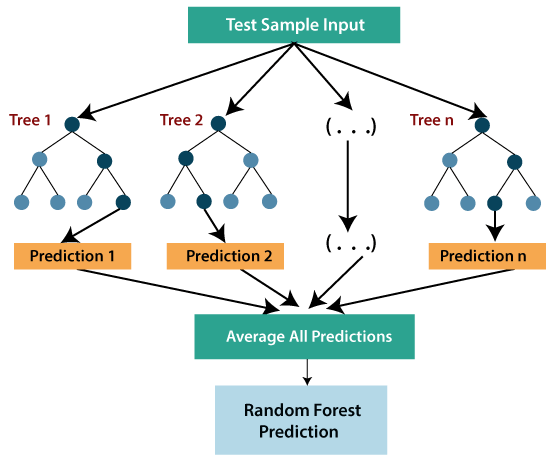
## **Key factors:**

### **Entropy:**

Entropy refers to a common way to measure impurity. In the decision tree, it measures the randomness or impurity in data sets.

### **Information Gain:**

Information Gain refers to the decline in entropy after the dataset is split. It is also called **Entropy Reduction**. Building a decision tree is all about discovering attributes that return the highest data gain.



1. **XGBoost:**

To understand XGBoost we have to know gradient boosting beforehand.

* **Gradient Boosting-**

Gradient boosted trees consider the special case where the simple model is a decision tree

****

In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w= [2, 0.1, -1]).

When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

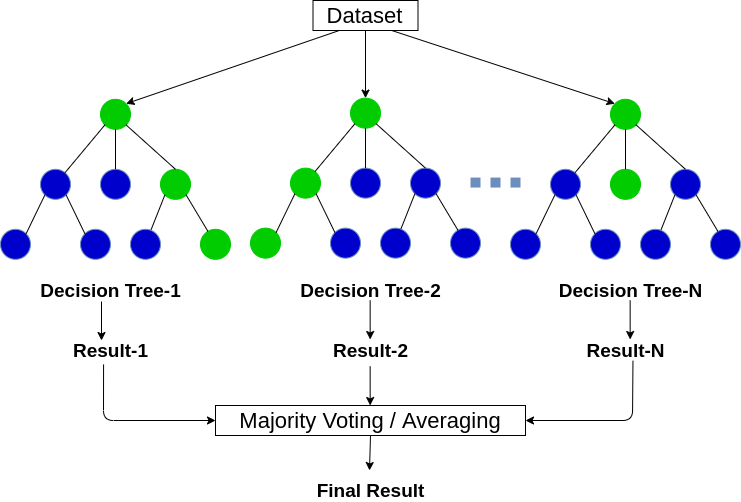
**XGBoost** is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

1. **Random Forest Classifier:**

Random forest is a [supervised learning algorithm](https://builtin.com/data-science/supervised-learning-python). The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the [bagging method](https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies) is that a combination of learning models increases the overall result.

Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor.Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).



**Model Adjustment on various Evaluation matrices:**

**HyperParameter Tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV**

Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

1. **Randomized Search CV**

In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist’s control

# Bayesian Optimization

# Bayesian Hyperparameter optimization is a very efficient and interesting way to find good hyperparameters. In this approach, in naive interpretation way is to use a support model to find the best hyperparameters. A hyperparameter optimization process based on a probabilistic model, often Gaussian Process, will be used to find datafrom data observed in the later distribution of the performance of the given models or set of tested hyperparameters.

**Conclusion:**

On performing various operations on the Dataset the conclusion is as follows:

* The chance of CHD based or education level are totally independent based on different level of education.
* People who are on BPMeds are tend to have chances of CHD.
* If a person is having prevalent stroke then the chance of CHD is **99**%.
* People who are hypertensive can also be a patient to CHD.
* The chance of getting CHD drastically increases after **60**+ age.
* Diabetic patients are also prone to CHD as **86**% people got CHD who were diabetic.
* The number of CHD patients are more in male as compare to females.
* Systolic BP of more than 150 is really dangerous and in case of more than **250** person can die due to stroke.
* High blood pressure is also not good for CHD as more than **70%** got CHD who had high BP and systolic blood pressure and diastolic BP.