***A Data Driven Approach: Heart Disease Prediction by using Clustering and Supervised ML methods.***

Sriraman Gurumanchi

Ml Final Project

***Background:***

***Introduction:***

The project aims to explore the use of clustering and supervised machine learning algorithms to predict the risk of developing heart disease based on patient data.

Cardiovascular disease is a leading cause of death worldwide, and early diagnosis is crucial for successful treatment. Every forty seconds, one person is killed by cardiovascular disease. Heart disease prediction is one of the growing areas for prediction in the field of medical science. On a daily basis, a vast amount of patient-related information is maintained. Keep data are frequently utilized as a source of predicting the possibility of future diseases, causing ML techniques to play a vital role in information extraction and prediction. Several data processing strategies for the prediction of cardiovascular diseases are proposed, with various levels of accuracy. However, the accuracy of every technique is determined by the number of attributes included and the Machine Learning techniques used.

***Problem Statement :***

Our problem statement is to predict the risk of developing heart disease based on patient data using clustering and Supervised machine learning techniques.

***Description of Dataset :***

The dataset used in this project is the Framingham Heart Study dataset, which contains information on risk factors for heart disease in 4,434 patients over a period of 10 years. The dataset includes demographic, behavioral, and medical information, such as age, gender, blood pressure, cholesterol levels, smoking habits, and history of diabetes and cardiovascular disease. By identifying groups of patients with similar risk factors and selecting the most important features for prediction, the aim is to build an accurate and robust model for early diagnosis and treatment of heart disease.

***Solution Aprroach :***

Our solution approach involves the following steps:

1. ***Data Cleaning and Preprocessing:*** We will start by cleaning and preprocessing the dataset. This will include handling missing values, scaling and standardizing the features, and encoding categorical variables.
2. **Exploratory Data Analysis(EDA) and Data visualization :** Once the data gets preprocessed , data visualization is done by statistical insights from the data , Histogram Plots and the distributions of the different attributes, correlations of the attributes with each other and the target variable
3. **Clustering analysis:** Once after EDA and visualization, clustering techniques such as k-means will be used to group similar patients based on their demographic and health characteristics. The number of clusters will be determined using techniques such as the elbow method.
4. **Feature Selection:** We will use feature selection techniques such as correlation analysis and the Select Best method to identify the most important features for predicting heart disease risk.
5. **Supervised Learning:** We will train and evaluate several supervised learning models such as logistic regression, decision tree, and random forest on the preprocessed and clustered data. We will compare the performance of these models and select the best performing one for heart disease prediction based on accuracy.
6. **Model Evaluation and Cross validation :** We will evaluate the performance of the models on a separate test set using metrics such as accuracy, precision, recall, and F1-score. We will also perform cross-validation and hyperparameter tuning to ensure the robustness of our model. Based on results, we will select best model for heart disease prediction.

Work flow :

Loading of Dataset

Data PreProcessing

EDA and Data visualization

Clustering Analysis

Feature Selection

Supervised Methods

Model Evaluation and Cross Validation

**Dataset :**

The Framingham Heart study is a long-term prospective analysis of the origin of cardiovascular disease in a free-living population in Framingham, Massachusetts.

The dataset is a rather small subset of possible FHS dataset, having 4240 observations and 16 variables. The variables are as follows:

sex : male: 1 if male, 0 if female

age: age of the participant in years

education: level of education of the participant (1 = some high school, 2 = high school or GED, 3 = some college or vocational school, 4 = college)

currentSmoker: whether or not the participant is currently a smoker (1 = yes, 0 = no)

cigsPerDay: number of cigarettes smoked per day

BPMeds: whether or not the participant is on blood pressure medication (1 = yes, 0 = no)

prevalentStroke: whether or not the participant has had a stroke (1 = yes, 0 = no)

prevalentHyp: whether or not the participant has hypertension (1 = yes, 0 = no)

diabetes: whether or not the participant has diabetes (1 = yes, 0 = no)

totChol: total cholesterol level of the participant in mg/dL

sysBP: systolic blood pressure of the participant in mmHg

diaBP: diastolic blood pressure of the participant in mmHg

BMI: body mass index of the participant

heartRate: resting heart rate of the participant in beats per minute

glucose: fasting blood glucose level of the participant in mg/dL

TenYearCHD: whether or not the participant has a 10-year risk of coronary heart disease (CHD) (1 = yes, 0 = no)

**Data PreProcessing :**

The steps involved in data cleaning and preprocessing are as follows:

Handling Missing Values: We need to check for missing values in the dataset and handle them appropriately. We can either drop the rows with missing values or fill in the missing values using imputation techniques.

Scaling and Standardizing: Scaling and standardizing the data helps to bring all the variables to the same scale and avoids any issues with variable units. We can use techniques normalization for scaling and standardizing the data.

Encoding Categorical Variables: We need to encode categorical variables into numerical values to use them in modeling. We can use techniques such as one-hot encoding or label encoding for this purpose.

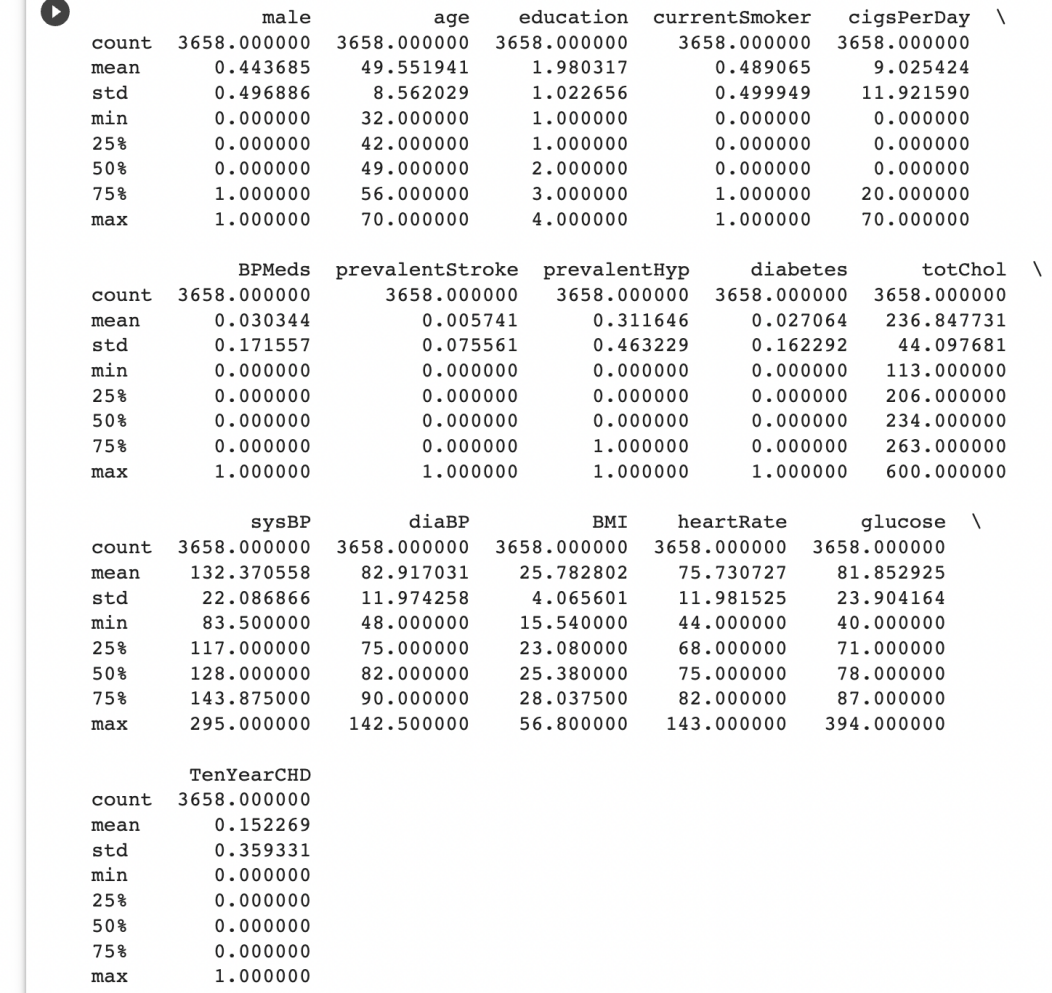
**Exploratory Data Analysis(EDA) and Data visualization :**

Once the data gets preprocessed , data visualization is done by statistical insights from the data,

Summary Statistics of data :

The mean age of the individuals is around 49 years, and the majority of individuals are male (44%). The mean education level is around 2, which corresponds to some college education. About half of the individuals are current smokers, with an average of 9 cigarettes per day.

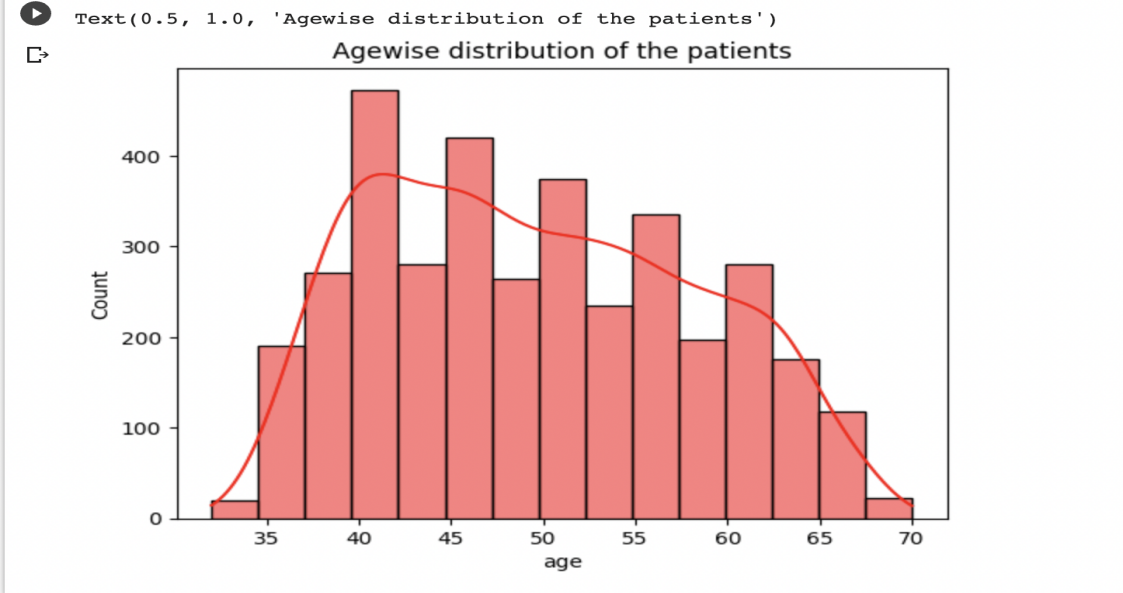
The prevalence of stroke, hypertension, and diabetes in the population is relatively low, at less than 1% for stroke, around 31% for hypertension, and around 3% for diabetes. The mean total cholesterol level is around 237 mg/dL, and the mean BMI is around 26. The mean heart rate is around 76 beats per minute, and the mean glucose level is around 82 mg/dL. The overall prevalence of coronary heart disease in the next 10 years is around 15%



Data Distribution :

Age wise Distribution :

Patients ranging from Age 40 to 50 are in Majority followed by 50 to 70.



Number of people who have disease vs age :

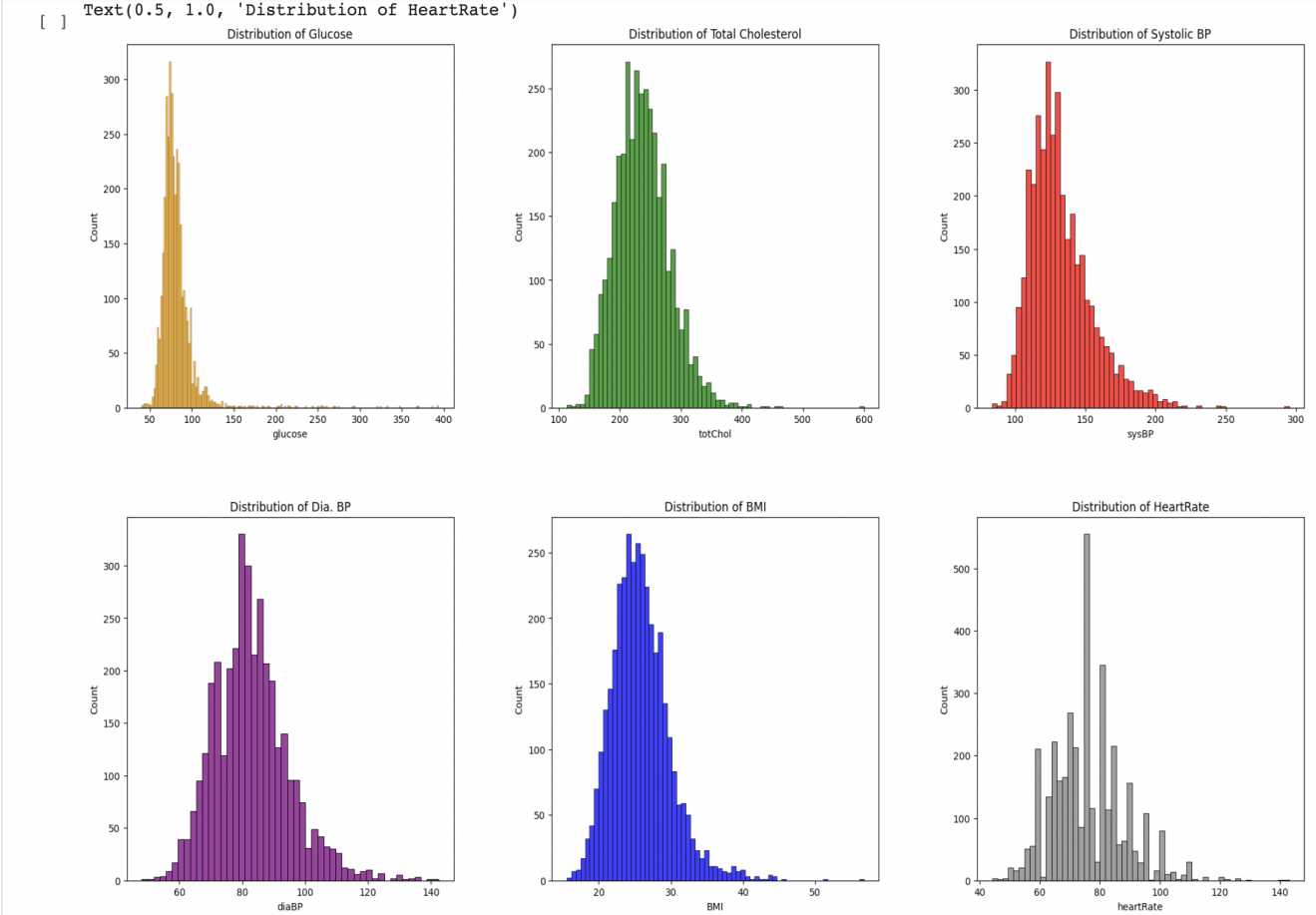
The people with the highest risk of developing CHD are betwwen the ages of 51 and 63.



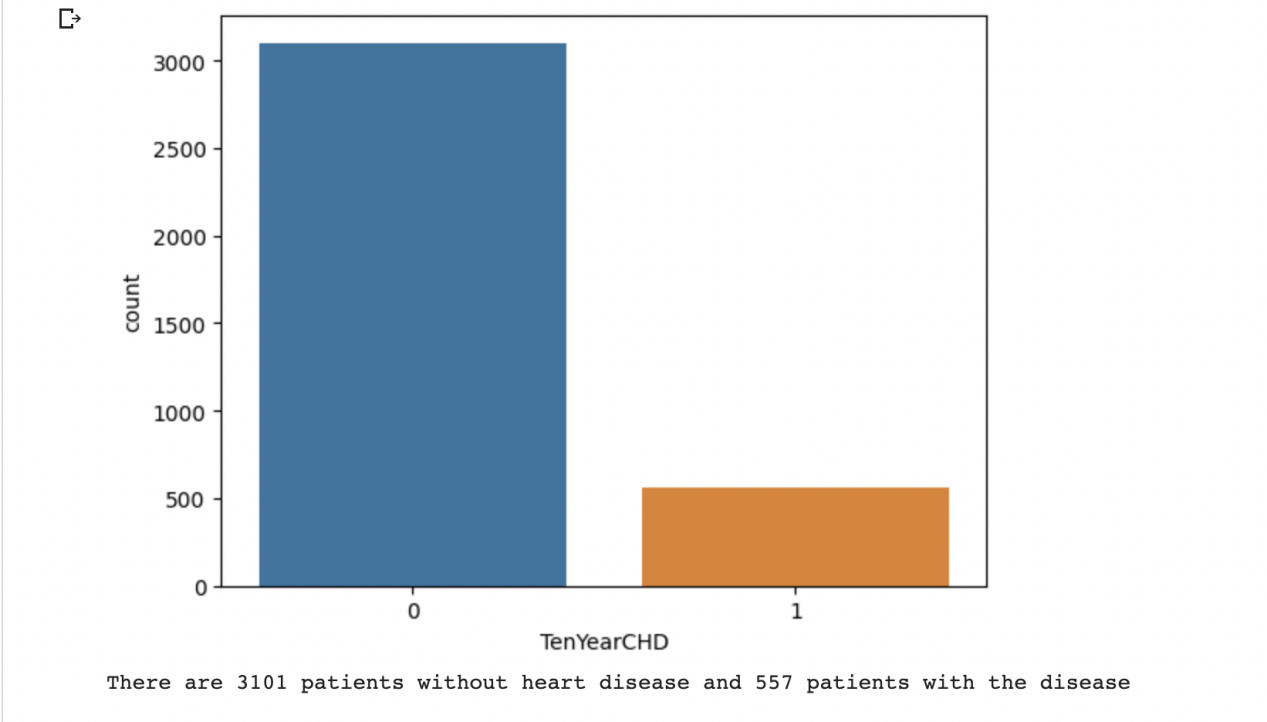
Distribution of continous variables :

We can see Glucose, Total Cholesterol, Systolic BP & BMI is Right Skewed.

While Diastolic BP & Heart Rate are close to Normal / Gaussian Distribution.

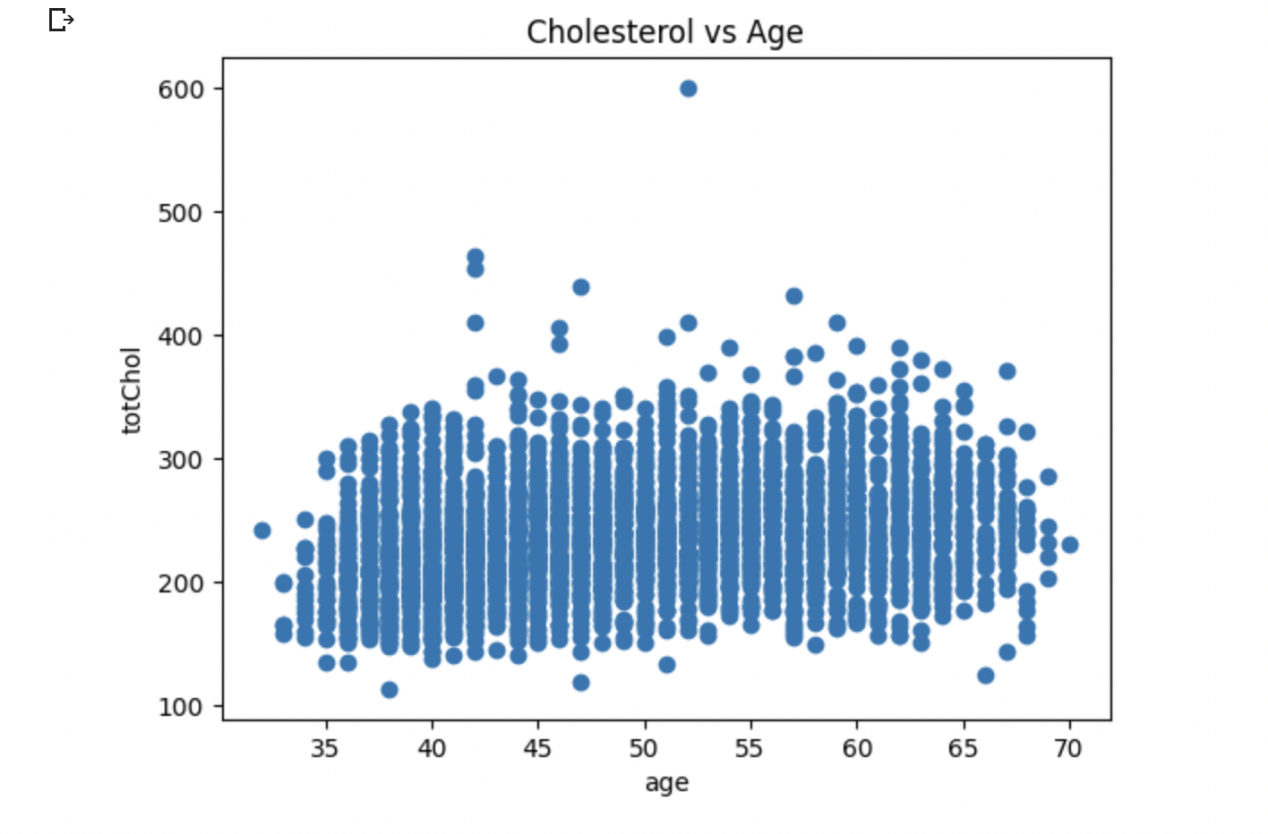


Case Counts :



Scatter plot :

Cholesterol vs Age



There appears to be a positive correlation between age and cholesterol levels, with cholesterol levels increasing as age increases. However, there is also a lot of variability in the data, with some individuals having very high cholesterol levels at a relatively young age and vice versa. This suggests that age is not the only factor influencing cholesterol levels.

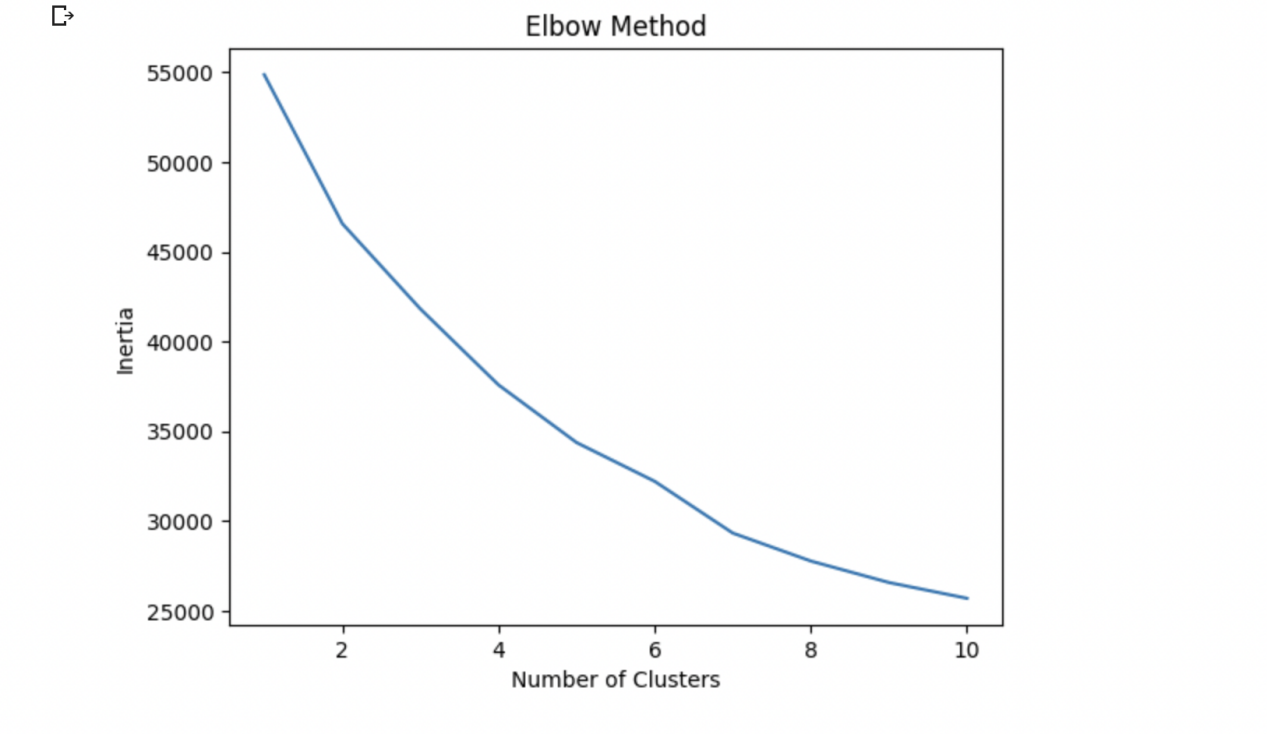
**Clustering analysis :**

In this clustering analysis, we used the K-means clustering to group patients based on their demographic and health characteristics.

1. means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into k distinct non-overlapping clusters. The objective is to minimize the sum of distances between the data points and the cluster centroid, to identify the correct group each data point should belong to.

The algorithm works by iteratively assigning each data point to one of the k clusters based on their similarity to the centroid (mean) of each cluster. The algorithm then computes the mean of all the data points in each cluster to update the centroid.

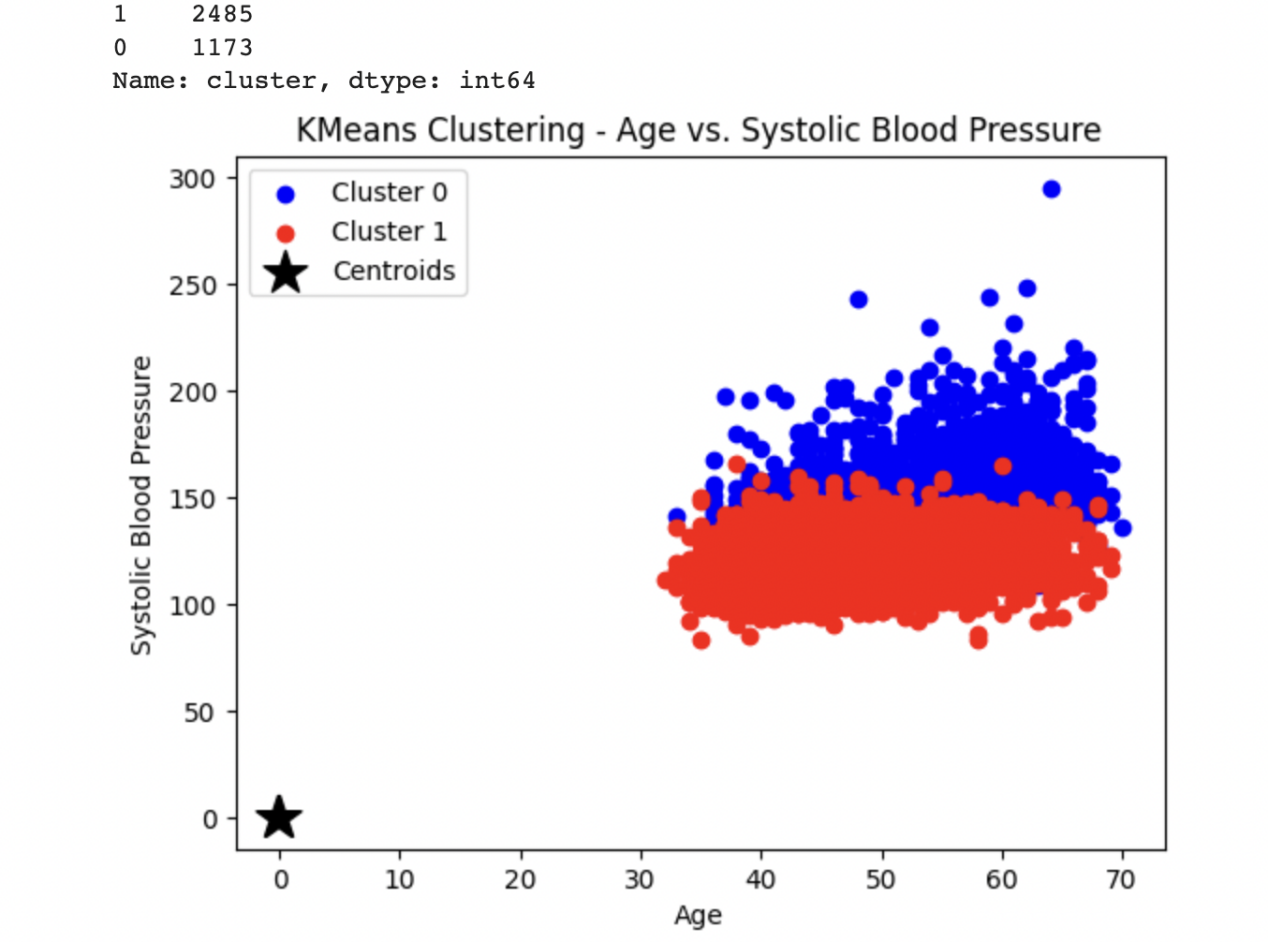
The algorithm requires the specification of the number of clusters k, which is often determined using methods such as the elbow method. The elbow method involves plotting the within-cluster sum of squares (WCSS) (inertia) against the number of clusters and identifying the point where the reduction in WCSS begins to level off.



We used the elbow method to determine the optimal number of clusters, and found that two clusters provided the best balance between variance explained and number of clusters.

After applying K-means clustering with k=2, we obtained two clusters with different sizes, cluster 0 (n=2485) and cluster 1 (n=1173). We visualized the clusters using a scatter plot with age on the x-axis and systolic blood pressure (sysBP) on the y-axis. The plot showed that the two clusters had different ranges of ages and blood pressures.

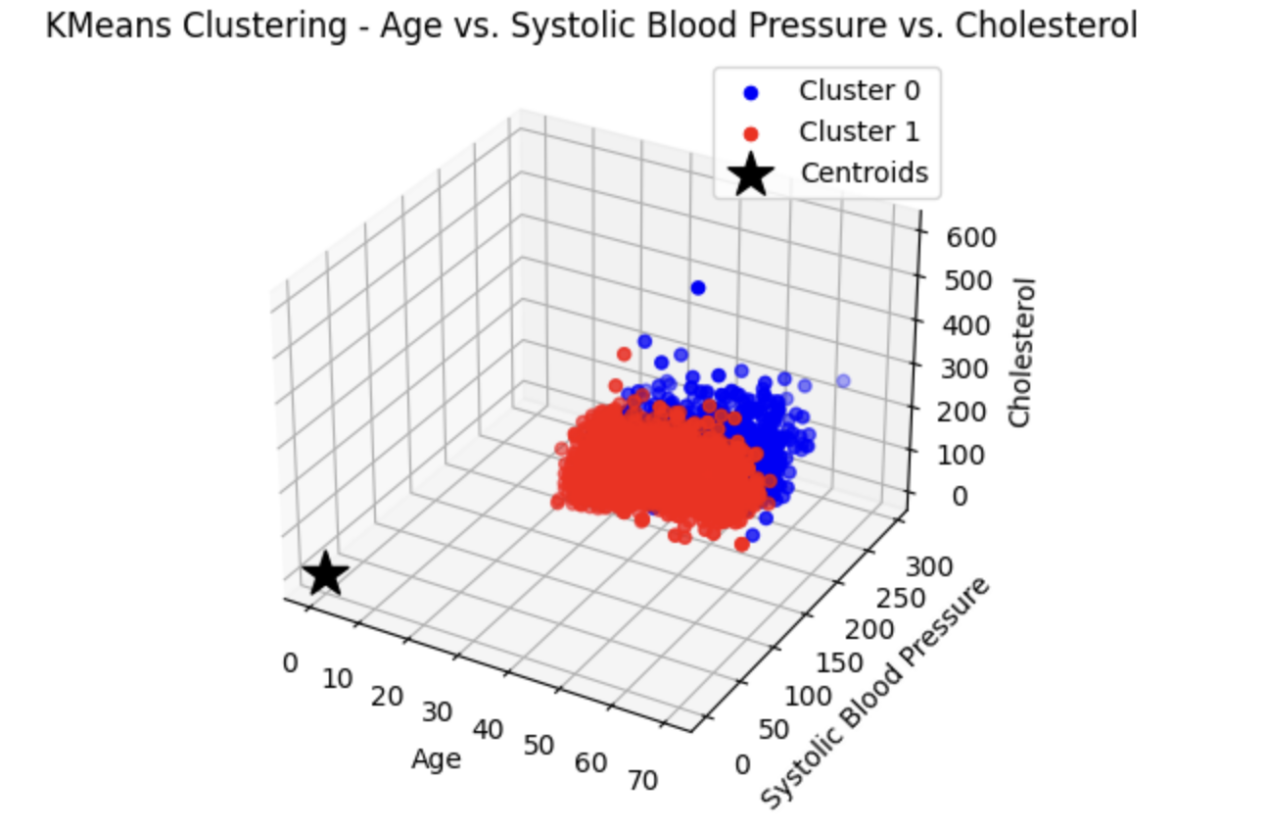
The scatter plot shows that there are two distinct clusters of individuals: one with low age and low systolic blood pressure (shown in blue) and another with higher age and higher systolic blood pressure (shown in red). The black star markers on the plot represent the centroids of the clusters, which indicate the center point of each cluster.



Cluster Visualization in 3D scatter plot :

The 3D scatter plot shows that the two clusters identified by the K-means clustering algorithm are somewhat separable in three dimensions. The blue cluster tends to have lower values for all three variables (age, systolic blood pressure, and cholesterol) compared to the red cluster, which tends to have higher values for all three variables. This suggests that the algorithm has identified a pattern where individuals in the blue cluster may be at lower risk of developing coronary heart disease (CHD) compared to individuals in the red cluster.

However, there is still some overlap between the two clusters, and the centroids are not located at the center of each cluster.



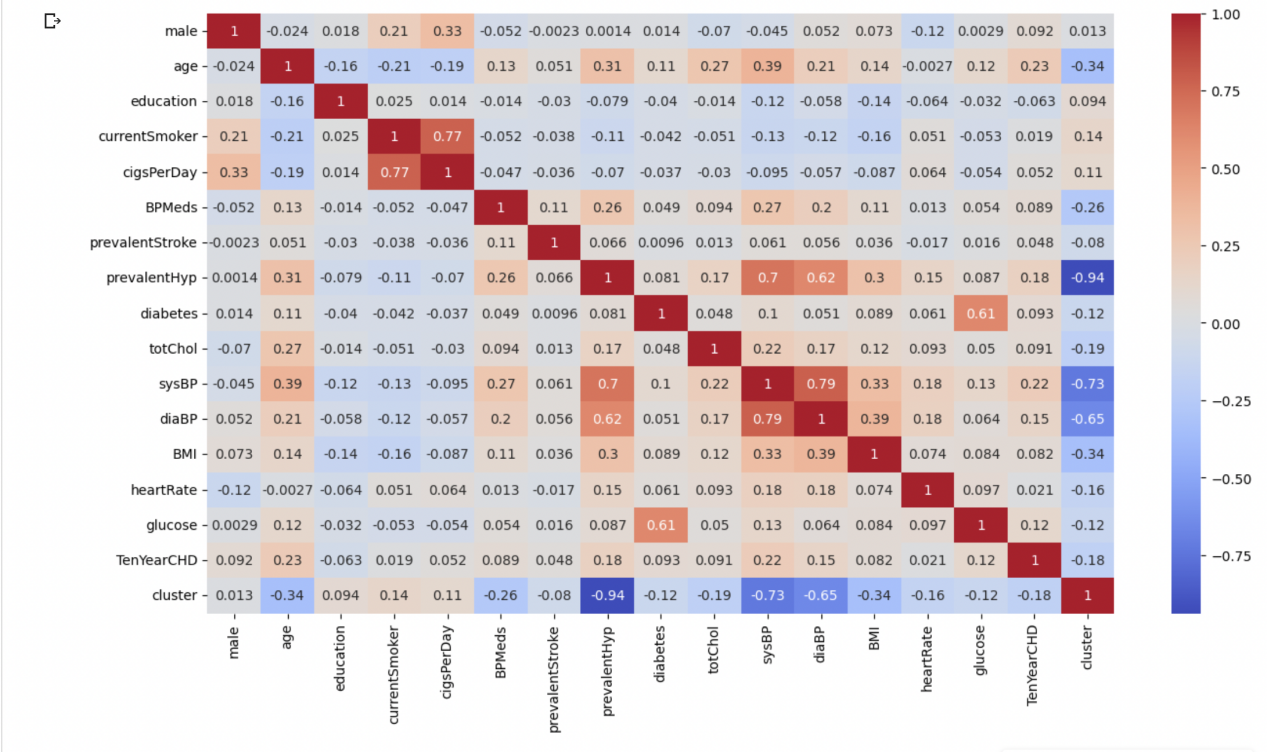
**Feature Selection :**

Feature selection is an important step in building a predictive model. In this dataset, Correlation analysis and Select Best method are techniques used to identify the most important features.

Co relation Analysis :

In correlation analysis, we can calculate the correlation coefficient between each feature and the target variable (TenYearCHD) and select the features with the highest correlation coefficients. The correlation coefficient ranges from -1 to 1, with values closer to 1 indicating a strong positive correlation, values closer to -1 indicating a strong negative correlation, and values close to 0 indicating no correlation.

Based on the correlation analysis, it appears that age, systolic blood pressure (sysBP), prevalent hypertension (prevalentHyp), diastolic blood pressure (diaBP), glucose, diabetes, male gender, total cholesterol (totChol), blood pressure medications (BPMeds), BMI, and clustering labels have the strongest correlation with the target variable, TenYearCHD. On the other hand, education, heart rate, current smoker status, cigsPerDay, and prevalent stroke have a weaker correlation with the target variable. This information can be used to select the most important features for predicting heart disease risk.



Select Best Method :

By select best method algorithm, we performed the feature selection using SelectKBest with f\_classif scoring function, and assigned k = 2, number of top features to select. The results were observed as

**Selected features: Index(['age', 'sysBP'], dtype='object')**

We see that age and the systolic blood pressures are selected as the most important features for predicting the Ten year risk of developing CHD.

**Supervised Learning Methods :**

Supervised learning is a type of machine learning where the algorithm learns from a labeled dataset, meaning that it has access to both input features and corresponding output labels. In this case, we have a labeled dataset of patients with their respective demographic, health characteristics, and a target variable indicating whether they have developed heart disease within ten years or not.

We will use this labeled dataset to train and evaluate several supervised learning models such as logistic regression, decision tree, and random forest. These models will learn to predict the target variable based on the input features, and we will evaluate their performance using metrics such as accuracy, precision, recall, and F1 score.

Once we have evaluated the performance of these models, we will select the best performing one for heart disease prediction. This best-performing model will be used to predict the likelihood of a patient developing heart disease based on their demographic and health characteristics.

Splitting of data into test and train data :

In supervised machine learning, we typically split the available data into two subsets: a training set and a testing set. The purpose of this split is to use the training set to train our model on how to make predictions, and then use the testing set to evaluate the performance of the model on data that it hasn't seen before.

The train\_test\_split() function from the sklearn.model\_selection module allows us to split our dataset into random train and test subsets. We typically use 80% of the data for training and 20% for testing, although this split can vary depending on the size of the dataset.

In this model, we split the data into test and train data by using important features, where we split the preprocessed and selected feature data into training and testing data sets and begin to train the models.

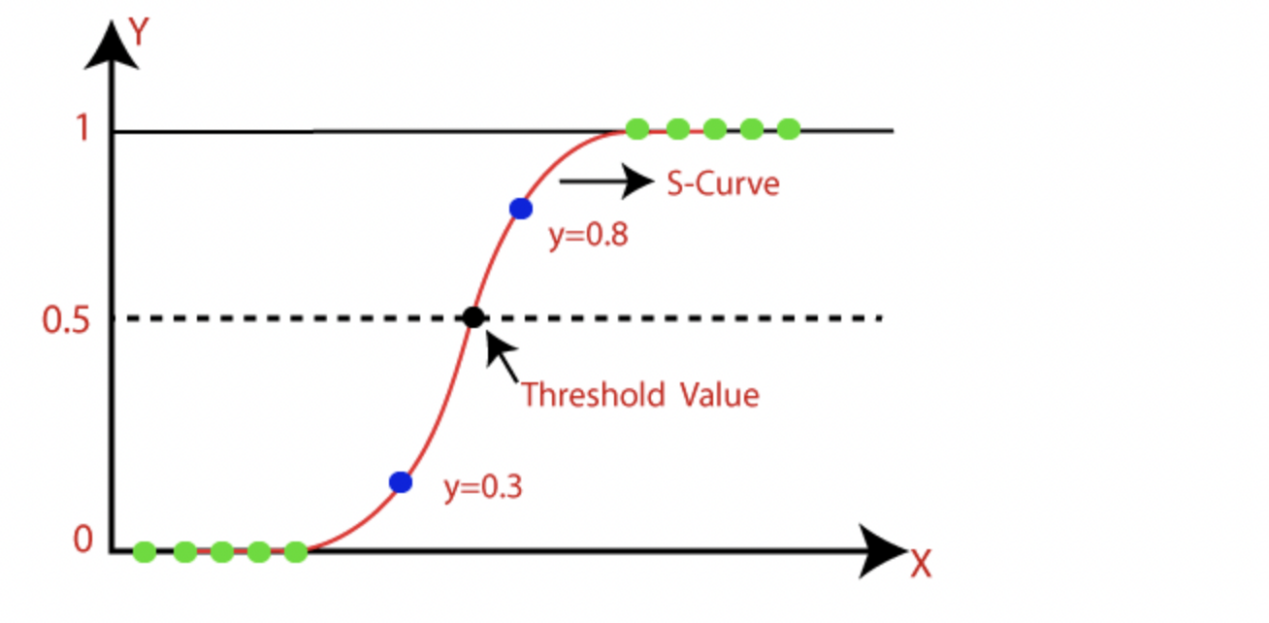
**Logistic Regression :**

Logistic Regression is a type of supervised learning algorithm used for binary classification problems, where the target variable is categorical with two classes. It models the probability of an event occurring, such as the occurrence of a disease, based on one or more input features. The output of the logistic regression model is a probability score between 0 and 1, and a decision threshold is used to classify the input data as belonging to one of the two classes. Logistic Regression uses a logistic function to model the relationship between the input features and the probability of the target variable.

The logistic function is a sigmoid function that transforms the input features into a probability score. The model is trained using maximum likelihood estimation to find the optimal coefficients for the input features that maximize the likelihood of the observed data. The performance of the model can be evaluated using metrics such as accuracy, precision, recall, and F1 score.

Logistic Function :

* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1.
* The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.



Once we have our training and testing sets, we create a logistic regression model object using the LogisticRegression() function from the sklearn.linear\_model module. We then fit this model to our training data using the .fit() method of the model object.

After training the model, we use the .predict() method to generate predictions on the testing set. We then evaluate the performance of the model using the accuracy\_score() function from the sklearn.metrics module, which compares the predicted values to the actual values in the testing set to calculate the accuracy of the model.

Finally we got the Accuracy score for Logistic regression model is 83.46%.

**Random Forest :**

Random Forest is a supervised machine learning algorithm made up of decision trees. Random Forest is used for both classification and regression. It builds a forest in the form of an ensemble of decision trees which adds more randomness while growing the trees. While splitting a node, the algorithm searches for the best features from the random subset of features which adds more diversity, thereby resulting in a better model.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



The following steps to build and evaluate a Random Forest model:

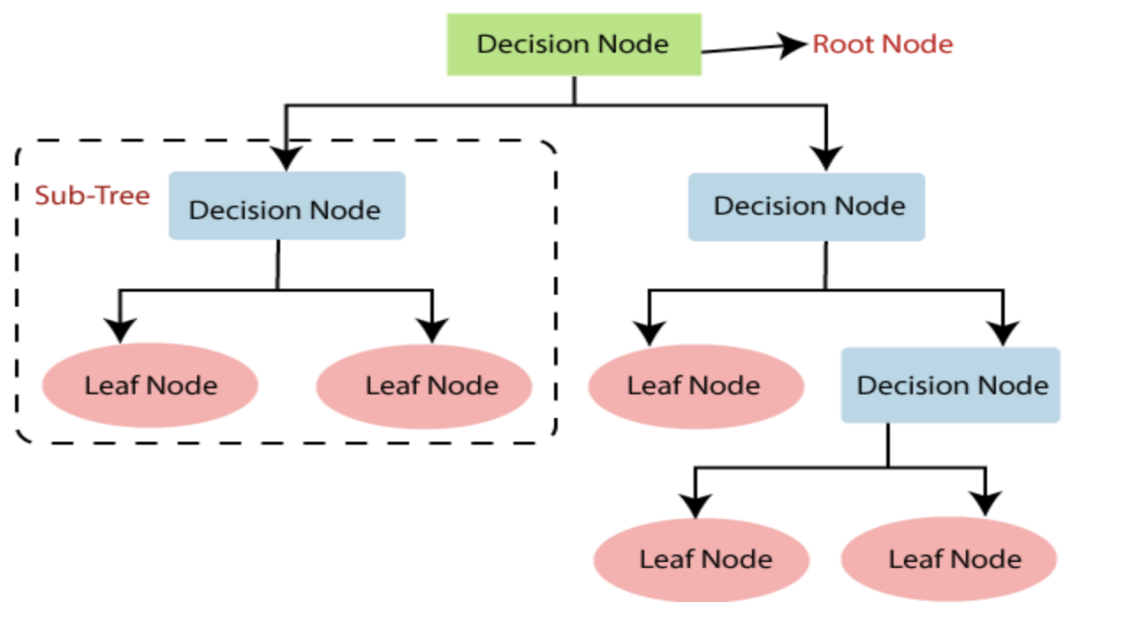
* First, we splits the preprocessed and selected feature data into training and testing sets using the train\_test\_split function from scikit-learn.
* Next, creates an instance of the Random Forest classifier using the RandomForestClassifier class from scikit-learn, and sets the number of decision trees (n\_estimators) to 100 and the random state to 42.
* Then trains the model on the training data using the fit method. After that, it makes predictions on the testing data using the predict method.
* Finally, evaluates the performance of the model using the accuracy\_score function from scikit-learn and we got the accuracy for Random forest is 83.74%.

**Decision Tree :**

A decision tree is a supervised learning algorithm that can be used for both classification and regression tasks. It works by recursively splitting the data based on the features and their values to create a tree-like model of decisions and their possible consequences. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decision tree algorithm works by recursively splitting the data into smaller subsets based on the features and their values. The algorithm starts at the root node of the tree and selects the best feature to split the data into two subsets. It continues to split the data at each subsequent node until it reaches the leaf nodes, which represent the final classification. Each node in the decision tree corresponds to a test on a particular feature, and each branch corresponds to the outcome of the test. The leaves of the tree represent the class labels.

.

In the decision tree model, we first split our preprocessed and selected feature data into training and testing sets using the train\_test\_split() function from sklearn.model\_selection. Then, we initialize the decision tree classifier using the DecisionTreeClassifier() function from sklearn.tree. We then fit the model on the training data using the fit() function.

Next, we use the trained decision tree model to predict the target values on the testing data using the predict() function. Finally, we calculate the accuracy score of the model using the accuracy\_score() function from sklearn.metrics.

We got the accuracy score for Decision tree is 73.63%.

**Comparison of models :**

By comparing the accuracy scores of three Supervised models- Logistic regression, Random Forest and Decision tree algorithms, The best model selected is Random Forest because the Random Forest model performed slightly better than the Logistic Regression model, with an accuracy of 0.8374 compared to 0.8347. The Decision Tree model had the lowest accuracy score of 0.7363, suggesting that it may not be the best model for this particular dataset.

However , accuracy alone may not always be the best metric to evaluate a model's performance, and other metrics such as precision, recall, and F1 score may also be considered

**Model Evaluation and Cross validation:**

Model evaluation is a crucial step in the machine learning pipeline as it helps us understand how well our model performs on unseen data. In this step, we will evaluate the performance of the models on a separate test set using metrics such as accuracy, precision, recall, and F1-score. We will also perform cross-validation and hyper parameter tuning to ensure the robustness of our model. Based on results, we will select best model for heart disease prediction.

Accuracy measures the overall accuracy of the model in predicting the target variable correctly. Precision measures the proportion of true positives (correctly identified positive samples) among all the positive predictions made by the model. Recall measures the proportion of true positives among all actual positive samples. F1-score is the harmonic mean of precision and recall and provides a balance between the two.

In addition to these evaluation metrics, we also perform cross-validation to ensure the robustness of our model. Cross-validation involves dividing the data into several subsets, training the model on different subsets, and evaluating its performance on the remaining subset. By doing so, we can obtain a more accurate estimate of the model's performance on unseen data.

Hyper parameter tuning is another crucial step in ensuring the robustness of our model. Hyper parameters are parameters that are not learned during training and need to be set manually. By tuning the hyper parameters, we can find the best combination of hyper parameters that maximizes the model's performance on the validation set. This helps us avoid over fitting or under fitting of the model.

The following steps were done for Model evaluation and Cross vallidation :

* Imported necessary libraries such as scikit-learn's RandomForestClassifier, LogisticRegression, Decisiontree, GridSearchCV, StratifiedKFold, and evaluation metrics such as accuracy\_score, precision\_score, recall\_score, and f1\_score.
* Defined hyper parameters for each model, including:
* Random Forest: number of estimators, maximum depth, minimum samples split, and minimum samples leaf.
* Logistic Regression: regularization parameter C and penalty (l1 or l2).
* Decision Tree: maximum depth, minimum samples split, and minimum samples leaf.
* Defined a cross-validation strategy, StratifiedKFold, with 5 folds and a random state of 42.
* Performed a grid search using GridSearchCV to find the best hyperparameters for each model based on accuracy score. The grid search was performed on the training data using the defined hyperparameters and cross-validation strategy.
* Evaluated the best model on the test set using the evaluation metrics such as accuracy\_score, precision\_score, recall\_score, and f1\_score, which will help us determine if our model is performing well enough or needs further improvement.

**Results Interpretation :**

**Random forest :**

Best hyper parameters: {'max\_depth': 20, 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 200}

Accuracy: 0.8387978142076503

Precision: 0.8333333333333334

Recall: 0.040983606557377046

F1-score: 0.07812499999999999

**Logistic regression :**

Best hyper parameters: {'C': 0.1, 'penalty': 'l2'}

Accuracy: 0.8319672131147541

Precision: 0.4

Recall: 0.01639344262295082

F1-score: 0.031496062992125984.

**Decision tree :**

Best hyper parameters: {'max\_depth': 5, 'min\_samples\_leaf': 2, 'min\_samples\_split': 2}

Accuracy: 0.8306010928961749

Precision: 0.4375

Recall: 0.05737704918032787

F1-score: 0.10144927536231883

Based on the results obtained, we can see that the Random Forest classifier outperformed both the Logistic Regression and Decision Tree classifiers, as it achieved the highest accuracy and F1-score on the test set.

The Logistic Regression model had a very low recall and F1-score, indicating that it had difficulty identifying positive cases.

The Decision Tree model had a slightly higher recall than the Logistic Regression model, but still had a low F1-score.

Overall, After Model evaluation and cross validation analysis, Random forest model performed good as compared to both the Logistic Regression and Decision Tree classifiers.

**Working of built in Ml Model :**

Finally, We have give my new data as input for Heart disease prediction after ten years and the results are :

*The predicted risk of getting heart disease after ten years is: 2.08 %.*

**Conclusion :**

Overall, the objective of this project was to develop a machine learning model for predicting the risk of heart disease in patients based on their demographic and health characteristics. We started by cleaning and preprocessing the dataset, including handling missing values, scaling and standardizing features, and encoding categorical variables.

Next, we used clustering techniques such as k-means to group similar patients based on their characteristics and The number of clusters are determined by using the elbow method,where we assigned the k value as ‘2’ . Cluster visualization with centroid is represented by scatter plot of age vs systolic blood pressure. 3d scatter plot of age vs sysbp vs cholesterol was observed.

We also performed feature selection using correlation analysis and SelectBest method to identify the most important features for predicting heart disease risk. The results are ‘Age’ and ‘sysBp’(Systolic Blood Pressure) identified as important features for predicting heart disease risk.

We then trained and evaluated several supervised learning models such as logistic regression, decision tree, and random forest on the preprocessed and clustered data. After comparing the performance of these models, based on accuracy we selected the Random forest classifier as the best performing model .

Finally, we evaluated the performance of the models on a separate test set using metrics such as accuracy, precision, recall, and F1-score. We also performed cross-validation and hyper parameter tuning to ensure the robustness of our model.After results interpretation, Random forest model performed good as compared to both the Logistic Regression and Decision Tree classifiers.

In summary, we were able to develop a machine learning model for predicting heart disease risk with a high level of accuracy and precision. This model could be used in clinical settings or can be used for health professionals to help identify patients who are at high risk of developing heart disease and provide early interventions to prevent the disease**.**

Further research is also needed to know exact prediction by handling large data , bulding other Machine learning algorithms and use of Deep learning models for heart disease prediction.

**References :**

Prediction of cardiac arrhythmia type using clustering and regression approach (P-CA-CRA) Prathibhamol C.P,Anjana Suresh,Gopika Suresh.

Heart Disease Prediction System Using Data Mining and Hybrid Intelligent Techniques: A Review J. Vijayashree, N Ch Sriman Narayana Iyenger.

[Influence of Optimal Hyperparameters on the Performance of Machine Learning Algorithms for Predicting Heart Disease](https://www.researchgate.net/publication/368979184_Influence_of_Optimal_Hyperparameters_on_the_Performance_of_Machine_Learning_Algorithms_for_Predicting_Heart_Disease). Ghulab Nabi, AhamadShafiullah, Hira FatimaMohammed Usman

Clustering of Countries for COVID-19 Cases based on Disease Prevalence, Health Systems and Environmental Indicators. Syeda Amna Rizvi, Muhammad Umair Mujahid, Muhammad Aamir Cheema.

Integrated k-means Clustering with Nature Inspired Optimization Algorithm for the Prediction of Disease on High Dimensional Data. Afroj Alam, Mohd Muqeem.

A Survey on Data Mining Techniques for Prediction of Heart Diseases. B Senthil Kumar.

Anomaly Detection in Heart Disease Using a Density-Based Unsupervised Approach. Y. A. NanehkaranZhu Licai, Junde ChenMohsen Abdollahzadeh Aghbolagh

Prediction of Heart Disease by Clustering and Classification Techniques Prediction of Heart Disease. Reetu Singh , Rajesh Elangovan.