**Predicting Cardiovascular Disease in Individuals Between the age of 35-70**

One of the main causes of death in the world is heart disease, and its effects can be minimized by early detection and prevention. It is now possible to forecast the risk of heart disease using a variety of lifestyle and health-related indicators thanks to developments in machine learning and data analytics. These models allow both individuals and healthcare providers to better understand their risk for heart disease and take preventative measures to lower that risk.  
The problem addressed in this report is the analysis of a model for predicting heart disease risk based on clinical variables. The goal is to develop a model that accurately classifies individuals as either having heart disease or not, using a dataset containing various clinical features such as age, sex, blood pressure, cholesterol levels, and other related measurements. The report aims to evaluate the model's performance, analyze the importance of different features in predicting heart disease, and provide insights into the data preprocessing and feature selection techniques employed.

Data Collection

The dataset used to build this model is sourced from Kaggle. The dataset is used for analyzing and predicting heart disease based on the given features:

* Age: The age of the individual.
* Sex: The biological sex of the individual (0 = female, 1 = male).
* Chest pain type: The type of chest pain experienced by the individual.
* BP: Blood pressure of the individual.
* Cholesterol: Cholesterol level of the individual.
* FBS over 120: Whether the fasting blood sugar level of the individual is over 120 mg/dl (1 = true, 0 = false).
* EKG results: Results of the electrocardiogram.
* Max HR: Maximum heart rate achieved during exercise.
* Exercise angina: Whether the individual experienced angina (chest pain) during exercise (1 = yes, 0 = no).
* ST depression: ST segment depression induced by exercise relative to rest.
* Slope of ST: The slope of the ST segment during peak exercise.
* Number of vessels fluro: The number of major blood vessels colored by fluoroscopy.
* Thallium: Results of thallium stress test.
* Heart Disease: Presence or absence of heart disease (1 = presence, 0 = absence).

The dataset contains these information from different individuals and it will be used to build the model that will be used for prediction.

Data Preprocessing

The necessary libraries are imported and with their help the data is preprocessed by the following steps:

* Reading the dataset:

The dataset was read using pd.read\_csv() function, and the resulting DataFrame was assigned to the variable df.

* Handling missing values:

The code did not explicitly handle missing values in the dataset. Missing values were not imputed or removed.

* Handling categorical features:

The code used LabelEncoder from sklearn.preprocessing to convert the "Heart Disease" column (assumed to be a categorical feature) to numerical labels.

* Separating features and target variable:

The code separated the features by dropping the "Heart Disease" column and assigning the result to the variable X.

The target variable "Heart Disease" was assigned to the variable y

* Splitting data into training and testing sets:

The code used train\_test\_split from sklearn.model\_selection to split the data into training and testing sets. The testing set size was set to 20% of the data, with a random state of 42.

Feature Selection  
In my work feature selection was not explicitly performed. However, it does include the calculation of feature importances using a Random Forest Classifier.

Model Selection  
The model used to aid in the prediction is the Random Forest Model. Random Forest is a popular machine-learning algorithm used for both classification and regression tasks. It is an ensemble model that combines multiple decision trees to make predictions.

It works by :

* Training Data: Random Forest is trained on a dataset with examples that have input features and corresponding labels.
* Creating Many Trees: Random Forest creates a bunch of trees, let's say 100 trees.
* Random Sampling: Each tree is trained on a random subset of the data, where some examples may be repeated and some may be left out.
* Decisions in Trees: Each tree makes decisions based on the input features. It asks questions about the features to figure out the labels.
* Combining Predictions: When you want to predict something, like whether an email is spam or not, each tree gives its own prediction. The final prediction is the one that gets the most votes from all the trees.
* Easy and Accurate: Random Forest is good at making predictions because it combines the knowledge of many trees. It's like asking many people their opinion and going with the majority.

Model Training

The model is trained by first splitting the data into 2 parts which will be the features (“X”) and the target variable (“y”)  
A second split is also done on the data. This is the train and test split. This is where 80% of the data is used to train the model and the remaining 20% of the data is used to test the model.  
The Random Forest classifier is instantiated by creating an instance of the RandomForestClassifier class from scikit-learn. In this code, the number of estimators (decision trees) is set to 200 (n\_estimators=200). This parameter represents the number of decision trees in the Random Forest.  
  
Model Evaluation  
After building and testing the model. The accuracy was tested with data and model used and it provided 84.2% accuracy on its prediction.

In conclusion, using a Random Forest Classifier to forecast the likelihood of developing heart disease is a viable strategy. This algorithm can handle non-linear correlations between the features and the target variable as well as a high number of input features.

In comparison to other machine learning algorithms, the random forest approach performs well in terms of accuracy and stability. But it's crucial to remember that the caliber of the data used to train the model affects how accurate the predictions turn out to be. Therefore, before training the model, it is essential to gather high-quality data and rigorously pre-process it.

In order to make sure the model generalizes effectively to fresh, untested data, it is also crucial to verify the model using a different test dataset. Machine learning models, such as Random Forest Classifier, can be a useful tool in forecasting the risk of heart disease and enabling people and healthcare professionals to take preventative measures. With the correct data and approach.