CARDIOVASCULAR DISEASE DETECTION

Course Code: CS354N

PREPARED BY :

Atish Kumar (210001006)
Nilesh Kumar (210001043)
Vartesh (210001073)

PRESENTED TO :

Dr. Aruna Tiwari

Introduction:

Project Title:

Detection of cardiovascular disease using different types of machine learning models.

Problem Description:

Cardiovascular disease (CVD) remains one of the leading causes of mortality globally, accounting for a significant portion of deaths each year. Early detection and intervention are crucial in managing and preventing the progression of cardiovascular conditions. Machine learning (ML) techniques offer promising avenues for improving the accuracy and efficiency of CVD detection.

The problem at hand involves developing a robust machine learning model capable of accurately predicting the presence or likelihood of cardiovascular disease based on various patient attributes and clinical indicators. This predictive model aims to assist healthcare professionals in identifying individuals at high risk of developing CVD, thereby enabling timely interventions and personalized treatment plans.

Analysis And Design:

Data Collection:

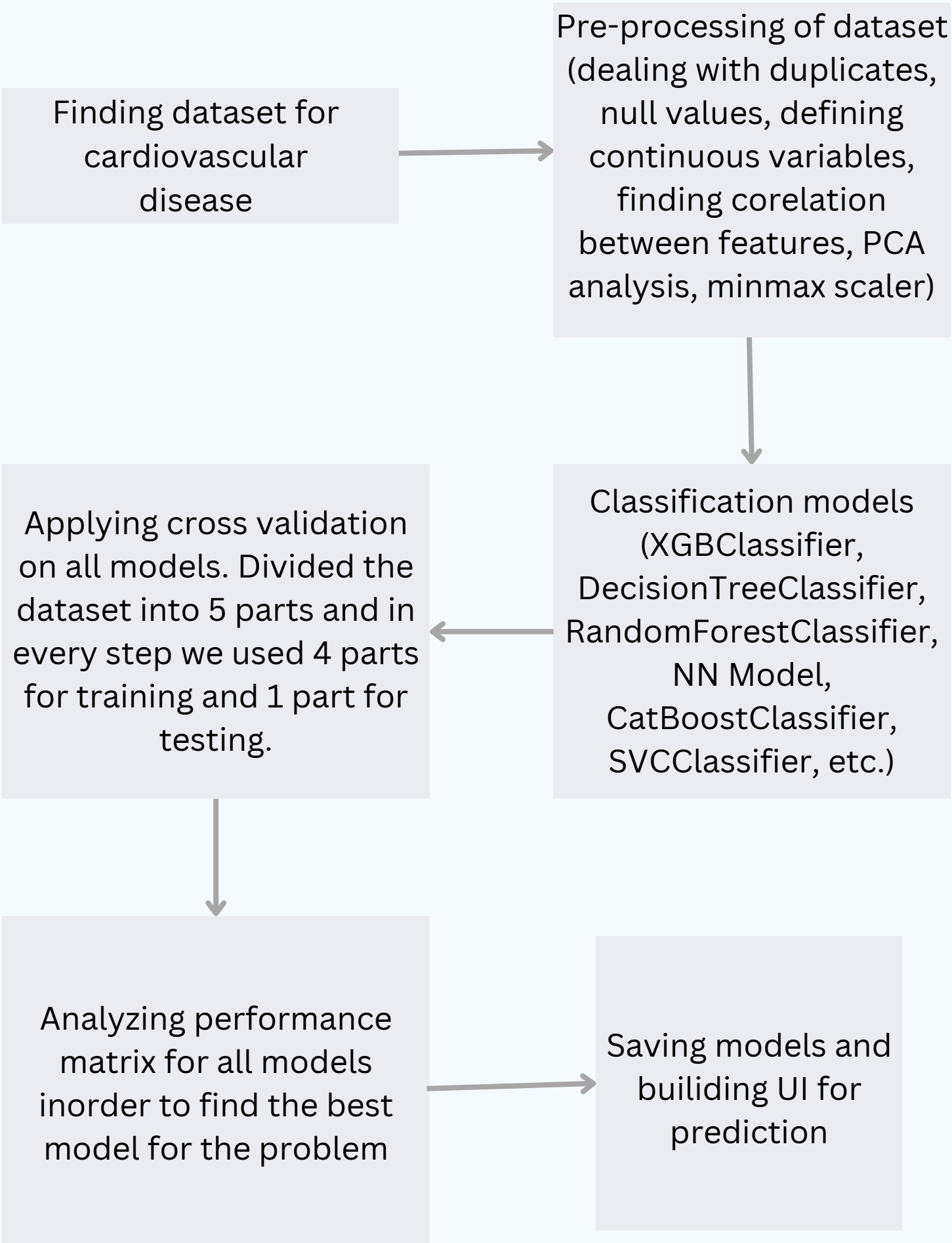
This heart disease dataset is acquired from one of the multispecialty hospitals in India. Over 14 common features which makes it one of the heart disease dataset available so far for research purposes. This dataset consists of 1000 subjects with 12 features. This dataset will be useful for building a early-stage heart disease detection as well as to generate predictive machine learning models. Dataset Link

Data Preprocessing:

The preprocessing of the dataset commenced with data reading and a thorough check for duplicate entries, ensuring data integrity from the outset. Subsequently, the presence of null values was examined, revealing a single column, 'serumCholesterol,' with missing data. Interpolation was then employed to fill in these null values, ensuring completeness of the dataset. To streamline the dataset for machine learning, the patient ID column was dropped as it held no relevance to the training of the model.

Additionally, a comprehensive analysis of continuous columns, defined as those with more than ten unique values, was conducted using the '.describe()' function, offering insights into the distribution and statistics of these features. Potential outliers within these continuous columns were identified through box plots and further explored via histograms, facilitating a deeper understanding of the data distribution.Leveraging Principal Component Analysis (PCA), the dataset was transformed from its original 12-dimensional space into a two-dimensional representation, aiding visualization and comprehension. A correlation matrix was then constructed to assess the relationships between pairs of features, highlighting age as the least correlated with the target variable, prompting its removal from the feature set. Finally, to standardize the data for machine learning algorithms, MinMaxScaler was applied to scale the feature values into the range of 0 to 1, optimizing model performance. The dataset was ultimately partitioned into feature and target arrays, denoted as X and y respectively, in preparation for model training. Through these preprocessing steps, the dataset was meticulously prepared to extract meaningful insights and facilitate accurate cardiovascular disease detection using machine learning techniques.

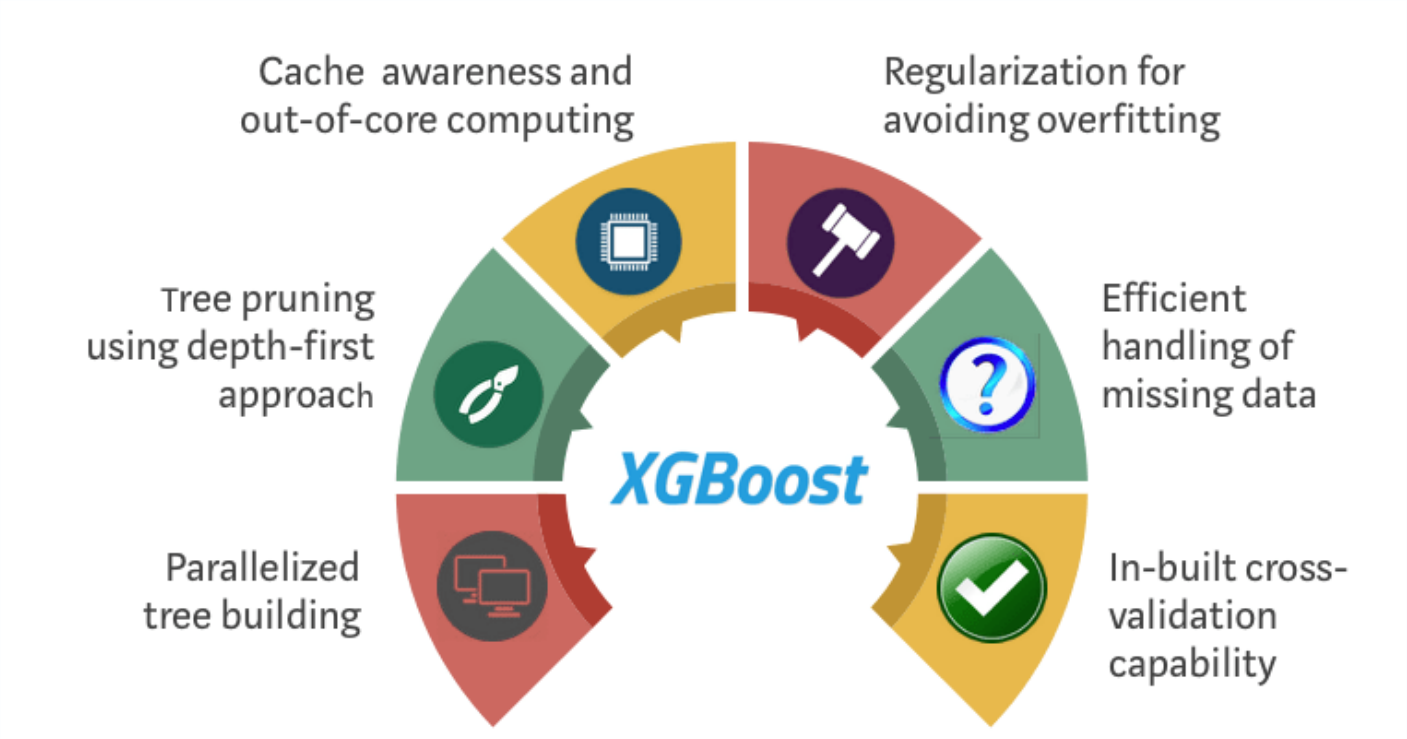
Flow Chart:


Finding dataset for cardiovascular disease
Pre-processing of dataset (dealing with duplicates, null values, defining continuous variables, finding corelation between features, PCA analysis, minmax scaler)
Classification models (XGBClassifier, DecisionTreeClassifier, RandomForestClassifier, NN Model, CatBoostClassifier, SVCClassifier, etc.)
Applying cross validation on all models. Divided the dataset into 5 parts and in every step we used 4 parts for training and 1 part for testing.
Analyzing performance matrix for all models inorder to find the best model for the problem
Saving models and builiding UI for prediction

Algorithms:

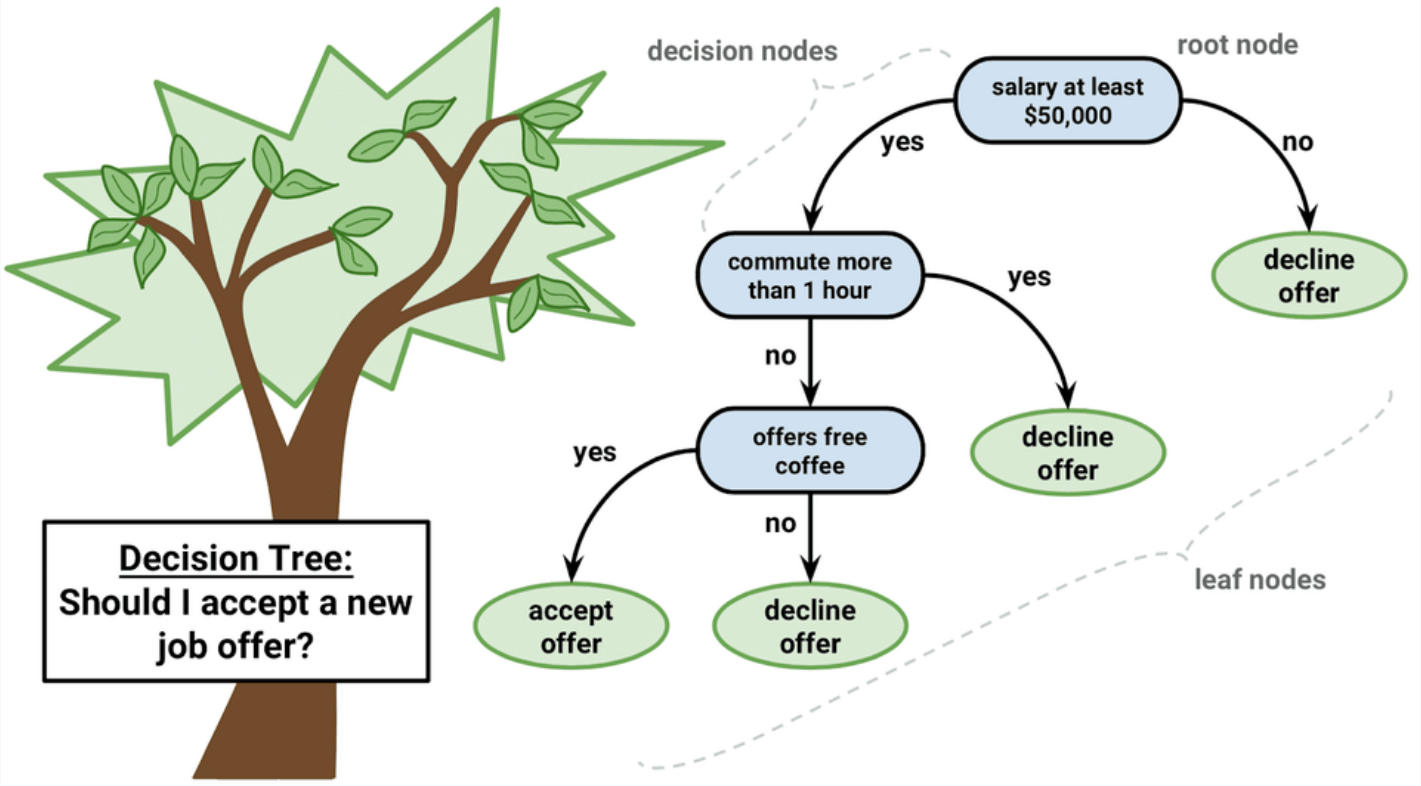
XGBoost:

XGBoost is a machine learning algorithm known for its speed and accuracy. It builds multiple decision trees sequentially, correcting errors of the previous ones, to predict outcomes. It's popular for its ability to handle large datasets and is widely used for classification and regression tasks.



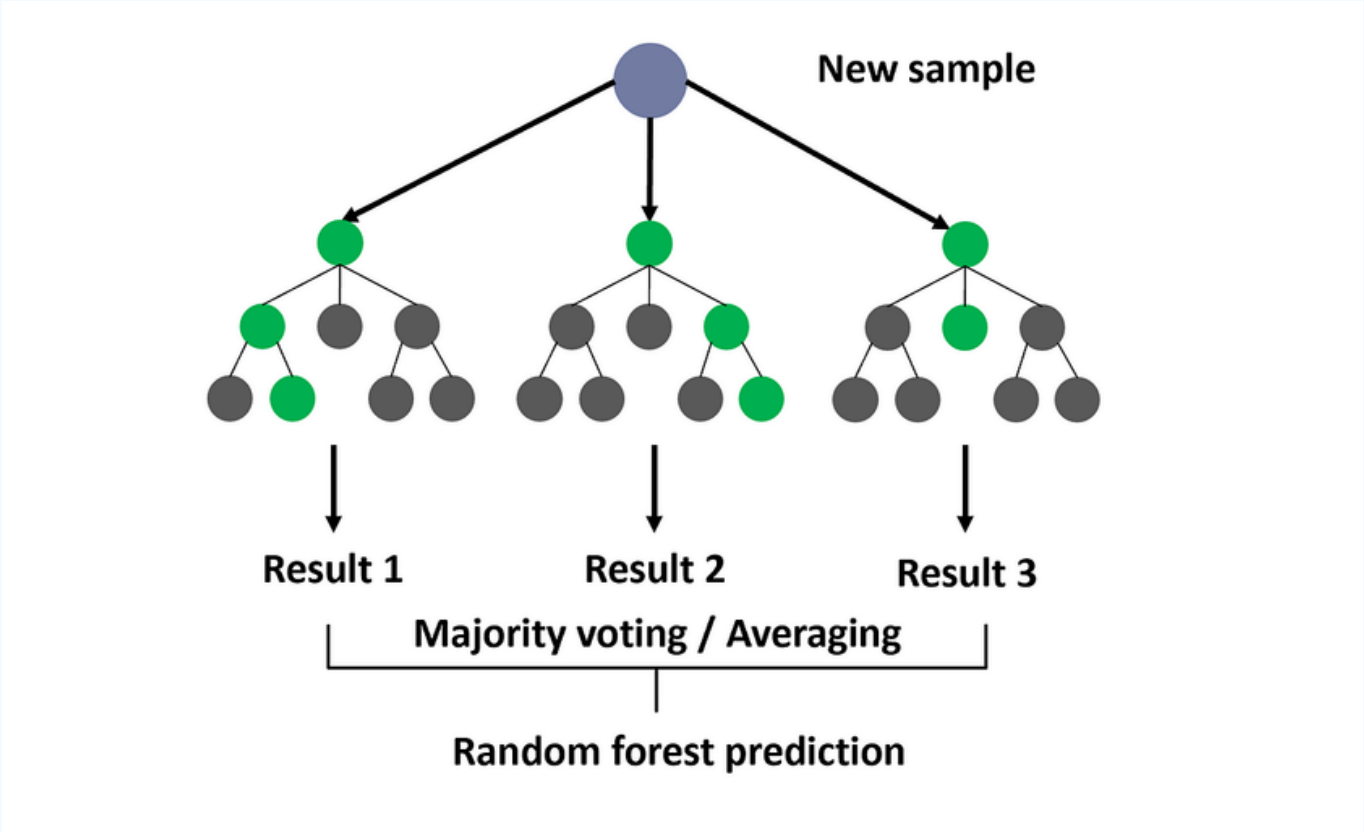
DescisionTreeClassifier:

DecisionTreeClassifier is a machine learning algorithm used for classification tasks. It creates a tree-like structure by splitting the dataset into subsets based on features, aiming to classify instances accurately. It's simple, interpretable, and capable of handling both numerical and categorical data. However, it may overfit if not properly tuned.



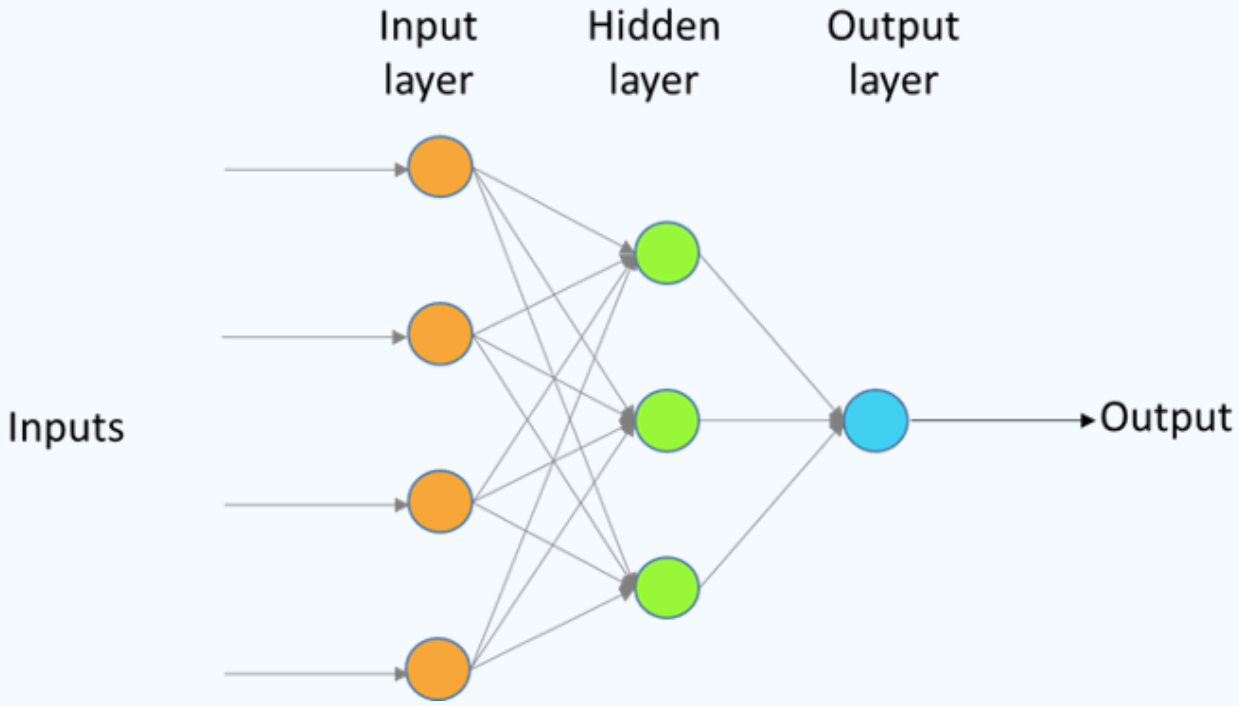
RandomForestClassifier:

Random Forest is an ensemble learning method for classification and regression tasks. It operates by constructing a multitude of decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random Forest introduces randomness both in the selection of the data samples used for training each tree and the features considered for each split, which helps to decorrelate the trees and reduce overfitting. It's known for its robustness, scalability, and ability to handle high-dimensional data. Random Forest is widely used in various machine learning applications due to its excellent performance and simplicity in implementation.



NN Model:

A NN model, or Neural Network, is a machine learning algorithm inspired by the human brain. It consists of interconnected nodes organized into layers, learning patterns from data to make predictions. It's versatile and widely used for tasks like classification and regression due to its flexibility and accuracy.
Our model is a simple sequential Neural Network model with three hidden layers of 32, 64, and 32 neurons respectively, all using ReLU activation. The output layer consists of one neuron with sigmoid activation, suitable for binary classification tasks.



CatBoostClassifier:

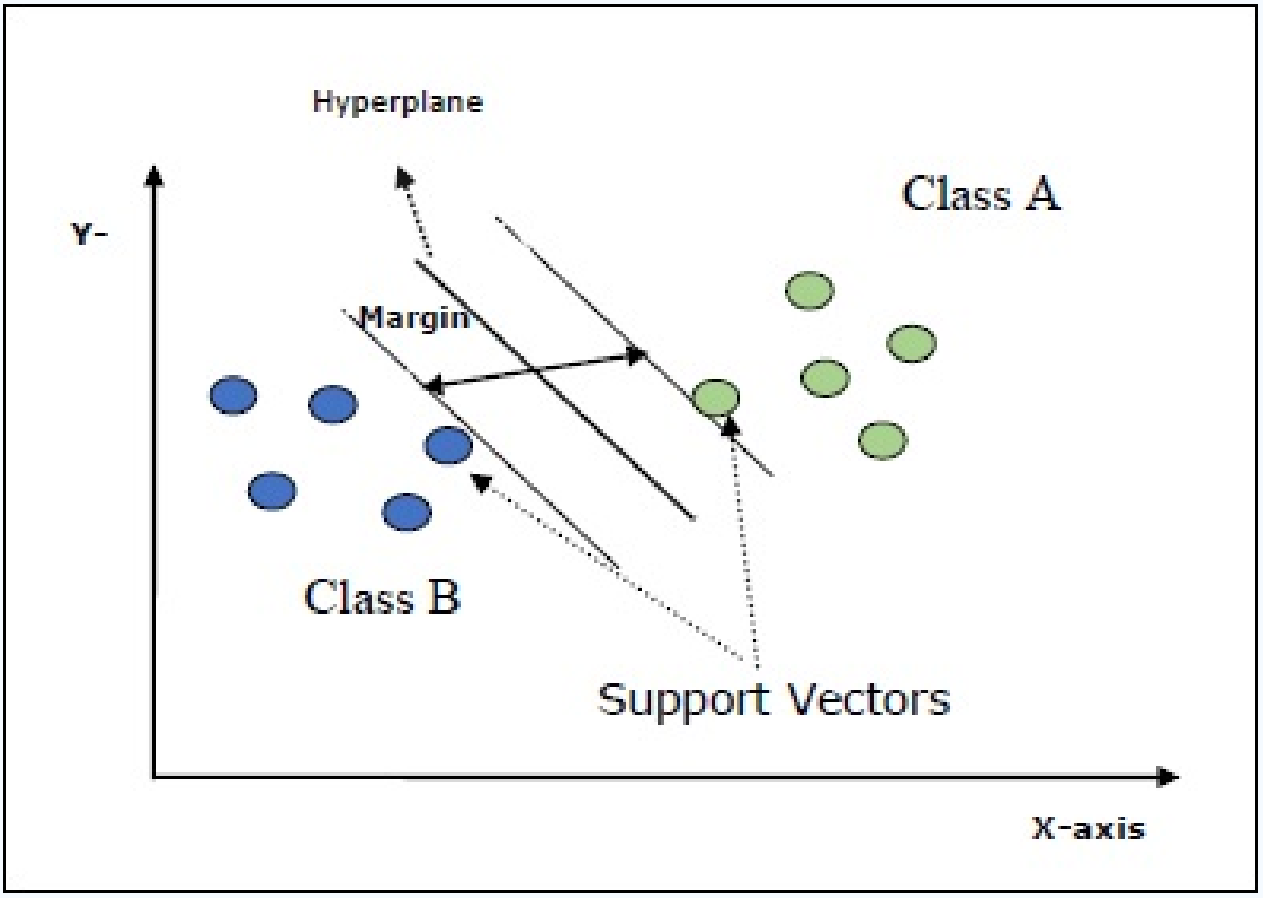
CatBoostClassifier is a machine learning algorithm specifically designed for gradient boosting on decision trees. It's particularly adept at handling categorical features, hence the name "CatBoost." It automatically handles categorical variables, eliminating the need for manual preprocessing. CatBoost implements an innovative algorithm for handling categorical data, which results in faster training times and better predictive performance. It's known for its robustness, efficiency, and ability to handle large datasets with high dimensionality.

LGBMClassifier:

LGBMClassifier, short for LightGBM Classifier, is a gradient boosting framework that uses tree-based learning algorithms. It's known for its efficiency and speed, particularly with large datasets. LightGBM employs a novel technique called Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to enhance training speed and reduce memory usage. It's highly scalable and performs well on a variety of machine learning tasks, including classification and ranking. Overall, LightGBM is favored for its high performance, flexibility, and ease of use.

SVC:

SVC, or Support Vector Classifier, is a supervised learning algorithm used for classification tasks. It works by finding the hyperplane that best separates different classes in the feature space. SVC aims to maximize the margin between classes, effectively maximizing the distance between the closest data points (support vectors) from each class to the decision boundary. It's effective in high-dimensional spaces and is versatile with different kernel functions such as linear, polynomial, and radial basis function (RBF). SVC is particularly useful for binary classification but can also be extended to handle multi-class classification problems.



Performance Measures:

Accuracy: The proportion of correctly classified instances over the total number of instances. It provides an overall measure of model correctness.

Precision: The proportion of true positive predictions over the total number of positive predictions. It indicates the accuracy of positive predictions made by the model.

Recall (Sensitivity): The proportion of true positive predictions over the total number of actual positive instances. It measures the model's ability to identify all relevant instances.

F1 Score: The harmonic mean of precision and recall, providing a balanced measure of both metrics. It is useful when the classes are imbalanced.

Confusion Matrix: A table that summarizes the model's performance by comparing predicted classes against actual classes. It includes metrics such as true positives, true negatives, false positives, and false negatives.

Experimentation and Results:

Cross-validation is used in machine learning to estimate a model's performance, prevent overfitting, and optimize hyperparameters by iteratively partitioning data into training and validation sets. It ensures robust evaluation and efficient use of available data.
Recall will be our primary performance measure because we want to maximize the accuracy of model which will predict to have positive CVD and actual value is also positive.

XGBoost

```
************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.93      0.94      0.94        89
         1.0       0.95      0.95      0.95       111

    accuracy                           0.94       200
   macro avg       0.94      0.94      0.94       200
weighted avg       0.95      0.94      0.95       200

Accuracy score of xgb   0.945
```

DecisionTreeClassifier

```
************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.93      0.94      0.94        89
         1.0       0.95      0.95      0.95       111

    accuracy                           0.94       200
   macro avg       0.94      0.94      0.94       200
weighted avg       0.95      0.94      0.95       200

Accuracy score of dec   0.945
```

RandomForest

```
************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.93      0.94      0.94        89
         1.0       0.95      0.95      0.95       111

    accuracy                           0.94       200
   macro avg       0.94      0.94      0.94       200
weighted avg       0.95      0.94      0.95       200

Accuracy score of rfc   0.945
```

NN Model

```
************** FOR fold No. 2 ***************
25/25 [==============================] - 0s 2ms/step - loss: 0.1184 - accuracy: 0.9563
7/7 [==============================] - 0s 2ms/step
              precision    recall  f1-score   support

         0.0       0.85      0.99      0.91        73
         1.0       0.99      0.90      0.94       127

    accuracy                           0.93       200
   macro avg       0.92      0.94      0.93       200
weighted avg       0.94      0.93      0.93       200

Accuracy score of NN Model is:  0.93
```

CatBoost

```
              precision    recall  f1-score   support

         0.0       0.97      0.98      0.97        90
         1.0       0.98      0.97      0.98       110

    accuracy                           0.97       200
   macro avg       0.97      0.98      0.97       200
weighted avg       0.98      0.97      0.98       200

Accuracy score of clf   0.975
************** FOR fold No. 4 ****************
```

LGBMClassifier

```
              precision    recall  f1-score   support

         0.0       0.98      0.98      0.98        90
         1.0       0.98      0.98      0.98       110

    accuracy                           0.98       200
   macro avg       0.98      0.98      0.98       200
weighted avg       0.98      0.98      0.98       200

Accuracy score of lgb_c   0.98
```

SVC Classifier

```
*************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.95      0.97      0.96        89
         1.0       0.97      0.95      0.96       111

    accuracy                           0.96       200
   macro avg       0.96      0.96      0.96       200
weighted avg       0.96      0.96      0.96       200

Accuracy score of svc_c   0.96
```

Logistic Regression

```
*************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.94      0.98      0.96        89
         1.0       0.98      0.95      0.96       111

    accuracy                           0.96       200
   macro avg       0.96      0.96      0.96       200
weighted avg       0.96      0.96      0.96       200

Accuracy score of log_r   0.96
```

KNN Model

```
************** FOR fold No. 1 ****************
              precision    recall  f1-score   support

         0.0       0.88      0.88      0.88        86
         1.0       0.91      0.91      0.91       114

    accuracy                           0.90       200
   macro avg       0.90      0.90      0.90       200
weighted avg       0.90      0.90      0.90       200

Accuracy score of KNN_Model   0.9
```

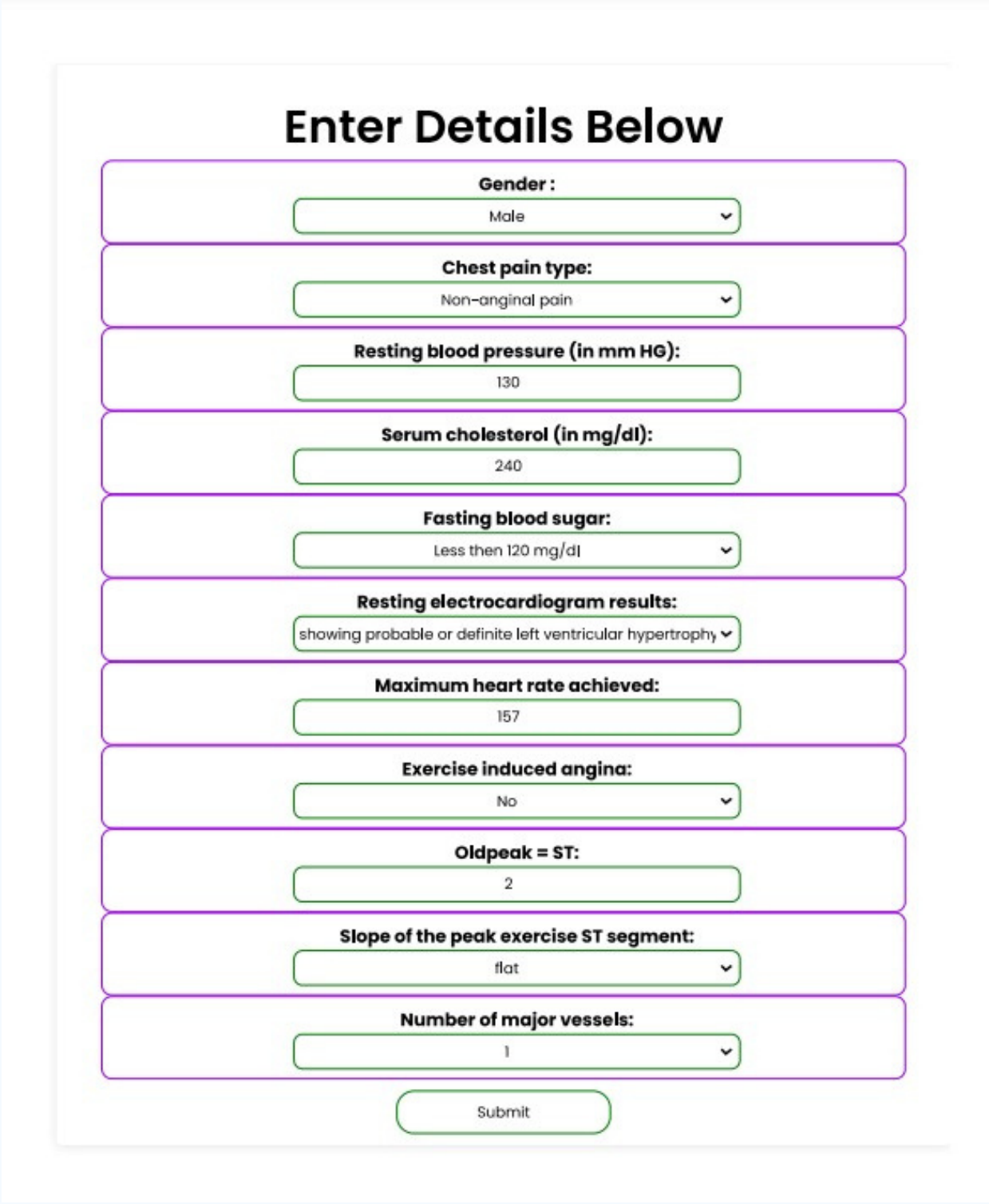
GaussianNB

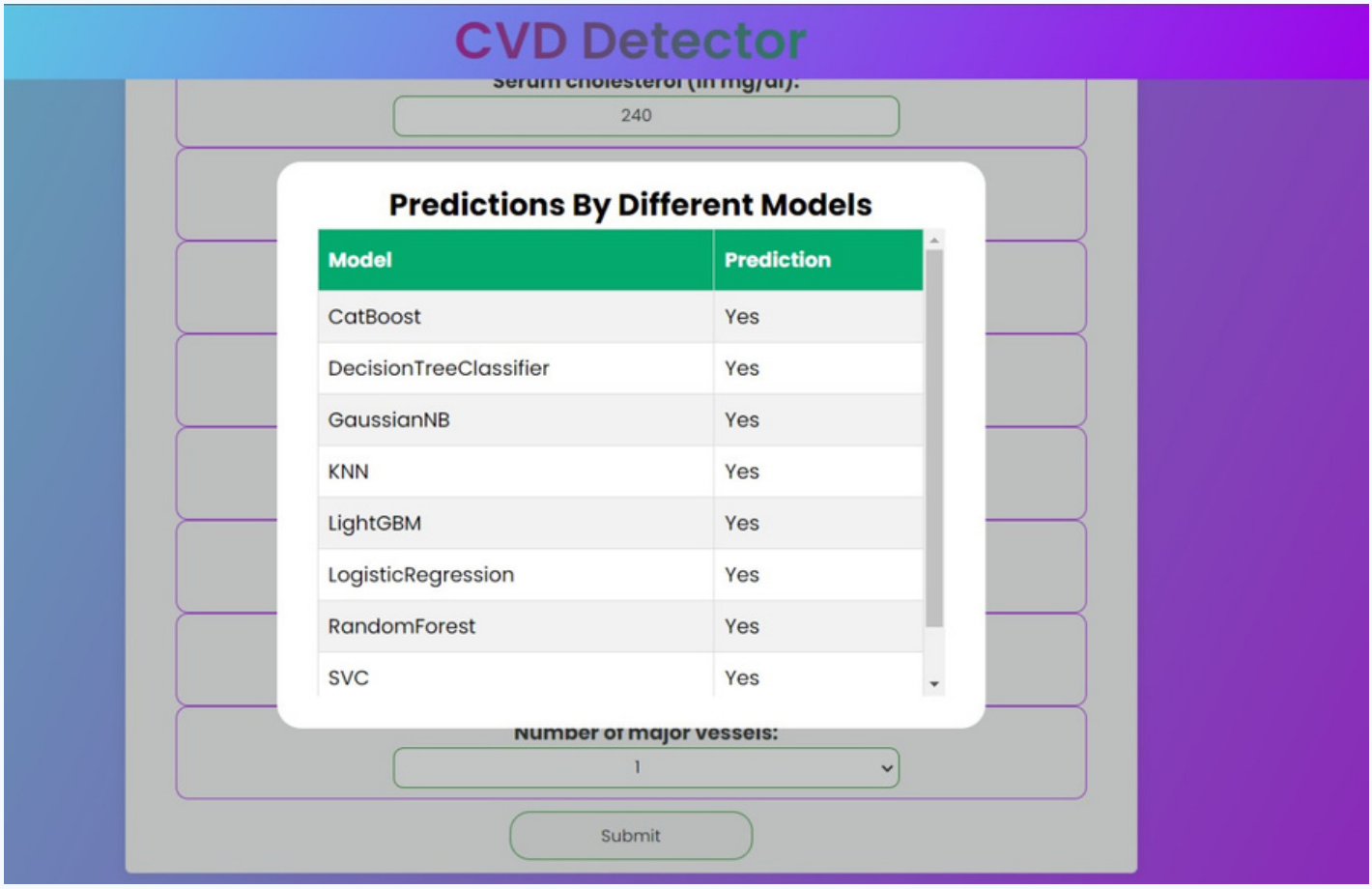
```
************** FOR fold No. 4 ****************
              precision    recall  f1-score   support

         0.0       0.91      0.96      0.93        89
         1.0       0.96      0.93      0.94       111

    accuracy                           0.94       200
   macro avg       0.94      0.94      0.94       200
weighted avg       0.94      0.94      0.94       200

Accuracy score of GNB_Model   0.94
```

Conclusion:

Based on above results, the best model for our problem is LGBMClassifier which has a recall of positive samples of 98%.

UI:




CVD Detector
240
Predictions By Different Models
Model
Prediction
CatBoost
Yes
DecisionTreeClassifier
Yes
GaussianNB
Yes
KNN
Yes
LightGBM
Yes
LogisticRegression
Yes
RandomForest
Yes
SVC
Yes
1
Submit

Thank You!!