# *Cardiovascular Disease Prediction - Machine Learning Final Project 2022 Fall*

Nafiz Mahamud
Eliza Wu

## I. Abstract

The objective of our model is to predict whether a person has heart disease or not based on several features of our dataset. To implement our model, we applied multiple classification algorithms such as KNN, Logistic Regression, Multiple Layer Perceptron, Decision Tree and Random Forest. Another important part of our model is we tuned every hyperparameter of each algorithm and pick the best hyperparameters and then compare the results of those algorithms.

## II. Introduction

A vital organ in the human body is the heart. It circulates blood and supplies to all the body's organs. Predicting the occurrence of heart illnesses is a difficult task in the medical industry. Machine Learning, Internet of Things and deep learning of these things have been proven to be successful solutions for healthcare challenges, biological community and clinical management. They also aid in the early detection of disease through effective medical data interpretation.

## III. Data

The Dataset used for this project was collected from kaggle[1]. It is a quite huge dataset of 70,000 rows and 13 columns and also a pretty much balanced dataset of positive ratio 0.50.[pic1].

The dataset contains three types of features:

**1. Objective:**

It is related to factual information of a patient like Age, Height, Weight,Gender.

**2.Examination:**

It is the medical reports of a patient like Systolic Blood Pressure(ap_hi); Diastolic Blood Pressure(ap_lo); Cholesterol(1-normal; 2-above normal; 3-well above normal) ; Glucose(gluc) (1-normal; 2-above normal; 3-well above normal).

**3.Subjective:**

It is a kind of personal information given by the patient such as Smoking, Alcohol intake, and Physical Activity. We analyzed our dataset such as work on the missing values, checking the coefficient[pic] between our independent column and dependent columns. We also applied feature importance for the feature selection and based on the results of both the feature importance[pic] and coefficient, we selected the most relevant features for our models.

We split our dataset into 80% for training and 20% for testing. We check the outliers and remove the outliers applying the Z -Score.

## IV. Models

Since the label of our dataset is categorical, we implemented several classification models to build our project, and the following sections will describe each model in detail. We implemented an embedded system to improve our accuracy.

- **Random Forest Classification**

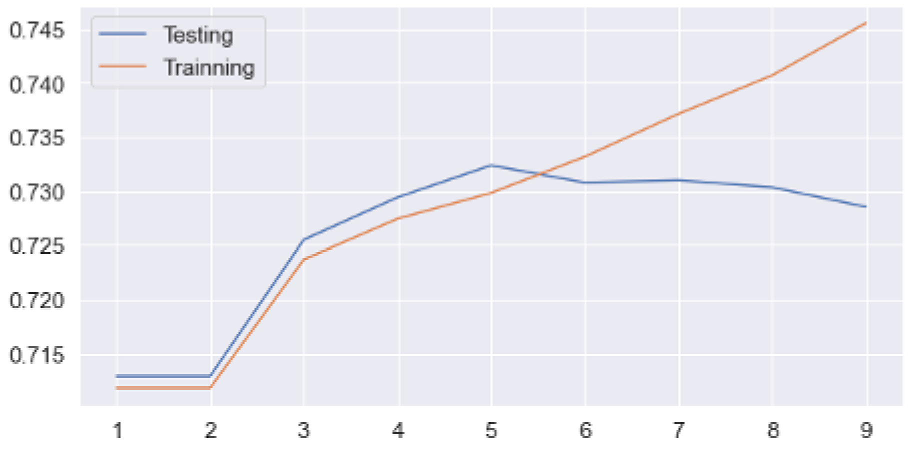
  A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. We tuned the hyperparameters of this algorithm using GridSearchCV to get the best hyperparameters so that the prediction will be more accurate.

  The parameters that we used:

- **Decision Tree Classification**

  Decision trees build classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed[2] . The core algorithm for building decision trees is called **ID3** by **J. R. Quinlan** which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree. To get the best accuracy we tuned the hyperparameters applying GridSearchCV.

  We found the best d is 5.



```
The best d is 5 and the highest accuracy 0.732377019488517
```

- **Logistic Regression**

  Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

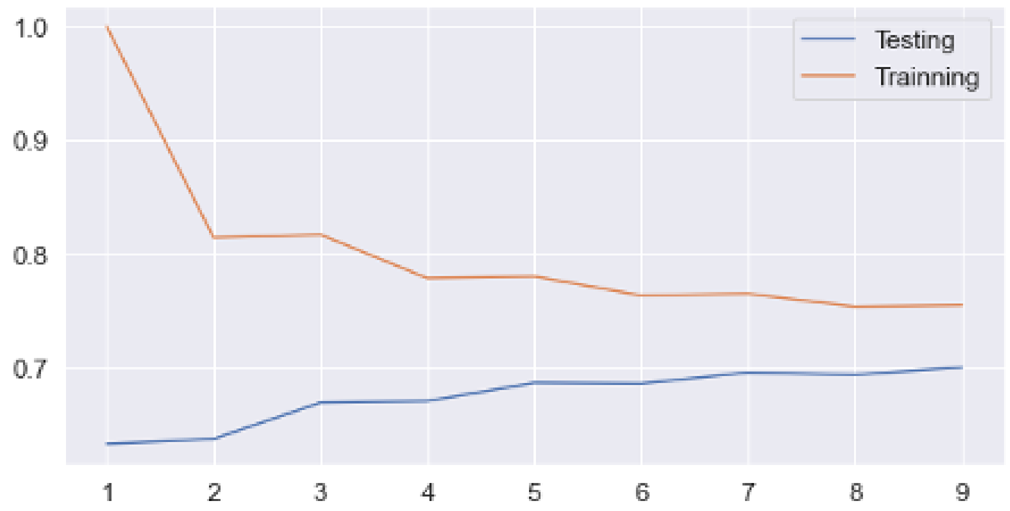
- **Multilayer Perceptron Classification**

  Multilayer Perceptron falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. For the hyperparameter tuning in the model, we applied three activation functions such as a)Logistic, b)Tanh and c)Rectified Linear Unit (Relu). And two solvers 1) Stochastic Gradient Descent(SGD) and 2) Adam with different numbers of hidden layers.

- **K-Nearest Neighbor Classification**

  The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

  We found the best k is 9.



```
The best k is 9 and the highest accuracy 0.7002825472723321
```

- **Deep Neural Network**

  We implement a simple sequential model of deep neural network with tensorflow and keras.

- **Ensemble Method**

  In our ensemble method, we combine Logistic Regression, K-Nearest Neighbor, Random Forest Classification, and Decision Tree.

- **Supports Vector Machine**

  The objective of the support vector machine algorithm is to find a hyperplane in an N dimensional space(N — the number of features) that distinctly classifies the data points.

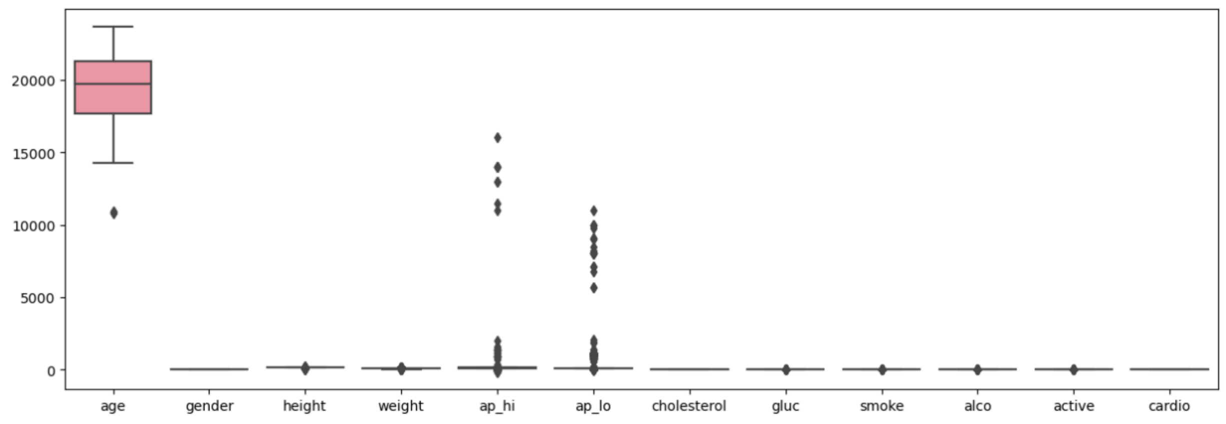


Fig 1: Outliers

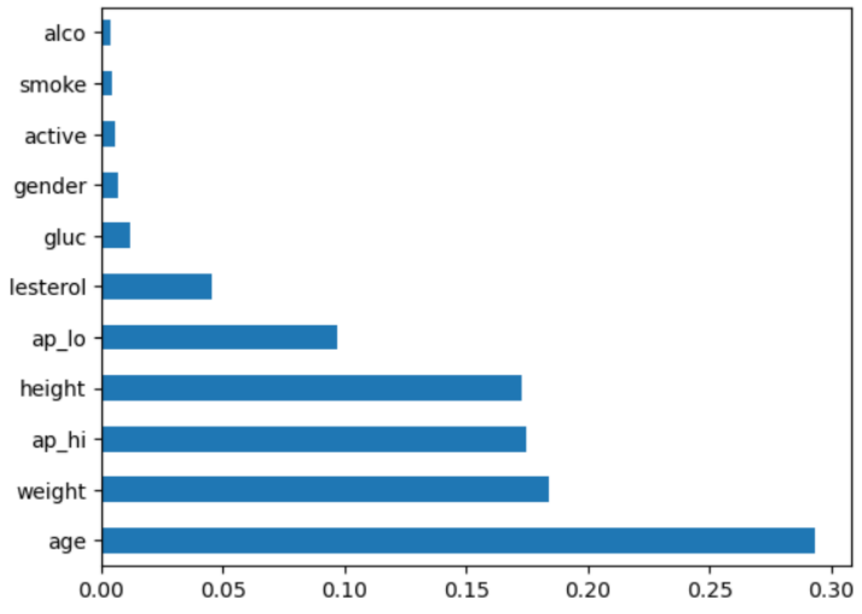
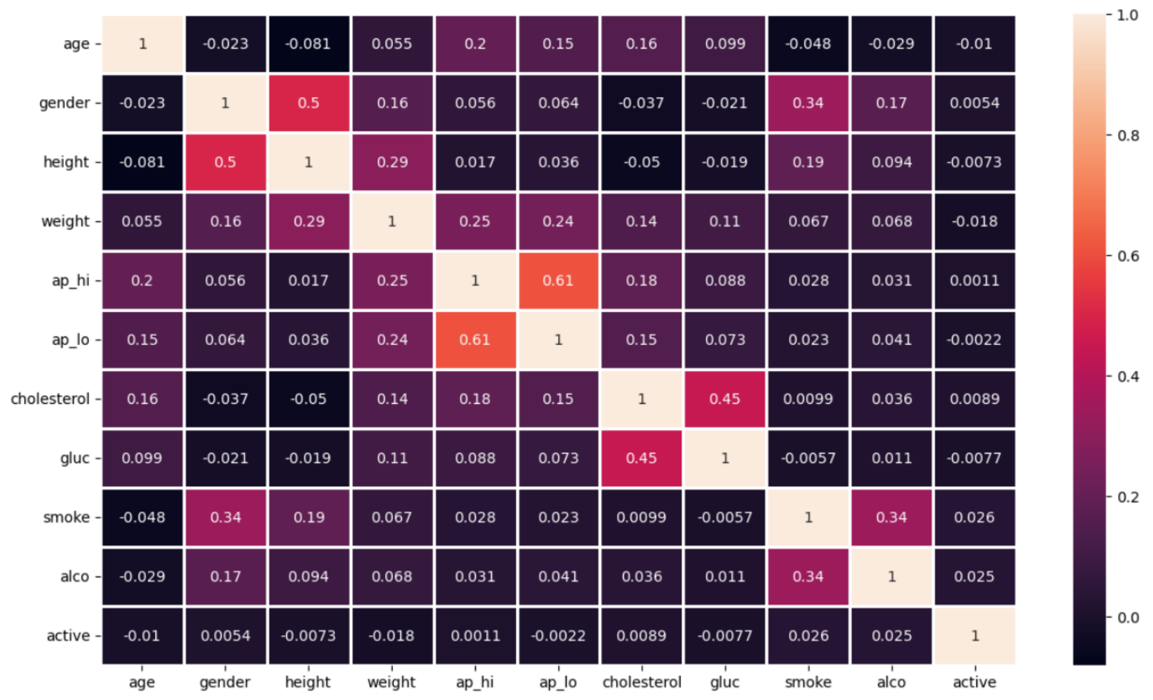


Fig 2: Feature Importance



| | age | gender | height | weight | ap_hi | ap_lo | cholesterol | gluc | smoke | alco | active |
|---|---|---|---|---|---|---|---|---|---|---|---|
| age | 1 | -0.023 | -0.081 | 0.055 | 0.2 | 0.15 | 0.16 | 0.099 | -0.048 | -0.029 | -0.01 |
| gender | -0.023 | 1 | 0.5 | 0.16 | 0.056 | 0.064 | -0.037 | -0.021 | 0.34 | 0.17 | 0.0054 |
| height | -0.081 | 0.5 | 1 | 0.29 | 0.017 | 0.036 | -0.05 | -0.019 | 0.19 | 0.094 | -0.0073 |
| weight | 0.055 | 0.16 | 0.29 | 1 | 0.25 | 0.24 | 0.14 | 0.11 | 0.067 | 0.068 | -0.018 |
| ap_hi | 0.2 | 0.056 | 0.017 | 0.25 | 1 | 0.61 | 0.18 | 0.088 | 0.028 | 0.031 | 0.0011 |
| ap_lo | 0.15 | 0.064 | 0.036 | 0.24 | 0.61 | 1 | 0.15 | 0.073 | 0.023 | 0.041 | -0.0022 |
| cholesterol | 0.16 | -0.037 | -0.05 | 0.14 | 0.18 | 0.15 | 1 | 0.45 | 0.0099 | 0.036 | 0.0089 |
| gluc | 0.099 | -0.021 | -0.019 | 0.11 | 0.088 | 0.073 | 0.45 | 1 | -0.0057 | 0.011 | -0.0077 |
| smoke | -0.048 | 0.34 | 0.19 | 0.067 | 0.028 | 0.023 | 0.0099 | -0.0057 | 1 | 0.34 | 0.026 |
| alco | -0.029 | 0.17 | 0.094 | 0.068 | 0.031 | 0.041 | 0.036 | 0.011 | 0.34 | 1 | 0.025 |
| active | -0.01 | 0.0054 | -0.0073 | -0.018 | 0.0011 | -0.0022 | 0.0089 | -0.0077 | 0.026 | 0.025 | 1 |

Fig 3: Correlation Matrix

## V. Implementation

All the models are implemented in Python 3.0 language and using several libraries like pandas, numpy, tensorflow and many more.

Experiments were run on local machines (Hp Pavilion,core i7).

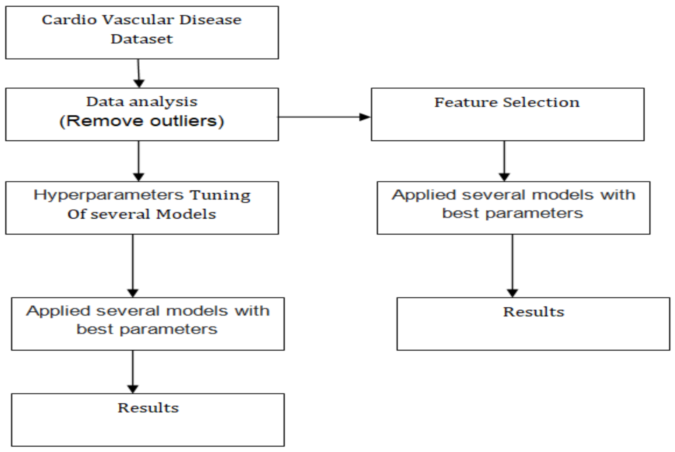

Cardio Vascular Disease Dataset
Data analysis (Remove outliers)
Feature Selection
Hyperparameters Tuning Of several Models
Applied several models with best parameters
Applied several models with best parameters
Results
Results

Fig 4: Project Architecture

Table 1: Hyperparameters Tuning

| Model | Hyperparameters Tuning (GridSearchCV) | Best Parameters |
|---|---|---|
| Random Forest Classification | Estimators: range (1 to 10)<br>Max Depth: range(1 to 4)<br>Min Samples Split: 2, 5, 8<br>Min Samples Leaf: 1, 2, 4 | Estimators: 7<br>Max Depth: 3<br>Min Samples Split: 2<br>Min Samples Leaf: 1 |
| Decision Tree Classification | Max Depth: range(1 to 10) | Max Depth: 5 |
| Logistic Regression | Penalty: L1 and L2<br>C: np.logspace(-4,4,20)<br>Solver: lbfgs and liblinear<br>Max Iter: [50, 100] | Penalty: L1<br>C: 3792.690190732246<br>Solver: liblinear<br>Max Iter: 50 |
| Multilayer Perceptron Classification | Activation: logistic, tanh, and relu<br>Hidden layer Sizes:(20,30), (25,), and (10,10)<br>Solver': adam and sgd<br>Learning rate init: [0.01, 0.1] | Activation: relu<br>Hidden layer Sizes:(20,30)<br>Solver': adam<br>Learning rate init: 0.01 |
| KNN Classification | Neighbors(k): range (1 to 10) | k: 9 |

Table 2: Model Performance on the Test Set before Feature Selection

| Model | Disease [ 0->No; 1->Yes] | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| Random Forest Classification | 0 | 0.71 | 0.78 | 0.74 | 0.72 |
| | 1 | 0.75 | 0.67 | 0.71 | |
| Decision Tree Classification | 0 | 0.71 | **0.80** | 0.75 | 0.73 |
| | 1 | 0.76 | **0.66** | 0.71 | |
| Multy-Layer Perceptron Classification | 0 | 0.73 | 0.76 | 0.75 | 0.74 |
| | 1 | 0.75 | 0.71 | 0.73 | |
| KNN Classification | 0 | 0.69 | 0.73 | 0.71 | 0.70 |
| | 1 | 0.71 | 0.67 | 0.69 | |
| Deep Neural Network | 0 | 0.66 | 0.66 | 0.66 | 0.65 |
| | 1 | 0.65 | 0.65 | 0.65 | |
| Support Vector Machine | 0 | 0.71 | 0.79 | 0.75 | 0.73 |
| | 1 | 0.76 | 0.68 | 0.71 | |
| Logistic Regression | 0 | 0.71 | 0.78 | 0.75 | 0.73 |
| | 1 | 0.75 | 0.68 | 0.71 | |

Table 3: Model Performance on the Test Set after Feature Selection

| Model | Disease [ 0->No; 1->Yes] | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| Random Forest Classification | 0 | 0.71 | 0.77 | 0.74 | 0.72 |
| | 1 | 0.74 | 0.68 | 0.71 | |
| Decision Tree Classification | 0 | 0.71 | **0.80** | 0.75 | 0.73 |
| | 1 | 0.76 | **0.67** | 0.71 | |
| Multy-Layer Perceptron Classification | 0 | 0.70 | **0.84** | 0.76 | 0.73 |
| | 1 | 0.79 | **0.62** | 0.70 | |
| KNN Classification | 0 | 0.70 | 0.73 | 0.71 | 0.71 |
| | 1 | 0.71 | 0.68 | 0.70 | |
| Deep Neural Network | 0 | 0.66 | 0.66 | 0.66 | 0.65 |
| | 1 | 0.65 | 0.65 | 0.65 | |
| Embedded Method | 0 | 0.71 | 0.79 | 0.75 | 0.73 |
| | 1 | 0.76 | 0.68 | 0.72 | |
| Support Vector Machine | 0 | 0.71 | 0.79 | 0.75 | 0.73 |
| | 1 | 0.76 | 0.68 | 0.71 | |
| Logistic Regression | 0 | 0.71 | 0.78 | 0.75 | 0.73 |
| | 1 | 0.75 | 0.67 | 0.71 | |

## VI. Results and Discussion

The performance of each model employed in this project before feature selection is summarized in Table 2.

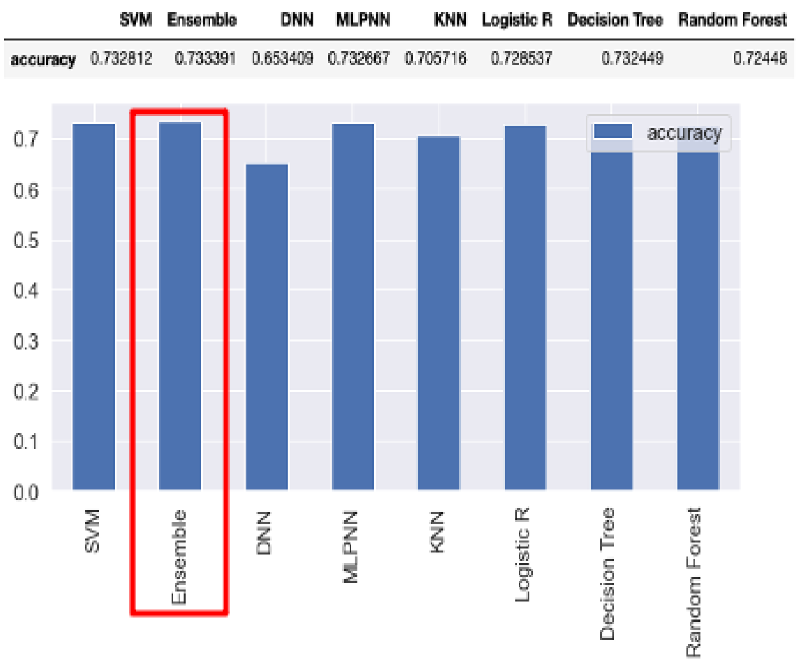
And, the performance of each model employed in this project after feature selection is summarized in Table 3.

The overall accuracy is not so good. Because in the dataset, the features were not well organized meaning the value of some features was not meaningful for a model. And also the features numbers were not many.

We saw that there were not much differences in precision,  recall, and f1-score among all the models except deep neural networks. The accuracy of the simple deep neural network was not much higher than other models.

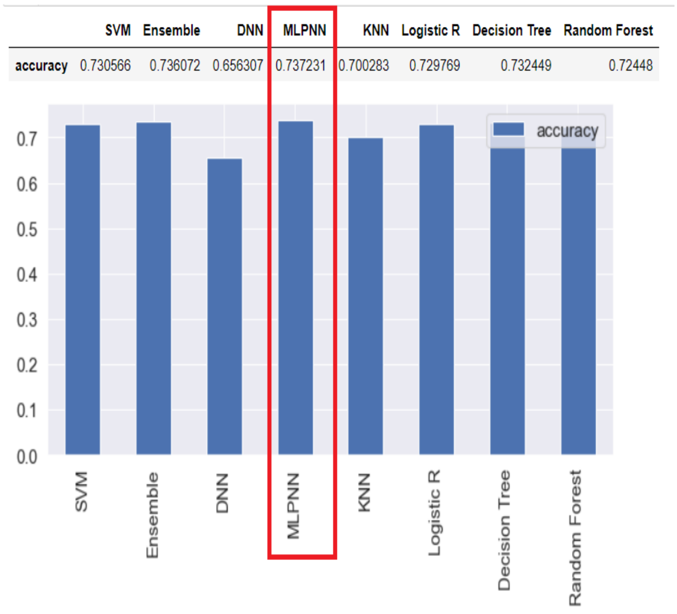
- Comparison

  As a result, the accuracy of the multi-layer perceptron classification method is the highest in our experiments before feature selection, which is 73.72%[Fig:6].
  On the other hand, the accuracy of the Ensemble method is the highest in our experiments before feature selection, which is 73.33%. The compare graph on the right shows that all the accuracy is very close to each other in both cases.



| | SVM | Ensemble | DNN | MLPNN | KNN | Logistic R | Decision Tree | Random Forest |
|---|---|---|---|---|---|---|---|---|
| accuracy | 0.732812 | 0.733391 | 0.653409 | 0.732667 | 0.705716 | 0.728537 | 0.732449 | 0.72448 |

Fig 5:Model Comparison after feature selection



| | SVM | Ensemble | DNN | MLPNN | KNN | Logistic R | Decision Tree | Random Forest |
|---|---|---|---|---|---|---|---|---|
| accuracy | 0.730566 | 0.736072 | 0.656307 | 0.737231 | 0.700283 | 0.729769 | 0.732449 | 0.72448 |

Fig 6:Model Comparison before feature selection

## VII. **Future Work**

First, we need to improve the training data preparation. This time we used training data features used the number to categorize the level, such as the physical activity. The feature categorize way may effect the result of the prediction and training process. In this project, we used the simple algorithm that we learned from the course. In the future, using different and complicated algorithms for higher accuracy is the target that we can investigate.

VIII.

## References

[1]. cardiovascular-disease-dataset, *https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset*

[2]. Decision tree classification, *https://www.saedsayad.com/decision_tree.htm*