PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE

DMW MINI-PROJECT
REPORT
ON
“**HEART DISEASE PREDICTION**”

SUBMITTED BY
Rajwinder Singh (41152)
Sahil Singh (41155)
Sanchit Raina (41157)

Under the guidance of
Prof. Shital Nayan Girme

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2021-22

LP - II Mini Project Report

Data Mining and Warehousing

Title: Heart Disease Prediction

Problem Statement: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class
labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix, and compare these models. Also, apply cross-validation while preparing the training and testing datasets. For Example Health Care Domain for predicting disease.

Abstract: The aim of this project is to demonstrate the concepts learned in Data Mining and Warehousing and use them towards the solution of real-world problems. The Heart Disease UCI Dataset is a classification dataset that has 13 predictor variables, and the label signifies whether a patient has heart disease or not. We will clean, transform and format the data appropriately and compare various classifiers in order to figure out which works best for this application.

Introduction: Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, heart rhythm problems (arrhythmias), and heart defects you're born with (congenital heart defects), among others.

The term "heart disease" is often used interchangeably with the term "cardiovascular disease". Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina), or stroke. Other heart conditions, such as those that affect your heart's muscle, valves, or rhythm, also are considered forms of heart disease.

Software and Hardware Requirements:

- Python
- Python libraries
- Jupyter Notebook

- 64-bit OS (Windows 10/ Ubuntu)

Data Preparation:

Pandas:

- It is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.
- It is free software.
- Provides a DataFrame object, which is a two-dimensional data structure in a tabular fashion.

Data Cleaning:

Data cleaning is an essential part of data science. Working with impure data can lead to many
difficulties. Techniques used:

- Remove Irrelevant Values
- Get Rid of Duplicate Values
- Avoid Typos (and similar errors)
- Convert data types
- Input Missing Values

Data Manipulation:

Data Manipulation techniques can be used in order to arrange data in a more beneficial manner.
Techniques used:

- Standardizing
- Normalizing
- Discretization
- Encoding Categorical features

Classification:

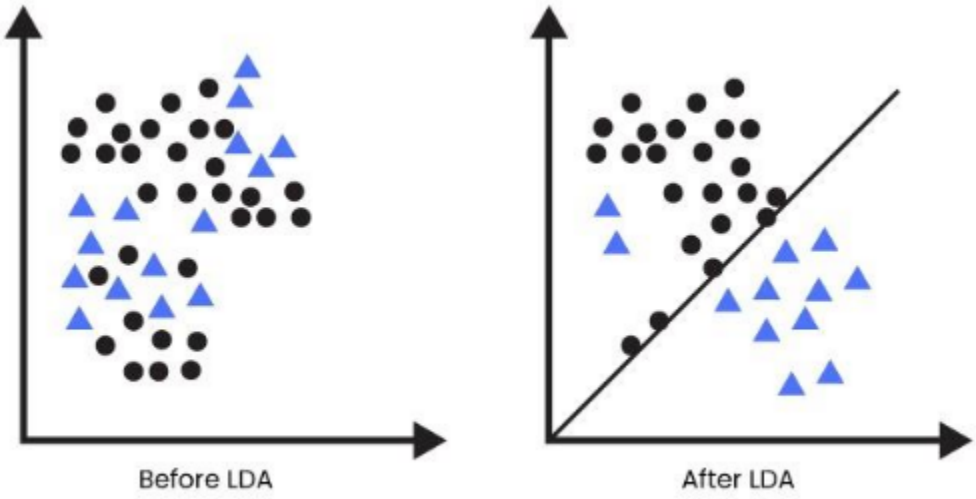
- In statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

- Examples are assigning a given email to the "spam" or "non-spam" class and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.).
- It is considered an instance of supervised learning.
- Types of classification: ○ Binary Classification When we have to categorize given data into 2 distinct classes. ○ Multiclass Classification The number of classes is more than 2.
- There are various types of classifiers. Some of them are:
    - Linear Classifiers: Logistic Regression
    - Tree-Based Classifiers: Decision Tree Classifier
    - Support Vector Machines
    - Artificial Neural Networks
    - Bayesian Regression
    - Gaussian Naive Bayes Classifiers
    - Stochastic Gradient Descent (SGD) Classifier
    - Ensemble Methods: Random Forests, AdaBoost, Bagging Classifier, Voting Classifier,
    - ExtraTrees Classifier

Linear Discriminant Analysis
Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

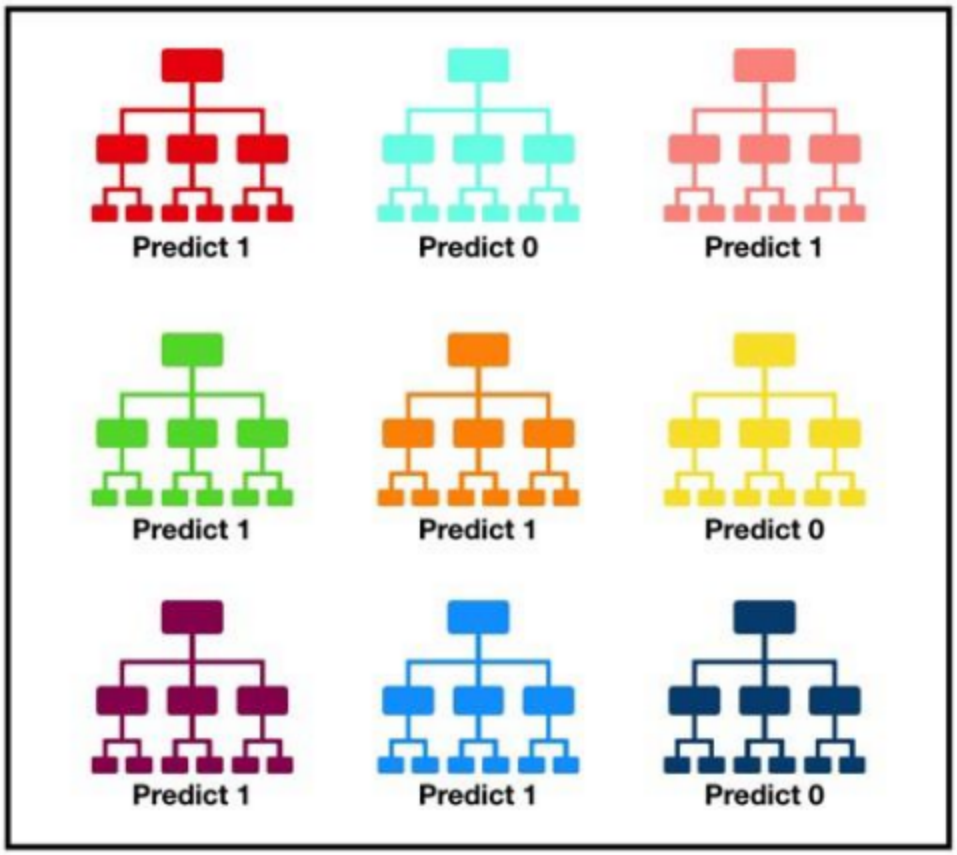
LDA is closely related to the analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable (i.e. the class label). Logistic regression and probit regression are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the

independent variables are normally distributed, which is a fundamental assumption of the LDA method.



Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (See figure below).



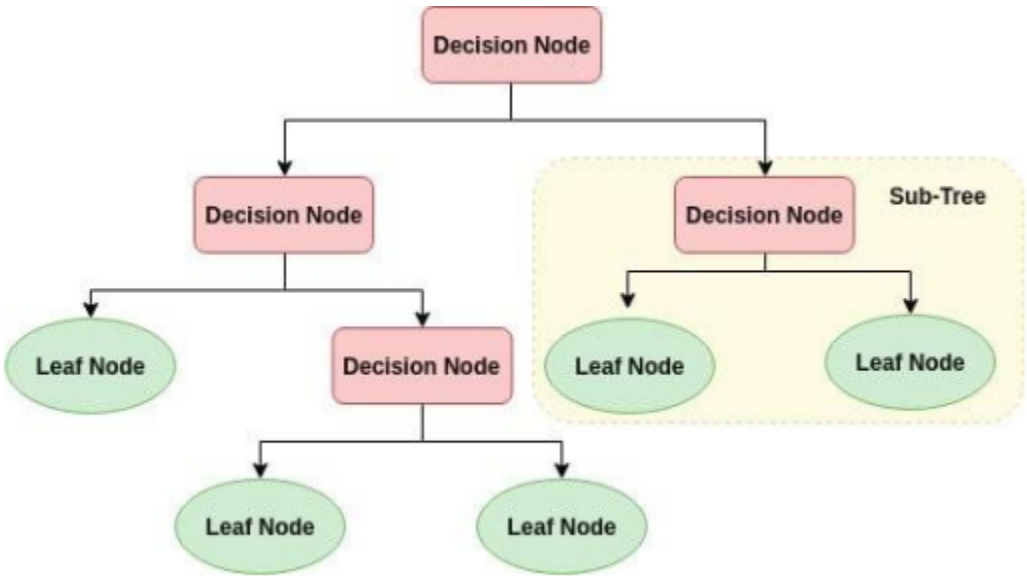
Tally: Six 1s and Three 0s
Prediction: 1

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. **The reason for this wonderful effect is that the trees protect each other from their individual errors** (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So, the prerequisites for the random forests to perform well are:

1. There needs to be some actual signal in our features so that models built using those features do better than random guessing.
2. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

Decision Tree Classifier

Decision tree learning is one of the predictive modeling approaches used in statistics, data mining, and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Decision trees are among the most popular machine learning algorithms, given their intelligibility and simplicity



Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model

in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be
interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient
boosting algorithms were subsequently developed by Jerome H. Friedman, simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. The latter two papers introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

Results:

	algo_name	acc_test	acc_train
2	RandomForestClassifier_	84.62	100.00
3	RandomForestClassifier_FS	83.52	95.28
1	LinearDiscriminantAnalysis_sfs	82.42	83.96
4	RandomForestClassifier_sfs	82.42	86.32
7	GradientBoostingClassifier_	81.32	100.00
0	LinearDiscriminantAnalysis_	80.22	85.38
6	DecisionTreeClassifier_sfs	78.02	86.79
8	GradientBoostingClassifier_sfs	78.02	86.79
5	DecisionTreeClassifier_	71.43	100.00

Conclusion:
We have successfully implemented multiple classifiers on the Heart Disease UCI dataset and have understood that the Random Forest Classifier produces the best results on it.