BREAST CANCER PREDICTION USING MACHINE LEARNING

Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Technology in the field of Computer Science and Engineering

BY

SWARNAVA DUTTA (380116010119)

Under the supervision
of
Dr.PRANATI RAKSHIT

Department of Computer Science and Engineering
JIS College of Engineering

Block-A, Phase-III, Kalyani, Nadia, Pin-741235
West Bengal, India
May, 2020

JIS College of Engineering
Block 'A', Phase-III, Kalyani, Nadia, 741235
Phone: +91 33 2582 2137, Telefax: +91 33 2582 2138
Website: www.jiscollege.ac.in, Email: info@jiscollege.ac.in

CERTIFICATE

This is to certify that **SWARNAVA DUTTA** has completed his/her project entitled **BREAST CANCER PREDICTION USING MACHINE LEARNING,** under the guidance of **Dr. Pranati Rakshit** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** from JIS college of Engineering (An Autonomous Institute) is an authentic record of their own work carried out during the academic year 2019-20 and to the best of our knowledge, this work has not been submitted elsewhere as part of the process of obtaining a degree, diploma, fellowship or any other similar title.

Signature of the Supervisor **Signature of the HOD** **Signature of the Principal**

Signature of the External Expert

Place:

Date:

ACKNOWLEDGEMENT

The analysis of the project work wishes to express our gratitude to Guide Name for allowing the degree attitude and providing effective guidance in development of this project work. His conscription of the topic and all the helpful hints, he provided, contributed greatly to successful development of this work, without being pedagogic and overbearing influence.

We also express our sincere gratitude to **Dr. Dharmpal Singh**, Head of the Department of Computer Science and Engineering of JIS College of Engineering and all the respected faculty members of Department of CSE for giving the scope of successfully carrying out the project work.

Finally, we take this opportunity to thank Prof. **(Dr.) Partha Sarkar**, Principal of JIS College of Engineering for giving us the scope of carrying out the project work.

Date:

....

SWARNAVA DUTTA
B.TECH in Computer Science and Engineering
4thYEAR/8th SEMESTER
Univ Roll--380116010119

List of Figures

Fig 1: Sigmoid Activation Function

Fig 2: Visualization of a Random Forest Model Making a Prediction
Fig 3: Possible hyperplane

Fig 4: Possible hyperplanes

Fig 5: hyperplane in 2D and 3D

Fig 6: support vectors
Fig 7: cumulative test statistic
Fig 8: fitness function
Fig 9: crossover point
Fig 10: offspring
Fig 11: population
Fig 12: mutation
Fig 13: supervised learning model
Fig 14: machine learning pipeline
Fig 15: heatmap
Fig 16: graph plot

Fig 17: accuracy of svm classifier (classification methods)
Fig 18: accuracy of logistic regression (classification methods)
Fig 19: accuracy of knn classifier (classification methods)
Fig 20: accuracy of naïve bayes (classification methods)
Fig 21: accuracy of decision tree algorithm (classification methods)
Fig 22: accuracy random forest classifier (classification methods)

Fig 23: accuracy of svm classifier (feature selection)
Fig 24: accuracy of logistic regression (feature selection)
Fig 25: accuracy of knn classifier (feature selection)
Fig 26: accuracy of naïve bayes (feature selection)
Fig 27: accuracy of decision tree algorithm (feature selection)
Fig 28: accuracy random forest classifier (feature selection)

List of Tables

Table 1. : This table is created with selecting all the features and getting the accuracy with normalization and svm classifier gives the best accuracy of 96.49% and lowest accuracy is given by logistic regression at about 45.61%.

Table 2.: This table is created with selecting all the features and getting the accuracy without normalization and highest accuracy is achieved by random forest classifier at 97.37% and lowest by KNN at 93.85%.

Table 3.: This table represents all the physical features that are being used.there are total of 30 attributes.

Table 4.: this table has the best 15 features acoording to their score to give the maximum accuracy possible for chi-squared test.

Table 5. Gives the accuracy results of chi-squared test with and without normalization and the acuuracy are 92.98% and 98.24% respectively gained by svc.

Table 6. 9 best features are taken into consideration for better accuracy and genetic algorithm is used.

Table 7.: Accuracy result of genetic algorithm using 9 best possible features.

CONTENTS

Title page 1-2

Certificate 3

Acknowledgement 4

List of Figures & Tables 5

Abstract

1. Introduction 9

1.1 Problem definition 9

1.2 Objective 10

1.3 Project overview/specifications 11

1.4 Hardware specifications 12

1.5 Software specification 12

2. Literature Survey 13

2.1 Existing systems 13

2.2 Feasibility study 17

3. Methodology and implementation 18

3.1 Dataset design 18

3.2 Methodology 20

3.2.1 logistic regression 20

3.2.2 random forest classifier 22

3.2.3 support vector machine classifier 24

3.2.4 k-nearest neighbor 27

3.2.5 decision tree 29

3.2.6 naïve bayes 32

3.2.7 Feature selection 35

3.2.7.1 chi-squared test 35

3.2.7.2 genetic algorithm 37

4. Implementation 40

5. Result and Discussion 41

5.1 heat map 45

5.2 graph plot 46

5.3 code snippet 47

5.3.1 Classification methods

5.3.2 Feature selection

6. Conclusion 53

7. Discussion 54

8. References 55

ABSTRACT:

Cancer has been characterized as a heterogeneous disease consisting of many different subtypes. The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients. The importance of classifying cancer patients into high or low risk groups has led many research teams, from the biomedical and the bioinformatics field, to study the application of machine learning (ML) methods.

Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice. In this work, we present a review of recent ML approaches employed in the modeling of cancer progression.

This present work may have been utilized as an aim to model the progression of protection and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance. A variety of these techniques including Naïve Bayes, logistic regression, KNN, Support Vector Machines (SVMs) and Decision Trees (DTs) and feature selection methods like chi-squared test and genetic algorithm test to improve the accuracy and optimization of the present classification models which will be widely applied in cancer research for the development of predictive models, resulting in effective and accurate decision making.

Keyword: breast cancer, malignant, benign, classifier, chi-squared, genetic algorithm, knn, naïve bayes, logistic regression, svc, random forest, decision tree

1. INTRODUCTION:

1.1 Problem Definition

Breast Cancer becomes dangerous disease in today's era. The most common type of this type of breast cancer is ductal carcinoma, which begins in the lining of the milk ducts. It is nothing but only thin tubes that carry milk from the lobules of the breast to the little nipple. Another type of breast cancer is lobular carcinoma, which begins in the lobules of the breast. Invasive breast cancer is breast cancer that has spread from where it began in the breast ducts or lobules to surrounding normal tissue. Breast cancer occurs in both men and women, although male breast cancer is rare.

According to the survey of in 2014, there are 232,670 females and 2,360 males having this type of new cases regarding the breast cancer. Among them 40,000 females and 430 males was death during the period this survey .These are the signs and symptoms for the early detection of the breast cancer. Machine Learning is a powerful tool and technique to handling this task. In data mining breast cancer research has been one of the important research topics in medical science during the recent years The classification of Breast Cancer data can be useful to predict the result of some diseases or discover the genetic behavior of tumors. There are many techniques to predict and classification breast cancer pattern. This paper empirically compares performance of different classification rules that are suitable for direct interpretability of their results.

Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice. In this work, we present a review of recent ML approaches employed in the modeling of cancer progression.

1.2 OBJECTIVE:

The objective of our project is to predict the occurrence of breast cancer with utmost accuracy possible using different classification techniques, feature selection methods, feature importance, hypothesis and heuristic test like chi squared test and genetic algorithm.

The dataset we are using is UCI dataset which includes different physical features that is required to determine whether the person has a probability of breast cancer and how much accurate the result is (whether it is benign or malignant).

1.3 Project Overview/Specifications

Cancer prediction is a machine learning application, trained by a UCI dataset. All the features will be trained and the classifier algorithm will calculate the probability of presence of disease. The result/accuracy will be displayed. Thus, minimizing the cost and time required to predict the disease. Format of data plays crucial part in this application. Our system will be implementing the following algorithms:

- Support Vector Machine (SVM)
- Logistic Regression
- Random Forest
- KNN
- Naïve bayes
- Decision tree

Furthermore, some steps will be taken for optimizing the algorithms thereby improving the accuracy. These steps include cleaning the dataset and data pre-processing. The algorithms were judged based on their accuracy and it is observed that the is the most accurate out of the three with efficiency. Hence, it is selected for the main application .The main application is a web application which accepts the various parameters from the user as input and computes the result. The result is displayed along with the accuracy of prediction .

1.4 Hardware Specification

- Operating System: Windows 7 or above.
- Dual core processor or more.
- RAM: 4GB or more.
- Hard Disk: 20GB

1.5 Software Specification

- CODELAB
- JUPYTER
- PANDAS LIBRARY
- PANDAS LIBRARY
- SKLEARN LIBRARY

2. LITERATURE SURVEY

2.1 Existing System:

The existing system has various problems which are mentioned below-

- System is only using one data set for validation which does not predictable enough to generate outcomes.
- System is only exploring the common predictable performance of their models without considering the F-score and precision as measures.
- Most studies do not provide statistical test results to demonstrate the level of significance of their experimental results

- Most studies related to ensemble classifier do not compare the performance difference between individual classifiers and an ensemble classifier consisted of individual classifiers.

M. Ravishankar et al. [1] used classification of detected abnormality as benign or malignant using Support vector machine (SVM) classifier. The proposed method was evaluated using Mini Mammographic Image Analysis Society (Mini-MIAS) dataset. The proposed method has achieved 92% accuracy.

Majid Iranpour , Sanaz Almassi , Morteza Analoui et al. [2] applied support vector machines (SVMs) and radial basis function (RBF) for breast cancer detection. Results demonstrate that SVM classifiers with the proposed automatic parameter tuning systems and the RBF classifier can be used as one of most efficient tools for breast cancer detection, with the detection accuracy up to 98%.

Vikas Chaurasia and Saurabh Pal, BB Tiwari et al. [3] compare the performance criterion of supervised learning classifiers; such as Naïve Bayes, SVM-RBF kernel, RBF neural networks, Decision trees (J48) and simple CART; to find the best classifier in breast cancer datasets. The results (based on average accuracy Breast Cancer dataset) indicated that the Naive Bayes is the best predictor with 97.36% accuracy on the holdout sample, RBF Network came out to be the second with 96.77% accuracy, J48 came out third with 93.41% accuracy.

Hiba Asria ,Hajar Mousannifb ,Hassan Al Moatassime ,Thomas Noeld et al. [4] compare the performance of C4.5, Naïve Bayes, Support Vector Machine (SVM) and K- Nearest Neighbor (K-NN), NB to find the best classifier in WBC dataset. Experimental results show that SVM gives the highest accuracy (97.13%).

Author BichenZheng,Sang WonYoon et al.[5] proposed, a support vector machine (SVM) is used to obtain the new classifier to differentiate the incoming tumors. Based on 10-fold cross validation, the proposed methodology improves the accuracy to 97.38%, when tested on the Wisconsin Diagnostic Breast Cancer (WDBC) data set from the University of California – Irvine machine learning repository.

Emina Aličković & Abdulhamit Subasi, et al. [6] authors used Wisconsin Breast Cancer datasets to evaluate the system proposed.The performance of the methods is evaluated using classification accuracy, area under receiver operating characteristic curves and *F*-measure. Results obtained with the Rotation Forest model with GA-based 14 features show the highest classification accuracy (99.48 %), and when compared with the previous works, the proposed approach reveals the enhancement in performances.

Authors Hussein AttyaLafta, Noor KdhimAyoob, Asraa Abdullah Hussein et al. [7] introduced an approach for diagnosing breast cancer via classifying a well-known WBCD dataset based on a hybrid neurogenetic system. The suggested approach showed a good behavior and excellent classification accuracy.The accuracy of system has been reached to 100% in the best trial and it is exceeding 97% in other experiments.

Authors Shokoufeh Aalaei, Hadi Shahraki, Alireza Rowhanimanesh, and Saeid Eslami et al. [8] proposed in their paper about the comparison of average accuracies for the three classifiers (ANN, PS-classifier, GA-classifier) with and without feature selection on WBC dataset showed that without feature selection the accuracy of ANN (96.8%) is the best and the accuracy obtained by PS-classifier is better than that produced by GA-classifier (96.2 vs. 96.08). It is observed that feature selection improved the accuracy of all classifiers expect of ANN and the best accuracy with feature selection achieved by PS-classifier (96.9%).

With respect to all related work mentioned above, our work compares the behavior of machine learning algorithms SVM, NB, k-NN, SDC, logistic regression and decision tree using Wisconsin Breast Cancer (original) datasets in both diagnosis and analysis to make decisions. The goal is to achieve the best accuracy with the lowest error rate in analyzing data.

2.2 FEASIBILITY STUDY

In feasibility study phase we had undergone through various steps which are described below:

1. **Technical Feasibility:**

   The project entitles “BREAST CANCER PREDICTION USING MACHINE LEARNING” is technically feasibility because it provides the high level of reliability, availability and compatibility.

2. **Economical Feasibility:**

   The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

3. METHODOLOGYAND IMPLEMENTATION:

3.1 DATASET DESIGN:

Attribute Description

radius	mean of distances from center to points on the perimeter
texture	standard deviation of gray-scale values
perimeter	Circumference of affected area
area	Affected area of disease
smoothness	local variation in radius lengths
compactness	perimeter^2 / area - 1.0
concavity	severity of concave portions of the contour
Concave points	number of concave portions of the contour
symmetry	Same on both sides or not
Fractal dimension	coastline approximation - 1

Data Set Characteristics:

Number of Instances: 569

Number of Attributes: 30 numeric, predictive attributes and the class

Cancer prediction is a machine learning application, trained by a UCI dataset. The user inputs its specific medical details to get the prediction of heart disease for that user. The algorithm will calculate the probability of presence of disease, minimizing the cost and time required to predict the disease.

Furthermore, some steps will be taken for optimizing the algorithms thereby improving the accuracy. These steps include cleaning the dataset and data pre-processing. The algorithms were judged based on their accuracy.

3.2 METHODOLOGY

3.2.1 LOGISTIC REGRESSION:

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Simple Logistic Regression

Model

Output = 0 or 1

Hypothesis => $Z = WX + B$

$h\Theta(x) = \text{sigmoid}(Z)$

Sigmoid Function

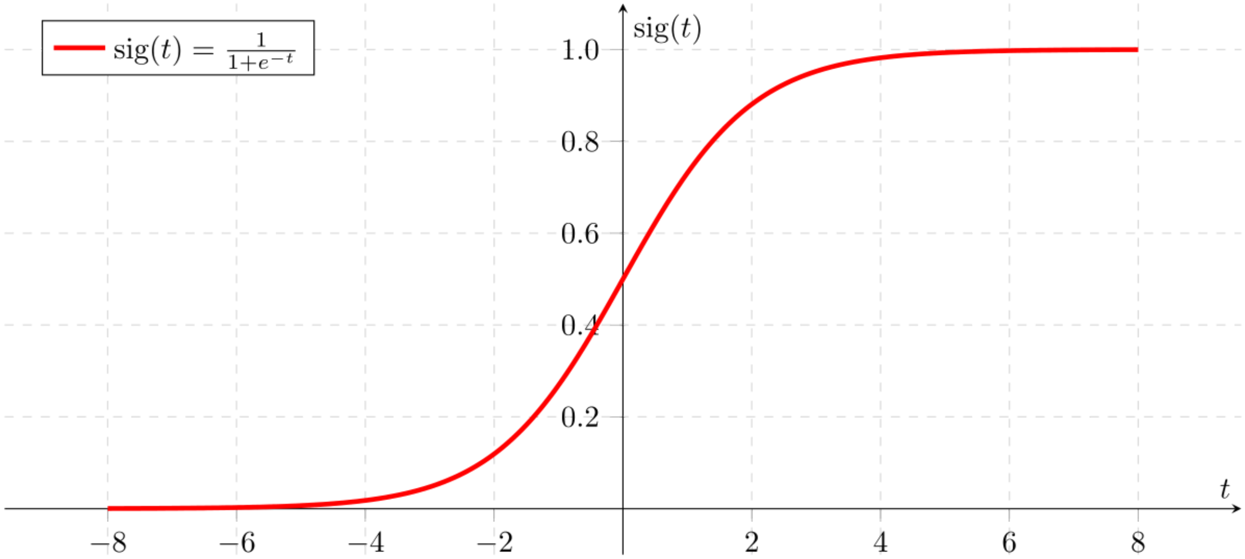


Figure 1: Sigmoid Activation Function

If ‘Z’ goes to infinity, Y(predicted) will become 1 and if ‘Z’ goes to negative infinity, Y(predicted) will become 0.

Analysis of the hypothesis

The output from the hypothesis is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input X. Consider the below example,

X = [x0 x1] = [1 IP-Address]

Based on the x1 value, let’s say we obtained the estimated probability to be 0.8. This tells that there is 80% chance that an email will be spam.

3.2.2 THE 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).

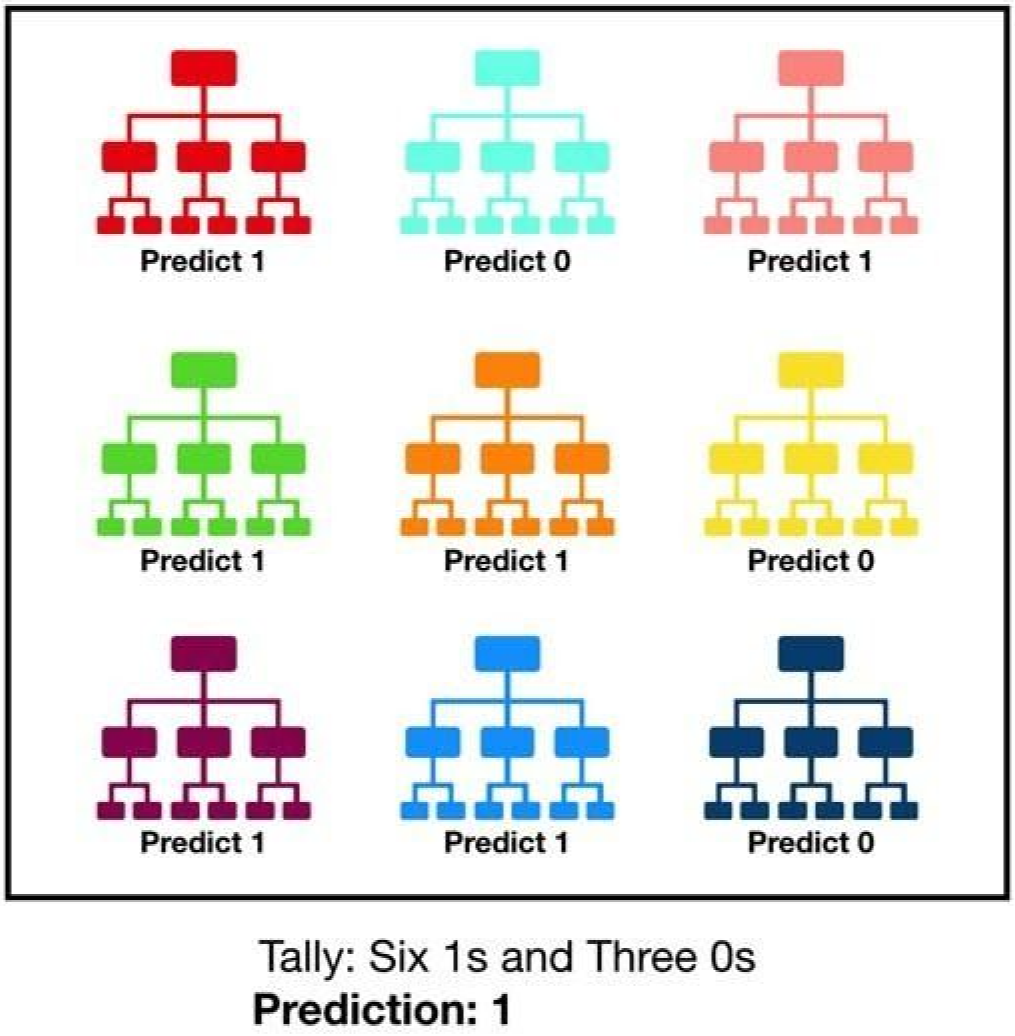


Fig 2: Visualization of a Random Forest Model Making a Prediction

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

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).

3.2.3 SUPPORT VECTOR MACHINE CLASSIFIER

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.

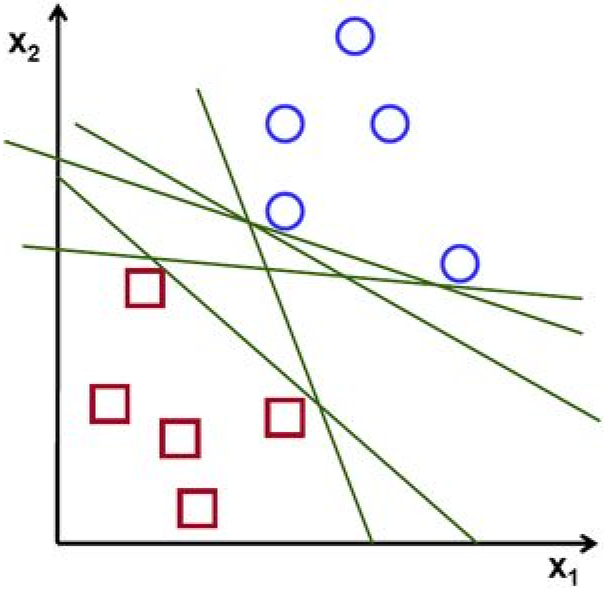


fig 3

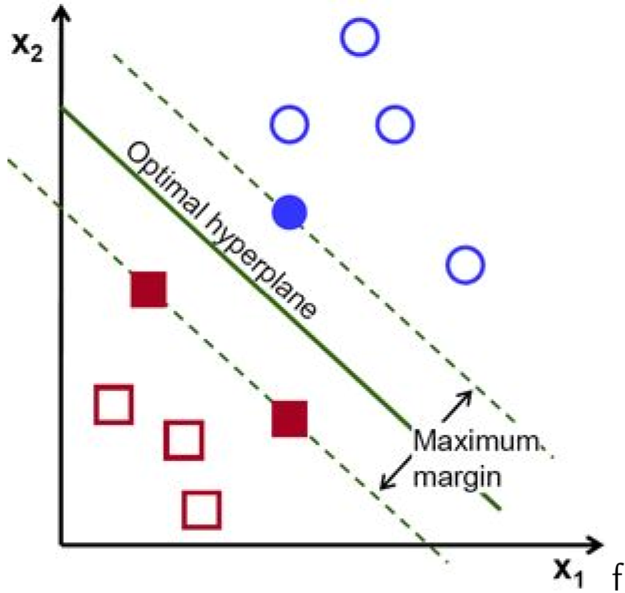
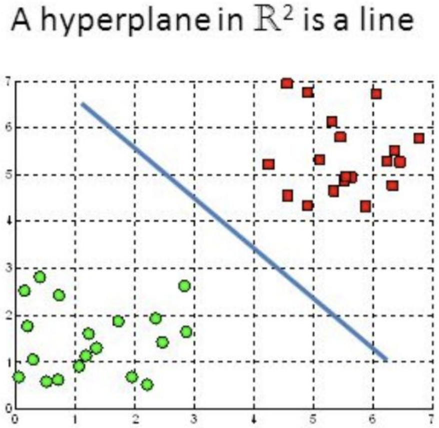


fig 4

Possible hyperplanes

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes and Support Vectors



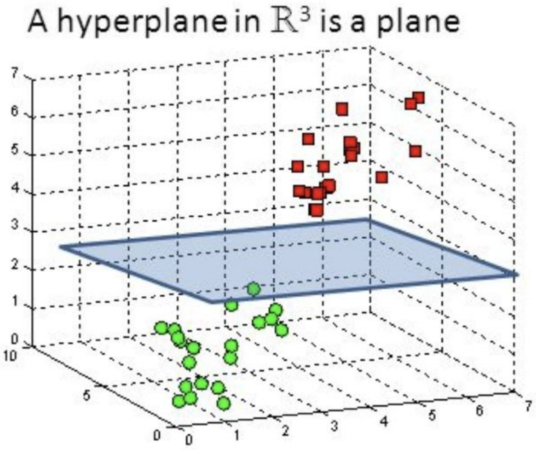


Fig 5: Hyperplanes in 2D and 3D feature space

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

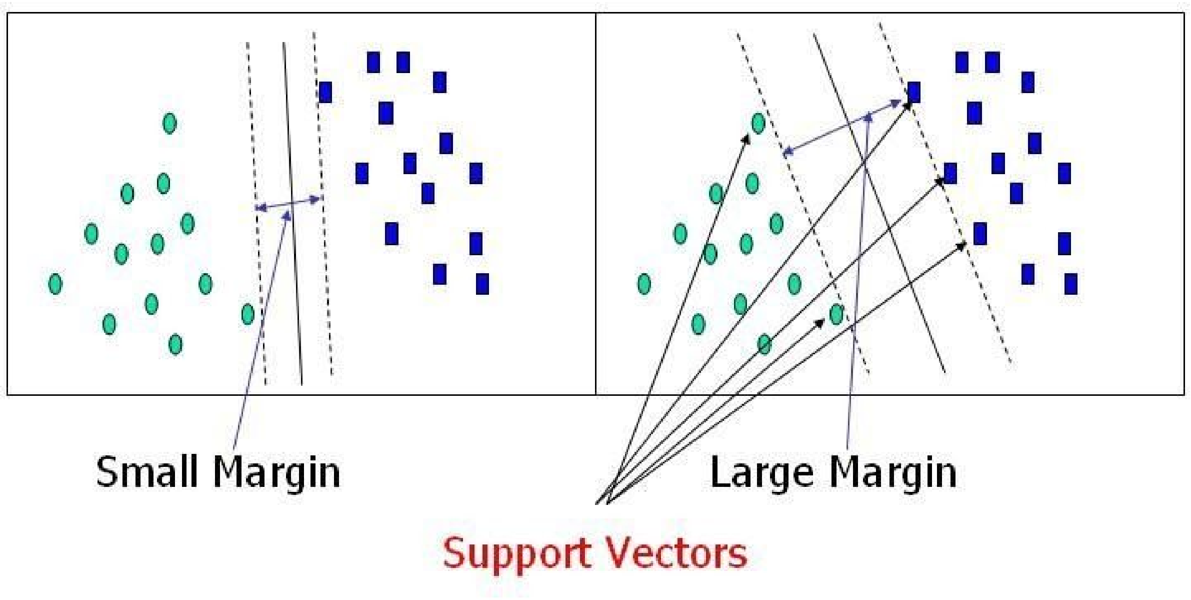


Fig 6: Support Vectors

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM

3.2.4 K-nearest neighbors (KNN)

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

- **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

Working of KNN Algorithm:

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- **3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- **3.2** − Now, based on the distance value, sort them in ascending order.
- **3.3** − Next, it will choose the top K rows from the sorted array.
- **3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

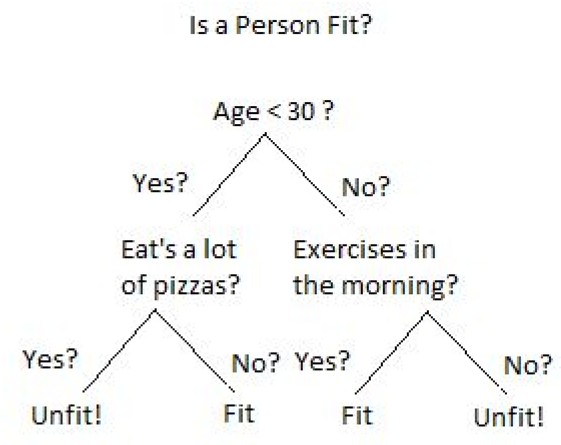
3.2.5 Decision tree:

Decision tree classifiers are utilized as a well known classification technique in different pattern recognition issues, for example, image classification and character recognition. Decision tree classifiers perform more successfully, specifically for complex classification problems, due to their high adaptability and computationally effective features.

Decision Tree consists of :

1. **Nodes** : Test for the value of a certain attribute.
2. **Edges/ Branch** : Correspond to the outcome of a test and connect to the next node or leaf.

3. **Leaf nodes** : Terminal nodes that predict the outcome (represent class labels or class distribution).



Source: xoriant.com

To understand the concept of Decision Tree consider the above example. Let’s say you want to predict whether a person is fit or unfit, given their information like age, eating habits, physical activity, etc. The decision nodes are the questions like ‘What’s the age?’, ‘Does he exercise?’, ‘Does he eat a lot of pizzas’? And the leaves represent outcomes like either ‘fit’, or ‘unfit’.

There are two main types of Decision Trees:

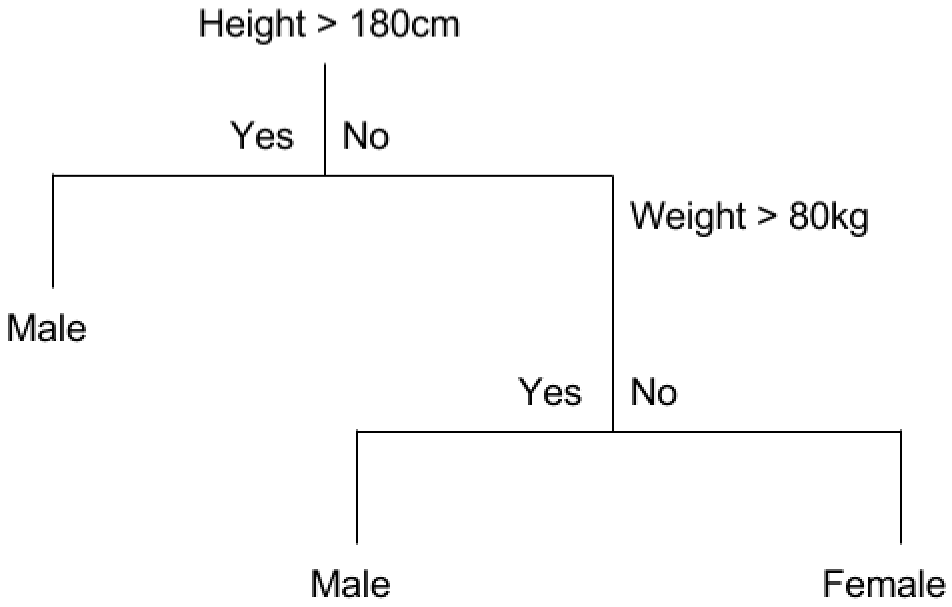
1. Classification Trees.

2. Regression Trees.

1. Classification trees (Yes/No types) :

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is **Categorical/ discrete**.

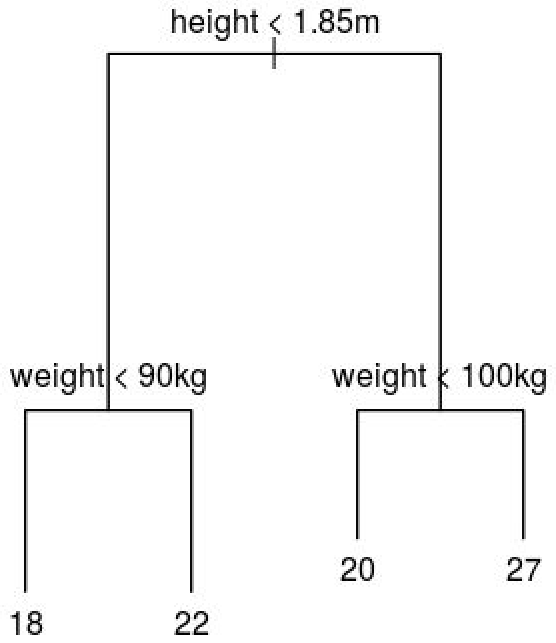
Such a tree is built through a process known as **binary recursive partitioning**. This is an iterative process of **splitting the data into partitions**, and then splitting it up further on each of the branches.

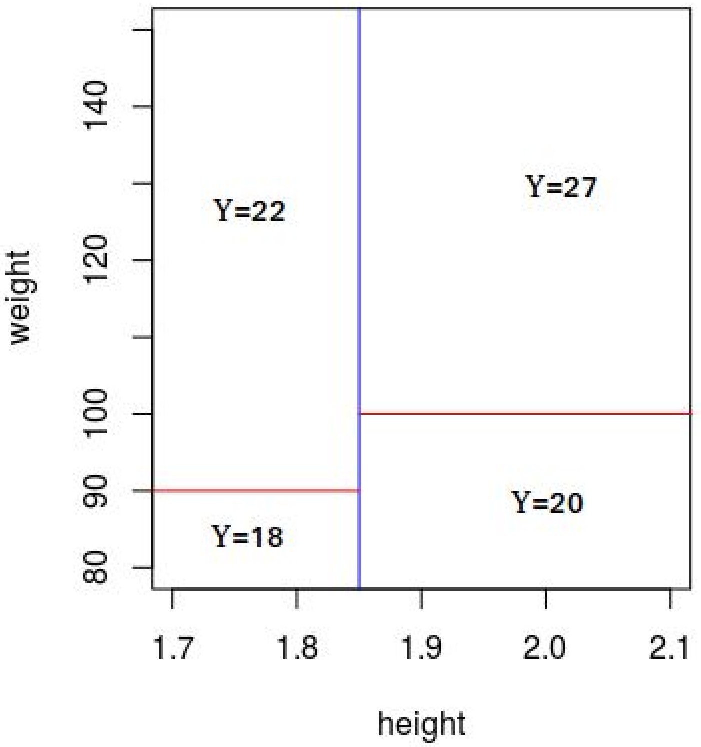


Example of a Classification Tree

2. Regression trees (Continuous data types) :

Decision trees where the target variable can take **continuous values** (typically real numbers) are called **regression trees**. (e.g. the price of a house, or a patient's length of stay in a hospital)





3.2.6 Naïve Bayes

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

The fundamental Naive Bayes assumption is that each feature makes an:

- independent
- equal

contribution to the outcome.

With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be **independent**.
- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can't predict the outcome accuratey. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

Bayes' Theorem

Bayes’ Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are events and P(B) ? 0.

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes’ theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size *n*) where:

$$X = (x_1, x_2, x_3, ..., x_n)$$

Hence, we reach to the result:

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

which can be expressed as:

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, as the denominator remains constant for a given input, we can remove that term:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

Now, we need to create a classifier model. For this, we find the probability of given set of inputs for all possible values of the class variable *y* and pick up the output with maximum probability. This can be expressed mathematically as:

$$y = argmax_y P(y)\prod_{i=1}^{n} P(x_i|y)$$

3.2.7 FEATURE SELECTION:

3.2.7.1 CHI-SQUARED TEST

The term "chi-squared test," also written as χ^2 **test**, refers to certain types of statistical hypothesis tests that are valid to perform when the test statistic is chi-squared distributed under the null hypothesis. Often, however, the term is used to refer to *Pearson's* chi-squared test and variants thereof. Pearson's chi-squared test is used to determine whether there is a statistically significant difference (i.e., a magnitude of difference that is unlikely to be due to chance alone) between the expected frequencies and the observed frequencies in one or more categories of a so-called contingency table.

In the standard applications of this test, the observations are classified into mutually exclusive classes. If the so-called null hypothesis is true, the test statistic computed from the observations follows a χ^2 distribution. The purpose of the test is to evaluate how likely the observed frequencies would be assuming the null hypothesis is true.

Test statistics that follow a χ^2 distribution occur when the observations are independent and normally distributed, which assumptions are often justified under the central limit theorem. There are also χ^2 tests for testing the null hypothesis of independence of a pair of random variables based on observations of the pairs.

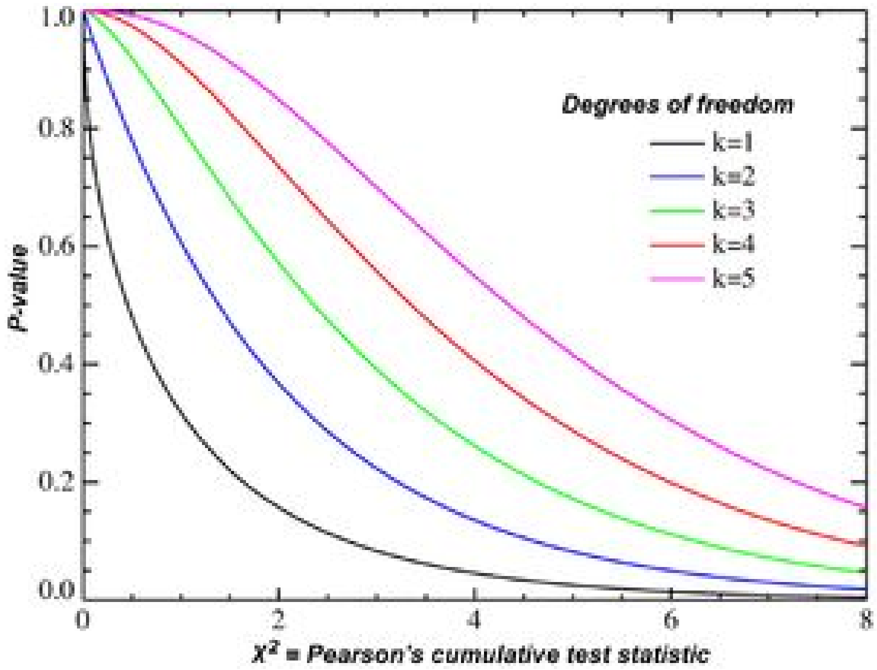

Degrees of freedom
k=1
k=2
k=3
k=4
k=5
P-value
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
X^2 = Pearson's cumulative test statistic

fig 7

3.2.7.2 GENETIC ALGORITHM

Genetic algorithms (GA) like neural networks are biologically inspired and represent a new computational model having its roots in evolutionary sciences. Usually GAs represent an optimization procedure in a binary search space, and unlike traditional hill climbers they do not evaluate and improve a single solution but a set of solutions or hypotheses, a so-called population. The GAs produce successor hypotheses by mutation and recombination of the best currently known hypotheses. Thus, at each iteration a part of the current population is replaced by offspring of the most fit hypotheses.

A living being is only a decoded structure of the chromosomes. Natural selection is the link between chromosomes and the performance of their decoded structures. In genetic algorithms the design variables or features that characterize an individual are represented in an ordered list called a string. Each design variable corresponds to a gene and the string of genes corresponds to a chromosome. A group of chromosomes is called a population. During each iterative procedure (referred to as generation), a new set of strings with improved performance is generated using three GA operators (namely, reproduction, crossover, and mutation).

Five phases are considered in a genetic algorithm are: Initial population, Fitness function, Selection, Crossover, Mutation.

Initial Population

The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve.An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution).In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.

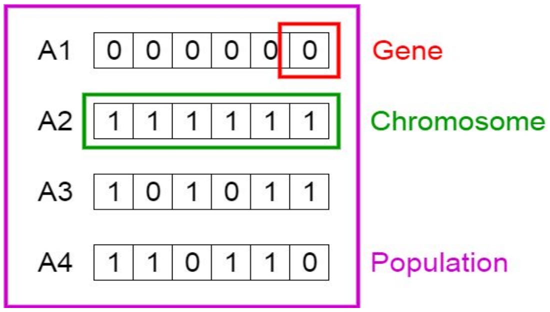


fig 8

Fitness Function

The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

Selection

The idea of **selection** phase is to select the fittest individuals and let them pass their genes to the next generation.Two pairs of individuals (**parents**) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Crossover

Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a **crossover point** is chosen at random from within the genes.

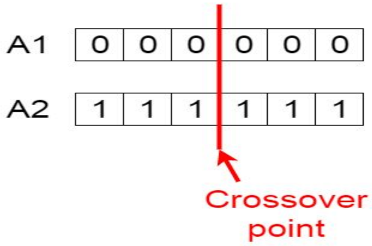


fig 9

Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.

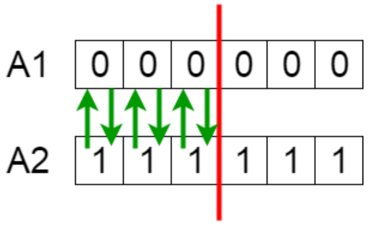


fig 10 The new offspring are added to the population**.**

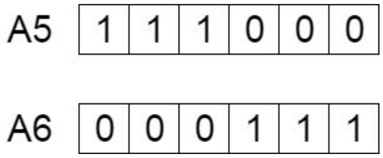


fig 11

Mutation

In certain new offspring formed, some of their genes can be subjected to a **mutation** with a low random probability. This implies that some of the bits in the bit string can be flipped.

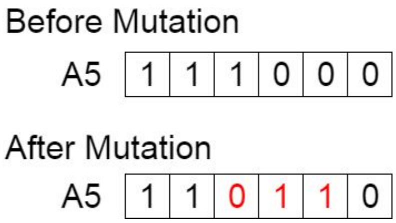


fig 12

Mutation occurs to maintain diversity within the population and prevent premature convergence.

4. IMPLEMENTATION:

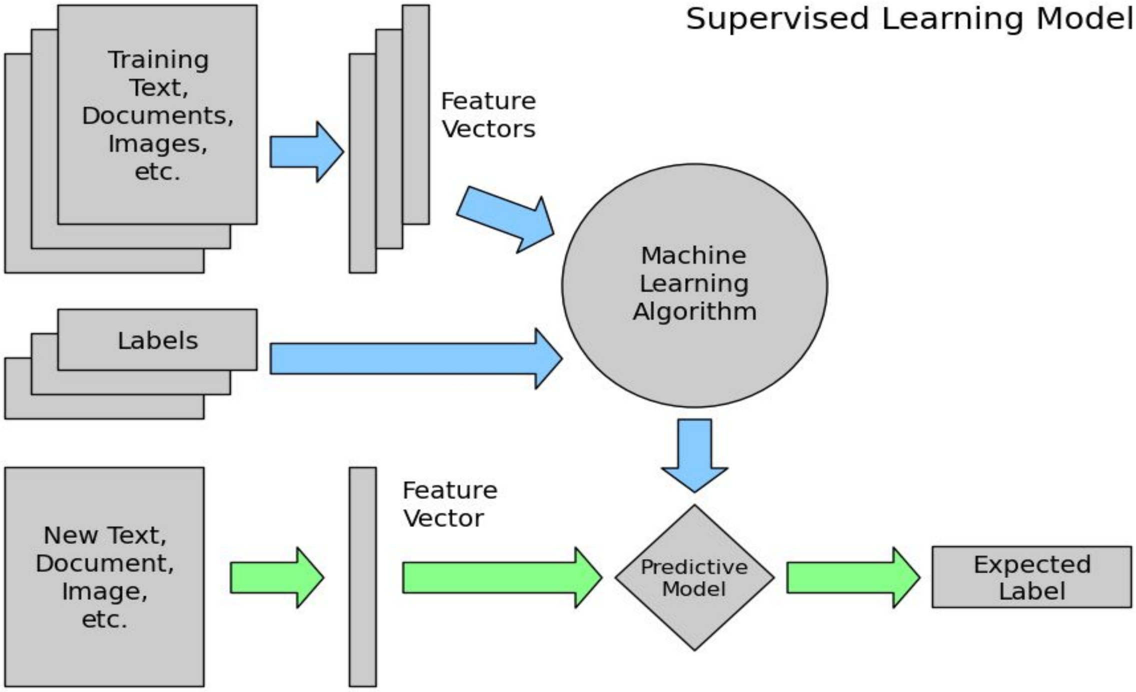
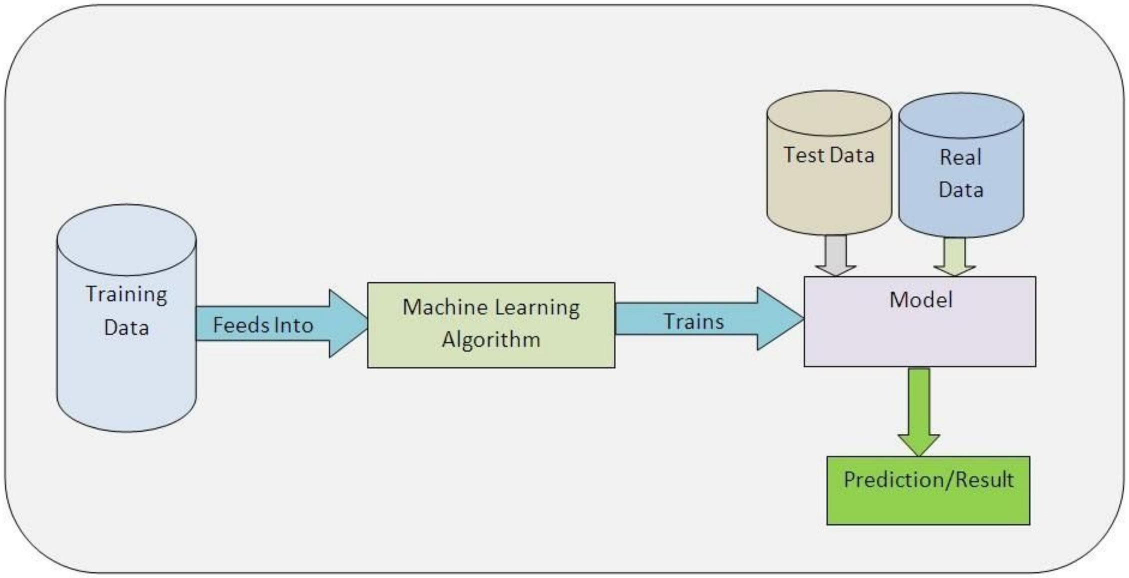

Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Fig 13



A Simple Machine Learning Pipeline Explanation

Fig 14

5. RESULT AND DISCUSSION:

This table is created with selecting all the features and getting the accuracy with normalization and without normalization.

With Normalization

Algorithm	Accuracy
1. Support vector machine classifier	96.49%
2. Logistic regression	45.61%
3. K-nearest neighbor classifier	57.89%
4. Naïve bayes	93.86%
5. Decision tree	75.44%
6. Random forest classifier	75.44%

Table 1.

Without Normalization

Algorithm	Accuracy
1. Support vector machine classifier	93.86%
2. Logistic regression	95.61%
3. K-nearest neighbor classifier	93.85%
4. Naive bayes	94.74%
5. Decision tree	94.74%
6. Random forest classifier	97.37%

Table 2.

The features that are being used are as follows:

S.no	**Specs**	**Score**
1	Mean radius	266.104917
2	Mean texture	93.897508
3	Mean perimeter	2011.102864
4	Mean area	53991.655924
5	Mean smoothness	0.149899
6	Mean compactness	5.403075
7	Mean concavity	19.712354
8	Mean concave points	10.544035
9	Mean symmetry	0.257380
10	Mean fractal dimension	0.000074
11	Radius error	34.675247
12	Texture error	0.009794
13	Perimeter error	250.571896
14	Area error	8758.504705
15	Smoothness error	0.003266
16	Compactness error	0.613785
17	Concavity error	1.044718
18	Concave point error	0.305232
19	symmetry error	0.000080
20	fractal dimension error	0.006371
21	Worst radius	491.689157
22	Worst texture	174.449400
23	Worst perimeter	3665.035416
24	Worst area	112598.431564

25	Worst smoothness	0.397366
26	worst compactness	19.314922
27	worst concavity	39.516915
28	worst concave points	13.485419
29	worst symmetry	1.298861
30	Worst fractal dimension	0.231522

Table 3.

Feature selection:

The Chi Square statistic is commonly used for testing relationships between categorical variables .The null hypothesis of the **Chi-Square test** is that no relationship exists on the categorical variables in the population; they are independent. After performing chi Square Test we are getting the accuracy of 98.24%.

Final 15 features are selected after selecting common features for chi squared test. The features are taken into account for their best score.

S.no	**specs**	**score**
1	Worst area	112598.431564
2	Mean area	53991.655924
3	Area error	8758.504705
4	Worst perimeter	3665.035416
5	Mean perimeter	2011.102864
6	Worst radius	491.689157
7	Mean radius	266.104917
8	Perimeter error	250.571896
9	Worst texture	174.449400
10	Mean texture	93.897508
11	Worst concavity	39.516915
12	Radius error	34.675247
13	Mean concavity	19.712354
14	Worst compactness	19.314922
15	Worst concave points	13.485419

Table 4.

Chi-squared test

classifiers	**Accuracy without normalization**	**Accuracy with normalization**
1. Support vector machine classifier	**92.98%**	**98.24%**
2. Logistic regression	**94.73%**	**36.84%**
3. K-nearest neighbor classifier	**93.85%**	**57.89%**
4. Naive bayes	**95.61%**	**95.61%**
5. Decision tree	**95.61%**	**92.98%**
6. Random forest classifier	**97.36%**	**63.15%**

Table 5.

After selecting the best 9 features from the attributes table, genetic algorithm is used to get the accuracy.

S.no	**specs**	**score**
1	Worst area	112598.431564
2	Mean area	53991.655924
3	Area error	8758.504705
4	Worst perimeter	3665.035416
5	Mean perimeter	2011.102864
6	Worst radius	491.689157
7	Mean radius	266.104917
8	Perimeter error	250.571896
9	Worst texture	174.449400

Table 6.

Genetic Algorithm

classifiers	**Accuracy without normalization**	**Accuracy with normalization**
Support vector machine classifier	**94.77%**	**97.37%**
Logistic regression	**94.74%**	**42.11%**
K-nearest neighbor classifier	**92.98%**	**57.89%**
Naive bayes	**95.61%**	**95.61%**
Decision tree	**95.61%**	**57.89%**
Random forest classifier	**96.49%**	**57.89%**

Table 7.

5.1 HEATMAP GENERATED (ALL FEATURES)

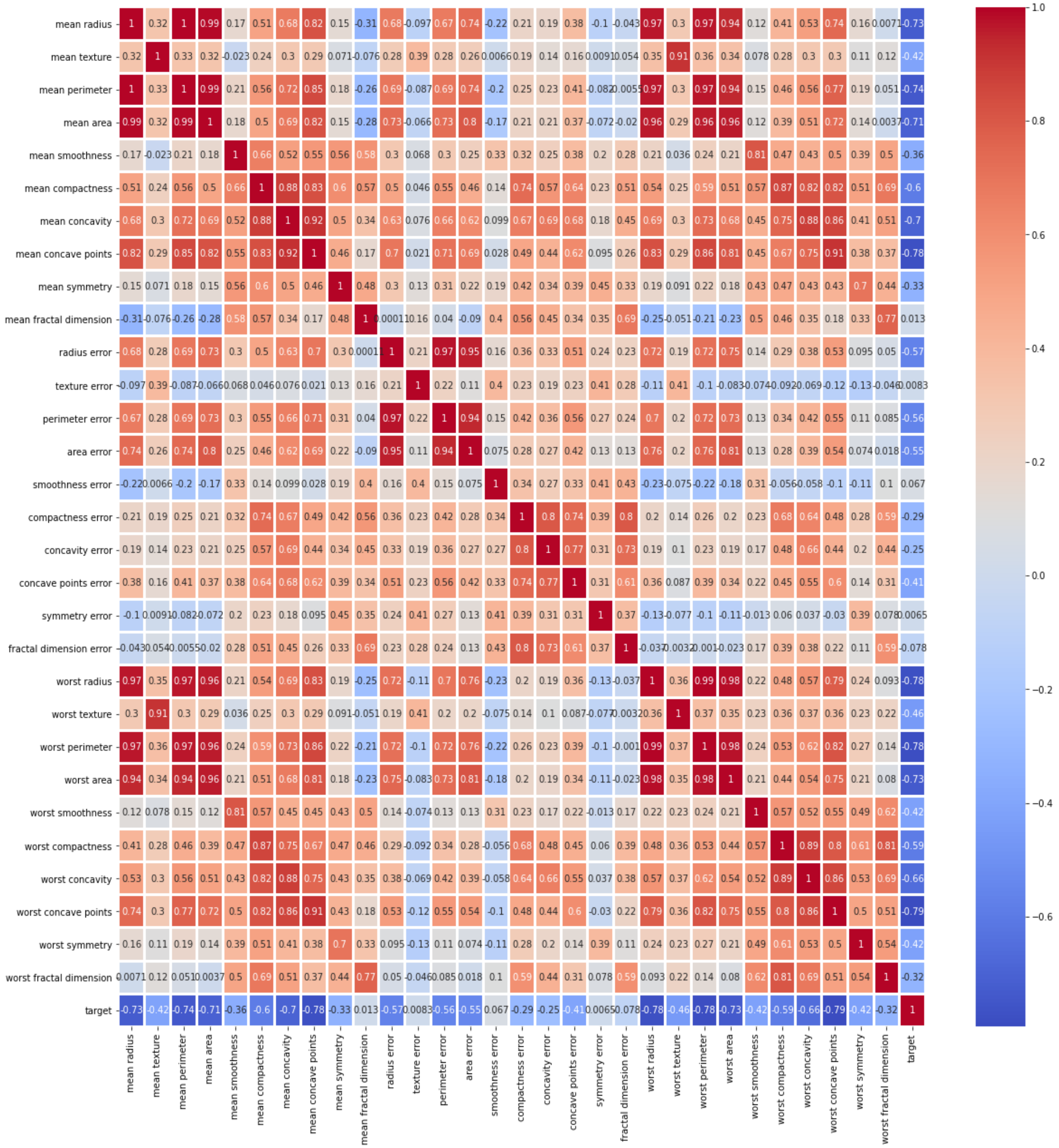


Fig 15

5.2 GRAPHPLOT

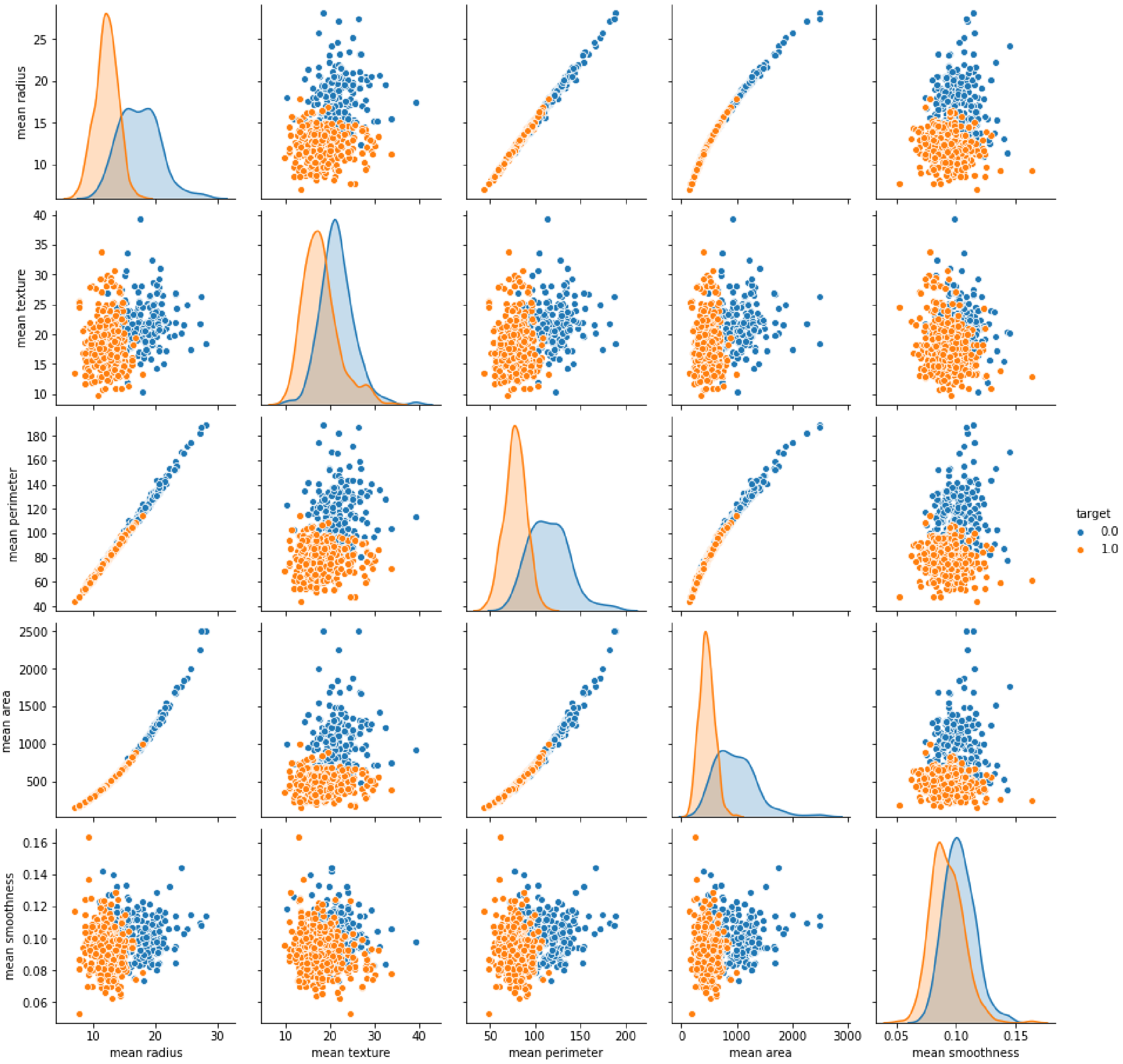
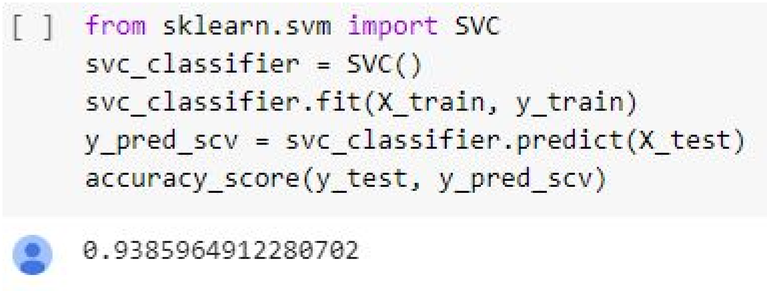


Fig 16

5.3 CODE SNIPPET:

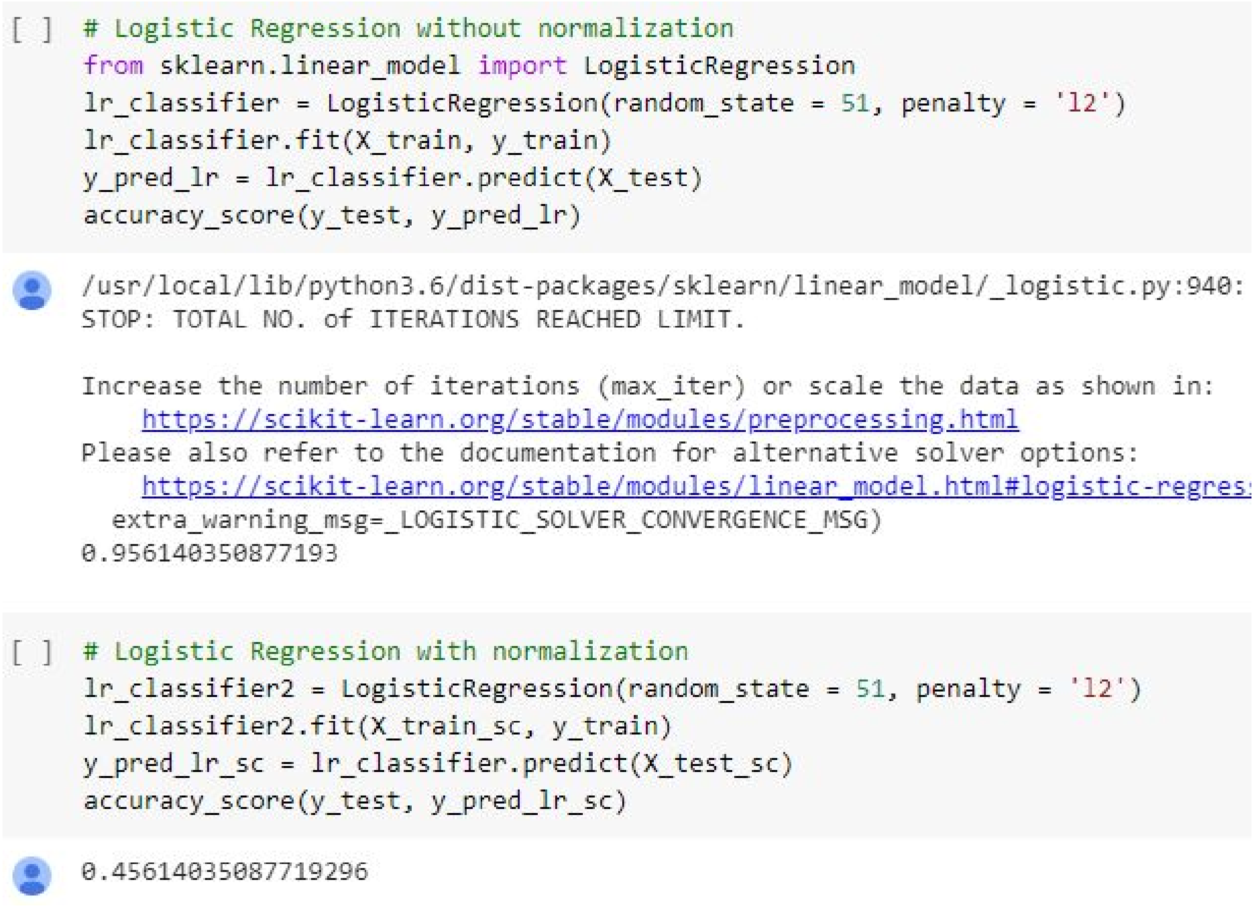
5.3.1 CLASSIFICATION METHODS (ALL FEATURES)



```
[ ]  from sklearn.svm import SVC
     svc_classifier = SVC()
     svc_classifier.fit(X_train, y_train)
     y_pred_scv = svc_classifier.predict(X_test)
     accuracy_score(y_test, y_pred_scv)
```

```
0.9385964912280702
```

fig 17



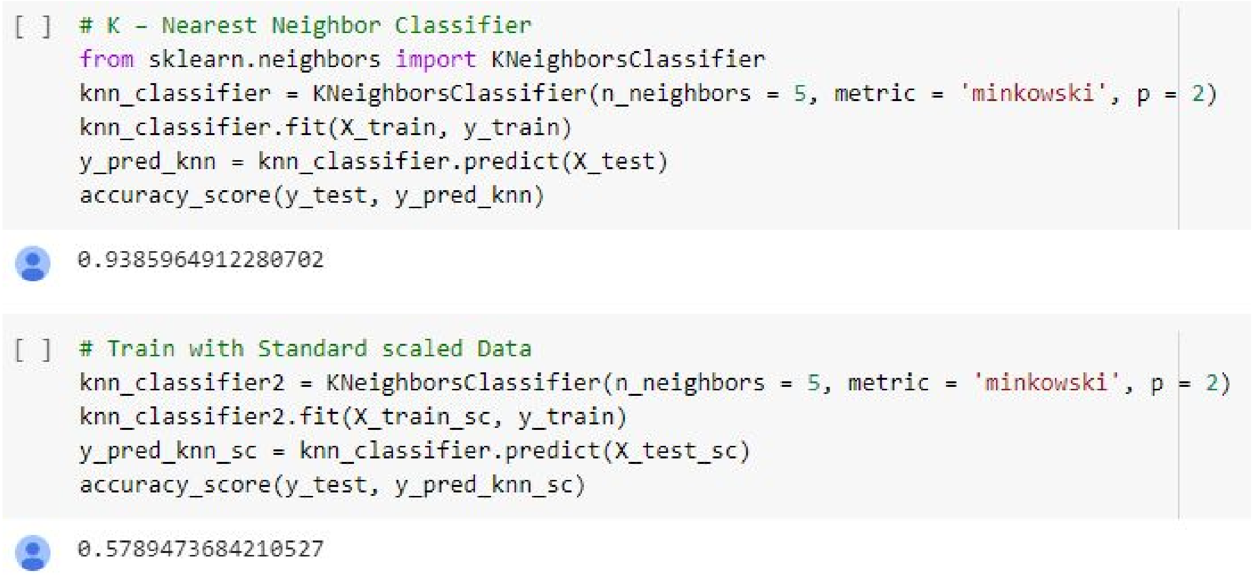
```
[ ]  # Logistic Regression without normalization
     from sklearn.linear_model import LogisticRegression
     lr_classifier = LogisticRegression(random_state = 51, penalty = 'l2')
     lr_classifier.fit(X_train, y_train)
     y_pred_lr = lr_classifier.predict(X_test)
     accuracy_score(y_test, y_pred_lr)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_logistic.py:940:
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
0.956140350877193
```

```
[ ]  # Logistic Regression with normalization
     lr_classifier2 = LogisticRegression(random_state = 51, penalty = 'l2')
     lr_classifier2.fit(X_train_sc, y_train)
     y_pred_lr_sc = lr_classifier.predict(X_test_sc)
     accuracy_score(y_test, y_pred_lr_sc)
```

```
0.45614035087719296
```

Fig 18



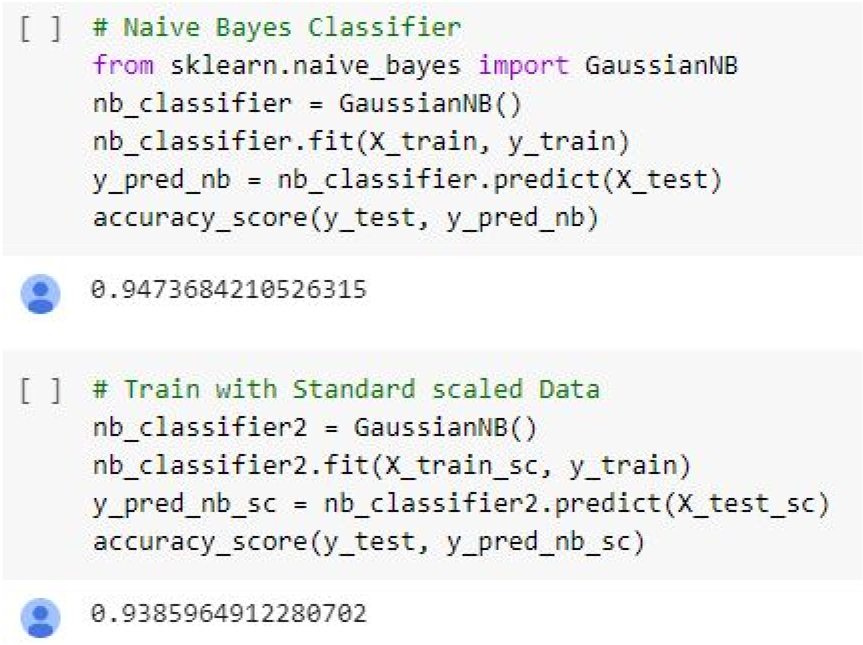
```
[ ] # K - Nearest Neighbor Classifier
    from sklearn.neighbors import KNeighborsClassifier
    knn_classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
    knn_classifier.fit(X_train, y_train)
    y_pred_knn = knn_classifier.predict(X_test)
    accuracy_score(y_test, y_pred_knn)
```

```
0.9385964912280702
```

```
[ ] # Train with Standard scaled Data
    knn_classifier2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
    knn_classifier2.fit(X_train_sc, y_train)
    y_pred_knn_sc = knn_classifier.predict(X_test_sc)
    accuracy_score(y_test, y_pred_knn_sc)
```

```
0.5789473684210527
```

fig 19



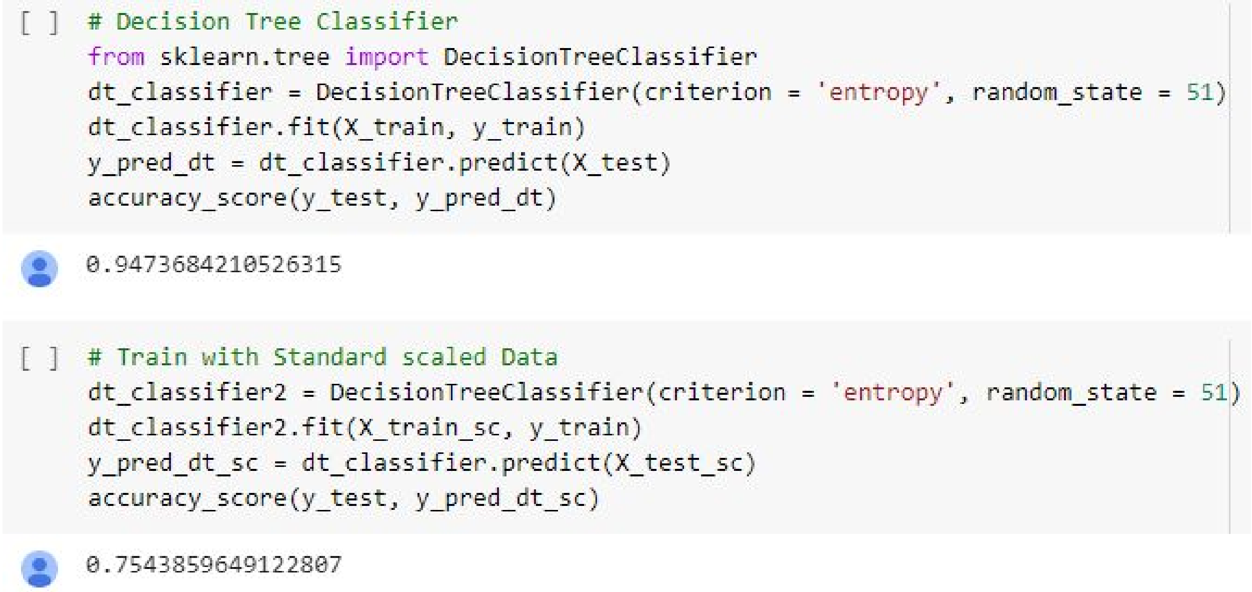
```
[ ] # Naive Bayes Classifier
    from sklearn.naive_bayes import GaussianNB
    nb_classifier = GaussianNB()
    nb_classifier.fit(X_train, y_train)
    y_pred_nb = nb_classifier.predict(X_test)
    accuracy_score(y_test, y_pred_nb)
```

```
0.9473684210526315
```

```
[ ] # Train with Standard scaled Data
    nb_classifier2 = GaussianNB()
    nb_classifier2.fit(X_train_sc, y_train)
    y_pred_nb_sc = nb_classifier2.predict(X_test_sc)
    accuracy_score(y_test, y_pred_nb_sc)
```

```
0.9385964912280702
```

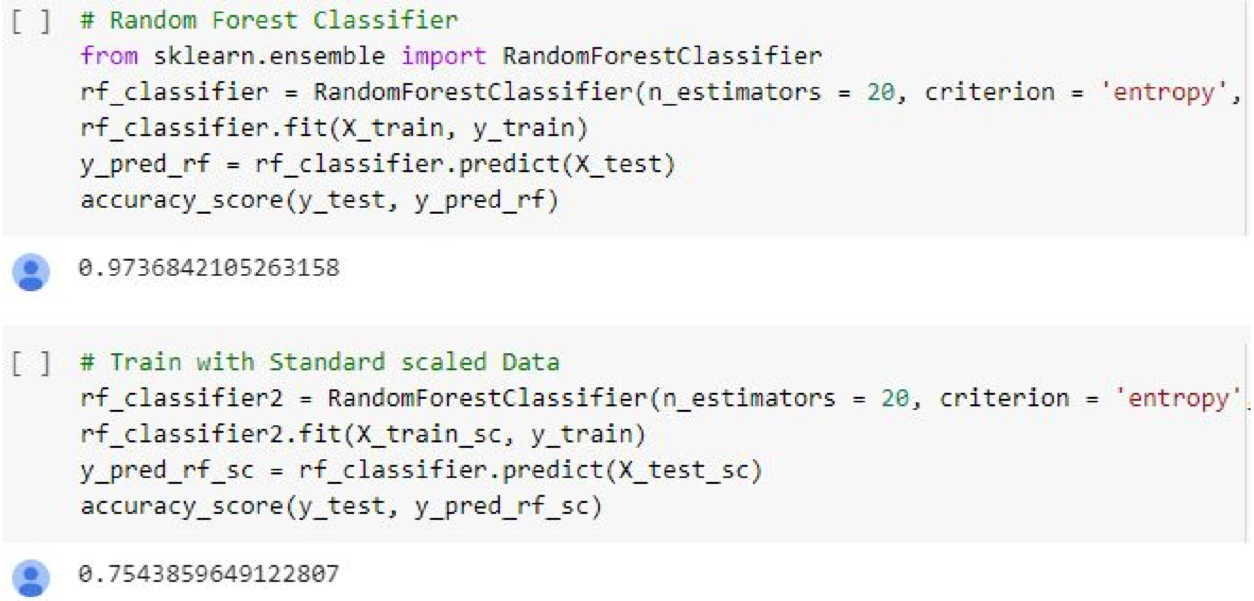
Fig 20



```
[ ] # Decision Tree Classifier
    from sklearn.tree import DecisionTreeClassifier
    dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
    dt_classifier.fit(X_train, y_train)
    y_pred_dt = dt_classifier.predict(X_test)
    accuracy_score(y_test, y_pred_dt)

    0.9473684210526315
```

```
[ ] # Train with Standard scaled Data
    dt_classifier2 = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
    dt_classifier2.fit(X_train_sc, y_train)
    y_pred_dt_sc = dt_classifier.predict(X_test_sc)
    accuracy_score(y_test, y_pred_dt_sc)

    0.7543859649122807
```

fig 21

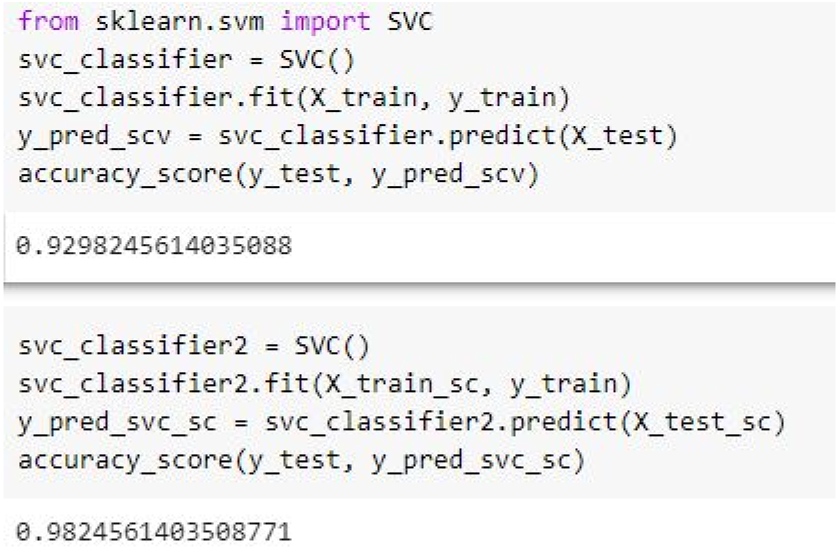


```
[ ] # Random Forest Classifier
    from sklearn.ensemble import RandomForestClassifier
    rf_classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy',
    rf_classifier.fit(X_train, y_train)
    y_pred_rf = rf_classifier.predict(X_test)
    accuracy_score(y_test, y_pred_rf)

    0.9736842105263158
```

```
[ ] # Train with Standard scaled Data
    rf_classifier2 = RandomForestClassifier(n_estimators = 20, criterion = 'entropy'
    rf_classifier2.fit(X_train_sc, y_train)
    y_pred_rf_sc = rf_classifier.predict(X_test_sc)
    accuracy_score(y_test, y_pred_rf_sc)

    0.7543859649122807
```

Fig 22

5.3.2 FEATURE SELECTION AND FEATURE IMPORTANCE



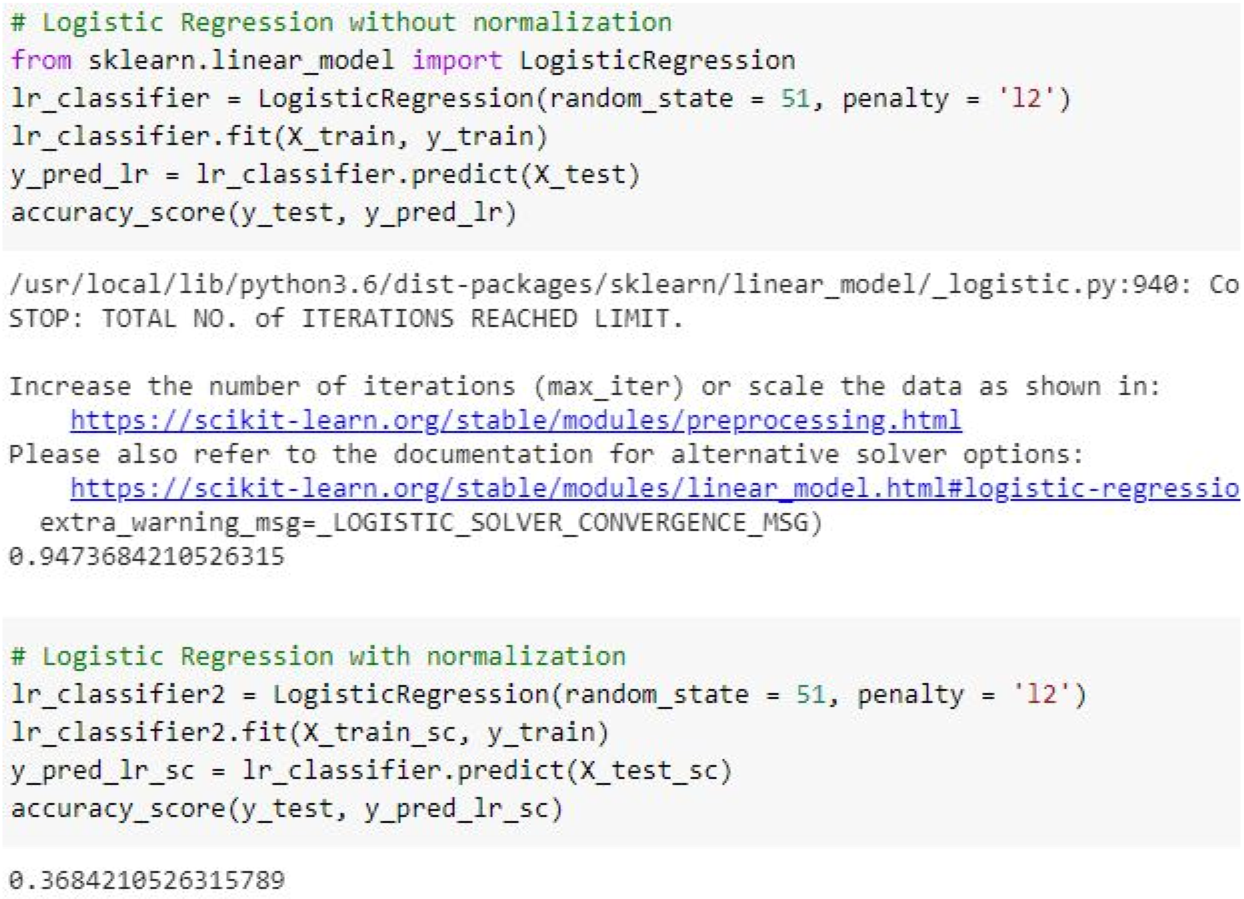
```
from sklearn.svm import SVC
svc_classifier = SVC()
svc_classifier.fit(X_train, y_train)
y_pred_scv = svc_classifier.predict(X_test)
accuracy_score(y_test, y_pred_scv)
```

```
0.9298245614035088
```

```
svc_classifier2 = SVC()
svc_classifier2.fit(X_train_sc, y_train)
y_pred_svc_sc = svc_classifier2.predict(X_test_sc)
accuracy_score(y_test, y_pred_svc_sc)
```

```
0.9824561403508771
```

Fig 23



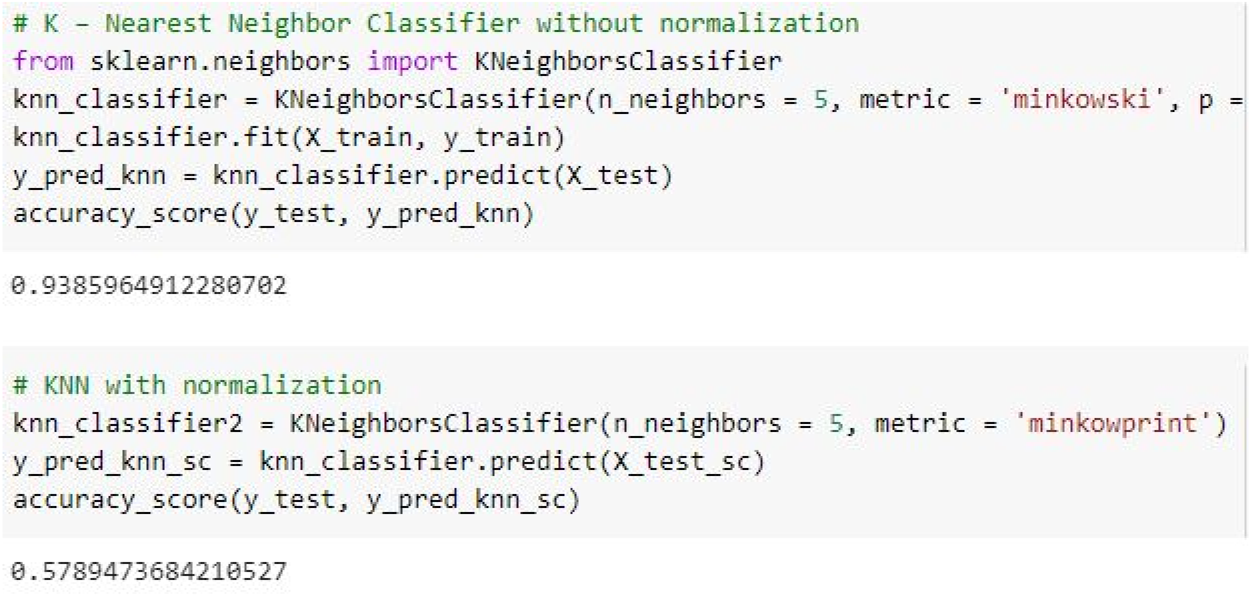
```
# Logistic Regression without normalization
from sklearn.linear_model import LogisticRegression
lr_classifier = LogisticRegression(random_state = 51, penalty = 'l2')
lr_classifier.fit(X_train, y_train)
y_pred_lr = lr_classifier.predict(X_test)
accuracy_score(y_test, y_pred_lr)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_logistic.py:940: Co
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressio
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
0.9473684210526315
```

```
# Logistic Regression with normalization
lr_classifier2 = LogisticRegression(random_state = 51, penalty = 'l2')
lr_classifier2.fit(X_train_sc, y_train)
y_pred_lr_sc = lr_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_lr_sc)
```

```
0.3684210526315789
```

Fig 24



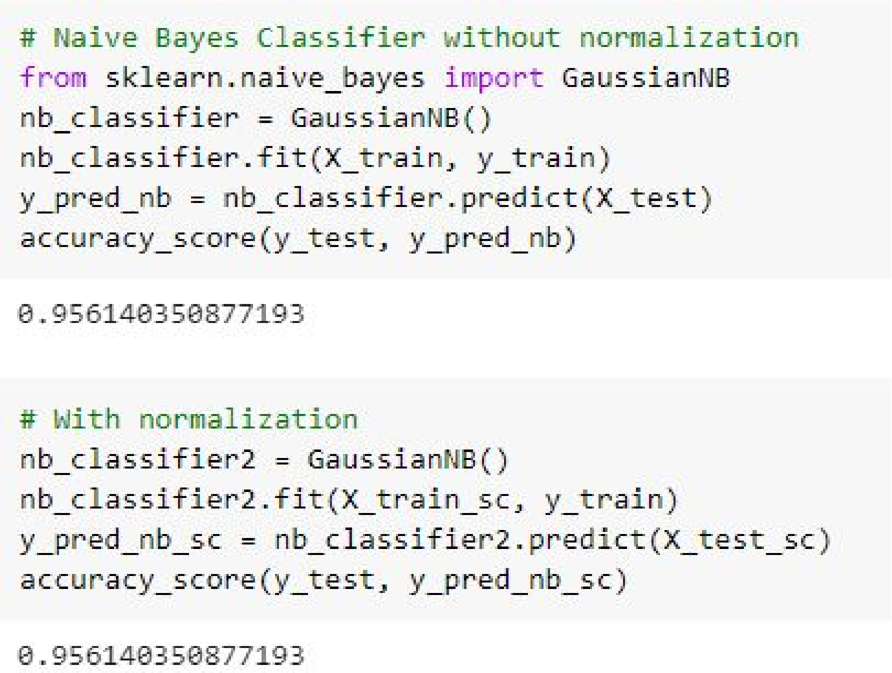
```
# K - Nearest Neighbor Classifier without normalization
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p =
knn_classifier.fit(X_train, y_train)
y_pred_knn = knn_classifier.predict(X_test)
accuracy_score(y_test, y_pred_knn)
```

```
0.9385964912280702
```

```
# KNN with normalization
knn_classifier2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowprint')
y_pred_knn_sc = knn_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_knn_sc)
```

```
0.5789473684210527
```

Fig 25



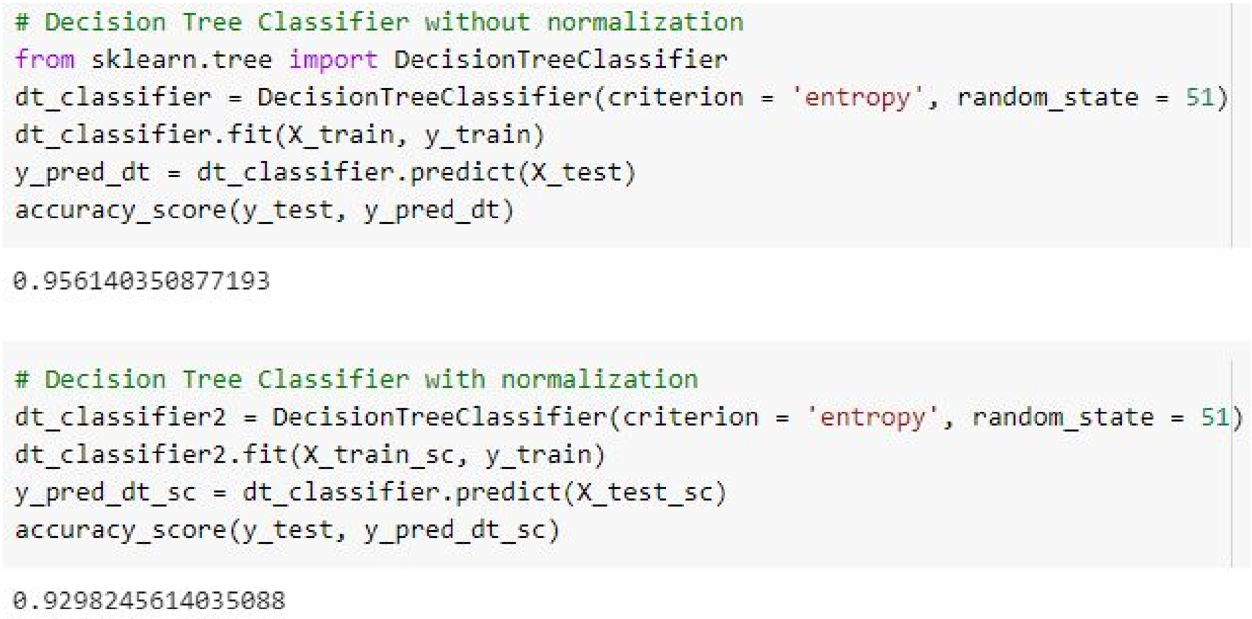
```
# Naive Bayes Classifier without normalization
from sklearn.naive_bayes import GaussianNB
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred_nb = nb_classifier.predict(X_test)
accuracy_score(y_test, y_pred_nb)
```

```
0.956140350877193
```

```
# With normalization
nb_classifier2 = GaussianNB()
nb_classifier2.fit(X_train_sc, y_train)
y_pred_nb_sc = nb_classifier2.predict(X_test_sc)
accuracy_score(y_test, y_pred_nb_sc)
```

```
0.956140350877193
```

Fig 26



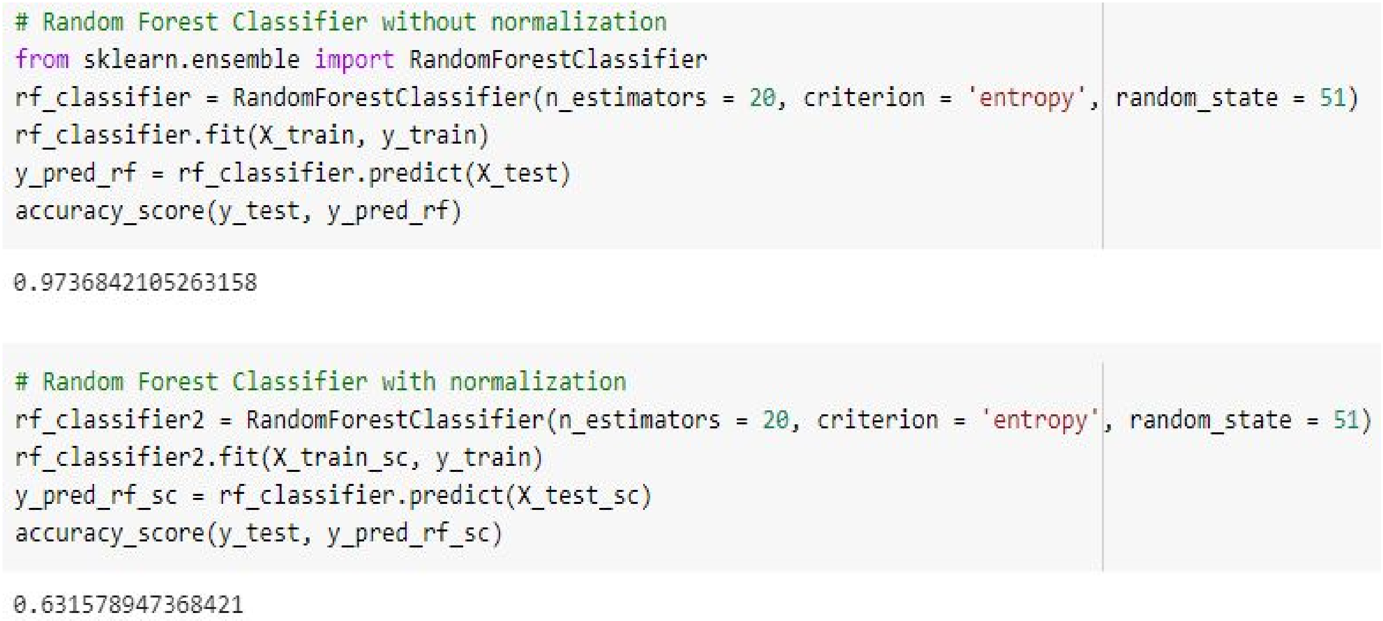
```
# Decision Tree Classifier without normalization
from sklearn.tree import DecisionTreeClassifier
dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
dt_classifier.fit(X_train, y_train)
y_pred_dt = dt_classifier.predict(X_test)
accuracy_score(y_test, y_pred_dt)
```

```
0.956140350877193
```

```
# Decision Tree Classifier with normalization
dt_classifier2 = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
dt_classifier2.fit(X_train_sc, y_train)
y_pred_dt_sc = dt_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_dt_sc)
```

```
0.9298245614035088
```

Fig 27



```
# Random Forest Classifier without normalization
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
accuracy_score(y_test, y_pred_rf)
```

```
0.9736842105263158
```

```
# Random Forest Classifier with normalization
rf_classifier2 = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier2.fit(X_train_sc, y_train)
y_pred_rf_sc = rf_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_rf_sc)
```

```
0.631578947368421
```

Fig 28

6. CONCLUSION:

Comparing to all other cancers, breast cancer is one of the major causes of death in women. So, the early detection of breast cancer is needed in reducing life losses. In this paper we have applied techniques namely feature selection, and classification methods to further optimize the feature model and increasing the accuracy. It helps in predicting breast cancer as accurately as possible. Our study reveals that **support vector machine classifier** gives the maximum accuracy at **96.49%** with normalization and **random forest classifier** gives best accuracy at **97.37%** without normalization.

Using best 15 features in chi-squared test we get **random forest classifier** to give best accuracy of **97.36**% without normalization and **svc** with accuracy of **98.24%.**

9 best feautres were further refined to get more accuracy out of **genetic algorithm** we used to get the best accuracy of **97.37%** and **94.77%** with and without normalization acquired by **svm classifier**.

7. DISCUSSION:

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model. To attain feature selection many irrelevant features are discarded and only the best scoring features used to acquire better accuracy. The dataset needs to be more diverse with the features and more attributes to get more features which ultimately results in better accuracy in predicting the cancer. Modern science has advanced so much that we can now predict and diagnose with greater accuracy saving lives. This project is intended to do greater work in the future and more improvement can be done like using it on web application, making interface etc.

8. REFERENCES:

[1] International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-11, September 2019Breast Cancer Detection using Machine Learning.

[2] V. Vishrutha, M. Ravishankar Early Detection and Classification of Breast Cancer. Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 327).

[3] Majid Iranpour , Sanaz Almassi , Morteza Analoui Breast Cancer Detection from FNA Using SVM and RBF Classifier Published 2007.

[4] Vikas Chaurasia , Saurabh Pal and BB Tiwari, Prediction of benign and malignant breast cancer using data mining techniques. Journal of Algorithms & Computational Technology 2018, Vol. 12(2).

[5] Hiba Asria ,Hajar Mousannifb ,Hassan Al Moatassime ,Thomas Noeld Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis. The 6th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS 2016).

[6] Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms.Expert Systems with Applications Volume 41, Issue 4, Part 1, March 2014.

[7] Breast cancer diagnosis using GA feature selection and Rotation Forest Emina Aličković & Abdulhamit Subasi *Neural Computing and Applications*, published 18 nov 2015.

[8] Breast cancer diagnosis using genetic algorithm for training feed forward back propagation. Hussein AttyaLafta ; Noor KdhimAyoob ; Asraa Abdullah Hussein. 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT).

[9] Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets Shokoufeh Aalaei, Hadi Shahraki, Alireza Rowhanimanesh, and Saeid Eslami. Published Iran J Basic Med Sci. 2016 May,19.