VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A Project Report
on

"Analysis of Machine Learning Algorithms on Real Time Data Sets"

Submitted in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* by

Prashanth M	**1RN14CS068**
Praveen M	**1RN14CS072**
Vinyas R	**1RN14CS120**
Yugandhar G	**1RN14CS123**

Under the Guidance of:
Mrs. Sudhamani.M.J
Asst. Professor
Dept. of Computer Science & Engineering

Department of Computer Science and Engineering
RNS Institute of Technology
Channasandra, Dr.Vishnuvardan Road, Bengaluru-560 098

2017-2018

RNS Institute of Technology

Channasandra, Dr. Vishnuvardan Road, Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work entitled **"*Analysis of Machine Learning Algorithms On Real Time Data Sets*"** has been successfully carried out by **Prashanth M, Praveen M, Vinyas R and Yugandhar G** bearing USNs **1RN14CS068, 1RN14CS072, 1RN14CS120 and 1RN14CS123** respectively**,** bona fide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2017-2018. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Internal Guide:

Mrs.Sudhamani.M.J	**Dr. G T Raju**	**Dr. M K Venkatesha**
Asst. Professor	**Dean, Prof. and HOD**	**Principal**

External Viva:

Name of the Examiners	**Signature with Date**
1._____	
2._____	

ABSTRACT

Machine learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalisation in order to get a system that performs well on yet unseen data instances. Machine learning is a relatively new discipline within Computer Science that provides a collection of data analysis techniques. Some of these techniques are based on well-established statistical methods. Fisher's Iris data base is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. Cluster analysis plays a vital role in various fields in order to group similar data from the available database. There are various clustering algorithms available in order to cluster the data but the entire algorithms are not suitable for all applications. Classification and clustering are both used to group data (observations) into multiple groups where those in the same group are in some way similar and those in different groups are not. In this project, we envisage distinct classification techniques and assessed the effectiveness of the algorithms on Fisher's Iris dataset. Finger vein recognition is a kind of biometric authentication system. This is one among many forms of biometrics used to recognize the individuals and to verify their identity. This Project presents a finger vein authentication system using Classification and Clustering algorithms. The grading of the agar wood oil to the high and low quality is done using manually such as human trained grader. It was performed based on the agar wood oil physical properties such as human experience and perception and the oil colour, odour and long lasting aroma.

ACKNOWLEDGMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible. Many are responsible for the knowledge and experience that we have gained during our project and throughout the course. Hence, we feel that expressing our deepest gratitude is just not formality but a part and parcel of the project.

We would like to profoundly thank the **Management** of **R N S Institute of Technology** for providing such a healthy environment for the successful completion of this Project.

We would like to express our thanks to **Dr. H N Shivashankar, Director** and **Dr. M K Venkatesha, Principal,** for their encouragement that motivated us for the successful completion of this Project.

We are extremely grateful to our very own and beloved **Dean, Prof. and HOD - CSE**, **Dr.GTRaju**, for having accepted to patronize us in the apt direction with all his wisdom. We would also like to express earnest thanks to our project guide **Mrs.Sudhamani.M.J, Asst. Prof.**, and coordinator, **Mr. Devaraju B M, Asst. Professor, Dept. of CSE**. They are our motivator, guide and constant source of knowledge and inspiration for us towards the preparation of this project and their consent and whole hearted cooperation in providing all the facilities and resources that we had required for successful implementation of this project work. We would also like to thank them for their encouragement and support throughout this project work.

Last but not the least, we thank all our friends who have helped us directly or indirectly during this project and made it successful. At the same time, we thank all our faculty and lab assistants of the Computer Science and Engineering Department for their kind co-operation.

PRASHANTH M
PRAVEEN M
VINYAS R
YUGANDHAR G

CONTENTS

LIST OF FIGURES

Fig No. Figure Name	Page No.

LIST OF TABLES

Chapter 1

INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

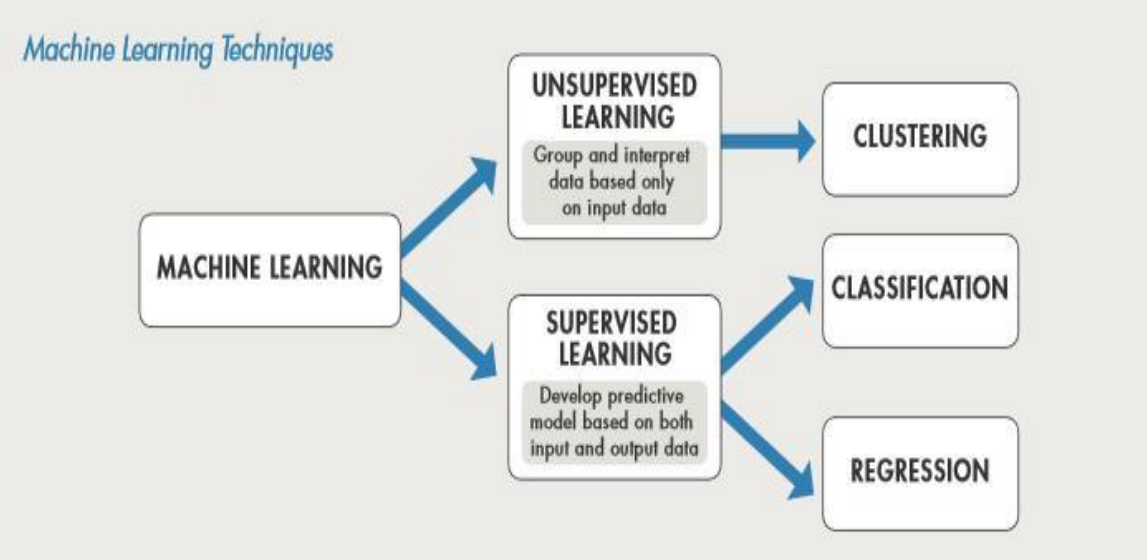


Fig 1: Machine learning classification

1.1 Machine Learning

Machine learning teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. Machine learning algorithms find natural patterns in data that generate

insight and help you make better decisions and predictions. They are used every day to make critical decisions in medical diagnosis, stock trading, energy load forecasting, and more.

1.1.1 Supervised Learning

The aim of supervised machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models.

1.1.2 Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. A central case of unsupervised learning is the problem of density estimation in statistics, though unsupervised learning encompasses many other problems (and solutions) involving summarizing and explaining key features of the data.

1.2 Clustering Algorithms

Clustering is the act of assigning a set of elements into subsets, or clusters, so that elements in the same cluster are, in some sense, similar. Cluster analysis groups the given data objects based on only information found in the data and describes the objects and their relationships. The data objects have the maximum similarity within a group. Cluster analysis can be applied to many areas including biology, medicine, and market research. Each of these areas has the potential to amass large amounts of data. There are dozens of different methods used to cluster data, each with its own shortcomings and limitations. Another important issue is visualizing the strength, or connectivity, of clusters. It is often impossible to visualize the original data if it is high-dimensional, so the ability to visualize the strength of a clustering would benefit the applications of data mining and cluster analysis. Clustering algorithms fall into two broad groups:

• Hard clustering, where each data point belongs to only one cluster

• Soft clustering, where each data point can belong to more than one cluster

1.3 Classification Algorithm

Classification of remotely sensed data is used to assign corresponding levels with respect to groups with homogeneous characteristics, with the aim of discriminating multiple objects from each other within the image. A classification problem arises when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object. Many problems in business, science, industry, and medicine can be treated as classification problems. It predicts discrete responses—for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging. Speech recognition and credit scoring. When you are working on a classification problem, begin by determining whether the problem is binary or multiclass. In a binary classification problem, a single training or test .item (instance) can only be divided into two classes—for example, if you want to determine whether an email is genuine or spam. In a multiclass classification problem, it can be divided into more than two.

1.4 Regression

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in electricity demand. Applications include forecasting stock prices, handwriting recognition, and acoustic signal processing. Regression analysis is used when you want to predict a continuous dependent variable from a number of independent variables. If the dependent variable is dichotomous, then logistic regression should be used. The independent variables used in regression can be either continuous or dichotomous. Independent variables with more than two levels can also be used in regression analyses, but they first must be converted into variables that have only two levels. This is called dummy coding. Usually, regression analysis is used with naturally-occurring variables, as opposed to experimentally manipulated variables, although you can use regression with experimentally manipulated variables.

Chapter 2

LITERATURE SURVEY

As a biometric characteristic, finger vein has several desirable properties, such as universality, distinctiveness, permanence and acceptability. In addition to, compared with other biometric characteristics (for example, face, gait, fingerprint and so on), it has other distinct advantages in the following two points. Living body identification. It means that only vein in living finger can be captured, and further used to perform identification. Internal characteristic. It is hard to copy or forge finger vein, and very little external factor can damage finger vein, which guarantee the high security of finger vein recognition. These two advantages make finger vein an irreplaceable biometric characteristic, and attract more and more attentions from research teams. A typical finger vein identification system mainly includes image acquisition, pre-processing, feature extraction and matching, as shown in Fig 2.1.

Fig 2.1: Finger vein identification system

Japanese medical researchers, proposed finger vein based identity identification, and gave an effective feature extraction method. Yanagawa et al proved the diversity of human finger vein patterns and the usefulness of finger veins for identity identification on 2, 024 fingers of 506 persons. They show that, two finger vein patterns are identical if and only if they are from the same finger in the same hand of the same person. These two literatures are the foundation of finger vein recognition, which open the era of finger vein recognition. In the early days of finger vein recognition, there are two significant literatures, which are all from Miura et al. The first one is about a feature extracted method, named repeated line tracking. Line tracking starts at various positions, and moves along the direction of vein pattern pixel by pixel. In the second literature, in order to overcome the influence of vein patterns' various widths and brightness, maximum curvature was developed to extract the centerlines of vein.

By the development of the past decade, finger vein recognition ushers in the evolution period now. The most representative literature is, in which authors used Gabor to extract finger vein patterns, and fuse finger vein and finger texture. Beside, Yang et al proposed to use the width of phalangeal joint as a soft biometric trait to enhance the recognition accuracy for finger vein. Although, there are some valuable works in finger vein recognition, lots of key problems are unsolved, for example, the acquisition of high quality image, the high recognition rate, the large scale applications. In this paper, first we comprehensively review main techniques of finger vein recognition, which include image acquisition devices, existing public databases and some typical feature extraction and matching methods. And then, the unsolved key problems and potential development directions in finger vein recognition are analyzed. There are two ways of finger vein image acquisition, i.e., light reflection method and light transmission method, The main difference between two methods is the position of near-infrared light. In detail, in light reflection method, near-infrared light is placed in finger palmar side, and finger vein pattern is captured by the reflected light from finger palmar surface. Conversely, near-infrared light is placed in finger dorsal side in light transmission method, and the light will penetrate finger. Compared with light reflection method, light transmission method can capture high-contrast image, so most of image acquisition devices employ light transmission method.

There are multiple public finger vein databases, and five typical databases are introduced. The first one was built by Shandong University, named SDUMLA-FV database, and it was a part of a homologous multimodal database. Another finger vein database was published by Ajay and Zhou, and it also was a part of a homologous multimodal database. We call it HKPU-FV database. Light reflection; Light transmission. Following. The third database was from University of Twente, abbreviated UTFV database. Recently, two finger vein databases were published, which were from Tsinghua University and Chonbuk Nation University respectively. The previous database is a part of a homologous multimodal database, and we call it THU-FV database in this paper. The other one is named MMCBNU_6000 database. Light transmission-based image acquisition advice was used on all three databases. And for three databases, the number of subject/finger is limited. Besides, images from different databases have different sizes, different contrast, different backgrounds, and different quality. There are six typical vein pattern-based feature extraction

methods, including repeated line tracking, maximum curvature, Gabor, mean curvature, region growth , and modified repeated line tracking .This group of method is the mainstream in finger vein extraction. In these methods, the vein patterns are segmented firstly, and then the geometric shape or topological structure of vein pattern is used for matching.

Although some advancement has been made, there are still some problems in finger vein recognition. The first problem is the distinctiveness of finger vein pattern. Yanagawa et al proved the diversity of human finger vein patterns on 2, 024 fingers of 506 persons, but medical evidence is not enough. So, in large scale applications, we cannot confidently predict how the recognition rate will be and if the classification result is reliable. And it also concerns if finger vein can be used in judiciary like fingerprint and face. Besides, the medical evidence about the stability of finger vein is not enough, either. In practical applications, the corresponding problem is the effectiveness of the enrolled finger vein template. In other word, it means if it is necessary to replace the enrolled template every 5 or 10 years. And if the surrounding environment and diseases can affect the finger vein pattern is uncertain. The second problem is about image acquisition. The price of finger vein acquisition device is still high now, which is one factor that limits the application of finger vein recognition. In public databases, there are some common issues about image quality, for example, low contrast, image blurring, excessive brightness, excessive dark and stains. So, there is a space for the performance improvement of image acquisition device. Dai et al used non uniform intensity infrared light to capture finger vein image, and the quality of captured image has been improved at certain extent. In total, the device with low price and high performance will vastly promote the development of finger vein recognition. The third problem is finger displacement during image acquisition. Finger displacement can be divided into 2 dimensional posture changes, i.e. shift along x-axis, y-axis and z-axis, and 3 dimensional posture changes, i.e., rotation around x-axis, y-axis, z-axis. Compared with 2 dimensional posture changes, it is harder to handle 3 dimensional posture changes. Transformation models, which were based on binary finger vein pattern and minutia points, were used to finger alignment. And some works align displaced fingers in preprocessing, but these methods mainly focus on overcoming 2 dimensional posture changes. It may be easier to handle this problem from device, for example, adding a groove to fix finger. The last one is lack of large scale practical application. Hitachi LTD. has researched finger vein recognition

since 1997, and applied finger vein recognition into many domains, for example, ATM automatic teller machine and car lock. The inland industrial communities, which research product of finger vein recognition, start late, and the scale of application is relatively small.

The methodology for Iris Flower Species System is described. In this work, IRIS flower classification using Neural Network. The problem concerns the identification of IRIS flower species on the basis of flower attribute measurements. Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS flower and how the prediction was made from analyzing the pattern to form the class of IRIS flower. By using this pattern and classification, in future upcoming years the unknown data can be predicted more precisely. Artificial neural networks have been successfully applied to problems in pattern classification, function approximations, optimization, and associative memories. In this work, Multilayer feed-forward networks are trained using back propagation learning algorithm. The model for Iris Flower Species System is described. Existing iris flower dataset is preloaded in MATLAB and is used for clustering into three different species. The dataset is clustered using the k-means algorithm and neural network clustering tool in MATLAB. Neural network clustering tool is mainly used for clustering large data set without any supervision. It is also used for pattern recognition, feature extraction, vector quantization, image segmentation, function approximation, and data mining. Results/Findings: The results include the clustered iris dataset into three species without any supervision. The model for Iris Flower Species System is described. The proposed method is applied on Iris data sets and classifies the dataset into four classes. In this case, the network could select the good features and extract a small but adequate set of rules for the classification task. For Class one data set we obtained zero misclassification on test sets and for all other data sets the results obtained are comparable to the results reported in the literature.

Chapter 3

PROPOSED SYSTEM

This project has the goal of finding the best classification method and select the best voted classifier i.e. classifier that has most accuracy percentage. The loaded dataset is preprocessed. After preprocessing of dataset, we apply different machine learning algorithms and analyzed the performance of each algorithm.

3.1. NEAREST NEIGHBOUR ALGORITHM

The nearest neighbor algorithm was one of the first algorithms used to determine a solution to the travelling salesman problem. In it, the salesman starts at a random city and repeatedly visits the nearest city until all have been visited. It quickly yields a short tour, but usually not the optimal one.

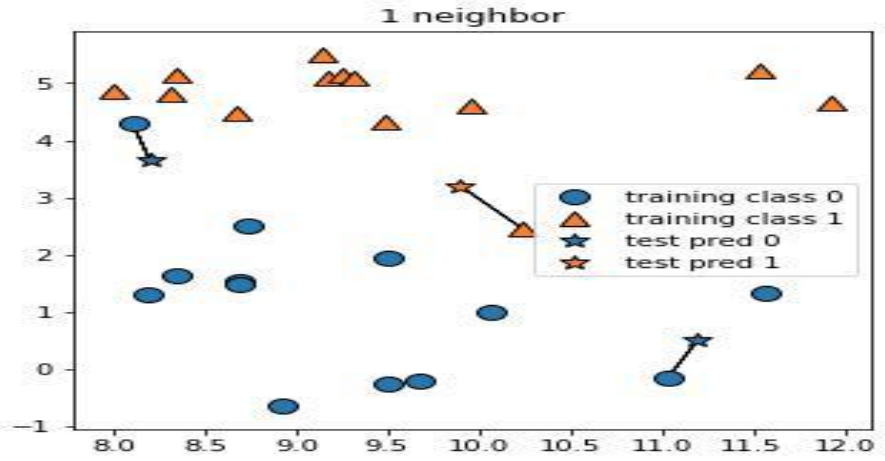


Fig 3.1: Example of nearest neighbor algorithm

Nearest neighbor analysis examines the distances between each point and the closest point to it, and then compares these to expected values for a random sample of points from a CSR (complete spatial randomness) pattern. CSR is generated by means of two assumptions: 1) that all places are equally likely to be the recipient of a case (event) and 2) all cases are located independently of one another. The mean nearest neighbor distance

$$\bar{d} = \frac{\sum_{i=1}^{N} d_i}{N}$$

where N is the number of points. d_i is the nearest neighbor distance for point I .

3.2 K NEAREST NEIGHBOUR ALGORITHM

In pattern recognition, the *k*-nearest neighbours algorithm (*k*-NN) is a non-parametric method used for classification and regression, n both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbor's.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.

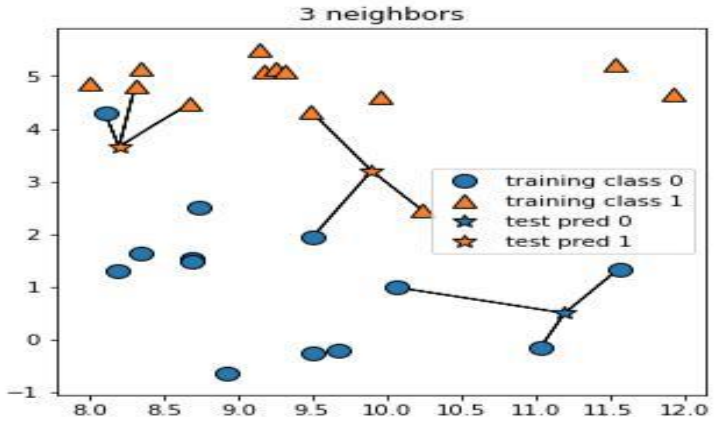


Fig 3.2: Example of k nearest neighbour algorithm

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the

average than the more distant ones. For example, a common weighting scheme consists in giving each neighbour a weight of 1/*d*, where *d* is the distance to the neighbour.

The neighbours are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.A peculiarity of the *k*-NN algorithm is that it is sensitive to the local structure of the data. Thealgorithm is not to be confused with *k*-means, another popular machine learning technique. For estimating the density at a point x, place a hypercube centred at x and keep increasing its size till k neighbours are captured. Now estimate the density using the formula,

$$p(x) = \frac{k/n}{V}$$

Where n is the total number of V is the volume of the hypercube. Notice that the numerator is essentially a constant and the density is influenced by the volume.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

3.3 LINEAR REGRESSION

Linear regression is perhaps one of the most well-known and well understood algorithms in statistics and machine learning.Machine learning, more specifically the field of predictive modelling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of explain ability. In applied machine

learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.

As such, linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables, but has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm.

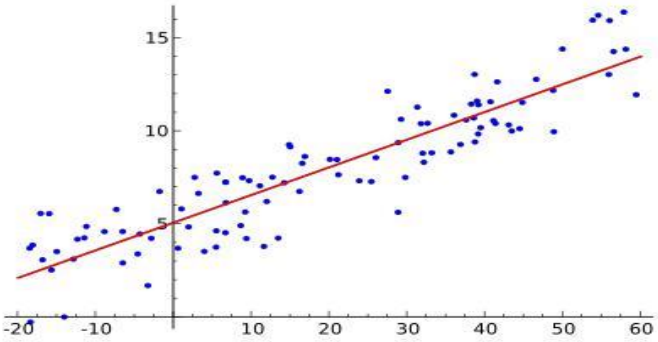


Fig 3.3: Example of Linear regression

It is common to make the additional hypothesis that the ordinary least squares method should be used to minimize the residuals(vertical distances between the points of the data set and the fitted line). Under this hypothesis, the accuracy of a line through the sample points ismeasured by the sum of squared residuals, and the goal is to make this sum as small aspossible. Other regression methods that can be used in place of ordinary least squares include least absolute deviations (minimizing the sum of absolute values of residuals) and the Theil–Sen estimator (which chooses a line whose slope is the median of the slopes determined by pairs of sample points). Deming regression (total least squares) also finds a line that fits a set of two-dimensional sample points, but (unlike ordinary least squares, least absolute deviations, and median slope regression) it is not really an instance of simple linear regression, because it does not separate the coordinates into one dependent and one independent variable and could potentially return a vertical line as its fit.

The equation for linear regression

$$Y = \beta_0 + \beta_1 X$$

$$\beta_1 = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

3.4 LOGISTIC REGRESSION

In statistics, logistic regression, or logit regression, or logit model is a regression model Where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analysed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model.

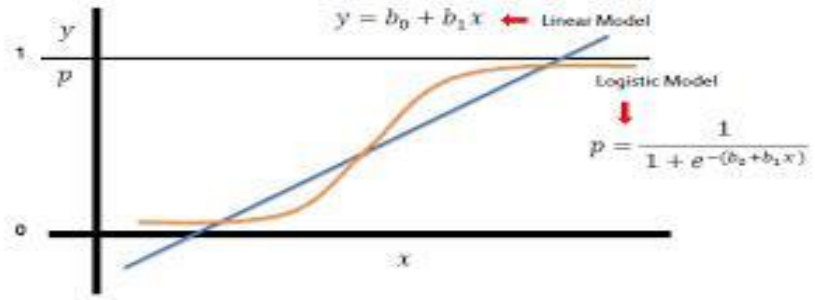


Fig 3.4 Example of Logistic regression

The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor. The model is a direct probability model and not a classifier.

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score

(TRISS), which iswidely used to predict mortality in injured patients, was originally developed by Boyd et al. using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression. Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observedcharacteristics of the patient. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.Logistic regression can be binomial,ordinal or multinomial. Binomial or binary logistic regression deals with situations in which the observed outcome for a dependent variable can have only two possible types, "0" and "1". Multinomial logistic regression deals with situations where the outcome can have three or more possible types that are not ordered. Ordinal logistic regression deals with dependent variables that are ordered.

Logistic regression achieves this by taking the log odds of the event ln(P/1−P), where, P is the probability of event. So P always lies between 0 and 1.

$$Z_i = ln\left(\frac{P_i}{1-P_i}\right) = \alpha + \beta_1 x_1 + .. + \beta_n x_n$$

Taking exponent on both sides of the equation gives.

$$P_i = E(y = 1|x_i) = \frac{e^z}{1+e^z} = \frac{e^{\alpha+\beta_i x_i}}{1+e^{\alpha+\beta_i x_i}}$$

3.5 SUPPORT VECTOR MACHINES

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and

predicted to belong to a category based on which side of the gap they fall.In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

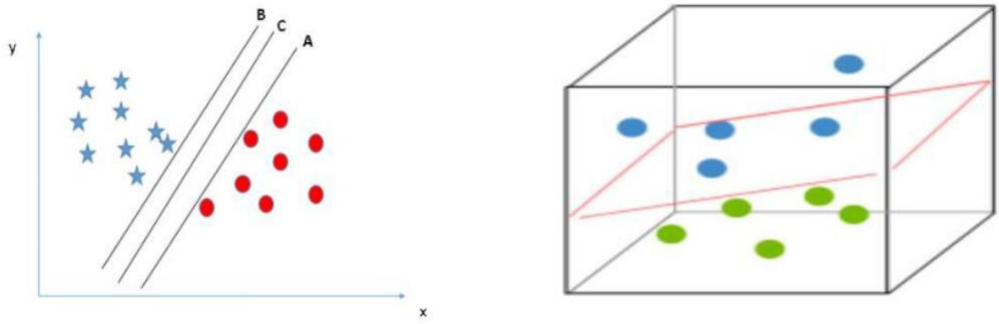


Fig 3.5 Example of SVM

When data are not labelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clusteringalgorithm created applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabelled data, and is one of the most widely used clustering algorithms in industrial applications.

SVMs can be used to solve various real world problems:

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labelled training instances in both the standard inductive and transductive settings.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinementschemes after just three to four rounds of relevance feedback. This is also true of imagesegmentation systems, including those using a modified version SVM.
- Hand-written characters can be recognized using SVM.
- The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism

for interpretation of SVM models. Support vector machine weights have also been used to interpret SVM models in the past.

In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. For example:

$$B0 + (B1 * X1) + (B2 * X2) = 0$$

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

The equation for making a prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows: f(x)

$$= B0 + sum(ai * (x,xi))$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

The dot-product is called the kernel and can be re-written as:

$$K(x, xi) = sum(x * xi)$$

Instead of the dot-product, we can use a polynomial kernel, for example:

$$K(x,xi) = 1 + sum(x * xi)^d$$

3.6 K-MEANS ALGORITHM

k-means clustering is a method of vector quantization, originally from signal processing, thatis popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectationmaximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both *k*-means and Gaussian Mixture Modelling.

Additionally, they both use cluster centres to model the data; however, *k*-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

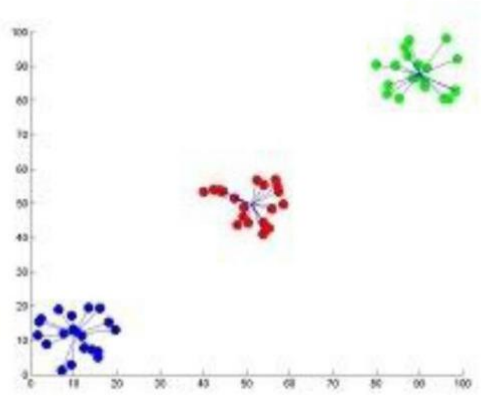


Fig 3.6: Example of K-means algorithm

The algorithm has a loose relationship to the *k*-nearest neighbour classifier, a popular machine learning technique for classification that is often confused with *k*-means because of the *k* in the name. One can apply the 1-nearest neighbour classifier on the cluster centres obtained by *k*-means to classify new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

This algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

3.7 RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Decision trees are a

popular method for various machine learning tasks. Tree learning come closest to meeting the requirements for serving as an off-the-shelf procedure for data mining because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspect able models. However, they are seldom accurate.

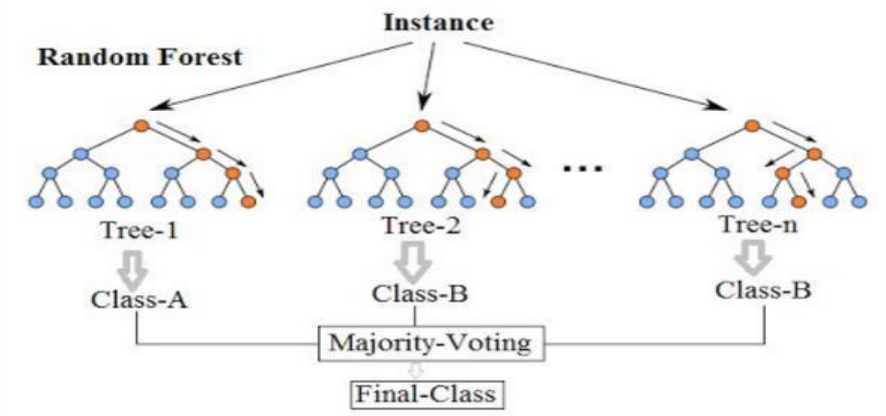


Fig 3.7: Example of Random forest

In particular, trees that are grown very deep tend to learn highly irregular patterns: they over fit their training sets, i.e. have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

3.8 ADVANTAGES OF PROPOSED SYSTEM

1. Relatively scalable and simple.

2. Suitable for datasets with compact clusters that are well-separated.

3. Discovery of arbitrary-shaped clusters with varying size.

Chapter 4

REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

4.1 FUNCTIONAL REQUIREMENTS

The functional requirements for a system describe what the system should do. Examples of functional requirements for a university library system called LIBSYS, used by students and faculty to order books and documents from other libraries.

- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.
- Every order shall be allocated a unique identifier, which the user shall be able to copy to the account's permanent storage area.

The LIBSYS system is a single interface to a range of article databases. It allows users to download copies of published articles in magazines, newspapers and scientific journals. The second example requirement for the library system that refers to 'appropriate viewers' provided by the system. The library system can deliver documents in a range of formats; the intention of this requirement is that viewers for all of these formats should be available. However, the requirement is worded ambiguously; it does not make clear that viewers for each document format should be provided. A developer under schedule pressure might simply provide a text viewer and claim that the requirement had been met. Completeness means that all services required by the user should be defined. Consistency means that requirements should not have contradictory definitions.

4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy.

- Failing to meet a non-functional requirement can mean that the whole system is unusable. For example, if an aircraft system does not meet its reliability requirements, it will not be certified as safe for operation.
- Non-functional requirements constrain the process that should be used to develop the system. Examples of process requirements include a specification of the quality standards that should be used in the process, a specification that the design must be produced with a particular CASE toolset and a description of the process that should be followed.
- Classification of non-functional requirements. You can see from this diagram that the non-functional requirements may come from required characteristics of the software (product requirements), the organization developing the software (organizational requirements) or from external sources.

4.3 SOFTWARE REQUIREMENTS

- Windows 10,Windows 8.1,Windows 7 Service Pack 1,Windows Server 2016,Windows Server 2012 R2,Windows Server 2012.
- Any Intel or AMD x86-64 processor.
- Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support.
- No specific graphics card is required.

4.4 HARDWARE REQUIREMENTS

- 2 GB of HDD space for MATLAB only, 4-6 GB for a typical installation.
- An SSD is recommended.
- A full installation of all Math Works products may take up to 22 GB of disk space.
- Minimum 4 GB RAM.
- For Polyspace, 4 GB per core is recommended.
- Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.

Chapter 5

SYSTEM DESIGN

5.1 HIGH-LEVEL DESIGN

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. In contrast, low-level design further exposes the logical detailed design of each of these elements for programmers. A high-level design document or HLDD adds the necessary details to the current project description to represent a suitable model for coding. This document includes a high-level architecture diagram depicting the structure of the system, such as the database architecture, application architecture (layers), application flow (navigation), security architecture and technology architecture.

5.1.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).System architecture conveys the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that an architecture description may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

5.1.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data, flows as a unified model. Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM.

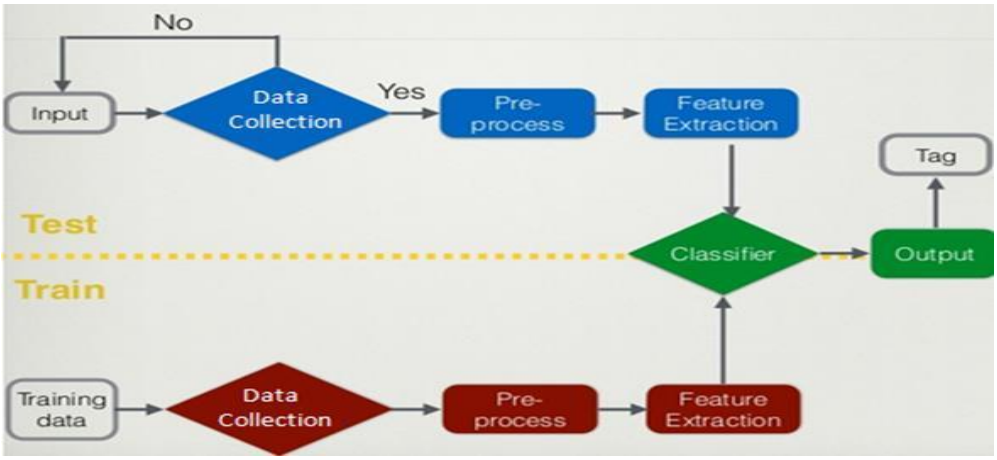


Fig 5.1: Data flow diagram

The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's data flow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model.

5.1.3 FLOWCHART

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

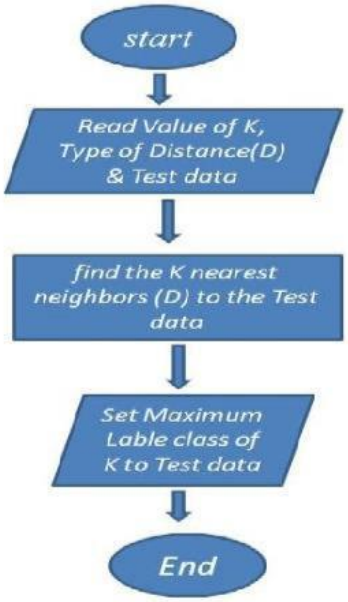


Fig 5.2: Flowchart for k nearest neighbor algorithm

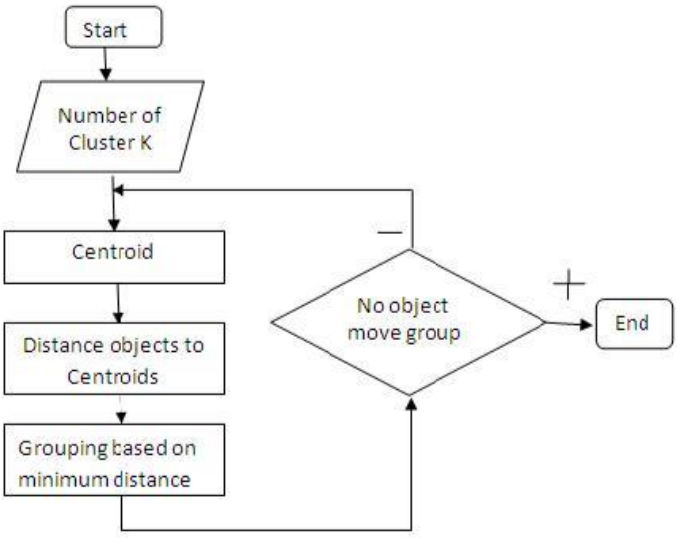


Fig 5.3: Flowchart for k-means algorithm

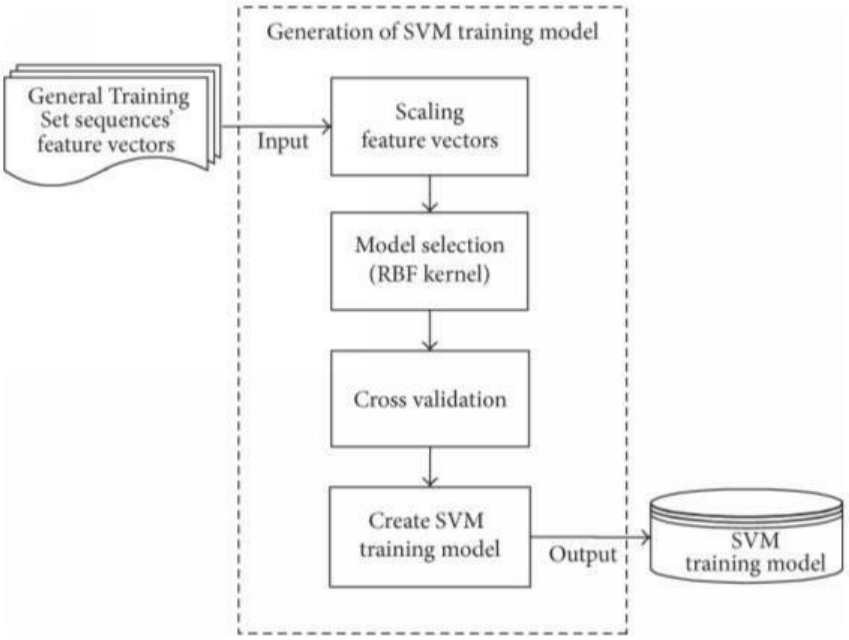


Fig 5.4: Flowchart for SVM

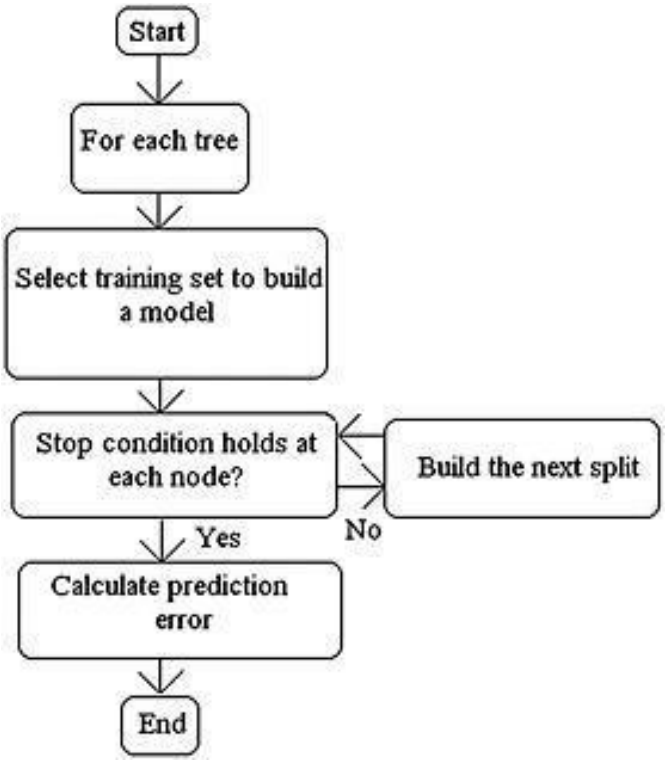


Fig 5.5 Flowchart for Random forest

Chapter 6

IMPLEMENTATION

6.1 ALGORITHMS

In mathematics and computer science, an algorithm is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing and automated reasoning tasks. As an effective method, an algorithm can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function. Starting from an initial state and initial input, the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing output and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input

6.1.1 ALGORITHM FOR NEAREST NEIGHBOUR ALGORITHM

Step 1: start on an arbitrary vertex as current vertex.

Step 2: find out the shortest edge connecting current vertex and an unvisited vertex V.

Step 3: set current vertex to V.

Step 4: mark V as visited.

Step 5: if all the vertices in domain are visited, then terminate.

Step 6: Go to step 2

6.1.2 ALGORITHM FOR K-NEAREST NEIGHBOUR ALGORITHM

Step 1: Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).

Step 2: for i=0 to m:

Calculate Euclidian distance d(arr[i],p).

Step 3: Make set S of K smallest distances obtained. Each of these distances correspond to an

already classified data point.

Step 4: Return the majority label among S.

6.1.3 ALGORITHM FOR K-MEANS

Step 1: Place K points into the space represented by the objects that are being clustered. These

points represent initial group centroids.

Step 2: Assign each object to the group that has the closest centroid.

Step 3: When all objects have been assigned, recalculate the positions of the K centroids.

Step 4: Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of

the objects into groups from which the metric to be minimized can be calculated.

6.1.4 ALGORITHM FOR LINEAR REGRESSION

Step 1: Got a bunch of points in R 2 , {(x i , y i)}.

Step 2: Want to fit a line y = ax + b that describes the trend.

Step 3: We define a cost function that computes the total squared error of our predictions w.r.t.

observed values y i J(a, b) = P(ax i + b − y i) 2 that we want to minimize.

Step 4: See it as a function of a and b: compute both derivatives, force them equal to zero, and

solve for a and b.

Step 5: The coefficients you get give you the minimum squared error. 6. Can do this for specific

points, or in general and find the formulas.

6.1.5 ALGORITHM FOR LOGISTIC REGRESSION

Step 1:Initialize a = h1, .., 1i^T

Step 2: Perform feature scaling on the examples' attributes

Step 3: Repeat until convergence

for each j = 0, .., n:

a 0 j = aj + α P i (y i − ha(x i))x i j

for each j = 0, .., n:

aj = aj

Step 4: Output a

6.1.6 ALGORITHM FOR SUPPORT VECTOR MACHINE

Step 1: candidateSV= { closest pair from opposite classes }

Step 2: while there are violating points do

Step 3: Find a violator

candidateSV= candidateSV S violator
if any p<0 due to addition of c to s then

candidateSV=candidateSV/p

Step 4: repeat until all points are pruned

Step 5: end if

Step 6: end while.

6.1.7 ALGORITHM FOR RANDOM FOREST

Step 1: Randomly select "k" features from total "m" features, where k<<m

Step 2: Among the "k" features, calculate the node "d" using the best split points

Step 3: split the node into daughter nodes using the best split.

Step 4: Repeat 1 to 3 steps until "l" number of nodes has been reached .

Step 5: Build forest by repeating steps 1 to 4 for "n" number of times to create "n" number of Tress.

Chapter 7

CODE

Code Snippet of NEAREST NEIGHBOUR IMPLEMENTATION

```
loadfisheriris

X = meas(51:100,3:4);
X=X';
Y=meas(101:150,3:4);
Y=Y';
P=0.5+(2.9-0.5).*rand(1);

Q=2.5+(7.5-2.5).*rand(1);
R=[Q,P];

R=R';

plot(R(1,1),R(2,1),'g*');

I = nearestneighbour(R, X, 'NumberOfNeighbours', 1)
J=nearestneighbour(R, Y, 'NumberOfNeighbours', 1)
```

Code Snippet of K-NEAREST NEIGHBOUR IMPLEMENTATION

```
loadfisheriris

X = meas(51:150,3:4);
X=X';
P=0.5+(2.9-0.5).*rand(1);

Q=2.5+(7.5-2.5).*rand(1);
R=[Q,P];

R=R';

plot(R(1,1),R(2,1),'g*');

disp(R)

I = nearestneighbour(R, X, 'NumberOfNeighbours', 3)
```

Code Snippet of K-MEANS IMPLEMENTATION

```
loadfisheriris
```

```
F=meas(1:150,3:4);
K     = 3;
KMI   = 20;
CENTS = F( ceil(rand(K,1)*size(F,1)) ,:);
DAL = zeros(size(F,1),K+2);
CV    = '+r+b+g';

for n = 1:KMI
fori = 1:size(F,1)
for j = 1:K
DAL(i,j) = norm(F(i,:) -
CENTS(j,:)); end
      [Distance CN] = min(DAL(i,1:K));
DAL(i,K+1) = CN;
DAL(i,K+2) = Distance;
end
fori = 1:K
      A = (DAL(:,K+1) == i);
CENTS(i,:) = mean(F(A,:));
if sum(isnan(CENTS(:))) ~= 0
         NC = find(isnan(CENTS(:,1)) == 1);
forInd = 1:size(NC,1)
CENTS(NC(Ind),:) = F(randi(size(F,1)),:);
end
end
end
```

Code Snippet of LINEAR REGRESSION IMPLEMENTATION

```
fprintf('Plotting Input Data ...\n')
data = csvread('data.csv');
X = data(:, 1);
y = data(:, 2);
m = length(y);
plotData(X, y);
fprintf('See the plotted graph. Press enter to continue.\n');
pause;

X = [ones(m, 1), data(:,1)];
theta = zeros(2, 1);
iterations = 1500;
alpha = 0.01;
```

Code Snippet of LOGISTIC REGRESSION IMPLEMENTATION

```
data = load('iris.mat');
```

```
 X = data.meas(1:100,[1 2]);
 y = data.species(1:100,: );
X(:,1)=X(:,1)*10;
X(:,2)=X(:,2)*10;
label='setosa';
 y= ismember(y,label);

fprintf(['Plotting data with + indicating (y = 1) examples and o '
...
          'indicating (y = 0) examples.\n']);

plotData(X, y);
```

Code Snippet of SUPPORT VECTOR MACHINE IMPLEMENTATION

```
kernel_function = 'linear';
svm_classifier(meas_1,meas_2,label,one_label,kernel_function);

figure;
kernel_function = 'rbf';
svm_classifier(meas_1,meas_2,label,one_label,kernel_function);

figure;
kernel_function = 'quadratic';
svm_classifier(meas_1,meas_2,label,one_label,kernel_function);

figure;
kernel_function = 'polynomial';
svm_classifier(meas_1,meas_2,label,one_label,kernel_function);

figure;
kernel_function = 'mlp';
svm_classifier(meas_1,meas_2,label,one_label,kernel_function);
```

Code Snippet of RANDOM FOREST IMPLEMENTATION

```
BaggedEnsemble =
TreeBagger(iNumBags,X,Y,'OOBPred','On','Method',str_method)

oobErrorBaggedEnsemble = oobError(BaggedEnsemble);
figID = figure;
plot(oobErrorBaggedEnsemble)
xlabel 'Number of grown trees';
ylabel 'Out-of-bag classification error';
print(figID, '-dpdf',
sprintf('randomforest_errorplot_%s.pdf', date));
```

Chapter 8

Performance Analysis

This project has the goal of finding the best classification method and select the best voted classifier i.e. classifier that has most accuracy percentage. Each dataset are divided into 3 categories. 60% of data is for training, 20% data is for cross validation and rest 20% data is for testing. Table 8 shows the comparison of different algorithms with respect to accuracy.

Table 8: Performance of Machine Learning Algorithms

Dataset / Algorithm	Finger Vein	Fisher Iris	Agar wood Oil Composition
SVM	89.24 %	100 %	100 %
Linear Regression	-	61.26 %	51.4%
Logistic Regression	87.65 %	100 %	89 %
KNN	89.92 %	85.93%	78.95 %
NN	75.53 %	79.81 %	52.46 %

The loaded dataset is preprocessed. After preprocessing of dataset, we tried to apply different machine learning algorithms and analyzed the performance of each algorithm. We evaluated each algorithm on 10 fold cross validation method to test re-sampling of data by executing it 10 times for each algorithm which is repeated for 3 times.

We study the performance of various Machine learning algorithms along K-Nearest Neighbors (NN) and Support Vector Machine (SVM) classification techniques, Linear Regression, Logistic Regression, and Nearest Neighbors.

Table 8 summarizes the results obtained using the aforementioned training and/or classification techniques. As Table 8 shows, using the Support Vector Machine (SVM) algorithm for classification yields better results than all other used techniques.

However, there seems to be somewhat of a consensus as far as the overall detection accuracy between the classifiers Logistic Regression as well SVM as the accuracies are quite close. The peak accuracy of SVM and Logistic Regression gets into 100%, for KNN it reaches 89.92%, and 79.81 % for Nearest Neighbors algorithm.

Chapter 9

RESULT ANALYSIS

9.1 SNAPSHOTS

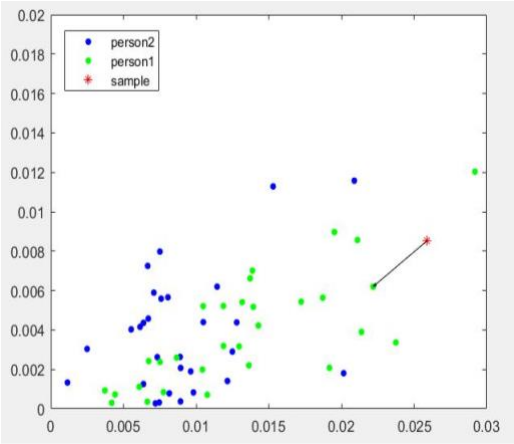


Fig 9.1 Nearest Neighbor for Finger vein.

The above NN algorithm has predicted the sample as **Person1** which being the nearest neighbor compared to others. If two or more neighbors are at equal distance from the sample, the algorithm randomly selects one point. Hence a reduced efficiency has been recorded for Nearest Neighbor algorithm.

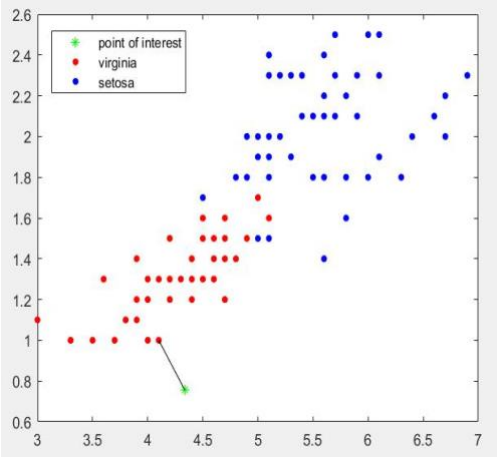


Fig 9.2 Nearest Neighbor for Fisher iris.

The above NN algorithm has predicted the point of interest as **Virginia** which being the nearest neighbor compared to others. If two or more neighbors are at equal distance from the point of interest, the algorithm randomly selects one point. Hence a reduced efficiency has been recorded for Nearest Neighbor algorithm.

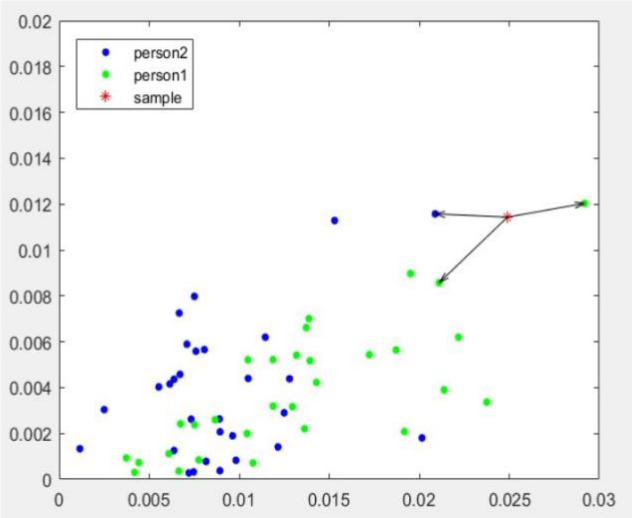


Fig 9.3 K-Nearest Neighbor for Finger vein.

The above KNN algorithm has predicted the sample as **Person1**. Even though it has predicted one point as Person 2, a voting method is used so as to determine the winner. Hence odd numbers of neighbors are considered so as to resolve the conflict of equal voting. Hence a reduced efficiency has been recorded for K-Nearest Neighbor algorithm.

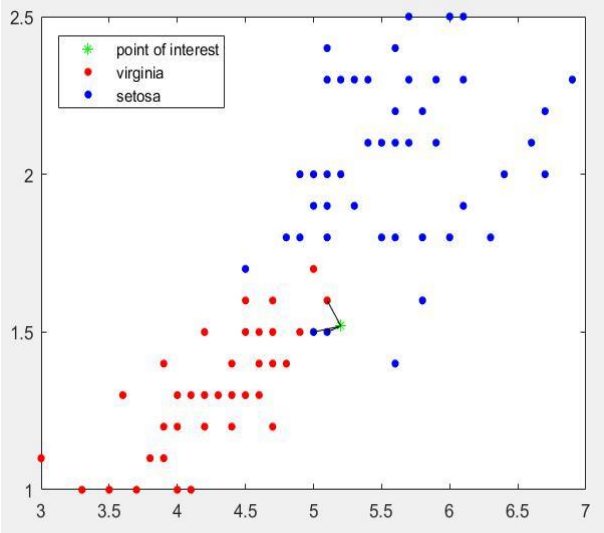


Fig 9.4 K-Nearest Neighbor for Fisher iris.

The above KNN algorithm has predicted the sample as **Setosa**. Even though it has predicted one point as Virginia, a voting method is used so as to determine the winner. Hence odd numbers of neighbors are considered so as to resolve the conflict of equal voting. Hence a reduced efficiency has been recorded for K-Nearest Neighbor algorithm.

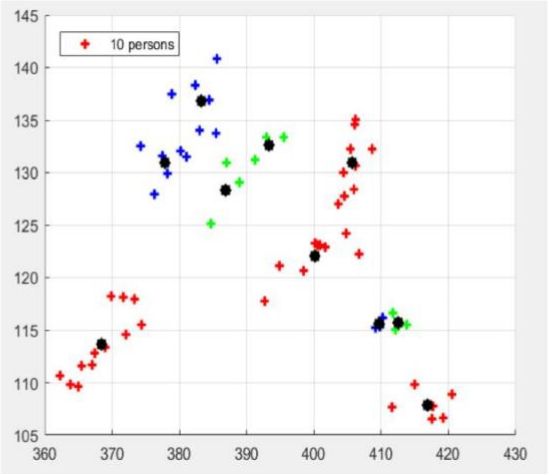


Fig 9.5 K-Means for Finger Vein.

The above snapshot has clustered **10 persons** into groups based on their finger vein characteristics namely **Major axis length and Minor axis length**. Each cluster contains a centroid which marks for differentiating persons. Higher the clustering iteration, better results but at the cost of elapsed run time. Hence according to the user requirements, elapsed time and clustering efficiency must be balanced.

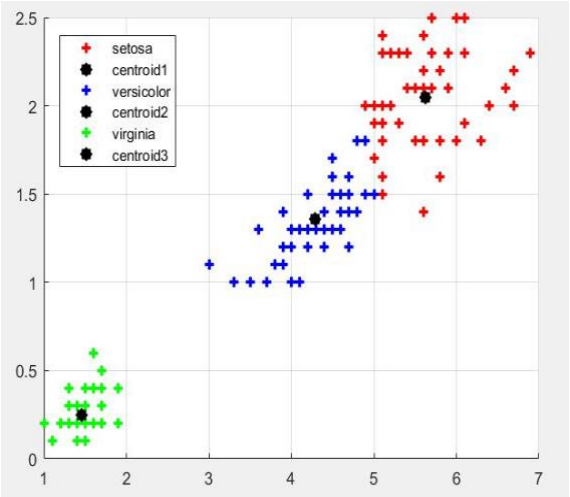


Fig 9.6 K-Means for fisher iris.

The above snapshot has clustered 3 flowers-**Virginia, Setosa, Versicolor** into groups based on Sepal width and Sepal length characteristics. Each cluster contains a centroid which marks for differentiating the flowers. Higher the clustering iteration, better results but at the cost of elapsed run time. Hence according to the user requirements, elapsed time and clustering efficiency must be balanced. Fisher iris data set has recorded a good efficiency as the classes of flowers are well distinguished.

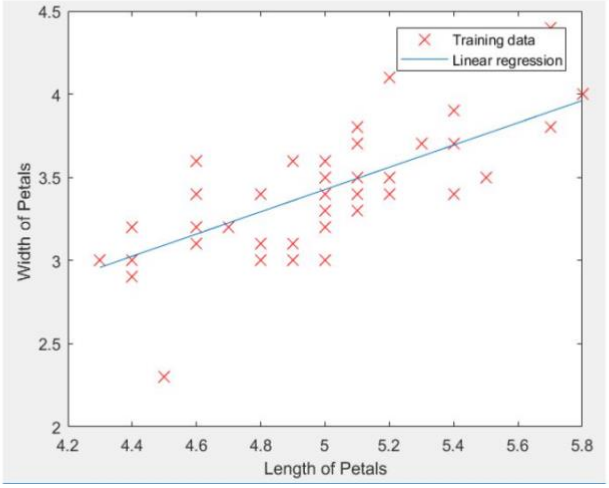


Fig 9.7 Linear Regression for Fisher iris.

The above graph depicts the prediction of **Setosa** flower based on **Petal width and Petal length**. Since Linear Regression predicts the future value based on the current scenario, any sample flower which is closer to the linear line has more close features of Setosa. This line is derived from the basic line equation Y=MX+C.

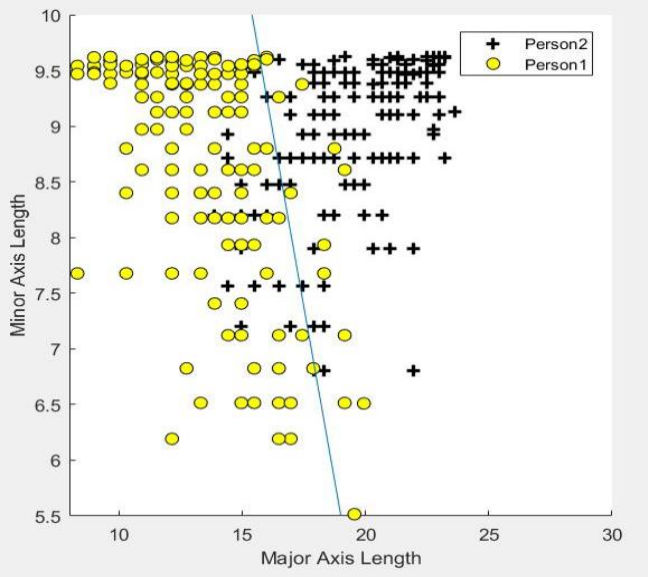


Fig 9.8 Logistic Regression for Finger vein.

The above graph depicts the classification of **Peron1 or Person2** based on their **Major Axis Length and Minor Axis length**. Logistic Regression predicts the outcome as Zero or One (True/False), thus the line is predicted in such way that each side of the line contains a balance of maximum points. Closer to the predicted line, lower is the classification efficiency.

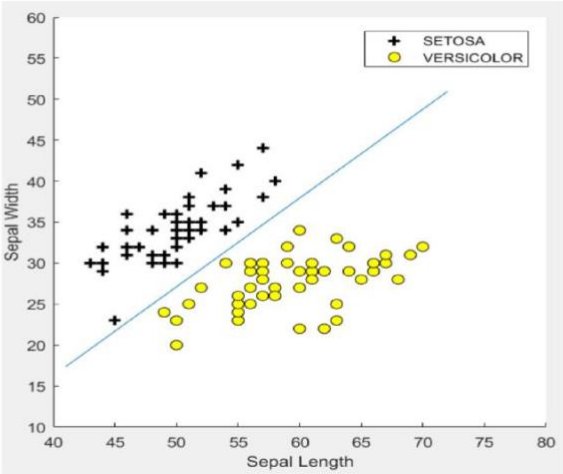


Fig 9.9 Logistic Regression for Fisher iris.

The above graph depicts the classification **of Setosa or Versicolor** based on their **Sepal Length and width**. Linear Regression predicts the outcome as Zero or One (True/False), thus the line is predicted in such way that each side of the line contains a balance of maximum points. Closer to the predicted line, lower is the classification efficiency.

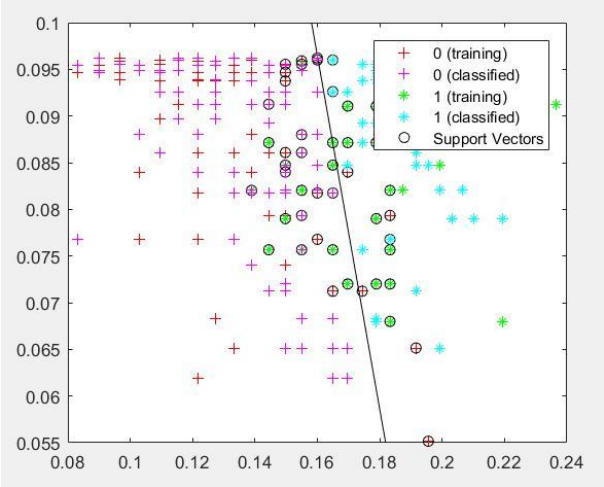


Fig 9.10 SVM for Finger vein.

Support Vector Machines- a supervised learning method has classified **Person1 & Person2** based **Major Axis & Minor Axis length** using a **linear kernel**. Here the support vectors are calculated (encircled) which are at max distance with respect to each other. This distance is directly proportional to the classification efficiency. As seen from the graph, some of the support vectors are on the line, which suggests finger vein has a lesser classification efficiency compared to the other data-sets. But in contrast SVM has performed the best when compared to the other algorithms.

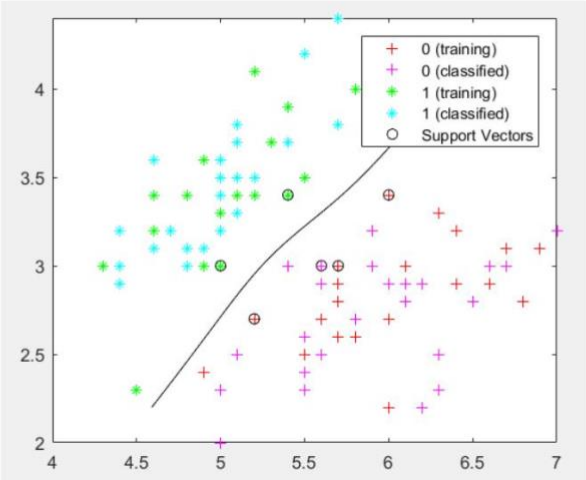


Fig 9.11 SVM for Fisher iris.

Support Vector Machines- a supervised learning method has classified **Setosa & Versicolor** based Sepal width & length using **RBF Kernel**. Here the support vectors are calculated (encircled) which are at max distance with respect to each other. This distance is directly proportional to the classification efficiency. As seen from the graph, the support vectors are clearly at a distance, which suggests fisher iris has a greater classification efficiency compared to the other data-sets.

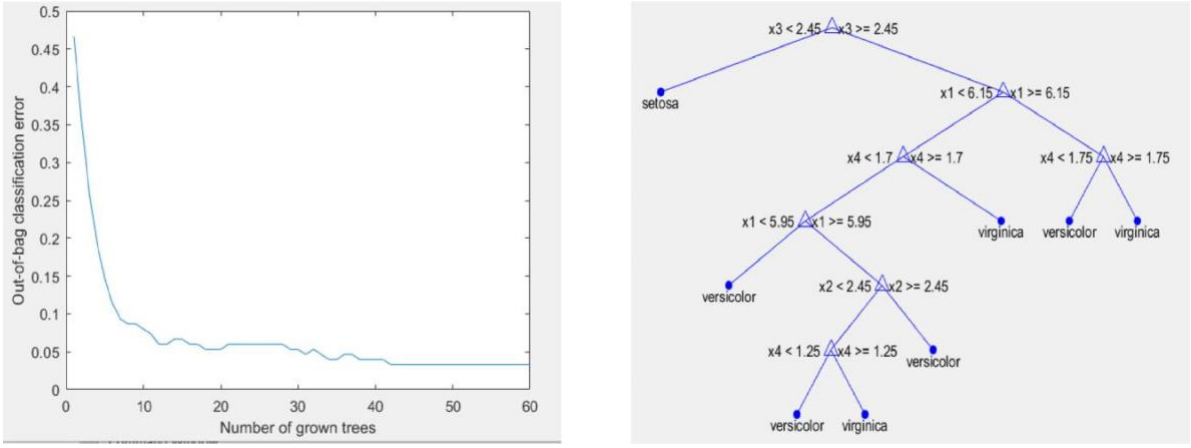
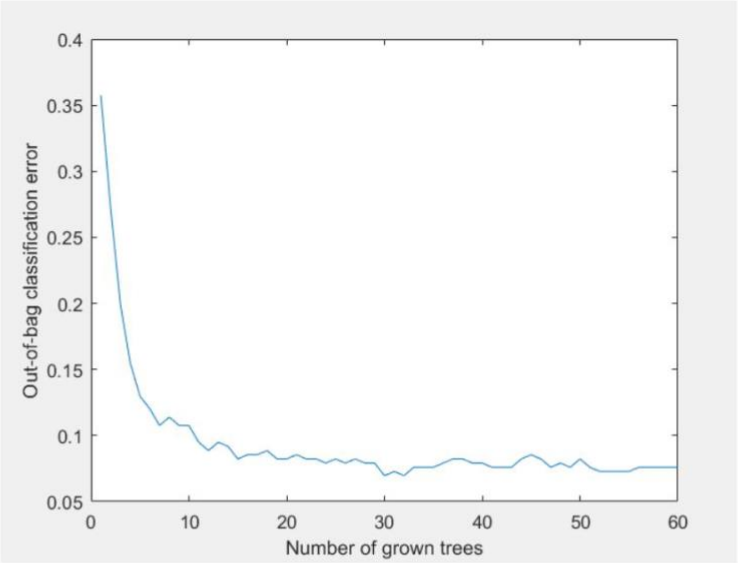


Fig 9.12 Random forest for Fisher iris.

The above snapshots show the classification tree viewer and the efficiency based on the number of grown tress. A voting method is used to predict the flower. In the above snapshots **Versicolor** is the predicted flower since it contains the max votes. As the number grown trees increases, the classification efficiency gets better. Thus Random forests classification efficiency increases with increase in the number of a decision trees (equivalent to number of data samples).



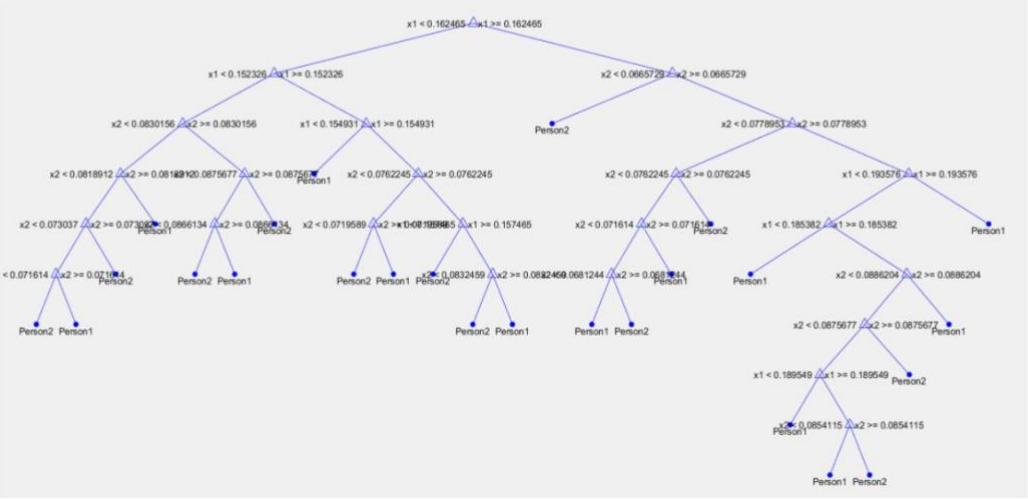


Fig 9.13 Random forest for Finger Vein.

The above snapshots show the classification tree viewer and the efficiency based on the number of grown tress. A voting method is used to predict the person. In the above snapshots **Person1** is predicted since it contains the max votes. As the number grown trees increases, the classification efficiency gets better. Thus Random forests classification efficiency increases with increase in the number of a decision trees (equivalent to number of data samples).

Chapter 10

CONCLUSION& FUTURE WORK

Algorithms implemented in this project provides a basis for machine learning, whose test cases and outcomes can be used for further research and optimizing the algorithms. Experiments were carried out using different real time datasets and UCI data sets to analyze and compare their classification performance. Feature extraction concept used here can be implemented in the field of image processing which will result in adaptable and resilient image processing technology. Machine learning techniques are being widely used to solve real-world problems by storing, manipulating, extracting and retrieving data from large sources. Supervised machine learning techniques have been widely adopted however these techniques prove to be very expensive when the systems are implemented over wide range of data. This is due to the fact that significant amount of effort and cost is involved because of obtaining large labeled data sets. Thus active learning provides a way to reduce the labeling costs by labeling only the most useful instances for learning.

In future we plan to extend our comparisons and weigh both accuracy and comprehensibility. These comparisons should include systems from other non-symbolic fields. A multimodal scheme where several approach systems will be combined to form a unit identification system can be studied. As a future work, the proposed approach will be tested on other datasets. Focus can be given to integration and comparison between the ROC (Receiver Operating Characteristics) curves. The proposed approach contains several phases, namely, acquisition, enhancement, feature extraction and classification. In field of active learning future work involves combining active learning with a subfield of machine learning called transfer learning. It is applicable in situations when we have a training set available for one problem but not for another similar problem. It involves transferring knowledge from one domain to another to speed up learning.

REFERENCES

[1] Miura, N., Nagasaka, A.: Feature extraction of finger-vein pattern based on repeated line tracking and its application to personal identification. Machine Vision and Applications, 15(4): 194-203, 2004.

[2] Hashimoto, J.: Finger vein authentication technology and its future. In Proceedings of the VLSI Symposium on Circuits, PP: 5-8, Honolulu, HI, 2006.

[3] Kono, M., Ueki, H., Umemura, S.: A new method for the identification of individuals by using of vein pattern matching of a finger. In Proceedings of the 5th symposium on pattern measurement, PP: 9-12, Yamaguchi, Japan, 2000.

[4] Yanagawa, T., Aoki, S., Ohyama, T.: Human finger vein images are diverse and its patterns are useful for personal identification, MHF Preprint Series, Kyushu University, pages 1–7, 2007.

[5] Miura, N., Nagasaka, A., Miyatake, T.: Extraction of finger-vein patterns using maximum curvature points in image profiles. IEICE Transactions on Information and Systems, E90-D (8): 1185–1194, 2007.

[6] Kumar, A., Zhou, Y.B.: Human identification using finger images. IEEE Transactions on Image Process, 21(4): 2228–2244, 2012.

[7] Yang, L., Yang, G.P., Yin, Y.L., Xi, X.M.: Exploring soft biometric trait with finger vein recognition. Neurocomputing, 135: 218-228, 2014.

[8] Ton, B.T., and Raymond N.V.: A high quality finger vascular pattern dataset collected using a custom designed capturing device. In Proceedings of International Conference on Biometrics, PP: 1-5, Madrid, Spain, 2013.

[9] Lee, E. C., Jung, H., Kim, D.: New finger biometric method using near infrared imaging. Sensors, 11 (3): 2319–2333, 2011.

[10] Yang, G.P., Xi, X.M., Yin, Y.L.: Finger vein recognition based on a personalized best bit map. Sensors 12 (2): 1738-1757, 2012.

[11] Yin, Y.L., Liu, L.L., Sun, X.W.: SDUMLA-HMT: a multimodal biometric database. The 6th Chinese Conference on Biometric Recognition, LNCS 7098, pp. 260-268, Beijing, China, 2011.

[12] Yang, W.M., Huang, X.L., Zhou, F., Liao, Q.M.: Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion.

[13] Lu, Y., Xie, S.J., Yoon, S., Wang, Z., Park, D.S.: An Available Database for the Research of Finger Vein Recognition. In Proceedings of International Congress on Image and Signal Processing, PP: 386-392, Hangzhou, China, 2013.

[14] Song, W., Kim, T., Kim, H.C., Choi, J.H., Kong, H. J., Lee, S.R.: A finger-vein verification system using mean curvature. Pattern Recognition Letter, 32 (11):1541-1547, 2011. 15. Qin, H.F., Yu, C.B., Qin, L.: Region growth–based feature extraction method for finger-vein recognition. Optical Engineering, 50(5): 057208-057208, 2011.

[15] Liu, T., Xie, J. B., Yan, W., Li, P.Q., Lu, H.Z.: An algorithm for finger-vein segmentation based on modified repeated line tracking. The Imaging Science Journal, 61(6): 491-502, 2013.