A Major Project Report

On

Disease Prediction Using Machine Learning

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

(Computer Science Engineering and Technology)

Submitted by:

Satyam Singh Virat (180970101042)

Sapna Rawat (180970101041)

Under the guidance of

Mr. Manish Kumar

(Assistant Professor)

in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THDC INSTITUTE OF HYDROPOWER ENGINEERING AND TECHNOLOGY,
TEHRI, UTTARAKHAND, INDIA

(Uttarakhand Technical University, Dehradun)

(2018-2022)

CERTIFICATE

I hereby certify that the work which is being presented in the project entitled "**Disease Prediction Using Machine Learning"** in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** and submitted in Department of CSE of THDC Institute of Hydropower Engineering & Technology, Tehri, is an authentic record of our own work carried out under the supervision of Mr. **Manish Kumar**, Assistant Professor, Department of CSE, THDC Institute of Hydropower Engineering & Technology, Tehri.

The matter presented in the report is not submitted by us anywhere for the award of any other degree of this or any other institute.

Satyam Singh Virat (180970101042)

Sapna Rawat (180970101041)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date: **(Mr. Manish Kumar**)

Supervisor

HOD Mr. Vivek Kumar

ACKNOWLEDGEMENT

It gives us a great sense of a pleasure to present the report of the B.Tech project undertaken during B.Tech final year. We owe special debt of gratitude to professor Mr. Manish Kumar, Department of CSE, THDC Institute of Hydropower Engineering & Technology, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only due to his cognizant efforts that our endeavors have seen the light of the day. We would also take the opportunity to acknowledge the contribution of Professor Mr. VIVEK KUMAR to support and assistance during the development of the project.

We would also like to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Satyam Singh Virat (180970101042)

Sapna Rawat (180970101041)

ABSTRACT

The project **"Disease Prediction Using Machine Learning with Python"** is a GUI based desktop application which is developed in Python platform. This is simple and basic level small project for learning purpose. It is developed using Machine Learning with Python and Database Local Storage. The Script is written by Sapna Rawat and Satyam Singh Virat completely in python.

With the help of machine learning algorithm, our application decides whether a patient has a disease or not, based on the symptoms visible on him/her and their medical history. Our application then predicts the disease the patients could be suffering from, by executing 3 different machine learning algorithms.

With the new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum. Therefore, we used it in our project for disease prediction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Full-forms
RNN	Recurrent neural networks
DNN	Deep Neural Network
EMH	Efficient Market Hypothesis
ML	Machine Learning
GRNN	General Regression Neural Network
API	Application Programming Interface
SVM	Support Vector Machine
ANN	Artificial Neural Network
LSTM	Long Short Term Memory
ICA	Independent Component Analysis
URL	Uniform Resource Locator
KNN	K-Nearest Neighbor
RMS	Root Mean Square
SMA	Simple Moving Average
WMA	Weighted Moving Average
SLP	Single Layer Perception
MLP	Multi Layer Perception
MSE	Mean Square Error
MAPE	Mean Absolute Percentage Error
CAD	Computer-aided diagnosis
DT	Decision tree
FH	Family history of cancer

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

[1] The main objective of this project is to predict the diseases a patient could possibly be having based on the symptoms in order to make more informed and accurate decisions to handle the disease before it reaches it critical state. Diseases predictor aims to determine the possible development of diseases in a patient. The accurate prediction of diseases will lead to proper treatment at the right time.

[2] Machine learning itself employs different models to make prediction easier and authentic. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Dataset is a vital part of machine learning. In this project, supervised machine learning is employed on a dataset obtained from AIIMS Delhi. This dataset comprises of following many variables: itching, skin rash, nodal skin eruptions, continuous sneezing, shivering, chills, joint pain, stomach pain, acidity, ulcers on tongue, muscle wasting, vomiting, burning micturition, spotting urination, fatigue, weight gain, anxiety, cold hands and feet, mood swings, weight loss, restlessness, lethargy, patches in throat, irregular sugar level, cough, high fever, sunken eyes, breathlessness, sweating, dehydration, indigestion, headache, yellowish skin, dark urine, nausea, loss of appetite, pain behind the eyes, back pain, constipation, abdominal pain, diarrhea, mild fever, yellow urine, yellowing of eyes, acute liver failure and many more such symptoms. The model is then tested on the test data.

1.2 INTRODUCTION TO MACHINE LEARNING

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy[3]. The healthcare industry produces large amounts of health-are data daily that can be used to extract information for predicting disease that can happen to a patient in future while using the treatment history and health data. This hidden information in the healthcare data will be later used for affective decision making for patient's health. Also, this areas need improvement by using the informative data in healthcare.

One such implementation of machine learning algorithms is in the field of healthcare. Medical facilities need to be advanced so that better decisions for patient diagnosis and treatment options can be made. Machine learning in healthcare aids the humans to process huge and complex medical datasets and then analyze them into clinical insights. This then can further be used by physicians in providing medical care. Hence machine learning when implemented in healthcare can leads to increased patient satisfaction. The Decision tree, Random Forest and Naive Bayes algorithm is used to predict diseases using patient treatment history and health data.[4].

[3]Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics.

[4] Spot-on accuracy may not be practical but sometimes even simple linear models can be surprisingly close. In this project, we'll train a regression model using historic pricing data and technical indicators to make predictions on possible diseases. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them. Decision tree, random forest and naive bayes models are engaged for this conjecture separately. Error calculations are done by taking mean square error root mean square error. All the programming is done in python. Python libraries and modules like scikit-learn, numpy, pandas, seaborn and datetime are used for various operations.

1.3 TRADITIONAL APPROACH FOR DISEASE PREDICTION

[5] Prediction using traditional disease risk model usually involves a machine learning and supervised learning algorithm which uses training data with the labels for the training of the models. High-risk and Low-risk patient classification is done in groups test sets. But these models are only valuable in clinical situations and are widely studied. A system for sustainable health monitoring using smart clothing by Chen et.al. He thoroughly studied heterogeneous systems and was able to achieve the best results for cost minimization on the tree and simple path cases for heterogeneous systems.

[5] The information of patient's statistics, test results, and disease history is recorded in EHR which enables to identify potential data-centric solutions which reduce the cost of medical case studies. Bates et al. propose six applications of big data in the healthcare field. Existing systems can predict the diseases but not the subtype of diseases. It fails to predict the condition of people.

1.4 MACHINE LEARNING

[6] Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum.

[7] Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that take patterns out of raw data by using an algorithm or method. The key focus of ML is to allow computer systems to learn from experience without being explicitly programmed or human intervention.

[8] It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, computer vision and stock analysis where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

[8] A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way

that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

[7] Tom Mitchell in his book Machine Learning provides a definition in the opening line of the preface:
The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

In his introduction he provides a short formalism that you'll see much repeated:

[7] "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

[7] The above definition is basically focusing on three parameters, also the main components of any learning algorithm, namely Task (T), Performance (P) and experience (E). In this context, we can simplify this definition as –

ML is a field of AI consisting of learning algorithms that –

- Improve their performance (P)
- At executing some task (T)
- Over time with experience (E)

Based on the above, the following figure 1.1 [2] represents a Machine Learning Model –

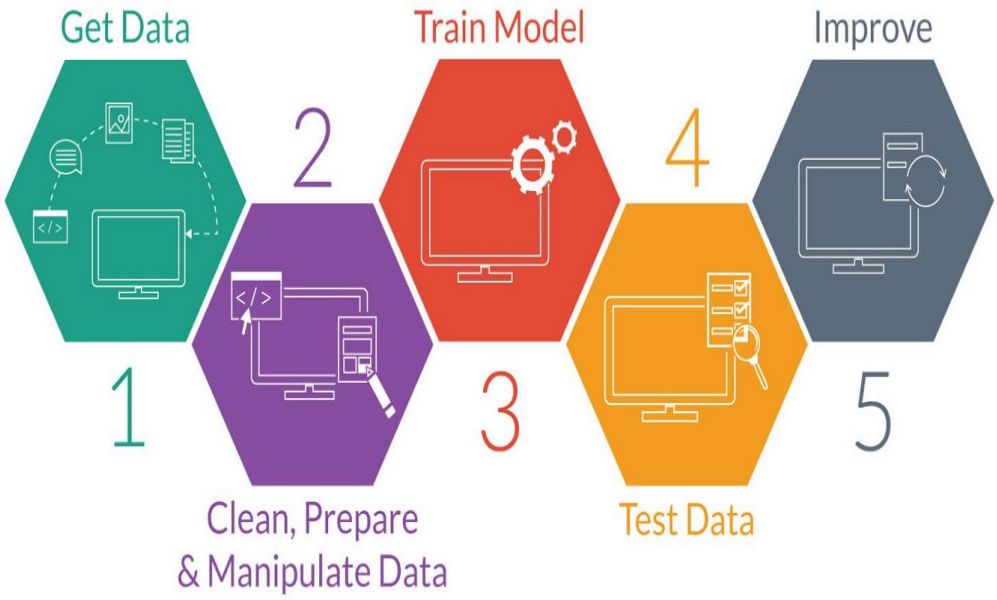


Fig1.1 Machine Learning Model

[9] Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Disease Prediction is one of the application of machine learning.

[10] Classical machine learning is often categorised by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm chosen depends on what type of data we want to predict. Here we have used supervised learning approach.

1.4.1 SUPERVISED MACHINE LEARNING

[11] Known as supervised machine learning, is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, support vector machine (SVM).

Supervised learning is when we teach or train the machine using data that is well labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data. For instance**,** suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:

- If the shape of the object is rounded and has a depression at the top, is red in colour, then it will be labelled as –Apple.
- If the shape of the object is a long curving cylinder having Green-Yellow colour, then it will be labelled as –Banana.

[12]**To solve a given problem of supervised learning, one has to perform the following steps:**

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set.
2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support-vector machines or decision trees.
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.

6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

[11] Supervised learning is classified into two categories of algorithms:

- Classification: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.
- Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
- In this project we’ll be using linear regression imported from sklearn library.

1.4.2 LIBRARIES USED

1.4.2.1 SCIKIT LEARN

[13] Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. [14] The scikit-learn project started as scikits.learn, a Google Summer of Code project by French data scientist David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from the French Institute for Research in Computer Science and Automation in Rocquencourt, France, took leadership of the project and made the first public release on February the 1st 2010. [14] Of the various scikits, scikit-learn as well as scikit-image were described as "well-maintained and popular" in November 2012. Scikit-learn is one of the most popular machine learning libraries on GitHub. Scikit-learn is largely written in Python, and use NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

1.4.2.2 PANDAS

[15] Pandas is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users. Pandas generally provide two data structures for manipulating data. They are:
Series: is a one-dimensional labelled array capable of holding data of any type (integer, string, float, python objects, etc.).

Dataframe: is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes (rows and columns).
[16] Key features of pandas are-

- It has a fast and efficient DataFrame object with the default and customized indexing.
- Used for reshaping and pivoting of the data sets.
- Group by data for aggregations and transformations.
- It is used for data alignment and integration of the missing data.
- Provide the functionality of Time Series.
- Process a variety of data sets in different formats like matrix data, tabular heterogeneous, time series.
- Handle multiple operations of the data sets such as subsetting, slicing, filtering, groupBy, re-ordering, and re-shaping.
- It integrates with the other libraries such as SciPy, and scikit-learn.
- Provides fast performance, and if you want to speed it, even more, you can use the Cython.

1.4.2.3 SEABORN

[17] Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.
[18] Plots are generally used to make visualization of the relationships between the given variables. These variables can either be a category like a group, division, or class or can be completely numerical variables. There are various different categories of plots that we can create using the seaborn library.
[19] In the seaborn library, the plot that we create is divided into the following various categories:

- **Distribution plots:** This type of plot is used for examining both types of distributions, i.e., univariate and bivariate distribution.
- **Relational plots:** This type of plot is used to understand the relation between the two given variables.
- **Regression plots:** Regression plots in the seaborn library are primarily intended to add an additional visual guide that will help to emphasize dataset patterns during the analysis of exploratory data.
- **Categorical plots:** The categorical plots are used to deals with categories of variables and how we can visualize them.
- **Multi-plot grids:** The multi-plot grids are also a type of plot that is a useful approach is to draw multiple instances for the same plot with different subsets of a single dataset.
- **Matrix plots:** The matrix plots are a type of arrays of the scatterplots.

1.4.2.4 NUMPY

[20] NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy

was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
[20] The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important. [21] Numpy arrays are written mostly in C language. Being written in C, the NumPy arrays are stored in contiguous memory locations which makes them accessible and easier to manipulate. This means that you can get the performance level of a C code with the ease of writing a python program.

1.4.3 Challenges of machine learning

As machine learning technology advances, it has certainly made our lives easier. However, implementing machine learning within businesses has also raised a number of ethical concerns surrounding AI technologies. Some of these include-

Technological singularity

While this topic garners a lot of public attention, many researchers are not concerned with the idea of AI surpassing human intelligence in the near or immediate future. This is also referred to as super intelligence, which Nick Bostrum defines as “any intellect that vastly outperforms the best human brains in practically every field, including scientific creativity, general wisdom, and social skills.” Despite the fact that Strong AI and super intelligence is not imminent in society, the idea of it raises some interesting questions as we consider the use of autonomous systems, like self-driving cars

Privacy

Privacy tends to be discussed in the context of data privacy, data protection and data security, and these concerns have allowed policymakers to make more strides here in recent years. As a result, investments within security have become an increasing priority for businesses as they seek to eliminate any vulnerabilities and opportunities for surveillance, hacking, and cyber-attacks.

Bias and discrimination

Instances of bias and discrimination across a number of intelligent systems have raised many ethical questions regarding the use of artificial intelligence. Bias and discrimination aren’t limited to the human resources function either; it can be found in a number of applications from facial recognition software to social media algorithms. As businesses become more aware of the risks with AI, they’ve also become more active this discussion around AI ethics and values.

Accountability

Since there isn't significant legislation to regulate AI practices, there is no real enforcement mechanism to ensure that ethical AI is practiced. The current incentives for companies to adhere to these guidelines are the negative repercussions of an unethical AI system to the bottom line. To fill the gap, ethical frameworks have emerged as part of a collaboration between ethicist and researchers to govern the construction and distribution of AI models within society. [22]

All the programming is done in python using Jupyter notebook. [23] The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

Chapter 2

Literature Review

Mehrbakhsh Nilashi1,2, Othman Ibrahim1 & Ali Ahani1 [5] emphasized that the main medical challenge is to correctly recognize the PD affected subjects at the early stage. The early diagnosis can assist the patients improve and maintain their quality of life19. However, due to symptom overlap with other diseases PD may be difficult to diagnose accurately, especially at the early stages of the illness20. In addition, traditional diagnosis of PD involves a clinician taking a neurological history of the patient and observing motor skills in various situations. Since there is no definitive laboratory test to diagnose PD, diagnosis is often difficult, particularly in the early stages when motor effects are not yet severe. Monitoring progression of the disease over time requires repeated clinic visits by the patient. There is no cure, but pharmacological treatment to manage the condition includes dopaminergic drugs.

Antonio Bazarra-Fernandeza [24] proposed conditions that would affect the risk of fracture would be: the strength of the bone, the weight and height of the person, the frequency of falls, the nature of the falls (on the side, forwards or whatever) the state of neuromuscular coordination, whether they drink lots of alcohol, the thickness of the muscle, the amount of fat and clothing in the relevant area, and so on.

Physicians rely heavily on BMD T-score to decide on osteoporosis treatment initiation. Although guidelines suggest using clinical risk factors to guide decision making, we did not see evidence of this because sodium fluoride increase BMD but not decrease fractures and low BMD does not mean decrease of fractures. More explicit methods of reporting fracture risk may help physicians select patients who are likely to derive the largest benefit from osteoporosis treatment.

Wei Yu*, Tiebin Liu, Rodolfo Valdez, Marta Gwinn, Muin J Khoury [25] present a potentially useful alternative approach based on support vector machine (SVM) techniques to classify persons with and without common diseases. We illustrate the method to detect persons with diabetes and pre-diabetes in a cross-sectional representative sample of the U.S. population. Methods: We used data from the 1999-2004 National Health and Nutrition Examination Survey (NHANES) to develop and validate SVM models for two classification schemes: Classification Scheme I (diagnosed or undiagnosed diabetes vs. pre-diabetes or no diabetes) and Classification Scheme II (undiagnosed diabetes or pre-diabetes vs. no diabetes). The SVM models were used to select sets of variables that would yield the best classification of individuals into these diabetes categories.

They proposed A supervised machine learning method, the support vector machine (SVM) algorithm has demonstrated high performance in solving classification problems in many biomedical fields, especially in bioinformatics. In contrast to logistic regression, which depends on a pre-determined model to predict the occurrence or not of a binary event by fitting data to a logistic curve, SVM discriminates between two classes by generating a hyperplane that optimally

separates classes after the input data have been transformed mathematically into a high-dimensional space.

Mohammad R. Mohebian, Hamid R. Marateb, Marjan Mansourian, Miguel Angel Mañanas, Fariborz Mokarian [26] proposed the use of Computer-aided diagnosis (CAD) is using computers and software to interpret medical information. The purpose of CAD is to improve the diagnosis accuracy. In fact, CAD is used as a second opinion by the physicians to make the final diagnosis decision. An important issue is whether we can optimize the treatments to increase the therapeutic efficacy. In fact, 5-year recurrence-free survival is an important treatment quality measure. In principle, it is possible to predict 5-year cancer recurrence using clinico-pathologic characteristics of cancer patients. Such a prediction could be used by doctors to make proper treatment plan to considerably prolong patient life.

The prediction systems were used for cancer diagnosis in the literature. However, there are few studies focusing on cancer prognosis (including recurrence or survival analysis). Since the focus of the current study is prediction of cancer recurrence, the literature review on cancer recurrence prediction models is provided. Meanwhile, the table of available methods was provided in the Supplementary material S1.

Zeng. suggested a mixture classification model containing a two-layer structure called mixture of rough set and support vector machine (SVM) for breast cancer prognosis with the average accuracy of 91%.

Truyen Tran1,2, Wei Luo1, Dinh Phung1, Sunil Gupta1, Santu Rana1, Richard Lee Kennedy3, Ann Larkins4 and Svetha Venkatesh1 [27] proposed that the feature engineering is a time consuming component of predictive modeling. We propose a versatile platform to automatically extract features for risk prediction, based on a pre-defined and extensible entity schema. The extraction is independent of disease type or risk prediction task. We contrast auto-extracted features to baselines generated from the Elixhauser comorbidities.

Hospital medical records was transformed to event sequences, to which filters were applied to extract feature sets capturing diversity in temporal scales and data types. The features were evaluated on a readmission prediction task, comparing with baseline feature sets generated from the Elixhauser comorbidities. The prediction model was through logistic regression with elastic net regularization. Predictions horizons of 1, 2, 3, 6, 12 months were considered for four diverse diseases: diabetes, COPD, mental disorders and pneumonia, with derivation and validation cohorts defined on non-overlapping data-collection periods.

For unplanned readmissions, auto-extracted feature set using socio-demographic information and medical records, outperformed baselines derived from the socio-demographic information and Elixhauser comorbidities, over 20 settings (5 prediction horizons over 4 diseases). In particular over 30-day prediction, the AUCs are: COPD —baseline: 0.60 (95% CI: 0.57, 0.63), auto-extracted: 0.67 (0.64, 0.70); diabetes— baseline: 0.60 (0.58, 0.63), auto-extracted: 0.67 (0.64, 0.69); mental disorders— baseline: 0.57 (0.54, 0.60), auto-extracted: 0.69 (0.64,0.70); pneumonia—baseline: 0.61 (0.59, 0.63), auto-extracted: 0.70 (0.67, 0.72).

LiMin Wang1'2 [28] proposed the working mechanisms of 3 classical restricted Bayesian classifiers, namely, NB, TAN and KDB, were analysed and summarised. To retain the properties of global optimisation and high-order dependency representation, the proposed learning algorithm, i.e., flexible ^-dependence Bayesian network (FKBN), applies the greedy search of conditional mutual information space to identify the globally optimal ordering of the attributes and to allow the classifiers to be constructed at arbitrary points (values of K) along the attribute dependence spectrum. This method represents the relationships between different attributes by using a directed acyclic graph (DAG) model. A total of 12 data sets were selected from the SEER database and KRBM repository by 10-fold cross-validation for evaluation purposes. The findings revealed that the FKBN model outperformed NB, TAN and KDB.

A Bayesian classifier can graphically describe the conditional dependency among attributes. The proposed algorithm offers a trade-off between probability estimation and network structure complexity. The direct and indirect relationships between the predictive attributes and class variable should be considered simultaneously to achieve global optimisation and high-order dependency representation. By analysing the DAG inferred from the breast cancer data set of the SEER database we divided the attributes into two subgroups, namely, key attributes that should be considered first for cancer diagnosis and those that are independent of each other but are closely related to key attributes. The statistical analysis results clarify some of the causal relationships implicated in the DAG.

Fatma Patlar Akbulut1*, Erkan Akkur2, Aydin Akan3 and B Siddik Yarman3 [29] purposed the use of decision support systems in health provides improvement in terms of the quality of health and care, early diagnosis of diseases, prevention of person related errors, lowering costs and providing the patients with the optimum treatment. It describes a decision support system proposing the ventilator settings required to be applied in the treatment according to the patients' physiological information. The proposed model has been designed to minimize the possibility of making a mistake and to encourage more efficient use of time in support of the decision making process while the physicians make critical decisions about the patient. Artificial Neural Network (ANN) is implemented in order to calculate frequency, tidal volume, FiO2 outputs, and this classification model has been used for estimation of pressure support/volume support outputs. For the obtainment of the highest performance in both models, different configurations have been tried. Various tests have been realized for training methods, and a number of hidden layers mostly affect factors regarding the performance of ANNs.

Ya-Ju Chang1, Hui-Chun Huang1, Yuan-Yu Hsueh2, Shao-Wei Wang3 [30] proposed Prediction and validation of endothelium damage in vein graft. To monitor the dynamic changes in vein graft restenosis at early stages in living animals, a 30-MHz HFU system was used to scan the detailed vascular structures. The dynamic geometrical changes in the vein graft of the same rat after surgery one, two, and three weeks were acquired by serial scanning of cross-section HFU images at 50-^m intervals, starting from the artery 2 mm proximal to the proximal suture site, through the entire graft vessel, and then to the artery 2 mm distal to the distal suture site.

Increases in echogenicity in ultrasonography were observed on the vessel wall of CAV, thereby suggesting structural changes in the intima. The vessel lumen area, which was determined from the vessel border identified in serial HFU images, became progressively narrowed over the 3-week period after vein graft surgery.

Jacob K Kariuki1*, Eileen M Stuart-Shor1,2+, Suzanne G Leveille1 + and Laura L Hayman1 + [31] proposed to evaluate the performance of existing non-laboratory based CV risk assessment algorithms using the benchmarks for clinically useful CV risk assessment algorithms outlined by Cooney and colleagues. Methods: A literature search to identify non-laboratory based risk prediction algorithms was performed in MEDLINE, CINAHL, Ovid Premier Nursing Journals Plus, and PubMed databases. The identified algorithms were evaluated using the benchmarks for clinically useful cardiovascular risk assessment algorithms outlined by Cooney and colleagues. Both the Gaziano and Framingham non-laboratory based algorithms met most of the criteria outlined by Cooney and colleagues. External validation of the algorithms in diverse samples is needed to ascertain their performance and applicability to different populations and to enhance clinicians' confidence in them.

SENTHILKUMAR MOHAN 1 , CHANDRASEGAR THIRUMALAI1 ,
AND GAUTAM SRIVASTAVA [32] proposed it is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate and many other factors. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. The severity of the disease is classified based on various methods like K-Nearest Neighbor Algorithm (), Decision Trees (), Genetic algorithm (), and Naive Bayes ().
We have also seen decision trees be used in predicting the accuracy of events related to heart disease. Various methods have been used for knowledge abstraction by using known methods of data mining for prediction of heart disease. In this work, numerous readings have been carried out to produce a prediction model using not only distinct techniques but also by relating two or more techniques. These amalgamated new techniques are commonly known as hybrid methods. We introduce neural networks using heart rate time series. This method uses various clinical records for prediction.

CHUNYAN GUO, ZHIQIANG HAN , JIABING ZHANG, AND JIANSHE YU [33] proposed the use of deep learning is a machine learning approach that utilizes multiple neural network layers and vast amounts of data to optimize a host of algorithms for a specific task. The potential for therapeutic development, from discovery to prediction, to decision-making is high in machine learning. Within patient electronic healthcare records, ML can detect patterns of certain diseases and inform clinicians of all abnormalities. Due to several contributory risk factors like diabetes, blood pressure, abnormal pulse rate, high cholesterol, and several other factors, it is critical to Machine Learning Model for image processing determines heart disease. Different techniques were employed to reduce the severity of cardiac disease among humans in the mining of data and neural networks. Neural networks are widely

considered to be the best way to predict diseases such as brain disease and heart disease. The generated results using the ANN which delivers good results in heart disease prediction. Neural network models incorporate not only later probabilities but expected values from several previous techniques. Neural network methods were implemented. The heart data set Cleveland with a neural network is used for all experiments to increase the performance of cardiac diseases.

The main contribution of this paper,

- To evaluate the accuracy in the prediction of heart disease using Recursion Enhance Random Forest with an improved linear model (RFRF-ILM) method has been proposed.

- Designing an Artificial Neural Network with feature selection and backpropagation learning technique for classification of cardiovascular disease.

Dhiraj Dahiwade, Gajanan Patle, Ektaa Meshram [34] proposed the prediction of disease at earlier stage becomes important task. But the accurate prediction on the basis of symptoms becomes too difficult for doctor. The correct prediction of disease is the most challenging task. To overcome this problem data mining plays an important role to predict the disease. Medical science has large amount of data growth per year. Due to increase amount of data growth in medical and healthcare field the accurate analysis on medical data which has been benefits from early patient care. With the help of disease data, data mining finds hidden pattern information in the huge amount of medical data. We proposed general disease prediction based on symptoms of the patient. For the disease prediction, we use K-Nearest Neighbor (KNN) and Convolutional neural network (CNN) machine learning algorithm for accurate prediction of disease. For disease prediction required disease symptoms dataset. In this general disease prediction the living habits of person and checkup information consider for the accurate prediction. The accuracy of general disease prediction by using CNN is 84.5% which is more than KNN algorithm. And the time and the memory requirement is also more in KNN than CNN. After general disease prediction, this system able to gives the risk associated with general disease which is lower risk of general disease or higher.

Chun-XiaoNie et al. [35] proposed the systematic studies of the structure of the financial kNN (k-nearest neighbor) network. First, we use the eigenvalues and eigenvectors of the financial correlation matrix to analyze the structure of the network. We find that the degree is related to the average correlation coefficient, and furthermore, it also has a relationship between the components of the eigenvector corresponding to the maximum eigen value. We apply existing research to confirm that the community structure of the kNN network can be used to cluster financial time series. Finally, empirical studies based on financial markets in three countries show that there is a high correlation between the community structure and dimensions. Therefore, this study shows that the structure of the financial kNN network is related

to the properties of the correlation matrix, and it extracts a meaningful correlation structure.

Sateesh Ambesange, Vijayalaxmi A, Rashmi Uppin, Shruthi Patil, Vilaskumar Patil [36] proposed using a machine learning prediction models, liver diseases can be predicted using those health parameters in early stages. In this work to build the machine-learning model, Indian Liver Patient Dataset (ILPD) hosted at UCI.edu [1] is used, which is based on Indian patient and Random Forest (RF) algorithm is used to predict the disease with different preprocessing techniques. Data set is checked for skewness, outliers and imbalance using univariate and bivariate analysis and then suitable algorithms used to remove outliers and various oversampling and under sampling techniques are used to balance the data. Further refinement of model is done through hyper parameter tuning using grid search and feature selection. The final model provides 100% accuracy and also good score across different metrics.

Tao Ban et al. [37] proposed a new multivariate regression approach for financial time series forecasting based on knowledge shared from referential nearest neighbors. This approach defines a two-tier architecture. In the top tier, the nearest neighbors that bear referential information for a target time series are identified by exploiting the financial correlation from the historical data. Next, the future status of the target financial time series is inferred from heritage of the time series by using a multivariate k-Nearest Neighbor (kNN) regression model exploiting the aggregated knowledge from all relevant referential nearest neighbors. The performance of the proposed multivariate kNN approach is assessed by empirical evaluation on the 9-year S&P 500 stock data. The experimental results show that the proposed approach provides enhanced forecasting accuracy than the referred univariate kNN regression.

Ankita Dewan, Meghna Sharma [38] proposed that many hospitals use hospital information systems to manage their healthcare or patient data. These systems produce huge amounts of data in the form of images, text, charts and numbers. Sadly, this data is rarely used to support the medical decision making. There is a bulk of hidden information in this data that is not yet explored which give rise to an important query of how to make useful information out of the data. So there is necessity of creating an excellent project which will help practitioners predict the heart disease before it occurs. The main objective of this paper is to develop a prototype which can determine and extract unknown knowledge (patterns and relations) related with heart disease from a past heart disease database record. It can solve complicated queries for detecting heart disease and thus assist medical practitioners to make smart clinical decisions which traditional decision support systems were not able to. By providing efficient treatments, it can help to reduce costs of treatment.

Md. Touhidul Islam, Sanjida Reza Rafa, Md. Golam Kibria [39] proposed in this paper, Principal Component Analysis (PCA) has been used to reduce attributes. Apart from a Hybrid genetic algorithm (HGA) with k-means used for final clustering. Typically, the k-means method is using for clustering the data. This type of clustering can get stuck in the local optima because this method is heuristic. We used the Hybrid Genetic Algorithm (HGA) for data clustering to avoid this problem. Our proposed method can predict early heart disease with an accuracy of 94.06%.

Lucas Nunno [40] proposed the work that was done on investigating applications of regression techniques on stock market price prediction. Linear and polynomial regression methods were applied along with the accuracies obtained using these methods. It was found that support vector regression was the most effective out of the models used, although there are opportunities to expand this research further using additional techniques and parameter tuning. It was found that for long term projected market fluctuations linear regression performed well. This case was especially true when a polynomial method would overfit the training data and have increased performance at the beginning of the testing data, but at the cost of very inaccurate results in the later prediction dates. Conversely, linear regression was less accurate at the beginning of the prediction, but wouldn't perform as badly as a polynomial regression method that diverged.

Pahulpreet Singh Kohli, Shriya Arora. [41] proposed The application of machine learning in the field of medical diagnosis is increasing gradually. This can be contributed primarily to the improvement in the classification and recognition systems used in disease diagnosis which is able to provide data that aids medical experts in early detection of fatal diseases and therefore, increase the survival rate of patients significantly. In this paper, we apply different classification algorithms, each with its own advantage on three separate databases of disease (Heart, Breast cancer, Diabetes) available in UCI repository for disease prediction. The feature selection for each dataset was accomplished by backward modeling using the p-value test. The results of the study strengthen the idea of the application of machine learning in early detection of diseases.

Narendra Mohan, Vinod Jain, Gauranshi Agrawal[42] proposed that Predicting and detecting cardiac disease has always been a difficult and time-consuming undertaking for doctors. To treat cardiac disorders, hospitals and other clinics are giving costly therapies and operations. As a result, anticipating cardiac disease in its early stages will be beneficial to people all around the world, allowing them to take required treatment before it becomes serious. Heart disease has been a major issue in recent years, with the primary causes being excessive alcohol use, tobacco use, and a lack of physical activity. Machine learning methods are utilized to forecast cardiac illnesses in this article. For training and testing, a data collection containing diverse human health parameters is used. Many AI&ML algorithms are used to predict cardiac disorders. The performance of the machine learning algorithm is compared after it has been implemented.

Deeraj Shetty, Kishor Rit, Sohail Shaikh, Nikita Patil[43] reveals that the goal of the data mining methodology is to think data from a data set and change it into a reasonable structure for further use. Our examination concentrates on this part of Medical conclusion learning design through the gathered data of diabetes and to create smart therapeutic choice emotionally supportive network to help the physicians. The primary target of this examination is to assemble Intelligent Diabetes Disease Prediction System that gives analysis of diabetes malady utilizing diabetes patient's database. In this system, we propose the use of algorithms like Bayesian and KNN (K-Nearest Neighbor) to apply on diabetes patient's database and analyze them by taking various attributes of diabetes for prediction of diabetes disease.

Shivendra Kaura, Assem Chandel, Nitin Kumar Pal[44] proposed that there's a demand for practitioners of medical to predict cardio diseases even way before when their patients is prone to any CVD. The options that increase the probabilities of the heart attacks are drinking and smoking, lack of physical activities, high level of vital sign, dangerous extent of cholesterol levels, unhealthy and unhygienic diet, damaging use of alcoholic beverage, and high sugar content level food. Cardio vascular diseases (CVD) constitutes coronary disease heart, vas or Stroke, and other hypertensive cardio disease, innate heart, other peripheral artery causes, rheumatic cardio disease, and inflammatory type cardio disease. Prophecies associated descriptions are principal goals of information mining; in observe Prediction of the processed data involves the attributes or the physiological variables in the data set to find a prediction probability or future state values of an alternative attributes. The Description will emphasize on the discovering a common recognizable patterns that describes that information that can be understood by humans.

Rukhsar Syed, Rajeev Kumar Gupta, Nikhlesh Pathik[45] proposed that Data mining is one of the emerging area in the field of computer science it's enable to deal with large dataset with different characteristic. In the current scenario it is used in every field like Medical. Education, Agriculture etc., but in the past few decades use of data mining approaches is increasing exponentially because it required prediction based on data for quick decision. Sometimes it is very challenging to predict accurately on large study data. Classification and observing them is one of the proper solution which driven by algorithms. In this paper a proposed algorithm is given which take advantage of partitioning based on tree, further working with adaptive SVM approach for classification. The proposed architecture used pre-processing under sampling SMORT which enable in pruning the data. The approach is experimented using the Weka tool on diabetic dataset and compared with traditional tree based RF, RT and J48 Approach. The observed outcome shows the efficiency of proposed algorithm over the traditional solution of processing diabetic data and finding efficient classification from it.

Tina Khajeh, Derek Reiman, Ryan Morley, Yang Dai[46] Using deep neural networks consisting of an autoencoder for extracting latent representations and a multilayer neural network for disease prediction, we show that gut metabolome is more predictive of inflammatory bowel disease (IBD) than gut microbiome. In addition, we design a new multi-task autoencoder to extract the latent profiles from the combined microbiome and metabolome data. We further demonstrate that the combined latent profiles can further improve the performance of prediction. In summary, our work shows that autoencoders are useful apparatuses in generating low dimensional profiles that contribute to the improved performance and robustness for IBD prediction.

Mahmood Hussain Kadhem, Ahmed M. Zeki[47] proposed that Data mining (DM) has a wide range of applications in the health care field. DM can be used to discover hidden patterns among different diagnoses or to predict the disease of patients based on certain number of symptoms. It can be used also to analyze the success major of a given treatment for a group of patients based on a number of characteristics and parameters available. This paper demonstrates the ability of DM to develop a prediction model for a presumptive diagnosis of two familiar urinary diseases: the acute inflammation of the urinary bladder and nephritis of renal pelvis. The dataset used in this work includes a number of characteristics, which are important in diagnosing any patient with an acute inflammation of urinary bladder or nephritis. This research evaluates the supervised machine learning algorithms Ridor, OneR, and J48 in terms of performance and accuracy to determine the best classification algorithm which will be used to develop the accurate prediction model. The decision tree (J48) shows a powerful accuracy and capability in prediction, and has been used to classify the patients' data with the proper acute inflammation diseases. The analyzed dataset has been trained using the 10-fold cross validation. The decision tree for the acute urinary bladder and nephritis has been generated.

CHAPTER 3:

IMPLEMENTATION AND SYSTEM DESIGN

3.1 Working of Machine Learning

The learning system of a machine learning algorithm is divided into three main parts.

- **A Decision Process**: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabelled, your algorithm will produce an estimate about a pattern in the data.

- **An Error Function**: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.

- **A Model Optimization Process**: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met. [48]

3.2 Machine Learning Methods

Machine learning classifiers fall into three categories-

Supervised machine learning

Known as supervised machine learning, is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, support vector machine (SVM).

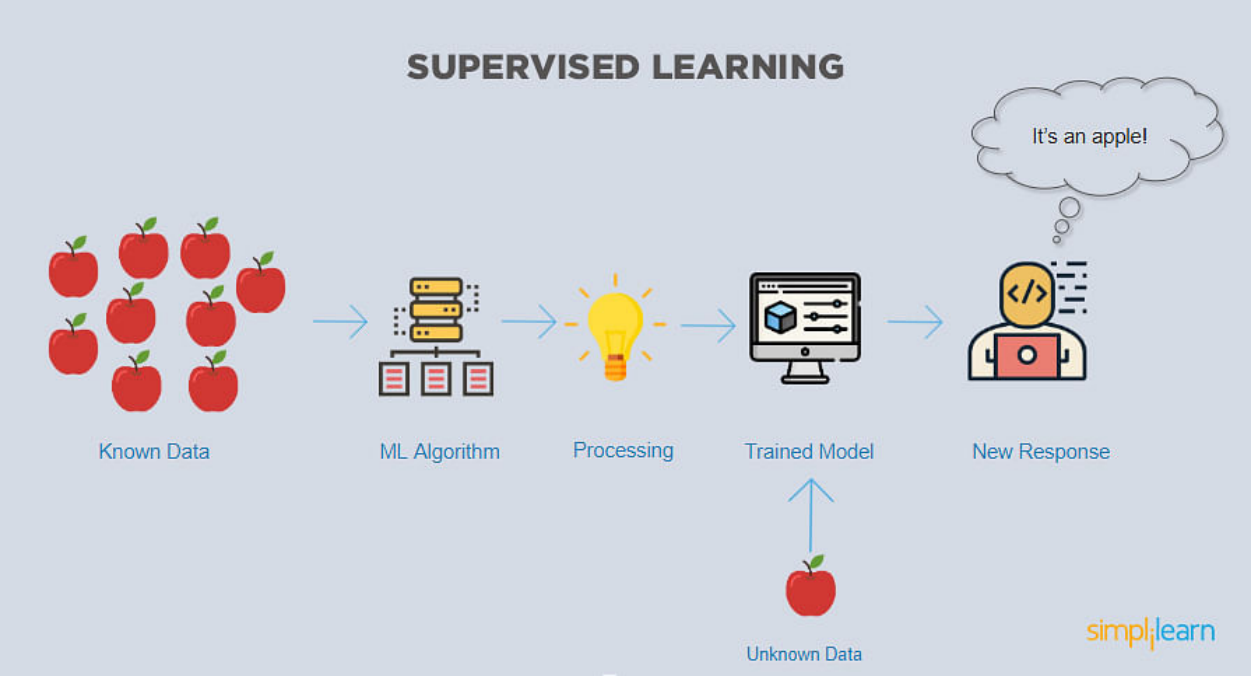


Fig. 3.1 Simplified Supervised Learning Diagram

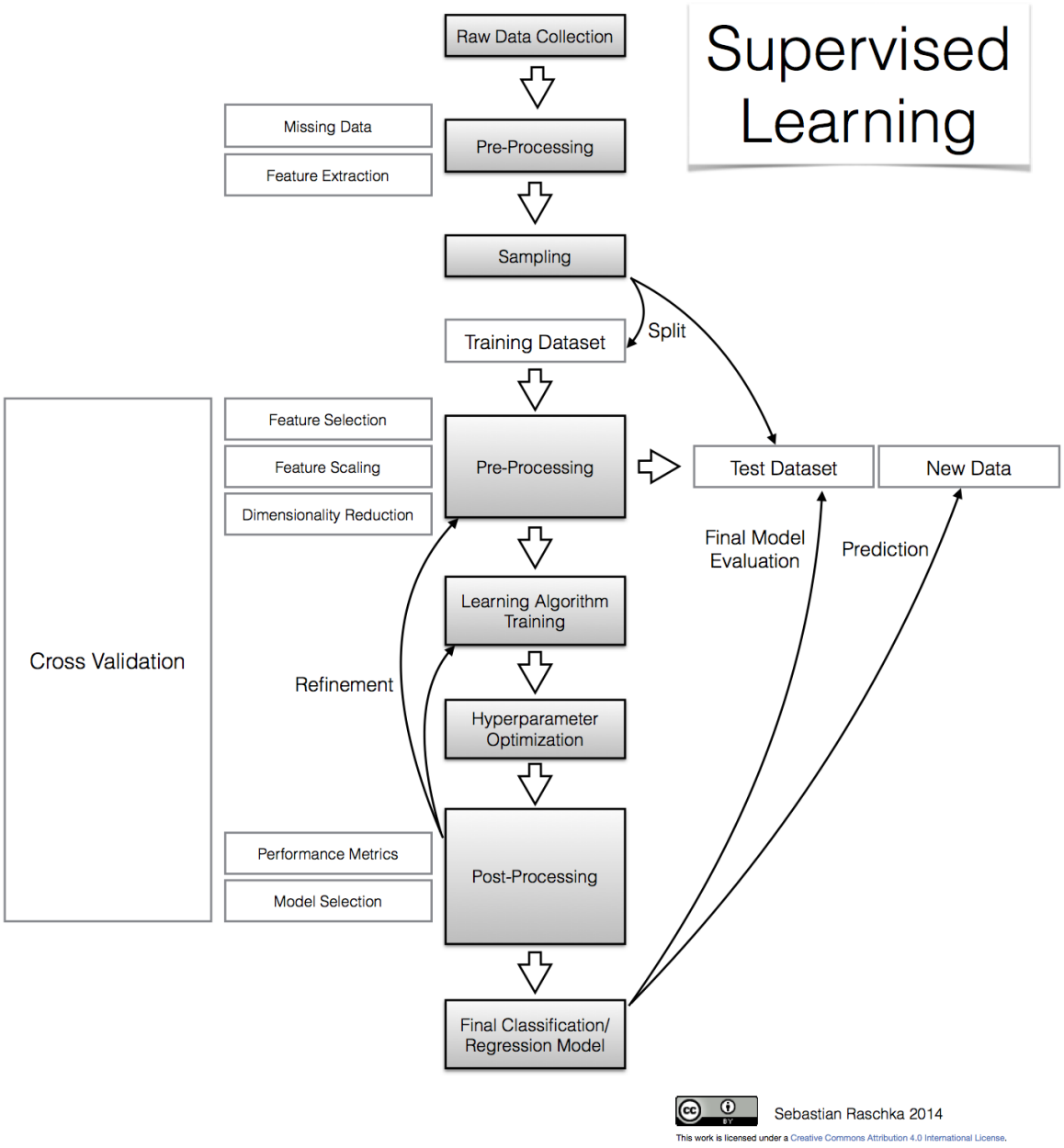


Fig.3.2 Workflow of supervised learning

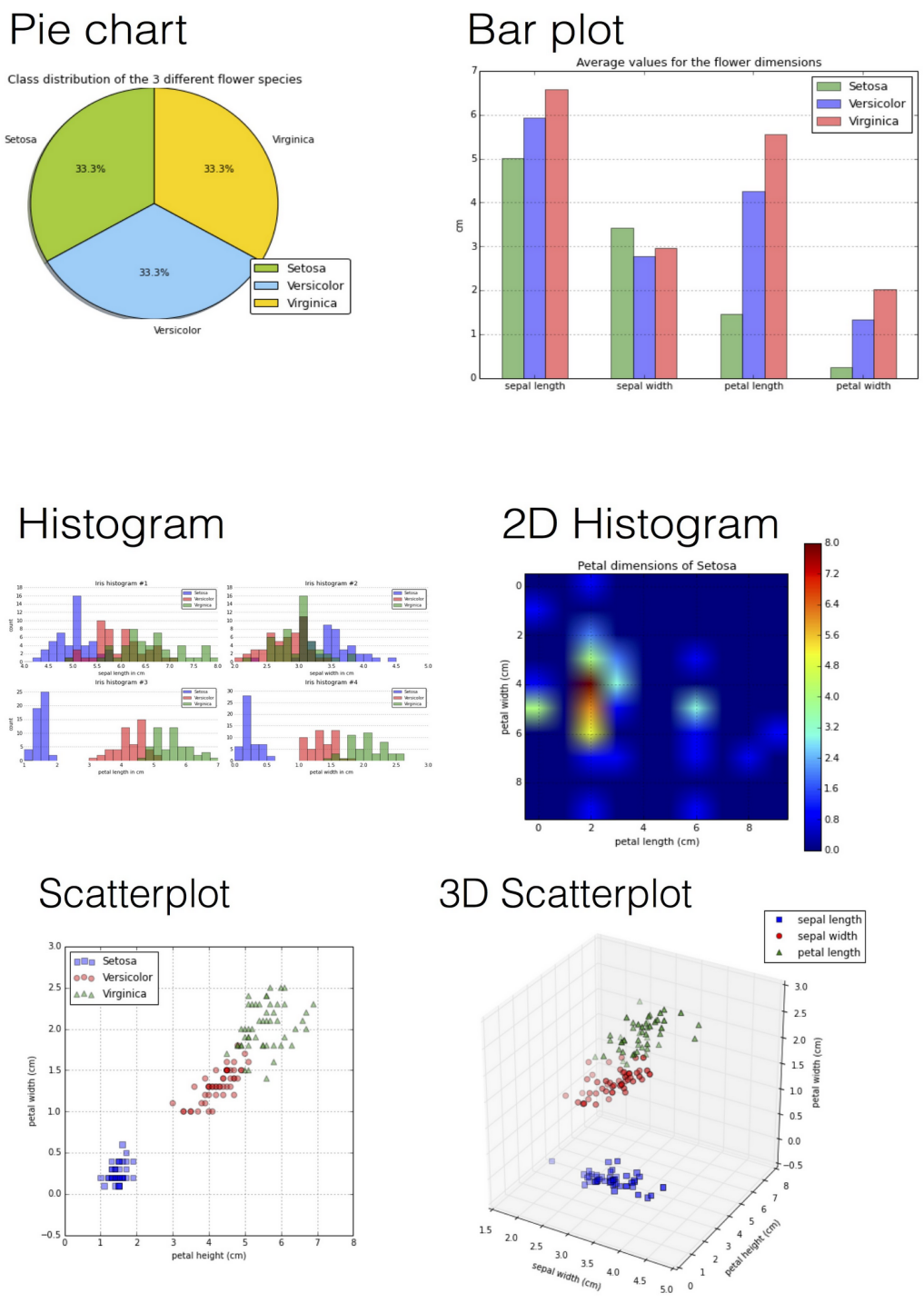


Fig. 3.3 Visualization of Supervised learning on a given dataset

Unsupervised machine learning

Unsupervised learning, uses machine learning algorithms to analysis and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction; principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, probabilistic clustering methods, and more.

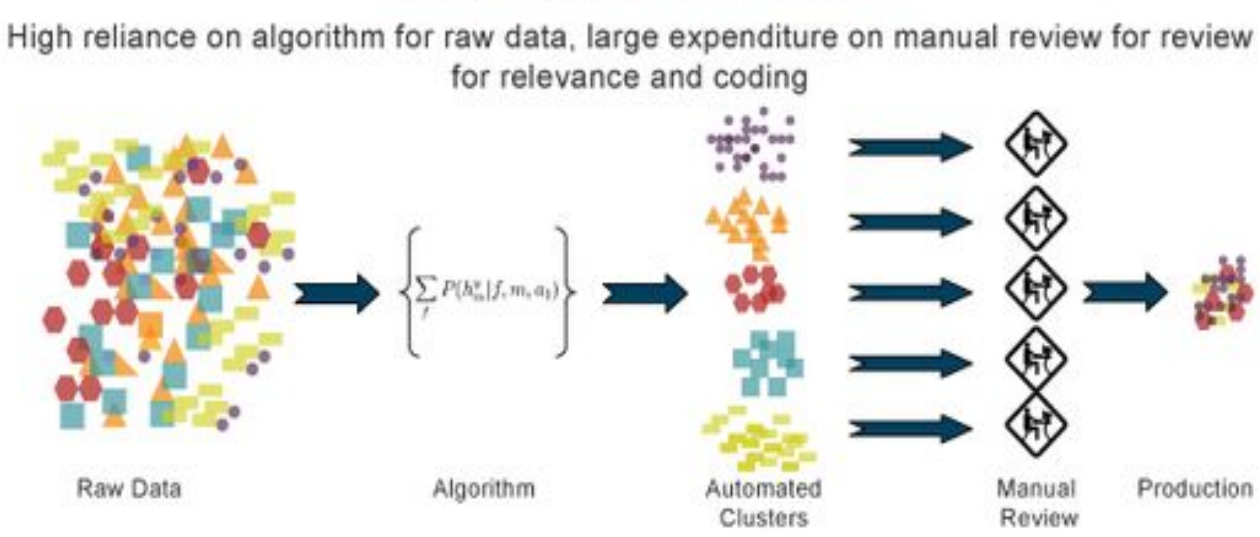


Fig.3.4 Workflow of Unsupervised Learning

Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labelled data set to guide classification and feature extraction from a larger, unlabelled data set. Semi-supervised learning can solve the problem of having not enough labelled data (or not being able to afford to label enough data) to train a supervised learning algorithm.

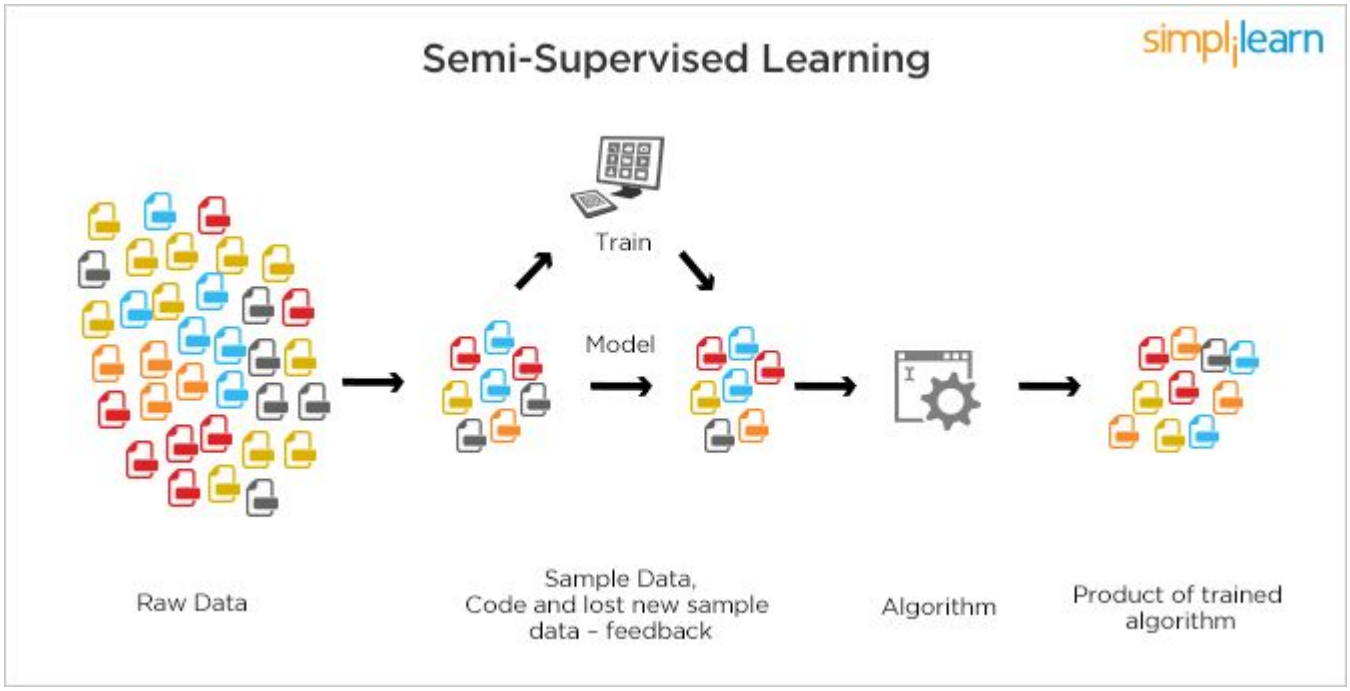


Fig. 3.5 Semi-Supervised Learning Model

3.3 Algorithms Used and Code Snippets

3.3.1 Decision tree -

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.[51]

In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

Before we dive deep, let's get familiar with some of the terminologies:

- Instances: Refer to the vector of features or attributes that define the input space
- Attribute: A quantity describing an instance
- Concept: The function that maps input to output
- Target Concept: The function that we are trying to find, i.e., the actual answer
- Hypothesis Class: Set of all the possible functions
- Sample: A set of inputs paired with a label, which is the correct output (also known as the Training Set)
- Candidate Concept: A concept which we think is the target concept
- Testing Set: Similar to the training set and is used to test the candidate concept and determine its performance[52]

Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, it’s also widely used in machine learning, which will be the main focus of this article.

3.3.1.1 How can an algorithm be represented as a tree?

For this let’s consider a very basic example that uses titanic data set for predicting whether a passenger will survive or not. Below model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).
A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn’t split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.[53]

3.3.1.2 Pruning

The performance of a tree can be further increased by pruning. It involves removing the branches that make use of features having low importance. This way, we reduce the complexity of tree, and thus increasing its predictive power by reducing overfitting. Pruning can start at either root or the leaves. The simplest method of pruning starts at leaves and removes each node with most popular class in that leaf, this change is kept if it doesn't deteriorate accuracy. Its also called reduced error pruning. More sophisticated pruning methods can be used such as cost complexity pruning where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree. This is also known as weakest link pruning.

3.3.1.3 Issues in decision trees

Avoiding overfitting

Since the ID3 algorithm continues splitting on attributes until either it classifies all the data points or there are no more attributes to splits on. As a result, it is prone to creating decision trees that overfit by performing really well on the training data at the expense of accuracy with respect to the entire distribution of data.

There are, in general, two approaches to avoid this in decision trees: - Allow the tree to grow until it overfits and then prune it. - Prevent the tree from growing too deep by stopping it before it perfectly classifies the training data.

A decision tree's growth is specified in terms of the number of layers, or depth, it's allowed to have. The data available to train the decision tree is split into training and testing data and then trees of various sizes are created with the help of the training data and tested on the test data. Cross-validation can also be used as part of this approach. Pruning the tree, on the other hand, involves testing the original tree against pruned versions of it. Leaf nodes are removed from the tree as long as the pruned tree performs better on the test data than the larger tree.

Incorporating continuous valued attributes

Our initial definition of ID3 is restricted to attributes that take on a discrete set of values. One way to make the ID3 algorithm more useful with continuous variables is to turn them, in a way, into discrete variables. Let's say in our example of Play Badminton the temperature is continuous (see the following table), we could test the information gain of certain partitions of the temperature values, such as temperature > 42.5. Typically, whenever the classification changes from no to yes

or yes to no, the average of the two temperatures is taken as a potential partition boundary.
For eg. 42 corresponds to No and 43 corresponds to Yes, 42.5 becomes a candidate. If any of the partitions end up exhibiting the greatest information gain, then it is used as an attribute and temperature is removed from the set of potential attributes to split on.

3.3.1.4 Decision Tree Code Snippet used-

```
def DecisionTree():
    from sklearn import tree
    clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
    clf3 = clf3.fit(X,y)

    # calculating accuracy---------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf3.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),
                         Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        # print (k,)
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1
    inputtest = [l2]
    predict = clf3.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t1.delete("1.0", END)
        t1.insert(END, disease[a])
    else:
        t1.delete("1.0", END)
        t1.insert(END, "Not Found")
```

```
def DecisionTree():
    from sklearn import tree
    clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
    clf3 = clf3.fit(X,y)

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf3.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(), ·····Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        # print (k,)
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1
    inputtest = [l2]
    predict = clf3.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break
    .          .
    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t1.delete("1.0", END)
        t1.insert(END, disease[a])
    else:
        t1.delete("1.0", END)
        t1.insert(END, "Not Found")
```

Fig. 3.6 Decision Tree Algorithm

3.3.2 Random Forest -

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.
Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

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). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

1. There needs to be some actual signal in our features so that models built using those features do better than random guessing.
2. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other. [54]

3.3.2.1 Ensuring that the Models Diversify Each Other

So how does random forest ensure that the behavior of each individual tree is not too correlated with the behavior of any of the other trees in the model? It uses the following two methods:

Bagging (Bootstrap Aggregation)— Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging.

Notice that with bagging we are not subsetting the training data into smaller chunks and training each tree on a different chunk. Rather, if we have a sample of size N, we are still feeding each tree a training set of size N (unless specified otherwise). But

instead of the original training data, we take a random sample of size N with replacement. For example, if our training data was [1, 2, 3, 4, 5, 6] then we might give one of our trees the following list [1, 2, 2, 3, 6, 6]. Notice that both lists are of length six and that “2” and “6” are both repeated in the randomly selected training data we give to our tree (because we sample with replacement).

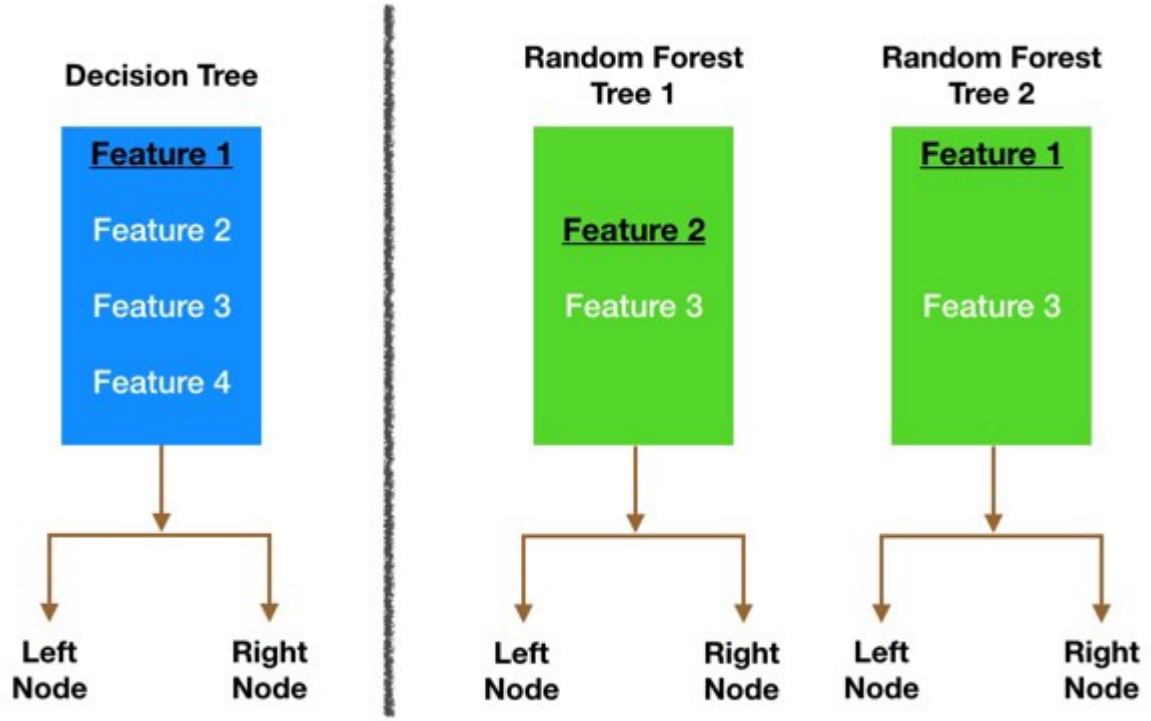


Fig 3.7 Node splitting in a random forest model is based on a random subset of features for each tree.

Feature Randomness — In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.[54]

Let’s go through a visual example — in the picture above, the traditional decision tree (in blue) can select from all four features when deciding how to split the node. It decides to go with Feature 1 (black and underlined) as it splits the data into groups that are as separated as possible.

Now let’s take a look at our random forest. We will just examine two of the forest’s trees in this example. When we check out random forest Tree 1, we find that it it can only consider Features 2 and 3 (selected randomly) for its node splitting decision. We know from our traditional decision tree (in blue) that Feature 1 is the best feature for splitting, but Tree 1 cannot see Feature 1 so it is forced to go with Feature 2 (black and underlined). Tree 2, on the other hand, can only see Features 1 and 3 so it is able to pick Feature 1.

3.3.2.2 Random Forest Code Snippet used

```
def randomforest():
    from sklearn.ensemble import RandomForestClassifier
    clf4 = RandomForestClassifier()
    clf4 = clf4.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf4.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # ---------------------------------------------------

    psymptoms =
[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = clf4.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t2.delete("1.0", END)
        t2.insert(END, disease[a])
    else:
        t2.delete("1.0", END)
        t2.insert(END, "Not Found")
```

```
[*]: def randomforest():
        from sklearn.ensemble import RandomForestClassifier
        clf4 = RandomForestClassifier()
        clf4 = clf4.fit(X,np.ravel(y))

        # calculating accuracy-------------------------------------------------------------
        from sklearn.metrics import accuracy_score
        y_pred=clf4.predict(X_test)
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        # ---------------------------------------------------------

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf4.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
    inputtest = [l2]
    predict = clf4.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t2.delete("1.0", END)
        t2.insert(END, disease[a])
    else:
        t2.delete("1.0", END)
        t2.insert(END, "Not Found")
```

Fig.3.8 Random Forest Algorithm

3.3.3 Naive Bayes -

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.[55]

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

3.3.3.1 How Naive Bayes algorithm works?

Let's understand it using an example. Below we have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Fig.3.9 A training data set of weather and corresponding target

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

3.3.3.2 Naïve Bayes Code Snippet used-

```
def NaiveBayes():
    from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb=gnb.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=gnb.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),
                            Symptom4.get(),Symptom5.get()]


    for k in range(0,len(l1)):
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = gnb.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t3.delete("1.0", END)
        t3.insert(END, disease[a])
    else:
        t3.delete("1.0", END)
        t3.insert(END, "Not Found")
```

```
[*]: def NaiveBayes():
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb=gnb.fit(X,np.ravel(y))

        # calculating accuracy-------------------------------------------------------------------
        from sklearn.metrics import accuracy_score
        y_pred=gnb.predict(X_test)
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        # ---------------------------------------------------

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]


        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = gnb.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
```

```
    .          .

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t3.delete("1.0", END)
        t3.insert(END, disease[a])
    else:
        t3.delete("1.0", END)
        t3.insert(END, "Not Found")
```

Fig. 3.10 Naive Bayes Algorithm

[56] 3.6 A Qualitative Comparison between Traditional Statistical Approach and Machine Learning Approach

Diseases predictions have been performed using different approaches. Some statistical approaches like Simple Moving Average (SMA), Random Forest, Exponential Smoothing, and Naive Approach were used traditionally to predict diseases in the earlier days. Since statistical approaches are linear in nature, it hampers prediction performances. As disease is unpredictable, chaotic, random and depends on several surrounding parameters, statistical approaches are not found to be so accurate.

In modern days of artificial intelligence, machine learning plays an important role in working on the modeling part. Using K-Fold cross-validation to evaluate the machine learning models. We used Support Vector Classifier, Gaussian Naive Bayes Classifier, and Random Forest Classifier for cross-validation. Some promising techniques like Simple Linear Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest and neural network models like Single Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Long Short Term Memory (LSTM) can be used for better prediction.

3.6.1 Comparing the Approches:

Data Preparation and Evaluation:

A. Gathering the Data:

Data preparation is the primary step for any machine learning problem. We will be using a **dataset** from Kaggle for this problem. This dataset consists of two CSV files one for training and one for testing. There is a total of 133 columns in the dataset out of which 132 columns represent the symptoms and the last column is the prognosis.

B. Cleaning the Data:

Cleaning is the most important step in a machine learning project. The quality of our data determines the quality of our machine learning model. So it is always necessary to clean the data before feeding it to the model for training. In our dataset all the columns are numerical, the target column i.e. prognosis is a string type and is encoded to numerical form using a **label encoder**.

C. Model Building:

After gathering and cleaning the data, the data is ready and can be used to train a machine learning model. We will be using this cleaned data to train the Support Vector Classifier, Naive Bayes Classifier, and Random Forest Classifier. We will be using a **confusion matrix** to determine the quality of the models.

D. Inference:
After training the three models we will be predicting the disease for the input symptoms by combining the predictions of all three models. This makes our overall prediction more robust and accurate.

Traditional Methods	Proposed Methods
Algorithms Statistical Methods 1) Simple Moving Average (SMA): In this method, an unweighted mean of a specific number of previous data is considered to be the predicted value for the next day. 2) Weighted Moving Average (WMA): The difference between SMA and WMA is that a weight is used with the previous values to predict the future value. 3) Exponential Smoothing: An smoothing constant, α is used for smoothing the prediction value from the previous prediction in Exponential Smoothing method. This smoothing constant maximizes prediction accuracy from the last prediction.	Algorithms Machine Learning Methods 1) 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. 2) Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. 3) The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically *a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.*
Prediction Techniques: Prediction has been performed using both traditional statistical and machine learning approaches. In case of statistical prediction, training of model is not necessary. This approach predicts data by using statistical mean of previous data. We have performed four such methods.	Prediction Techniques: In case of machine learning approaches, the models are trained first using the training data and then performs prediction on the testing data. Three machine learning methods are performed. Decision tree method is applied for trainig attributes. Moreover, Lasso and Ridge regressions are also used to minimize error and increase prediction accuracy. Scaled values are used as training attributes. Prediction has been performed using Random Forest algorithm for several number of estimators.

Table 3.1 Comparing Traditional and Proposed Methods

3.6.2 CONCLUSION

A comparative study between statistical approaches and machine learning approaches has been done in terms of prediction performances and accuracy. After studying all the methods individually, machine learning methods, Random forest and Naive Bayes are found to be the most accurate to predict diseases
.

CHAPTER 4:

RESULT AND DISCUSSION

4.1 MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes things easy and hard things possible [57]. Matplotlib is open source and we can use it freely. It is mostly written in python [58].

4.1.1 PYPLOT

[60] Pyplot is a matplotlib module which provides a matlab-like interface. matplotlib is designed to be as usable as matlab, with the ability to use python and the advantage of being free and open-source. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. the various plots we can utilize using pyplot are **line plot, histogram, scatter, 3d plot, image, contour and polar.**

[60] Parameters:

- **plot(x, y):** plot x and y using default line style and color.
- **plot.axis([xmin, xmax, ymin, ymax])**: scales the x-axis and y-axis from minimum to maximum values.
- **plot.xlabel('x-axis')**: names x-axis.
- **plot.ylabel('y-axis')**: names y-axis.

4.2 ACCURACY OF PREDICTED DATA

4.2.1 Naive Bayes Classifier

```
[*]: X = data.iloc[:,:-1]
     y = data.iloc[:, -1]
     X_train, X_test, y_train, y_test =train_test_split(
       X, y, test_size = 0.2, random_state = 24)

     print(f"Train: {X_train.shape}, {y_train.shape}")
     print(f"Test: {X_test.shape}, {y_test.shape}")

[ ]:
```

Fig 4.1 Importing Data and Organizing it into Testing and Training Sets

We have our data and now we want to see how well it can be fit to a linear model. After splitting the data, we will be now working on the modeling part. We will be using K-Fold cross-validation to evaluate the machine learning models. We will be using Decision Tree, Gaussian Naive Bayes Classifier, and Random Forest Classifier for cross-validation.

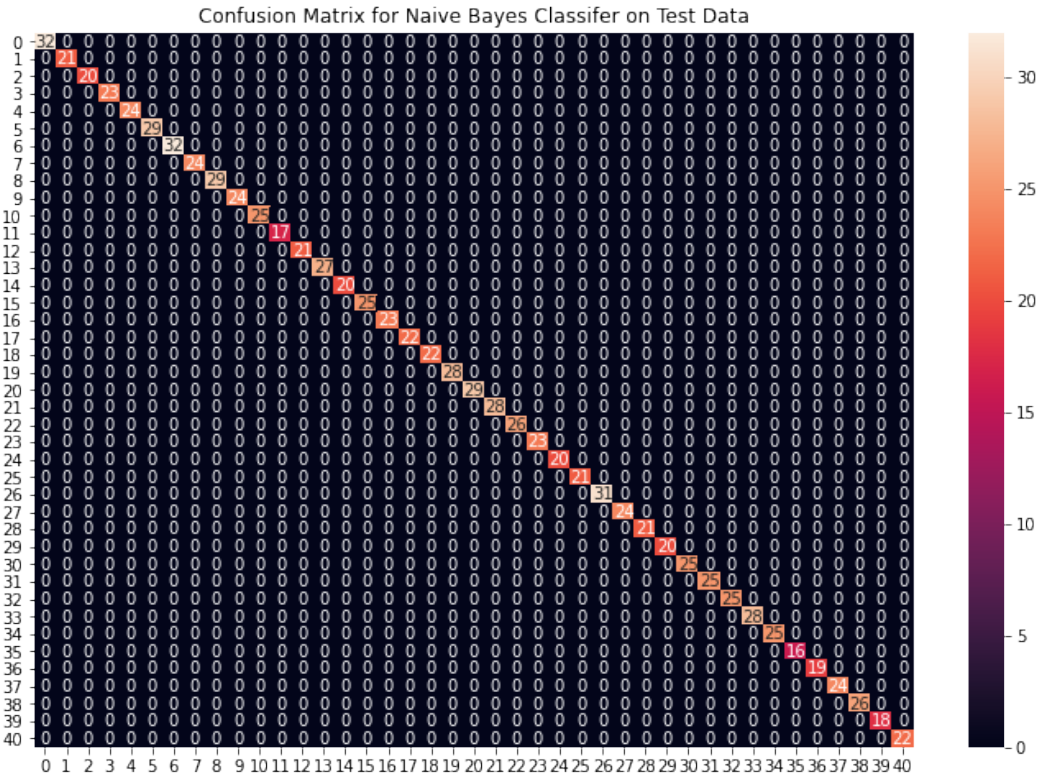
```
[*]: # Defining scoring metric for k-fold cross validation
def cv_scoring(estimator, X, y):
    return accuracy_score(y, estimator.predict(X))

# Initializing Models
models = {
    "Decision Tree":DecisionTree(),
    "Gaussian NB":GaussianNB(),
    "Random Forest":RandomForestClassifier(random_state=18)
}

# Producing cross validation score for the models
for model_name in models:
    model = models[model_name]
    scores = cross_val_score(model, X, y, cv = 10,
                             n_jobs = -1,
                             scoring = cv_scoring)
    print("=="*30)
    print(model_name)
    print(f"Scores: {scores}")
    print(f"Mean Score: {np.mean(scores)}")

[ ]:
```

Fig 4.2 Using K-Fold Cross-Validation for model selection

4.2.2 Accuracy through Naive Bayes

Accuracy on train data by Naive Bayes Classifier: 100.0
Accuracy on test data by Naive Bayes Classifier: 100.0

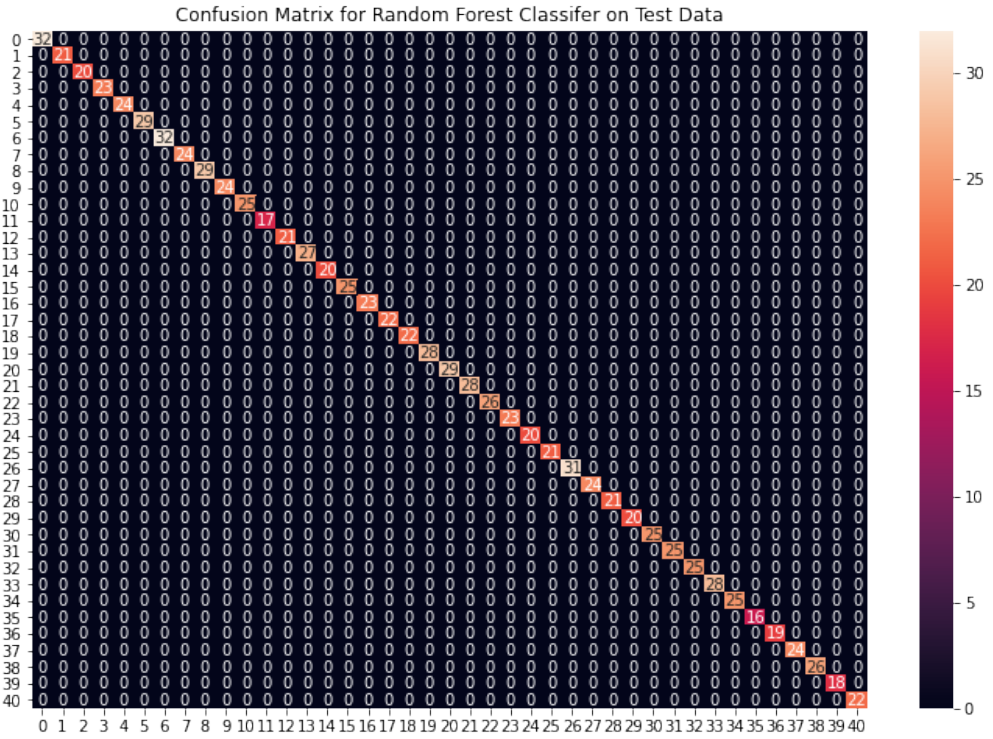
Fig 4.3



4.2.2 Accuracy through Random Forest

Accuracy on train data by Random Forest Classifier: 100.0
Accuracy on test data by Random Forest Classifier: 100.0

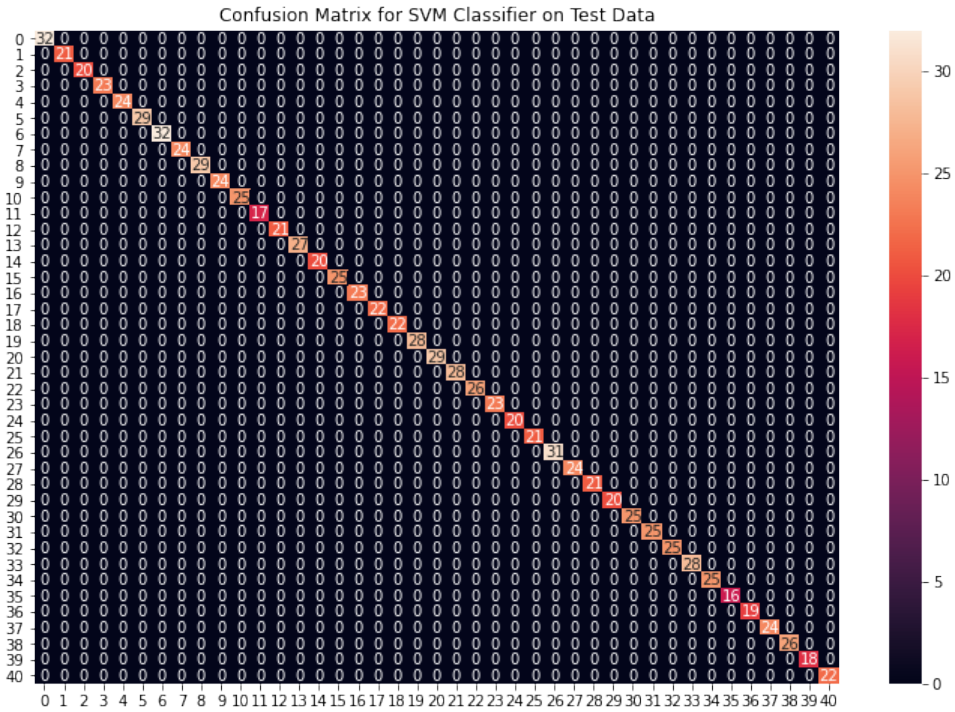
Fig 4.4



4.2.3 Accuracy through Decision Tree

Accuracy on train data by Decision Tree: 100.0
Accuracy on test data by Decision Tree Classifier: 100.0

Fig 4.5



CHAPTER 5:

CONCLUSION AND FUTURE SCOPE

Three techniques have been utilized in this project: Decision Tree, Random Forest and Naive Bayes. All the techniques have shown an improvement in the accuracy of predictions, thereby yielding positive results. With the following detailed studies and code practices, we studied what is machine learning, how it works and how it is implemented. We therefore concludes that our application works fine in prediction of the disease of a patient.

It has led to the conclusion that it is possible to predict diseases with more accuracy and efficiency using machine learning techniques. In the future, the disease prediction system can be further improved by utilizing a much bigger dataset than the one being utilized currently so that feeding more refined data sets can unimaginably improve the application's usability. . This would help to increase the accuracy of our prediction models. Furthermore, other models of Machine Learning could also be studied to check for the accuracy rate resulted by them.

For future recommendations, we can include more refined algorithms and data sets for training a better model, which can give more accurate results. Also, more symptoms and diseases can be added in the future for better detection.

On being refined, this application can be put to use in public hospitals and clinics for patients to get to know what problems they have based on the symptoms and consult the respective doctors accordingly.

Reference:

[1] Disease Prediction Using Machine Learning: Akash C. Jamgade, Prof. S. D. Zade ISO 9001:2008 Certified Journal https://www.irjet.net/archives/V6/i5/IRJET-V6I5977.pdf

[2] Jinesh Maloo https://blog.usejournal.com/machine-learning-for-beginners-from-zero-level-8be5b89bf77c

[3] Avijeet Biswal, “An Easy Guide to Stock Price Prediction Using Machine Learning” https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction using-machine-learning.

[4] Zack West, “Predicting Stock Prices with Linear Regression in Python” https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/.

[5] ACCURACY IMPROVEMENT FOR PREDICTING PARKINSON’S DISEASE PROGRESSION HTTPS://CYBERLENINKA.ORG/ARTICLE/N/1413930

[6] 2022 SAS Institute Inc., “Machine Learning What it is and why it matters” https://www.sas.com/en_in/insights/analytics/machine-learning.html.

[7] tutorialspoint, “Machine Learning with Python-Basics”, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_basics.htm

[8] WikipediaThe Free Encyclopedia. “Machine Learning”, https://en.wikipedia.org/wiki/Machine_learning.

[9] Robust Results Pvt. Ltd., Prutor Online Academy “Machine Learning with Python – Basics” https://prutor.ai/machine-learning-with-python-basics/

[10] Ed Burns, Executive Editor, TechTarget, “machine learning”, https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

[11] Shubham Bansal, GeeksforGeeks, “Supervised and Unsupervised Machine Learning”, https://www.geeksforgeeks.org/supervised-unsupervised-learning/

[12] Tutorialspoint “Scikit-Learn Tutorial”, https://www.tutorialspoint.com/scikit_learn/index.html

[13] Tutorialspoint “Scikit-Learn Introduction”, https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.html

[14] Wikipedia The Free Encyclopedia, “scikit-learn”, https://en.wikipedia.org/wiki/Scikit-learn

[15] nikhilaggarwal3, GeeksforGeeks, “Introduction to Pandas in Python”, https://www.geeksforgeeks.org/introduction-to-pandas-in-python/

[16] Tutorialspoint, “Python Pandas-Quick Guide” https://www.tutorialspoint.com/python_pandas/python_pandas_quick_guide.htm

[17] Michael Waskom, seaborn, “An Introduction to Seaborn”, https://seaborn.pydata.org/introduction.html

[18] Choco_Chips, GeeksforGeeks, “Seaborn | Categorical plots” https://www.geeksforgeeks.org/seaborn-categorical-plots/

[19] Javatpoint, “Python seaborn Library”, https://www.javatpoint.com/python-seaborn-library

[20] W3Schools, “Numpy Introduction” https://www.w3schools.com/python/numpy/numpy_intro.asp

[21] sareendivyansh, GeeksforGeeks, “Numpy array in Python” https://www.geeksforgeeks.org/numpy-array-in-python/

[22] nikhilaggarwal3, GeeksforGeeks, “Python datetime module”, https://www.geeksforgeeks.org/python-datetime-module/

[23] Jupyter, “JupyterLab: A Next-Generation Notebook Interface”, https://jupyter.org/

[24] Antonio Bazarra-Fernandez (Member, IEEE), BONE EDUCATION: INNOVATION AND CREATIVITY HTTPS://CYBERLENINKA.ORG/ARTICLE/N/947817

[25] WEI YU*, TIEBIN LIU, RODOLFO VALDEZ, MARTA GWINN, MUIN J KHOURY: APPLICATION OF SUPPORT VECTOR MACHINE MODELING FOR PREDICTION OF COMMON DISEASES: THE CASE OF DIABETES AND PRE-DIABETES HTTPS://CYBERLENINKA.ORG/ARTICLE/N/906197

[26] Mohammad R. Mohebian, Hamid R. Marateb, Marjan Mansourian, Miguel Angel Mañanas, Fariborz Mokarian, “A HYBRID COMPUTER-AIDED-DIAGNOSIS SYSTEM FOR PREDICTION OF BREAST CANCER RECURRENCE (HPBCR) USING OPTIMIZED ENSEMBLE LEARNING ” https://cyberleninka.org/article/n/696527

[27] Truyen Tran1,2, Wei Luo1, Dinh Phung1, Sunil Gupta1, Santu Rana1, Richard Lee Kennedy3, Ann Larkins4 and Svetha Venkatesh1 “A framework for feature extraction from hospital medical data with applications in risk prediction”, Tran et al. BMC Bioinformatics (2014) 15:425 DOI 10.1186/s12859-014-0425-8 https://cyberleninka.org/article/n/1060043

[28] LiMin Wang1'2, “Mining causal relationships among clinical variables for cancer diagnosis based on Bayesian analysis”, Wang BioData Mining (2015) 8:13 DOI 10.1186/s13040-015-0046-4

[29] Fatma Patlar Akbulut1*, Erkan Akkur2, Aydin Akan3 and B Siddik Yarman3 A decision support system to determine optimal ventilator settings https://cyberleninka.org/article/n/428290

[30] Ya-Ju Chang1, Hui-Chun Huang1, Yuan-Yu Hsueh2, Shao-Wei Wang3 “ROLE OF EXCESSIVE AUTOPHAGY INDUCED BY MECHANICAL OVERLOAD IN VEIN GRAFT NEOINTIMA FORMATION: PREDICTION AND PREVENTION

”, SCIENTIFIC REPpRTS https://cyberleninka.org/article/n/1456002

[31] Jacob K Kariuki1*, Eileen M Stuart-Shor1,2+, Suzanne G Leveille1 + and Laura L Hayman1 + “Evaluation of the performance of existing non-laboratory based cardiovascular risk assessment algorithms”, Cardiovascular Disorders: https://cyberleninka.org/article/n/1163544

[32] SENTHILKUMAR MOHAN 1 , CHANDRASEGAR THIRUMALAI1, AND GAUTAM SRIVASTAVA, “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” doi 10.1109/ACCESS.2019.2923707

[33] CHUNYAN GUO, ZHIQIANG HAN, JIABING ZHANG , AND JIANSHE YU "Recursion Enhanced Random Forest With an Improved Linear Model (RERF-ILM) for Heart Disease Detection on the Internet of Medical Things Platform", Digital Object Identifier 10.1109/ACCESS.2020.2981159

[34] Dhiraj Dahiwade; Gajanan Patle; Ektaa Meshram “Designing Disease Prediction Model Using Machine Learning Approach”, doi: 10.1109/ICCMC.2019.8819782

[35] Chun-XiaoNie et al, “Analyzing the stock market based on the structure of kNN network”, Chaos, Solitones and Fractals, 2018 Elsevier Ltd., https://doi.org/10.1016/j.chaos.2018.05.018

[36] Sateesh Ambesange; Vijayalaxmi A; Rashmi Uppin; Shruthi Patil; Vilaskumar Patil “Optimizing Liver disease prediction with Random Forest by various Data balancing Techniques”, DOI:10.1109/CCEM50674.2020.00030

[37] Tao Ban et al, “Referential kNN Regression for Financial Time Series Forecasting”, International Conference on Neural Information Processing, 2022 Springer Nature Switzerland AG.

[38] Ankita Dewan, Meghna Sharma, “Prediction of heart disease using a hybrid technique in data mining classification”, INSPEC Accession Number:15110061

[39] Pahulpreet Singh Kohli, Shriya Arora, “Application of Machine Learning in Disease Prediction”, INSPEC Accession Number: 18868504, DOI: 10.1109/CCAA.2018.8777449

[40] Lucas Nanno et al, “Stock Market Price Prediction Using Linear and Polynomial Regression Models”, University of New Mexico Computer Science Department Albuquerque, New Mexico, United States lnunno@cs.unm.edu

[41] Md. Touhidul Islam, Sanjida Reza Rafa, Md. Golam Kibria, “Early Prediction of Heart Disease Using PCA and Hybrid Genetic Algorithm with k-Means”, DOI: 10.1109/ICCIT51783.2020.9392655

[42] Narendra Mohan, Vinod Jain, Gauranshi Agrawal” Heart Disease Prediction Using Supervised Machine Learning Algorithms”. DOI:10.1109/ISCON52037.2021.9702314

[43] Deeraj Shetty, Kishor Rit, Sohail Shaikh, Nikita Patil“Diabetes disease prediction using data mining” DOI:10.1109/ICIIECS.2017.8276012

[44] Shivendra Kaura, Assem Chandel, Nitin Kumar Pal“Heart disease-Sinus arrhythmia prediction system by neural network using ECG analysis”. DOI: 10.1109/PEEIC47157.2019.8976829

[45] Rukhsar Syed; Rajeev Kumar Gupta; Nikhlesh Pathik “An Advance Tree Adaptive Data Classification for the Diabetes Disease Prediction” DOI:10.1109/ICRIEECE44171.2018.9009180

[46] Tina Khajeh; Derek Reiman; Ryan Morley; Yang Dai “Integrating microbiome and metabolome data for host disease prediction via deep neural networks” DOI: 10.1109/BHI50953.2021.9508601

[47] Mahmood Hussain Kadhem; Ahmed M. Zeki, “Prediction of Urinary System Disease Diagnosis: A Comparative Study of Three Decision Tree Algorithms” DOI: 10.1109/CASH.2014.25

[48] Greg Brand, “Yahoo Finance API-A Complete Guide”, AlgoTrading101 Blog, https://algotrading101.com/learn/yahoo-finance-api-guide/

[49] Ran Aroussi, “fix-yahoo-finance 0.1.37”, https://pypi.org/project/fix-yahoo-finance/

[50] Machine Learning IBM https://www.ibm.com/cloud/learn/machine-learning

[51] Medium: Decision Tree https://medium.com/@imparth/decision-tree-4d0bc22620ab

[52] Decision Tree Algorithm overview explained, “https://towardsmachinelearning.org/decision-tree-algorithm/

[53] Prashant Gupta “Decision Trees in Machine Learning”, https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

[54] Khalid Alkhatib et al, “Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm”, International Journal of Business, Humanities and Technology Vol. 3 No. 3; March 2013 Centre for Promoting Ideas, USA www.ijbhtnet.com

[55] Javatpoint, “K-Nearest Neighbor(KNN) Algorithm for Machine Learning”, https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[56] Indronil Bhattacharjee et al, “Stock Price Prediction: A Comparative Study between Traditional Statistical Approach and Machine Learning Approach”, Conference: 4th International Conference on Electrical Information and Communication Technology (EICT) At: Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh DOI: 10.1109/EICT48899.2019.9068850

[57] 2021The Matplotlib Development Team, “Matplotlib: Visualization with Python”, https://matplotlib.org/

[58] W3Schools, “Matplotlib Tutorial”, Refsnes Data, https://www.w3schools.com/python/matplotlib_intro.asp

[59] KattamuriMeghna, “Python | Introduction to Matplotlib”, GeeksforGeeks, https://www.geeksforgeeks.org/python-introduction-matplotlib/

[60] akshay_sharma08, “Pyplot in Matplotlib”, GeeksforGeeks, https://www.geeksforgeeks.org/pyplot-in-matplotlib/

[61] RajuKumar19, “Matplotlib.dates.DateFormatter class in Python”, GeeksforGeeks, https://www.geeksforgeeks.org/matplotlib-dates-dateformatter-class-in-python/