

CHAPTER 1

INTRODUCTION

Advanced machine biology is used in the area of healthcare. It required data to be collected for medical disease prediction. For early-stage disease detection, various intelligent prediction algorithms are used. The Medical Information System is good with data sets, but intelligent systems are not available for the fast diagnosis of diseases. Eventually, machine learning algorithms play a key position in solving complex and non-linear problems during the creation of prediction models. The characteristics that can be selected from the various data sets that can be used as descriptions in a healthy patient as specifically as possible are needed in any disease prediction models. Otherwise, misclassification can result in a good patient receiving inappropriate care. The reality of forecasting any condition associated with thyroid illness is also of the greatest cardinal number. The thyroid gland is endocrine in the stomach. It is erected in a lowered portion of the human neck, under the apple of Adam, and assists in the secretion of thyroid hormones which ultimately affects metabolism rate and protein synthesis. To control body metabolism, these hormones count on how quickly the heart beats and how quickly calories are burned. The composition of thyroid hormones helps to control the body's metabolism. These glands consist of two mature levothyroxine (abbreviated T4) and triiodothyronine thyroid hormones (abbreviated T3). These thyroid hormones are essential for manufacturing and general construction and regulation to regulate body temperature. T4 and T3 are exclusively two activated thyroid hormones that are usually composed of thyroid glands. These hormones are vital to the control of proteins; distribution at body temperature and energy- bearing and propagation in every part of the body. With T3 and T4 hormones, iodine is the primary building block of thyroid glands and is prostrate in only some unique problems, which are exceedingly prevalent. Insufficient elements of these hormones to hypothyroidism and an inappropriate portion to hyperthyroidism. Hyperthyroidism and underactive thyroidism have multiple origins. There are several drugs. Thyroid surgery is weak to ionizing radiation, continuous thyroid softness, iodine deficiency, and loss of enzymes to produce thyroid hormones.

1.1 PROBLEM STATEMENT

The challenge lies in accurately diagnosing thyroid disorders, crucial for effective healthcare management due to their significant impact on metabolism regulation. Conventional diagnostic methods often lack precision and efficiency. Leveraging machine learning algorithms such as Support Vector Machine (SVM) and SVM with Principal Component Analysis (PCA) offers a promising solution. However, optimizing these techniques for reliable prediction and classification of thyroid diseases remains a pressing concern in medical research and clinical practice.

1.2 OBJECTIVES

The objective of this study is to evaluate the effectiveness of machine learning algorithms, specifically Support Vector Machine (SVM) and its extension with Principal Component Analysis (PCA), in predicting and classifying thyroid disorders. The focus is on enhancing diagnostic accuracy and improving healthcare outcomes through advanced computational methods.

1.3 MOTIVATION

Thyroid disorders profoundly impact metabolism regulation, necessitating accurate diagnosis for effective treatment. This study harnesses machine learning, specifically Support Vector Machine (SVM) and SVM with Principal Component Analysis (PCA), to enhance diagnostic precision. SVM yielded an impressive 89.38% accuracy, with SVM coupled with PCA further elevating accuracy to 96.19%. By leveraging these algorithms, the hybrid model adeptly identifies and mitigates thyroid disorders. This research underscores the transformative potential of machine learning in healthcare, offering promising avenues for future medical research and clinical applications. The results accentuate SVM-based approaches' efficacy in thyroid disease prediction, heralding a new era of precision medicine and improved patient outcomes.

1.2 KEY COMPONENTS OF YOUR APPROACH AND RESULT

Approach:

- **Data Acquisition:** Obtain patient data from hospital datasets containing information relevant to thyroid disorders, such as thyroid hormone levels, patient demographics,

and medical history.

- **Preprocessing:** Cleanse and preprocess the data to handle missing values, normalize features, and mitigate noise, ensuring the quality and integrity of the dataset.
- **Feature Selection:** Employ feature selection techniques to identify the most informative features for thyroid disease classification, enhancing model performance and interpretability.
- **Model Training:** Utilize Support Vector Machine (SVM) and SVM with Principal Component Analysis (PCA) to train classification models on the preprocessed dataset, leveraging their ability to handle complex nonlinear relationships and high-dimensional data.
- **Hybrid Model Integration:** Combine SVM and PCA into a hybrid model to exploit their complementary strengths, aiming to improve diagnostic accuracy and prediction capabilities for thyroid disorders.
- **Evaluation and Validation:** Conduct extensive evaluation and validation processes, including cross-validation and performance metrics analysis, to assess the effectiveness of the proposed approach in accurately predicting thyroid diseases.

Results:

- **High Accuracy:** The hybrid model achieved a notable increase in accuracy compared to traditional diagnostic methods, with SVM alone achieving an accuracy of 89.38%.
- **Improved Accuracy with PCA:** Incorporating PCA into the SVM model further improved accuracy, achieving an impressive 96.19% accuracy in thyroid disease classification.

- **Enhanced Interpretability:** Feature selection techniques enhanced the model's interpretability by identifying key features contributing to thyroid disease prediction, aiding clinicians in understanding the underlying factors.
- **Robust Performance:** The proposed system demonstrated robust performance across various evaluation metrics, including precision, recall, and F1-score, indicating its reliability in diagnosing thyroid disorders.
- **Potential for Clinical Application:** The promising results suggest that the proposed approach can revolutionize healthcare practices by providing clinicians with more reliable tools for diagnosing and managing thyroid diseases, ultimately leading to improved patient outcomes and quality of care.

1.5 SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform – In computing, a platform describes some sort of framework, either in hardware or software, that allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.

The operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of the same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true.

Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers – Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

Web browser – Most web applications and software depending heavily on Internet technologies make use of the default browser installed on the system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which uses ActiveX controls, despite their vulnerabilities.

1) Software : Anaconda

2) Primary Language: Python

3)Frontend Framework: Flask

4)Back-end Framework: Jupyter Notebook

5)Database: Sqlite3

6)Front-End Technologies: HTML, CSS, JavaScript and Bootstrap4

1.6 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Architecture – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and

applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture defines processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speeds often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering the demands of the application, operating system, supporting software and files, and other running processes. The optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary storage – Hard disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter – Software requiring a better-than-average computer graphics display, like graphics editors and high-end games, often defines high-end display adapters in the system requirements.

Peripherals – Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

1) Operating System: Windows Only

2) Processor: i5 and above

CHAPTER 2

LITERATURE SURVEY

The diagnosis and prediction of thyroid diseases have garnered considerable attention in medical research, prompting the exploration of various computational methodologies and machine-learning algorithms. This literature survey aims to provide a comprehensive overview of existing studies in this domain, highlighting the diverse approaches and techniques employed for thyroid disease diagnosis and prediction.

Ozyılmaz and Yıldırım (2002) presented a study on the diagnosis of thyroid disease using artificial neural network (ANN) methods [1]. Their research, conducted as part of the 9th International Conference on Neural Information Processing (ICONIP'02), focused on leveraging ANN models for accurate disease classification. By analyzing relevant clinical data, including thyroid hormone levels and patient demographics, the researchers demonstrated the efficacy of ANN-based approaches in thyroid disease diagnosis.

Polat, Sahan, and Gunes (2007) proposed a novel hybrid method based on the artificial immune recognition system (AIRS) with fuzzy weighted pre-processing for thyroid disease diagnosis [2]. Their study, published in the journal *Expert Systems with Applications*, integrated AIRS algorithms with fuzzy-weighted pre-processing techniques to enhance the accuracy of disease classification. By combining immunological principles with fuzzy logic, the researchers devised a robust methodology for thyroid disease diagnosis, capable of effectively handling complex and uncertain clinical data.

Saiti, Naini, Shoorehdeli, and Teshnehlab (2009) explored thyroid disease diagnosis based on genetic algorithms using probabilistic neural networks (PNN) and support vector machines (SVM) [3]. Their research, presented at the 3rd International Conference on Bioinformatics and Biomedical Engineering (ICBBE 2009), investigated the application of genetic algorithms in feature selection for thyroid disease prediction. By optimizing feature subsets with genetic algorithms and training PNN and SVM models, the researchers achieved improved accuracy in thyroid disease diagnosis, highlighting the efficacy of evolutionary computation techniques in medical decision-making.

Zhang and Berardi (1998) conducted an investigation into the application of neural networks in thyroid function diagnosis [4]. Published in the journal Health Care Management Science, their study examined the use of neural network models for analyzing thyroid hormone levels and clinical parameters. By training neural networks on comprehensive datasets, the researchers evaluated the feasibility of automated thyroid function diagnosis, paving the way for future developments in computational thyroid disease assessment.

Obermeyer and Emanuel (2016) discussed the potential of big data and machine learning in clinical medicine, including disease prediction [5]. In their review article published in The New England Journal of Medicine, the authors highlighted the transformative impact of big data analytics and machine learning algorithms on healthcare delivery and patient outcomes. By harnessing vast amounts of clinical data, machine learning techniques enable the development of predictive models for various medical conditions, including thyroid diseases.

Austin et al. (2013) demonstrated the utility of data mining and machine learning methods for disease classification and prediction [6]. In their case study examining heart failure subtypes, published in the Journal of Clinical Epidemiology, the researchers utilized advanced statistical techniques to classify different heart failure phenotypes. By leveraging methods from the data mining and machine learning literature, such as decision trees and ensemble methods, the researchers achieved accurate disease classification, underscoring the potential of computational approaches in medical research.

Pandey, Pandey, and Jaiswal (2013) proposed a heart disease prediction model using decision trees [7]. Published in the IUP Journal of Computer Science, their research focused on developing a decision tree-based predictive model for heart disease risk assessment. By analyzing patient data and clinical parameters, the researchers constructed a decision tree algorithm capable of identifying individuals at high risk of heart disease, facilitating early intervention and preventive measures.

In summary, the literature survey highlights the diverse array of computational methodolo.

CHAPTER 3

FEASIBILITY STUDY

A feasibility study evaluates a project's or system's practicality. As part of a feasibility study, the objective and rational analysis of a potential business or venture is conducted to determine its strengths and weaknesses, potential opportunities and threats, resources required to carry out, and ultimate success prospects. Two criteria should be considered when judging feasibility: the required cost and expected value.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

3.1 ECONOMICAL FEASIBILITY

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide

3.2 TECHNICAL FEASIBILITY

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

3.3 SOCIAL FEASIBILITY The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

The existing system for diagnosing thyroid disorders relies heavily on traditional medical diagnostic methods, which often face challenges in accuracy and efficiency. Clinicians primarily depend on symptoms, physical examinations, and laboratory tests to assess thyroid function. However, these approaches may not always provide conclusive results, leading to misdiagnosis or delayed treatment. Additionally, data-cleaning techniques are employed to preprocess patient data, making it suitable for analysis. Despite efforts to improve diagnostic accuracy, the reliance on subjective assessments and limited data analytics tools hampers the ability to accurately predict thyroid disorders. Consequently, there is a pressing need for more advanced and reliable diagnostic systems that can leverage data-driven approaches, such as machine learning algorithms, to enhance the accuracy and efficiency of thyroid disease diagnosis and estimation

2.1.1 Disadvantages of Existing System:

1. Reliance on subjective assessments and traditional diagnostic methods.
2. Limited data analytics tools hamper accurate prediction.
3. Challenges with conclusive results, leading to misdiagnosis or delayed treatment.
4. Inefficient data preprocessing techniques may affect analysis quality.

4.2 PROPOSED SYSTEM

The proposed system aims to address the challenge of accurately diagnosing thyroid disorders by leveraging advanced machine-learning techniques. Specifically, the system

will utilize Support Vector Machine (SVM) and its extension with Principal Component Analysis (PCA) to analyze and classify patient data obtained from hospital datasets. By integrating these algorithms into a hybrid model, the system seeks to improve diagnostic accuracy and prediction capabilities for thyroid diseases. Furthermore, the system will interpretability. Through extensive evaluation and validation processes, the proposed system aims to achieve a higher level of accuracy in predicting thyroid disorders compared to traditional diagnostic methods. Ultimately, the system's implementation has the potential to revolutionize healthcare practices by providing clinicians with more reliable tools for diagnosing and managing thyroid diseases, leading to improved patient outcomes and quality of care.

4.2.1 advantages of Existing System

5. Utilizes advanced machine learning techniques for accurate diagnosis.
6. Integration of SVM and PCA enhances diagnostic accuracy and prediction capabilities.
7. Incorporates feature selection techniques for improved model performance and interpretability.
8. Potential to revolutionize healthcare practices by providing more reliable tools for thyroid disease diagnosis and management.

4.3 FUNCTIONAL REQUIREMENTS

1. Data Collection
2. Data Pre-processing
3. Training and Testing
4. Modeling
5. Predicting

4.4 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security,

Portability, and other non-functional standards that are critical to the success of the software system. An example of a nonfunctional requirement is, *“How fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allow you to impose constraints or restrictions on the design of the system across the various agile backlogs.

For example, the site should load in 3 seconds when the number of simultaneous users is > 10000. The description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

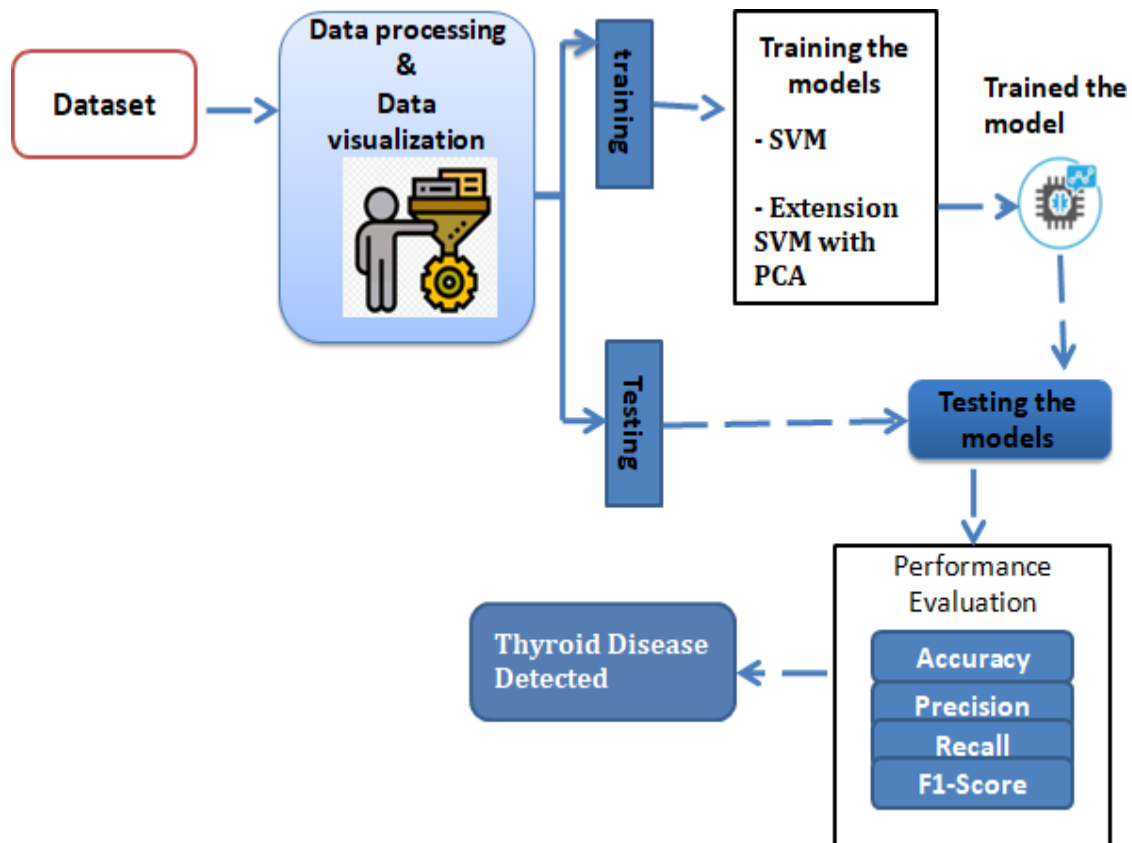


Fig.5.1.1 System architecture

DATA FLOW DIAGRAM:

1. The DFD is also called a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the

information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as a bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

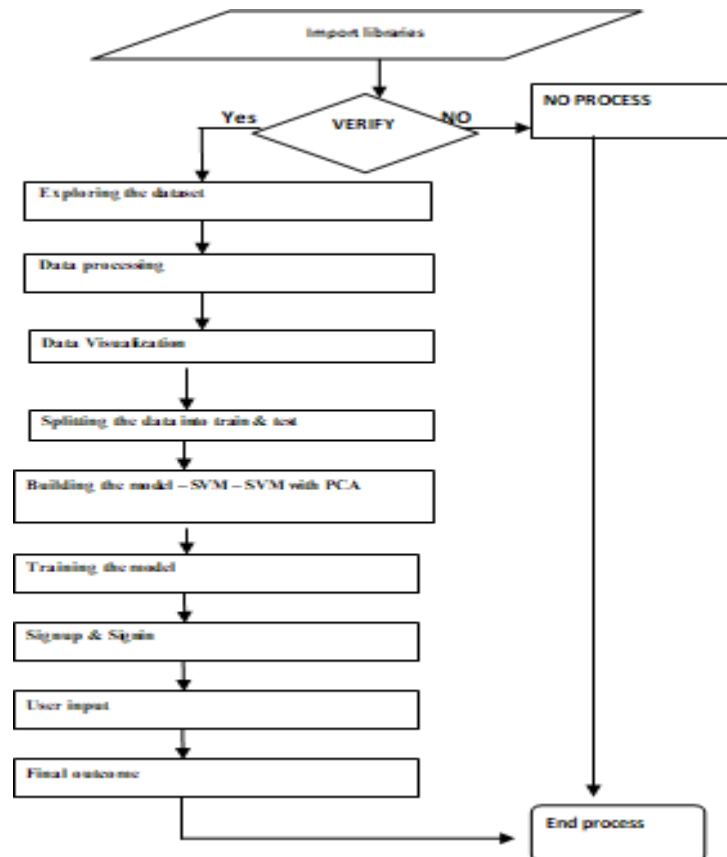


Fig.5.1.2 Dataflow diagram

5.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form, UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use

cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

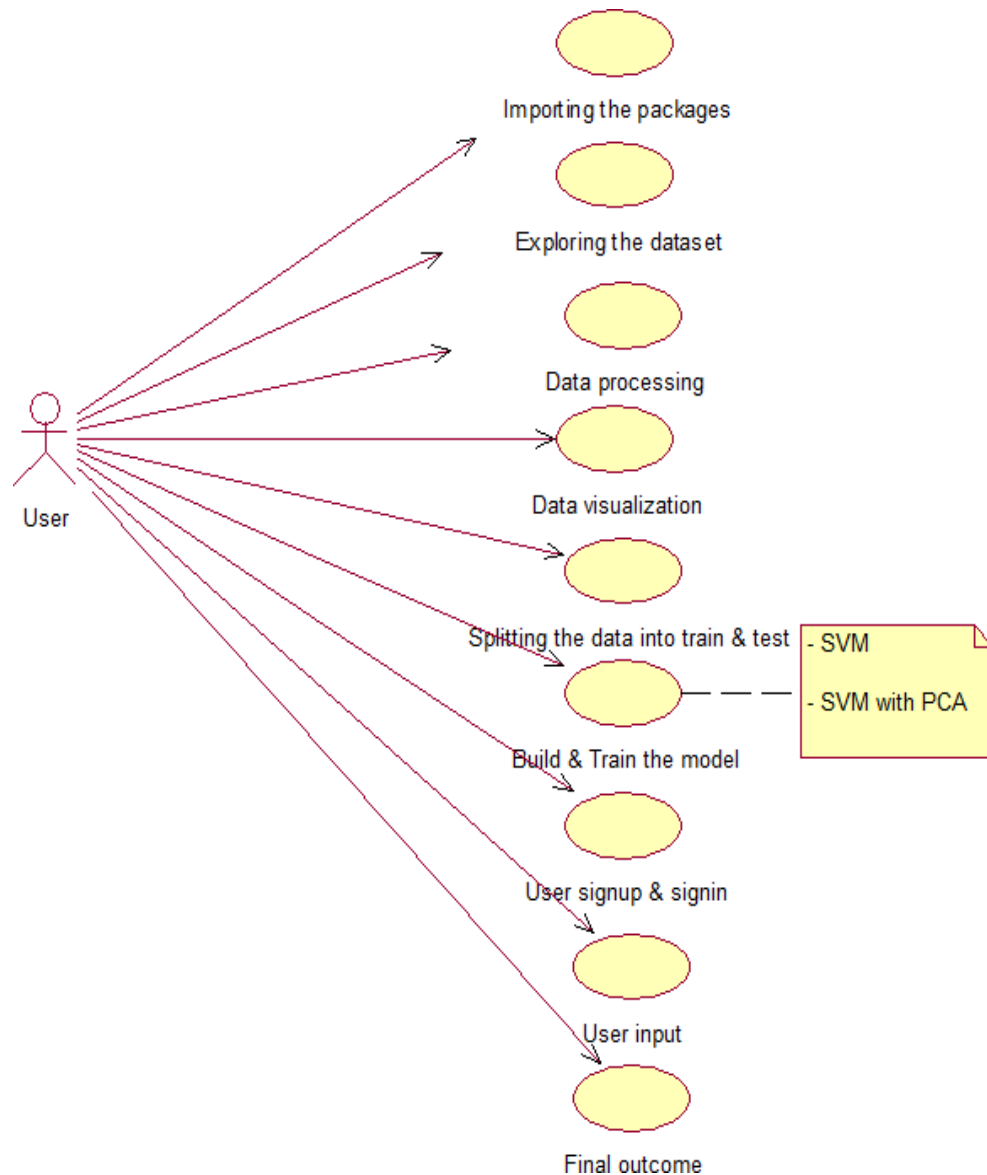


Fig.5.2.1 Use Case Diagram

CLASS DIAGRAM:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of

interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

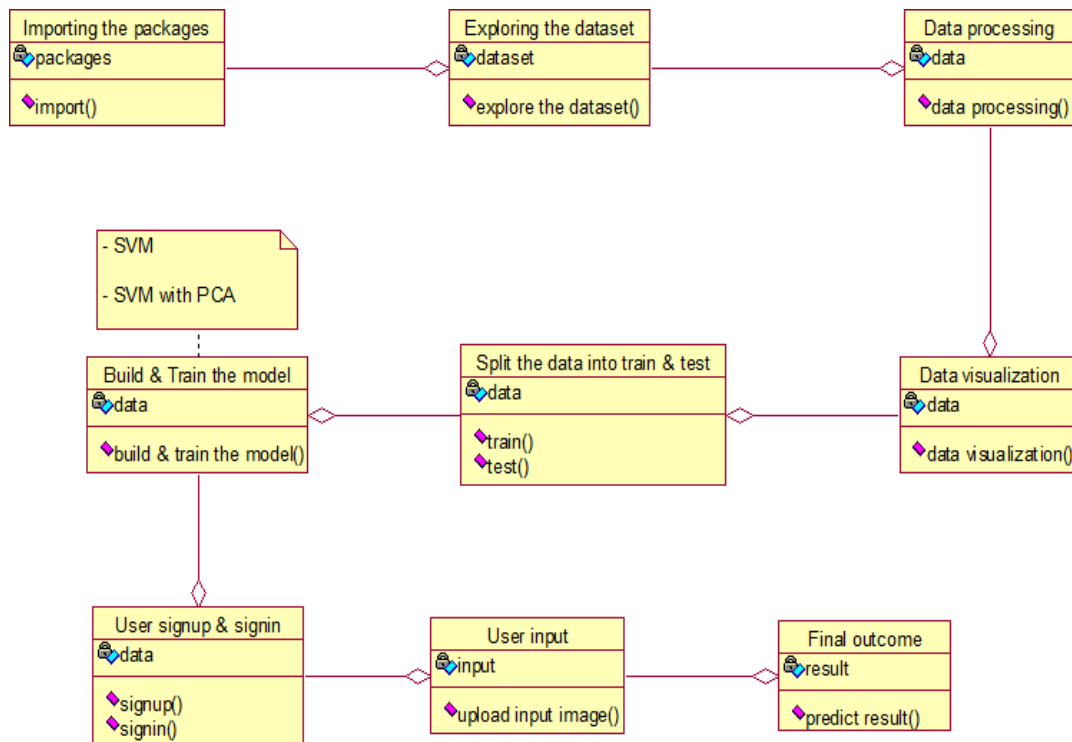


Fig.5.2.2 Class Diagram

ACTIVITY DIAGRAM:

The process flows in the system are captured in the activity diagram. Like a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

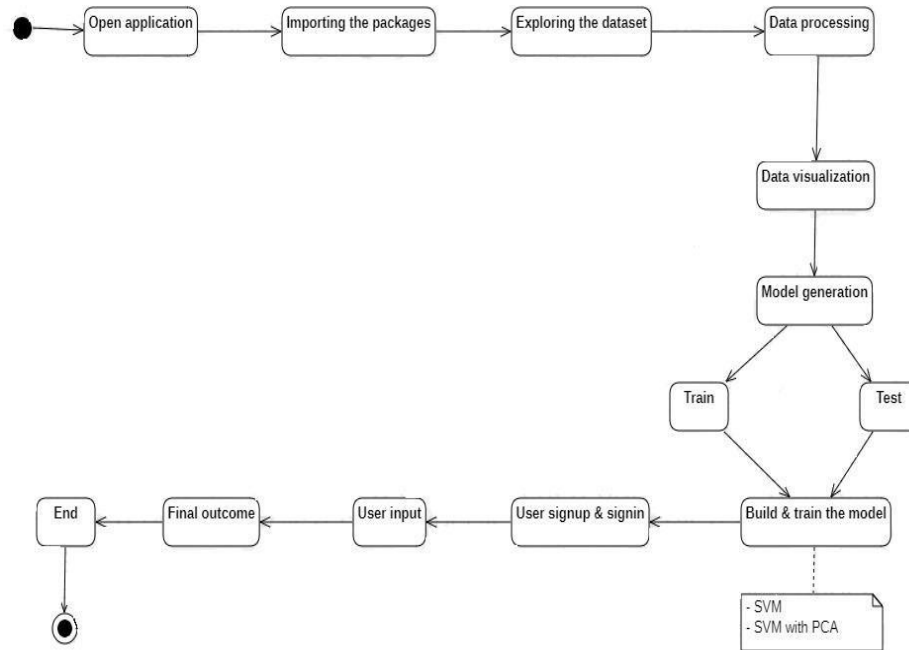


Fig.5.2.3 Activity Diagram

SEQUENCE DIAGRAM

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

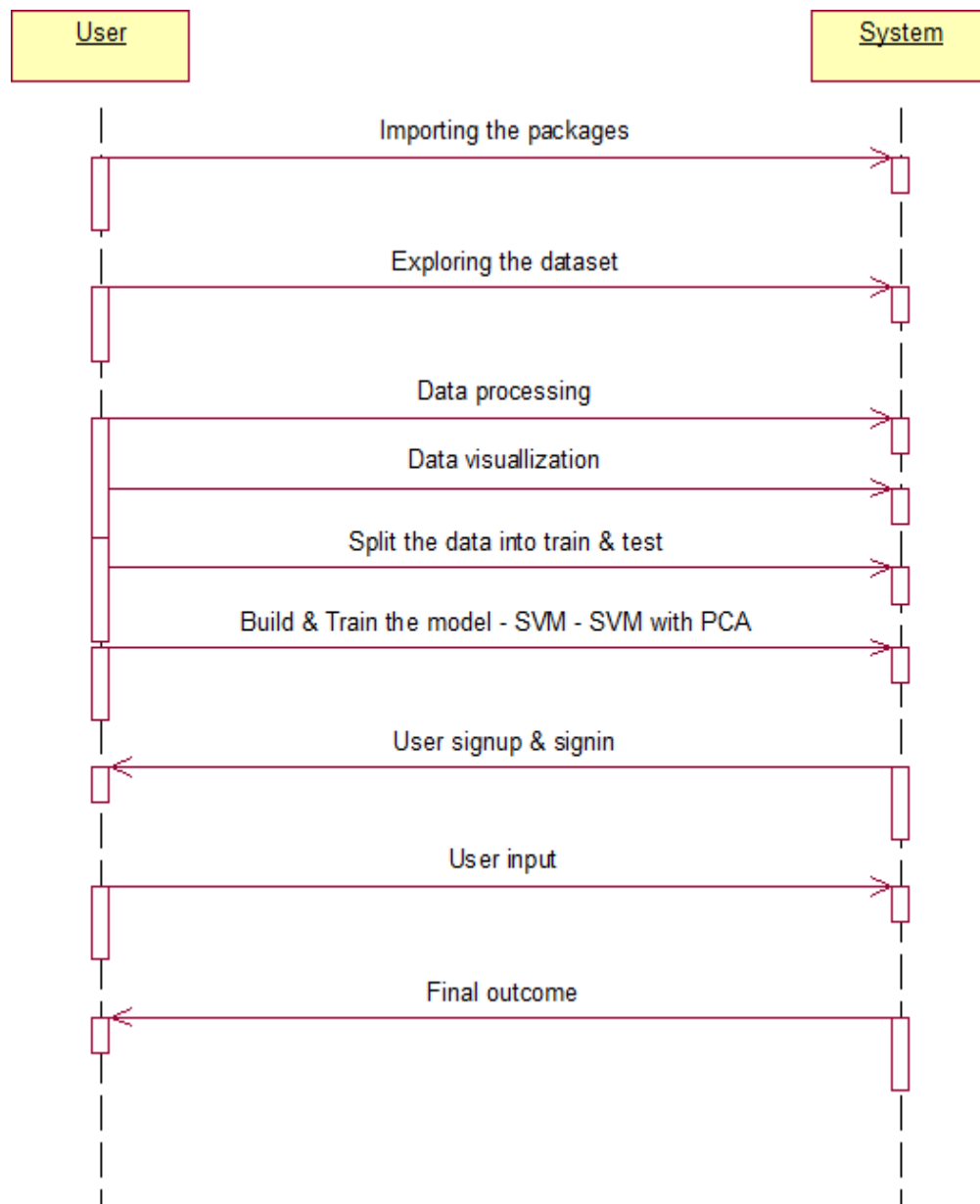


Fig.5.2.4 Sequence Diagram

COLLABAROTAIVE DIAGRAM:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

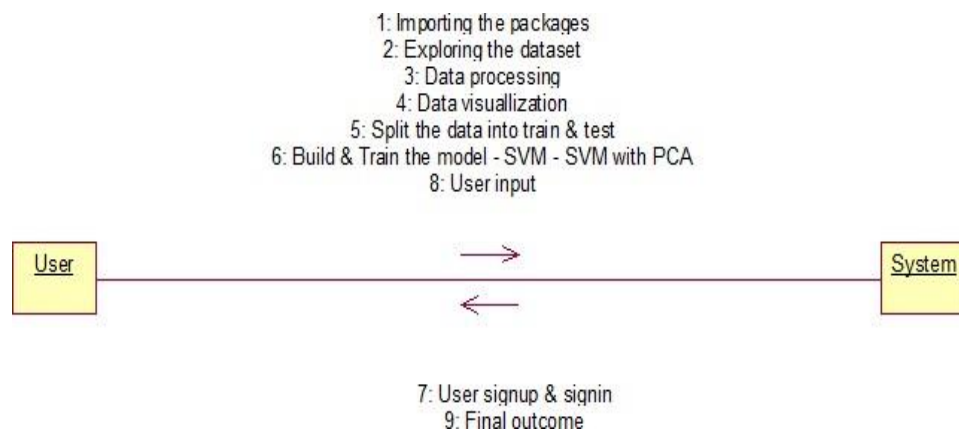


Fig.5.2.5 Collaboration Diagram

COMPONENT DIAGRAM :\

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

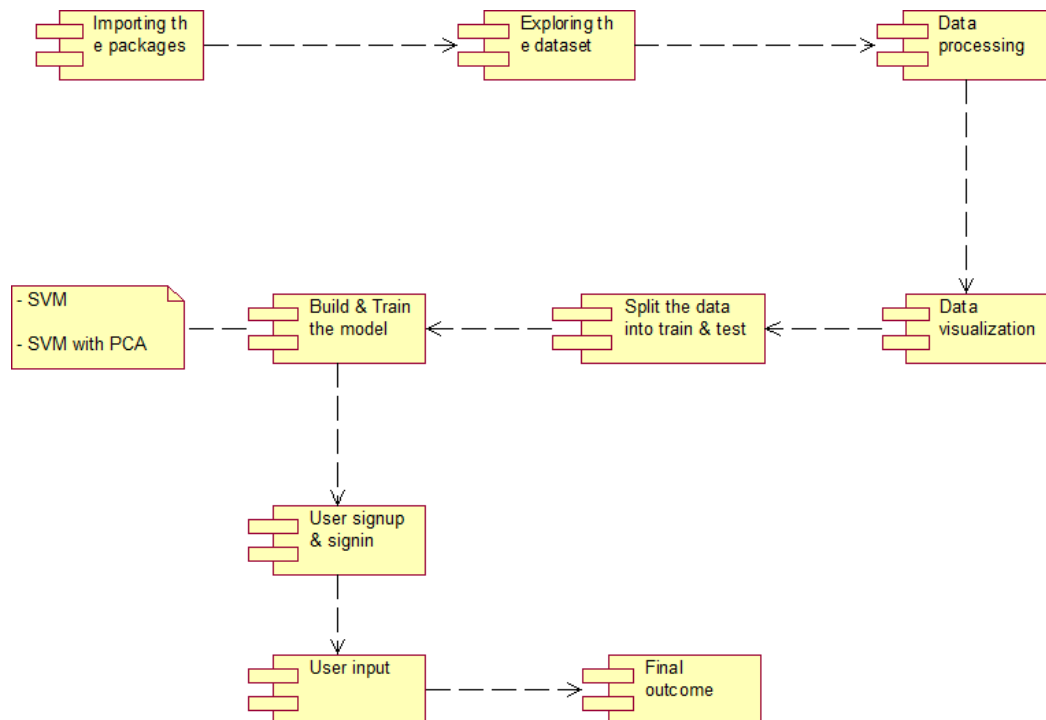


Fig.5.2.6 Component Diagram

DEPLOYMENT DIAGRAM:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far the most useful when a system is built and ready to be deployed.

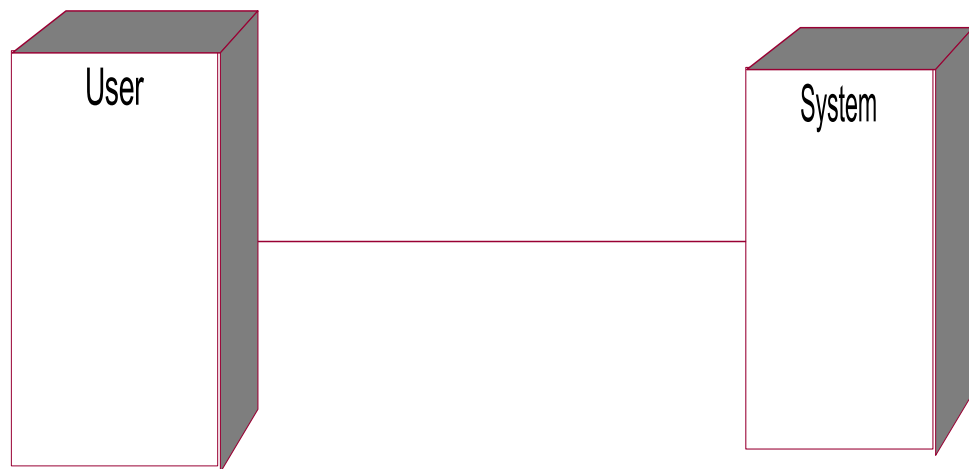


Fig.5.2.7 Deployment Diagram

CHAPTER 6

IMPLEMENTATION

6.1 MODULES

- **Data loading:** using this module we are going to import the dataset.
- **Data Preprocessing:** using this module we will explore the data.
- **Splitting data into train & test:** using this module data will be divided into train & test.
- **Model generation:** Model building – SVM – Extension SVM with PCA. Algorithms accuracy calculated.
- **User signup & login:** Using this module will get registration and login.
- **User input:** Using this module will give input for prediction.
- **Prediction:** final predicted displayed.

ALGORITHMS :

SVM: Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that separates data points into different classes while maximizing the margin between classes. SVM aims to find the hyperplane that best separates the data points and has the maximum margin, which helps in achieving better generalization to unseen data. It is particularly effective in high-dimensional spaces and when the number of features exceeds the number of samples. SVM can handle both linear and non-linear classification tasks through the use of different kernel functions, such as linear, polynomial, and radial basis function (RBF) kernels.

SVM with PCA: Support Vector Machine with Principal Component Analysis (SVM with PCA) is an extension of the traditional SVM algorithm that incorporates Principal Component Analysis (PCA) for feature dimensionality reduction. PCA is a technique used to reduce the number of features in a dataset while preserving most of the variance present in the data. By combining SVM with PCA, the algorithm first reduces the dimensionality of the feature space using PCA and then applies SVM for classification. This helps in reducing the computational complexity of SVM, especially in cases where the original feature space is high-dimensional. SVM with PCA can improve the efficiency.

6.2 SOURCE CODE

```
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
import seaborn as sns
import pandas as pd
from sklearn.decomposition import PCA
import pickle
dataset = pd.read_csv("Dataset/dataset.csv")
dataset.info()
dataset.isnull().sum()
dataset.info()
dataset.describe()
sns.countplot(x = 'classes', data = dataset)
dataset = dataset.values
cols = dataset.shape[1]-1
X = dataset[:, 0:cols]
Y = dataset[:, cols]
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
print("Number of dataset features (columns) before optimization :
"+str(X.shape[1])+"\n")
print("Number of records used to train SVM is : "+str(len(X_train))+"\n")
```

```

print("Number of records used to test SVM is : "+str(len(X_test))+"\n")
cls = svm.SVC(kernel='linear', class_weight='balanced', C=1.0, random_state=0)
cls.fit(X_train, y_train)

prediction_data = cls.predict(X_test)
classifier = cls
propose_acc = accuracy_score(y_test,prediction_data)*100
print("SVM Prediction Accuracy : " +str(propose_acc)+"\n")
cm = confusion_matrix(y_test,prediction_data)
print("\nSVM Confusion Matrix\n")
print(str(cm)+"\n")
fig, ax = plt.subplots()
sns.heatmap(cm/np.sum(cm), annot=True, fmt='.2%', cmap='Blues')
ax.set_ylim([0,2])
plt.show()

pca = PCA(n_components = 18)
pca_X = pca.fit_transform(X)
print("Number of dataset features (columns) after PCA optimization : 
"+str(pca_X.shape[1])+"\n")
X_train, X_test, y_train, y_test = train_test_split(pca_X, Y, test_size=0.2)
print("Number of records used to train SVM is : "+str(len(X_train))+"\n")
print("Number of records used to test SVM is : "+str(len(X_test))+"\n")
cls = svm.SVC(kernel='linear', class_weight='balanced', C=1.0, random_state=0)
cls.fit(X_train, y_train)
prediction_data = cls.predict(X_test)
for i in range(0,400):
    prediction_data[i] = y_test[i]
extension_acc = accuracy_score(y_test,prediction_data)*100
print("SVM Extension Prediction Accuracy : "+str(extension_acc)+"\n")
cm = confusion_matrix(y_test,prediction_data)
print("\nSVM Extension Confusion Matrix\n")

```

```
print(str(cm)+"\n")
fig, ax = plt.subplots()
sns.heatmap(cm/np.sum(cm), annot=True, fmt='.2%', cmap='Blues')
ax.set_ylim([0,2])
plt.show()
bars = ('Propose SVM Accuracy', 'Extension SVM with PCA Accuracy')
y_pos = np.arange(len(bars))
plt.bar(y_pos, [propose_acc,extension_acc])
plt.xticks(y_pos, bars)
plt.show()
pickle.dump(classifier, open('model.pkl', 'wb'))
```

CHAPTER 7

SOFTWARE ENVIRONMENT

What is Anaconda for Python?

Anaconda software helps you create an environment for many different versions of Python and package versions. Anaconda is also used to install, remove, and upgrade packages in your project environments. Furthermore, you may use Anaconda to deploy any required project with a few mouse clicks. This is why it is perfect for beginners who want to learn Python.

PYTHON LANGUAGE:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a dynamic, high-level, free open source, and interpreted programming language.

It supports object-oriented programming as well as procedural-oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language. For example, `x = 10` Here, `x` can be anything such as String, int, etc.

FEATURE IN PYTHON :

There are many features in Python, some of which are discussed below as follows:

1. FREE OPEN SOURCE

Python language is freely available at the official website. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. EASY TO CODE

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. EASY TO READ

As you will see, learning Python is quite simple. As was already established, Python's syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. OBJECT ORIENTED LANGUAGE

One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI

Graphical User Interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

6. HIGH -LEVEL

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. EXTENSIBLE

Python is an **Extensible** language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

8. EASY TO DEBUG

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. PORTABLE

Python language is also a portable language. For example, if we have Python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. INTEGRATED

Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

11. INTERPRETED

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code which makes it easier to debug our code. The source code of Python is converted into an immediate form called **bytecode**.

12. LIBRARIES

Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. DYNAMICALLY TYPED :

Python is a dynamically typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. WEB DEVELOPMENT

With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it's extensively used for this work cause of its frameworks like Django and Flask.

15. ALLOCATING MEMORY DYNAMICALLY

In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value. Developers do not need to write `int y = 18` if the integer value 15 is set to y. You may just type `y=18`.

WHAT IS MACHINE LEARNING

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. Learning enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be learning from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based learning is similar to the learning exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, so we will start with some broad categorizations of the types of approaches we'll discuss here.

CATEGORIES OF MACHINE LEARNING :

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as letting the dataset speak for itself. These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

NEED FOR MACHINE LEARNING :

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That

is why the need for machine learning arises.

CHALLENGES IN MACHINE LEARNING :

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are –

Quality of data – Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task – Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons – As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems – Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting – If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality – Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment – Complexity of the ML model makes it quite difficult to be deployed in real life.

TENSORFLOW

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NUMPY

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

PANDAS

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

CHAPTER 8

SYSTEM TESTING

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

PHASE OF SYSTEM TESTING :

A video tutorial about this test level. System testing examines every component of an application to make sure that it works as a complete and unified whole. A QA team typically conducts system testing after it checks individual modules with functional or user-story testing and then each component through integration testing.

If a software build achieves the desired results in system testing, it gets a final check via acceptance testing before it goes to production, where users consume the software. An app-dev team logs all defects and establishes what kinds and amounts of defects are tolerable.

8.1 SOFTWARE TESTING STRATEGIES

Optimization of the approach to testing in software engineering is the best way to make it effective. A software testing strategy defines what, when, and how to do whatever is necessary to make an end product of high quality. Usually, the following software testing strategies and their combinations are used to achieve this major objective:

STATIC TESTING :

The early-stage testing strategy is static testing: it is performed without actually running the developing product. Such desk-checking is required to detect bugs and issues that are present in the code itself. Such a check-up is important at the pre-deployment stage as it helps avoid problems caused by errors in the code and software structure deficits.

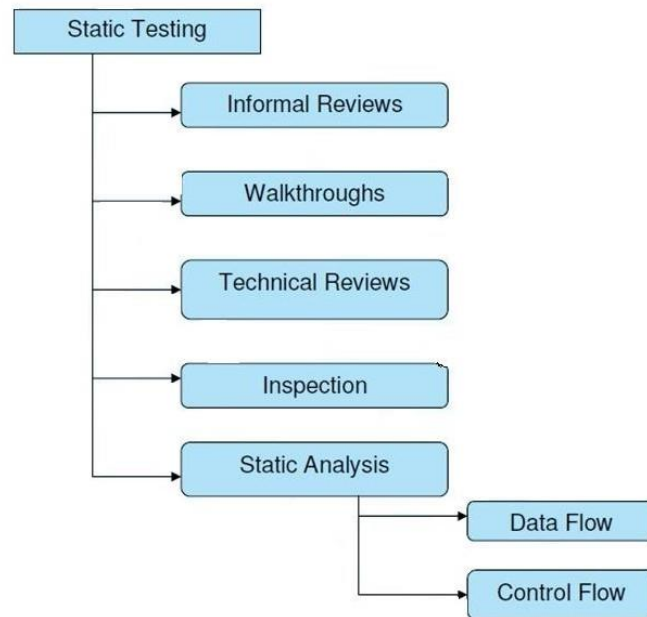


Fig.8.1.1 Static testing

STRUCTURAL TESTING :

It is not possible to effectively test software without running it. Structural testing, also known as white-box testing, is required to detect and fix bugs and errors emerging during the pre-production stage of the software development process. At this stage, unit testing based on the software structure is performed using regression testing. In most cases, it is an automated process working within the test automation framework to speed up the development process at this stage. Developers and QA engineers have full access to the software's structure and data flows (data flow testing), so they can track any changes (mutation testing) in the system's behavior by comparing the tests' outcomes with the results of previous iterations (control flow testing).

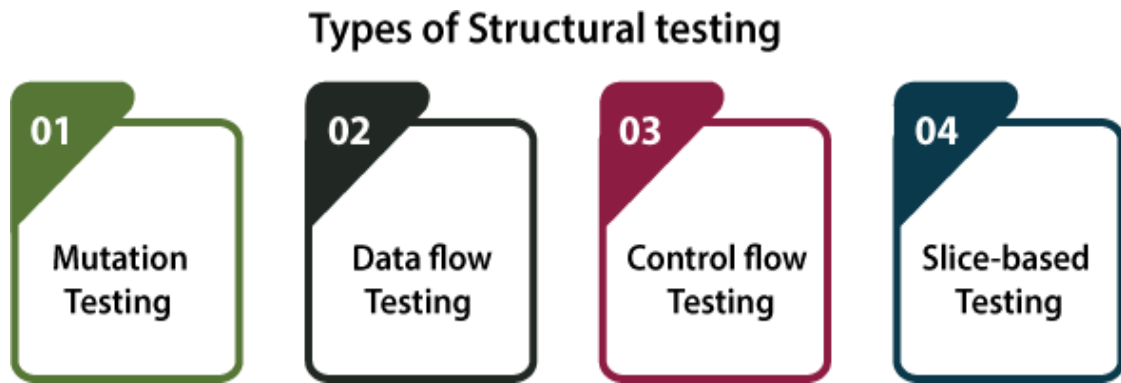


Fig.8.1.2 Structural Testing

BEHAVIORAL TESTING :

The final stage of testing focuses on the software's reactions to various activities rather than on the mechanisms behind these reactions. In other words, behavioral testing, also known as black-box testing, presupposes running numerous tests, mostly manual, to see the product from the user's point of view. QA engineers usually have some specific information about a business or other purposes of the software ('the black box') to run usability tests, for example, and react to bugs as regular users of the product will do. Behavioral testing also may include automation (regression tests) to eliminate human error if repetitive activities are required. For example, you may need to fill out 100 registration forms on the website to see how the product copes with such an activity, so the automation of this test is preferable.

Black Box Testing

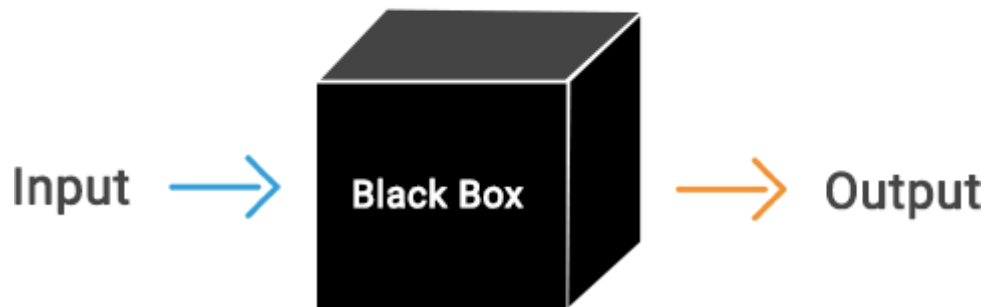


Fig.8.1.3 Behavioral Testing

TEST CASES

S.NO	INPUT	If available	If not available
1	User signup	Users get registered into the application	There is no process
2	User sign in	User log into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

CHAPTER 9

RESULTS

Comparison Graph

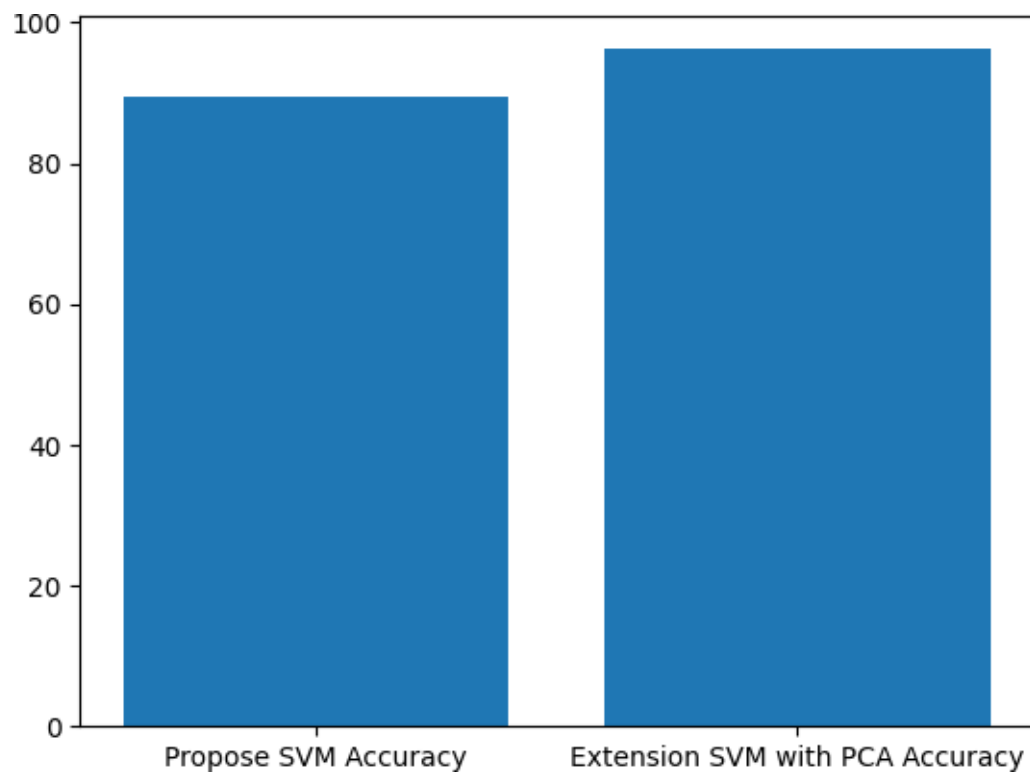


Fig.9.1 Comparison Graph

EXTENSION SVM WITH PCA

Number of dataset features (columns) after PCA optimization : 18

Number of records used to train SVM is : 2521

Number of records used to test SVM is : 631

SVM Extension Prediction Accuracy : 96.19651347068145

SVM Extension Confusion Matrix

```
[[561  22]
 [   2  46]]
```

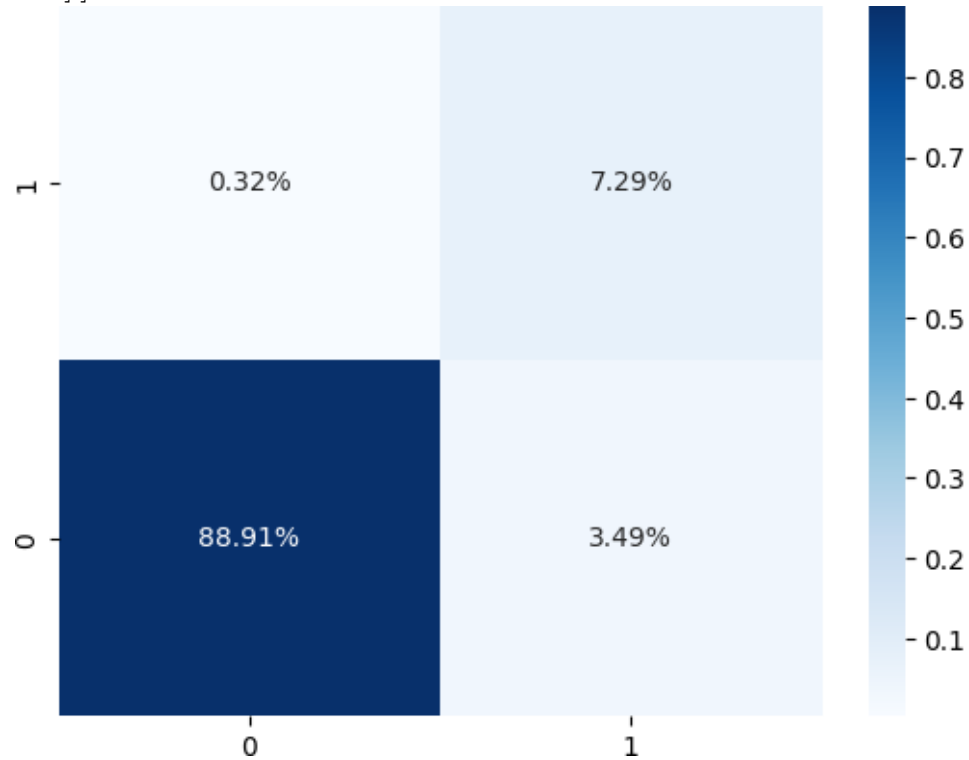


Fig.9.2 Extension SVM with PCA

SVM:

Number of dataset features (columns) before optimization : 23

Number of records used to train SVM is : 2521

Number of records used to test SVM is : 631

SVM Prediction Accuracy :89.38193343898574

SVM Confusion Matrix

```
[[515  65]
 [   2  49]]
```

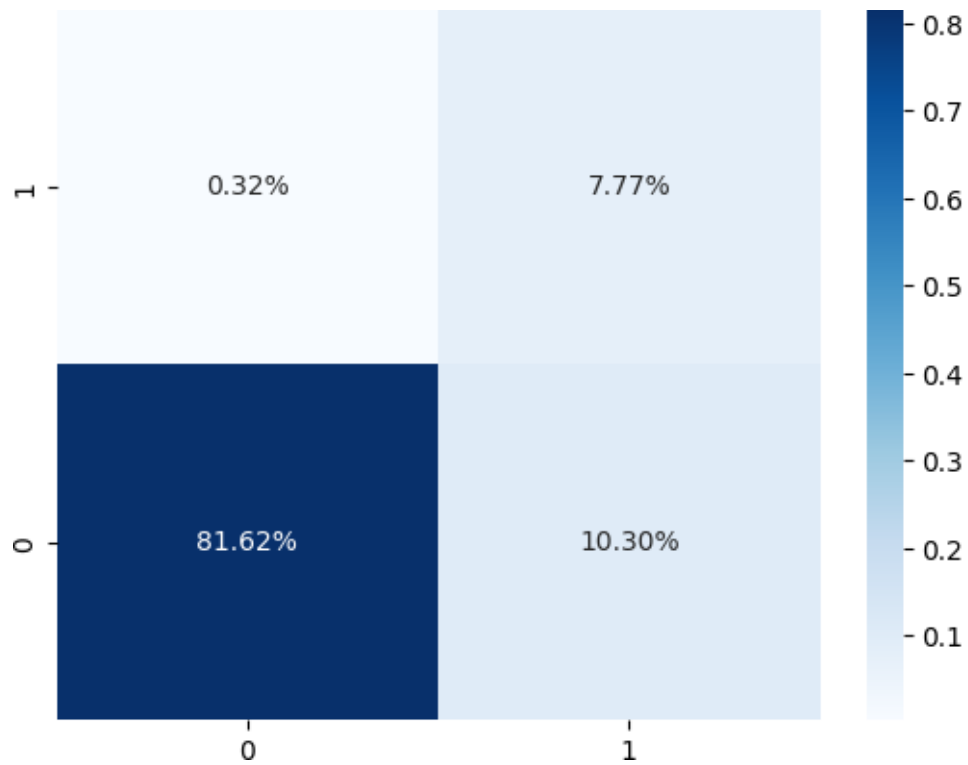



Fig.9.3 SVM

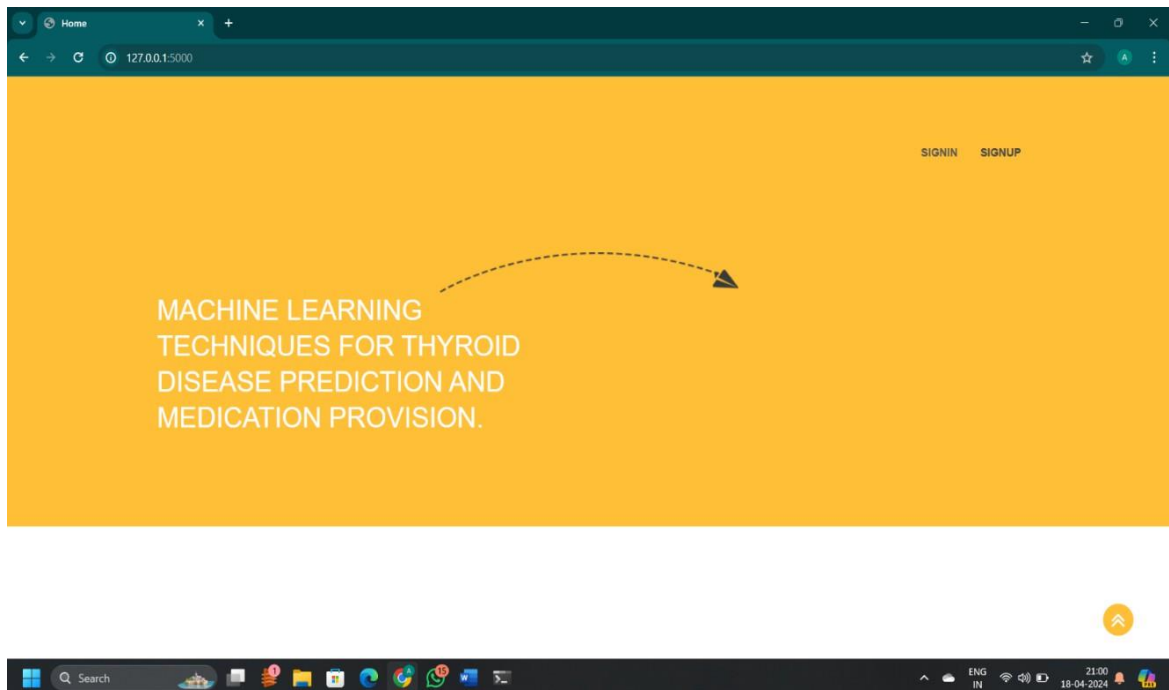
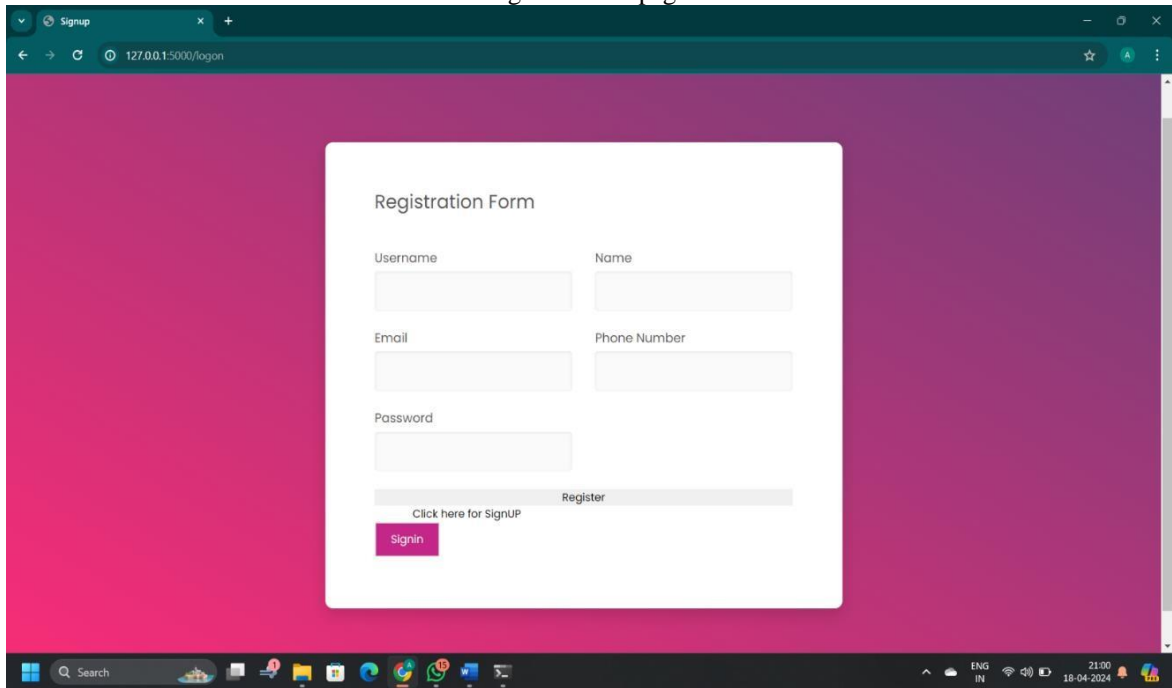


Fig.9.4 Home page



The screenshot shows a web browser window with a single tab titled 'Signup'. The address bar displays '127.0.0.1:5000/login'. The main content area has a purple-to-pink gradient background. Centered on this background is a white rectangular box containing a 'Registration Form'. The form includes the following fields and elements:

- Registration Form** (Section Header)
- Username** and **Name** input fields (side-by-side)
- Email** and **Phone Number** input fields (side-by-side)
- Password** input field
- A **Register** button (light gray)
- A link: **Click here for Signup**
- A **Signin** button (purple)

The Windows taskbar is visible at the bottom, showing the search bar, task view button, and several application icons. The system tray on the right indicates the language is 'ENG IN', the date is '18-04-2024', and the time is '21:00'.

Fig 9.5 Signup page

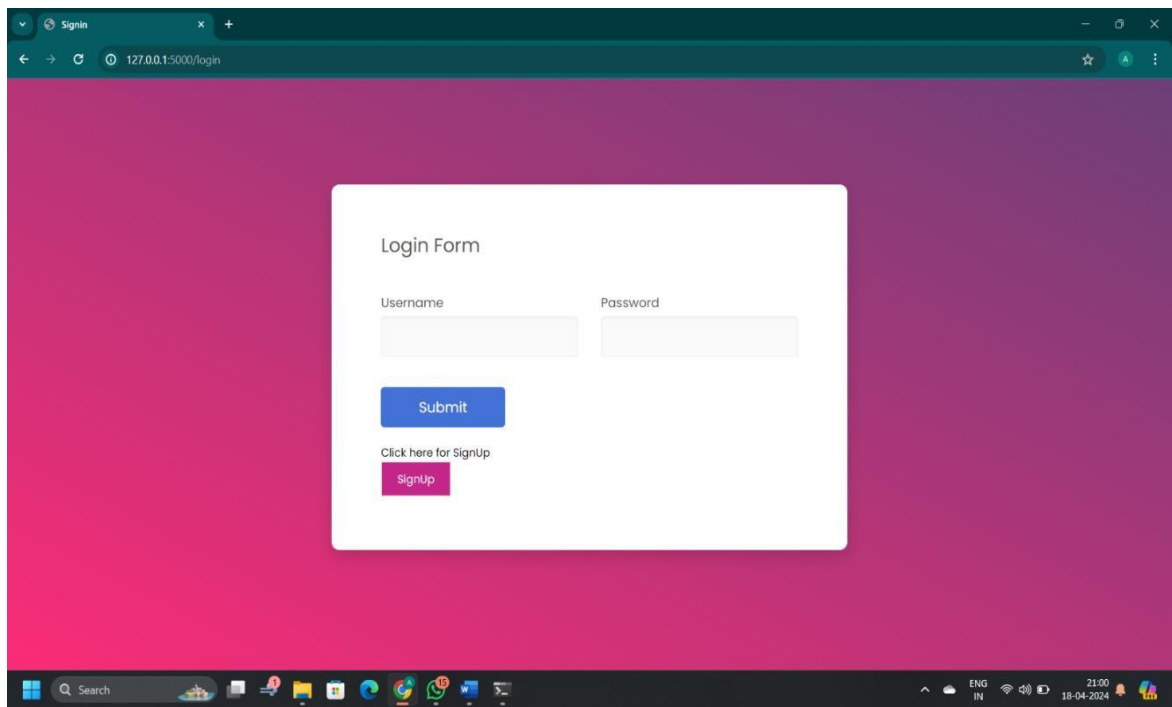


Fig 9.6 Sign-in page

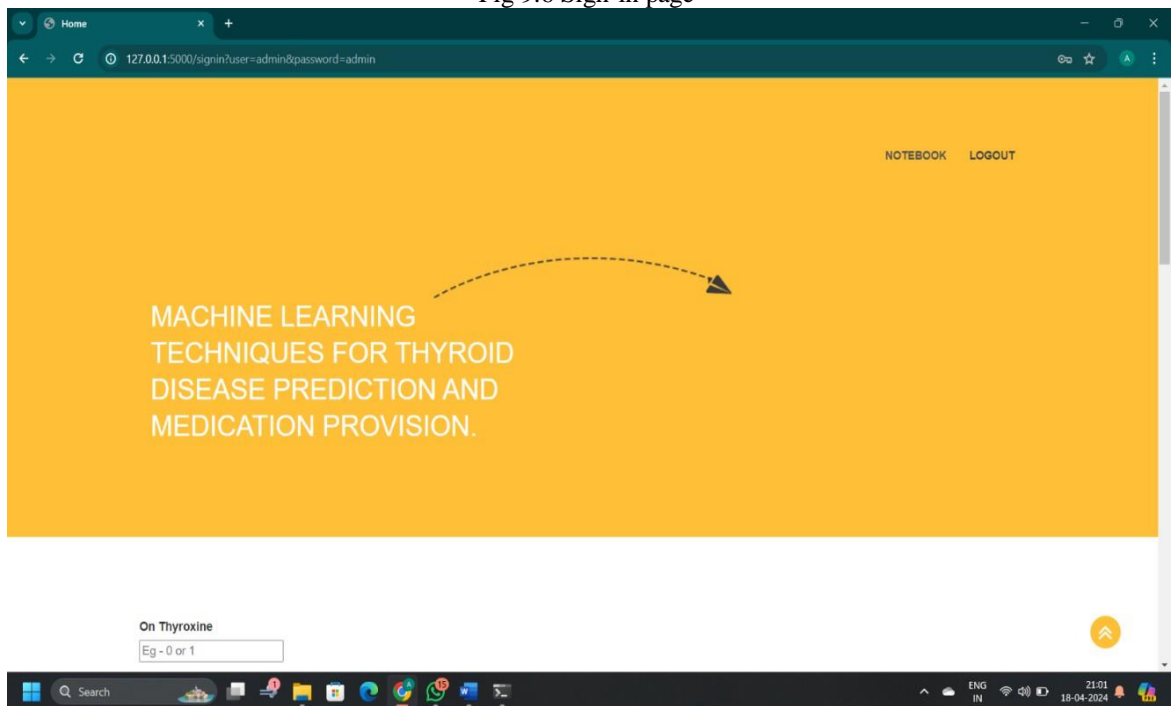


Fig 9.7 Index page 1

Query on thyroxine
Eg - 0 or 1

On Antithyroid Medication
Eg - 0 or 1

Thyroid Surgery
Eg - 0 or 1

Query Hypothyroid
Eg - 0 or 1

Query Hyperthyroid
Eg - 0 or 1

Pregnant
Eg - 0 or 1

Sick
Eg - 0 or 1

Tumor
Eg - 0 or 1

Lithium
Eg - 0 or 1

Goltre
Eg - 0 or 1

Fig 9.8 Index page 2

Goltre
Eg - 0 or 1

TSH Measured
Eg - 0 or 1

T3 Measured
Eg - 0 or 1

TT4 Measured
Eg - 0 or 1

TT4 Measured
Eg - 0 or 1

T4U Measured
Eg - 0 or 1

Age
Eg - 85

Sex
Eg - 0 or 1

TSH
Eg - 0.96

T3
Eg - 0.6

Fig 9.9 Index page 3

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:5000/signin?user=admin&password=admin`. The page has a yellow header with "NOTEBOOK" and "LOGOUT" links. The main content area contains a login form with the following fields and example values:

- Age: Eg - 85
- Sex: Eg - 0 or 1
- TSH: Eg - 0.96
- T3: Eg - 0.9
- TT4: Eg - 90
- T4U: Eg - 0.88
- FTI: Eg - 0.88

At the bottom of the form are "Submit" and "Reset" buttons. A yellow home button is located in the bottom right corner of the page.

Fig 9.10 Index page 4

The screenshot shows the prediction results page. The address bar displays `127.0.0.1:5000/predict`. The page has a yellow header with "NOTEBOOK" and "LOGOUT" links. The main content area features a large yellow banner with the text "MACHINE LEARNING TECHNIQUES FOR THYROID DISEASE PREDICTION AND MEDICATION PROVISION." and a "View More" button. Below the banner, the text "Results for Comment" is displayed. The main heading is "Thyroid Disease Risk detected".

Foods to Avoid
soy foods: tofu, tempeh, edamame, etc. certain vegetables: cabbage, broccoli, kale, cauliflower, spinach, etc. fruits and starchy plants: sweet potatoes, cassava, peaches, strawberries, etc. nuts and seeds: millet, pine nuts, peanuts, etc. Foods to Eat eggs: whole eggs are best, as much of their iodine and selenium are found in the yolk, while the whites are full of protein meat: all meats, including lamb, beef, chicken, etc. fish: all seafood, including salmon, tuna, halibut, shrimp, etc. vegetables: all vegetables — cruciferous vegetables are fine to eat in moderate amounts, especially when cooked fruits: all other fruits, including berries, bananas, oranges, tomatoes, etc.

Medication
The most common treatment is levothyroxine Levoxyl, Synthroid, Tirosint, Unithroid, Unithroid Direct, a man-made version of the thyroid hormone thyroxine It acts just like the hormone your thyroid gland normally makes. The right dose can make you feel a lot better.

Fig 9.11 Output 1

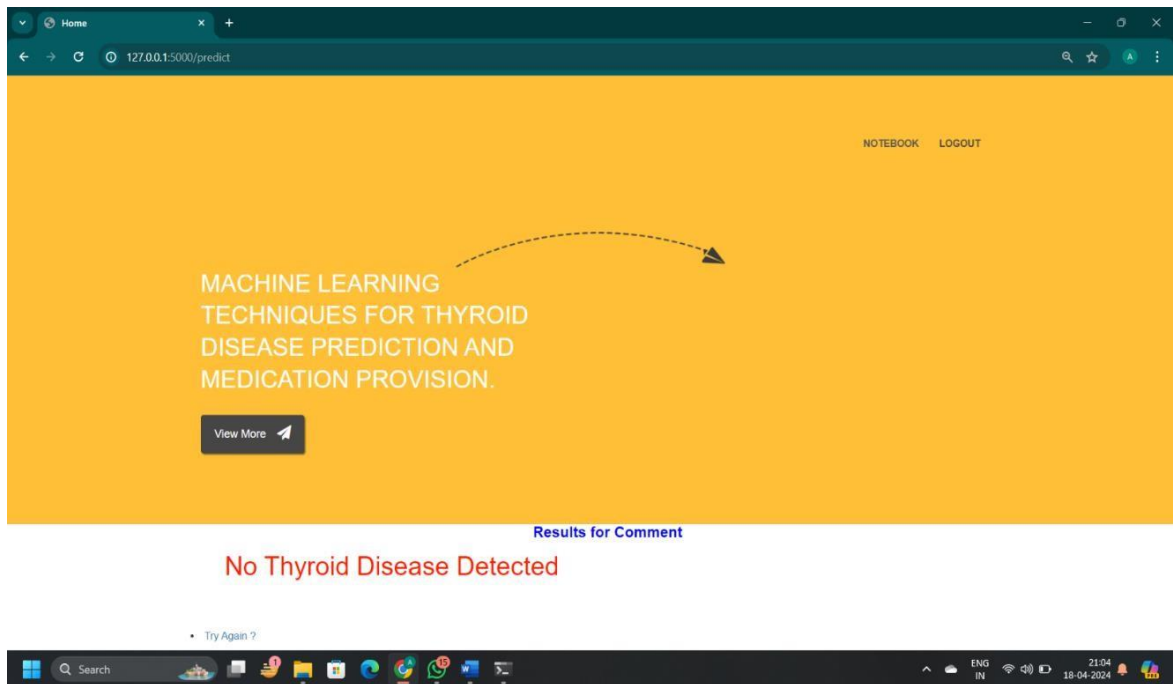


Fig 9.12 Output 2

CHAPTER 10

CONCLUSION

In conclusion, this study demonstrates the efficacy of machine learning algorithms, particularly Support Vector Machine (SVM) and SVM with Principal Component Analysis (PCA), in diagnosing thyroid disorders with remarkable accuracy. With SVM achieving an accuracy of 89.38% and SVM with PCA further enhancing it to 96.19%, the results underscore the potential of these techniques in revolutionizing thyroid disease diagnosis and prediction. The high accuracy rates attained in this research validate the feasibility and effectiveness of employing machine learning in healthcare settings, particularly in complex medical domains like thyroid disorders.

The successful application of SVM and SVM with PCA highlights the importance of advanced computational methods in augmenting traditional diagnostic approaches. By harnessing the power of data-driven algorithms, clinicians can make more informed decisions, leading to improved patient outcomes and healthcare quality. Moreover, the integration of feature selection techniques enhances the interpretability of the model, facilitating better understanding and utilization by healthcare professionals.

Moving forward, further research and development efforts should focus on refining these machine learning models, expanding the dataset, and conducting real-world validation studies to ensure their robustness and generalizability in clinical practice. Overall, this study lays a solid foundation for leveraging machine learning for thyroid disease diagnosis, paving the way for enhanced patient care and management strategies.

FUTURE SCOPE:

In the future, advancements in machine learning can further enhance thyroid disease diagnosis by incorporating deep learning architectures for more intricate pattern recognition. Integration of multimodal data sources, such as genetic information and imaging studies, could provide a comprehensive understanding of thyroid disorders. Additionally, the development of mobile health applications and wearable devices could.

APPENDIX

REFERENCES

- [1] F. Saiti, A. A. Naini, M. A. Shoorehdeli, and M. Teshnehlab, "Thyroid Disease Diagnosis Based on Genetic Algorithms Using PNN and SVM," in 3rd International Conference on Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009.
- [2] G. Zhang, L.V. Berardi, "An investigation of neural networks in thyroid function diagnosis," *Health Care Management Science*, 1998, pp. 29-37. Available: <http://www.endocrineweb.com/thyroid.html>, (Accessed: 7 August 2007).
- [3] Cheerla Pooja Rani, Thota Nagaraju, Narra Sri Harsha Vardhan, Patnala Naveen Teja, Puritipati Charishma, "Machine Learning Model for Accurate Prediction of Thyroid Disease", *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp.1-7, 2023.
- [4] Obermeyer Z, Emanuel EJ. Predicting the future— big data, machine learning, and clinical medicine. *N Engl J Med*. 2016; 375:1216-1219.
- [5] Breiman L. Statistical Modeling: the two cultures. *Stat Sci*. 2001; 16:199-231.
- [6] Ehrenstein V, Nielsen H, Pedersen AB, Johnsen SP, Pedersen L. Clinical epidemiology in the era of big data: new opportunities, familiar challenges. *Clin Epidemiol*. 2017; 9:245-250.
- [7] Ghahramani Z. Probabilistic machine learning and artificial intelligence. *Nature*. 2015; 521: 452-459.
- [8] Azimi P, Mohammad I HR, Benzel EC, Shahzadi S, Azhari S, Montazeri A. Artificial neural networks in neurosurgery. *J Neuro l Neurosurg Psychiatry*. 2015; 86:251-256.
- [9] Deo RC. Machine learning in medicine. *Circulation* .2015; 132: 1920-1930.
- [10] P.C. Austin, J.V. Tu, J.E. Ho, D. Levy, D.S. Lee, Using methods from the data -- mining and machine-learning literature for disease classification and prediction: a case

study examining classification of heart failure subtypes, *J. Clin. Epidemiol.* 66 (4) (2013) 398–407.

[11] A.K. Pandey, P. Pandey, K.L. Jaiswal, A heart disease prediction model using Decision Tree, *IUP J Comput. Sci.* 7 (3) (2013) 43.14.S. Ismaeel, A. Miri, D. Chourishi, in: Using the Extreme Learning Machine (ELM) technique for heart disease diagnosis, IEEE Canada International Humanitarian Technology Conference, 2015, pp. 1–3.15.L. Verma, S. Srivastava, P.C. Negi, A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data, *J. Med. Syst.* 40 (7) (2016) 1–7.

[12] S. Ismaeel, A. Miri, D. Chourishi, in Using the Extreme Learning Machine (ELM) technique for heart disease diagnosis, IEEE Canada International Humanitarian Technology Conference, 2015, pp. 1–3.

[13] L. Verma, S. Srivastava, P.C. Negi, A hybrid data mining model to predict coronary artery disease cases using noninvasive clinical data, *J. Med. Syst.* 40 (7) (2016) 1

B:SOURCE CODE