A PROJECT REPORT

on

EARLY DETECTION: MACHINE LEARNING TECHNIQUES IN PANCREATIC CANCER DIAGNOSIS

`Submitted in partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

MALLIPUDI DEVI SIVA SAI	-	**20BQ1A1299**
PALAPARTHI PRUDHVI	-	**20BQ1A12C4**
GOLLAPUDI M N V SAI GOPI	-	**20BQ1A1264**
INDLA GANESWARA NAGA SAI RAM	-	**20BQ1A1266**
MANDADI RAM SANDEEP	-	**20BQ1A12A2**

Under the Supervision of

Mr. NAGA BABU PACHHALA **M. Tech(Ph.D);**

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508, TEL no: 0873 2118036,

www.vvitguntur.com, approved by AICTE, permanently affiliated to JNTUK

Accredited by NAAC with “A” grade, Accredited by NBA for 3 years

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY:: NAMBUR

VASIREDDY VENKATADRI
INSTITUTE OF TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the project report “EARLY DETECTION: MACHINE LEARNING TECHNIQUES IN PANCREATIC CANCER DIAGNOSIS” is the Bonafide work done by “M. Devi Siva Sai (20BQ1A1299), P. Prudhvi (20BQ1A12C4), G. Naga Venkata Sai Gopi(20BQ1A1264), I. Ganeswara Naga Sai Ram(20BQ1A1266), M. Ram Sandeep (20BQ1A12A2)”, who carried out the project under my guidance during the year 2024 towards partial fulfillment of the requirementsof the Degree of Bachelor of Technology in Information Technology from Vasireddy Venkatadri Institute of Technology, Nambur. The results embodied in this report have not beensubmitted to any other University for the award of any degree.

Signature of the Supervisor	Signature of the Head of the Department
NAGA BABU PACHHALA **Assistant Professor, IT.**	**Dr KALAVATHI ALLA** **Professor, IT.**

Submitted for Viva voce Examination held on ______________

EXTERNAL EXAMINER

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY:: NAMBUR

CERTIFICATE OF AUTHENTICATION

I solemnly declare that this project report **"EARLY DETECTION: MACHINE LEARNING TECHNIQUES IN PANCREATIC CANCER DIAGNOSIS"** is Bonafide work done purely by me/us, carriedout under the supervision of Mr. Naga Babu Pachhala, towards partial fulfillment of the requirements ofthe Degree of Bachelor of Technology in Information Technology from Vasireddy VenkatadriInstitute of Technology, Nambur during the year 2023-24. It is further certified that this work has not been submitted, either in part or in full, to any other department of the Vasireddy Venkatadri Institute of Technology, or any other University, institution, or elsewhere, or for publication in any form.

Signature of the Student

M. DEVI SIVA SAI	**– 20BQ1A1299**
P. PRUDHVI	**– 20BQ1A12C4**
G. M N V SAI GOPI	**– 20BQ1A1264**
I. G N SAI RAM	**– 20BQ1A1266**
M. RAM SANDEEP	**– 20BQ1A12A2**

ACKNOWLEDGEMENT

We take this opportunity to express our deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and help metowards the successful completion of this project work.

First and foremost, we express our deep gratitude to **Sri. Vasireddy Vidya Sagar, Chairman,** Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the Information Technology program.

We express our sincere thanks to **Dr. Y. Mallikarjuna Reddy, Principal,** Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the Information Technology program.

We express our sincere gratitude to **Dr. A. Kalavathi, Professor & HOD,** Information Technology, Vasireddy Venkatadri Institute of Technology for her constant encouragement, motivation and faith by offering different places to look to expand my ideas. We would like to express our sincere gratefulness to our guide **Mr. Naga Babu Pachhala, Assistant Professor** for his/her insightful advice, motivating suggestions, invaluableguidance, help and support in successful completion of this project and also our project coordinator **Mr. K. Kranthi Kumar, Associate Professor** for her advice and support. Wewould like to take this opportunity to express our thanks to the teaching and nonteaching staff in Department of Information Technology, VVIT for their invaluable help and support.

.

M. Devi Siva Sai
P. Prudhvi
G. M N V Sai Gopi
I. G Naga Sai Ram
M. Ram Sandeep

ABSTRACT

Pancreatic cancer is a malignant tumor that poses a significant threat to patients' lives. Malignant growth is the abnormal development of cell tissue. Pancreatic illness is one of the most obvious causes of mortality across the world. Pancreatic malignant development begins in the pancreatic tissues. The pancreas secretes proteins that aid in digestion as well as hormones that direct sugar breakdown. Pancreatic cancer is typically identified in its late stages, spreads quickly, and has a terrible prognosis. Biomarkers are critical in the management of patients with invasive malignancies. Pancreatic Ductal Adenocarcinoma has a dismal prognosis due to its advanced appearance and limited treatment choices. This is compounded by the lack of validated screening and predicting biomarkers for early detection and precision therapy, respectively. In this paper, we have attempted to discuss various Machine Learning methods to detect pancreatic cancer. The selected. urinary biomarkers values are provided as the input of Support Vector Machine (SVM), Extra Tree Classifier (ETC), Decision Tree (DT), and Random Forest (RF) methods. The diagnosing accuracy of pancreatic cancer using SVM, ETC, DT, and RF classifiers are 50, 82.16, 81.03, and 86 respectively. The experimental results prove that the Random Forest classifier is more feasible and promising for clinical applications for the diagnosis of pancreatic cancer when compared to ETC, DT, and SVM.

.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Introduction **01**

1.2 Need of the Project **02**

2 SYSTEM ANALYSIS

2.1 Requirement Analysis. **03**

2.1.1 *Nonfunctional requirements*.. **03**

2.2 Hardware requirements. **03**

2.3 Software requirements. **04**

2.4 Existing System. **04**

2.5 Proposed System. **04**

2.6 Modules. **05**

3 SYSTEM DESIGN

3.1 System Architecture. **09**

3.2 UML Diagrams. **09**

3.2.1 *Use case diagram.* **11**

3.2.2 *Sequence diagram* **12**

3.2.3 *Class diagram.* **14**

4 SYSTEM IMPLEMENTATION

4.1 Technologies Used. **16**

4.1.1 *Python.* **16**

4.1.2 *Anaconda Terminal.* **17**

4.1.3 *Jupyter Notebook.* **19**

4.2 Algorithm. **21**

4.2.1 *Random Forest Classifier* **21**

4.2.2 *Decision Tree Classifier* **21**

4.2.3 *Etra Trees Classifier*... **21**

4.2.4 *Support Vector Machine Classifier* **21**

4.3 Sample Code. **21**

4.4 Output Screens. **35**

5 SYSTEM TESTING

5.1 Purpose of Testing. **38**

5.2 Testing Strategies. **38**

5.2.1 *Unit Testing.* **39**

5.2.2 *Integration Testing.* ***39***

5.2.3 *System Testing*. ***40***

6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion. **41**

6.2 Future Scope. **41**

APPENDIX **42**

LIST OF FIGURES

S. NO	TITLE	PAGE NO
2.1	Importing the Libraries	05
2.2	Read The DataSet	06
3.1	System Architecture	09
3.2	Use Case diagram	12
3.3	Sequence Diagram	14
3.4	Class diagram	15
4.1	Jupyter Notebook	19
4.2	Jupyter Notebook Interface	20
4.3	Model Comparsion	35
4.4	Performance Analysis	35
4.5	Home Page	36
4.6	Predict Page	36
4.7	Submit Page	37
5.1	Types of Testing	39

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Pancreatic cancer (PC) is a highly malignant tumor of the digestive system that provides significant hurdles in both early detection and subsequent therapy. In 2020, around 57,600 persons were diagnosed with PC, and 47,050 died from it. This makes PC an incurable disease. PCs continue to be widely used in poor nations [1]. As a result, complete PC diagnosis and staging are very crucial, as they may assist doctors provide the best therapy regimen for PC and allow patients to obtain early medical therapies before severe PC develops. PC is a disorder that causes malignant (cancerous) cells to develop in pancreatic tissues. The pancreas is a gland that sits behind the stomach and in front of the spine. The pancreas generates digestive juices and hormones that help regulate blood sugar levels. Exocrine pancreatic cells generate digestive fluids, whereas endocrine pancreatic cells create hormones. The majority of PCs begin in exocrine cells. PC can be treated with surgery, chemotherapy, or radiation therapy. Chemotherapy utilizes medications to treat cancer, whereas radiation treatment employs X-rays or other types of radiation to destroy cancer cells. Surgery is done to remove tumors or cure PC symptoms.

Pancreatic cancer, one of the most aggressive and lethal forms of cancer, poses significant challenges to early detection and effective treatment. According to recent statistics, pancreatic cancer has one of the lowest survival rates among major cancers, with less than 10% of patients surviving beyond five years from diagnosis. This alarming statistic underscores the urgent need for innovative approaches to improve early detection and prognosis.

Machine learning, a subset of artificial intelligence, has emerged as a promising tool in the field of medical diagnostics and prognostics. With its ability to analyze vast amounts of data and identify complex patterns, machine learning offers a new paradigm for early detection of pancreatic cancer. By leveraging advanced algorithms and computational techniques, machine learning models can analyze medical images, genetic markers, and clinical data to detect subtle signs of pancreatic cancer at its earliest stages, when treatment is most effective.

According to the American Cancer Society, only around 23% of people with exocrine pancreatic cancer survive a year following diagnosis. Five years after their diagnosis, around 8.2% are still living. Early identification of PC is challenging, hence many PC cases are detected late. When PC is discovered, the cancer is typically advanced. Machine learning is a branch of artificial intelligence that can identify PCs early.

1.2 NEED OF THE PROJECT

PC is becoming a leading cause of cancer related death in societies. Rapid and accurate diagnosis of a pancreatic mass is crucial for improving outcomes. Early detection of PC is challenging because cancer-specific symptoms occur only at an advanced stage, and a reliable screening tool to identify high-risk patients is lacking. Machine learning technique is a better way to address this challenge. There are exciting developments of new diagnostic techniques that open the possibility of personalised cancer medicine

CHAPTER II

SYSTEM ANALYSIS

2.1 REQUIREMENT ANALYSIS

2.1.1 Non-Functional Requirements

Performance Requirements:

The system must be interactive and the delays involved must be less if we test with small amount of data. So, in every action- response of the system, there are no immediate delays with less test set.

Software Quality Attributes:

Accuracy

Based on performance, this project determines the fake/generated images of human faces and it also predicts the best classifier by analyzing with the dataset. If the accuracy rate is high, then the performance of the discriminator is also high.

Reliability

This project predicts the output with less error-rate. As efficiency increases reliability also increases in producing the results and comparing all the classifiers.

2.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by the software engineers as the starting point for the system design. In hardware requirement we require all those components which will provide the platform for the development of project. The minimum hardware required for the development of this project is as follows –

- LAPTOP
- HARD DISK : 500 GB - 1 TB
- RAM : 4GB
- PROCESSOR : Intel core i5

2.3 SOFTWARE REQUIREMENTS

The software requirements are the software specifications of the system. It should include both a definition and specification of requirements. It is a set of what the system should do rather than how it should do it. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s progress throughout the development activity.

●	OPERATING SYSTEM	:	WINDOWS 7 AND ABOVE
●	TECHNOLOGY STACK	:	PYTHON 3.6
●	TOOLS	:	JUPYTER NOTEBOOK 6.0.3, ANACONDA NAVIGATOR 4.10.1

2.4 EXISTING SYSTEM

In the Existing System the classification algorithms of machine learning with the best accuracy are needed to assist the medical field in classifying individuals with pancreatic cancer. In this research, classification algorithms of Decision Tree and Logistic Regression were used. Furthermore, these two methods were compared to discover which has the best performance based on accuracy. The results showed that the Decision Tree and Logistic Regression yielded 50% and 72.68% respectively as their highest accuracy. Therefore, the Logistic Regression is a better method based on accuracy for classifying pancreatic cancer.

2.5 PROPOSED SYSTEM

The proposed system analyzes the accuracy of prediction of PC using machine learning techniques: SVM, RF, ETC and DT. These classifiers come under the category of supervised learning in machine learning. The classifier or the algorithm will be trained with the dataset that has the features and labels regarding PC, hence it becomes a trained model to predict the label. The trained model will be tested with new data or with random features from dataset. The performance of SVM, ETC, DT and RF classifiers are compared to find out which classifier have better accuracy among them. Also predicts the outcome that is whether the chosen person has the disease or not.

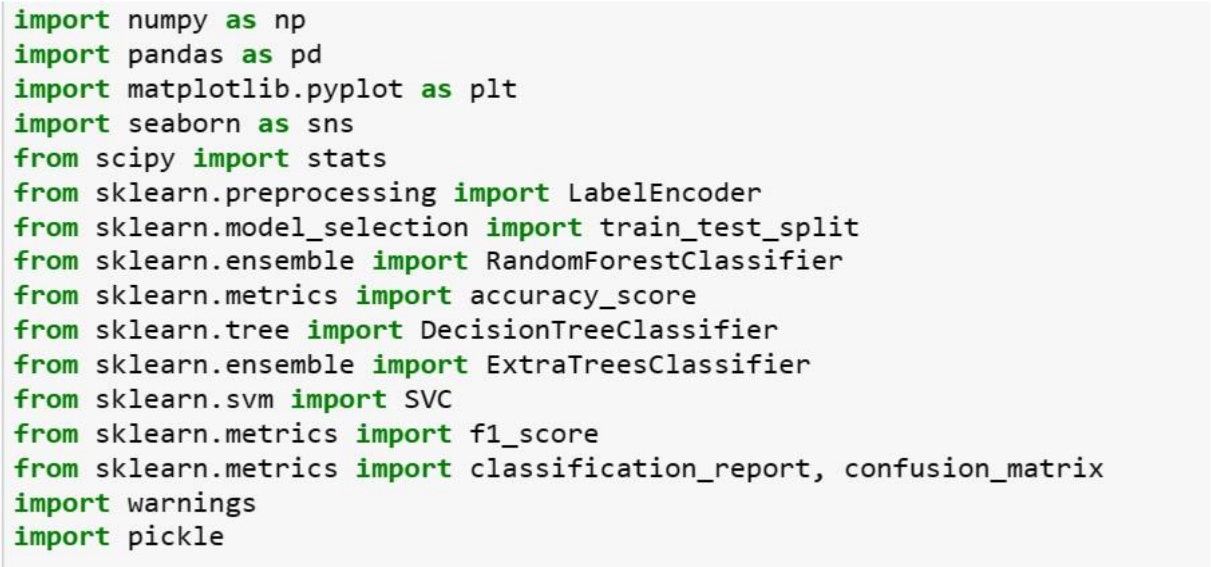
2.6 MODULES

In this project, there are seven modules –

1. Importing the libraries
2. Read the DataSet
3. Data processing
4. Analyzing Data and Visualization
5. Feature Selection
6. Model Training and Testing
7. Converting to .pkl file
8. Application Development
9. Predicting Output

Importing the Libraries:

Import the necessary libraries as shown in the image.

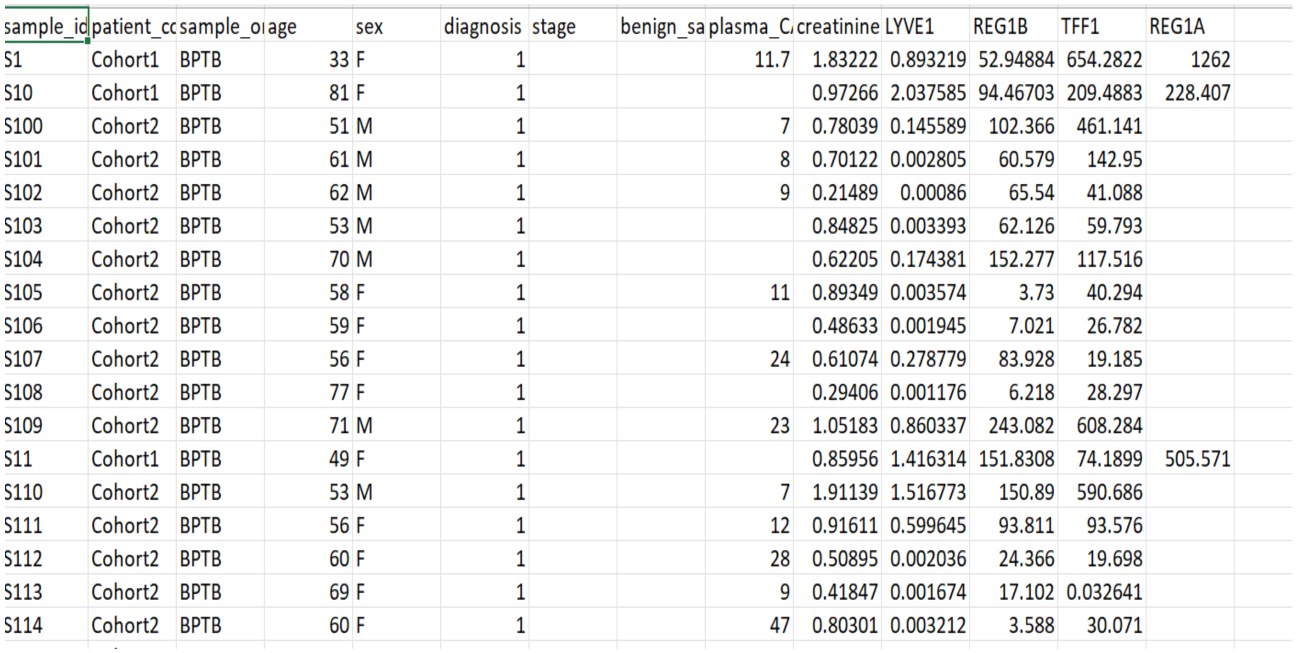


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.svm import SVC
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix
import warnings
import pickle
```

Fig : 2.1 Importing the Libraries

Read the DataSet:

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas. In pandas we have a function called read_excel() to read the dataset. As a parameter we have to give the directory of excel file..



sample_id	patient_cc	sample_o	age	sex	diagnosis	stage	benign_sa	plasma_C	creatinine	LYVE1	REG1B	TFF1	REG1A
S1	Cohort1	BPTB	33	F	1			11.7	1.83222	0.893219	52.94884	654.2822	1262
S10	Cohort1	BPTB	81	F	1				0.97266	2.037585	94.46703	209.4883	228.407
S100	Cohort2	BPTB	51	M	1			7	0.78039	0.145589	102.366	461.141	
S101	Cohort2	BPTB	61	M	1			8	0.70122	0.002805	60.579	142.95	
S102	Cohort2	BPTB	62	M	1			9	0.21489	0.00086	65.54	41.088	
S103	Cohort2	BPTB	53	M	1				0.84825	0.003393	62.126	59.793	
S104	Cohort2	BPTB	70	M	1				0.62205	0.174381	152.277	117.516	
S105	Cohort2	BPTB	58	F	1			11	0.89349	0.003574	3.73	40.294	
S106	Cohort2	BPTB	59	F	1				0.48633	0.001945	7.021	26.782	
S107	Cohort2	BPTB	56	F	1			24	0.61074	0.278779	83.928	19.185	
S108	Cohort2	BPTB	77	F	1				0.29406	0.001176	6.218	28.297	
S109	Cohort2	BPTB	71	M	1			23	1.05183	0.860337	243.082	608.284	
S11	Cohort1	BPTB	49	F	1				0.85956	1.416314	151.8308	74.1899	505.571
S110	Cohort2	BPTB	53	M	1			7	1.91139	1.516773	150.89	590.686	
S111	Cohort2	BPTB	56	F	1			12	0.91611	0.599645	93.811	93.576	
S112	Cohort2	BPTB	60	F	1			28	0.50895	0.002036	24.366	19.698	
S113	Cohort2	BPTB	69	F	1			9	0.41847	0.001674	17.102	0.032641	
S114	Cohort2	BPTB	60	F	1			47	0.80301	0.003212	3.588	30.071	

Fig 2.2 Read the DataSet

Data processing:

- For checking the null values, df.isnull() function is used. To sum those null values, we use. sum () function to it. we found that there are two null values present in our dataset. So, first we are exploringthe data.
- We have 1 missing value in Route column, and 1 missing value in Total stops column. We will meaningfully replace the missing values going further.
- We now start exploring the columns available in our dataset. The first thing we do is to create a list of categorical columns, and check the unique values present in these columns
- After dropping some columns, here we can see the meaningful columns to predict the flight price without the NAN values.
- Now we need to resolve the journey month and date, time of departure and time of arrival and handlethe categorical data.

Analyzing Data and Visualization:

Data analysis and visualization are iterative processes in machine learning, frequently requiringyou to return and improve your knowledge of the data as you unearth new insights. Effective visualization and analysis aid in improved model selection, feature engineering, and the dissemination of findings to non-technical stakeholders. To ensure openness and confidence in the model's predictions,it is a crucial phase in the machine learning workflow

Feature selection:

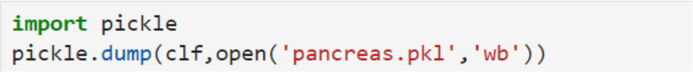
- Here, I have found various metrics with the help of various machine learning algorithms likeRandom Forest, SVM, Decision tree etc.
- We are getting the highest accuracy for random forest for testing dataset.

Model Training and Testing:

A subset of the dataset called training data is used to train the machine learning model toidentify patterns and make predictions. It is made up of input characteristics and the labels or goal values that correspond to the output features. Test data is a different subset of the dataset that is used to gauge the model's effectiveness and determine its capacity to make predictions onbrand-new, unforeseen data.

Converting to .pkl file:

Now we need to convert the file to pickle file and save the model as shown below.



```
import pickle
pickle.dump(clf,open('pancreas.pkl','wb'))
```

Application Development:

The process of creating software systems or applications that use machine learning models to address particular problems in the real world is known in the field of machine learning as "application building." These apps use the predictive power of machine learning models to make decisions, recommend actions, or automate operations.

1. Building HTML and CSS pages

2. Build python code

Predicting Output:

Predicting output is the process of utilizing a trained model to create predictions or forecasts based on input data. These predictions, which are basically the expected outcomes or answers produced by the model, might take a variety of shapes depending on the type of problemyou're attempting to address.

CHAPTER III

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

A System Architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A System Architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

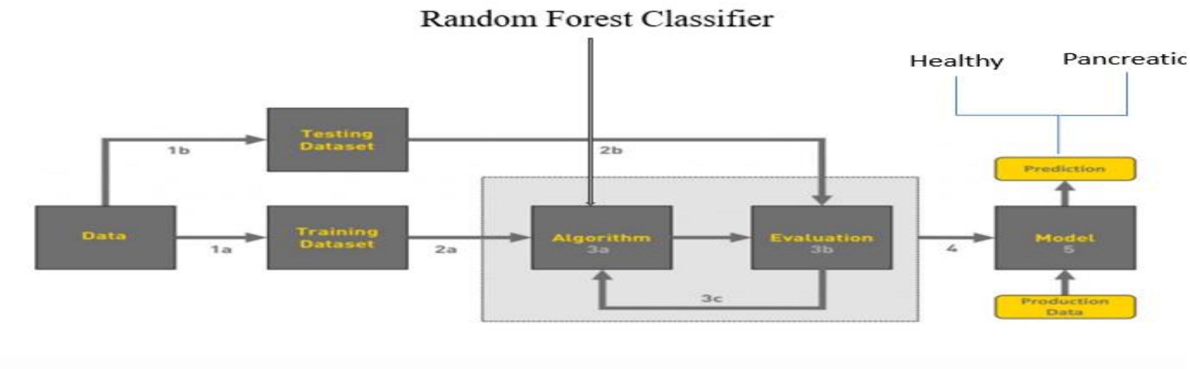


Fig 3.1 System Architecture

3.2 UML DIAGRAMS

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. There are two broad categories of diagrams and they are again divided into subcategories −

- Structural Diagrams
- Behavioral Diagrams

Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are −

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

Behavioral Diagrams

Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioral diagrams basically capture the dynamic aspect of a system. UML has the following five types of behavioral diagrams

- Use case diagram
- Sequence diagram
- Collaboration diagram
- Statechart diagram
- Activity diagram

3.2.1 USE CASE DIAGRAM

A use case diagram at the simplest is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a high-level view of the system. It has been said before that “Use case diagrams are the blueprints for your system”. They provide a simplified and graphical representation of what the system must actually do.

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed.

The purpose of the use case diagram is simply to provide the high-level view of the system and convey the requirements in laypeople’s terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view. In brief, the purpose of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.

- Identify the external and internal factors influencing the system.
- Shows the interactions among the requirements are actors.

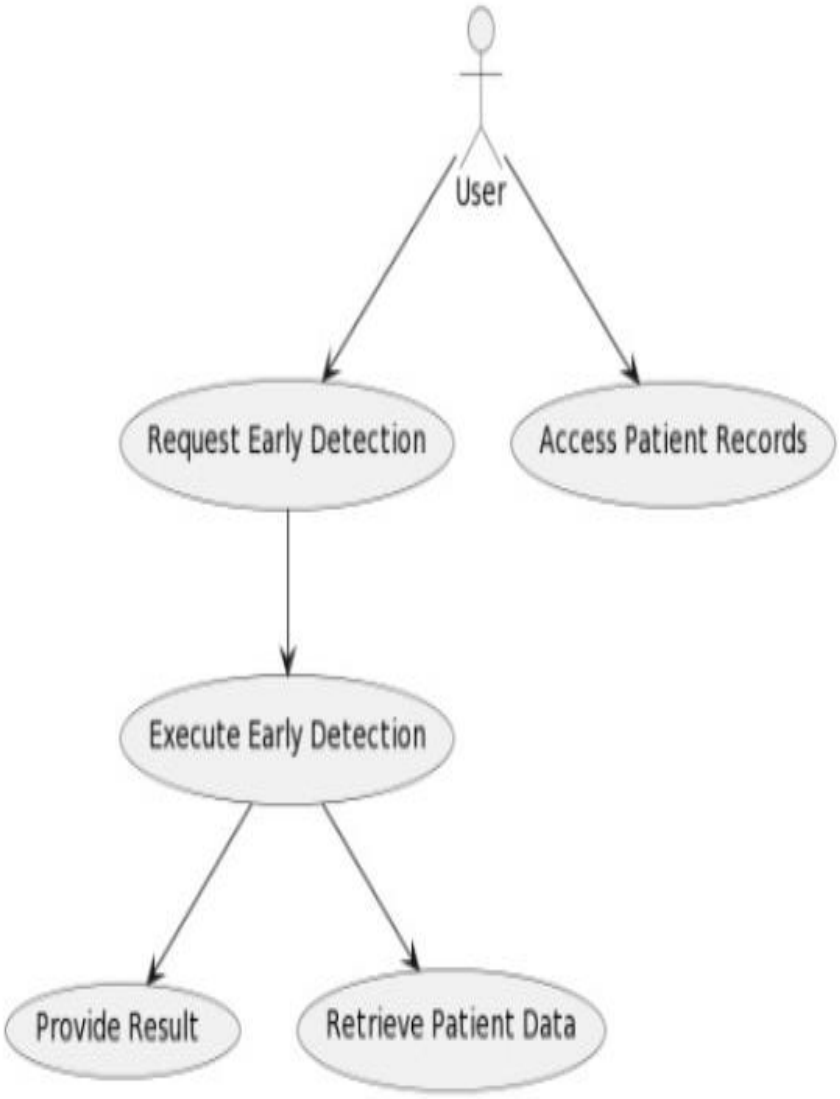


Fig 3.2 Use-case diagram

3.2.2 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios ina graphical manner.

If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances.

Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrowheads represent synchronous calls, open arrowheads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn’t have to wait for a response.

Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (Execution Specifications in UML).

Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. If an object is destroyed (removed from memory), an X is drawn on the bottom of the lifeline, and the dashed line ceases to be drawn below it. It should be the result of a message, either from the object itself, or another.

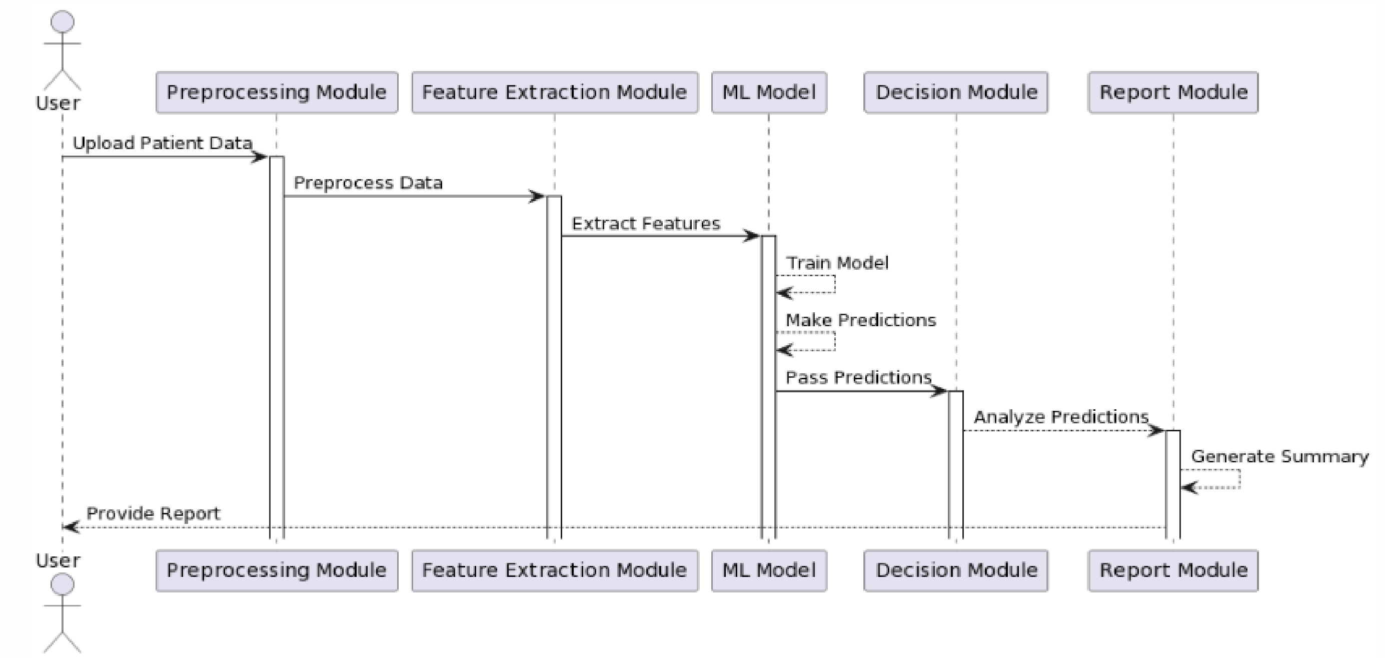


Fig 3.3 Sequence Diagram

3.2.3 CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The Class Diagram is the main building block of object- oriented modeling. It is usedfor general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

- The bottom compartment contains the operations the class can execute. They are also left- aligned and the first letter is lowercase.

In this project, we are considering the user and system as classes. Each of them has their own set of attributes and relevant operations as shown.

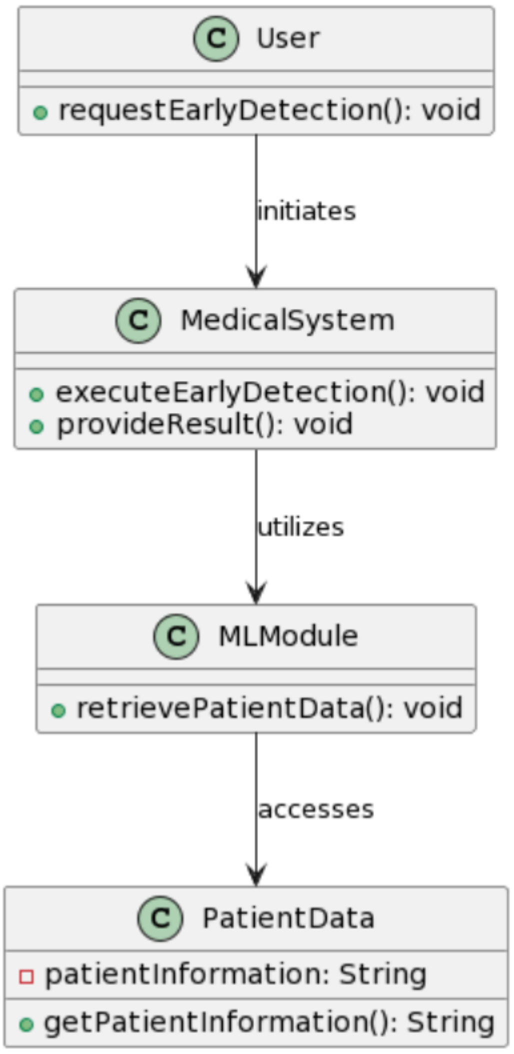

C User
requestEarlyDetection(): void
initiates
C MedicalSystem
executeEarlyDetection(): void
provideResult(): void
utilizes
C MLModule
retrievePatientData(): void
accesses
C PatientData
patientInformation: String
getPatientInformation(): String

Fig 3.4 Class Diagram

CHAPTER IV
SYSTEM IMPLEMENTATION

4.1 TECHNOLOGIES USED:

4.1.1 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non- profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)) Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle- detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the **off-side rule**, which some other languages share, but in most languages, indentation doesn't have any semantic meaning.

4.1.2 ANACONDA NAVIGATOR

AnacondaNavigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

Anacondais a data science platform built around the programming language Python. This open-source development tool works as an all-in-one data management tool, creating an environment that facilitates access to heavy amount of data. If you and your team need to **secure, interpret, scale, and store** critical data, this app can help.

Users must note, however, that Anaconda is targeted at a very niche audience. It focuses on large amounts of data, making it not suitable for small projects. Competitions, like Dev c++, are a better choice when you are working on producing a smaller amount of data.

Anaconda is primarily developed to support data science and machine learning tasks. It focuses on the distribution of R and Python programming languages and aimsat simplifying the data management and deployment of the mentioned languages. It offers allthe required packaged involved in data science at once. However, despite that, programmers still choose this program for fast installation and ease of use.

Installing the app is simple as it only requires you to follow the instructions from the wizard setup. Upon completion, the app provides you with more than **1,500 packages** in its distribution. In it, you will find the **Anaconda Navigator**, which is the graphical alternative to the command-line interface. This makes it easy for users to launch applications and manage packages and environments without using the command-line commands.

Anaconda is an enterprise-level software bundle that provides a host of innovative options to the end-user. As mentioned, it is great for managing all kinds of information and provides users an environment that facilitates access to heavy amounts of data. It enables organizations to successfully secure, interpret, scale, and store data critical to their operation. Not only that, but it also works to simplifytheprocess of working together on large batches of information.

Anaconda is **modular in nature**. You can adjust it depending on your or your organization's needs. Not only that, but the app also provides **access to other coding languages** besides Python. Be warned, though. Some programming languages may cause a few issues due to the real-time compilation. However, most of the time, restarting the program fixes the problem.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz

- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

4.1.3 JUPYTER NOTEBOOK

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Pérez.

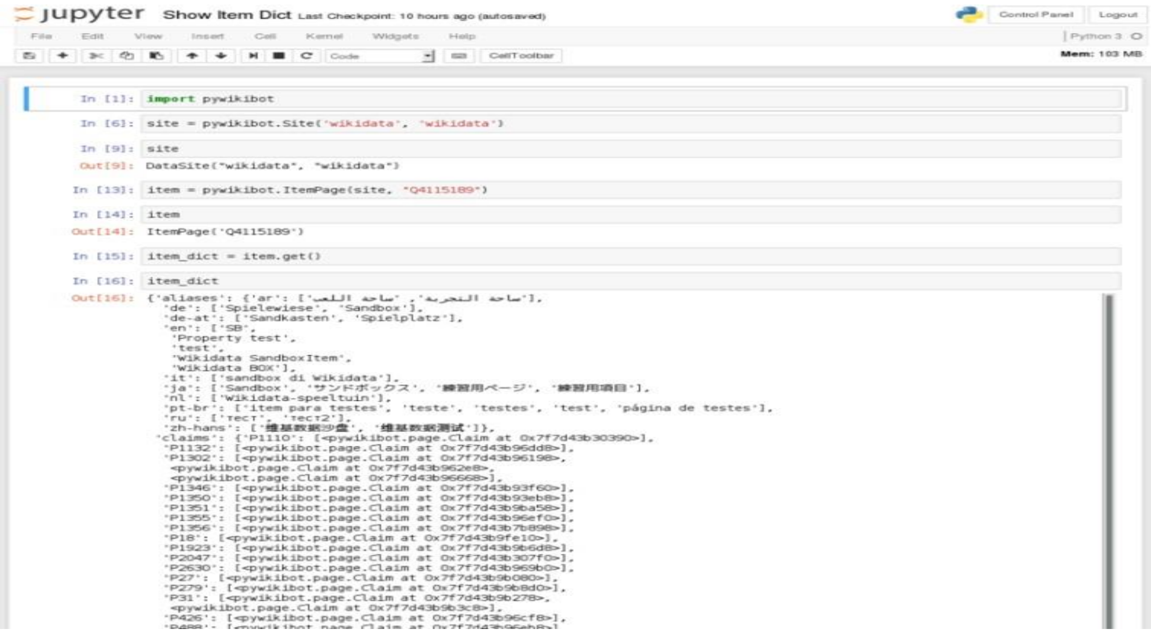
The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at project jupyter.

IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

Fig 4.1 Jupyter Notebook

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using python 3.



```
In [1]: import pywikibot

In [6]: site = pywikibot.Site('wikidata', 'wikidata')

In [9]: site
Out[9]: DataSite("wikidata", "wikidata")

In [13]: item = pywikibot.ItemPage(site, "Q4115189")

In [14]: item
Out[14]: ItemPage('Q4115189')

In [15]: item_dict = item.get()

In [16]: item_dict
Out[16]: {'aliases': {'ar': ['ساحة اللعب', 'ساحة التجربة'],
           'de': ['Spielewiese', 'Sandbox'],
           'de-at': ['Sandkasten', 'Spielplatz'],
           'en': ['SB',
            'Property test',
            'test',
            'Wikidata SandboxItem',
            'Wikidata BOX'],
           'it': ['sandbox di Wikidata'],
           'ja': ['Sandbox', 'サンドボックス', '練習用ページ', '練習用項目'],
           'nl': ['Wikidata-speeltuin'],
           'pt-br': ['item para testes', 'teste', 'testes', 'test', 'página de testes'],
           'ru': ['тест', 'тест2'],
           'zh-hans': ['维基数据沙盒', '维基数据测试']},
          'claims': {'P1110': [<pywikibot.page.Claim at 0x7f7d43b30390>],
           'P1132': [<pywikibot.page.Claim at 0x7f7d43b96dd8>],
           'P1302': [<pywikibot.page.Claim at 0x7f7d43b96198>,
            <pywikibot.page.Claim at 0x7f7d43b962e8>,
            <pywikibot.page.Claim at 0x7f7d43b96668>],
           'P1346': [<pywikibot.page.Claim at 0x7f7d43b93f60>],
           'P1350': [<pywikibot.page.Claim at 0x7f7d43b93eb8>],
           'P1351': [<pywikibot.page.Claim at 0x7f7d43b9ba58>],
           'P1355': [<pywikibot.page.Claim at 0x7f7d43b96ef0>],
           'P1356': [<pywikibot.page.Claim at 0x7f7d43b7b898>],
           'P18': [<pywikibot.page.Claim at 0x7f7d43b9fe10>],
           'P1923': [<pywikibot.page.Claim at 0x7f7d43b9b6d8>],
           'P2047': [<pywikibot.page.Claim at 0x7f7d43b307f0>],
           'P2630': [<pywikibot.page.Claim at 0x7f7d43b969b0>],
           'P27': [<pywikibot.page.Claim at 0x7f7d43b9b080>],
           'P279': [<pywikibot.page.Claim at 0x7f7d43b9b8d0>],
           'P31': [<pywikibot.page.Claim at 0x7f7d43b9b278>,
            <pywikibot.page.Claim at 0x7f7d43b9b3c8>],
           'P426': [<pywikibot.page.Claim at 0x7f7d43b96cf8>],
           'P488': [<pywikibot.page.Claim at 0x7f7d43b96eb8>]
```

Fig 4.2 Jupyter Notebook Interface

A Notebook’s cell defaults to using code whenever you first create one, and that cell uses the kernel that you chose when you started your Notebook.In this case, you started yours with Python 3 as your kernel, so that means you can write Python code in your code cells. Since your initial Notebook has only one empty cell in it, the Notebook can’t really do anything. Thus, to verify that everything is working as it should, you can add some Python code to the cell and try running its contents.Jupyter Notebook supports adding rich content to its cells. In this section, you will get an overview of just some of the things you can do with your cells using Markup and Code.

Jupyter Notebook provides a browser-based REPL built upon a number of popular open-source libraries:

- IPython
- ØMQ (ZeroMQ)
- Tornado (web server)
- jQuery
- Bootstrap (front-end framework)

- MathJax

4.2 ALGORITHM:

4.2.1 Random Forest Classifier:

RF is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. RF algorithm consists of many decision trees. The 'forest' generated by the RF algorithm is trained through bagging or bootstrap aggregating.

4.2.2 Decision Tree Classifier:

A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds 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.

4.2.2 Extra Trees Classifier:

A function named Extra Tree is created and train and test data are passed as the parameters. Inside the function, Extra Tree Classifier algorithm is initialised and training data is passed to the model with the. fit () function. Test data is predicted with. predict () function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

4.2.3 Support Vector Machine Classifier:

A function named Support Vector is created and train and test data are passed as the parameters. Inside the function, the Support Vector Classifier algorithm is initialised and training data is passed to the model with the. fit () function. Test data is predicted with. predict () function and saved in a new variable. For evaluating the model, confusion matrix and classification report is done

4.3 SAMPLE CODE

App.py

```
from flask import Flask, render_template, request
import numpy as np
import pickle
```

```
model = pickle.load(open(r"pancreas.pkl", 'rb'))
app = Flask(__name__)
@app.route("/")
def about():
    return render_template('home.html')
@app.route("/home")
def about1():
    return render_template('home.html')
@app.route("/predict")
def home1():
    return render_template('predict.html')

@app.route("/performance_analysis")
def performance_analysis():
    return render_template('Performance_analysis.html')
@app.route("/Abouting")
def aboutuss():
    return render_template('aboutuss.html')
@app.route("/pred", methods=['POST', 'GET'])
def predict():
    x = [[x for x in request.form.values()]]
    print(x)
    x = np.array(x)
    print(x.shape)
    print(x)
    pred = model.predict(x)
    if pred == 0:
        pred = 'Healthy'
    elif pred == 1:
        pred = 'Benign hepatobiliary Disease'
    else:
        pred = 'Pancreas Cancer Detected'
    return render_template('submit.html', prediction_text=str(pred))
```

```
if __name___== "_main_":
 app.run(debug=False)
```

Pancreas.ipynb

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.svm import SVC
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix
import warnings
import pickle df=pd.read_csv("pradataset.csv")
df.isnull().sum()
df=df.drop("sample_id",axis=1)
df["patient_cohort"].unique()
df["diagnosis"].unique()
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
df['patient_cohort']= label_encoder.fit_transform(df['patient_cohort'])
df['patient_cohort']=df['patient_cohort'].astype(int)
df['patient_cohort']=df['patient_cohort'].astype(int)
df['sample_origin']= label_encoder.fit_transform(df['sample_origin'])
df['sample_origin']=df['sample_origin'].astype(int)
df['sex']= label_encoder.fit_transform(df['sex'])
```

```
df['sex']=df['sex'].astype(int)
df['diagnosis'].unique()
df['stage']=df['stage'].fillna('0')
df['stage'].unique()
df['stage']= label_encoder.fit_transform(df['stage’])
df['stage']=df['stage'].astype(int)
df['benign_sample_diagnosis'].unique()
df['benign_sample_diagnosis']=df['benign_sample_diagnosis'].fillna("null")
df['plasma_CA19_9']=df['plasma_CA19_9'].fillna(df['plasma_CA19_9'].mean())
df['REG1A']=df['REG1A'].fillna(df['REG1A'].mean())
df=df.drop("benign_sample_diagnosis",axis=1)
X=df.drop("diagnosis",axis=1)
y=df['diagnosis']
# Build a Dataframe with Correlation between Features
corr_matrix = X.corr()
# Take absolute values of correlated coefficients
corr_matrix = corr_matrix.abs().unstack()
corr_matrix = corr_matrix.sort_values(ascending=False)
corr_matrix = corr_matrix[corr_matrix >= 0.8]
corr_matrix = corr_matrix[corr_matrix < 1]
corr_matrix = pd.DataFrame(corr_matrix).reset_index()
corr_matrix.columns = ['feature1', 'feature2', 'Correlation']
corr_matrix.head()
# Import label encoder
from sklearn import preprocessing
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
y= label_encoder.fit_transform(y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=60)
from sklearn.ensemble import ExtraTreesClassifier
Etc=ExtraTreesClassifier(n_estimators=100,max_depth=6,min_samples_split=2,min_weight_fraction_leaf =0.0,n_jobs=-1)
```

```
etc.fit(X_train, y_train)
print(etc.score(X_test, y_test)*100)
y_pred9 = etc.predict(X_test)
from sklearn.metrics import precision_recall_fscore_support
precision_recall_fscore_support(y_test, y_pred9,
average='macro')
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators = 100)
clf.fit(X_train, y_train)
y_pred8 = clf.predict(X_test)
from sklearn import metrics
print()
# using metrics module for accuracy calculation
print( metrics.accuracy_score(y_test, y_pred8)*100)
from sklearn.metrics import precision_recall_fscore_support
precision_recall_fscore_support(y_test, y_pred8, average='macro')
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_test_predict2=dtc.predict(X_test)
test_accuracy=(metrics.accuracy_score(y_test,y_test_predict2)*100)
test_accuracy
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
svc= SVC()
svc.fit(X_train,y_train)
y_test_predict4=svc.predict(X_test)
test_accuracy=(accuracy_score(y_test,y_test_predict4)*100)
test_accuracy
```

home.html:

```
<!doctype html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Home</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
        body
        {
         background-repeat:no-repeat;
        background-attached:fixed;
        background-size:cover;
        background-image:url({{url_for('static',filename='pics/07.png')}})

        }
        h3.big
        {
        line-height: 1.8;
        }
                  .navbar{
    padding :8px;
}
.navdiv{
    display: flex; align-items: center; justify-content: space-between;
}
.logo header{
    font-size: 15px; font-weight: 600; color: black;
}
li{
    list-style: none; display: inline-block;
}
li a{
    color: black; font-size: 18px; font-weight: bold; margin-right: 25px;
}
        strong{
        font-size : 80px;
        }

    </style>
</head>
<body>

   <nav class="navbar">
    <div class="navdiv">
<div class="logo">
    <header>
        <h1>HOME</h1>

    </header>
```

```
</div>
       <ul>

             <li> <a href="/home" >Home</a></li>
             <li> <a href="/predict" >Predict</a></li>
              <li><a href="/performance_analysis" >Performance analysis</a></li>
              <li><a href="/Abouting" >About</a></li>

       </ul>
       </div>
   </nav>
  <br>
  <br>
       <center>
           <h1><strong>Detection of Pancreatic Cancer</strong></h1>


           <h2>“When cancer happens, you don’t put life on hold. You live now.”</h2>
       </center>


              </div>

              <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
              <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
          </body>
          </html>
```

predict.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Predict</title>
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
   <style>
       body
       {
        background-repeat:no-repeat;
       background-attached:fixed;
       background-size:cover;
       background-image:url({{url_for('static',filename='pics/05.jpeg')}})

       }
        .navbar{
   padding :8px;
}
.navdiv{
   display: flex; align-items: center; justify-content: space-between;
}
.logo header{
   font-size: 15px; font-weight: 600; color: black;
}
li{
   list-style: none; display: inline-block;
}
li a{
   color: white; font-size: 18px; font-weight: bold; margin-right: 25px;
}

   </style>
</head>
   <body>
  <nav class="navbar">
   <div class="navdiv">
<div class="logo">
   <header>
      <h1>Prediction</h1>
   </header>
</div>
      <ul>

           <li> <a href="/home" >Home</a></li>
           <li> <a href="/predict" >Predict</a></li>
            <li><a href="/performance_analysis" >Performance analysis</a></li>
            <li><a href="/Abouting" >About</a></li>

      </ul>
      </div>
   </nav>
       <div class="container">
       <h4>
          <form action="/pred", method="POST">

                <div class="form-group row">
                   <div class="col-md-3">
                      <label for="plasma_CA19_9">plasma_CA19_9</label>
                      <input type="number" class="form-control" name="plasma_CA19_9" min=0 max=100000
```

```
step=0.0000001 id="plasma_CA19_9 up to 100000"  placeholder="plasma_CA19_9" required="required"/>
                         </div>
                     </div>
                      <div class="form-group row">
                          <div class="col-md-3">
                              <label for="creatinine">creatinine</label>
                              <input type="number" class="form-control" name="creatinine" min=0 max=2
step=0.0000001 id="creatinine 0-2"  placeholder="creatinine 0-2" required="required"/>
                          </div>
                      </div>

                     <div class="form-group row">
                        <div class="col-md-3">
                            <label for="LYVE1">LYVE1</label>
                            <input type="number" class="form-control" name="LYVE1" min=0 max=10
step=0.0000001 id="LYVE1"  placeholder="LYVE1 0-10" required="required"/>
                        </div>
                    </div>

                <div class="form-group row">
                    <div class="col-md-3">
                        <label for="REG1B">REG1B</label>
                        <input type="number" class="form-control" name="REG1B" min=0 max=500 step=0.0000001
id="REG1B"  placeholder="REG1B 0-500" required="required"/>
                    </div>
                </div>

            <div class="form-group row">
                <div class="col-md-3">
                    <label for="TFF1">TFF1</label>
                    <input type="number" class="form-control" name="TFF1" min=0 max=3000 step=0.0000001
id="TFF1"  placeholder="TFF1 up to 3ooo" required="required"/>
                </div>

        </div>
        <div class="form-group row">
            <div class="col-md-3">
                <label for="REG1A">REG1A</label>
                <input type="number" class="form-control" name="REG1A" min=0 max=14000 step=0.0000001
id="REG1A"  placeholder="REG1A up to 14k" required="required"/>
            </div>
        </div>
```

```
          <button type="submit" class="btn btn-success btn-lg">Submit</button>
  </form>
       <br>
       </h4>
    </div>


    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  </body>
  </html>
```

Performance.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Performance Analysis</title>

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        body
        {
         background-repeat:no-repeat;
        background-attached:fixed;
        background-size:cover;
        background-image:url({{url_for('static',filename='pics/06.png')}})

        }

        h1 {
            margin: 0;
        }
        .container {
            width: 80%;
            margin: 20px auto;
        }
        .analysis {
            background-color: #f4f4f4;
            padding: 20px;
            border-radius: 5px;
            margin-bottom: 20px;
        }
        .analysis h2 {
            margin-top: 0;
        }
        .results {
            display: flex;
            justify-content: space-between;
            margin-top: 10px;
        }
        .results p {
            margin: 5px 0;
        }
          .navbar{
    padding :8px;
}
.navdiv{
    display: flex; align-items: center; justify-content: space-between;
}
.logo header{
    font-size: 15px; font-weight: 600; color: black;
}
li{
    list-style: none; display: inline-block;
}
li a{
    color: black; font-size: 18px; font-weight: bold; margin-right: 25px;
}

    </style>
</head>
<body>
 <nav class="navbar">
    <div class="navdiv">
<div class="logo">
    <header>
        <h1>Performace Analysis</h1>
    </header>
</div>
        <ul>

               <li> <a href="/home" class="btn btn-info btn-lg">Home</a></li>
               <li> <a href="/predict" class="btn btn-primary  btn-lg">Predict</a></li>
                <li><a href="/performance_analysis" class="btn btn-primary btn-lg">Performance
analysis</a></li>
```

```
            <li><a href="/Abouting" class="btn btn-primary  btn-lg">About</a></li>

        </ul>
        </div>
      </nav>
    <br>

      <div class="container">
        <div class="analysis">
          <h2>Random Forest Classifier</h2>
          <div class="results">
            <p>Accuracy: 86%</p>
            <p>Precision: 85%</p>
            <p>Recall: 85%</p>
            <p>F1 Score: 85%</p>
          </div>
        </div>
        <div class="analysis">
          <h2>Extra Trees Classifier</h2>
          <div class="results">
            <p>Accuracy: 82.16%</p>
            <p>Precision: 84%</p>
            <p>Recall: 84%</p>
            <p>F1 Score: 83%</p>
          </div>
        </div>
        <div class="analysis">
          <h2>Decision Tree Classifier</h2>
          <div class="results">
            <p>Accuracy: 81.03%</p>
            <p>Precision: 80%</p>
            <p>Recall: 80%</p>
            <p>F1 Score: 80%</p>
          </div>
        </div>
        <div class="analysis">
          <h2>Support Vector Classifier</h2>
          <div class="results">
            <p>Accuracy: 50%</p>
            <p>Precision: 53%</p>
            <p>Recall: 52%</p>
            <p>F1 Score: 49%</p>
          </div>
        </div>

        <div class="analysis">
          <h2>Comparsion Of Algorithms</h2>
          <div class="results">
            <img src="{{url_for('static',filename='pics/algorithms.png')}}" alt="unloaded" />
            </div>
            </div>
            <br>
            <br>
            <br>
            <h2>Data Set Analysis</h2>
            <div class="analysis">
                <h2>Sex Analysis</h2>
                <div class="results">
                    <img src="{{url_for('static',filename='pics/sex.png')}}" />
                </div>
                <div class="analysis">
                    <h2>Diagnosis Analysis</h2>
                    <div class="results">
                        <img src="{{url_for('static',filename='pics/diagnosis.png')}}" alt="unloaded" />
                    </div>



            </div>
            </div>
        </div>
</body>
</html>
```

Submit.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Output</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
         body
        {
         background-repeat:no-repeat;
        background-attached:fixed;
        background-size:cover;
        background-image:url({{url_for('static',filename='pics/010.jpeg')}})

        }

                  .navbar{
    padding :8px;
}
.navdiv{
    display: flex; align-items: center; justify-content: space-between;
}
.logo header{
    font-size: 15px; font-weight: 600; color: black;
}
li{
    list-style: none; display: inline-block;
}
li a{
    color: black; font-size: 18px; font-weight: bold; margin-right: 25px;
}

    </style>
</head>

     <body>
   <nav class="navbar">
    <div class="navdiv">
<div class="logo">
    <header>
        <h1>Result</h1>
    </header>
</div>
        <ul>

               <li> <a href="/home" >Home</a></li>
               <li> <a href="/predict" >Predict</a></li>
                <li><a href="/performance_analysis" >Performance analysis</a></li>
                <li><a href="/Abouting" >About</a></li>

        </ul>
        </div>
    </nav>
<h1><strong>Detection of Pancreatic Cancer</strong></h1><br>
<h3>
    The predicted cancer for the pancreatic Disease is <B>{{prediction_text}} </B>
</h3>
</div>
</body>
</html>
```

4.4 OUTPUT SCREENS

Model Comparison :

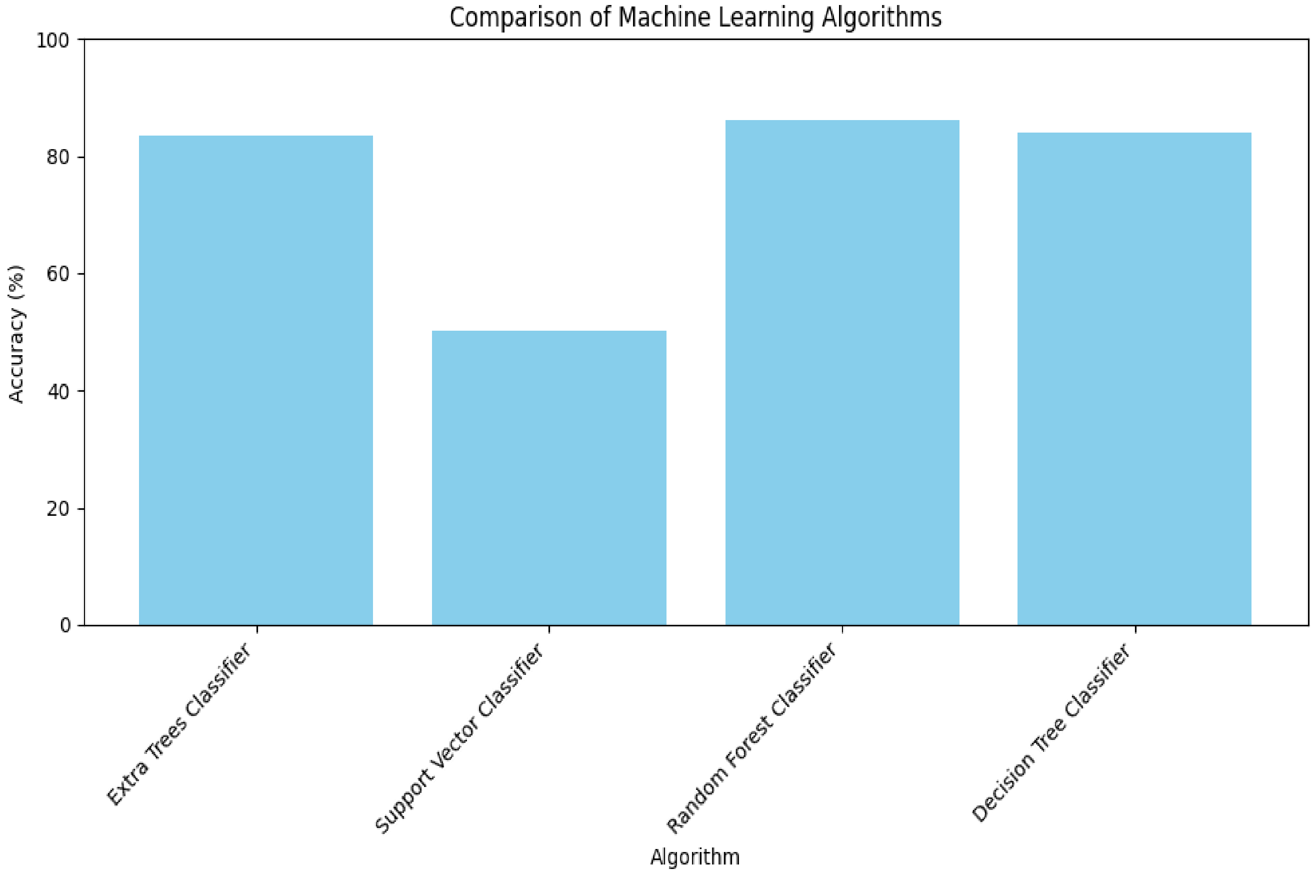
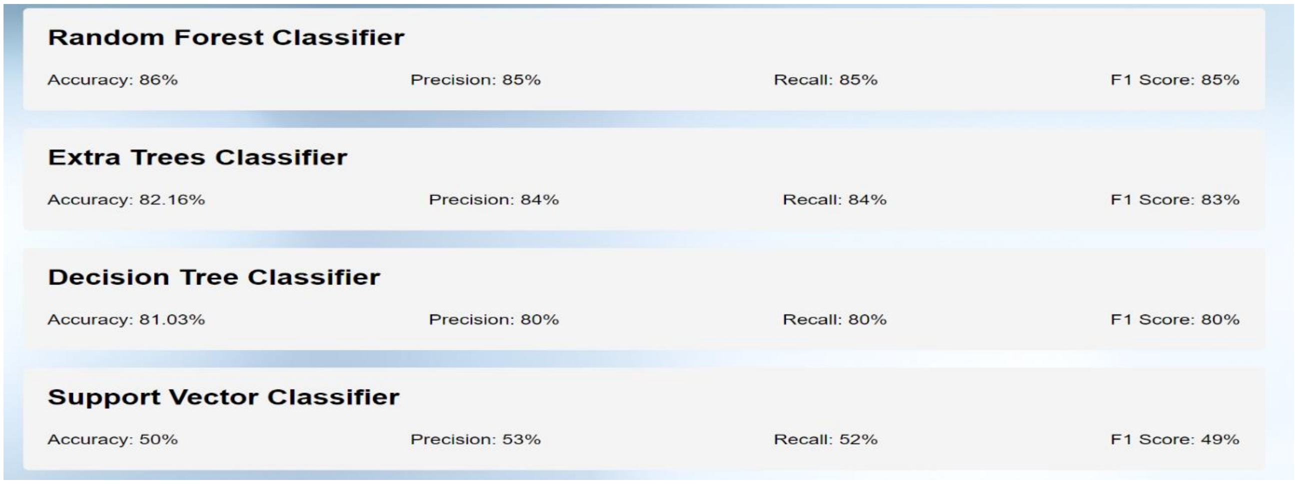


Fig 4.3 Model Comparsion

Performance Analysis:



Random Forest Classifier

Accuracy: 86% Precision: 85% Recall: 85% F1 Score: 85%

Extra Trees Classifier

Accuracy: 82.16% Precision: 84% Recall: 84% F1 Score: 83%

Decision Tree Classifier

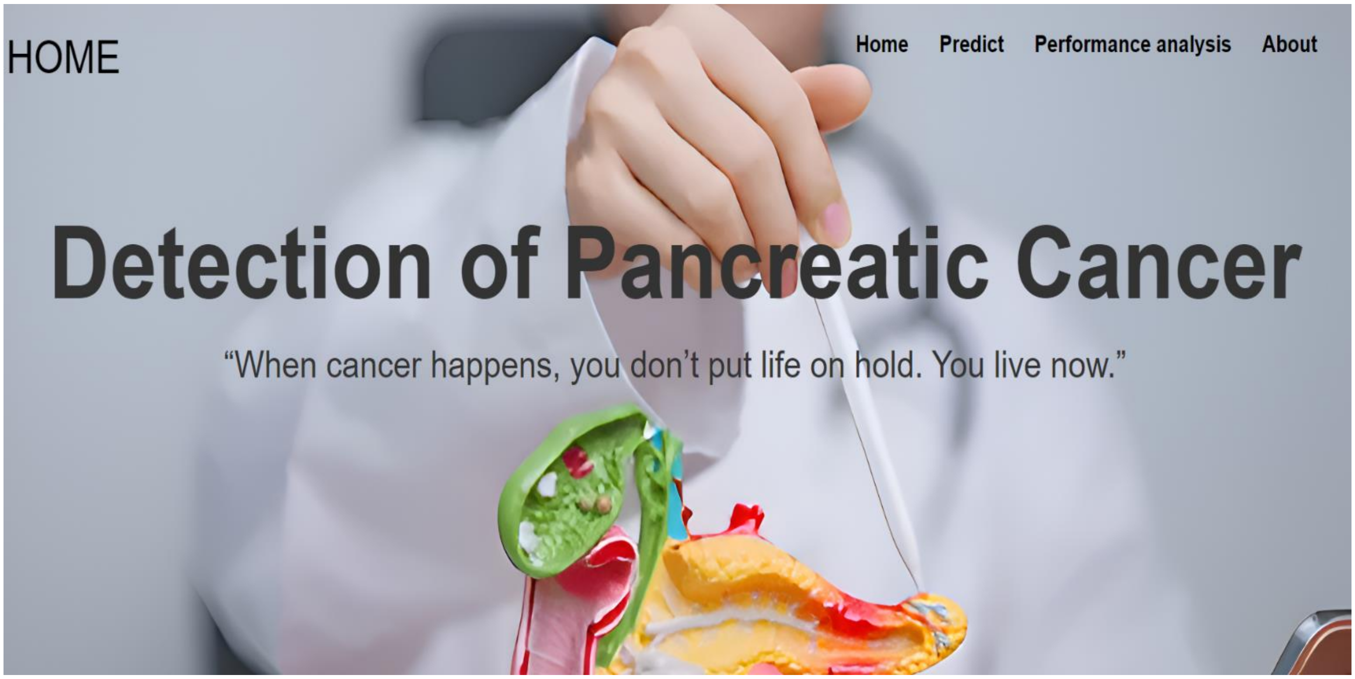
Accuracy: 81.03% Precision: 80% Recall: 80% F1 Score: 80%

Support Vector Classifier

Accuracy: 50% Precision: 53% Recall: 52% F1 Score: 49%

Fig 4.4 Performance analysis

Home Page :



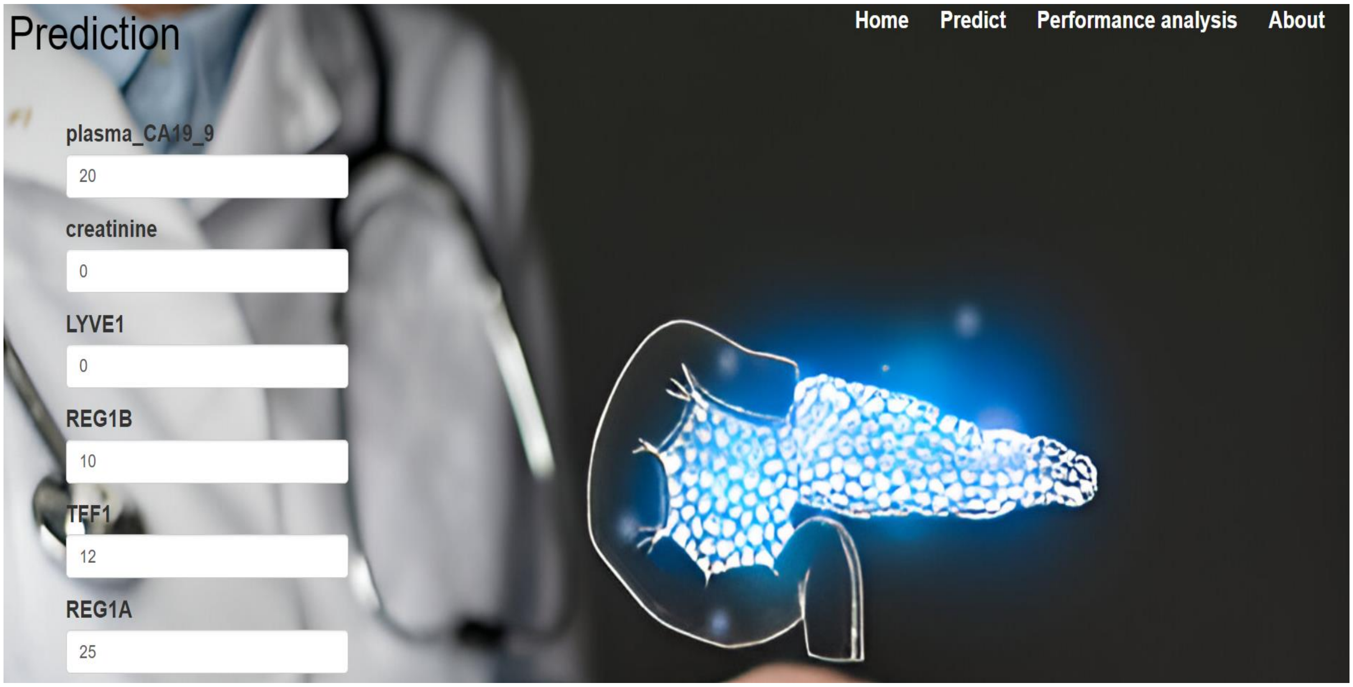
HOME

Detection of Pancreatic Cancer

“When cancer happens, you don’t put life on hold. You live now.”

Fig 4.5 Home Page

Predict Page :



Prediction

plasma_CA19_9

20

creatinine

0

LYVE1

0

REG1B

10

TFF1

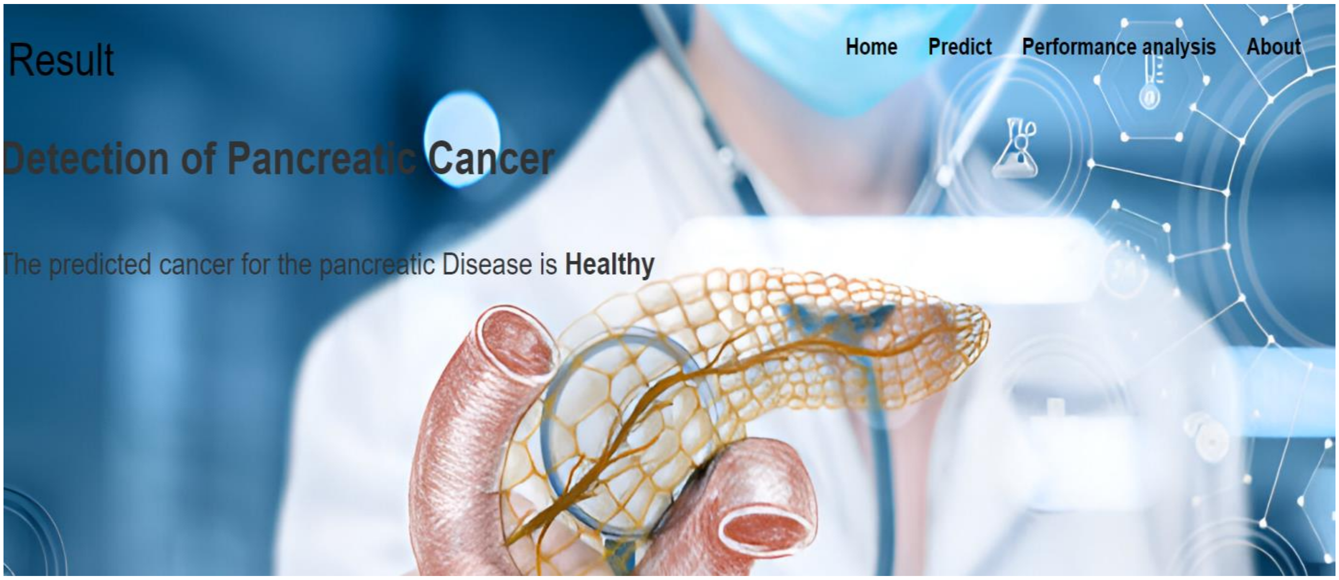
12

REG1A

25

Fig 4.6 Predict Page

Submit Page :



Result

Detection of Pancreatic Cancer

The predicted cancer for the pancreatic Disease is **Healthy**

Fig 4.7 Submit Page

Chapter-5

System Testing

5.1 PURPOSE OF TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests; each test type addresses a specific testing requirement. Testing and test design are parts of quality assurance that should also focus on bug prevention. A prevented bug is better than a detected and corrected bug. Testing consumes at least half of the time and work required to produce a functional program. History reveals that even well written programs still have 1-3 bugs per hundred statements. Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully, it will remove all the errors from the software.

5.2 TESTING STRATEGIES

In order to uncover the errors, present in different phases we have the concept of levels of testing. The Software testing has a prescribed order with the following list of software testing categories arranged in chronological order for our project testing and to generate test cases for output. These are the steps taken to fully test new software in preparation for marketing it:

Types of testing

- ***Unit testing*** performed on each module or block of code during development. Unit Testing is normally done by the programmer who writes the code.
- ***Integration testing*** done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.
- ***System testing*** done by a professional testing agent on the completed software product before it is introduced to the market.

- ***Acceptance testing*** - beta testing of the product done by the actual end users.

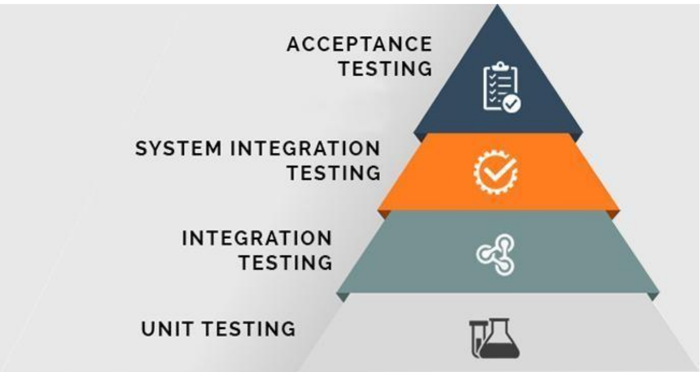


Fig 5.1 Types of Testing

5.2.1 UNIT TESTING

Unit testing is a crucial aspect of software development that aims to test the functionality of individual components or functions in isolation. In the case of online payment fraud detection, unit testing would involve testing the machine learning algorithms, data preprocessing, and feature engineering components separately. The primary goals of unit testing are to ensure that each unit performs as intended and to identify any potential issues at the lowest level of the system. This process is typically carried out by the development team using carefully designed test cases.

By conducting unit testing, developers can validate the performance of each component or function and ensure that they meet the expected requirements. This approach allows for the identification and resolution of any issues or bugs at an early stage, minimizing the impact on the overall system. In the context of online payment fraud detection, unit testing plays a crucial role in verifying the accuracy and effectiveness of the machine learning algorithms, data preprocessing techniques, and feature engineering methods. It provides developers with confidence in the reliability and functionality of these components before integrating them into the larger system.

5.2.2 INTEGRATION TESTING

Integration testing is a crucial phase in the software development process that aims to validate the proper functioning of different system components when they are combined. In the case of online payment fraud detection, this testing phase would involve examining the interactions between various machine learning models, data sources, and processing modules. The primary objective of integration testing is to identify and address any potential issues related to component interfaces, data flow, and the overall behavior of the system. By conducting integration testing, developers can ensure that the integrated system performs as intended and can effectively handle data flow and interactions.

Integration testing plays a vital role in ensuring the seamless integration of different components within a system. In the context of online payment fraud detection, this testing phase focuses on evaluating how machine learning models, data sources, and processing modules interact with each other. By scrutinizing the interfaces between these components, as well as the flow of data and the overall behavior of the system, integration testing helps identify and resolve any issues that may arise. This type of testing is essential to ensure that the integrated system functions as expected and can effectively handle data flow and interactions, ultimately enhancing the overall performance and reliability of the system.

5.2.3 SYSTEM TESTING

System testing, a comprehensive testing method, focuses on evaluating the entire system as a cohesive unit. In the case of online payment fraud detection, this testing approach entails assessing the complete end-to-end process, encompassing data collection, model deployment, and alert generation. Its primary objective is to gauge the system's effectiveness in fulfilling its intended functions and ensuring compliance with the specified requirements. Additionally, system testing may involve subjecting the system to diverse real-world scenarios, including testing with historical data and simulating various types of fraud attempts.

During system testing, the entire system is examined holistically to ensure its seamless integration and functionality. In the context of online payment fraud detection, this testing phase encompasses a thorough evaluation of the entire end-to-end process, starting from data collection and extending to model deployment and alert generation. The primary aim of system testing is to determine how well the system performs its intended functions and whether it meets the specified requirements. Furthermore, system testing may involve conducting tests under different real-world scenarios, such as using historical data and simulating various types of fraud attempts, to assess the system's performance and robustness.

CHAPTER VI

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Early detection of Pancreatic Cancer is very important so that the handling of Pancreatic Cancer does not occur too late, before the cancer spreads to other organs in the body. However, early detection of PC is difficult because this cancer has non-specific symptoms.

After classifying Pancreatic Cancer with SVM, Extra Tress, Decision Tree and Random Forest methods, it gets several results of accuracy. By comparing the values that are given from those methods , it is possible to conclude that Random Forest generates a better result than SVM, Extra Tress and Decision Tree. Because of the good results, Random Forest is suggested to help the medical staff to predict or classify a disease rather than SVM, Extra Tress and Decision Tree, especially for a dataset that is similar to this research.

The average rate of accuracy of Extra Trees Classifier is 82.1%, SVM is 50%, Decision Tree Classifier is 81.3% and for Random Forest Classifier is 86.34%. From this, it is clear that Random Forest gives an accurate result than the other three classifier algorithm. So, it can be concluded that Random Forest Classifier performs better than the other three classification algorithms.

6.2 FUTURE SCOPE

The future scope of the project could involve the following advancements:

- Keep making our computer tool smarter by trying out even better tricks and techniques beyond what we're using now.
- Imagine connecting our tool with cool gadgets people wear, like smartwatches or health monitors. This way, it can gather more information and become even better at spotting signs of pancreatic cancer.
- Think about how our tool could keep an eye on people's health all the time, not just once in a while. This might help predict problems early, like a superhero warning us about potential health issues.

APPENDIX

REFERENCES

Papers and articles

[1] A. Bosch, A. Zisserman and X. Munoz 2007 "Image Classification using Random Forests and Ferns,"IEEE 11th International Conference on Computer Vision, Rio de Janeiro, 10.1109/ICCV.2007.4409066

[2] Bhatt A, Dubey SK, Bhatt AK, Joshi M 2017, “Data Mining Approach to Predict and Analyze the Cardiovascular Disease”, Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications

[3] Bramhall, S.R., Neoptolemos, J.P., Stamp, G.W. and Lemoine, N.R 1998, Imbalance of expression of matrix metalloproteinases (MMPs) and tissue inhibitors of the matrix metalloproteinase (TIMPs) in human pancreatic carcinoma. J. Pathol., 182

[4] D. Arslan, M. E. Özdemir and M. T. Arslan2017, "Diagnosis of pancreatic cancer by pattern recognition methods using gene expression profiles", International Artificial Intelligence and Data Processing Symposium (IDAP), 10.1109/IDAP.2017.8090327.

[5] Daniele Ravi, Charence Wong 2017, “Deep Learning for HealthInformatics”, ieee journal of biomedical and health informatics

[6] D. Delen, G. Walker, and A. Kadam 2005, "Predicting breast cancer survivability: a comparison of three data mining methods," Artificial intelligence in medicine

[7] Dona Sara Jacob, RakhiViswan, V Manju, L PadmaSuresh, Shine Raj 2018, “A Survey on Breast Cancer Prediction Using Data Mining Techniques", IEEE Access

[8] Dr Prof. Neeraj, Sakshi Sharma, RenukaPurohit&Pramod Singh Rathore2017, “Prediction of Recurrence Cancer using J48Algorithm” Proceedings of the 2nd International Conference on Communication and Electronics Systems

[9] Dua D, Graff C 2019, “UCI machine learning repository”, School of Information and Computer Science, University of California, Irvine, CA

[10] Dwivedi AK, 2018 “Performance evaluation of different machine learning techniques for prediction of heart disease”, Neural Computer & Application 36

[11] Ellenrieder, V., Adler, G. and Gress, T.M 1999, Invasion and metastasis in pancreatic cancer. Ann. Oncol.10 (Suppl. 4)

[12] Escamilla AKG, El Hassani AH, Andres E 2019, “A Comparison of Machine Learning Techniques to Predict the Risk of Heart Failure”, Machine Learning Paradigms. Springer

[13] Eun Sun Lee, Jeong Min Lee 2014, “pancreatic cancer: A state-of - the-art review World”

[14] G.N. Satapathi, Dr.P.Srihari, Ch.ArunaJyothi, S. Lavanya 2013, “Prediction of cancer using DCP cells",

IEEE Access

[15] Ilias Tougui1,Abdelilah Jilbab1,Jamal El Mhamdi1 “Heart disease classification using data mining tools and machine learning techniques” . Health Technol, 2020

[16] Lola Rahib, Benjamin D Smith, Rhonda Aizenberg, Allison B Rosenzweig, Julie M Fleshman, and Lynn M Matrisian 2020, “Projecting cancer incidence and deaths to 2030: The unexpected burden of thyroid, liver, and pancreas cancers in the United States”

[17] Mohtadi K, Msaad R, Essadik R, Lebrazi H, Kettani A 2018, “Current risk factors of ischemic cardiovascular diseases estimated in a representative population of Casablanca”, EndocrinolMetabSyndr

[18] Ms. Rashmi G D, Mrs. A Lekha, Dr. NeelamBawane 2015, “Analysis of Efficiency of Classification and Prediction Algorithms (Naïve Bayes) for Breast Cancer Dataset", IEEE Access

[19] Sarfaraz Hussein, PujanKandel, Juan E. Corral CandiceW.Bolan, Michael B. Wallace and UlasBagci 2018, “Deep Multi-Modal Classification of Intraductal Papillary Mucinous Neoplasms (IPMN) with Canonical Correlation Analysis”, IEEE

[20] Shanjida Khan Maliha; Romana Rahman Ema; SimantaKumar Ghosh; Helal Ahmed; Md. RafsunJonyMollick; Tajul Islam 2019, “Cancer Disease Prediction Using Naive Bayes, Nearest Neighbor and J48 algorithm”