

# **PREDICTION OF CANCER USING MACHINE LEARNING**

## **MINOR PROJECT REPORT**

Submitted by

**PRAVEEN. A**

**17MSS039**

Under the Guidance of

**Mr. G. KARTHIKEYAN MCA., M.Phil**

Assistant Professor

Department of Software Systems

In partial fulfillment of the requirements for the award of the degree of

**MASTER OF SCIENCE IN SOFTWARE SYSTEMS**

(Five-years integrated course)

of Bharathiar University.



Since 1947



ISO 9001:2015

www.tuv.com  
ID 9105013918

**DEPARTMENT OF SOFTWARE SYSTEMS**

**PSG COLLEGE OF ARTS & SCIENCE**

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A' grade by NAAC (3<sup>rd</sup> Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore - 641 014

**October 2019**

**DEPARTMENT OF SOFTWARE SYSTEMS**

**PSG COLLEGE OF ARTS & SCIENCE**

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A' grade by NAAC (3<sup>rd</sup> Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2008 Certified Institution

Coimbatore - 641 014

**CERTIFICATE**

This is to certify that this minor project work entitled “**PREDICTION OF CANCER USING MACHINE LEARNING**” is a bonafide record of work done by **PRAVEEN. A (17MSS039)** in partial fulfillment of the requirements for the award of Degree of **Master of Science in Software Systems** (Five years Integrated Course) of Bharathiar University, Coimbatore – 641 046.

---

**Faculty Guide**

---

**Head of the Department**

Submitted for Viva-Voce Examination held on \_\_\_\_\_

---

**Internal Examiner**

---

**External Examiner**

## **DECLARATION**

I, **PRAVEEN. A (17MSS039)**, do hereby declare that this project minor work entitled **“PREDICTION OF CANCER USING MACHINE LEARNING”** is submitted to PSG College of Arts & Science, Coimbatore in partial fulfillment of the requirements for the award of the degree of Master of Science in Software Systems, is a record of original work done by me under the supervision and guidance of **Mr. G. KARTHIKEYAN MCA, M.Phil**, Assistant Professor, Department of Software Systems, PSG College of Arts & Science, Coimbatore - 14.

This minor Project Report has not yet been submitted by me for the award of any other Degree/ Diploma/ Associate ship/ Fellowship or any other similar degree to any other universities.

**Place:**

**PRAVEEN.A**

**Date:**

**(17MSS039)**

## ACKNOWLEDGEMENT

My venture stands imperfect without dedicating my gratitude to a few people who have contributed a lot towards the victorious completion for my project work.

I would like to thank **Thiru L. Gopalakrishnan, Managing Trustee, PSG & Sons Charities**, for providing me a prospect and surroundings that made the work possible.

I take this opportunity to express my deep sense of gratitude to **Dr. T. Kannaian, Secretary** of PSG College of Arts & Science, Coimbatore for permitting and doing the needful towards the successful completion of this project.

I express my deep sense of gratitude and sincere thanks to our Principal **Dr. D. Brindha M.Sc., M.Phil., Ph.D., MA (Yoga),** for her valuable advice and concern on students.

I am very thankful to **Dr. A Anguraj, M.Sc., M.Phil., Ph.D.,** Vice Principal(Academics), **Dr. Jayanthi M M.Com., MBA., M.Phil., Ph.D.,** Vice Principal (Student Affairs), **Prof. M Umarani, MBA, M.Phil., Faculty-In-Charge (Student Affairs),** for their support towards my project.

I sincerely thank **Dr. B.Murali., M.Sc., M.Phil., Ph.D.,** Head of the Department, Department of Software Systems for his whole hearted help to complete this project successfully by giving valuable suggestions.

I convey my heartiest and passionate sense of thankfulness to my project guide **G.Karthikeyan MCA.M.Phil,** Assistant Professor, Department of Software Systems, for her timely suggestion which had enable me in completing the project successfully.

This note of acknowledgement will be incomplete without paying my heartfelt devotion to my parents, my friends and other people, for their blessings, encouragement, financial support and the patience, without which it would have been impossible for me to complete the job.

## SYNOPSIS

This minor Project entitled “**PREDICTION OF CANCER USING MACHINE LEARNING**” is developed using **Python 3.7** as front end and **MS access 2010** as back end. This Project is used for interacting with the users and to store user inputs and generated outputs for future reference. Machine learning algorithm is used to predict the proper values from the user inputs.

In this Project , I have used University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg dataset to predict the values. This Dataset has details of the people living in America who has breast cancer problem. This dataset have the sample values of over 369 people.

Thus, this project is designed with a motive to predict chances of breast cancer for a individual with their biological factors like Clump Thickness , Marginal Adhesion and so on. The machine learning algorithm is trained with the dataset only.

## TABLE OF CONTENTS

| S.No      | CONTENTS                                   | PAGE NO   |
|-----------|--|-----------|
| <b>1.</b> | <b>INTRODUCTION</b>                        |           |
|           | 1.1 Project Overview                       | <b>1</b>  |
| <b>2</b>  | <b>SYSTEM ANALYSIS</b>                     |           |
|           | 2.1 Existing System                        | <b>2</b>  |
|           | 2.2 Proposed System                        |           |
| <b>3</b>  | <b>SYSTEM CONFIGURATION</b>                |           |
|           | 3.1 Hardware Specification                 | <b>3</b>  |
|           | 3.2 Software Specification                 |           |
|           | 3.3 Software Features                      | <b>4</b>  |
| <b>4</b>  | <b>SYSTEM DESIGN</b>                       |           |
|           | 4.1 System Flow Symbols                    | <b>12</b> |
|           | 4.2 Table Design                           | <b>13</b> |
|           | 4.3 Input Design                           | <b>14</b> |
|           | 4.4 Output Design                          | <b>15</b> |
|           | 4.5 Database Design                        |           |
| <b>5</b>  | <b>SYSTEM TESTING &amp; IMPLEMENTATION</b> |           |
|           | <b>5.1 System Testing</b>                  | <b>16</b> |
|           | <b>5.2 System Implementation</b>           | <b>19</b> |
| <b>6</b>  | <b>SCOPE FOR FUTURE ENHANCEMENT</b>        | <b>21</b> |
| <b>7</b>  | <b>CONCLUSION</b>                          | <b>22</b> |
| <b>8</b>  | <b>BIBLIOGRAPHY</b>                        | <b>23</b> |
| <b>9</b>  | <b>APPENDICES</b>                          |           |
|           | A. Screenshot                              | <b>24</b> |
|           | B. Sample Coding                           | <b>25</b> |

# 1. INTRODUCTION

## 1.1 OVERVIEW OF THE PROJECT

This minor Project entitled “**Prediction of cancer using machine learning**” is designed with a motive to predict the chances of breast cancer for an individual with their biological factors like Clump Thickness, Marginal Adhesion and so on. In this project I am trying to analyze the breast cancer dataset to extract some facts and also by using these facts we try to analyze the dataset. We will also use a data description to find the inequality among the data and also we will use the **k nearest neighbors(KNN)** to make a prediction model for predicting the breast cancer disease. We are using Wisconsin Breast Cancer Database - breast cancer dataset in order for analyzing the dataset. These are the table people living in phoenix Arizona in USA . This breast cancer dataset will be consisting of the sample taken from a survey among the women in the tribe. The predicted value’s accuracy varies with the values and the result will be displayed as positive or negative which is purely depends on the user defined input values. The sample of Wisconsin breast cancer dataset which is having 369 samples.

## **2. SYSTEM ANALYSIS**

### **2.1 EXISTING SYSTEM**

The existing system uses more hardware components for the detection of the detection of breast cancer from the person, which is complicated because the failure of any of the hardware component may affect the result. There is no system which is purely based on the user inputs.

#### **DRAWBACKS:**

- **It takes more time to predict.**
- **It is expensive.**
- **It has less accuracy**

### **2.2 PROPOSED SYSTEM**

The proposed system uses Machine Learning techniques to detect the breast cancer from the user inputs. The technique used here is Deep Learning where the model is trained to detect the breast cancer with a dataset of 369 samples. The trained model is then loaded and used to detect number breast cancer from the inputs. The advantage of using this system is that the model detects the breast cancer without any hardware components and then the result is stored in the database. The aim of this minor project is to develop an early stage and malignant stage breast cancer detection system which can automatically classify abnormalities in patients lab analyzed records. In this method, data pre-processing stage is to remove the noise . Data analyzation is used to predict the cancer is benign or malignant. We can insert new data of patient and get new analyzation data records. Apply training dataset to the KNN algorithms. Applying the KNN algorithms generate the machine learning model. Send test data to the model and first check the performance. Identify which algorithm is best for our system and then enter new test data to that model and predict the cancer stage.

#### **ADVANTAGES:**

- **It predicts fast.**
- **It is cheap of cost.**
- **It has high accuracy.**



## **3.SYSTEM CONFIGURATION**

### **3.1 HARDWARE SPECIFICATIONS**

- **PROCESSOR** : Intel i5 Processor 7<sup>th</sup> Gen
- **CPU CLOCK SPEED** : 1.6 GHz
- **RAM** : 4 GB or Higher
- **HDD** : 300 GB of available hard disk space
- **VIDEO CARD** : NVidia GeForce GTX 940MX or Higher
- **KEY BOARD** : Standard Windows Keyboard
- **MOUSE** : Two or Three Button Mouse
- **Monitor** : SVGA

### **3.2 SOFTWARE SPECIFICATIONS**

- **FRONT END** : HTML,CSS
- **BACK END** : Python 3.7 (scripting),Microsoft Access 2010
- **OPERATING SYSTEM** : Windows 10
- **IDLE** : Python IDLE

## 3.3 SOFTWARE FEATURES

### WINDOWS 10

**Windows 10** is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015.<sup>[13]</sup> Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

### GRAPHICAL USER INTERFACE

The **graphical user interface** is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

### FEATURES OF WINDOWS 10

- **Timeline:** A new feature to get a chronological view of the activities the user was previously doing and to switch back to those activities. Edge, File Explorer, Maps, and other built-in applications include support for Timeline. Any application written for Windows can interact with Timeline to give the system visibility into individual documents the user worked on inside that application.
- **Windows Hello:** Initial set up of Hello can be performed from the lock screen.
- **Bluetooth:** A new "Quick Pairing" capability that reduces the number of steps required to pair a nearby Bluetooth device. A toast notification is shown when a compatible device is near the computer and ready to be paired.

- **Task Bar:** The Acrylic visual style of the Fluent Design System is applied to the taskbar.
- **OneDrive status icons** are shown in File Explorer to show the synchronization status of files and folders.
- **Game Bar:** layout has changed to include a clock, and to provide new options to turn the microphone and camera on/off.
- **Nearby Sharing:** a new feature found in the Shared Experiences area of Settings that provides the ability to share files and web links to other Windows 10 machines over Wi-Fi or Bluetooth.
- **My People:** Acrylic visual style; three-contact limit is removed; new animations; drag-and-drop of contacts; app suggestions.
- **Text prediction:** A new optional feature to enable autocomplete and auto correction in any Windows application. The on-screen keyboard also has support for multilingual text prediction, which shows suggestions based on the top 3 installed Latin languages.

## Python (programming language)

---

**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 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released 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. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

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's design offers some support for functional programming in the Lisp tradition. It has `filter`, `map`, and `reduce` functions; list comprehensions, dictionaries, sets and generator expressions. The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from Haskell and Standard ML. The language's core philosophy is summarized in the document *The Zen of Python* (PEP 20), which includes aphorisms such as:

- **Beautiful is better than ugly.**
- **Explicit is better than implicit.**
- **Simple is better than complex.**
- **Complex is better than complicated.**
- **Readability counts.**

## Machine learning

---

**Machine learning (ML)** is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.<sup>[3][4]</sup> In its application across business problems, machine learning is also referred to as predictive analytics.

## **Python Language advantages and applications**

Python is a high level, interpreted and general purpose dynamic programming language that focuses on code readability. It has fewer steps when compared to Java and C. It was founded in 1991 by developer Guido Van Rossum. It is used in many organizations as it supports multiple programming paradigms. It also performs automatic memory management.

### **Advantages :**

- **Presence of third-party modules**
- **Extensive support libraries(NumPy for numerical calculations, Pandas for data analytics etc)**
- **Open source and community development**
- **Easy to learn**
- **User-friendly data structures**
- **High-level language**
- **Dynamically typed language(No need to mention data type based on value assigned, it takes data type)**
- **Object-oriented language**
- **Portable and Interactive**
- **Portable across Operating systems**

## **Applications :**

- **GUI based desktop applications(Games, Scientific Applications)**
- **Web frameworks and applications**
- **Enterprise and Business applications**
- **Operating Systems**
- **Language Development.**

## **INRODUCTION TO MICROSOFT ACCESS DATABASE 2016**

**Microsoft Access** is a database management system (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions or sold separately.

Microsoft Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases.

Software developers, data architects and power users can use Microsoft Access to develop application software. Like other Microsoft Office applications, Access is supported by Visual Basic for Applications (VBA), an object-based programming language that can reference a variety of objects including the legacy DAO (Data Access Objects), ActiveX Data Objects, and many other ActiveX components. Visual objects used in forms and reports expose their methods and properties in the VBA programming environment, and VBA code modules may declare and call Windows operating system operations.

## **NEW FEATURES IN MICROSOFT ACCESS**

Users can create tables, queries, forms and reports, and connect them together with macros. Advanced users can use VBA to write rich solutions with advanced data manipulation and user control. Access also has report creation features that can work with any data source that Access can access.

The original concept of Access was for end users to be able to access data from any source. Other features include: the import and export of data to many formats including Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server and Oracle. It also has the ability to link to data in its existing location and use it for viewing, querying, editing, and reporting. This allows the existing data to change while ensuring that Access uses the latest data. It can perform heterogeneous joins between data sets stored across different platforms. Access is often used by people downloading data from enterprise level databases for manipulation, analysis, and reporting locally.

There is also the Jet Database format (MDB or ACCDB in Access 2007) which can contain the application and data in one file. This makes it very convenient to distribute the entire application to another user, who can run it in disconnected environments.

One of the benefits of Access from a programmer's perspective is its relative compatibility with SQL (structured query language) — queries can be viewed graphically or edited as SQL statements, and SQL statements can be used directly in Macros and VBA Modules to manipulate Access tables. Users can mix and use both VBA and "Macros" for programming forms and logic and offers object-oriented possibilities. VBA can also be included in queries.

Microsoft Access offers parameterized queries. These queries and Access tables can be referenced from other programs like VB6 and .NET through DAO or ADO. From Microsoft Access, VBA can reference parameterized stored procedures via ADO.



Microsoft Access is a file server-based database. Unlike client–server relational database management systems (RDBMS), Microsoft Access does not implement database triggers, stored procedures, or transaction logging. Access 2010 includes table-level triggers and stored procedures built into the ACE data engine. Thus a Client-server database system is not a requirement for using stored procedures or table triggers with Access 2010. Tables, queries, forms, reports and macros can now be developed specifically for web based applications in Access 2010. Integration with Microsoft SharePoint 2010 is also highly improved.

The 2013 edition of Microsoft Access introduced a mostly flat design and the ability to install apps from the Office Store, but it did not introduce new features. The theme was partially updated again for 2016, but no dark theme was created for Access.

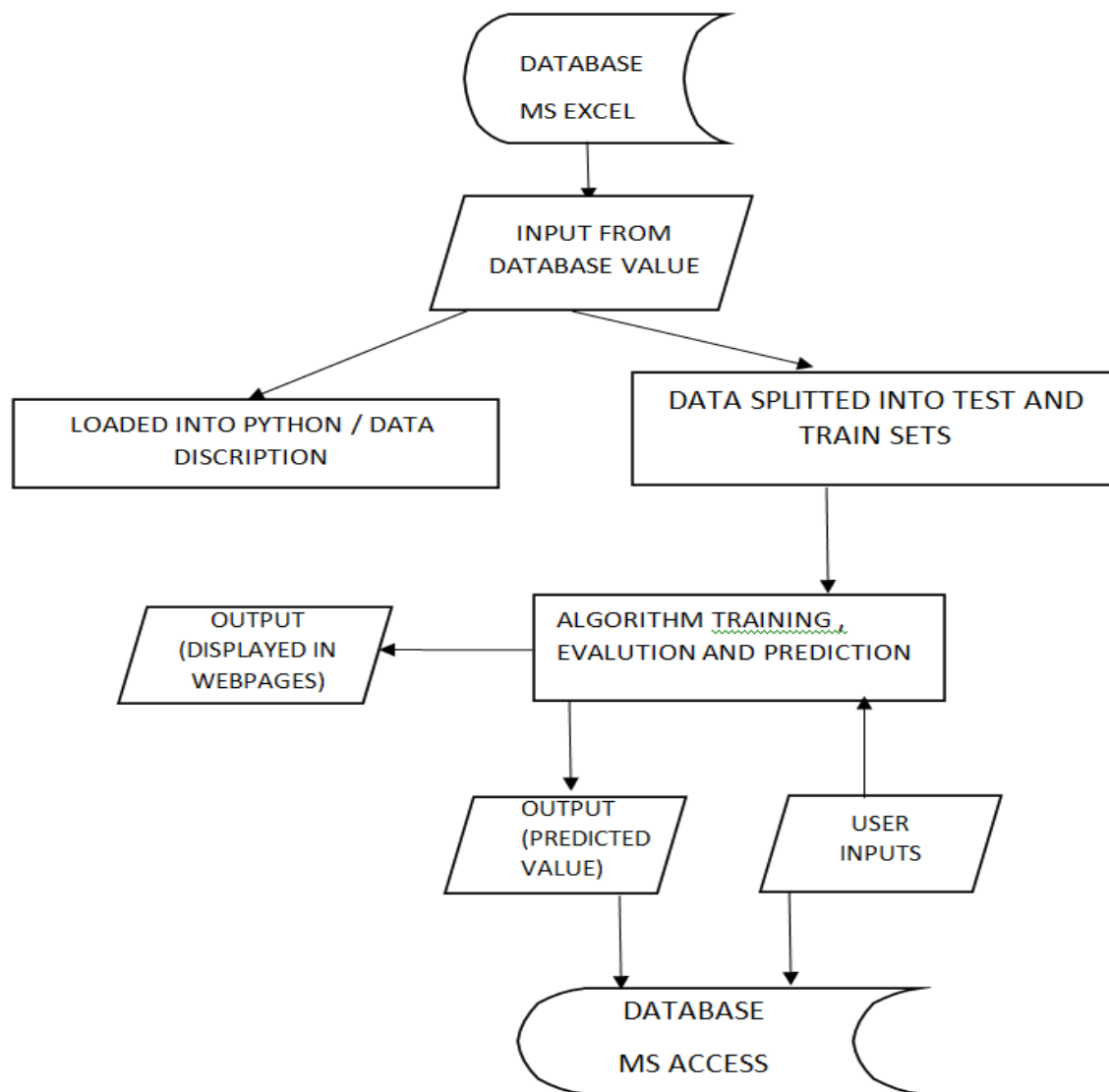
## **MAIN ELEMENTS OF ACCESS**

The Microsoft® Access Database is made up of 7 major components:

- **Tables;**
- **Relationships;**
- **Queries;**
- **Forms;**
- **Reports;**
- **Macros;**
- **Modules.**

## 4. SYSTEM DESIGN

### 4.1 SYSTEM FLOW DIAGRAM



## 4.2 TABLE DESIGN

Database Name : py\_db

Table Name : py\_db\_tb1

| Field Name                  | Data Type |
|-----------------------------|-----------|
| Sample code number          | Integer   |
| Clump Thickness             | Integer   |
| Uniformity of Cell Size     | Integer   |
| Uniformity of Cell Shape    | Integer   |
| Marginal Adhesion           | Integer   |
| Single Epithelial Cell Size | Integer   |
| Bare Nuclei                 | Float     |
| Bland Chromatin             | Float     |
| Normal Nucleoli             | Integer   |
| Mitoses                     | Integer   |
| Class                       | Integer   |

## 4.3 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer understandable format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. A large number of problems with a system can usually be tracked backs to fault input design and method. Every moment of input design should be analyzed and designed with utmost care.

The system takes input from the users, processes it and produces an output. Input design is link that ties the information system into the world of its users. The system should be user-friendly to gain appropriate information to the user. The decisions made during the input design are

- **To provide cost effective method of input.**
- **To achieve the highest possible level of accuracy.**
- **To ensure that the input is understand by the user.**

System analysis decide the following input design details like, what data to input, what medium to use, how the data should be arranged or coded, data items and transactions needing validations to detect errors and at last the dialogue to guide user in providing input.

Input data of a system may not be necessarily is raw data captured in the system from scratch. These can also be the output of another system or subsystem.

```
x1 :5
x2 :1
x3 :1
x4 :1
x5 :2
x6 :1
x7 :3
x8 :1
x9 :1
the data entered is : [5, 1, 1, 1, 2, 1, 3, 1, 1]
values predicted : 2
```

## 4.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with user and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information. All records from various Tables with same Database name are retrieved and consolidated, then displayed in a print Format.

| Id Number | Clump Thickr | Uniformity o | Uniformity o | Marginal Adf | Single Epithe | Bare Nuclei | Bland Chrom | Normal Nucl | Mitoses | Class |
|-----------|--------------|--------------|--------------|--------------|---------------|-------------|-------------|-------------|---------|-------|
| 10156     | 10           | 7            | 7            | 6            | 4             | 10          | 4           | 1           | 2       | 2     |
| 1486      | 7            | 4            | 6            | 4            | 6             | 1           | 4           | 3           | 1       | 2     |
| 158416    | 8            | 7            | 5            | 10           | 7             | 9           | 5           | 5           | 4       | 2     |
| 190       | 1            | 1            | 1            | 1            | 1             | 1           | 1           | 1           | 1       | 2     |
| 344       | 8            | 5            | 6            | 2            | 3             | 10          | 6           | 6           | 1       | 4     |
| 5481      | 10           | 5            | 5            | 3            | 6             | 7           | 7           | 10          | 1       | 2     |
| 79855     | 2            | 3            | 4            | 6            | 7             | 4           | 3           | 5           | 3       | 2     |
| 9         | 9            | 10           | 10           | 10           | 10            | 5           | 10          | 10          | 10      | 4     |
| 9464      | 8            | 10           | 10           | 8            | 7             | 10          | 9           | 7           | 1       | 4     |

## 4.5 DATABASE DESIGN

The database design involves creation of tables. Tables are represented in physical database as stored files. Database design has an own independent existence. A table consist of rows and columns. Each column corresponds to piece of information called field. A set field constitutes a record. Database design is required to manage the large amount of information. The management of data involves both the definition of structure of the storage of information and provision of mechanism for the manipulation of information.

The record contains all the information, specific to particular item. This is one, which transforms the information models created during analysis Numeric to data structure that will be required to implement the software.

## **5. SYSTEM TESTING & IMPLEMENTATION**

### **5.1 SYSTEM TESTING**

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs.

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information. Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing.

The test steps include

- **Unit testing**
- **Validation testing**
- **Acceptance Testing**
- **Black Box Testing**

## **Unit testing:**

Unit testing focuses verification efforts on the smallest of the software design of the module. This is also known as module testing. The modules of the “banking services” are tested separately. This test was carried out during programming stage.

## **Validation testing:**

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected and final series of software test begins. Validation testing can be defined in many ways, but a simple definition is the validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

After the validation test has been conducted, one of the three possible conditions exists.

- **The function of performance characteristics to specification and is accepted.**
- **A deviation from specification is uncovered and a deficiency list is created.**
- **Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.**

## **Acceptance testing:**

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

## **Black Box Testing**

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle. This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

Black Box Testing method is applicable to the following levels of software testing:

- **Integration Testing**
- **System Testing**
- **Acceptance Testing**

The higher the level, and hence the bigger and more complex the box, the more black-box testing method comes into use.

## **Techniques**

Following are some techniques that can be used for designing black box tests.

- ***Equivalence Partitioning:*** It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
- ***Boundary Value Analysis:*** It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.
- ***Cause-Effect Graphing:*** It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

## **Advantages**

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Tester need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
- Test cases can be designed as soon as the specifications are complete.



## 5.2 SYSTEM IMPLEMENTATION

The purpose of System Implementation can be summarized as follows:

It making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the performing organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the consumers on the use of the new system, placing the newly developed system into production ,confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system development to a system support and maintenance modes of operation, with ownership of the new system from the Project Team to the Performing Organization. System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- **Planning**
- **Training**
- **System testing and**
- **Changeover Planning**

Planning is the first task in the system implementation. Planning means deciding on the method and the time scale to be adopted . At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The committee considers ideas, problems and complaints user department, it must also consider;

- **The implication of system environment**
- **Self-selection and allocation form implementation tasks**
- **Consultation with unions and resources available**
- **Standby facilities and channels of communication**

During System Implementation it is essential that everyone involved be absolutely synchronized with the deployment plan and with each other. Often the performance of

deployment efforts impacts many of the Performing Organization's normal business operations. Examples of these impacts include:

Technical Services may be required to assume significant implementation responsibilities while at the same time having to continue current levels of service on other critical business systems.

Because of these and other impacts, the communication of planned deployment activities to all part involved in the project is critical. By this point in the project lifecycle, the team will have spent countless hours devising and refining the steps to be followed.

The final process within the System Development Lifecycle is to transition ownership of the system support responsibilities to the Performing Organization. In order for there to be an efficient and effective transition, the Project Manager should make sure that all involved parties are aware of the transition plan, the timing of the various transition activities, and their role in its execution.

## **6. SCOPE FOR FUTURE ENHANCEMENT**

The development of the **“PREDICTION OF CANCER USING MACHINE LEARNING”** has a good scope in the future as it can be expanded to the needed requirement. There can be few more samples can be added to dataset and the accuracy can be improved in such a way .Those enhancement can be done easily. The Proposed system is in initial stage. By involving few more options the proposed system can be developed. Various algorithmic approach can be used for different datasets so that the accuracy of the system will improve. User can be allowed to choose their own dataset , so that the system will be more interactive to the user. The well defined GUI can be used instead of compiling and running the program in the Python shell , so that it will be more interactive to the user too. The further development of this project can be to host this project to the Internet.

## **7. CONCLUSION**

The main purpose of this minor is predict breast cancer disease by using machine learning approaches. Many time people may have different symptoms and it may not be detected easily whether they have breast cancer or not. This system can help in detecting breast cancer. This proposed system breast cancer-based on Changes in breasts for individual person to Determining cancer for human Body to prevent from major health risks.

In this minor project, we constructed an expert system called cancer prediction system which predict specific cancer “breast cancer” risk, it help the user to predict the cancer. It can save cost and time. This system help the people to know their cancer risk and it also help the people to take appropriate decision based on their cancer risk status.

## **8. BIBLIOGRAPHY**

Machine Learning Mastery with Python BY Jason Brownlee

(Understand Your Data, Create Accurate Models and Work Projects End-To-End)

### **Website Reference:**

1. <https://stackoverflow.com>
2. <https://www.geeksforgeeks.org>
3. <https://www.javatpoint.com/>
4. <https://www.tutorialspoint.com/>
5. <https://www.google.com>

## 9.APPENDICES

### A.SCREENSHOT

**Input :**

```
x1 :5
x2 :1
x3 :1
x4 :1
x5 :2
x6 :1
x7 :3
x8 :1
x9 :1
the data entered is : [5, 1, 1, 1, 2, 1, 3, 1, 1]
values predicted : 2
```

**Output:**

| Id Number | Clump Thickr | Uniformity o | Uniformity o | Marginal Adh | Single Epithe | Bare Nuclei | Bland Chrom | Normal Nucl | Mitoses | Class |
|-----------|--------------|--------------|--------------|--------------|---------------|-------------|-------------|-------------|---------|-------|
| 10156     | 10           | 7            | 7            | 6            | 4             | 10          | 4           | 1           | 2       | 2     |
| 1486      | 7            | 4            | 6            | 4            | 6             | 1           | 4           | 3           | 1       | 2     |
| 158416    | 8            | 7            | 5            | 10           | 7             | 9           | 5           | 5           | 4       | 2     |
| 190       | 1            | 1            | 1            | 1            | 1             | 1           | 1           | 1           | 1       | 2     |
| 344       | 8            | 5            | 6            | 2            | 3             | 10          | 6           | 6           | 1       | 4     |
| 5481      | 10           | 5            | 5            | 3            | 6             | 7           | 7           | 10          | 1       | 2     |
| 79855     | 2            | 3            | 4            | 6            | 7             | 4           | 3           | 5           | 3       | 2     |
| 9         | 9            | 10           | 10           | 10           | 10            | 5           | 10          | 10          | 10      | 4     |
| 9464      | 8            | 10           | 10           | 8            | 7             | 10          | 9           | 7           | 1       | 4     |

## B.SAMPLE CODE

```
#load libraries
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
from collections import Counter
import pandas as pd
import random
from sklearn import neighbors
from sklearn.model_selection import cross_val_score
import pyodbc

#load dataset
df = pd.read_csv('E:\python\cancer.txt')
conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ=E:\python\py_db_1.accdb;')
cursor=conn.cursor()

#data description
print(df.dtypes)
print(df.shape)
print(df.head(20))
print(df.describe())

#graphical representation
df.hist()
plt.show()

#data cleaning
df.replace('?',-99999, inplace=True)
```

```

df.drop(['id'], 1, inplace=True)
full_data = df.astype(float).values.tolist()
print(full_data)

#dataset={2:[[[5.0, 1.0, 1.0, 1.0, 2.0, 1.0, 3.0, 1.0, 1.0],[2,3],[3,1]],4:[[[6,5],[7,7],[8,6]]]}

#train and test datasets
random.shuffle(full_data)
test_size = 0.2
train_set = {2:[], 4:[]} #2 & 4 is output data
test_set = {2:[], 4:[]} #2 is for the benign tumors 4 is for malignant tumors,
train_data = full_data[0:-int(test_size*len(full_data))]
test_data = full_data[-int(test_size*len(full_data)):]
for data in train_data:
train_set[data[-1]].append(data[0:-1])
for data in test_data:
test_set[data[-1]].append(data[0:-1])
test_set

#defining KNN algorithm from scratch
defknn(data, predict, k=3):
    distances = []
    for group in data:
        for features in data[group]:
            euclidean_distance = np.linalg.norm(np.array(features)-np.array(predict))
            distances.append([euclidean_distance,group])
            votes = [i[1] for i in sorted(distances)[:k]]

vote_result = Counter(votes).most_common(1)[0][0]
return vote_result # 2,4

```



```

correct=0
total =0
#new training datasets
for i in test_set:#4
    for pt in test_set[i]:
pr = knn(train_set,pt,200)
    if (pr == i):
        correct +=1
        total +=1
print(correct/total)
correct = 0
total=0
for group in test_set:
    for data in test_set[group]:
        print(group,end='->')
        print(data,end='->')
        vote = knn(train_set, data, k=6)
        print(vote)

    if group == vote:
        correct += 1
        total += 1
print('Accuracy:', correct/total)

#user defined input
new_pt=[]
id=int(input('enter the id :'))
i1=int(input('x1 :'))
i2=int(input('x2 :'))
i3=int(input('x3 :'))
i4=int(input('x4 :'))

```

```

i5=int(input('x5 :'))
i6=int(input('x6 :'))
i7=int(input('x7 :'))
i8=int(input('x8 :'))
i9=int(input('x9 :'))
new_pt.append(i1)
new_pt.append(i2)
new_pt.append(i3)
new_pt.append(i4)
new_pt.append(i5)
new_pt.append(i6)
new_pt.append(i7)
new_pt.append(i8)
new_pt.append(i9)
print('the data entered is :',new_pt)
s=knn(train_set,new_pt,500)
print('values predicted :',s)

#database

cursor.execute("INSERT INTO py_tb VALUES
((?),(?),(?),(?),(?),(?),(?),(?),(?),(?),(?))",(id),(i1),(i2),(i3),(i4),(i5),(i6),(i7),(i8),(i9),s)
conn.commit();

```

**CERTIFICATE**



**DECLARATION**



## **ACKNOWLEDGEMENT**



## SYNOPSIS



## TABLE OF CONTENTS



## INTRODUCTION





**SYSTEM ANALYSIS**

## SYSTEM CONFIGURATION



# SOFTWARE DESCRIPTION



**SYSTEM DESIGN**



## **SYSTEM TESTING AND IMPLEMENTATION**



**SCOPE FOR FUTURE ENHANCEMENTS**



**CONCLUSION**



## **BIBLIOGRAPHY**





## APPENDICES