**Thyroid Disease Classification Using Machine Learning Algorithms**

**1. INTRODUCTION**

Thyroid disease is a subset of endocrinology which is one of the most misunderstood and undiagnosed diseases. Thyroid gland diseases are among the most prevalent endocrine disorders in the world, second only to diabetes, according to the World Health Organization. Hyper function hyperthyroidism and hypothyroidism affect about 2% and 1% of individuals, respectively. Men have about a tenth of the prevalence of women. Hyper-and hypothyroidism may be caused by thyroid gland dysfunction, secondary to pituitary gland failure, or tertiary to hypothalamic malfunction. Due to dietary iodine deficiency, goiter or active thyroid nodules may become prevalent in some regions, with a prevalence of up to 15%. The thyroid gland can also be the location of different kinds of tumors and can be a dangerous place where endogenous antibodies wreak havoc (autoantibodies) . Early disease detection, diagnosis, and care, according to doctors, are vital in preventing disease progression and even death. For several different forms of anomalies, early identification and differential diagnosis raises the odds of good treatment. Despite multiple trials, clinical diagnosis is often thought to be a difficult task. The thyroid gland is a butterfly-shaped gland situated at the base of the throat. It comprises two active thyroid hormones, levothyroxine (T4) and triiodothyronine (T3), which are involved in brain functions such as body temperature control, blood pressure management, and heart rate regulation. Likewise, thyroid disease is one of the most prevalent diseases worldwide, and it is mostly caused by a deficiency of iodine, but it may also be caused by other factors. The thyroid gland is an endocrine gland that secretes hormones and passes them through the bloodstream. It is situated in the middle of the front of the body. Thyroid gland hormones are responsible for aiding in digestion as well as maintaining the body moist, balanced, and so on. Thyroid gland treatments such as T3 (triiodothyronine), T4 (thyroid hormone), and TSH (thyroid stimulating hormone) are used to assess thyroid activity (thyroid stimulating hormone). Thyroid disorder is classified into two types: hypothyroidism and hyperthyroidism. Data mining is a semi-automated method of looking for correlations in massive datasets. Machine learning algorithms are one of the best solutions to many problems that are difficult to solve Classification is a data extraction technique (machine learning) used to predict and identify many diseases, such as thyroid disease, which we researched and classified here because machine learning algorithms play a significant role in classifying thyroid disease and because these algorithms are high performing and efficient and aid in classification. Although the application of computer learning and artificial intelligence in medicine dates back to the early days of the field, there has been a new movement to consider the need for machine learning-driven healthcare solutions. As a result, analysts predict that machine learning will become commonplace in healthcare in the near future. Hyperthyroidism is a disorder in which the thyroid gland releases so many thyroid hormones. Hyperthyroidism is caused by an increase in thyroid hormone levels. Dry skin, elevated temperature sensitivity, hair thinning, weight loss, increased heart rate, high blood pressure, heavy sweating, neck enlargement, nervousness, menstrual cycles shortening, irregular stomach movements, and hands shaking are some of the signs. Hypothyroidism is a condition in which the thyroid gland is underactive Hypothyroidism is caused by a decline in thyroid hormone production. Hypo means deficient or less in medical terms. Inflammation and thyroid gland injury are the two primary causes of hypothyroidism. Obesity, low heart rate, increased temperature sensitivity, neck swelling, dry skin, hand numbness, hair issues, heavy menstrual cycles, and intestinal problems are some of the symptoms. If not treated, these symptoms can escalate over time.

* 1. **Objective of the Project**

With the vast amount of data and information difficult to deal with, especially in the health system, machine learning algorithms and data mining techniques have an important role in dealing with data. In our study, we used machine learning algorithms with thyroid disease. The goal of this study is to categorize thyroid disease into three categories: hyperthyroidism, hypothyroidism, and normal, so we worked on this study using data from Iraqi people, some of whom have an overactive thyroid gland and others who have hypothyroidism, so we used all of the algorithms. Support vector machines, random forest, decision tree, naïve bayes, logistic regression, k-nearest neighbors, multilayer perceptron (MLP), linear discriminant analysis. To classification of thyroid disease.

**2. LITERATURE SURVEY**

**Thyroid disorder is classified using different classification models based on parameters such as TSH, T4U, and goiter in this study**.

Several grouping methods, such as K-nearest neighbor, are used to justify this argument. The Naive Bayes and support vector machines algorithms are employed. The experiment was carried out using the Rapid miner instrument, and the findings indicate that K-nearest neighbor is more effective than Naive Bayes in detecting thyroid disease. To diagnose thyroid disorder, the researchers used data mining classifiers. Thyroid disorder is a vital factor to consider when diagnosing a disease. KNN and Naive Bayes classifiers were used in this study. The Rapid miner tool is used to compare these two classifiers. The findings revealed that the K-nearest neighbor classifier is the most reliable, with a 93.44 percent accuracy, while the Naive Bayes classifier has a 22.56 percent accuracy. The proposed KNN technique improves classification accuracy, which contributes to improved results. As a result, Naive Bayes can only have a linear, elliptic, or parabolic decision boundary, so the decision boundary consistency of KNN is a huge plus. KNN outperforms most methods since the factors are interdependent.

**Thyroid disease is one of the most common illnesses that humans suffer from.**

The hypothyroid data used in this study came from the data repository at the University of California, Irvine (UCI). The platform Waikato Environment of Information Analysis will be used for the whole research project (WEKA). The J48 technique was found to be more effective than the decision stump tree technique. In the world of health care, disease diagnosis is a difficult challenge. In the decision-making method, a number of data mining methods are used. In this analysis, we used dimensionality reduction to pick a subset of attributes from the original results, and we used J48 and decision stump data mining classification techniques to define hypothyroidism. The uncertainty matrix is used to assess classifier output in terms of precision and error rate. The J48 Algorithm has 99.58 percent accuracy, which is higher than decision stump tree accuracy, and it also has a smaller error rate than Decision stump.

**Classification, which is used to characterize predefined data sets, is one of the most popular supervised learning data mining techniques.**

In the healthcare sector, the classification is commonly used to aid in medical decision-making, diagnosis, and administration. The information for this study was gathered from a well-known Kashmiri laboratory. The entire research project will be conducted on the ANACONDA3-5.2.0 platform. In an experimental analysis, classification methods such as k nearest neighbors, Support vector machine, Decision tree, and Nave bayes may be used. The Judgment Tree has the greatest accuracy of the other classes, at 98.89 percent.

**Thyroid disorder is a chronic illness that affects people all over the world**.

Data mining in healthcare is producing excellent results in the prediction of different diseases. The accuracy of data mining techniques for prediction is high, and the cost of prediction is low. Another significant benefit is that prediction takes very little time. In this study, I used classification algorithms to analyze thyroid data and came up with a result. A model's efficacy is primarily determined by two factors. The first is prediction precision, and the second is prediction time. According to our findings, Nave Bayes took just 0.04 seconds to forecast. However, it is less accurate than J48 and Random Forest. When we looked at prediction accuracy, the Random Forest model came in at 99.3 percent. However, the model's construction time is longer than the other two iterations. So we can assume that J48 is the best model for hypothyroid prediction since its accuracy is 99 percent, which is among the highest, and it takes 0.2 seconds to run, which is significantly less time than the Random Forest model.

**The aim of this study is to propose a data mining-based method for enhancing the precision of hypothyroidism diagnosis by integrating patient questions with test results during the diagnosis process.**

Another goal is to reduce the risks that come with dialysis interventional trials. The logical conclusion It was determined if the new samples were hypothyroid using data from the UCI machine learning database, which included 3163 samples, 151 of which were hypothyroid and the others were hypothyroid. Different sampling techniques were used in the data collection to eradicate the unbalanced distribution, and models were developed to diagnose hypothyroidism using Logistic Regression, K Nearest Neighbor, and Support Vector Machine classifiers. The thesis demonstrated the impact of sampling techniques on the diagnosis of hypothyroidism in this regard. The Logistic Regression classifier produced the best results of all the models created. The precision was 97.8%, the F-Score was 82.26 percent, the region under the curve was 93.2 percent, and the Matthews correlation coefficient was 81.8 percent for this analysis, which was trained on the data set using over-sampling techniques.

**The aim of this paper is to create a method that can predict diabetes in a patient early and accurately using the Random Forest algorithm in a machine learning technique.**

Random Forest algorithms are a type of ensemble learning system that is commonly used for classification and regression tasks. As compared to other algorithms, the performance ratio is higher. The suggested model gives the best outcomes for diabetic prediction, and the results revealed that the prediction system is capable of correctly, effectively, and most importantly, immediately forecasting diabetes disease.

**After all other cancers, breast cancer is the second most common cancer in women.**

The aim of this research paper is to provide a breast cancer study that incorporates cutting-edge techniques. Improving breast cancer survivability modeling models by incorporating recent research advances. We used a broad dataset and three common data mining algorithms (Nave Bayes, RBF Network, and J48) to construct prediction models (683 breast cancer cases). For accuracy comparison, we used 10-fold cross-validation approaches to measure the unbiased estimation of the three prediction models. The findings suggest that the Bay is a safe place to visit (based on an average precision Breast Cancer dataset). The RBF Network is the second-best predictor, with 93.41 percent accuracy on the holdout sample (better than any other prediction accuracy reported in the literature), and Nave Bayes is the third-best predictor, with 97.36 percent accuracy on the holdout sample (better than any other prediction accuracy reported in the literature) (better than any other prediction accuracy published in the literature). In this study, we evaluated three breast cancer survivability prediction models using two criteria: benign and malignant cancer cases.

**The most recent research focuses on thyroid disease classification of two of the most frequent thyroid dysfunctions in the general population (hyperthyroidism and hypothyroidism).** The researchers looked at and compared four different classification models: Naive Bayes, Decision Trees, Multilayer Perceptron, and Radial Basis Function Networks. The findings reveal that all of the classification models listed above have a high degree of accuracy, with the Decision Tree model having the highest classification score. The classifier was built and validated using data from a Romanian data website and the UCI machine learning repository. KNIME Analytics Platform and Weka are two data sets. Data mining techniques were used as the foundation for developing and testing the classification models. A variety of studies in the field of thyroid classification use various data mining techniques to construct robust classifiers, according to the literature. The authors of this research explored the use of four classification models on thyroid data (Nave Bayes, Decision Tree, MLP, and RBF Network) to help classify thyroid dysfunctions such as hyperthyroidism and hypothyroidism. In all of the cases that were tested, the decision tree model was the correct classification model.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

At present, diseases have become dangerous and rapidly spread, and their exploration and diagnosis require a great deal of time and effort. The correct and accurate diagnosis of the disease early has become one of the problems that the health system suffers from it. The critical role of early and correct diagnosis of the disease, including thyroid disease, is vital because it increases patient treatment opportunities and reduces mortality. Among the vast amount of clinical data, early diagnosis is a challenging task. Today the machine learning has had impressive and good results in many sciences. Hence, it had a prominent and valuable role in diseases, so this study used machine learning algorithms with thyroid disease. To detect and classify thyroid disease into three types hyperthyroidisms, hypothyroidism, and normal.

**Disadvantage**

1. Low accuracy
2. high-power computing

**PROPOSED SYSTEM**

In this paper author employing various machine learning algorithms such as SVM, Naïve Bayes, Decision Tree, Random Forest, KNN and MLP to predict thyroid disease. Each algorithm gets trained on thyroid dataset which contains 3 different classes such Normal, Hyperthyroid and Hypothyroid. All algorithms performance is evaluated in terms of accuracy, precision, recall, FSCORE and confusion matrix and in all algorithms Random Forest and Decision Tree giving best accuracy.

**Advantage**

1. access to a significant amount of data

2. risk factors influence disease progression

3. more accurate

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify. Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to. the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user-friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**Operating Environment**

Windows XP.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows7/8
* Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic 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 objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

**Class Diagram:**



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**Sequence diagram:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. 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**, **event scenarios**, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



**Component Diagram:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**Deployment Diagram:**



**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

Upload Thyroid Disease Dataset

Process Dataset

Run Decision Tree Algorithm

Run SVM Algorithm

Run Random Forest

Run Naive Bayes

Run Logistic Regression

Run KNN Algorithm

Run MLP Algorithm

Comparison Graph

Predict Thyroid from Test Data

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1. Upload Thyroid Disease Dataset 2. Successfully Upload Thyroid Disease Dataset

3. Preprocess Dataset 4. Successfully Preprocess Dataset

5. Run Decision Tree Algorithm 6. Successfully Run Decision Tree Algorithm

7. Run SVM Algorithm 8. Successfully Run SVM Algorithm

9. Run Random Forest 10. Successfully Run Random Forest

11. Run Naive Bayes 12. Successfully Run Naive Bayes

13. Run Logistic Regression 14. Successfully Run Logistic Regression

15. Run KNN Algorithm 16. Successfully Run KNN Algorithm

17. Run MLP Algorithm 18. Successfully Run MLP Algorithm

19. Comparison Graph 20. Successfully Comparison Graph

21. Predict Thyroid from Test Data 22. Successfully Predict Thyroid from Test Data

**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQL db.

Standard libraries in Python are well tested and used by hundreds of people. So, you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

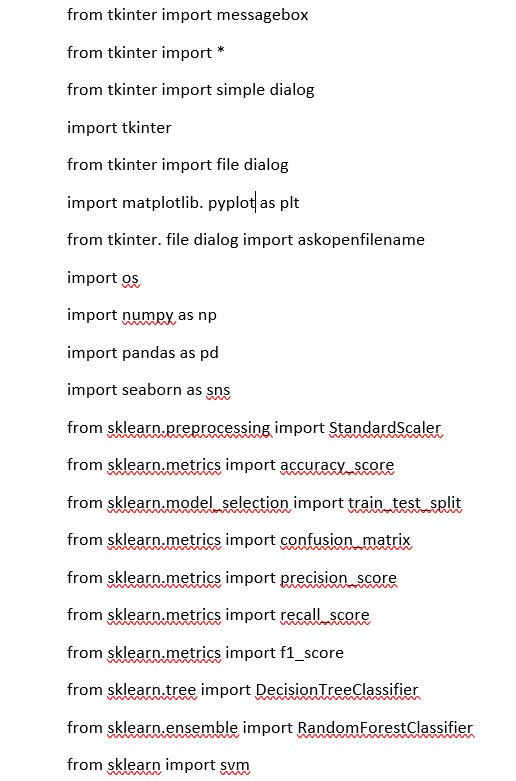
**3. Expressiveness of the language**

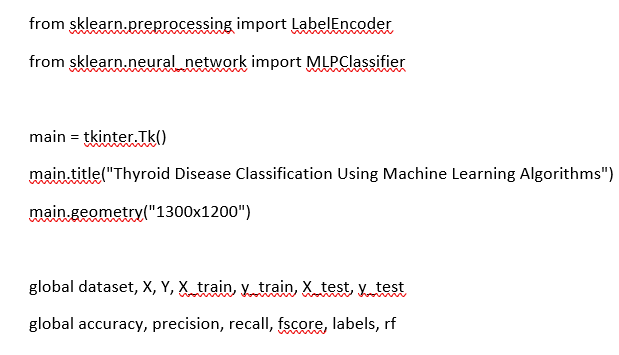
Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

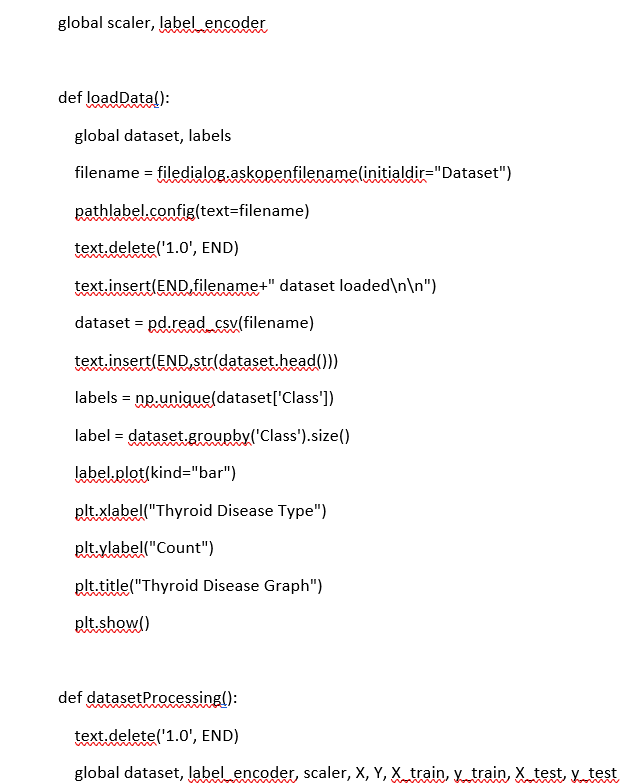
**4. Great Community and Support**

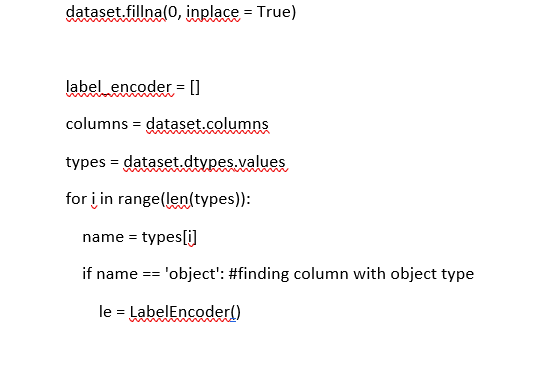
Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

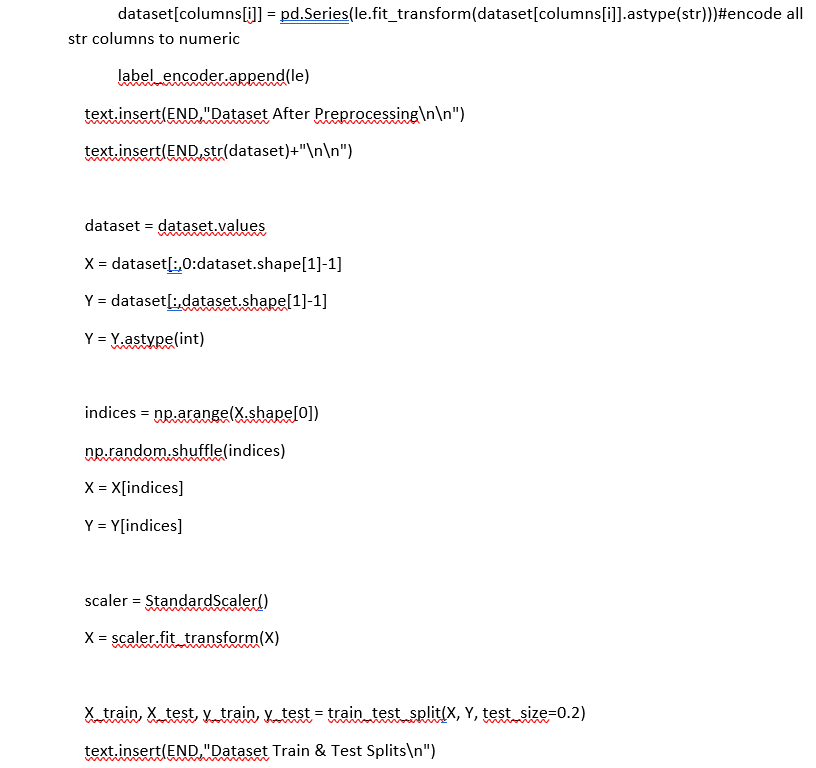
**5.2 Sample Code:**

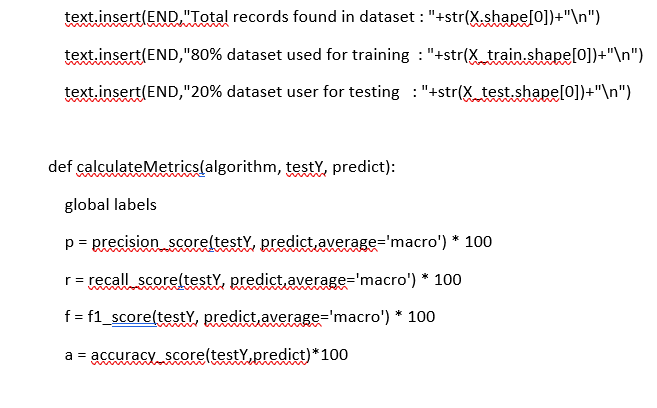


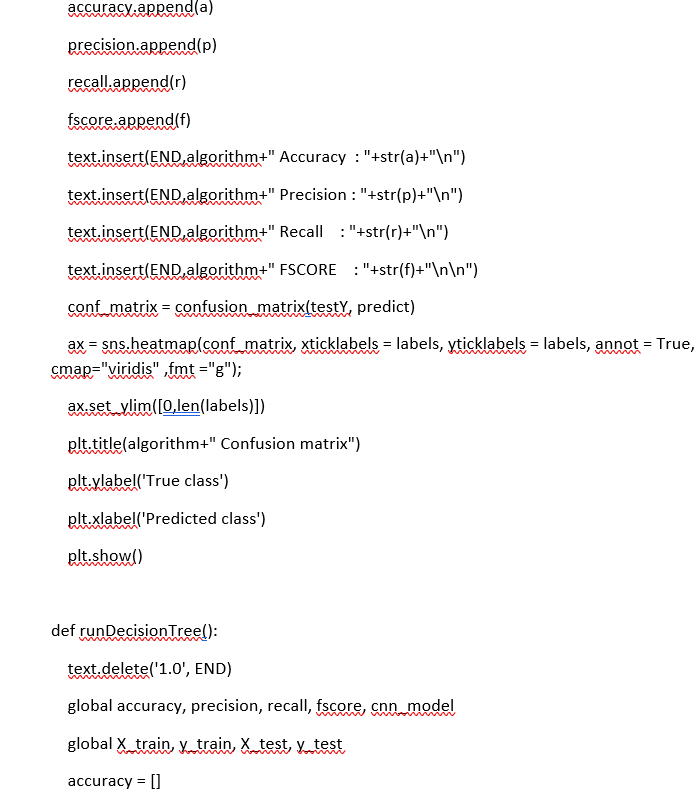


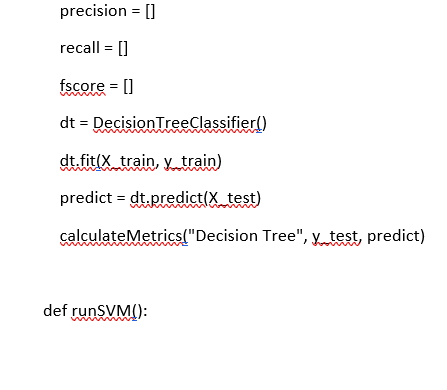


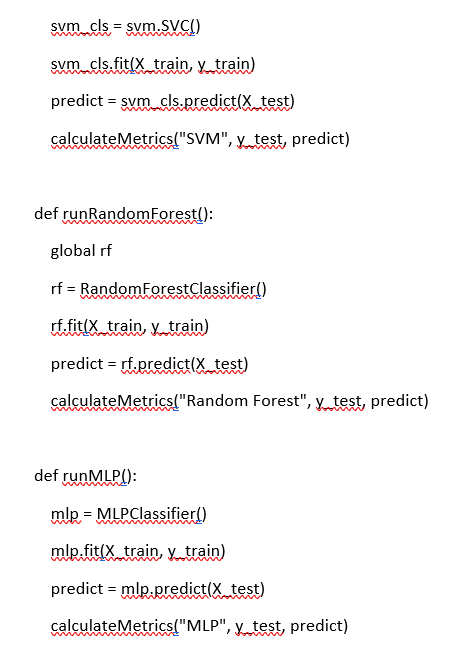


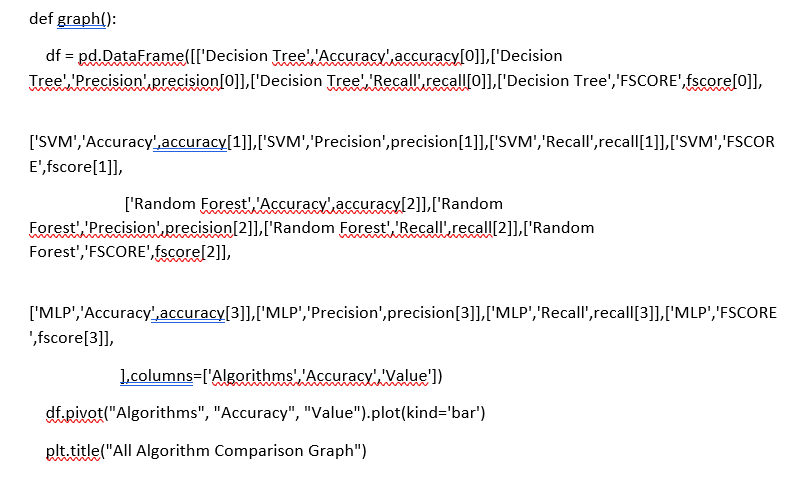


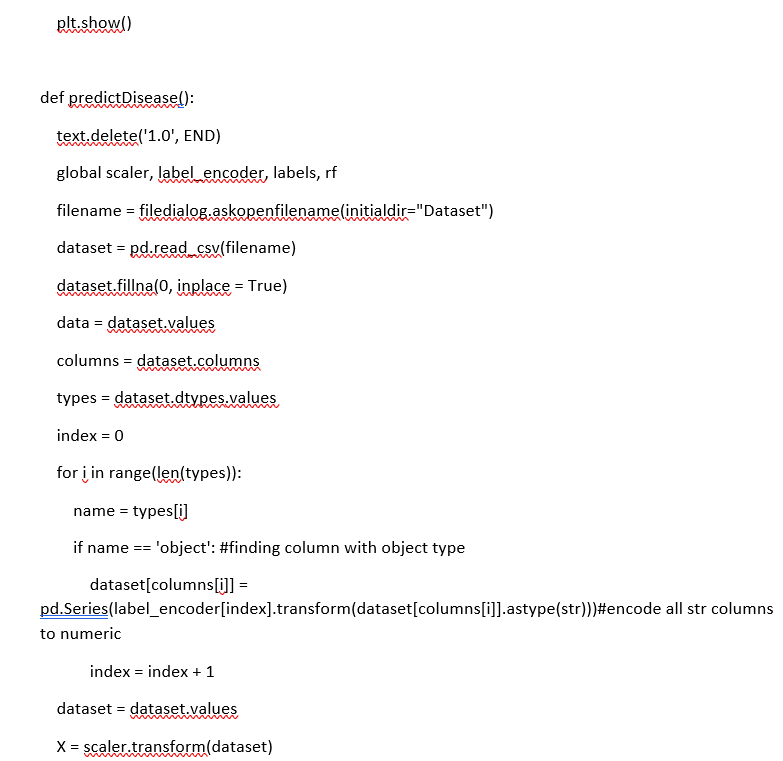


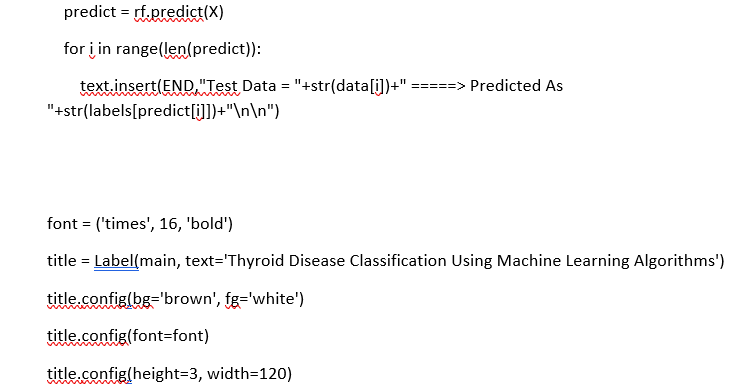


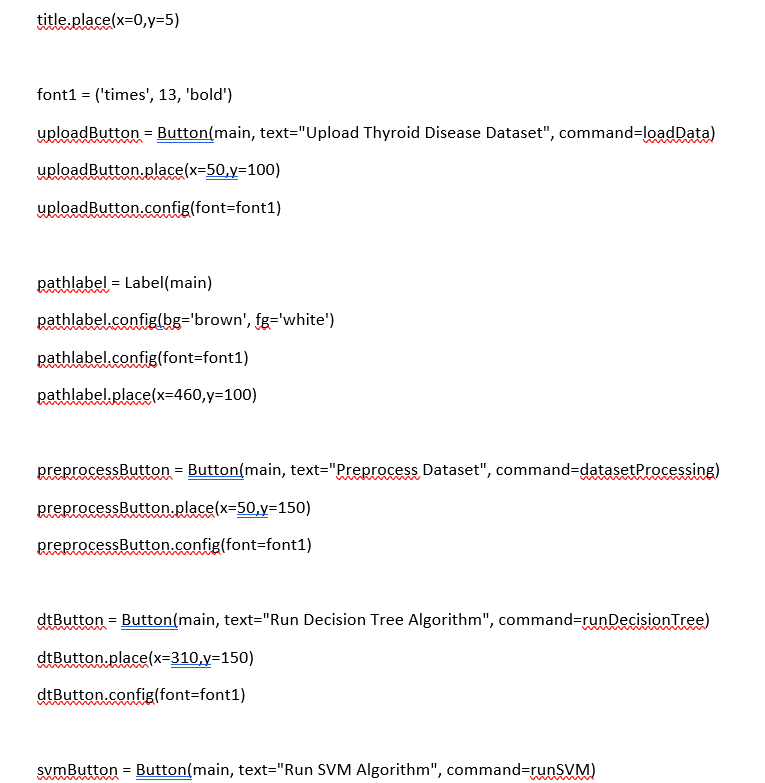


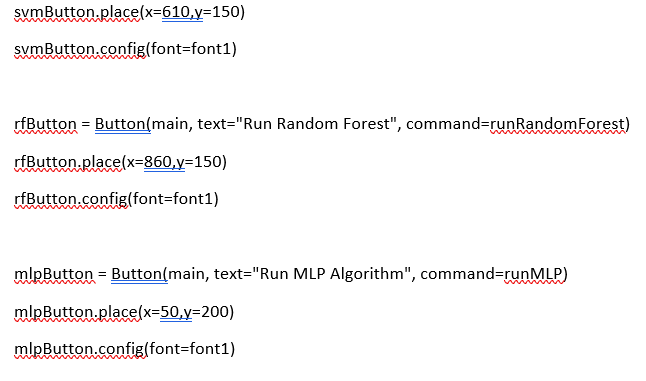


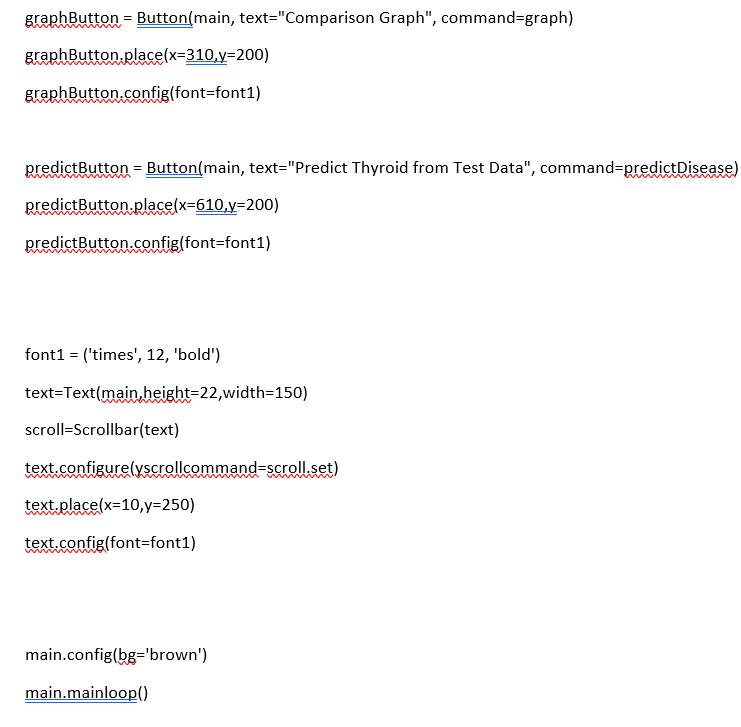












**6. TESTING**

**Testing and implementation:**

Implementation is one of the greatest and most crucial tasks, and it's also a time when caution must be used while all of the activities will include a lot of participation. Implementation is the most crucial phase in developing an efficient structure and earning users' confidence that the revised one is practical and effective. Each software program me is tested independently at the time of development using sample data to make sure it integrates into various programmers in the way specified in its own specification. the evaluation of a computing substrate and its surrounding environment for user satisfaction.

**Implementation**

Innovativeness in system design outweighs it in execution. Its primary focuses are user training and file conversion. The system could need intensive user retraining. Programming requires that the system's default settings be modified. A simple working method is provided to make it fast and simple for the user to understand the many possibilities. Both dots-per- inch and inkjet printers are available to the user, and both may be utilized for producing the different reports. The recommended approach is easy to put into practice. The process of converting a freshly developed or upgraded structure of systems into a functional one is often referred to as "implementation".

**Testing**

Prior to applying field validation throughout the procedure of testing, test data is prepared and used to test each of the modules independently. The system testing that follows ensures that each component of the system operates as a cohesive whole. The testing information should be selected such that it could withstand any scenario. Before the real operation begins, the system is put through actual testing to make sure it functions properly and effectively. The testing methods used throughout the testing period are described in the paragraphs that follow.

**system evaluation**

Testing has become an integral part of every system and project, particularly in the field of information technology. Testing prior advancement is crucial since it provides as a way of assessing preparedness to go forward, whether it's to assess one's capacity to resist the stresses of a certain setting or to confirm if they are capable of undertaking so. The application must be checked to make sure it is performing the desired purpose before being made accessible to users for use. This procedure employs a variety of testing techniques to ensure the reliability of the programming. The program's logic was tested, and each time the program me executed for a certain piece of data, the same pattern was followed. As a consequence, both the code and what emerged were carefully scrutinized for any possible suitable information.

**Module Testing**

Each module is checked independently to find any flaws. This makes it possible for us to identify errors and fix them without impacting other modules. The program me must be corrected if it fails to perform the needed function in order to provide the desired outcome. As a result, each module is evaluated separately beginning with the tiniest and weakest module and working up to the next phase. The system's modules are examined independently. For instance, the module for job categorization is evaluated individually. The course is put through a number of activities, each of which has an estimated completion time. The outcomes of the test are then contrasted with those that were produced manually. The comparison shows that the recommended system outperforms the existing one in terms of effectiveness. The modules of the system are individually inspected. The processing time is shortened by this system's independent testing of the resource category and job scheduling modules, which yields results that are consistent.

**Integration Testing**

Integrated testing is used after module examination. The possibility of problems while interconnecting the modules exists; these errors are corrected using this testing. This system's components are all connected and put through testing. The test results are highly precise. The technology properly maps assignments to resources as a result.

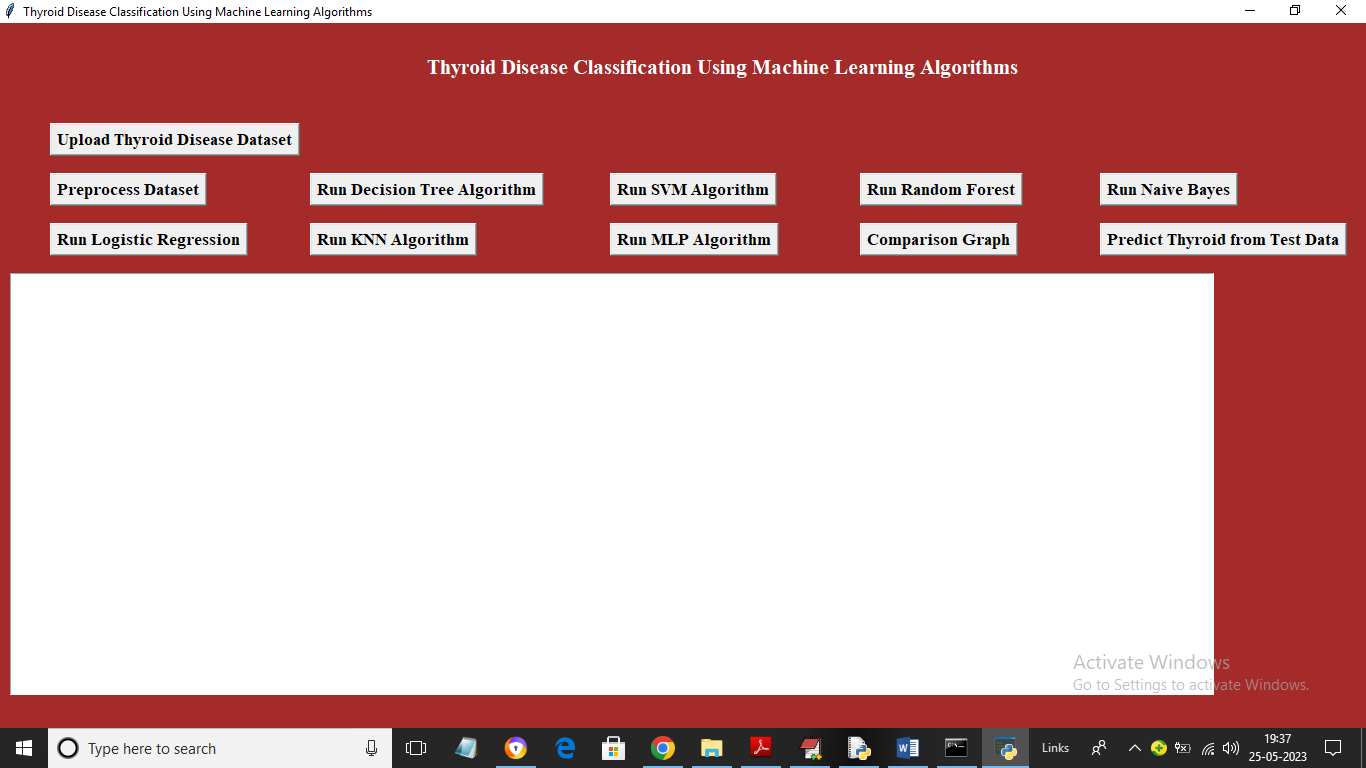
**Acceptance Testing**

A last acceptance test is passed by the system when the user found no significant accuracy issues. This test verifies that the system needs the original goals, objectives, and requirements specified during analysis without actually carrying them out, hence removing the need for time- and money-wasting evaluations of acceptance on the shoulders consumers and management.

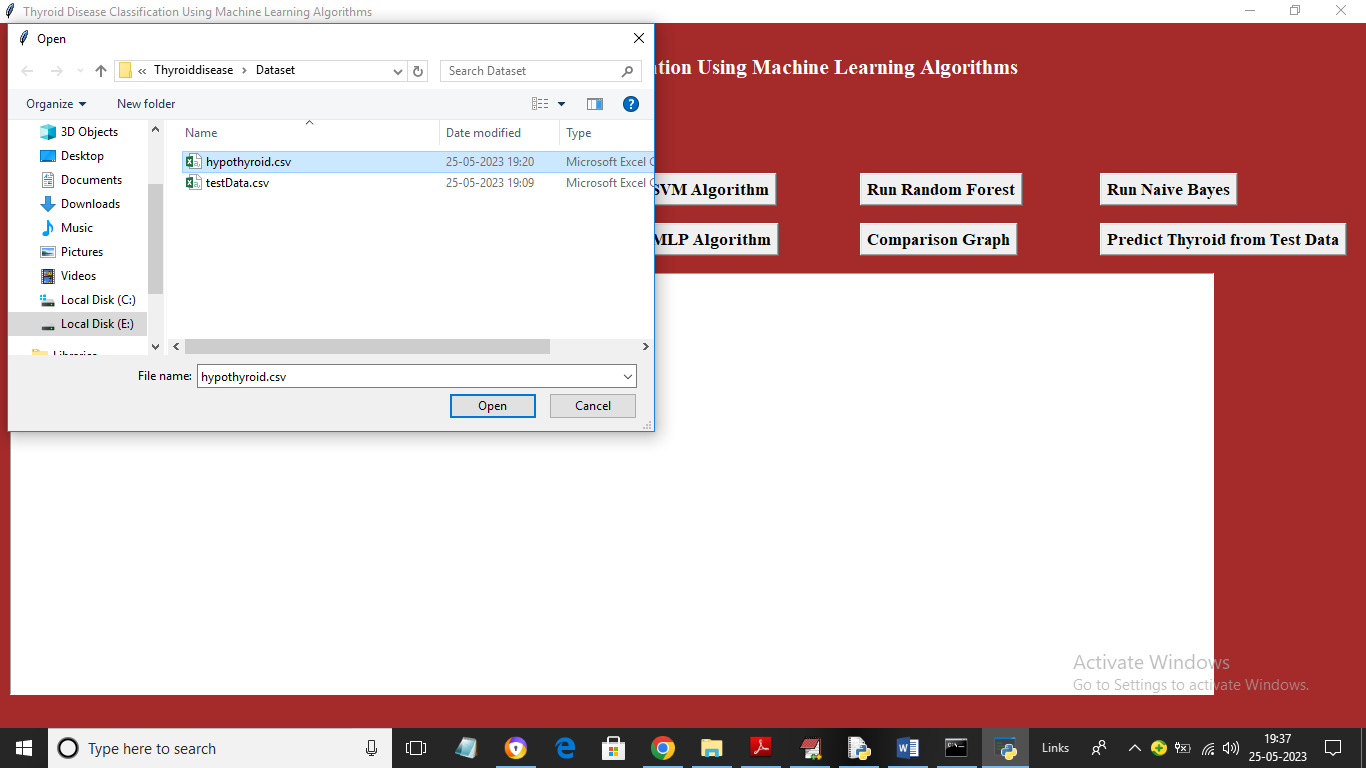
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload Dataset | Test whether the dataset is uploaded or not | If the Dataset is not uploaded | we cannot do further operations | If Dataset is uploaded we will do further operations | High | High |
| 02 | Preprocess Dataset | Verify the Dataset is Preprocessed  or not | If the Dataset is not preprocessed | We cannot do further operations | If Dataset is Preprocessed We Can do further operations | High | High |
| 03 | Run Decision Tree Algorithm | Verify whether Decision Tree runned or not | Without Running Decision Tree | we cannot do further operations | If Decision Tree Algorithm Runned We can do further operations | High | High |
| 04 | Run SVM | Verify whether SVM runned or not | Without Running SVM | we cannot do further operations | If SVM Algorithm Runned we can do further operations | High | High |
| 05 | Run Random Forest Algorithm | Verify whether Random Forest AlgorithmRunned or not | Without Running Random Forest Algorithm | We cannot do further operations | If Random Forest Algorithm Runned We can do further operations | High | High |
| 06 | Run Naïve Bayes Algorithm | Verify whether Naïve Bayes Algorithm Runned or not | Without Running Naïve Bayes Algorithm | we cannot do further operations | If Naïve Bayes Algorithm Runned we can do further operations | High | High |
| 07 | Run Logistic Regression | Verify whether Logistic Regression Runned or not | Without Running Logistic Regression | we cannot do further operations | If Logistic Regression Runned we can do further operations | High | High |
| 08 | Run KNN Algorithm | Verify whether KNN Algorithm Runned or not | Without Running KNN Algorithm | We cannot do further operations | If KNN Runned we can do further operations | High | High |
| 09 | Run MLP Algorithm | Verify whether MLP Runned or not | Without Running MLP Algorithm | We cannot do further operations | If MLP Runned we can do further operations | High | High |
| 10 | Comparison Graph | Verify Graph obtained or not | Without Graph | We cannot do further operations | If Graph is Obtained we can do further operations | High | High |
| 11. | Predict Thyroid | Verify Thyroid predicted or not | Without Thyroid | We cannot do further operations | If Thyroid predicted we can do further operations | High | High |

**7. SCREENSHOTS:**

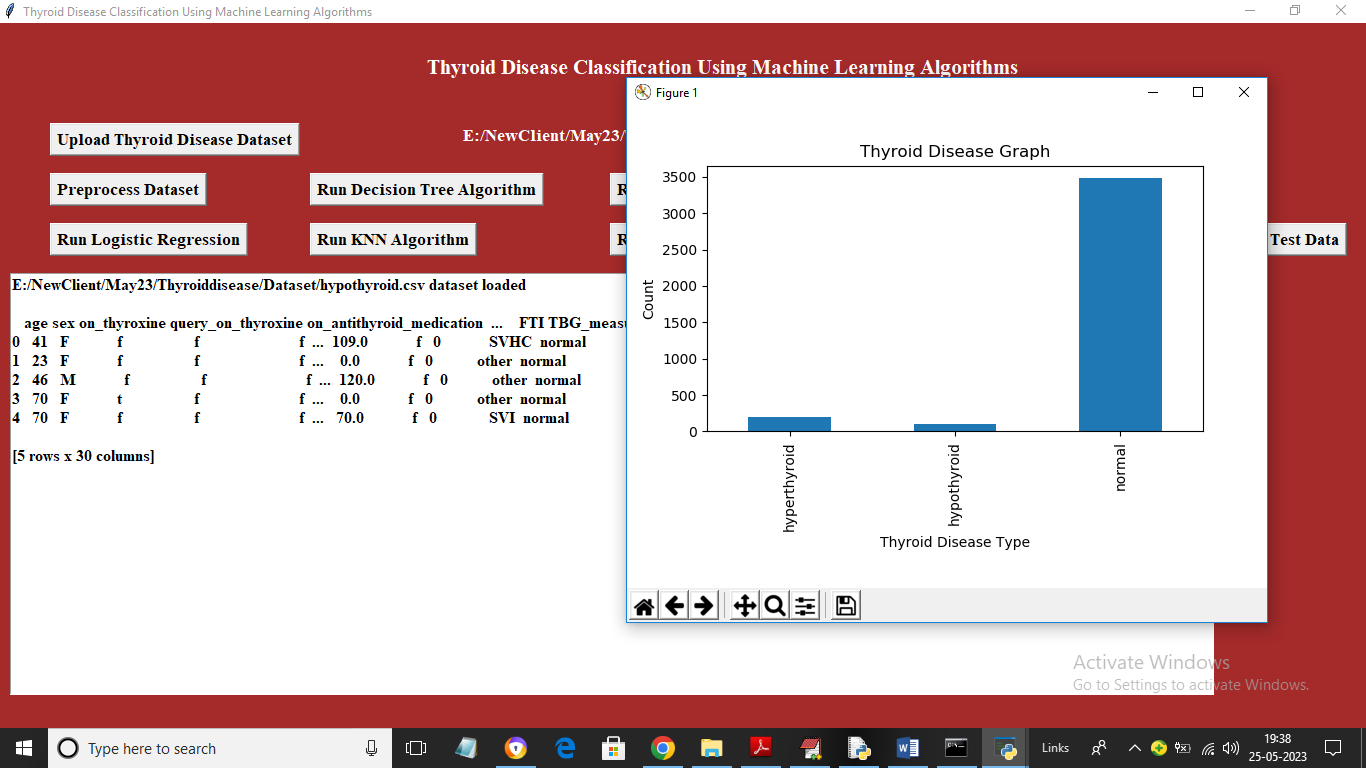
To run project double click on ‘run.bat’ file to get below screen



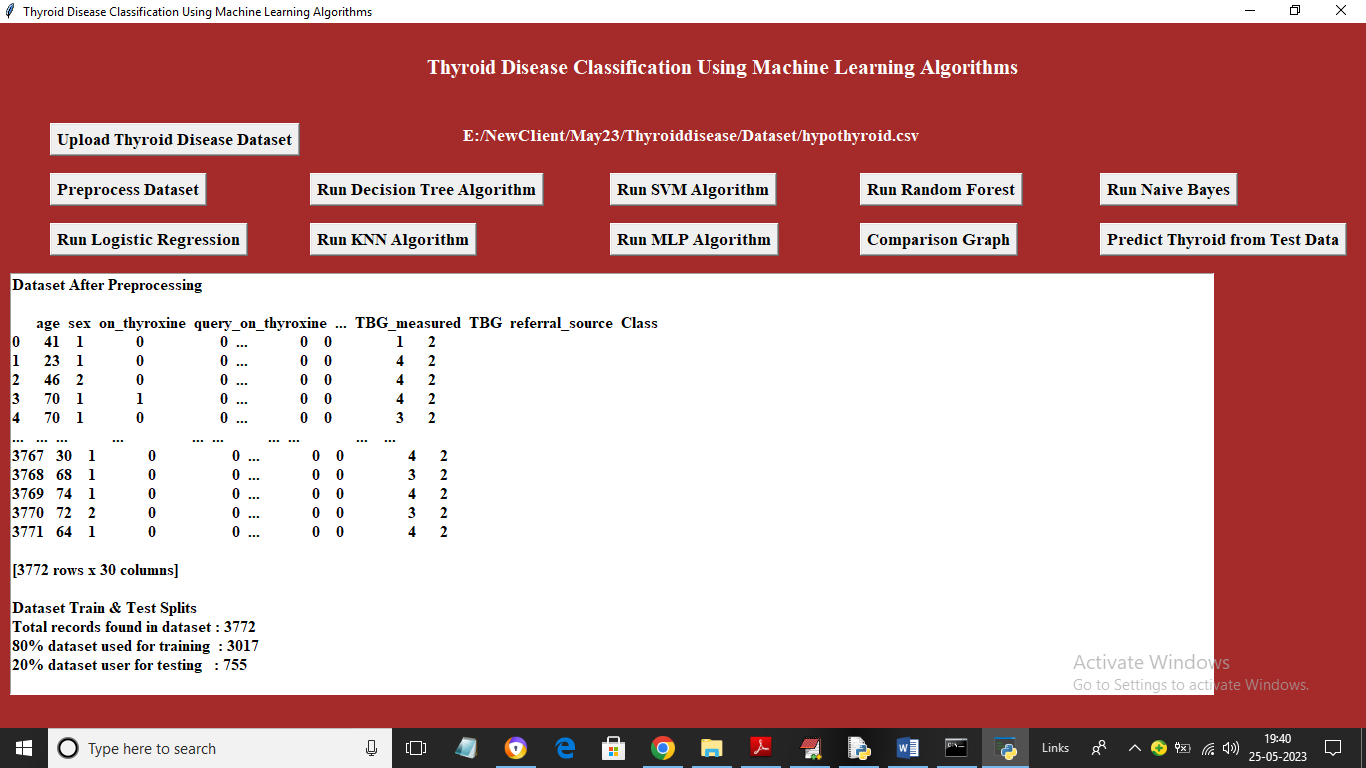
In above screen click on ‘Upload Thyroid Disease Dataset’ button to upload dataset and get below screen



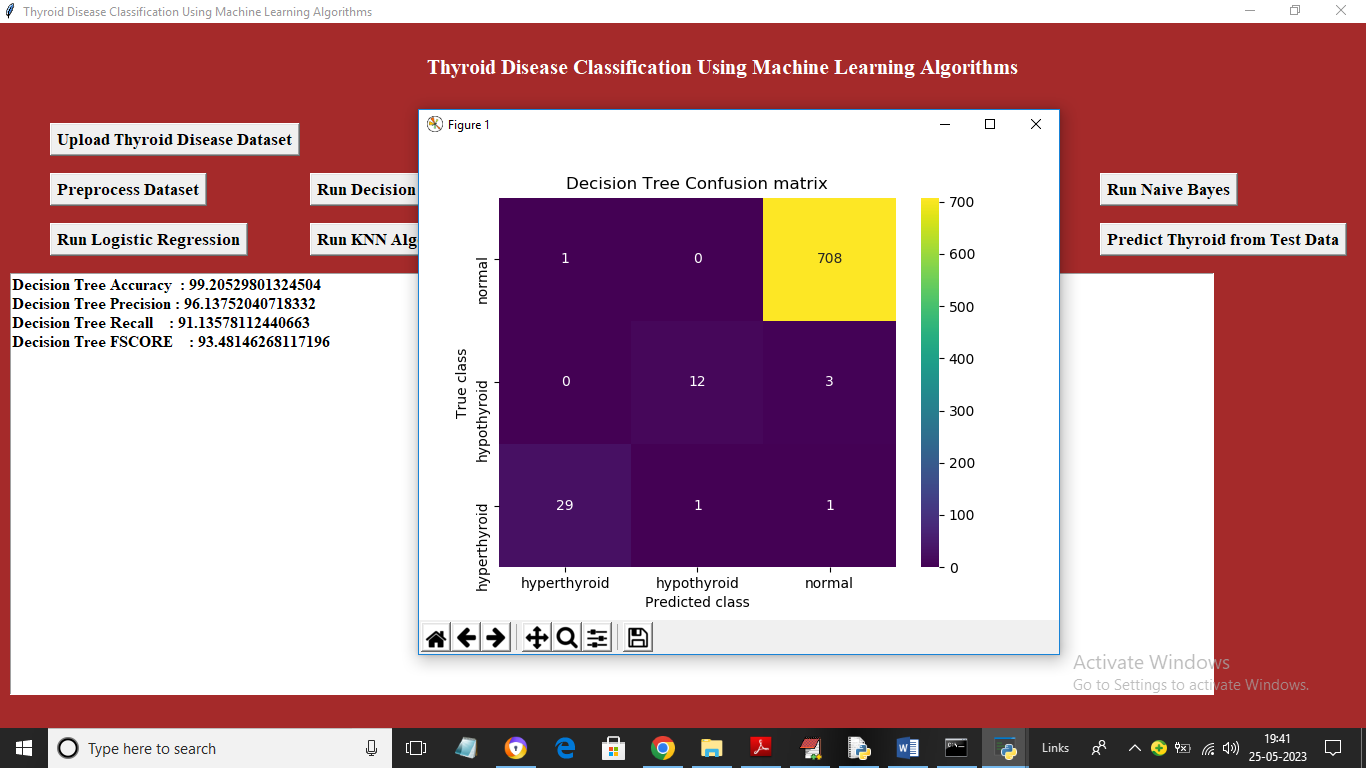
In above screen selecting and uploading dataset file and then click on ‘Open’ button to load dataset and get below output



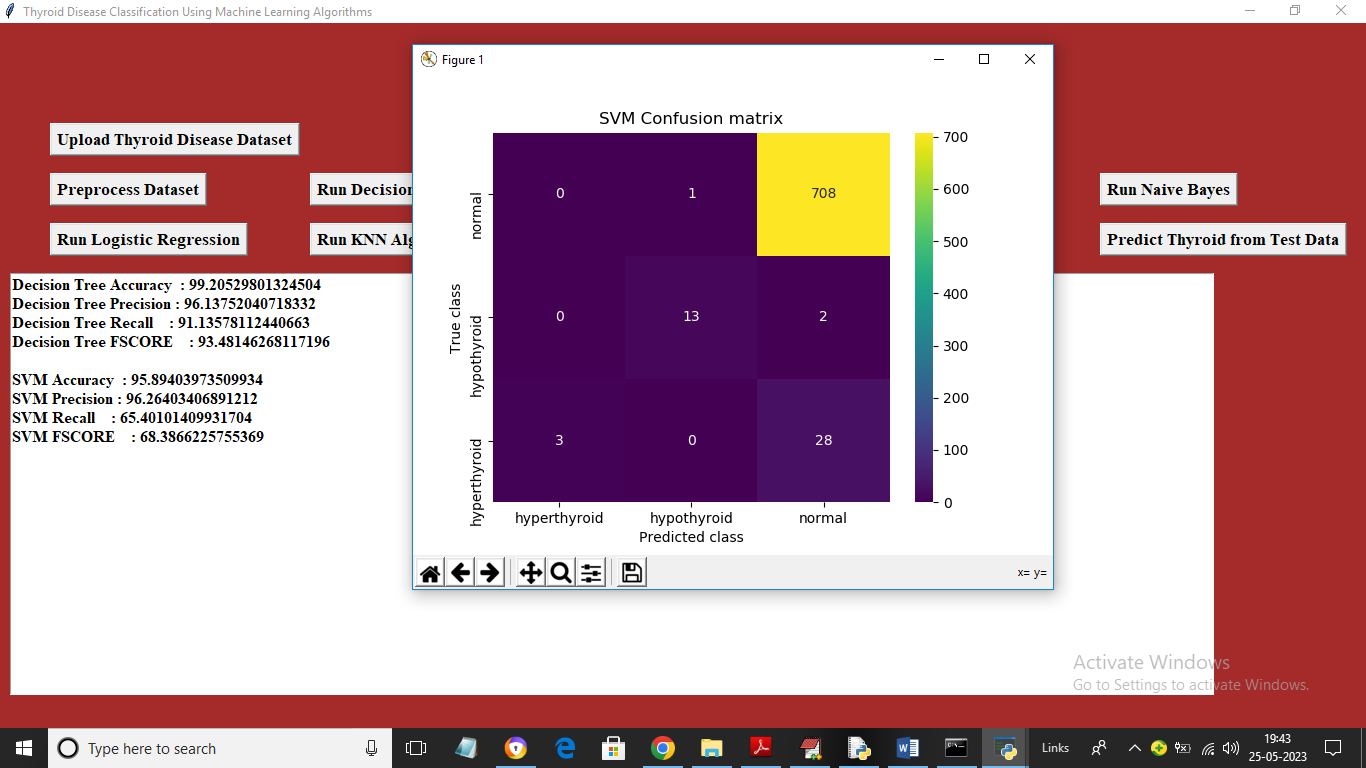
In above screen dataset loaded and we can see dataset contains some numeric and some non-numeric values so by applying Processing technique we can clean above data. In above graph x-axis represents thyroid disease type and y-axis represents count of each disease. Now close above graph and then click on ‘Preprocess Dataset’ button to process dataset and get below output



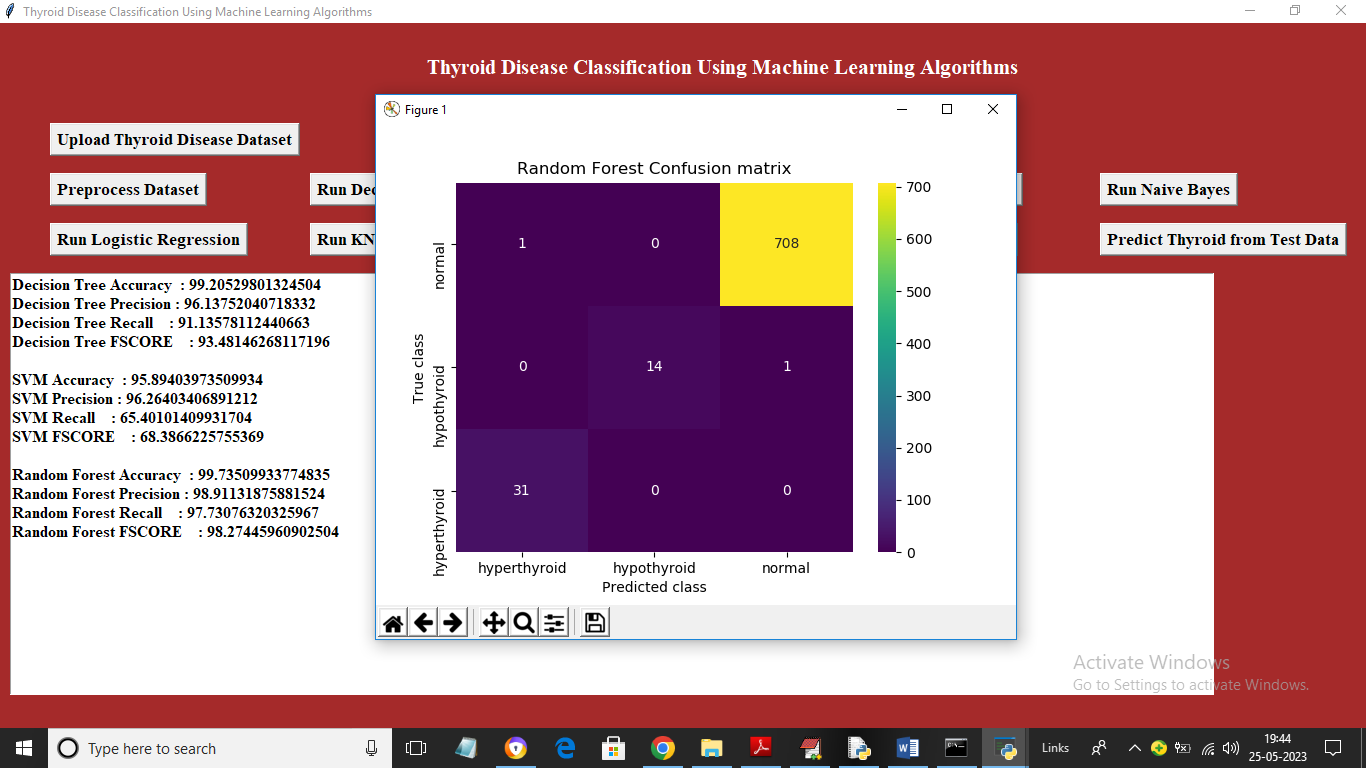
In above screen we can see entire dataset is now converted to numeric values and in last line we can see size of dataset and train and test split details. Now click on ‘Run Decision Tree Algorithm’ button to train decision Tree and get below output



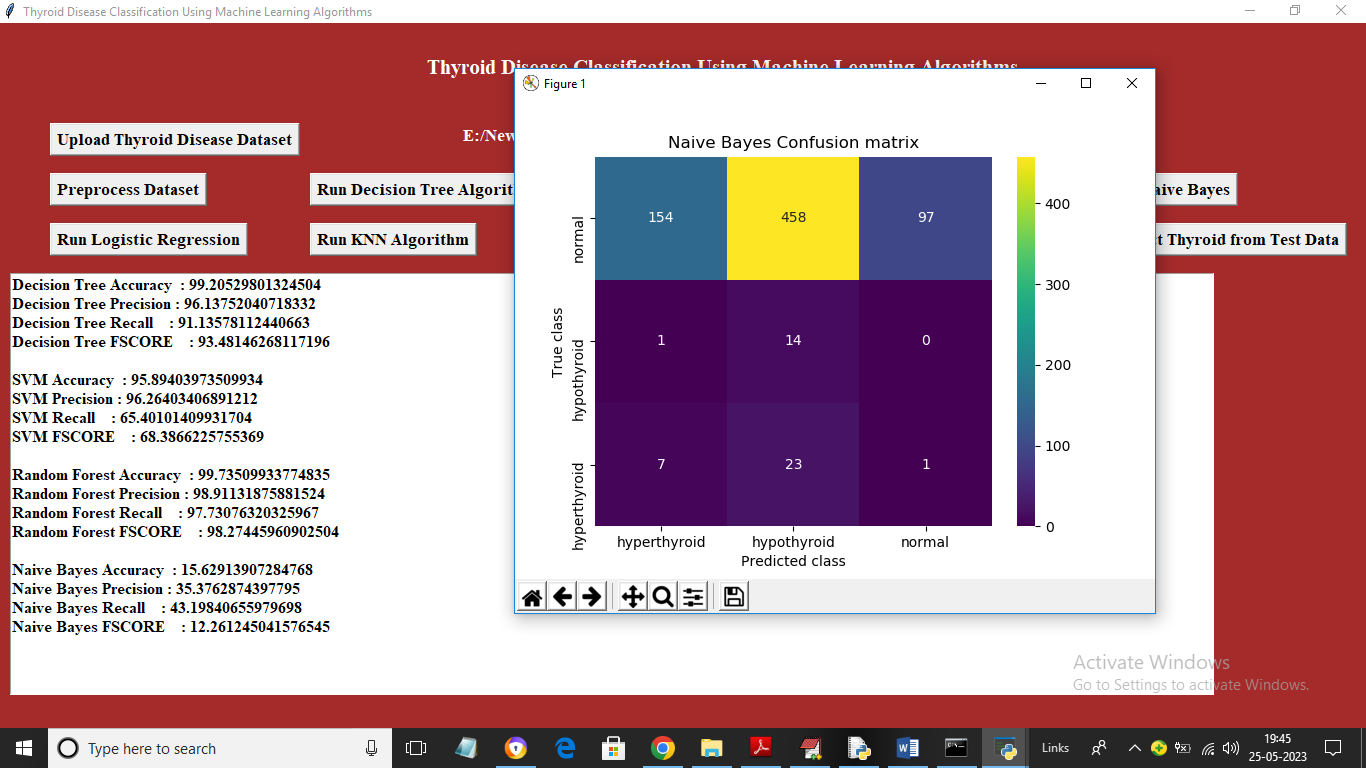
In above screen with decision tree we got accuracy as 99% and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all boxes in yellow colour box diagnol contains correct prediction count and remaining boxes contains incorrect prediction count which are very few. Now close above graph and then click on ‘Run SVM Algorithm’ button to get below output



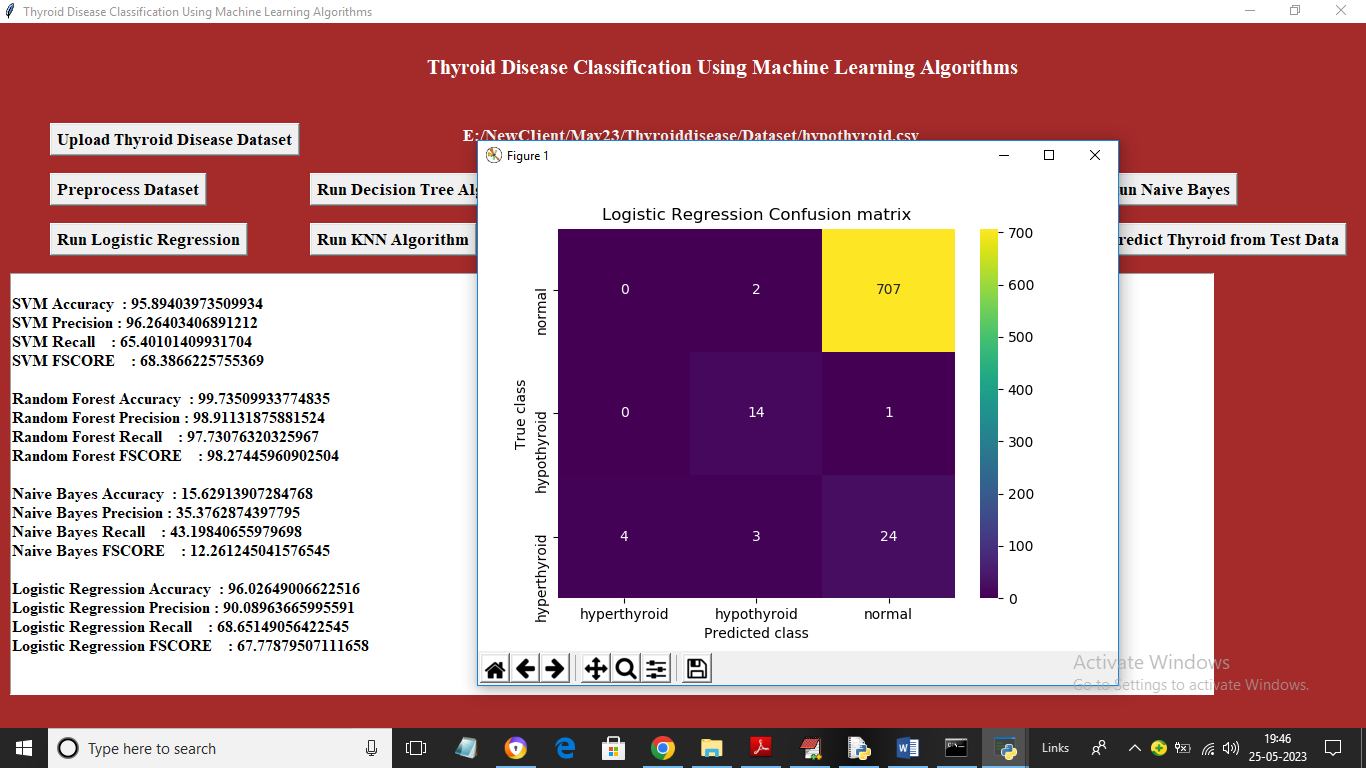
In above screen with SVM we got 95% accuracy and we can see other metrics and confusion matrix graph. Now click on ‘Run Random Forest Algorithm’ button to train Random Forest and get below output



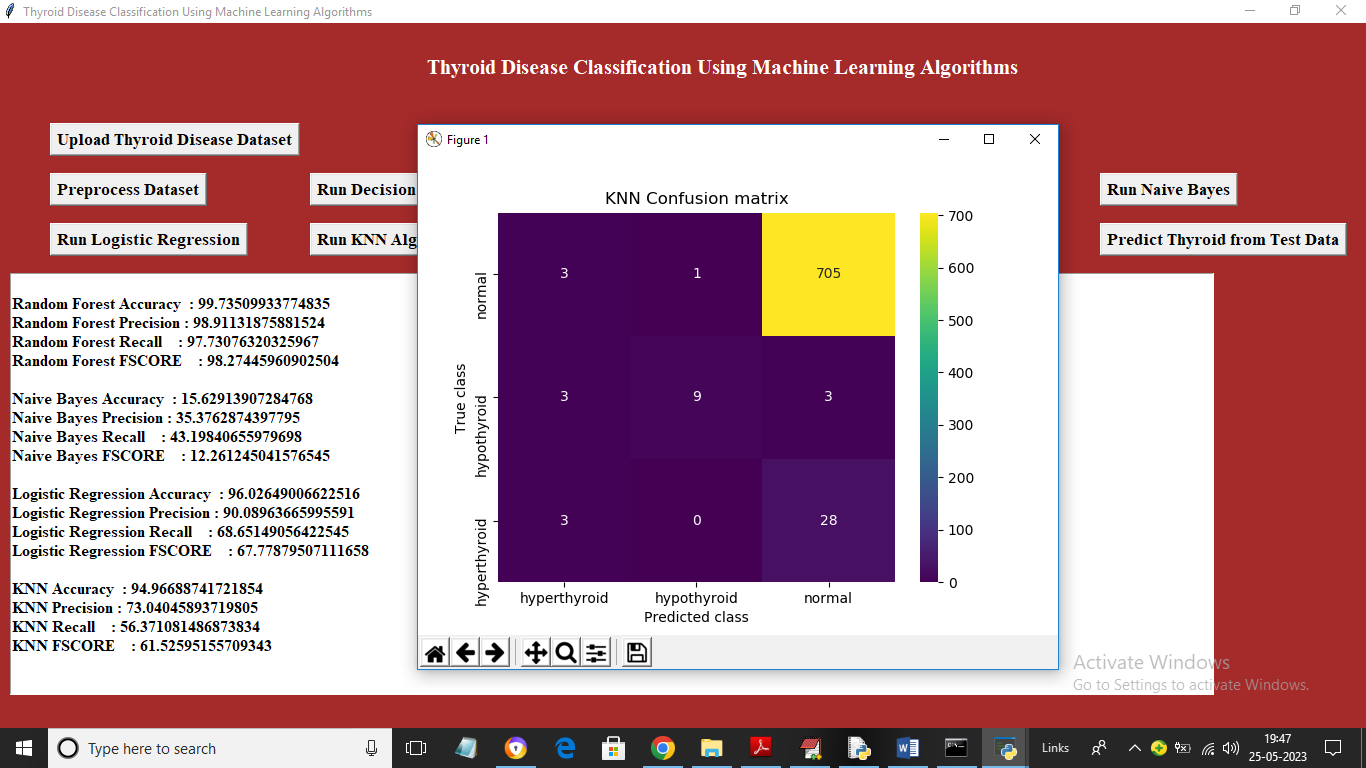
In above screen with Random Forest we got 99% accuracy and now click on ‘Run Naïve Bayes Algorithm’ button to get below output



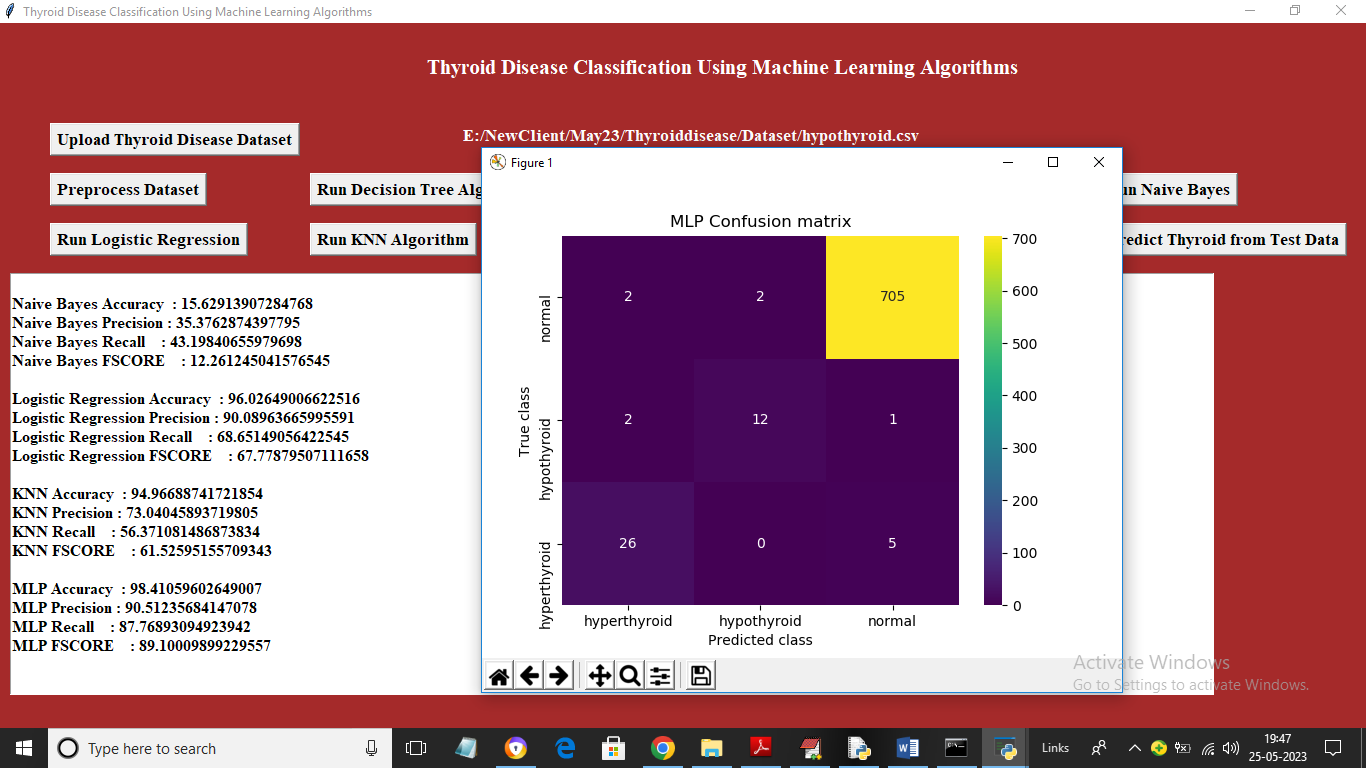
In above screen Random Forest got 15% accuracy and now click on ‘Run Logistic Regression’ button to get below output



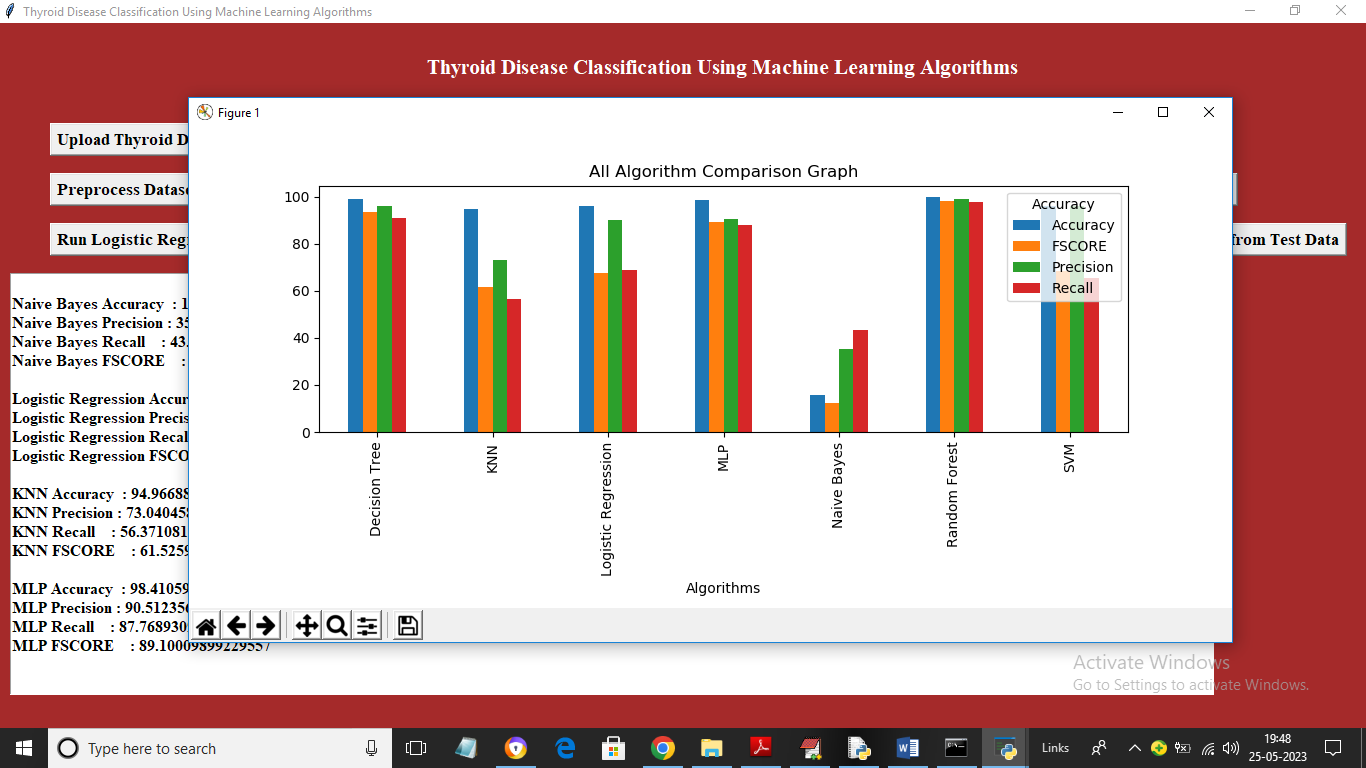
In above screen Logistic Regression got 96% accuracy



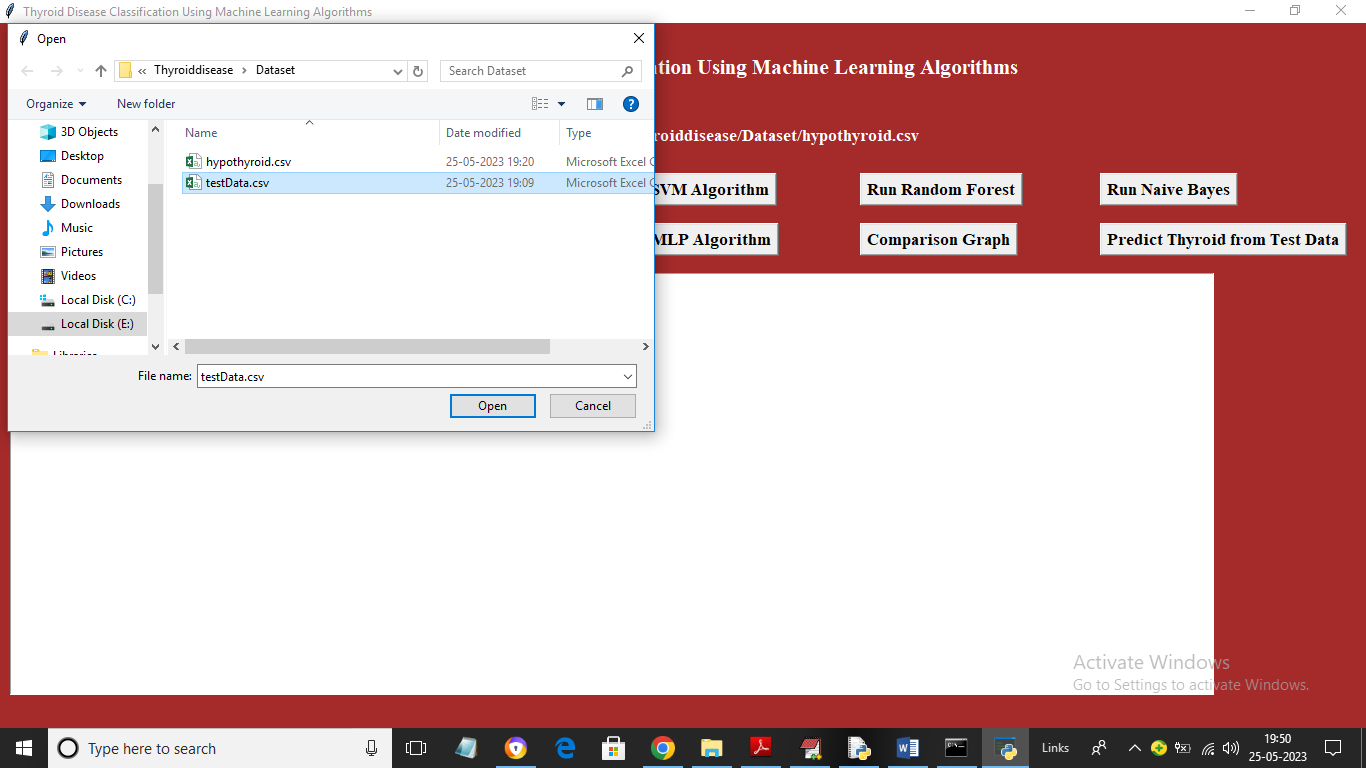
In above screen KNN got 94% accuracy



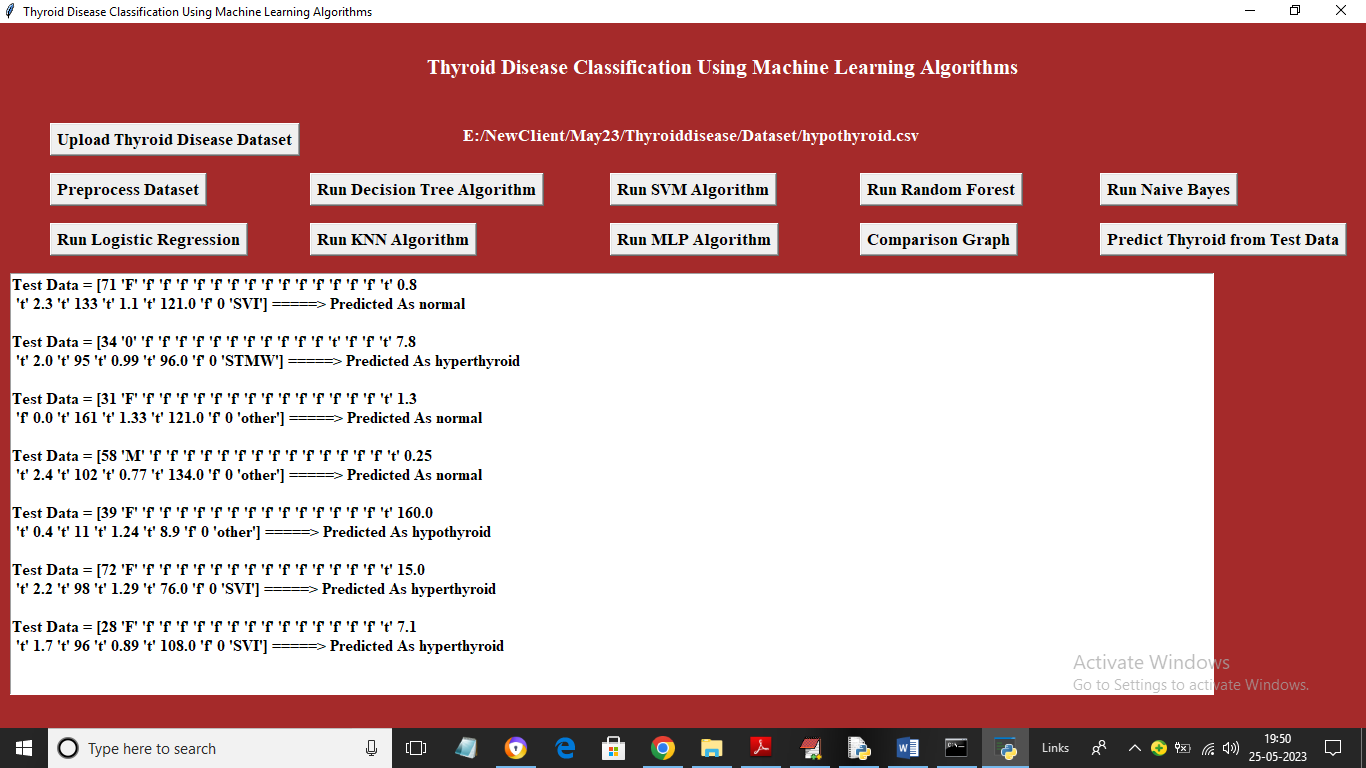
In above screen MLP got 98% accuracy and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different color bars and in all algorithms Random Forest and Decision Tree got highest accuracy and now close above graph and then click on ‘Predict Thyroid from Test Data’ button to upload test data and get prediction output



In above screen selecting and uploading ‘Test Data’ file and then click on ‘Open’ button to get below output



In above screen in square bracket we can see TEST data values and after =🡺 arrow symbol we can see predicted thyroid disease as Normal or Hypo or Hyper

**8.CONCLUSION:**

Thyroid disease is one of the diseases that afflict the world’s population, and the number of cases of this disease is increasing. Because of medical reports that show serious imbalances in thyroid diseases, our study deals with the classification of thyroid disease between hyperthyroidism and hypothyroidism. This disease was classified using algorithms. Machine learning showed us good results using several algorithms and was built in the form of two models. In the first model, all the characteristics consisting of 16 inputs and one output were taken, and the result of the accuracy of the random forest algorithm was 98.93, which is the highest accuracy among the other algorithms. In the second embodiment, the following characteristics were omitted based on a previous study. The removed attributes were 1- query thyroxine 2- query\_hypothyorid 3-query\_hyperthyroid. Here we have included the increased accuracy of some algorithms, as well as the retention of the accuracy of others. It was observed that the accuracy of Naive Bayes algorithm increased the accuracy by 90.67. The highest precision of the MLP algorithm was 96.4 accuracy.

**9.REFERENCES:**

[1] Azar, at, Hasaean, A.E. and Kim, T. Expert system based on neural fuzzy rules for thyroid diseases diagnosis, Computer Science, Artificial Intelligence, arXiv:1403.0522, Pp. 1-12,2012.

[2] Kales, A. ESTDD: Expert system for thyroid diseases diagnosis, Expert Syst Appl., Vol. 34, No.1, Pp.242–246,2008.

[3] a. Schuck, "World Health Organization," 2000. [Online]. Available: https://www.who.int/. 2nd International Conference on Physics and Applied Sciences (ICPAS 2021) Journal of Physics: Conference Series 1963 (2021) 012140 IOP Publishing doi:10.1088/1742-6596/1963/1/012140 11

[4] Koura, K., Exarches, T.P. Exarches, K.P., Karamus, M.V. and Fotiadis, D.I. (2015) Machine learning applications in cancer prognosis and prediction, Computational and Structural Biotechnology Journal, Vol. 13, Pp.8–17.

[5] Shukla, A. & Kaur, P. (2009). Diagnosis of thyroid disorders using artificial neural networks, IEEE International Advance computing Conference (IACC 2009)– Patiala, India, pp 1016-1020.

[6] Aswad, Salma Abdullah, and Eulah Sinus. "Classification of VPN Network Traffic Flow Using Time Related Features on Apache Spark." 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2020.

[7] Banu, G. Rasta. "A Role of decision Tree classification data Mining Technique in Diagnosing Thyroid disease." International Journal of Computer Sciences and Engineering 4.11 (2016): 64-70.

[8] Chindi, Jamil Ahmed, et al. "TDV: Intelligent system for thyroid disease visualization." 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube). IEEE, 2016.

[9] Travis B Murdoch and Allan S Ditsy. The inevitable application of big data to health care. Jama, 309(13):1351–1352, 2013.

[10] Dr. Srinivasan B, Pavia K “Diagnosis of Thyroid Disease: A Study” International Research Journal of Engineering and Technology Volume: 03 Issue:

11 | Nov – 2016 [11] Aytürk Kelsea and kales, Ali. "ESTDD: Expert system for thyroid diseases diagnosis." International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 11 | Nov -2017 34.1 (2017): 242- 246

[12] Khushboo Taneja, Parveen Sehgal, Preanal “Predictive Data Mining for Diagnosis of Thyroid Disease using Neural Network” International Journal of Research in Management, Science & Technology (E-ISSN: 2321- 3264) Vol. 3, No. 2, April 2016

[13] Chantel, Khushboo, et al. "A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques." CSI transactions on ICT 4.2-4 (2016): 313-319.

[14] Banu, G. Resita. "A Role of decision Tree classification data Mining Technique in Diagnosing Thyroid disease." International Journal of Computer Sciences and Engineering 4.11 (2016): 64-70.

[15] Umar Sidiqi, Dr, Syed Mutahar Saqib, and Rafi Ahmad Khan. "Diagnosis of various thyroid ailments using data mining classification techniques." Int J Sci Res Copout Sci Inf Technol 5 (2019): 131-6.

[16] Sandhya, Mrs. K. "EFFECTIVE PREDICTION OF HYPOTHYROID USING VARIOUS DATA MINING TECHNIQUES."