**DIABETES PREDICTION USING MACHINE**

**LEARNING**

***A Project report submitted in partial fulfillment of the requirements for the award of***

***the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**

**Submitted by**

M.G.S.S.Mohit 319106410073

D.Prajjwal 319106410070

Adja Kodjo Aménoagbé 319106410069

V.Lavanya 319106410095

Under the esteemed guidance of

**Dr. M Ramjee**

**Adjunct Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS** **ENGINEERING**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING**

**ANDHRA UNIVERSITY**

**VISAKHAPATNAM**

**APRIL 2023**



# CERTIFICATE

This is to certify that the project report entitled “**Diabetes Prediction Using Machine Learning** ” is a bonafide record of work carried out by **M.G.S.S.Mohit(319106410073), D.Prajjwal(319106410070),** **Adja Kodjo Aménoagbé(319106410069), V.Lavanya (319106410095)** students submitted in partial fulfillment of the requirement for the award of the degree of Bachelors of Technology in Computer Science Engineering during the period 2019-2023.

|  |  |
| --- | --- |
| **Project Guide** | **Head of the Department** |
| **Dr. M. Ramjee** | **Dr.K.Venkat Rao** |

**DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS**

**ENGINEERING**

**ANDHRA UNIVERSITY COLLEGE OF** **ENGINEERING(A)**

**ANDHRA UNIVERSITY**



**DECLARATION**

I/We, hereby declare that the project report entitled **“Diabetes Prediction Using** **Machine Learning”** has been prepared by us during the period December 2023 – April 2023 submitted in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering.

Place: Visakhapatnam

Date: 20/04/2023

|  |  |
| --- | --- |
| **Registration No(s).** | **Name(s)** |
|  |  |
|  |  |
| **319106410073** | **M.G.S.S.Mohit** |
|  |  |
| **319106410070** | **D.Prajjwal** |
|  |  |
| **319106410069** | **Adja Kodjo Aménoagbé** |
|  |  |
| **319106410095** | **V.Lavanya** |

**ACKNOWLEDGEMENT**

We have immense pleasure in expressing our earnest gratitude to our Project Guide **Dr. M. Ramjee**, Andhra University College of Engineering for his inspiring and scholarly guidance. Despite his preoccupation with several assignments, he has been kind enough to spare his valuable time and gave us the necessary counsel and guidance at every stage of planning and constitution of this work. We express sincere gratitude for having accorded us permission to take up this project work and for helping us graciously throughout the execution of this work.

We express sincere thanks to **Prof. K. Venkat Rao**, Head of the Department

Computer Science and Systems Engineering, Andhra University College of Engineering for his keen interest and providing necessary facilities for this project study.

We express sincere gratitude to **Prof. D. Lalitha Bhaskari**, Chairperson,

Board of Studies, Computer Science and Systems Engineering, Andhra University College of Engineering for her keen interest and for providing necessary facilities for this project study.

We express sincere thanks to **Prof. G. Sasi Bhushana Rao**, Principal, Electronics and Communications Engineering, Andhra University College of Engineering (A) for his keen interest and for providing necessary facilities for this project study.

We extend our sincere thanks to our academic teaching staff and nonteaching staff for their help throughout our study.

# ABSTRACT

Diabetes is one of the worst illnesses in the world. Diabetes is caused by a combination of variables, including obesity, excessive blood glucose levels, and other causes. It does this by altering the insulin hormone, which in turn causes an irregular metabolism in the crab and raises its blood sugar levels. This program's primary objective is to lessen the risk that people may acquire diabetes by making forecasts for them and urging them to take more care of their diet and lifestyle in the years to come.

The key goals of this research were to develop and execute a method for predicting diabetes using machine learning techniques, as well as investigate the strategies that would be used to achieve success in this endeavor. The suggested technique makes use of a wide variety of classification and ensemble learning algorithms, some examples of which include Knn, Label Encoder, and train test split. The results of the research may provide information that will help medical professionals make more accurate early predictions and judgments to better manage diabetes and save lives.

The method first extracts information from a dataset, such as certain symptoms that may be utilized to gain further knowledge about diabetes, and then validates that information using other data. This project's objective was to build classification models for the diabetes data set, develop models that can determine whether a person is sick, and get the greatest possible validation scores in the models that were developed. Massive datasets may be found in the healthcare business. By investigating enormous datasets in this manner, we may uncover previously unknown information and trends, which will enable us to draw conclusions based on the data and make accurate forecasts.

We categorize the dataset using random techniques since our major goal in doing this research is to determine the method that is the most accurate for predicting diabetes. The use of machine learning, which is becoming more important in the modern healthcare sector, will be the focus of this research.

# TABLE OF CONTENTS

* **Abstract**
* **List of Figures**
* **List of Tables**
* **Abbreviations/Notations**

## Chapter-1 Page Number

1. **INTRODUCTION 1**
   1. Introduction ………………………………..1
   2. Programming Language Used……………...2
   3. Problem Statement…………………………3
   4. Motivation…………………………………3
   5. Objectives of the Project…………………..4
   6. Existing System……………………………5
   7. Proposed System…………………………...5
   8. Aim of the Project………………………….6
   9. Scope of the Project………………………...6

**Chapter-2**

1. **LITERATURE SURVEY** **8**

## Chapter-3

1. **REQUIREMENT ANALYSIS** **12**
   1. Introduction to Requirements Analysis………….12
   2. Hardware Requirements…………………………14
   3. 3.3 Software Requirements………………………15

## Chapter-4

**4. SYSTEM ARCHITECTURE, DESIGN AND** **METHODOLOGY** **19**

* 1. System Architecture………………………19
  2. System Design……………………………20
     1. Design Implementation Methods.21
     2. Design Implementation Plan……21
     3. DFD…………………………….22

4.3 Algorithms………………………………..26

4.3.1 KNN Classifier………………….26

4.3.2 Decision Tree…………………...28

4.3.3 Random Forest………………….33

4.4 Methodology……………………………...36

4.4.1 Evaluation Metrics……………...37

## Chapter 5

1. **IMPLEMENTATION AND TESTING** **40**
   1. Implementation…………………………...40
      1. Importing the dependencies………...40
      2. Data collection and processing……...40
   2. Machine Learning Algorithms……………43
      1. KNN Classifier……………………...44
      2. Random Forest……………………...45
      3. Decision Tree……………………….46
   3. UI Design…………………………………46
      1. home.html..........................................46
      2. predict.html…………………………48
      3. views.py…………………………….50
      4. urls.py………………………………52
      5. Execution Command………………..53
   4. Testing……………………………………54
      1. Introduction…………………………54
      2. Types of tests………………………..55

## Chapter 6

1. **RESULTS, DISCUSSION AND COMPARISON** **58**
   1. Results……………………………………58

6.1.1 Screenshots of the output……………..59

* 1. Discussion………………………………...62
  2. Comparison……………………………….63

## Chapter 7

1. **CONCLUSION AND FUTURE ENHANCEMENT** **64**
   1. Conclusion………………………………..64
   2. Future Enhancements……………………..64

## Chapter 8

1. **REFERENCE** **66**
   1. Web Links………………………………...66
   2. References ………………………………..66

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Figure Description** | **Page No**. |
| 1. | System Architecture | 19 |
| 2. | Data Flow Diagram Level 1 | 22 |
| 3. | Data Flow Diagram Level 2 | 22 |
| 4. | Activity Diagram | 23 |
| 5. | Use Case Diagram | 24 |
| 6. | KNN Diagram | 25 |
| 7. | General Structure of Decision Tree | 27 |
| 8. | Working of Random Forest | 32 |
| 9. | Output screen 1 | 48 |
| 10. | Output screen 2 | 49 |
| 11. | Output screen 3 | 49 |
| 12. | Output screen 4 | 50 |
| 13. | Output result 1 | 50 |
| 14. | Output result 2 | 50 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Table Description** | **Page No.** |
| 1. | Accuracy comparison between algorithms | 47 |

## ABBREVIATIONS AND NOTATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| ML | Machine Learning |
| RF | Random Forest |
| SVM | Support Vector Machine |
| LR | Logistic Regression |
| CART | Classification And Regression Trees |
| TP | True Positive |
| TN | True Negative |
| FP | False Positive |
| FN | False Negative |
| EDA | Exploratory Data Analysis |
| ID3 | Iterative Dichotomiser 3 |
| NaN | Not a Number |
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| KNN | K-Nearest Neighbors |

## Chapter 1 INTRODUCTION

**1.1 Introduction**

The field of medicine has swiftly come to view the notion of machine learning as one that holds significant potential. The research community's ability to make accurate predictions and analyses based on medical data sets is an invaluable asset in the fight against disease and the development of effective preventative measures. Machine learning refers to the many types of algorithms that can assist in the process of decision making and prediction. We also talk about the many different uses of machine learning in the medical industry, with a particular emphasis on using machine learning to predict diabetes. Diabetes is one of the diseases that is rising at the fastest rate in the globe, and it has to be monitored constantly. In order to verify this, we investigate several different machine learning techniques that will assist in the accurate early prediction of this disease. This book provides an explanation of numerous aspects of machine learning, including the different sorts of algorithms that can assist with decision making and prediction. The predictions and analyses that have been performed by the research community for medical datasets benefit the public by allowing them to take the appropriate care and measures in order to prevent illnesses. Explore the many different uses of machine learning in the realm of medicine, concentrating specifically on how machine learning may be used to predict diabetes. Diabetes is one of the diseases that is increasing at the quickest rate throughout the world and has to be constantly monitored. In order to validate this, we are investigating a variety of machine learning methods that can assist us in making this baseline forecast. Decision Support Systems, Diabetes, Machine Learning, Support Vector Machine, Random Forest, K-Nearest Neighbor, and Logistics Regression are some of the keywords that might be associated with this topic. Intruder detection is discussed with regard to the application of machine learning techniques in this article. The analysis and discovery of patterns in data may be accomplished by machine learning algorithms through the application of artificial intelligence and data mining approaches. Diabetes is a condition that develops when blood glucose levels in the body, sometimes referred to as glucose, are abnormally high. Diabetes is referred to by the phrase "diabetes." According to the findings of experts, diabetes occurs when the pancreas is unable to generate an adequate amount of insulin. Insulin is a hormone that transports sugar from the circulatory system to the cells of the body, where it is converted into usable energy. Insulin is secreted by the pancreas. As a result, higher quantities of glucose are excreted in urine. If diabetes isn't treated properly, it can eventually cause organ failure, cardiovascular disease, and disturbances in a variety of other physiological systems. Diabetes is one of the four most significant noncommunicable diseases

(NCDs) impacting the globe today, as reported by the World Health Organization (WHO) (World Health Day, 2016). The most recent WHO findings are quite concerning. As was mentioned before, diabetes is a risk factor for a wide range of extra severe cardiovascular complications. Diabetes and cardiovascular illnesses are responsible for the deaths of 3.7 million people worldwide before they reach the age of 70, as reported by the World Health Organization (WHO). A blood glucose level that is not under control is the primary contributor to diabetes. Diabetes is one of the most serious diseases, and a large number of people are now afflicted with it. Diabetes can be caused by a variety of factors, including old age, obesity, abrupt weight loss, polyphagia, irritability, and muscular stiffness, among other things.

**1.2 Programming Languages Used:**

Python is a high-level, object-oriented programming language that is interpreted and has dynamic semantics. It's a great choice for Rapid Application Development, as well as for usage as a scripting or glue language to link pre existing components, thanks to its high-level built-in data structures, dynamic typing, and dynamic binding. The low cost of Python's upkeep is a direct result of the language's focus on readability. Python's module and package system facilitates and promotes program modularity and the reusability of code. The

Python programming language and its huge standard library are open-source, crossplatform, and free to use, modify, and redistribute in either source code or binary form. Python's greater productivity is a major draw for many programmers. The time it takes to get from editing a file to running a test and then fixing an issue is drastically reduced by the absence of the compilation phase. Unlike other programming languages, Python never generates a segmentation fault while debugging a program due to a bug or poor input. Instead, an exception is generated whenever the interpreter encounters a problem. The interpreter will provide a stack trace if the program does not handle the exception. Using a source-level debugger, you can do things like examine local and global variables, evaluate arbitrary expressions, place breakpoints, walk through the code line by line, and so on. The debugger is developed in Python, demonstrating the language's ability to look within. However, adding a few print statements to the source code is frequently the easiest method to debug a program; the short edit-test-debug cycle makes this basic technique quite successful.

**1.3 Problem Statement:**

Age, obesity, rapid weight loss, polyphagia, irritability, muscular stiffness, etc., are all risk factors for developing diabetes, making it one of the most pressing health problems today. This project aimed to create classification models for the diabetes data set, use those models to predict whether a person is sick, and achieve the greatest validation scores possible for those models.

**1.4 Motivation:**

Diabetes is considered to be one of the worst illnesses in the world. Diabetes is caused by a combination of variables, including obesity, excessive blood glucose levels, and other causes. It does this by altering the insulin hormone, which in turn causes an irregular metabolism in the crab and raises its blood sugar levels. This program's primary objective is to lessen the risk that people may acquire diabetes by making forecasts for them and urging them to take more care of their diet and lifestyle in the years to come. The key goals of this research were to develop and execute a method for predicting diabetes using machine learning techniques, as well as investigate the strategies that would be used to achieve success in this endeavor.

**1.5 Objectives of the Project:**

We categorize the dataset using random methods in order to determine the accurate algorithm for diabetes prediction, which is the primary goal of this research. Other objectives include employing machine learning, data visualization, and data interpretation. The application of machine learning, which is becoming increasingly significant in today's medical field, will be the focus of this particular research. Massive amounts of data are stored in the industry's databases. In this way, we are able to explore big datasets and uncover previously unknown information as well as trends. This allows us to derive knowledge from the data and make accurate predictions about future occurrences. The primary objective of this project is to decrease the risk that individuals may develop diabetes by the implementation of forecasts and the encouragement of individuals to be more careful in the future. Since the turn of the previous decade, there has been a considerable uptick in the number of persons who are afflicted with diabetes. The way that people live their lives nowadays is the key factor contributing to the rising prevalence of diabetes. In contemporary practices of medical diagnosis, errors might fall into one of three categories. Those categories are as follows: The false-negative kind is one in which a patient actually does have diabetes yet the test results indicate that the person does not have diabetes. The kind that gives a false positive. In this instance, the patient does not in fact have diabetes, despite the fact that the test results suggest that he or she does. The third category is the unclassifiable type, which describes situations in which a specific case cannot be diagnosed by a certain system. It is possible that a particular patient will be forecasted as belonging to an unclassified category as a result of inadequate information extraction from historical data. In practice, however, the patient needs to make a prediction as to whether or not they fall into the diabetic or non-diabetic categories. These kinds of diagnostic mistakes might result in needless therapies or even the absence of therapy altogether when it is warranted. In order to avoid or lessen the severity of an impact of this kind, there is a pressing need to develop a system that makes use of an algorithm for machine learning and various data mining techniques. This system should be able to produce accurate results while simultaneously cutting down on the amount of work done by humans.

**1.6 Existing System:**

The SVM Algorithm is used in the current method to calculate the nuances of diabetes. In order to organize any data set, the SVM Algorithm is a simple yet effective tool. Poor time management is the biggest problem. Large scale, heterogeneous, autonomous sources, transmitted, decentralized control, and the need to investigate complex, ever-evolving interconnections are only the beginning of the ways in which the patient's diabetes sets it apart.

* **Disadvantages of Existing System:**

Concerning accuracy, an automated framework does not, on its own, ensure precision, and the data from the distribution center falls into the same category as the data from the information flow that originated it. The framework does not fully automate its operations; in order to accomplish its purpose, it requires input from the customer.

**1.7 Proposed System:**

Due to developments in technology and medical science, the social insurance industry now interacts with medical records digitally rather than on paper. The information on the board may have been simplified, but keeping up with the constant updates is still difficult. Knowledge creation, on the other hand, is not only inevitable, but also directly proportional to the march of time. These days, medical information such as patients' documented wellness data, rehabilitative records, diagnostic reports, and prescription-related records are all kept track of by large-scale information management systems like Electronic Health Records (EHR). The Random Forest method excels in High dimensionality, Training Speed, and handling imbalanced Data.

* **Advantages of Proposed System:**
  + - Impressive in Versatility – handles binary features, numerical features
    - Parallelizable – we can split the process to multiple machines to run.
    - Great with high dimensionality – since we are working with split

data.

* 1. **Aim of Project:**

The primary objective of this project was to effectively achieve the goal of successfully designing and implementing Diabetes Prediction Using Machine Learning Approaches and then performing Performance Analysis of those methods. The suggested technique employs many classification and ensemble learning methods, some of which include Knn, Label Encoder, and train test split. The use of machine learning, which is becoming increasingly significant in today's medical field, is going to be the focus of this research. Massive amounts of data are stored in the industry's databases. In this way, we are able to explore big datasets and uncover previously unknown information as well as trends. This allows us to derive knowledge from the data and make accurate predictions about future occurrences. The findings of the experiment can provide assistance to medical professionals in making early predictions and decisions in order to treat diabetes and save the lives of individuals.

* 1. **Scope of the Project:**

We proposed a diabetes prediction model for the purpose of improving the classification of diabetes. This model combines a few of the external characteristics that are responsible for diabetes with regular factors such as delayed healing, partial paresis, irritability, itching, and visual blurring, among other symptoms. If diabetes can be predicted, then people will be able to take better care of themselves. Diabetes results from the body's inability to produce an adequate amount of insulin. Diabetes affects around 422 million people worldwide, the majority of whom live in countries with a low or moderate income, and the World Health Organization estimates that by the year 2030, this number might reach 490 billion. As a consequence of this endeavor, it's possible that many lives will be preserved.

## Chapter 2 LITERATURE SURVEY

**[1] Name of Paper:** An Innovative Approach for Diabetes Prediction.

**Author and Publication year:** Krishna Priya A S, Shemitha P A, Dr G.

Kiruthiga, Vol-8 Issue-3 2022. IJARIIE-ISSN(O)-2395-4396.

**Abstract:**

Algorithm Used: Naive Bayes Classification

**Drawbacks**:

Since it is possible for its predictions to be inaccurate in some circumstances, you shouldn't place too much stock in the probability outputs that it provides. It presupposes that each of the characteristics may exist on its own. When we come across terms in the test data for a certain class that were not included in the training data, we run the risk of having zero class probabilities for that class. The fundamental objective of this research is to investigate a database containing information on diabetic patients in order to make accurate diagnoses of diabetes at an earlier stage. Within the framework of this experiment suggestion, the Naive Bayes Classification points to diabetes. The process of collecting data from a dataset and organizing it into a format that may be used in further analysis is known as "information mining. "The findings indicate that the new strategy that was presented has a greater potential (0.96) for accurately predicting diabetes than the conventional methods that are currently in use. This system that is suggested and uses the Naive Bayes Classifier will provide a Web Interface as its output. This Web Interface will present the outcome of whether the individual has diabetes or does not have diabetes depending on the input factors such as insulin level, age, and so on. This results in an increase in the system's precision. The Bayes theorem serves as the foundation for the Naive Bayes method, which is a form of supervised learning that is used to solve classification problems. Its primary use is in text categorization jobs, which often demand a sizable amount of data for training. **[2]Name of Paper:** Diabetes Prediction using Machine Learning Algorithms.

**Author and Publication year:** Aishwarya Mujumdara, Dr. Vaidehi V,

VIT Chennai India, Mother Teresa Women’s University, Kodaikanal, India.

INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019 ICRTAC 2019.

**Abstract:**

Algorithm Used: Big Data Analytics

1. Split, Train and Test
2. ML using Pipeline

**Drawbacks:**

Big Data Analytics looks for patterns in the data that already exists; hence, the level of accuracy may vary depending on the data. Diabetes puts a person at high risk for a number of other ailments as well, including cardiovascular disease, renal disease, stroke, eye problems, nerve damage, and so on. The standard procedure followed in hospitals nowadays is to collect the necessary data for diabetes diagnosis by a battery of different tests, after which the correct therapy is administered depending on the diagnosis. The fields of medicine and healthcare find widespread use for big data analytics. The healthcare industry often maintains massive database volumes. With the use of big data analytics, one is able to investigate very large datasets, uncover previously unknown information and patterns within the data, and draw conclusions and make predictions based on those findings.

The categorization and prediction accuracy of the approach that is currently being used is not very good. Along with traditional risk factors like glucose, body mass index (BMI), age, and insulin, the model that we have proposed in this article for better diabetes classification is a diabetes prediction model. This model takes into account a few additional risk factors for diabetes in addition to the standard risk factors. In addition to this, a pipeline model was imposed for diabetes prediction with the goal of boosting the accuracy of categorization.

**[3]Name of Paper:** Diabetes Prediction using Machine Learning Techniques.

**Author and Publication year:** Mitushi Soni, Dr. Sunita Varma Dept of Computer Science and Engineering, Shri G.S. Institute of Technology and Science, Indore, India. International Journal of Engineering Research & Technology (IJERT), Vol.

9 Issue 09, September-2020.

**Abstract:**

Algorithm used: Random Forest, Support Vector Machine.

**Drawbacks:**

The SVM will not perform up to expectations in situations in which the number of features for each data point is greater than the number of training data samples.

**[4]Name of Paper:** Diabetes Prediction Using Machine Learning Techniques.

**Author and Publication year:** Tejas N. Joshi\*, Prof. Pramila M. Chawan\*\* \*M. Tech. student (Department of Computer Engg. and Info. Tech., V.J.T.I., Mumbai,

Maharashtra, India. \*\*Associate Professor (Department of Computer Engg. and Info. Tech., V.J.T.I., Mumbai, Maharashtra, India. Corresponding Author: Tejas N. JoshiS. Dewangan.et.al. Int. Journal of Engineering Research and Application ISSN: 2248-9622, Vol. 8, Issue 1, (Part -II) January 2018, pp.-09-13.

**Abstract:**

Algorithms used: ANN, SVM, Logistic Regression. In many different kinds of real-world problems, classification is one of the most significant methods for arriving at decisions. The primary goal of this effort is to increase the accuracy of the classification process by determining if the data represent diabetes or non-diabetic individuals and then classifying the data accordingly. In many classification problems, selecting a greater number of samples does not always lead to improved accuracy in the resulting classification. In many instances, the performance of an algorithm is great in terms of speed, but the accuracy of data categorization is low. [Case in point:] [Case in point:] Acquiring a high level of precision should be the primary focus of our model. The accuracy of the classification can be improved if we utilize a large portion of the data set for training and only a small portion of the data set for testing. The purpose of this survey was to investigate the efficacy of various categorization strategies for separating diabetes from non-diabetic data. As a result, it has been determined that methods such as the Support Vector Machine, Logistic Regression, and Artificial Neural Network are the ones that are best suited for putting the diabetes prediction system into action.

## 

## 

## Chapter 3 REQUIREMENTS ANALYSIS

**3.1 Introduction to Requirements Analysis**

Requirements analysis involves defining, analyzing, validating, and aligning stakeholders' expectations for new projects while considering all possible conflicts. It’s a process of identifying, analyzing, and managing project requirements to determine what the project should accomplish and eliminate any ambiguities or conflicting requirements in the project plan. As you conduct the requirements analysis process, remember that any accepted requirements must be:

* Documented
* Actionable
* Measurable
* Testable
* Traceable
* Defined with sufficient details
* Related to overall business needs

While requirements analysis is beneficial to any project, it is most common in software engineering. In software engineering, requirements analysis, known as requirement engineering, defines expectations for new software being built or modified.

Requirement Analysis which is also known as Requirement Engineering, is the process of defining the user expectations for a new software which is in the process of being built or modified. Requirements analysis encompasses those tasks that go into determining the needs or the conditions to meet for a new or an altered project, taking in account the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing system or software requirements.

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks which are conducted to identify the needs of the different users. Therefore, the requirements analysis means it is to analyze, make documents, validate and manage the software or the system requirements. High-quality requirements are actionable, documented, traceable, testable, measurable, helps to identify the business opportunities, and are defined to facilitate the system design.

A software requirement is the capability needed by the user to solve any problem or to achieve an objective. In other words, the requirement is the software capability that must be met or possessed by a system or the system component to satisfy a specification, standard, contract, or other formally imposed documentation. All members in a project such as developers, end users, software managers, customer managers must achieve some common understanding of what the product will be like and what we do, or someone will be surprised when it is delivered. Surprises in software are not always good news.

The main aim of requirement analysis is to fully understand the main objective of the requirement that includes why it is necessary. All these points are to be fully recognized in the problem recognition so that the requirements which are essential can be fulfilled to solve the business problems.

Evaluation means judging about something whether it is worth or not and the synthesis means to create or form something. Here are some of the tasks that are given that is important in the evaluation and the synthesis of the software requirement:

* To define all the functions of the software that are necessary.
* To define all the data objects that are present externally and are easily observable.
* To evaluate whether the flow of data is worth it or not.
* To fully understand the overall behavior of the system that means overall working of the system.
* To identify and discover the constraints that are meant to be designed.
* To define and establish the character of the system interface to fully understand how the system interacts with two or more components or with one another.

After completing the gathering of information from the above tasks, the functional and the behavioral models are established after checking the function and the behavior of the system using a domain model that is also known as the conceptual model. The software requirement specification (SRS) which means to specify the requirement whether it is functional or nonfunctional should be developed.

**3.2 Hardware Requirements**

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

* **Hardware requirements:**
  + 1. Processor: Intel Core i5/ i7
    2. RAM: 4GB
    3. Storage: 20GB
    4. Monitor with minimum 1024\*720 resolution

**Processor:**

An Intel Core i5/i7 is an Intel proprietary processor that is built on the framework of multiprocessor architecture. It is a type of dual-core processor with an integrated graphic processing unit (GPU).

**RAM:**

With 4 GB of RAM, you will have enough memory to run several programs at once. You can open lots of browser tabs at once, use photo or video editing programs, stream content, and play mid-to-high-end games.

**Hard disk:**

The storage required to run, train and execute the project. Hard disks for personal computers can store terabytes of information.

**Display Monitors:** Display monitor must be no less than 15” for spot view or sequence view. The high performance, long-term reliability, and flexible connectivity make a security monitor better meet the diverse requirements of any video surveillance environment.

**3.3 Software Requirements:**

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

**Software Requirements for Present Project:**

1. Operating System: Windows 7/8/10
2. Python
3. Anaconda
4. Jupyter Notebook
5. Django

**Python:**

Python is a high-level, object-oriented programming language that is interpreted and has dynamic semantics. It's a great choice for Rapid Application Development, as well as for usage as a scripting or glue language to link pre existing components, thanks to its high-level built-in data structures, dynamic typing, and dynamic binding. The low cost of Python's upkeep is a direct result of the language's focus on readability. Python's module and package system facilitates and promotes program modularity and the reusability of code. The Python programming language and its huge standard library are open-source, cross-platform, and free to use, modify, and redistribute in either source code or binary form.

Python's greater productivity is a major draw for many programmers. The time it takes to get from editing a file to running a test and then fixing an issue is drastically reduced by the absence of the compilation phase. Unlike other programming languages, Python never generates a segmentation fault while debugging a program due to a bug or poor input. Instead, an exception is generated whenever the interpreter encounters a problem. The interpreter will provide a stack trace if the program does not handle the exception. Using a source-level debugger, you can do things like examine local and global variables, evaluate arbitrary expressions, place breakpoints, walk through the code line by line, and so on. The debugger is developed in Python, demonstrating the language's ability to look within. However, adding a few print statements to the source code is frequently the easiest method to debug a program; the short edit-test-debug cycle makes this basic technique quite successful.

**Anaconda:**

It is a free and open-source Python and R distribution with the goal of simplifying package management and deployment for use in data science and machine learning-related applications (such as big data processing, predictive analytics, and scientific computing). The conda package management system handles all the different versions of the packages. There are nearly 6 million people (about twice the population of Arkansas) that use the Anaconda distribution, which includes over 250 widely used tools for data science and runs on Windows, Linux, and MacOS.

**NumPy:**

NumPy is a general-purpose array-processing package. It provides a higher performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Here NumPy is used to store the accessed dataset data in the arrays. It will be used to check if the user given symptoms is there in the array or not. Because NumPy is very fast to change the accessed data to an array within a low amount of time. Here we have used NumPy arrays to store Training dataset data and Testing dataset data.

**Pandas:**

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Here Pandas are used to access the datasets (CSV Files) of the Diseases and its Symptoms. Here Pandas will access two datasets. They are Training Dataset and Testing Dataset. Here Pandas are used to access the data in a low amount of time. Here accessing data is fast as compared to remaining modules. Because Pandas are able to access the data and store it in a tabular form.

**Scikit-learn:**

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including supportvector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

**Django:**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

## Chapter 4 SYSTEM ARCHITECTURE, DESIGN AND METHODOLOGY

**4.1 System Architecture:**

**Data Collection and Preprocessing:** The first step is to gather relevant data and preprocess it to remove any missing or incorrect values. The dataset should contain features such as age, BMI, glucose level, blood pressure, etc. The data preprocessing may include normalization, scaling, and feature selection.

**Model Selection:** Once the data is preprocessed, you need to choose an appropriate machine learning algorithm for the task. Several models can be used, such as logistic regression, decision trees, random forest, and support vector machines (SVM). You can evaluate different models using metrics such as accuracy, precision, recall, and F1 score.

**Training and Validation:** After selecting the model, you can train it using the preprocessed data. You need to split the data into training and validation sets to avoid overfitting. Cross-validation can also be used to evaluate the model's performance.

**Hyperparameter Tuning:** The performance of the model can be further improved by tuning the hyperparameters. You can use techniques such as grid search, random search, or Bayesian optimization to find the optimal hyperparameters.

**Deployment:** Once the model is trained and tuned, you need to deploy it in a production environment. This can be done using various tools and technologies, such as REST APIs, cloud services, or containerization. The deployment strategy should ensure scalability, availability, and security.

**User Interface:** You can design a user interface that allows users to enter their medical data, and the model can provide the predicted diabetes risk score. The user interface can be a web application, mobile application, or desktop application.

**Monitoring and Maintenance:** You need to monitor the model's performance in production and perform regular maintenance to ensure it continues to provide accurate predictions. You may also need to retrain the model periodically to keep up with changes in the data distribution.



Fig 1.System Architecture

**4.2 SYSTEM DESIGN:**

The transition from the old to needs new system design is an integral part of the implementation process, which encompasses all the actions involved in this transition. The proposed new system is operated in a completely different manner compared to the present system, which is composed of manual activities and is run in a totally different manner. To deliver a dependable system that can fulfill the criteria of the companies, it is necessary to carry out the implementation in the correct manner. The effectiveness of the computerized system may be jeopardized by an installation that is not performed correctly.

**4.2.1 DESIGN IMPLEMENTATION METHODS:**

There are a few different approaches that may be used to manage the transition from the older computerized system to the new one, as well as the implementation that follows. Operating both the old system and the new system concurrently is the strategy that offers the highest level of protection throughout the transition from the old to the new system. Under this strategy, a person can continue to operate in the manual older processing system while also beginning to run the new digital system.

This approach provides a high level of security due to the fact that we are able to rely on the manual system even in the event that there is a defect in the computerized system. However, the expense of keeping two systems running in parallel at the same time is rather considerable. This causes its advantages to be outweighed. A direct cut over from the manual system that was previously in place to the computerized system is another way that is regularly used.

The shift might take place within a week or it could take place today. There are no activities that run in parallel. However, there is no solution in the event that there is a problem. The execution of this technique demands meticulous preparation. It is also possible to establish a functional version of the system in a single section of the company. The employees in that section will serve as system pilots, and modifications to the system will be made as and when they are necessary. However, because the entire system is destroyed in this approach, it is not the technique of choice.

**4.2.2 DESIGN IMPLEMENTATION PLAN:**

The plan for putting the new system into operation and putting it into operation comprises a description of all the activities that need to occur in order to put the new system into operation. It also creates a time plan for the implementation of the system and identifies the persons who are accountable for the activities. The following are the steps that make up the overall implementation strategy. ● List all files required for implementation.

* Identify all data required to build new files during the implementation.
* List all new documents and procedures that go into the new system.

The Label Encoder implementation needs to be able to anticipate potential issues and ought to be able to solve such issues. The typical issues may include missing documents; data formats that are confused between the current and the files; faults in the translation of data; missing data; and so on.

**4.2.3 DFD:**

A data flow diagram, often known as a data flowchart or DFD, is a graphical depiction of the movement of data through an information system. The visualization of data processing may also be accomplished with the help of a data flow diagram (structured design). It is standard procedure for a designer to begin by sketching a DFD at the context level, which depicts the interaction between the system and things from the outside world. Following this, the context-level DFD is exploded to provide additional information on the system that is being modeled.

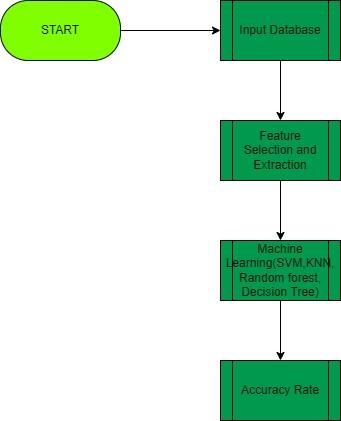


Fig 2. DFD Level 1

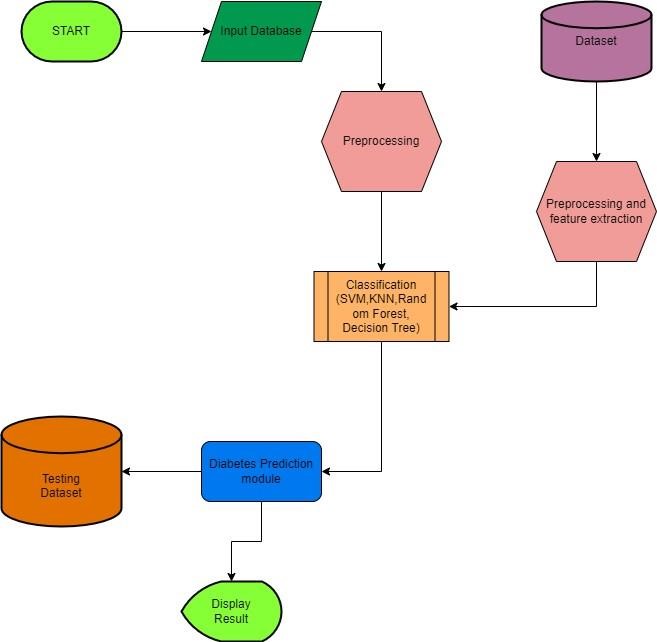


Fig 3.DFD Level 2

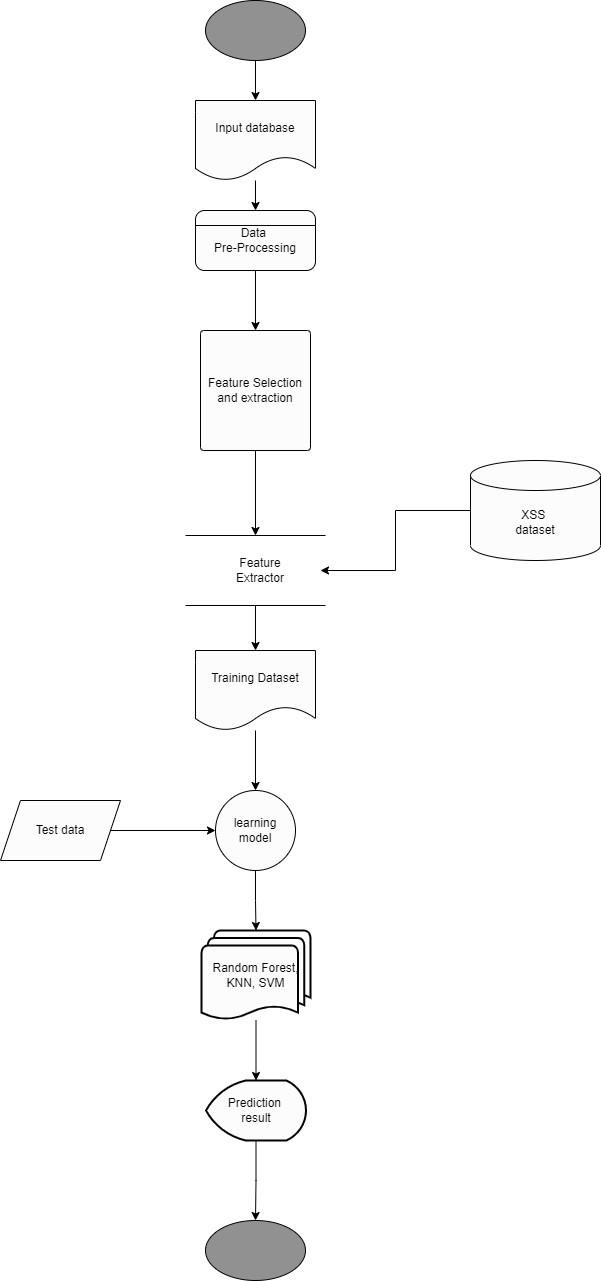


Fig 4. Activity Diagram

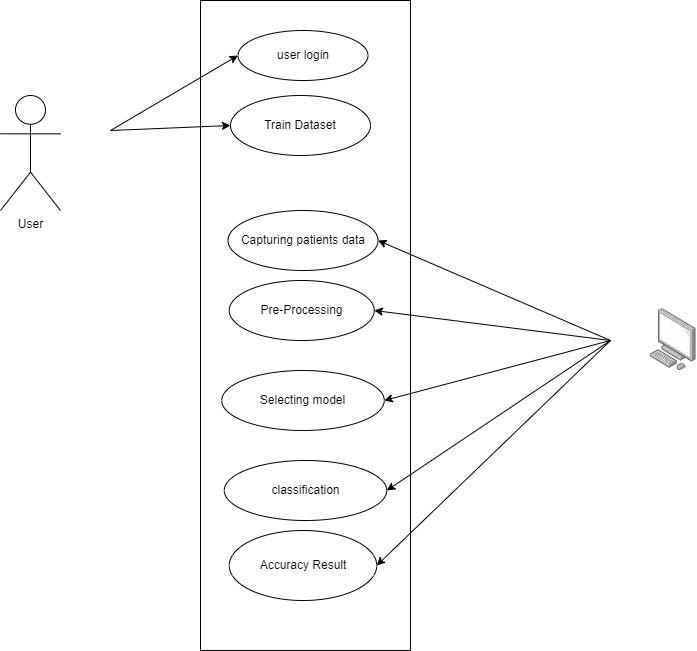


Fig 5.Use Case Diagram

**4.3 ALGORITHMS:**

**4.3.1 KNN Classifier**

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification of predictive problems in industry. The following two properties would define KNN well −

* **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
* **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

**Working of KNN Algorithm**

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

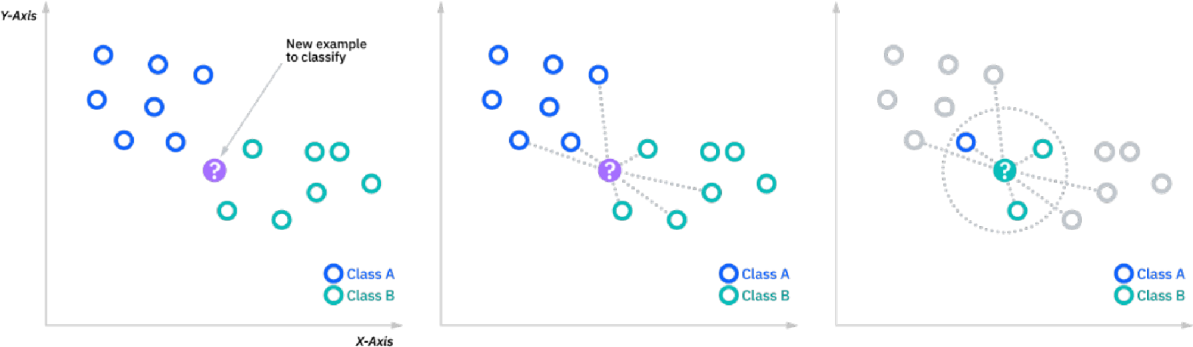


Fig 6. KNN Diagram

→ Step 1 − For implementing any algorithm, we need a dataset. So during the first step of KNN, we must load the training as well as test data.

→ Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

→ Step 3 − For each point in the test data do the following − o 3.1 − Calculate the distance between test data and each row of training data with the help of any of the methods, namely:

Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean. o 3.2 − Now, based on the distance value, sort them in ascending

order. o 3.3 − Next, it will choose the top K rows from the

sorted array.

* 3.4 − Now, it will assign a class to the test point based on the most frequent class of these rows.

→ Step 4 − End

**Advantages of KNN Algorithm:**

* It is simple to implement.
* It is robust to the noisy training data o It can be more effective if the training data is large.

**Disadvantages of KNN Algorithm:**

* Always needs to determine the value of K which may be complex sometimes.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

**4.3.2 Decision Tree Classification Algorithm**

* Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
* In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
* A decision tree simply asks a question and based on the answer (Yes/No), it further splits the tree into subtrees.
* A decision tree can contain categorical data (YES/NO) as well as numeric data.
* Below diagram explains the general structure of a decision tree:

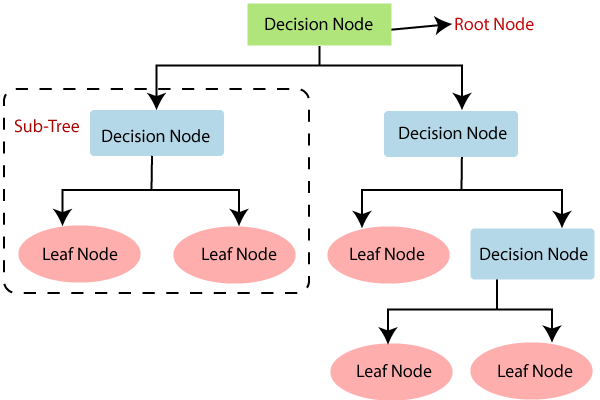


Fig 7. General structure of decision tree

**Decision Tree Terminologies**

* Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
* Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
* Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
* Branch/Subtree: A tree formed by splitting the tree.
* Pruning: Pruning is the process of removing the unwanted branches from the tree.
* Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. **Information Gain:**

When we use a node in a Decision Tree to partition the training instances into smaller subsets, the entropy changes. Information gain is a measure of this change in entropy.

*Information Gain= Entropy(S)- [(Weighted Avg) \*Entropy(each feature)*

Entropy is the measure of uncertainty of

a random variable, it characterizes the impurity of an arbitrary collection of examples.

The higher the entropy the more the information content.

*Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)* Where,

* + S= Total number of samples
  + P(yes)= probability of yes
  + P(no)= probability of no

2. **Gini Index:**

Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with a lower Gini index should be preferred. Sklearn supports “Gini” criteria for Gini Index and by default, it takes “gini” value.

The most notable types of Decision Tree algorithms are:-

1. **IDichotomiser 3 (ID3):**

This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

**1.2 C4.5:**

This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

2. **Classification and Regression Tree (CART):**

It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

**Working:**

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

* Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
* Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
* Step-3: Divide the S into subsets that contain possible values for the best attributes.
* Step-4: Generate the Decision Tree node, which contains the best attribute.
* Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.
* **Advantages of the Decision Tree**
* It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* **Disadvantages of the Decision Tree**
* The decision tree contains lots of layers, which makes it complex.

**Python Implementation of Decision Tree** Steps to implement:

* Data Preprocessing step
* Fitting a Decision-Tree algorithm to the Training set
* Predicting the test result
* Test accuracy of the result (Creation of Confusion matrix)
* Visualizing the test set result

**4.3.3 Random Forest Algorithm**

Random Forest is a supervised learning algorithm. It is an extension

of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees is the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)), where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees there are, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

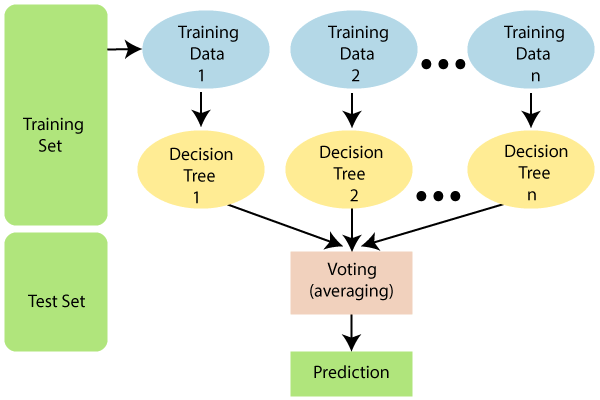


Fig 8. Working of Random Forest Algorithm

**Assumptions for Random Forest**

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

**Random Forest algorithm Steps:**

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes

**Advantages:**

* Random Forest is capable of performing bothClassification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

**Disadvantages:**

Although Random Forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

**4.4 SYSTEM METHODOLOGY:**

In terms of supervised classifiers, K-nearest Neighbors is your best bet.

When faced with a k-NN classification problem, it is the optimal solution. In order to predict the label of a new data point, KNN uses the distance between the labels of similar data points in the training set and the new data point. In most cases, the K variable in KNN is set between 0 and 10. The K-NN method uses an assumption of similarity between the new case/data and past cases to place the new case in the category most similar to the existing categories. The K-Nearest Neighbor method stores all previously collected information and uses it to assign categories to newly collected data. The optimal supervised classifier for K-NN is K-nearest Neighbors.

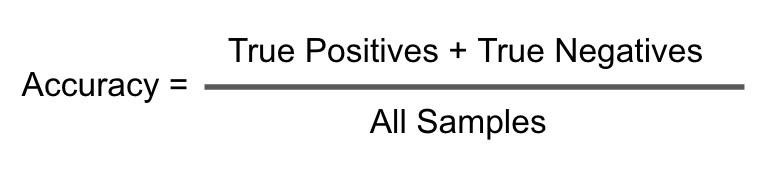
When faced with a k-NN classification problem, it is the optimal solution. KNN calculates the distance between a new test data point and the nearest class labels in the training data, using the given K value as a predictor. In most cases, the K variable in KNN is set between 0 and 10. The K-NN method uses an assumption of similarity between the new case/data and past cases to place the new case in the category most similar to the existing categories. The K-Nearest Neighbors method stores all available information and uses similarity to assign classes to newly collected data. Random Forest is "a classifier that comprises a number of decision trees on various subsets of the given dataset and takes the average to increase the predicted accuracy of that dataset," as the authors put it. Instead of depending on just one set of decision trees, a random forest takes the predictions made by each tree and makes an overall prediction based on the majority's choice. More trees in the forest mean better accuracy and less chance of overfitting. Supervised Learning Algorithm, or SVM (Support Vector Machine). The strategy for classifying data and looking for anomalies. SVM will be used to categorize the data according to the hyperplane. The hyperplane is used to effectively divide the two categories, and the best separator is often the one that lies at the hyperplane's most outer border. SVM Classifiers of both the Linear and Non-Linear varieties have been used. In the realm of Supervised Learning, Support Vector Machine (SVM) is widely used for both classification and regression tasks. On the other hand, it is widely employed in Machine Learning to address issues of categorization. Taking a look at the "decision tree" Supervised learning techniques like the decision tree are useful for both classification and regression issues. As a classifier, it takes the form of a tree with the nodes at its trunk storing information about the dataset, the branches representing the rules by which that dataset was classified, and the nodes at its leaves reflecting the final classification. It is a diagram that shows every feasible option for solving a problem or making a choice under conditions. Like a real tree, a decision tree has a central node from which branches extend in many directions.

**4.4.1 Evaluation Metrics:**

The factor is that you want to have deep expertise in the scoring metrics to decide how properly your version is performing.

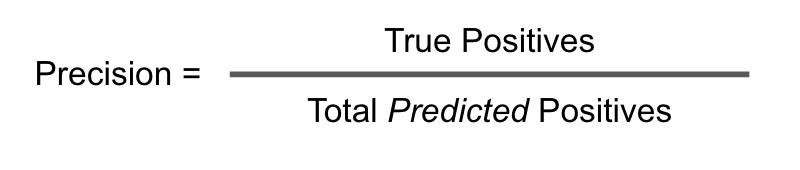
**Accuracy: Good** old accuracy is how properly our version predicts the

proper class or labels. If our dataset is reasonably balanced and all classes are similarly important, this ought to be our baseline metric to measure our version's performance.



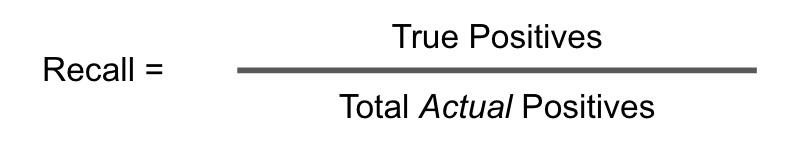
However, if our dataset is imbalanced, inclusive of a credit card fraud detection dataset, we might be 95 accurate. To confirm that our version tries to categorize check records factors into each class in preference to assigning a majority elegance, we want to seek advice from the confusion matrix.

**Precision:** In easy terms, Precision is the ratio of what our version expected efficiently to what our version expected. For every category/class, there may be one precision value.



We focus on accuracy when our predictions need to be correct, i.e., H. Ideally, we want to make sure that our model is correct when it predicts a label. For example, if we have a lending model that predicts whether a loan application will be approved or rejected, our priority is to get it right in all the cases where our model predicts the loan will be approved, since we will lose money if it is approved and becomes a loan that ideally should be declined. We don't lose money if you tell us to decline the loan because we still have that money with us. We use accuracy because the cost of making a wrong prediction is much higher than the cost of missing the right one.

**Recall:** In easy terms, Recall is the ratio of what our version expected efficiently to what the real labels are. Similar to precision, for every category/class, there may be one keep in mind value.



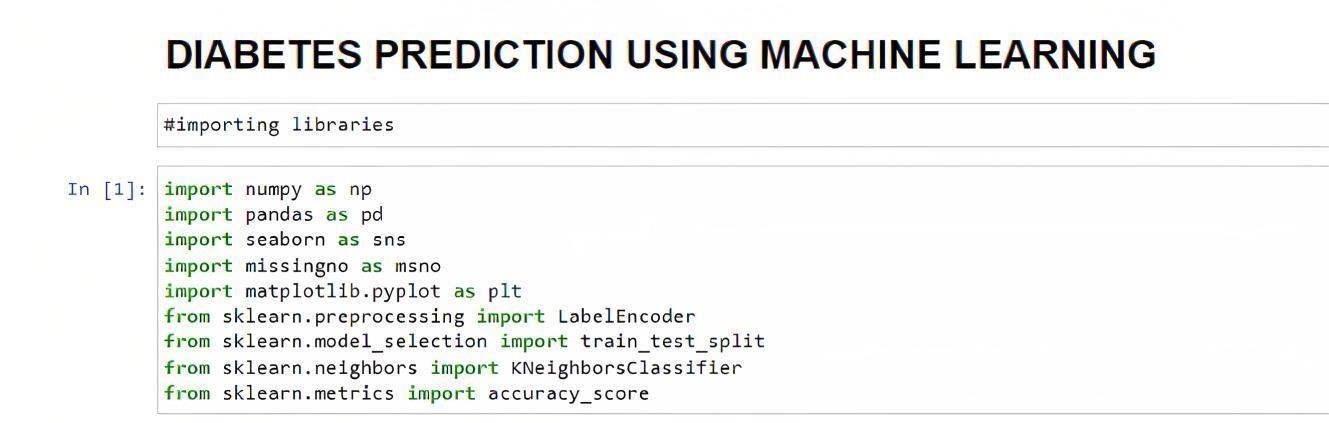
We focus on remembering when we are in a FOMO (fear of missing out) situation. Ideally, the model should capture all instances of a given class. For example, security scanners at airports have to make sure the detectors don't miss real bombs or dangerous objects, and that's why we can sometimes stop the wrong suitcase or traveler.

## 

## Chapter 5 IMPLEMENTATION AND TESTING

**5.1 Implementation**

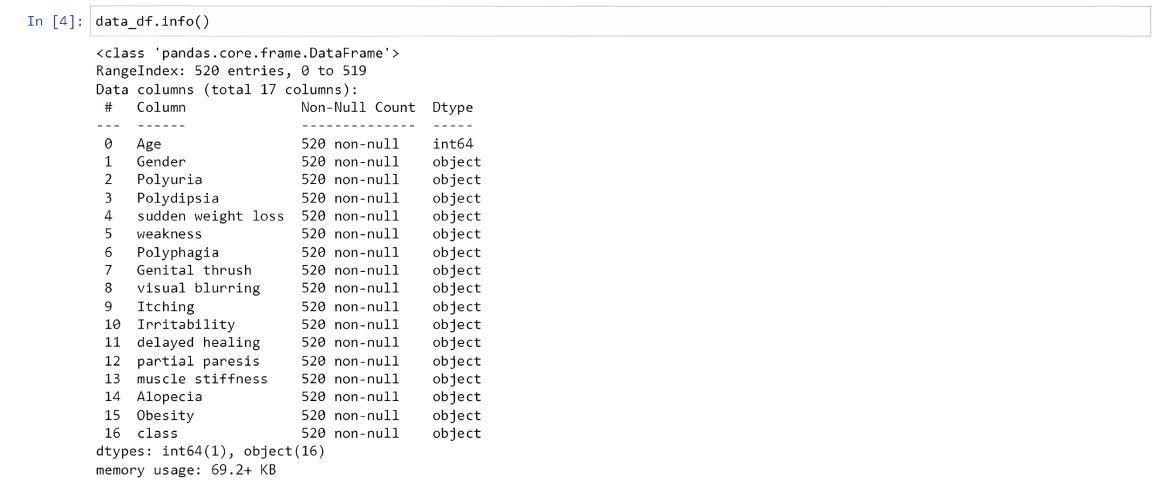
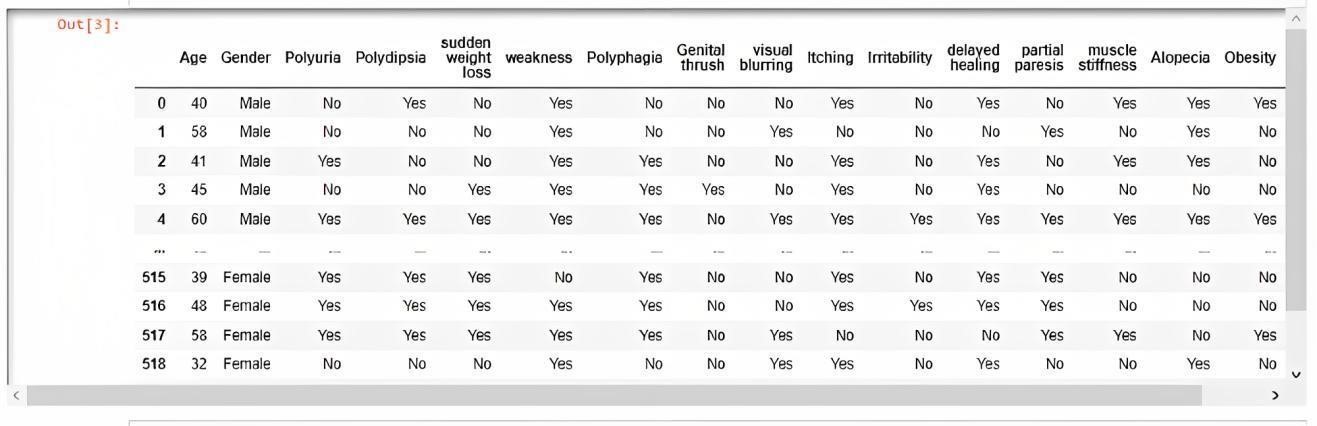
**5.1.1 Importing the dependencies:**

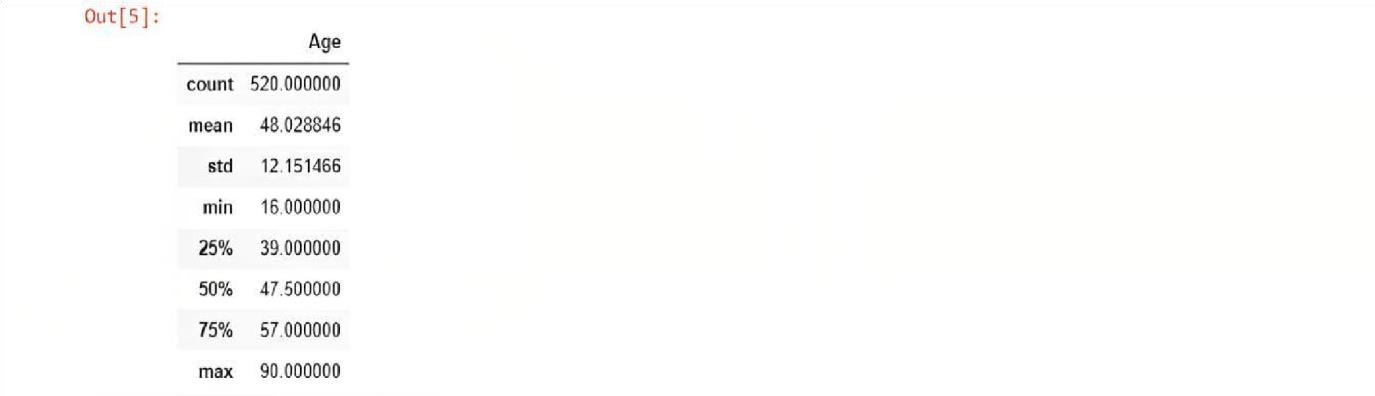


**5.1.2 Data collection and processing:**

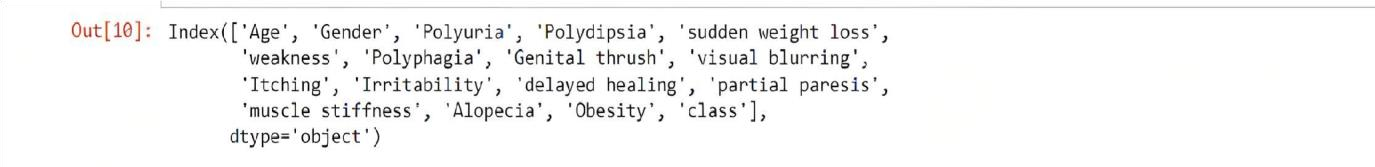


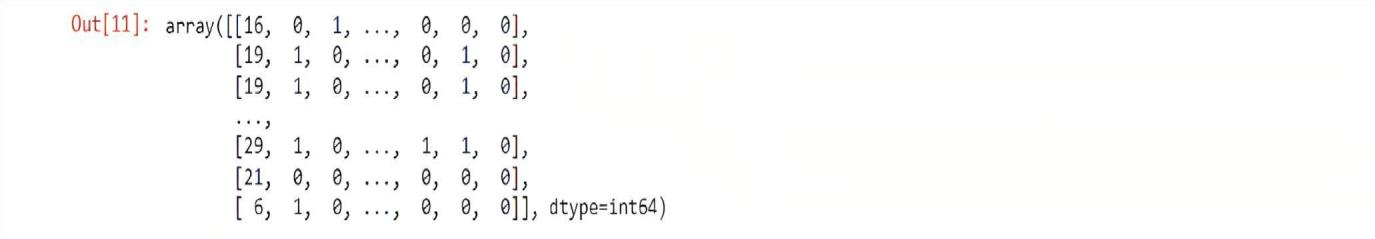
**Output:**



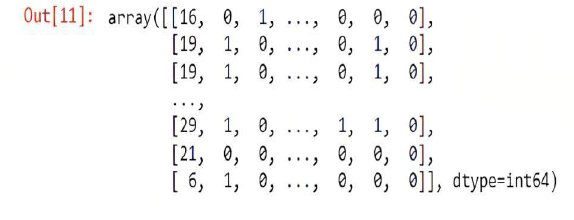
**Output:**



**Output:**

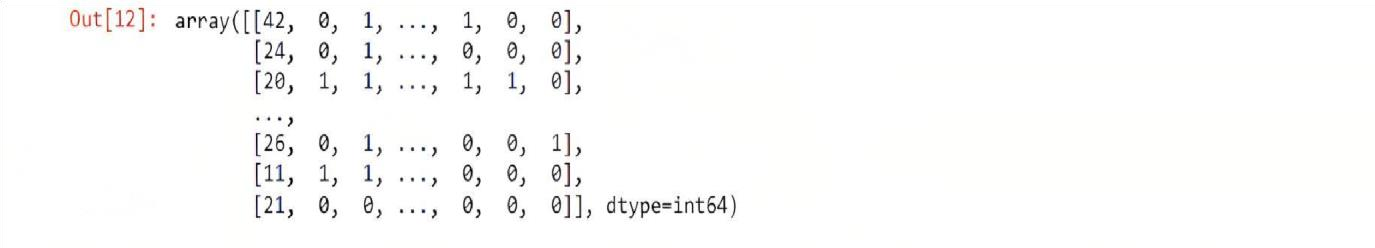


 **Output:**





**Output:**

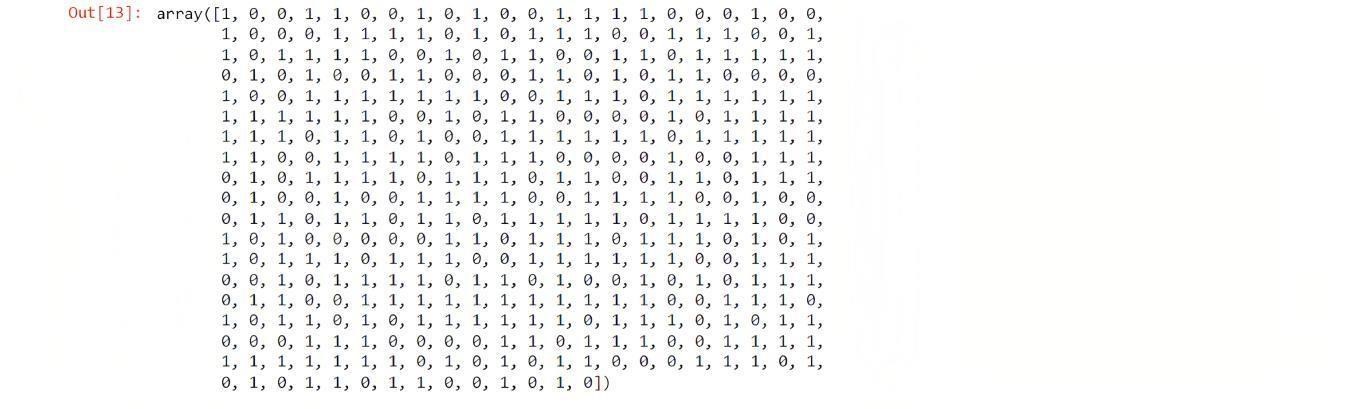




**Output:**

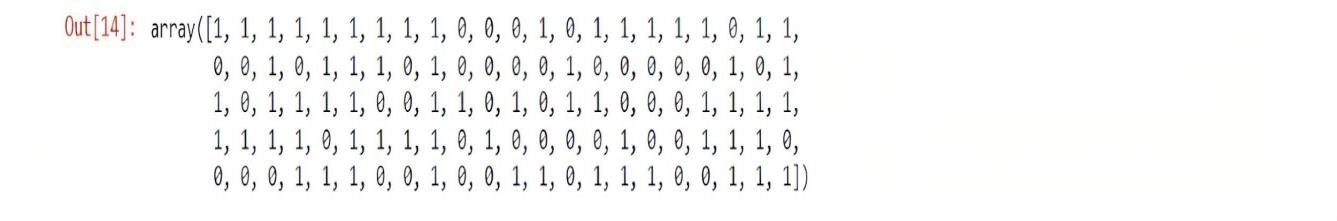


**Output:**



****

**Output:**

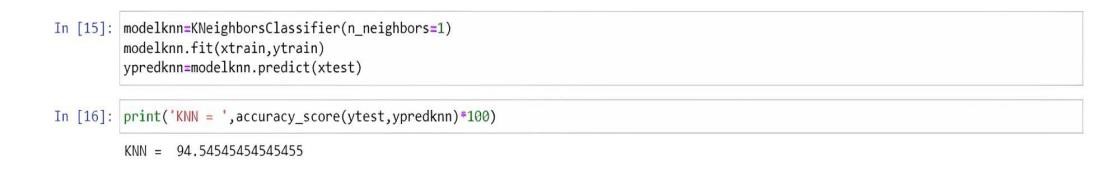
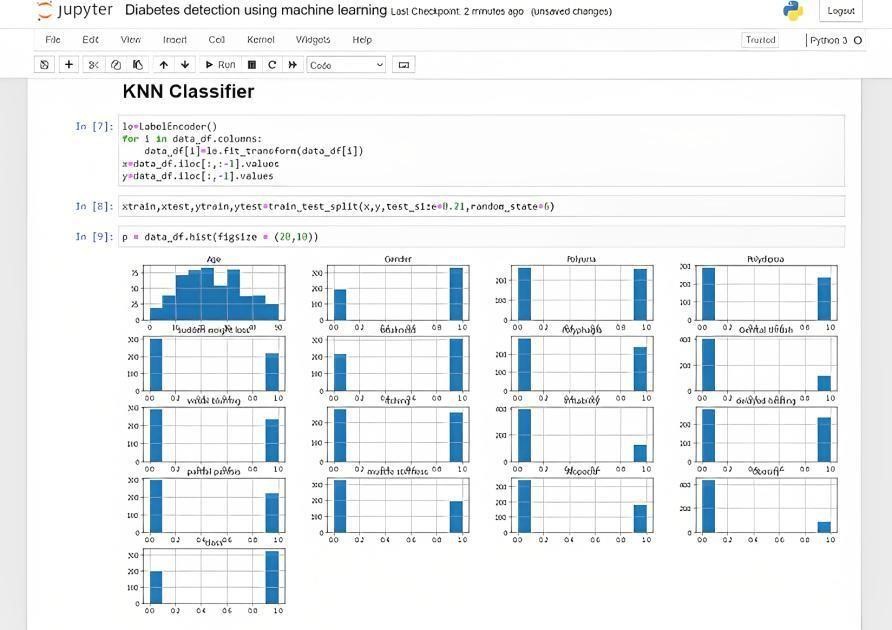


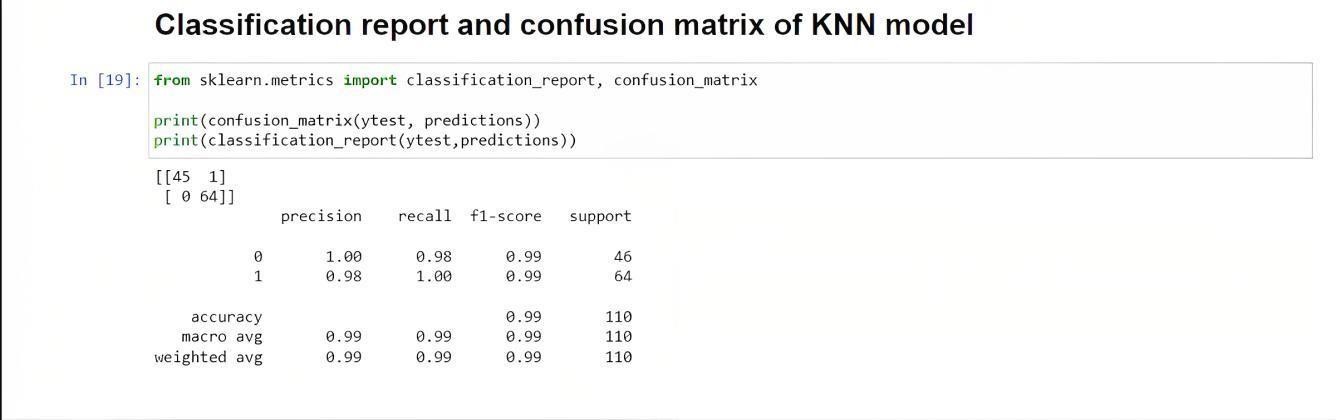
**5.2 Machine Learning Algorithms**

Here we take different machine learning algorithms and try to find algorithms which predict accurately.

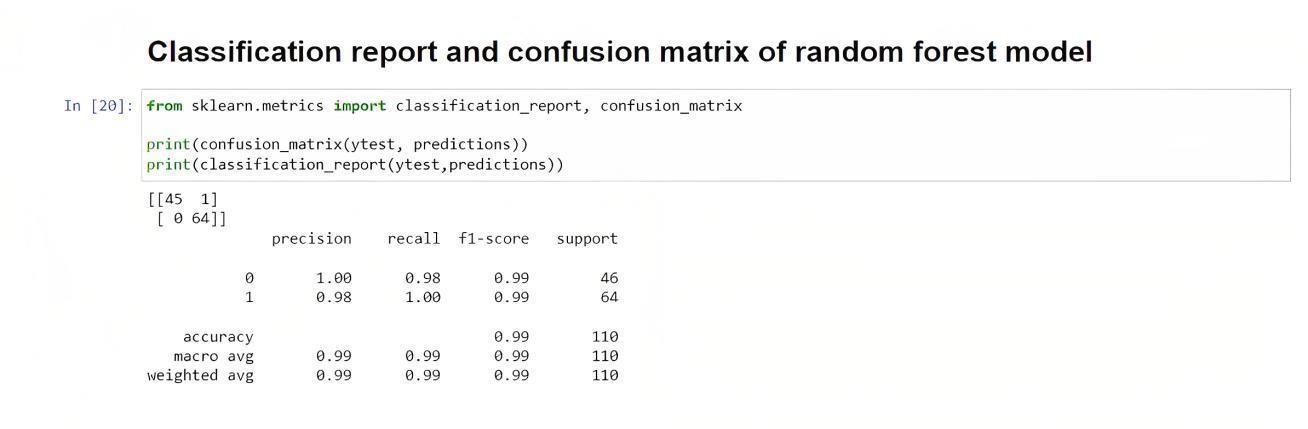
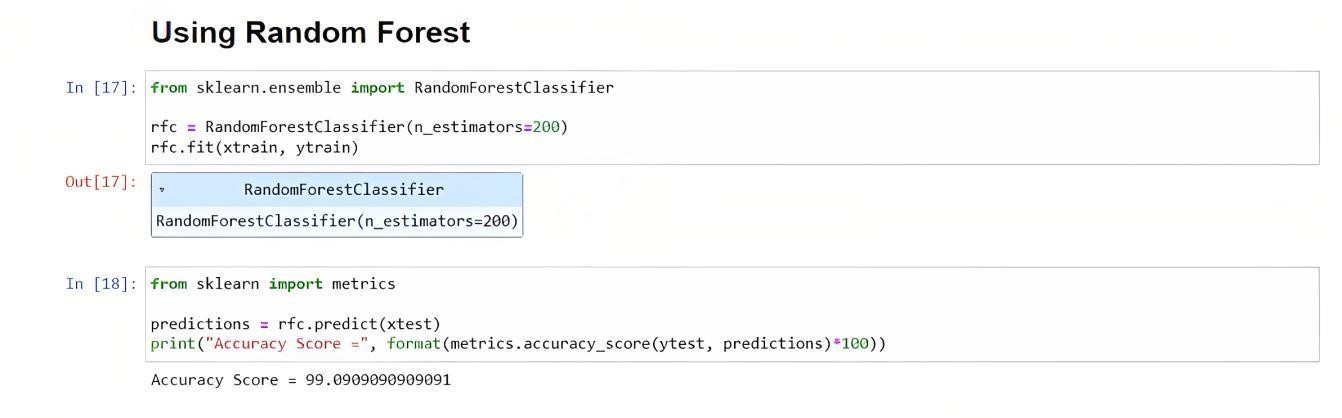
1. KNN Classifier
2. Random Forest
3. Decision Tree

**5.2.1 KNN Classifier:**

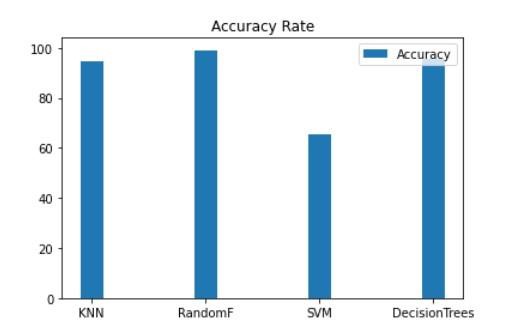
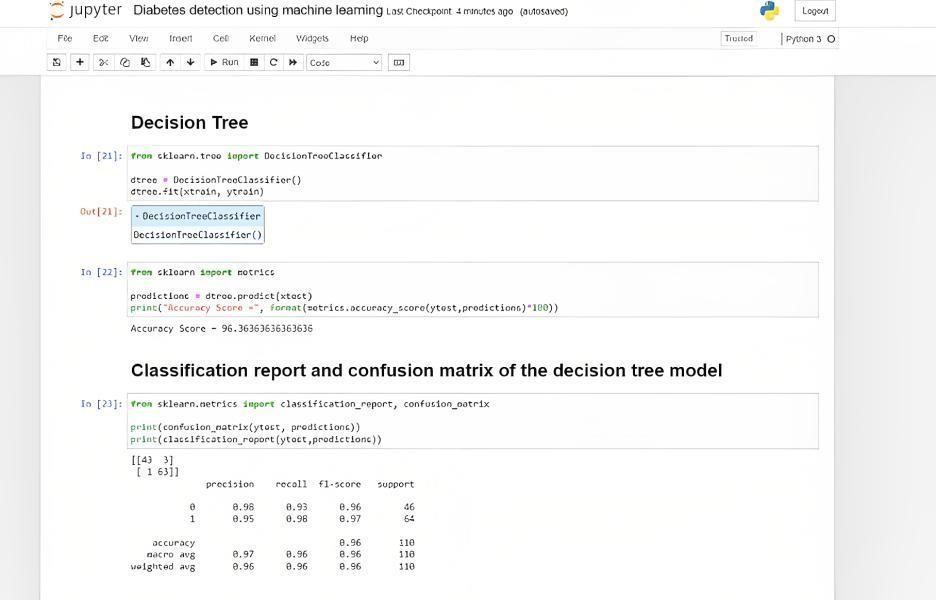




**5.2.2 Random Forest:**

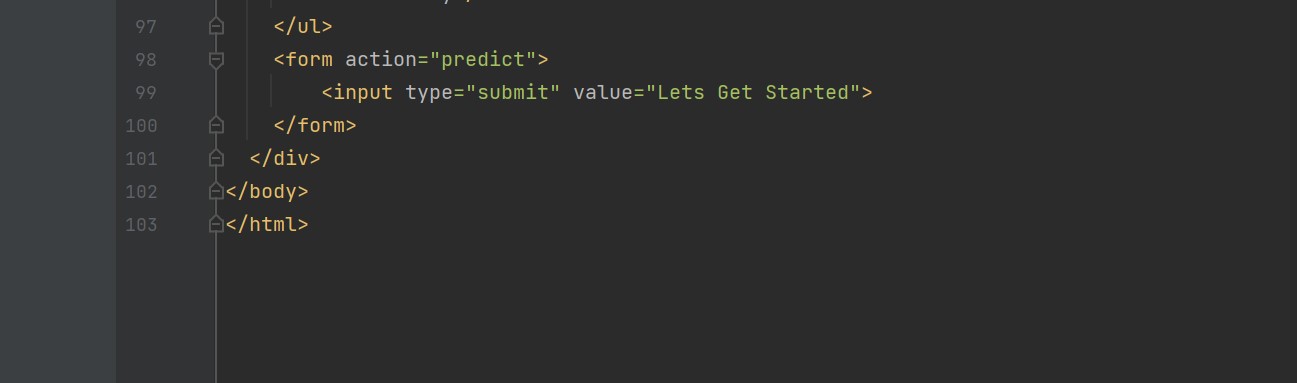
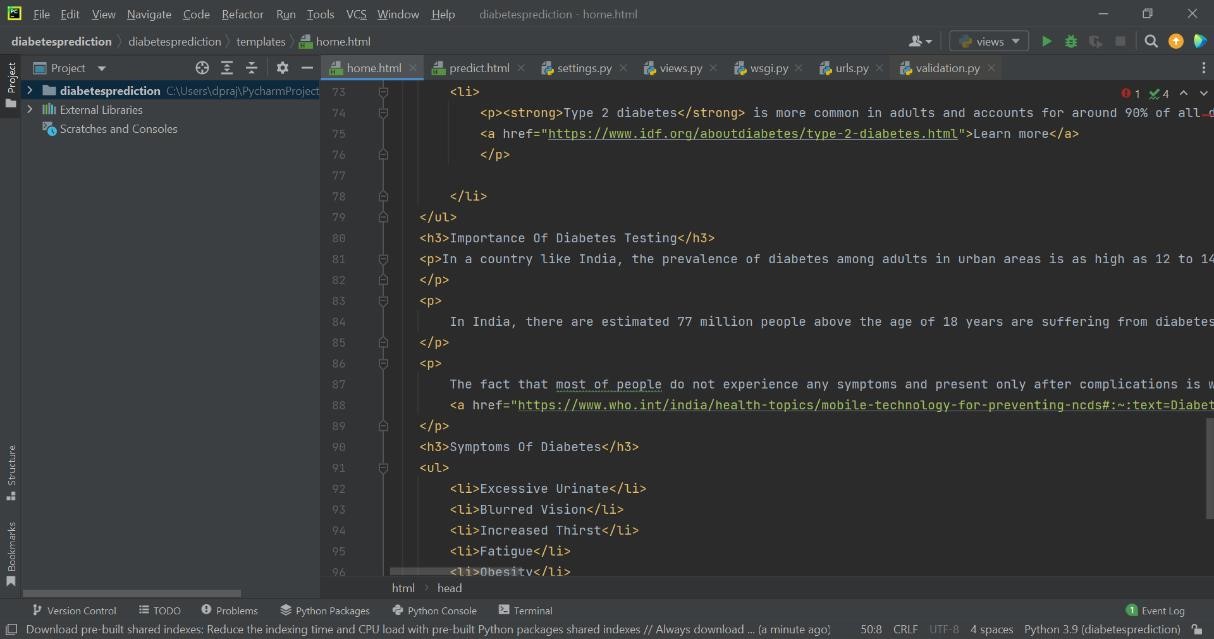
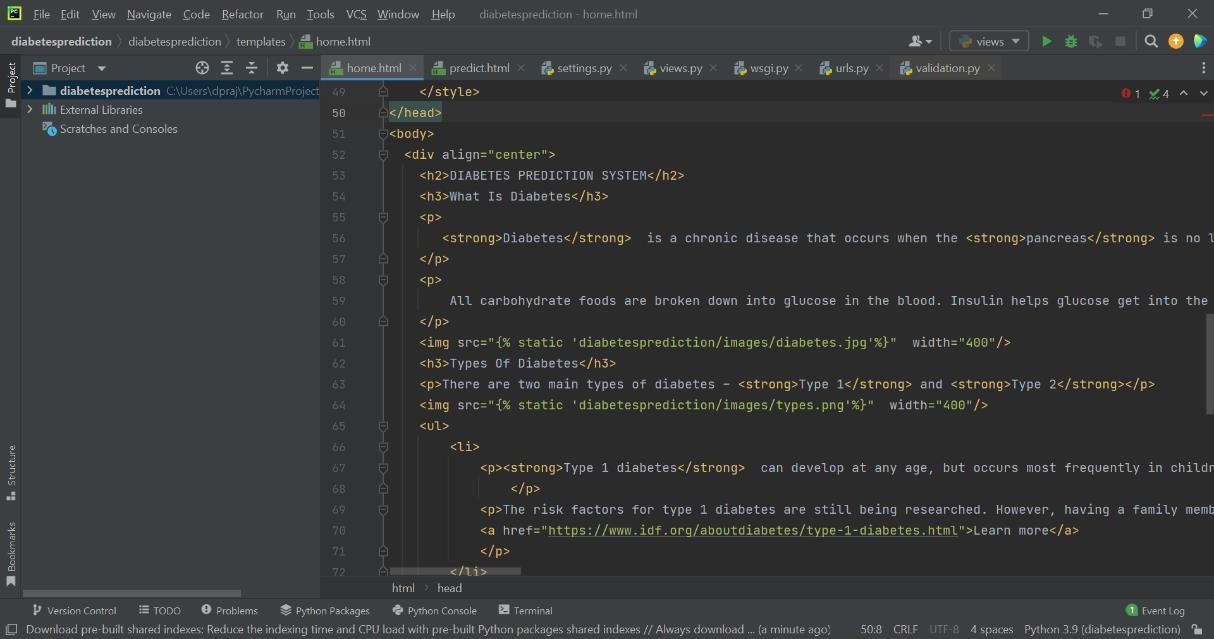
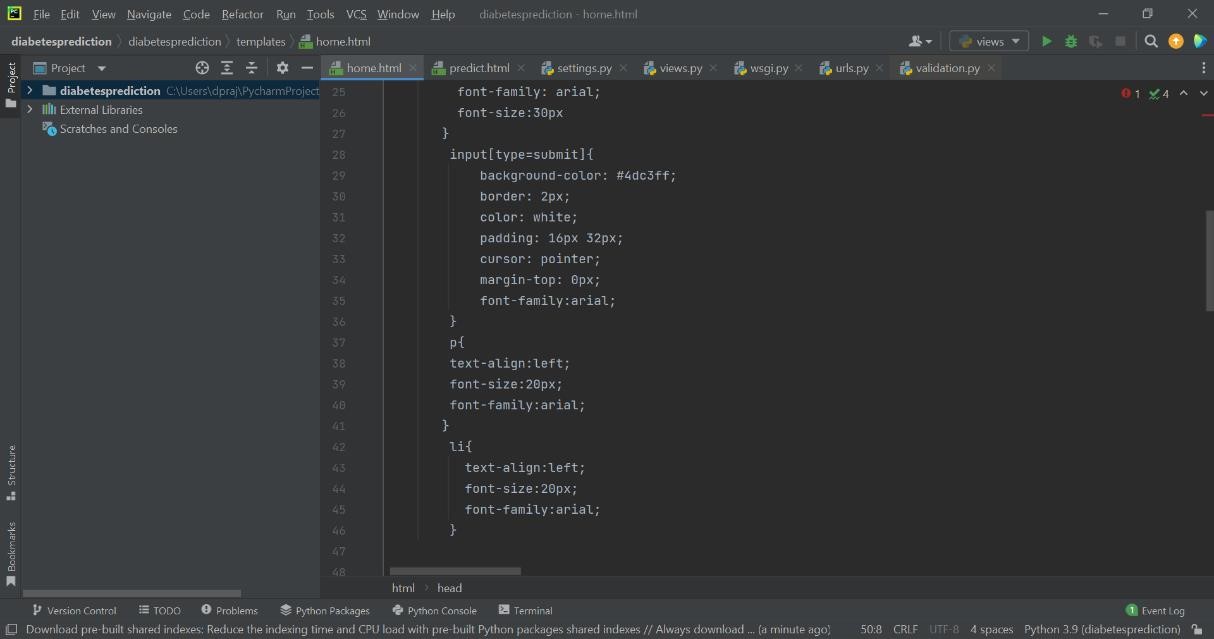
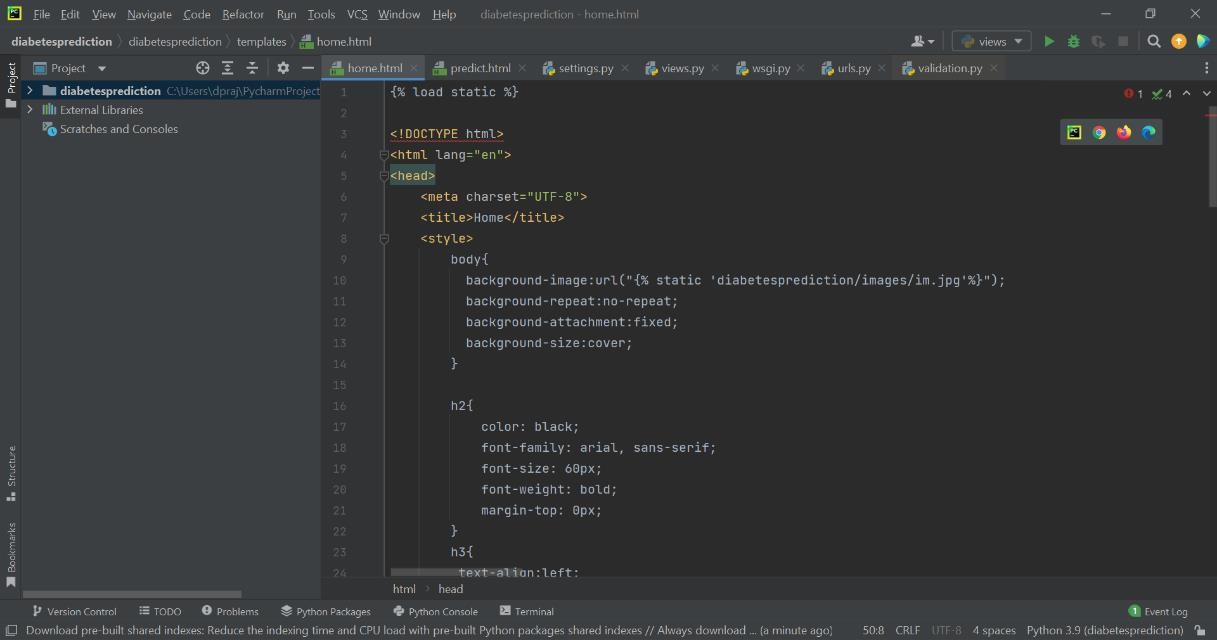


**5.2.3 Decision Tree:**

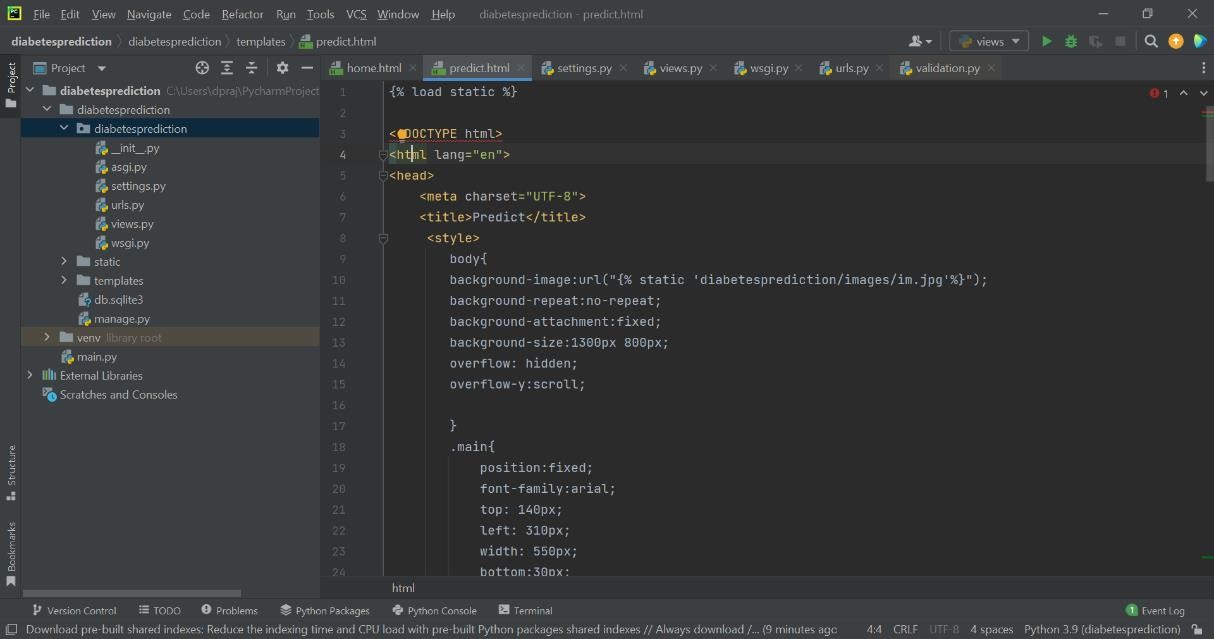


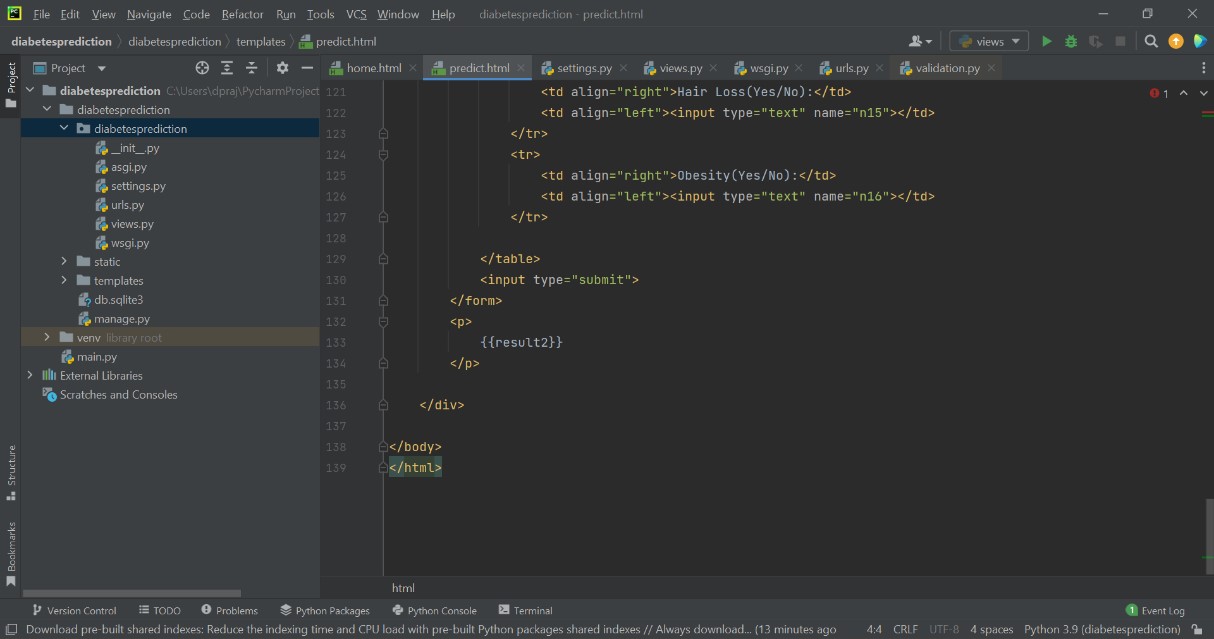
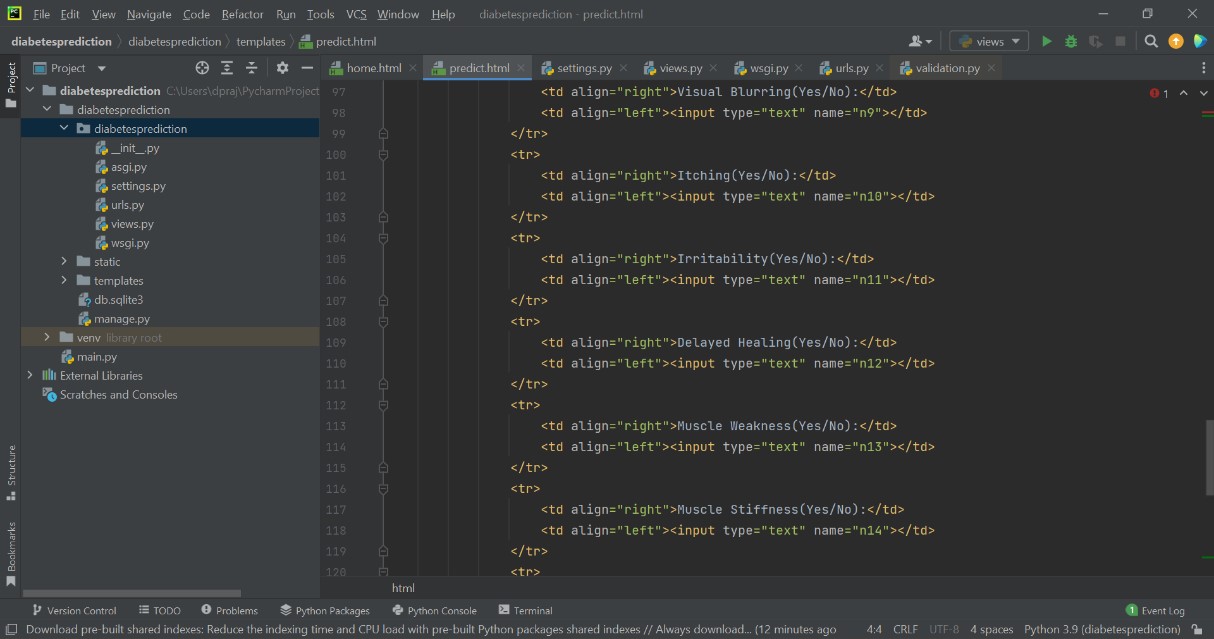
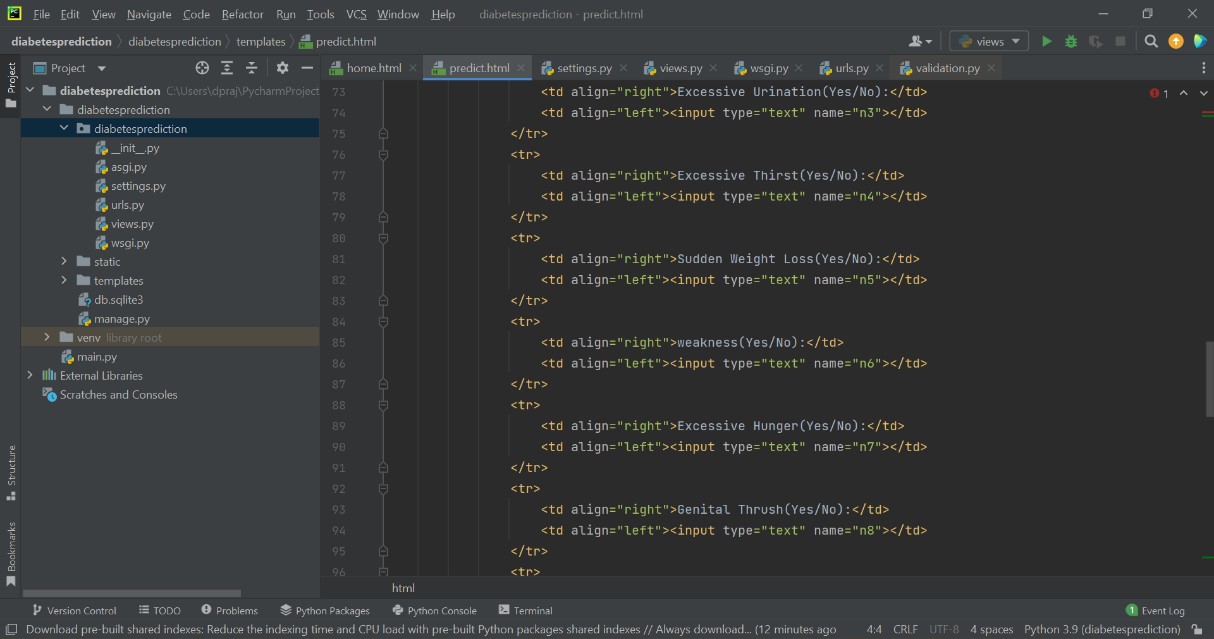
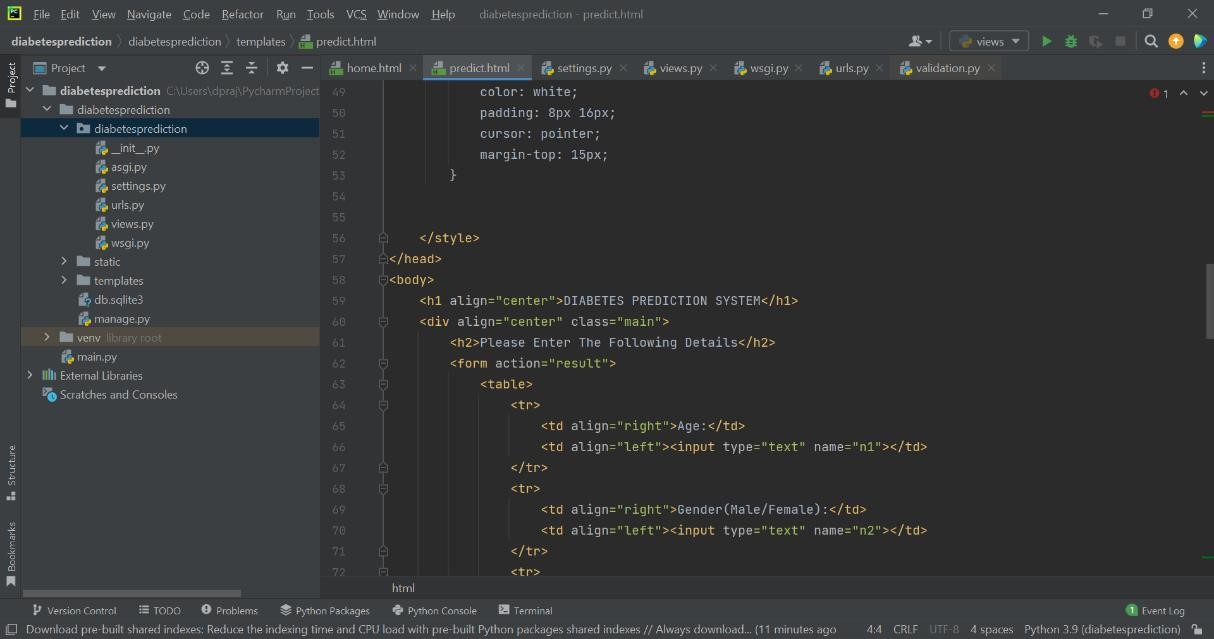
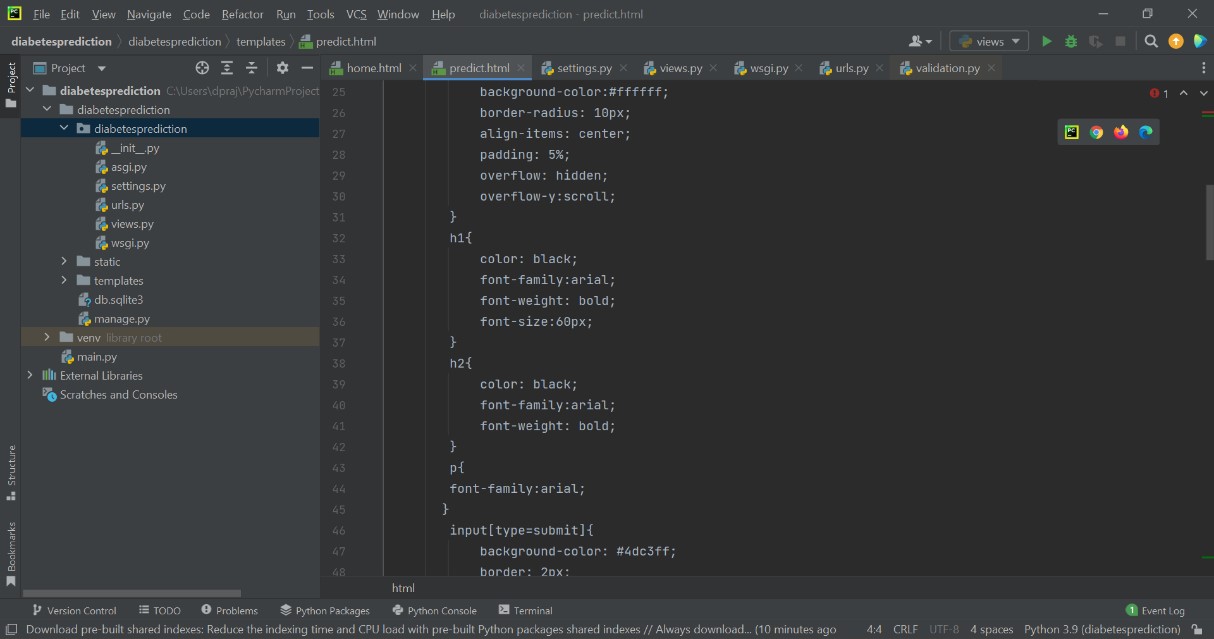
**5.3 UI DESIGN**

* + 1. **home.html:**

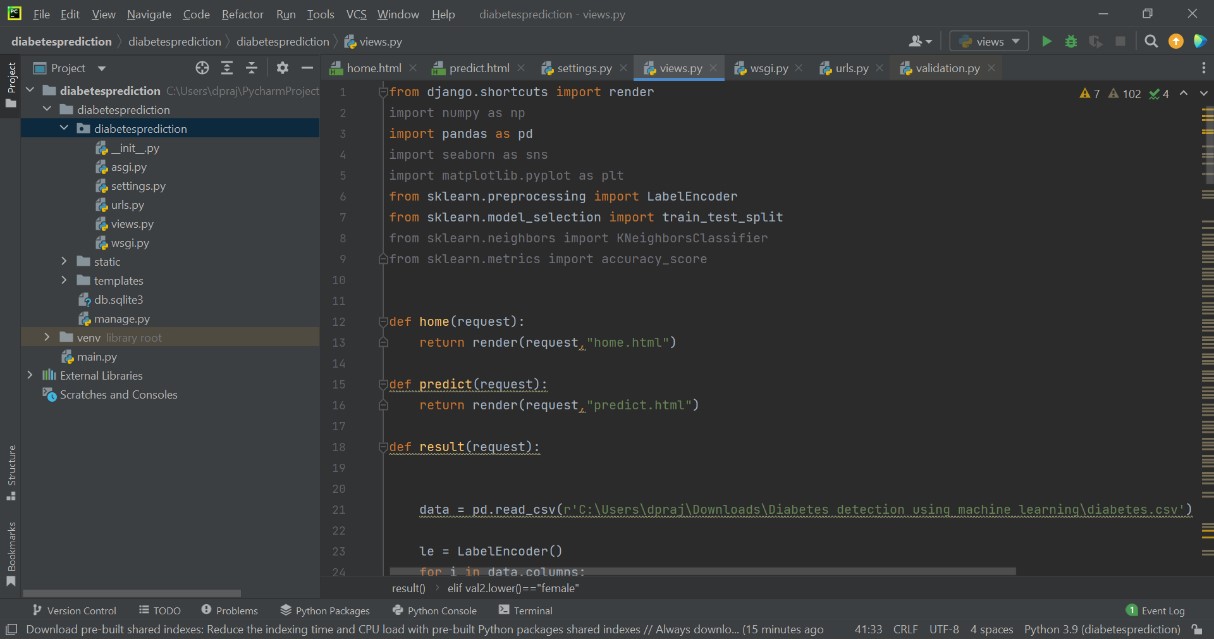


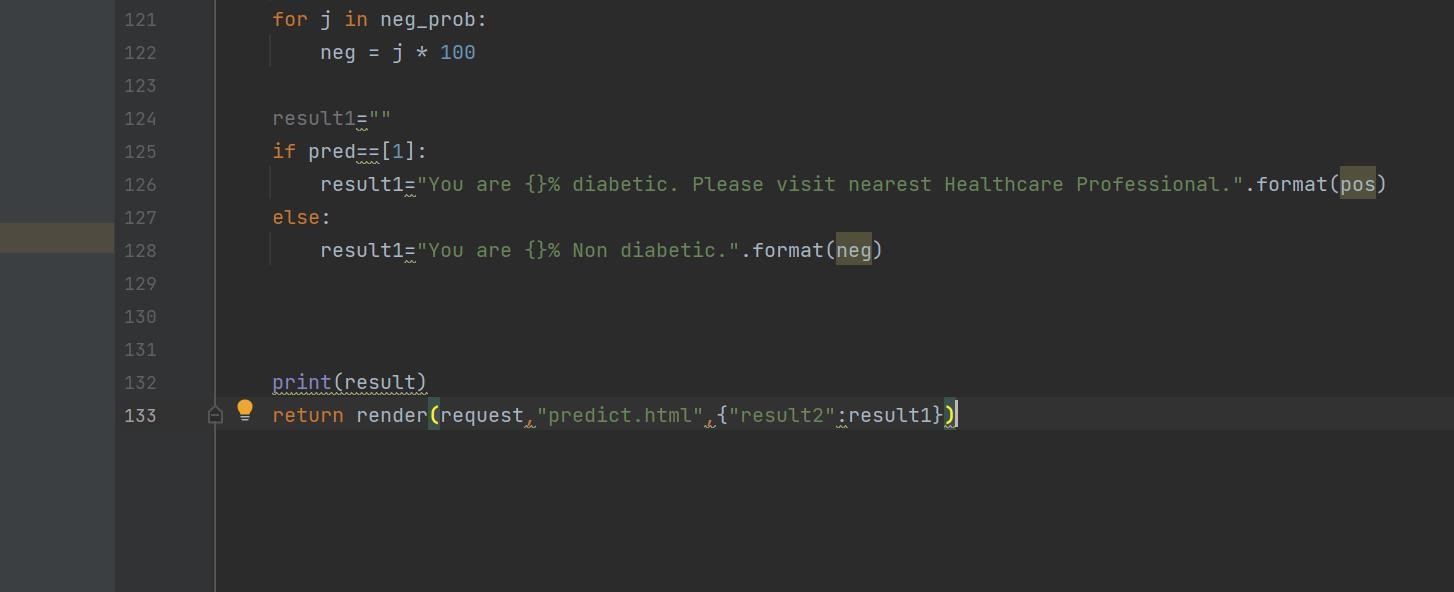
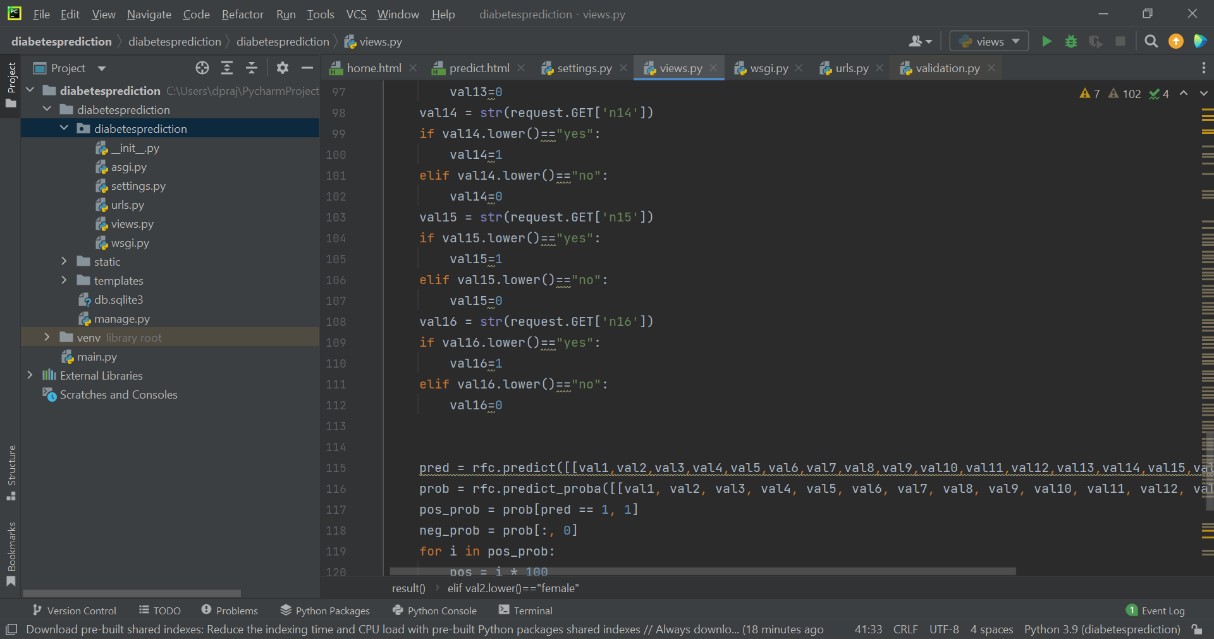
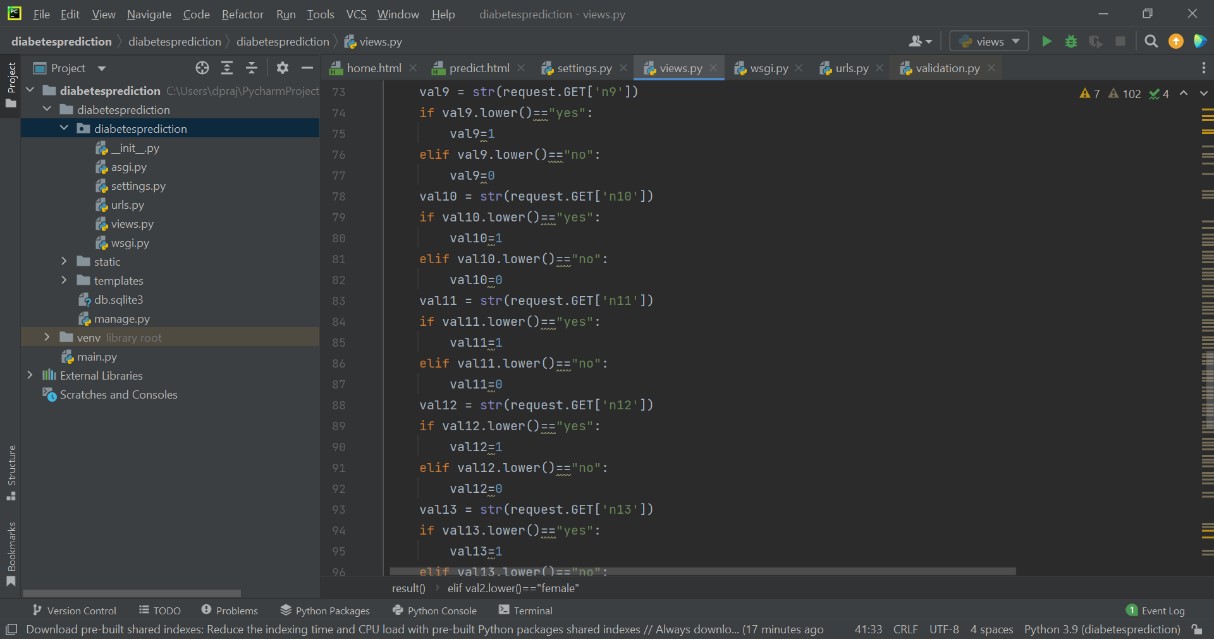
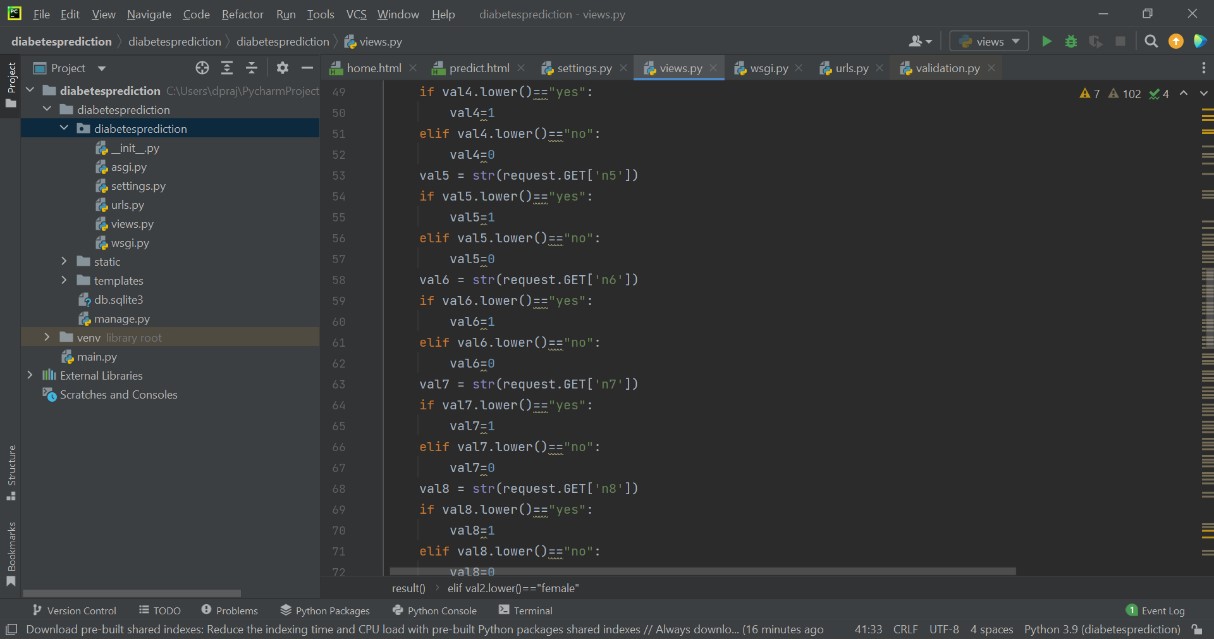
* + 1. **predict.html:**



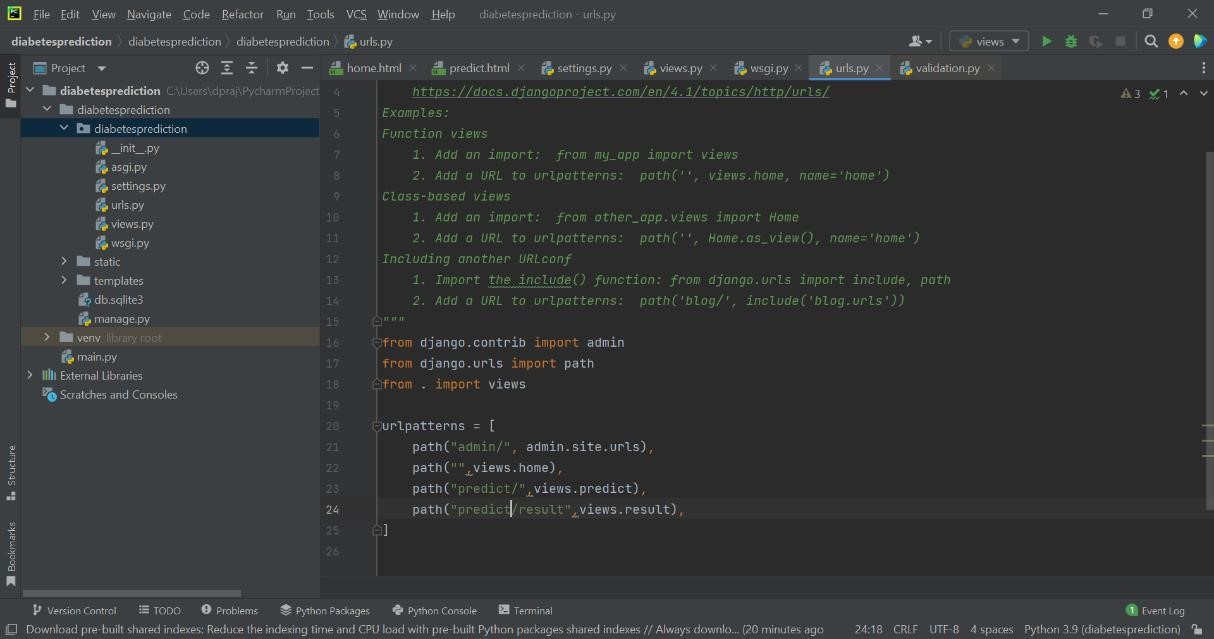


* + 1. **views.py:**

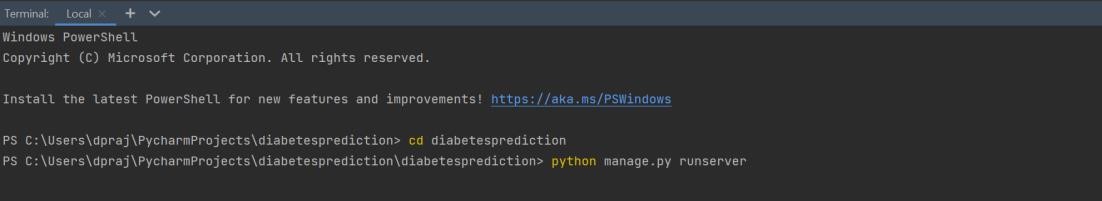




* + 1. **urls.py:**



* + 1. **Execution Command:**



**5.4 TESTING**

**5.4.1 Introduction**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement. Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Some prefer saying Software testing definition. In simple terms, Software Testing means the Verification of Application Under Test (AUT). Software Testing is important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software products ensure reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

* + - * **Cost-Effective:** It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case the bugs caught in the earlier stage of software testing, it costs less to fix.
      * **Security:** It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
      * **Product quality:** It is an essential requirement of any software product.

Testing ensures a quality product is delivered to customers.

* + - * **Customer Satisfaction:** The main aim of any product is to give satisfaction to their customers. UI/UX Testingensuresthe best user experience.

**5.4.2 TYPES OF TESTING:**

1. **UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

1. **INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

1. **FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be

exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

1. **SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

1. **WHITE BOX TESTING**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black box level.

1. **BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## Chapter 6 RESULTS, DISCUSSION AND COMPARISON

**6.1 RESULTS**

After performing the machine learning approach for training and testing we find that accuracy of the Random Forest is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that Random Forest is best with 99% accuracy and the comparison is shown below.

|  |  |  |
| --- | --- | --- |
| **ALGORITHM** |  | **ACCURACY** |
| KNN Classifier | 94.5% |  |
| Random Forest | 99.0% |  |
| Decision Tree | 96.3 |  |

Table 1. Accuracy comparison between algorithms

The Screenshots represent the output result of the screen here you can see the logo of the project. The user needs to provide some information in order to get the results of the diseases onto the screen.

The user is asked for his/her name and then there will be a dropdown list of some symptoms and the user needs to select at least three symptoms so that the model will predict the disease more accurately.

So, the user will select the symptoms from the dropdown. Now the user can select the algorithm in which they are going to know the result from. For that we have four buttons for four algorithms.

So based on the symptoms the machine learning model will predict the disease.

**6.1.1 Screenshots of the output**

* Homepage contains brief information about what diabetes is and types of diabetes.



Fig 9. Output screen 1

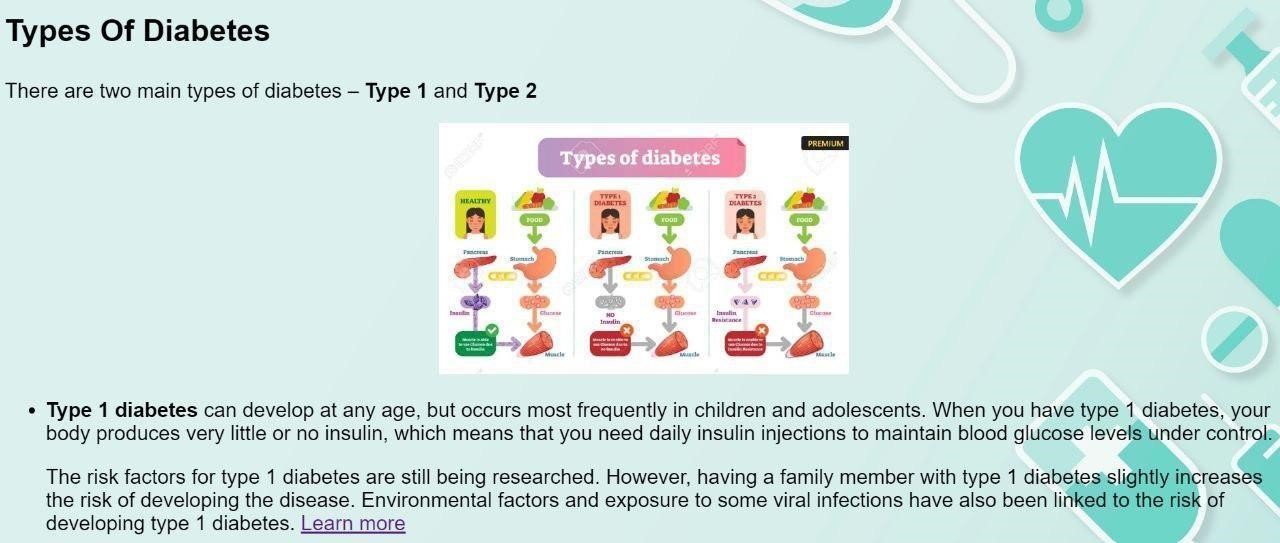


Fig 10. Output screen 2



Fig 11. Output screen 3

* Users should click on Let's Get Started to redirect to the next page.
* Users should enter the required values in the fields given.

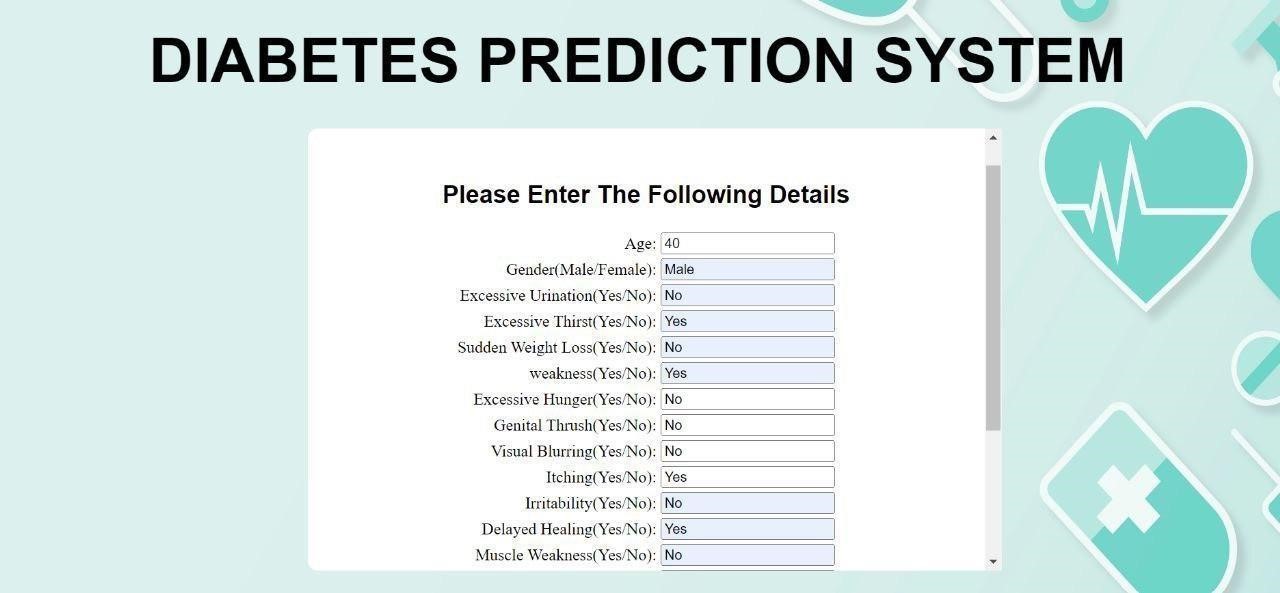


Fig 12. Output screen 4

* Click Submit to get results

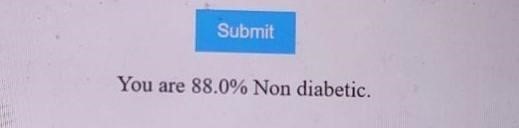


Fig 13. Output result 1

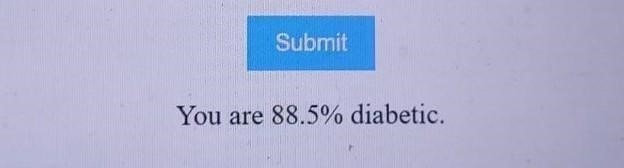


Fig 14.Output result 2

**6.2 Discussion**

Diabetes prediction using machine learning is a complex process that involves several challenges. One of the main challenges is data quality. Machine learning algorithms rely on large amounts of data to make accurate predictions. However, the data used in disease prediction is often incomplete, inconsistent, or inaccurate. This can lead to errors in prediction and compromise the effectiveness of the algorithm.

To overcome this challenge, data cleaning and preprocessing techniques are used to ensure that the data is of high quality. This involves removing any irrelevant or redundant data, correcting any errors, and standardizing the data format. These techniques help to ensure that the data is consistent and accurate, which improves the accuracy of the predictions.

Another challenge of disease prediction using machine learning is overfitting. Overfitting occurs when the machine learning algorithm learns the training data too well and is not able to generalize to new data. This can lead to inaccurate predictions and compromise the effectiveness of the algorithm.

To overcome this challenge, several techniques can be used, such as crossvalidation and regularization. Cross-validation involves splitting the data into training and validation sets and testing the algorithm on the validation set. This helps to ensure that the algorithm is not overfitting to the training data. Regularization involves adding a penalty term to the loss function, which helps to prevent the algorithm from overfitting.

In addition, the choice of machine learning algorithm is also critical to the accuracy of disease prediction. There are several machine learning algorithms available, such as logistic regression, decision trees, random forests, and neural networks. The choice of algorithm depends on the nature of the data and the problem at hand. For example, neural networks are often used for image recognition, while decision trees are used for classification problems.

**6.3 Comparison**

In general, machine learning-based diabetes prediction systems have several advantages over previously existing systems, such as:

* **Improved accuracy:** Machine learning algorithms can analyze large amounts of data and identify complex patterns that are difficult for humans to detect, leading to more accurate predictions.
* **Personalized predictions:** Machine learning algorithms can be trained on individual-level data, allowing for personalized disease risk assessments.
* **Early detection:** Machine learning algorithms can detect subtle changes in data patterns that can indicate the early stages of a disease, enabling earlier detection and potentially better treatment outcomes.
* **Reduced cost:** Machine learning algorithms can automate the prediction process, reducing the need for manual analysis and lowering the cost of disease prediction.
* **Scalability:** Machine learning algorithms can be easily scaled up to handle large amounts of data, making them ideal for population-level disease prediction tasks.

Overall, machine learning-based diabetes prediction systems have the potential to provide more accurate and personalized predictions than previously existing systems, potentially leading to better health outcomes for individuals and populations.

## Chapter 7 CONCLUSION AND FUTURE ENHANCEMENT

**7.1 Conclusion:**

It should come as no surprise that random forest is the model that delivers the most trustworthy results for this prediction given that it has an accuracy score of 0.98. Random Forest, which is one of the most well-known machine learning algorithms, falls under the umbrella term of supervised learning, which is a more comprehensive classification. Both classification and regression are examples of sorts of machine learning tasks that might potentially benefit from its use.

It is based on the concept of ensemble learning, which is the process of including a number of different classifiers in order to solve a challenging problem and improve the functionality of the model. This model will run with less computational resources and also it is better at classification so that the result will be more accurate than other models. This model is also effective when there are null values present in the dataset as those are ignored by using the probability estimation calculation. So it is one of the best models to predict diabetes using the symptoms provided by the user.

**7.2 Future Enhancements:**

There are several potential enhancements for a diabetes prediction using machine learning (ML) project that can improve its accuracy and usefulness.

1. **Feature engineering:** Feature engineering involves selecting the most relevant features from the available data to use in the ML model. It's important to identify the most important variables that may be predictive of diabetes, such as age, weight, blood glucose levels, and family history. This can be done using statistical methods or domain knowledge.
2. **Ensemble learning:** Ensemble learning involves combining multiple ML models to improve accuracy. For example, you can use a combination of decision trees, random forests, and neural networks to build an ensemble model that is more accurate than any individual model.
3. **Hyperparameter tuning:** Hyperparameters are variables that determine how the ML model is trained, such as the learning rate, regularization strength, and the number of layers in a neural network. Tuning these hyperparameters can help to optimize the model's performance and improve its accuracy.
4. **Data augmentation:** Data augmentation involves generating additional training data by applying various transformations to the existing data. For example, you can generate new samples by flipping, rotating, or scaling the images. This can help to increase the size of the training set and improve the robustness of the model.
5. **Explainable AI:** Explainable AI is a set of techniques that enable ML models to provide explanations for their predictions. This can help to increase the transparency and trustworthiness of the model. Some techniques for explainable AI include feature importance analysis, decision tree visualization, and LIME (Local Interpretable Model-Agnostic Explanations).
6. **Deployment:** Once the ML model is trained and validated, it needs to be deployed in a real-world setting. This can involve integrating it with other software systems, such as electronic health records or mobile health apps, and ensuring that it is secure and compliant with data privacy regulations.

By incorporating these enhancements, you can create a more accurate and robust diabetes prediction using the ML project that can have a significant impact on public health.

## Chapter 8 REFERENCES

**8.1 Web Links:**

1. https://www.ijraset.com/research-paper/disease-prediction-using-ml.
2. https://link.springer.com/chapter/10.1007/978-981-16-0538-3\_4.
3. https:<//www.springerprofessional.de/en/disease-prediction-based-on-sympto>ms-using-machine-learning/18334718.
4. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021- 00524-9.

**8.2 References:**

[1] “Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus” Md Faisal Faruque, Asaduzzaman, Iqbal H. Sarker, IEEE 2019. [2] “A Comprehensive Exploration to the Machine Learning Techniques for Diabetes Identification” Sidong Wei1, Xuejiao Zhao, Chunyan Miao Shanghai Jiao Tong University, China.

1. “Association Rule Extraction from Medical Transcripts of Diabetic Patients” Lakshmi K S, G Santhosh Kumar, 2014.
2. “Diabetes Care Decision Support System” 2nd International Conference on Industrial and Information Systems IEEE 2010.
3. “An Intelligent Mobile Diabetes Management and Educational System for Saudi Arabia: System Architecture” M.M. Alotaibi, R.S.H. Istepanian, A.Sungoor and N. Philip, IEEE 2014.
4. “Machine Learning Techniques for Classification of

Diabetes and Cardiovascular Diseases” by BerinaAlic, Lejla Gurbeta, IEEE 2017.

1. “Performance Analysis of Classification Approaches for the Prediction of Type II Diabetes” by M. Durgadevi, M. Durgadevi, IEEE 2017.
2. “Cloud-Based Diabetes Coaching Platform for Diabetes Management”

Elliot B. Sloane Senior Member IEEE, Nilmini Wickramasinghe, Steve Goldberg 2016. [9] Minyechil Alehegn and Rahul Joshi, “Analysis and prediction of diabetes diseases using machine learning algorithm”:International Research Journal of Engineering and Technology Volume: 04 Issue: 10 | Oct -2017

1. P. Suresh Kumar and V. Uma Tejaswi, “Diagnosing Diabetes using Data

Mining Techniques”,International Journal of Scientific and

ResearchPublications, Volume 7, Issue 6, June 2017 705 ISSN 2250-3153.

1. “Clustering Medical Data to Predict the Likelihood of Diseases” by Razan Paul, Abu Sayed Md. Latiful Hoque, IEEE 2010.
2. “Robust Parameter Estimation in a Model for Glucose Kinetics in Type 1 Diabetes Subjects” Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept 3, 2006.
3. Anjali C And Veena Vijayan V, Prediction and Diagnosis of Diabetes Mellitus, “A Machine Learning Approach” ,2015 IEEE in Intelligent Computational Systems (RAICS) | Trivandrum.
4. Ridam Pal ,Dr. Jayanta Poray, and Mainak Sen, ,“Application of Machine Learning Algorithms on Diabetic Retinopathy”, 2017 2nd IEEE International Conference On Recent Trends In Electronics Information & Communication Technology, May 19-20, 2017, India.
5. Dr. M. Renuka Devi and J. Maria Shyla, “Analysis of Various Data Mining Techniques toPredict Diabetes Mellitus”, International Journal ISSN 09734562 Volume 11, Number 1 (2016) pp 727-730
6. Gauri D. Kalyankar, Shivananda R. Poojara and Nagaraj V. Dharwadkar,” Predictive Analysis of Diabetic Patient Data Using Machine Learning and Hadoop”, International Conference On I-SMAC,978-1-5090-3243-3,2017.
7. Ayush Anand and Divya Shakti,” Prediction of Diabetes Based on Personal

Lifestyle Indicators”, 1st International Conference on Next Generation Computing Technologies, 978-1-4673-6809-4, September 2015.

1. B. Nithya and Dr. V. Ilango,” Predictive Analytics in Healthcare Using

Machine Learning Tools and Techniques”, International Conference on Intelligent Computing and Control Systems, 978-1-5386-2745-7,2017.

1. Dr Saravana kumar N M, Eswari T, Sampath P and Lavanya S,” Predictive

Methodology for Diabetic Data Analysis in Big Data”, 2nd International Symposium on Big Data and Cloud Computing,2015.

1. Aiswarya Iyer, S. Jayalalitha and Ronak Sumbaly,” Diagnosis of Diabetes Using Classification Mining Techniques”, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015.
2. P. Suresh Kumar and S. Pranavi “Performance Analysis of Machine Learning

Algorithms on Diabetes Dataset using Big Data Analytics”, International

Conference on Infocom Technologies and Unmanned Systems,

978-1-5386-0514-1, Dec. 18-20, 2017.

1. Mani Butwall and Shraddha Kumar,” A Data Mining Approach for the Diagnosis of Diabetes Mellitus using Random Forest Classifier”, International Journal of Computer Applications, Volume 120 - Number 8,2015.
2. K. Rajesh and V. Sangeetha, “Application of Data Mining Methods and

Techniques for Diabetes Diagnosis”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 3, September 2012.

[24]Humar Kahramanli and Novruz Allahverdi,”Design of a Hybrid System for Diabetes and Heart Disease”, Expert Systems with Applications: An International Journal, Volume 35 Issue 1-2, July, 2008.

1. B.M. Patil, R.C. Joshi and Durga Toshniwal,”Association Rule for

Classification of Type-2 Diabetic Patients”, ICMLC '10 Proceedings of the 2010 Second International Conference on Machine Learning and Computing, February 09 - 11, 2010.

1. Dost Muhammad Khan1, Nawaz Mohamudally2, “An Integration of Kmeans and Decision Tree (ID3) towards a more Efficient Data Mining

Algorithm ”, Journal Of Computing, Volume 3, Issue 12, December 2011