

DIABETES PREDICTION USING MACHINE LEARNING

A MINOR PROJECT REPORT

Submitted by

V. SIVA SUPRADEEP [Reg No: RA1911030010104]

P. SAI PHANIDHAR[Reg No: RA1911030010101]

Under the Guidance of

Mrs. G. Parimala

Assistant Professor, Department of Networking and Communications

in partial fulfillment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Cyber Security



DEPARTMENT OF NETWORKING AND COMMUNICATION

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

NOVEMBER 2022



Department of Networking and Communications

SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech, Computer Science Engineering

Student Name : Vakada Siva Supradeep

Registration Number : RA1911030010104

Title of Work : Diabetes Prediction using Machine Learning

I / We hereby certify that this assessment complies with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g.fellow students, technicians, statisticians, external sources)
- Complied with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the university policies and regulations.

DECLARATION:

I am aware of and understand the university's policy on academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign them with the date for every student in your group.



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203
BONAFIDE CERTIFICATE**

Certified that this B. Tech project report titled “**DIABETES PREDICTION USING MACHINE LEARNING**” is the Bonafide work of Mr. V. Siva Supradeep[Reg No: RA1911030010104] and Mr. P. Sai Phanidhar[Reg No: RA1911030010101] who carried out the project work under my/our supervision. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE

SIGNATURE

Mrs. G. Parimala

Dr. K. Annapurani Panaiyappa

GUIDE

HEAD OF THE DEPARTMENT

Assistant Professor
Dept. Of Networking and
Communications

Dept. Of Networking and
Communications

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his valuable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. We are incredibly grateful to our Head of the Department, **Dr. Annapurani Panaiyappan.K** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program co-ordinators **Dr. T. Y. J. Naga Malleswari** Associate Professor, and Panel Head, **Dr. C. N. S. Vinoth Kumar**, Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support. We register our immeasurable thanks to our Faculty Advisor, **Dr. C. N. S. Vinoth Kumar**, Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Mrs. G. Parimala**, Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Vakada Siva Supradeep

Pechetti Sai Phanidhar

ABSTRACT

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 Endeavour. 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 in order 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 or not 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. This will be accomplished by integrating machine learning, data visualization, and data interpretation. The use of machine learning, which is becoming more important in the modern healthcare sector, will be the focus of this research. Massive datasets may be found in the healthcare business.

INDEX

SR.NO	TITLE	PG.NO
1)	INTRODUCTION	7
2)	LITERATURE SURVEY	13
3)	PROBLEM DEFINITION	18
4)	REQUIREMENTS	22
5)	PLANNING AND ESTIMATION	28
6)	DESIGN AND IMPLEMENTATION	32
7)	IMPLEMENTATION	37
8)	ADVANTAGES	46
9)	CONCLUSION	47
10)	BIBLIOGRAPHY	48
11)	SCREENSHOTS	52
12)	SOURCE CODE	58
13)	PLAGIARISM CHECK	61
14)	PAPER SUBMISSION PROOF	62

Chapter 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 a number of 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.

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, LabelEncoder, 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.

Objectives of the Project

We categorise 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 visualisation, 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 practises 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 practise, however, the patient needs to make a prediction as to whether or not they fall into the diabetic or non-diabetic categories. These kind 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.

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 endeavour, it's possible that many lives will be preserved.

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 of 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 categorise 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 particular conditions. Like a real tree, a decision tree has a central node from which branches extend in many directions.

Chapter 2

LITERATURE SURVEY

Sr no.	Name Of Paper	Author and Publication year	Abstract
1.	An Innovative Approach for Diabetes Prediction.	Krishna Priya A S, Shemitha P A, Dr G. Kiruthiga, Vol-8 Issue-3 2022. IJARIE- ISSN(O)-2395-4396.	<p>Algorithm Used: Naive Bayes Classification</p> <p>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.</p> <p>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.</p> <p>Within the framework of this experiment suggestion, the Naive Bayes Classification points to diabetes. The process of collecting data from a dataset and organising it into a format</p>

			<p>that may be used in further analysis is known as "information mining."</p> <p>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.</p> <p>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.</p>
2.	Diabetes Prediction using Machine Learning Algorithms.	<p>Aishwarya Mujumbara, Dr. Vaidehi V, VIT chennai India, Mother Teresa Women's University, kodaikanal, India.INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019 ICRTAC 2019</p>	<p>Algorithm Used: Big Data Analytics</p> <ol style="list-style-type: none"> 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

			<p>diagnosis.</p> <p>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.</p> <p>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.</p> <p>In addition to this, a pipeline model was imposed for diabetes prediction with the goal of boosting the accuracy of categorization.</p>
3.	Diabetes Prediction using Machine Learning Techniques.	<p>Mitushi Soni, Dr. Sunita Varma</p> <p>Dept of Computer Science and Engineering, Shri G.S. Institute of Technology and Science, Indore, India.</p> <p>International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 09, September-2020</p>	<p>Algorithm used: Random Forest, Support Vector Machine.</p> <p>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.</p>

4.	Diabetes Prediction Using Machine Learning Techniques	<p>Tejas N. Joshi*, Prof. Pramila M. Chawan**</p> <p>*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. Joshi S. Dewangan.et.al. Int. Journal of Engineering Research and Application ISSN: 2248-9622, Vol. 8, Issue 1, (Part -II) January 2018, pp.-09-13</p>	<p>Algorithms used: ANN, SVM, Logistic Regression.</p> <p>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 utilise 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.</p>

Chapter 3

PROBLEM **DEFINITION**

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 or not a person is sick, and achieve the greatest validation scores possible for those models.

Existing System

The SVM Algorithm is used in the current method to calculate the nuances of diabetes. In order to organise 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, decentralised 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:

Concerns Regarding Accuracy An automated framework does not, on its own, ensure precision, and the data from the distribution centre 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.

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.

Methodology

1) Data Acquisition:

"Data acquisition is the process of sampling signals that measure actual physical conditions and converting these samples into digital numeric values," says Wikipedia. For the purposes of storage, analysis, and processing, data acquisition systems (DAS or DAQ) convert the physical conditions of analogue waveforms into digital values. Data refers to raw facts and figures that might be organised or unorganised, whereas acquisition refers to the process of acquiring data for a certain purpose. Data acquisition is the process of gathering information from various resources for the purpose of storing, cleaning, and preprocessing it before it is utilised in subsequent processes. This is the procedure of collecting relevant business data, reformatting it into a usable format, and entering it into the desired system. An average data scientist will spend 80% of their time on data discovery, cleaning, and analysis. Despite Machine Learning's widespread use, many use cases still require adequate labelled data. Inadequate data and improper data purification can hinder the performance of even the most sophisticated Machine Learning algorithms. Moreover, Deep Learning strategies need huge amounts of data since, unlike Machine Learning, they automatically construct features.

If not, we'd be getting trashed and sending trash out. Therefore, data acquisition or collecting is a crucial step.

2) Pre-Processing:

The details of our record-keeping practises as they pertain to our visual works are presented here. Now that we have our training set, it's time to begin utilising it in the classification process by extracting relevant features from it. After the data has been cleansed and normalised, it is then added to the network. The adjusted range for the pricing is between zero and one. The first step in using a machine learning model is known as "data preparation." It's the very first and crucial step in making an ML model.

It's not always possible to find thoroughly cleaned and organised data while building a machine learning application. Additionally, information must be cleaned and formatted before being used. This is why we use the data preparation process. The steps are: obtaining the data set, The Process of Bringing in New Materials Reconnaissance, Data Deficit, Categorical Data Encoding, Dissecting data into a test set and a training set, then adjusting the features.

3) Classification:

One way to evaluate the efficacy of a machine learning system is through the train-test split. It may be implemented in any supervised learning method and utilised for either classification or regression.

The procedure involves splitting the dataset in half. The first group is used to train the model; this is the training dataset. Instead of using the data from the second subset to train the model, it is utilised as input to the model, which then generates predictions that are compared to the actual values in the dataset. The second data set is the test data set. Training Dataset : The machine learning model is fitted using this dataset. When evaluating the accuracy of a machine learning model, the "test dataset" is what's put to use. The purpose is to evaluate the ML model's accuracy using data that wasn't utilised during training. In this way, we plan to put the model into action. To rephrase, we want to train it on historical data with known inputs and outputs, and then use it to generate predictions in future circumstances when we don't have access to training data or know the expected output or target values. Train-testing is a valid method when a large enough dataset is at hand.

Chapter 4

REQUIREMENTS

Requirements

Hardware:

1. Processor: Intel i3 or above.
2. RAM: 6GB or more

Software:

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

Technologies Used:-

This project we will develop using python and web technology.

1)Data Analysis:

What we term "data analysis" is the methodical use of statistical and/or logical tools to data in order to describe, illustrate, summarise, and assess it. The correct and suitable interpretation of study results is a crucial part of protecting the honesty of data. Information analysis is essential in business since without it, it is impossible to understand the challenges that a company is up against. Information analysis collects, sorts, and interprets raw data, then delivers it alongside contextual information. In addition to helping businesses receive dependable data on deal marketing and capital development, research enables teams within businesses to collaborate and produce better results. It helps businesses analyse previous performance and develop strategies for the future. Why learn it if you're creative this could be

the legitimate capacity to seek out data visualisation allows key chiefs in an incredibly business, typically non-tech senior executives, to see research presented externally in diagrams, graphs, etc. in order to set up patterns and designs and see advanced information.

2)Predictive Analysis:

The application of information, statistical algorithms, and machine learning approaches to determine the likelihood of future events based on previous data is what is referred to as predictive analytics. The objective here is to provide the most accurate possible forecast of what will occur in the future, which means going beyond simply being aware of what took occurred in the past. The purpose of predictive analytics is to base forecasts and assumptions on a collection of data relating to the recent and ongoing past. The foundation of predictive analytics is predictive modelling, which frequently makes use of machine learning. The goal of predictive analytics is to make use of past data in order to forecast the probability of a certain outcome in the future. Machine learning and predictive analytics go hand in hand since predictive models often include some kind of machine learning computation. These models are routinely updated after some period of time so that they can respond to new data or characteristics and deliver the results that the company desires.

Advantages of predictive analytics:

1. Find new product/service opportunities.
2. Optimize product and execution.
3. Gain a more profound comprehension of buyers.
4. Reduce cost and hazard.
5. Address issues before they happen.
6. Meet customer expectations.
7. Improved cooperation.

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 preexisting 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 programme 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 programme 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 programme 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 programme; 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 that use the Anaconda distribution, which includes over 250 widely-used tools for data science and runs on Windows, Linux, and MacOS.

KNN Classifier

K-Nearest Neighbors (K-NN) takes into account the likelihood of similarities between the new case/data and the existing cases and assigns it to the group with the highest degree of similarity. This means that the K- NN approach may be used to rapidly classify newly generated data into an appropriate category. Although the K-NN technique may be used for either regression or classification, it is often employed for the latter.

Random Forest

The ensemble learning method is used for classification and regression, and it works by creating a large number of decision trees during training period and then outputting the class, i.e. the mode of the classes, or the regression of the individual trees. Random Forest, as the name suggests, is a classifier made up of several decision trees applied to different subsets of the supplied dataset, with the average used to improve the projected accuracy of that dataset.

SVM

Supervised learning algorithms include the SVM. Methodology for analysing data and identifying patterns and anomalies (regression, classification, and outlier identification). On the hyperplane, SVM will categorise the data. However, its primary application is in Machine Learning, namely for Classification problems. The foundation of Support Vector Machine (SVM) is Supervised Learning. For SVM to work, there must be a training set containing labels. If test data is given into the model after training, it classifies the data accordingly. In linear classification tasks, it excels. When used to nonlinear classification, the kernel method of translating the inputs into high dimensional feature space allows for efficient operation. For the purpose of categorization, it builds a hyper plane. The hyper-plane is selected to optimise the separation between the hyper-plane and the nearest data point on each side.

Decision Tree

Even while the supervised learning technique of the decision tree may be applied to both classification and regression problems, the latter is where it finds the most widespread application. It is a classifier in the form of a tree, with core nodes containing attributes from the dataset, branches standing in for decision rules, and leaf nodes standing for the final outcome. It's a visual representation of all the options available when making a choice or solving an issue with fixed constraints. The name "decision tree" comes from its resemblance to a tree's structure, which begins with a central node and grows branches off of it.

Evaluation Metrics:

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

1) Accuracy:

Good old accuracy is actually 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.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{All Samples}}$$

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.

2) **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.

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total Predicted Positives}}$$

We focus on accuracy when our predictions need to be correct, i.e., H. Idealistically, 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.

3) Recall:

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

$$\text{Recall} = \frac{\text{True Positives}}{\text{Total *Actual* Positives}}$$

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.

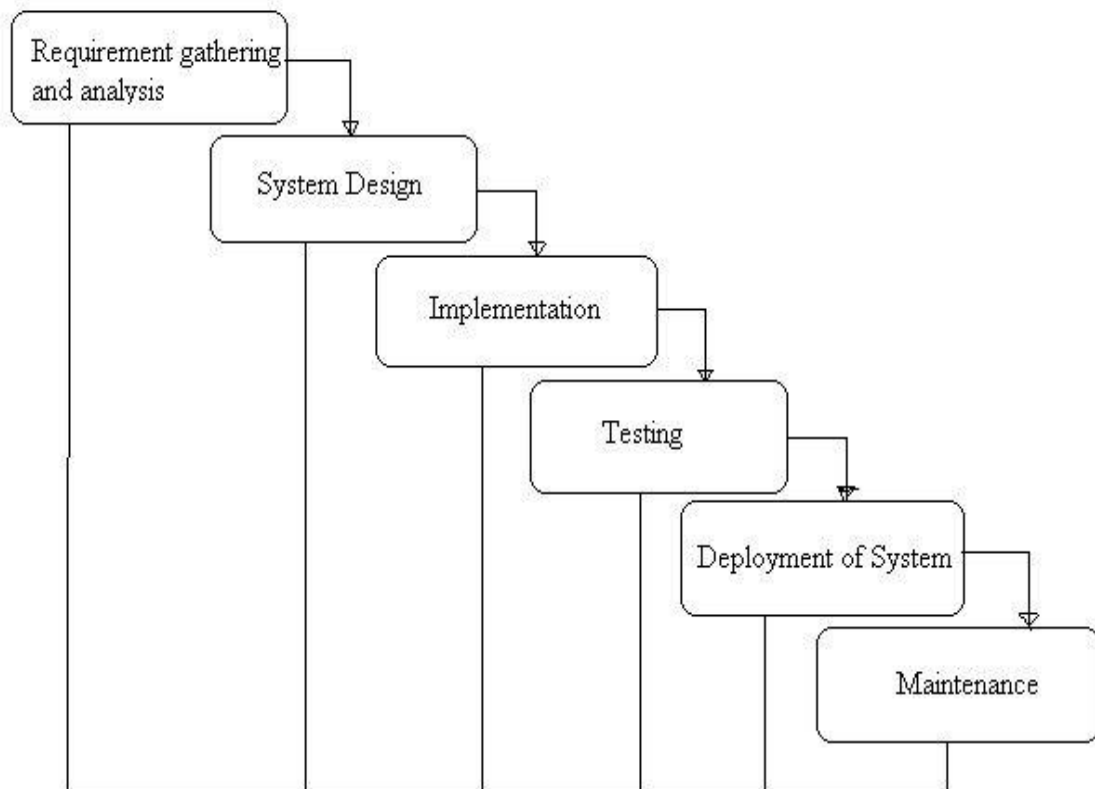
Chapter 5

PLANNING AND ESTIMATION

Software development Life Cycle

The entire project spanned for duration of 6 months. In order to effectively design and develop a cost-effective model the Waterfall model was practiced.

General Overview of "Waterfall Model"



Requirement gathering and Analysis phase:

This phase started at the beginning of our project, we had formed groups and modularized the project. Important points of consideration were

1. Define and visualize all the objectives clearly.

2. Gather requirements and evaluate them

Consider the technical requirements needed and then collect technical specifications of

various peripheral components (Hardware) required.

3. Analyze the coding languages needed for the project.

4. Define coding strategies.

5. Analyze future risks / problems.








6. Define strategies to avoid this risks else define alternate solutions to this risks.


7. Check financial feasibility.

8. Define Gantt charts and assign time span for each phase.

TimeLine

Please make changes as per your requirement

Task Name	ID	Start Date	Finish Date	Duration	30/07/2018 To 19/08/2018	19/08 / To 26/08 /18	27/08/2018 To 23/09/2018	24/08/2018 To 07/10/2018	08/10 To 15/10	08/10 To 15/10	08/10 To 15/10
Requirement Gathering	1	29/07/18	19/08/18	3 Weeks							
Problem Definition	2	12/08/18	26/08/18	1 Week							
Literature Survey	3	19/08/18	02/09/18	4 Weeks							
Analysis	4	02/09/18	02/09/18	2 Week							
Flow Chart	5	16/09/18	02/09/18	1 Week							
Block Diagram	6	30/09/18	07/10/18	2 weeks							
H/W	7			1 week							

Specifi cation		07/10/18	07/10/18								
S/W Specifi cation	8	07/10/18	07/10/18	1 week							

Chapter 7

Design & Implementation

System flowchart:

A flowchart is a graphical representation of an algorithm or process that uses a series of boxes and arrows to demonstrate the relationship between each step in the algorithm or process and the next. This graphical depiction shows how to address the issue at hand. These boxes and arrows describe processes; the operations themselves are suggested by the order in which they occur. Flowcharts have many applications and may be used to help with many different aspects of a process's or program's analysis, design, documentation, or management.

Start and end symbols

Represented as circles, ovals, or rounded (fillet) rectangles, these buttons typically feature the word "Start" or "End" or another phrase that indicates the beginning or conclusion of a process, such as "submit inquiry" or "receive product."

Arrows

providing evidence of "flow of control" An arrow that starts at one symbol and terminates at another symbol indicates that control has been transferred to the symbol that the arrow is pointing to. You have the option of using a solid or dashed line for the arrow. It is possible for the meaning of the arrow that has a dashed line to change from one flowchart to another; this is something that may be described in the legend.

Generic processing steps

Represented as rectangles Examples: "Add 1 to X"; "replace identified part"; "save changes" or similar.

Subroutines

Represented as rectangles with double-struck vertical edges; these are used to show complex processing steps which may be detailed in a separate flowchart. Example: PROCESS-FILES. One subroutine may have multiple distinct entry points or exit flows (see co routine); if so, these are shown as labeled 'wells' in the rectangle, and control arrows connect to these 'wells'.

Input/output

Presented in the form of a parallelogram Examples: Ask the user for X; provide X to the user Prepare conditional Conveyed in the form of a hexagon Displays activities that provide no use other than to set a value for a later step that is either conditional or decisional in nature (see below).

Conditional or decision

Presented in the form of a diamond (rhombus) to indicate the presence of a decision-making prompt, most frequently a Yes/No question or a True/False test. The unusual thing about the conditional symbol is that it has two arrows going out of it, often from the bottom point and the right point. One of the arrows corresponds to yes, which stands for true, and the other arrow corresponds to no, which stands for false. (The arrows ought to have labels attached to them at all times.) It is possible to use more than two arrows, however doing so is typically a clear indication that a complicated decision is being made. In this instance, the decision may need to be further broken down or replaced with the "pre-defined procedure" sign.

Junction symbol

Typically shown as a dark blob, this element indicates the point at which many different control flows combine to form a single exit flow. There will be several arrows leading into a junction sign, but there will only be one leading out of the symbol. In straightforward situations, one may just have an arrow point to another arrow instead of doing anything else. These are helpful when representing an iterative process, which is referred to as a loop in the field of computer science. One example of a loop has a connector

that serves as the point at which control is introduced for the first time, many processing stages, and a conditional that has one arrow leading out of the loop and one leading back to the connection. When making a drawing, if two lines happen to cross each other inadvertently, one of the lines might be drawn with a little semicircle around the other line to indicate that the junction was not intended. This provides further clarity.

Labeled connectors

characterised by an identifying label placed within a circle for presentation purposes. In more complicated or multi-sheet designs, labelled connections are sometimes used as an alternative to arrows. Although there can be any number of "inflow" connectors for a given label, the "outflow" connector is required to have a one-of-a-kind identifier at all times. In this particular instance, there is an implication of a junction in the control flow.

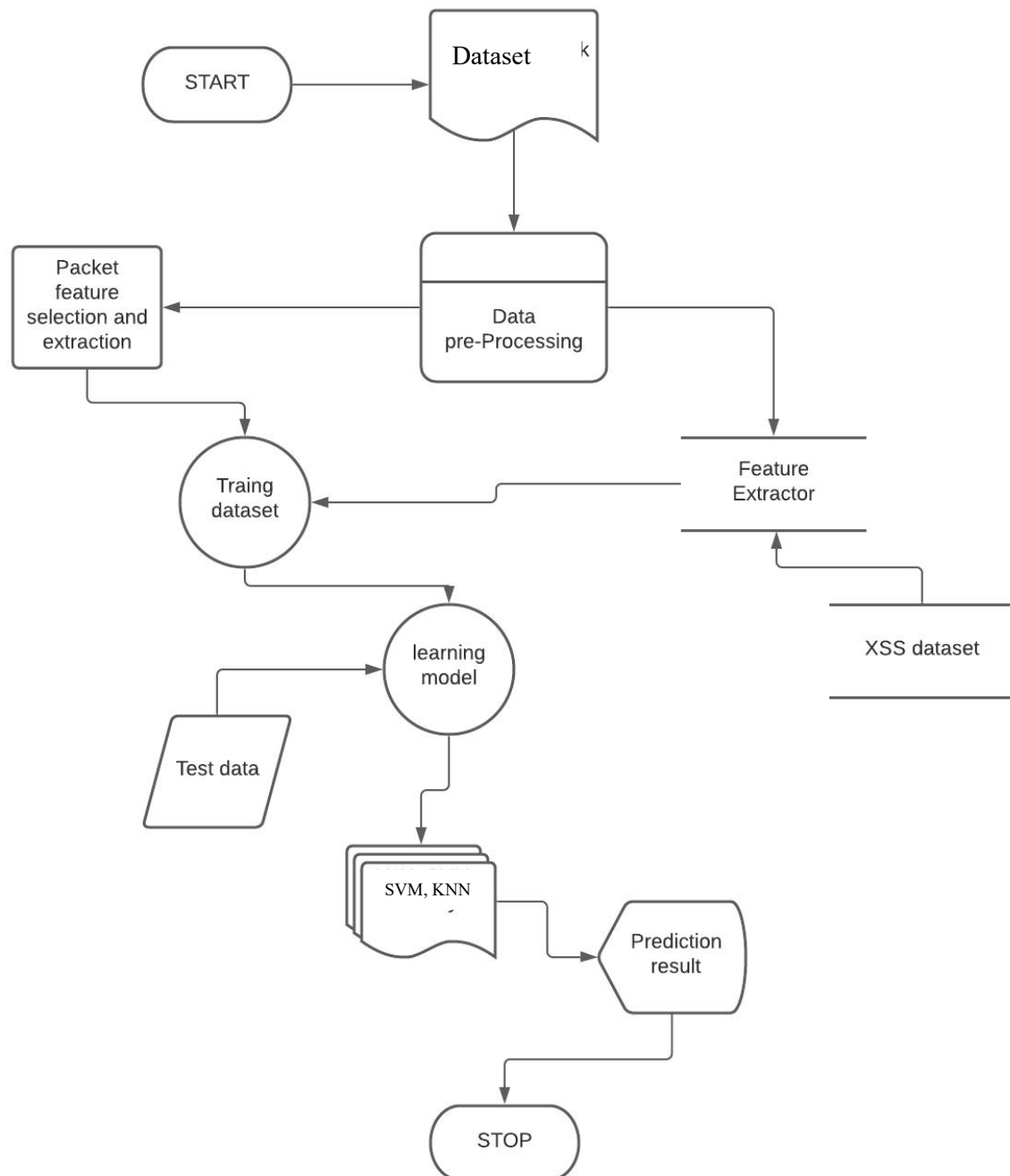
Concurrency symbol

Symbolized by a double horizontal line with an unspecified number of entry and exit arrows on either side. These symbols are utilised in situations in which two or more control flows need simultaneous operation. When all of the entrance flows have arrived at the concurrency symbol, the activation of the exit flows happens at the same time. A fork is a type of concurrency symbol that only has one flow entering it, whereas a join only has one flow leaving it. It is essential to bear in mind the significance of maintaining these linkages in a logical order. The direction of flow for all processes should be from left to right and top to bottom.

Flow chart:

Flowcharts are graphical representations of activities and action processes with support for choice, iteration, and concurrency. Flowcharts may be used to map out a series of steps in sequential order. Flowcharts may be utilised in the Unified Modeling Language to provide a step-by-step description of the business and operational processes that are associated with the components of a system. The activity diagram illustrates the control flow in its entirety. The overall process of control is depicted as a flowchart. Flowcharts are diagrams that outline processes and are made up of a constrained set of objects linked by arrows. Flowcharts are diagrams that outline processes and are made up of a constrained set of objects linked by arrows. The most common kinds of forms are as follows: the rectangle stands for the flow. The diamonds represent different options or choices. The split and join points for activities

that are happening at the same time are shown by the bars. The beginning (or starting state) of the workflow is represented by a rectangle in this diagram. The finish line is represented by a last rectangle (final state). The direction that the arrows are pointing in, which is from the beginning to the end, is meant to symbolise the sequence in which the activities occur.



SYSTEM IMPLEMENTATION:

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

IMPLEMENTATION METHODS:

There are a few different approaches that may be used to manage the transition from the older computerised 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 computerised 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 computerised 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.

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 strategy for implementation need 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.

DFD

A data flow diagram, often known as a data flow chart or DFD, is a graphical depiction of the movement of data through an information system. The visualisation 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 modelled.

Symbols:

The four components of a data flow diagram (DFD) are:

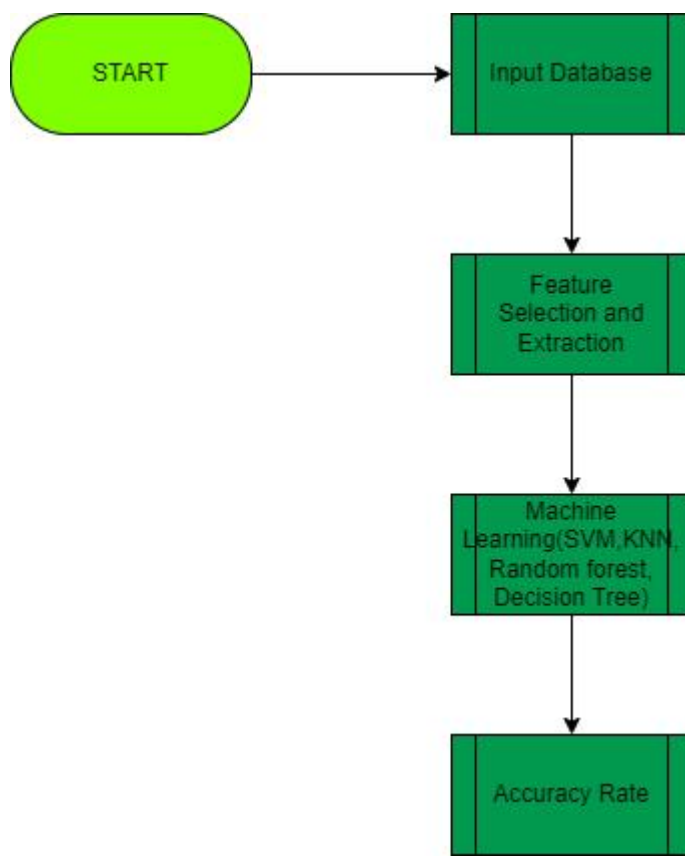
Entities and Terminators that are external to the system being represented are referred to as external. The origins and destinations of information are denoted by the corresponding terminators. When it comes to the design of a system, we have no understanding of the tasks that these terminators do or the process by which they perform them.

During the process of developing the outputs, processes make adjustments to the inputs.

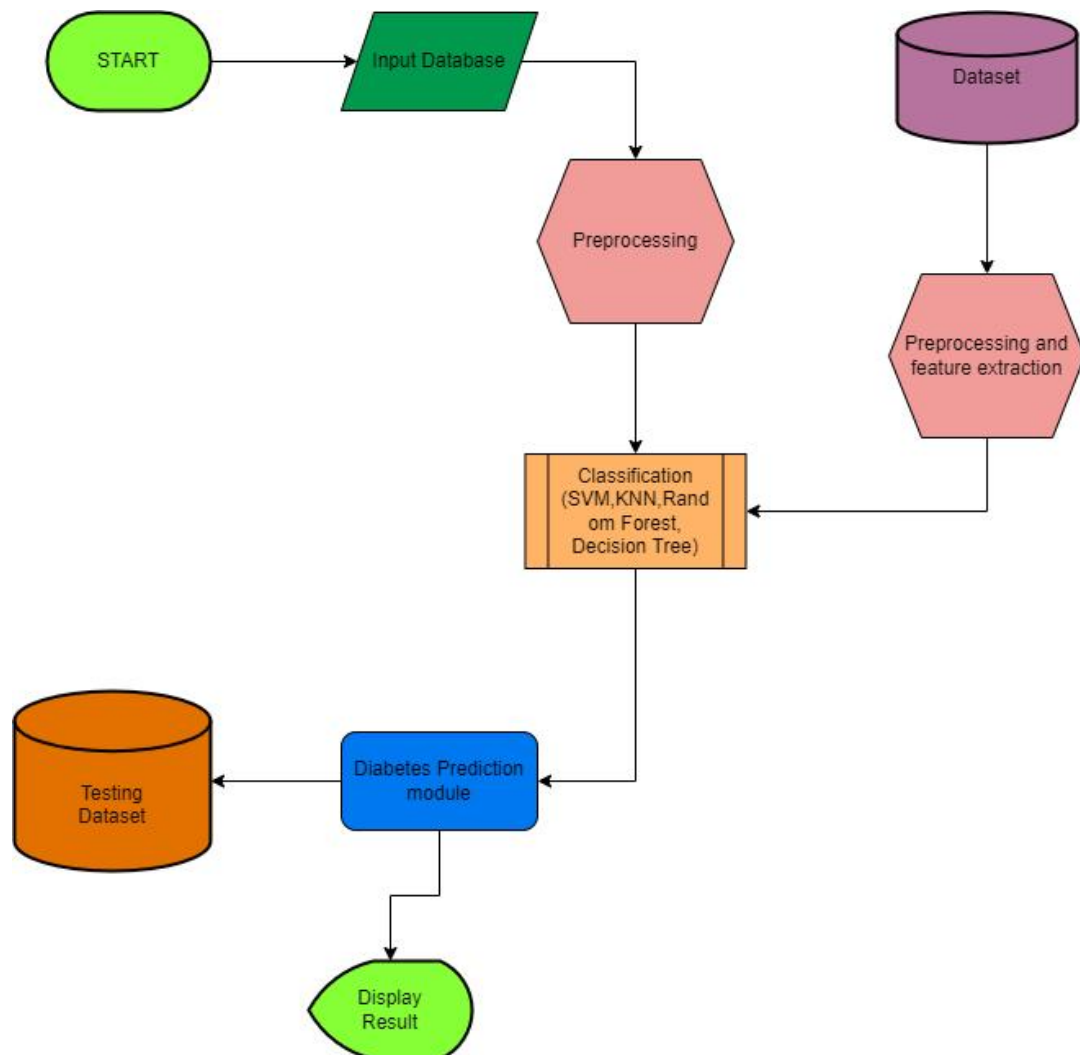
Data Stores are locations in the process where information is allowed to settle down for the night. A DFD does not say anything about the relative time of the processes,

therefore a data store may be a location to aggregate data over the course of a year for the yearly accounting process. This would be necessary since a DFD does not indicate anything about the relative timing of the activities.

DFD level 1:



DFD level 2



E-R Diagram:

The Entity Relational Model, sometimes known as ER, is a conceptual statistics version diagram with a high degree of detail. The belief in real-world entities and the relationships that exist between them underpins the entirety of the entity-dating version. Entity relationship diagrams, also known as entity dating diagrams (ER), are a type of flowchart that demonstrate the ways in which "entities" such as people, things, and ideas are related with one another inside a system. ER diagrams are similar to statistical shape diagrams (DSDs), which focus on the relationships of variables inside entities rather than the interactions between the entities themselves. ER diagrams are related with these types of diagrams. In the process of developing a logical database schema, ER modelling is regarded as a strategy that is considered to be comprehensive. This is incorrect due to the fact that the ER diagram is only a rough representation of the data. This description was developed by a completely subjective appraisal of the facts that were gathered during the requirements analysis. Entities, relationships, and attributes are the building blocks that go into constructing ER diagrams. In addition to this, they are what makes up the cardinality, which is what characterises relationships using numerical terms.

Entity

An object or component of the data is referred to as an entity. In an er diagram, each item, such as academics and educational institutions, is shown as a rectangle. As a result of the fact that most college students apply to just one school, college students engage in what is known as a "many-to-one courting." A susceptible entity is an entity that cannot be individually identified based on its own characteristics and instead relies on a courtship with any other entity in order to obtain a diagnosis. A pair of intersecting rectangles serves as a visual representation of the vulnerable entity.

Attribute

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

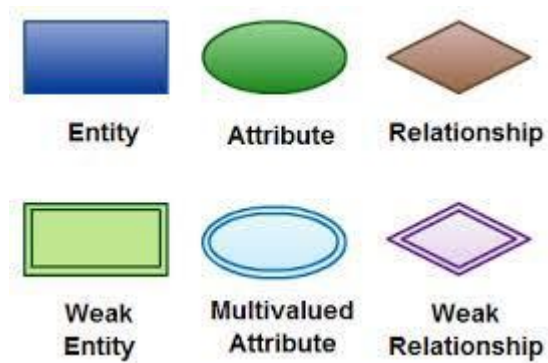
1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

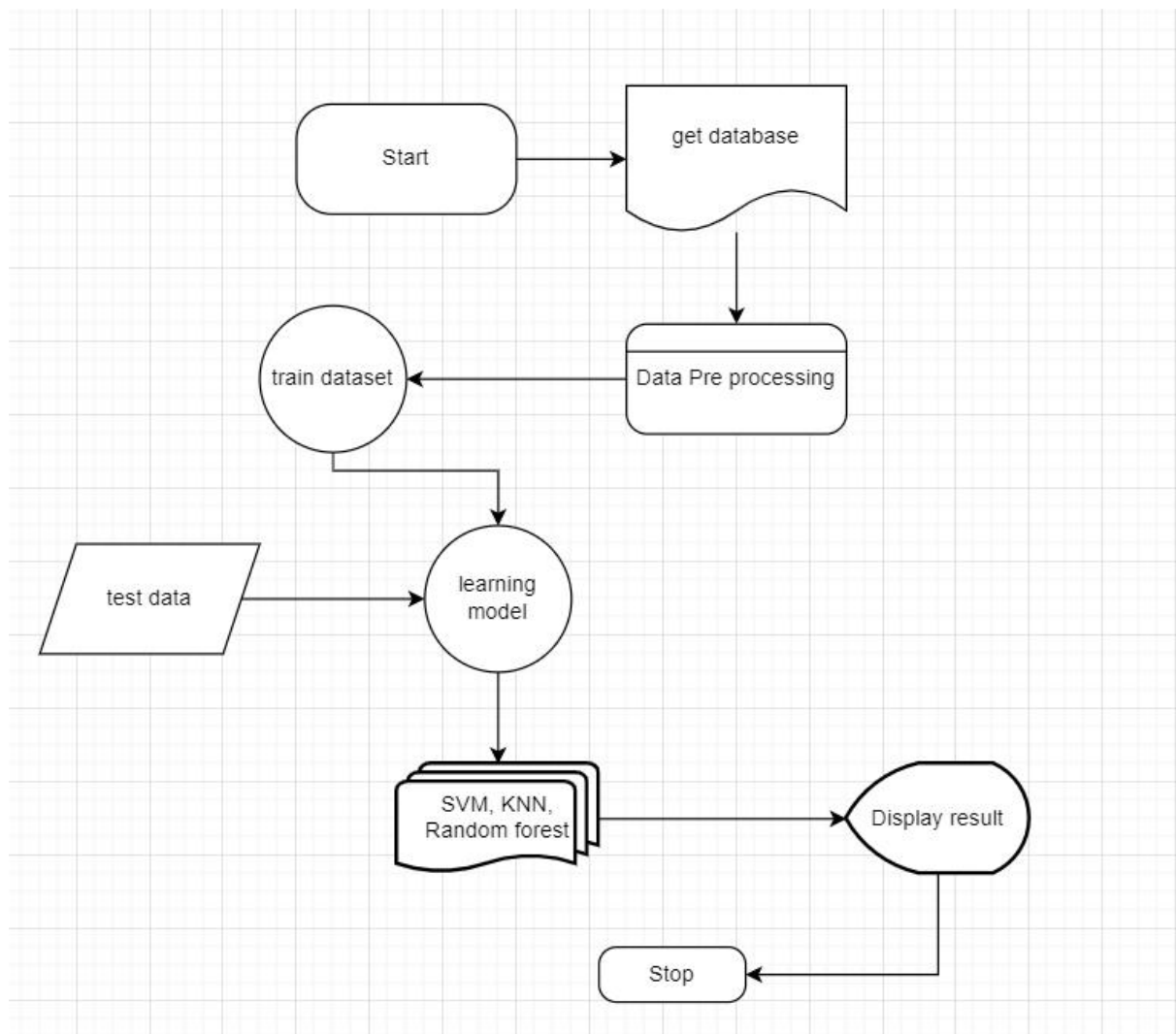
Relationship

A relationship is represented by diamond shape in the ER diagram, it shows the relationship among entities. There are four types of relationships:

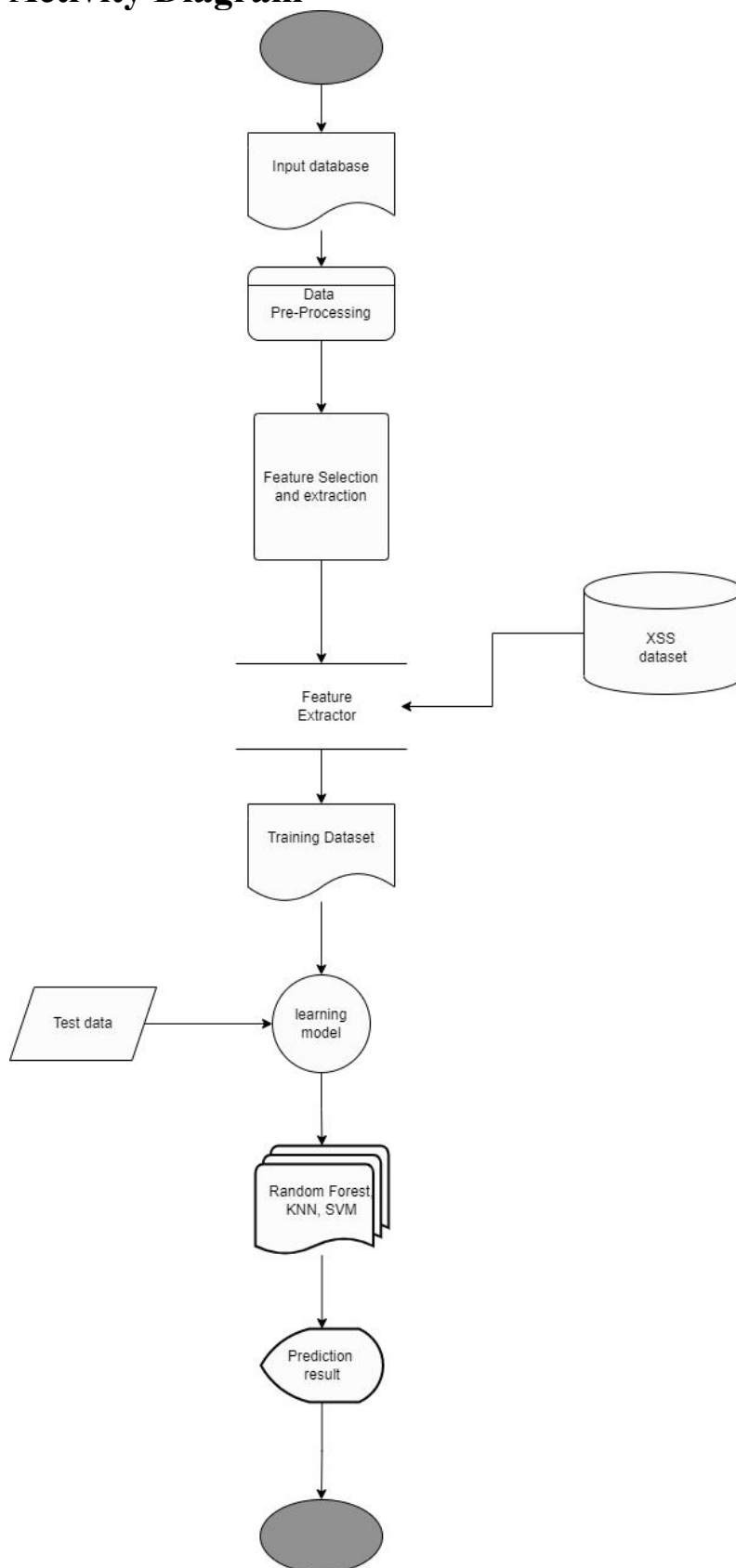
1. One to One
2. One to Many
3. Many to One
4. Many to Many

ER Diagram Symbols

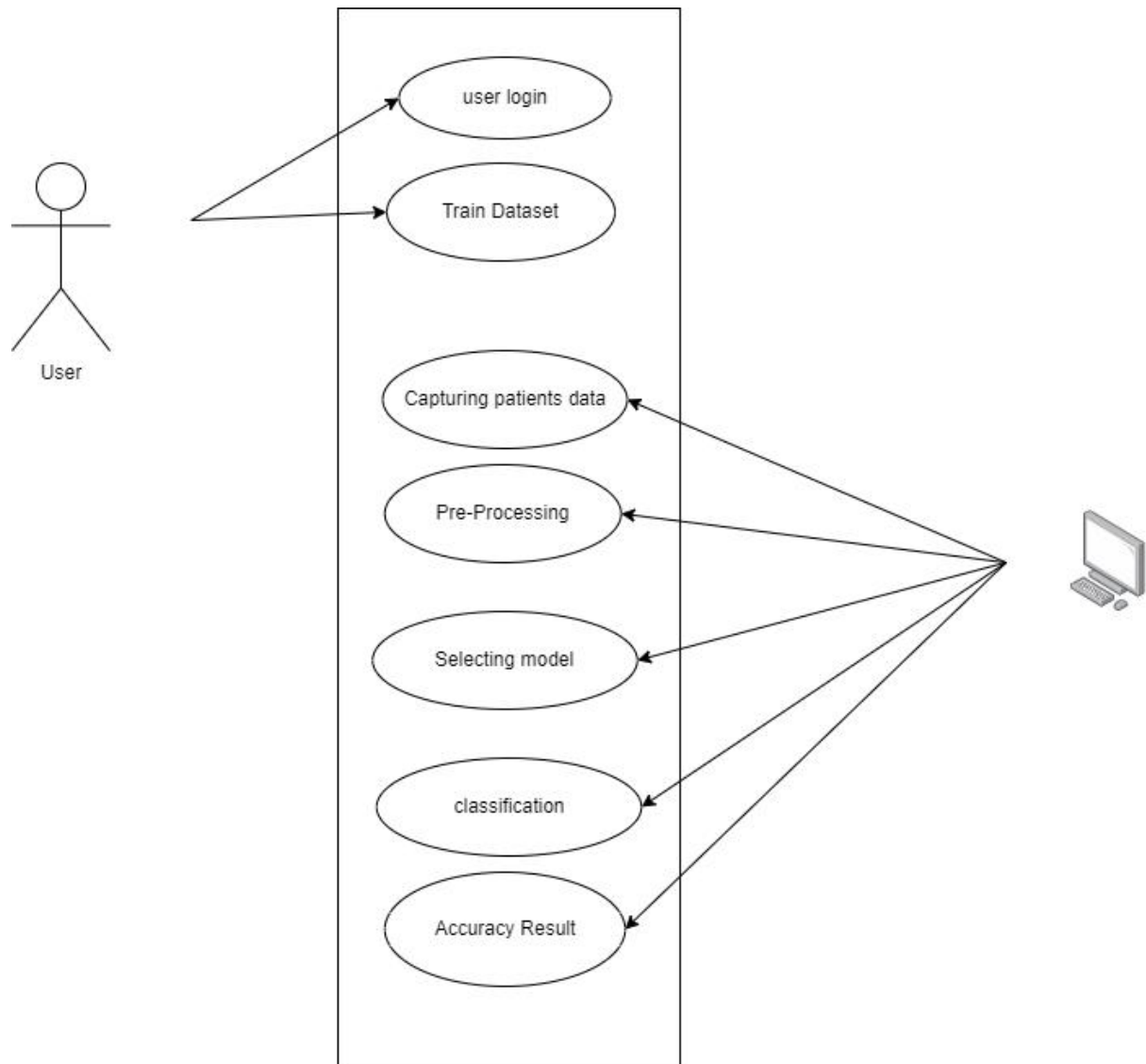




Activity Diagram



Use Case Diagram :-



Chapter 8

Advantages

Advantages:

We show that device mastering strategies employ the Random Forest approach to pick and extract capabilities to identify community visitors, and we accomplish this by reading the outcomes to find more community traffic and reduce the burden. SVM is used to categorise record sets. We categorise 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 visualisation, and data interpretation.

Chapter 09

CONCLUSION

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.

Chapter 10

BIBLIOGRAPHY

References

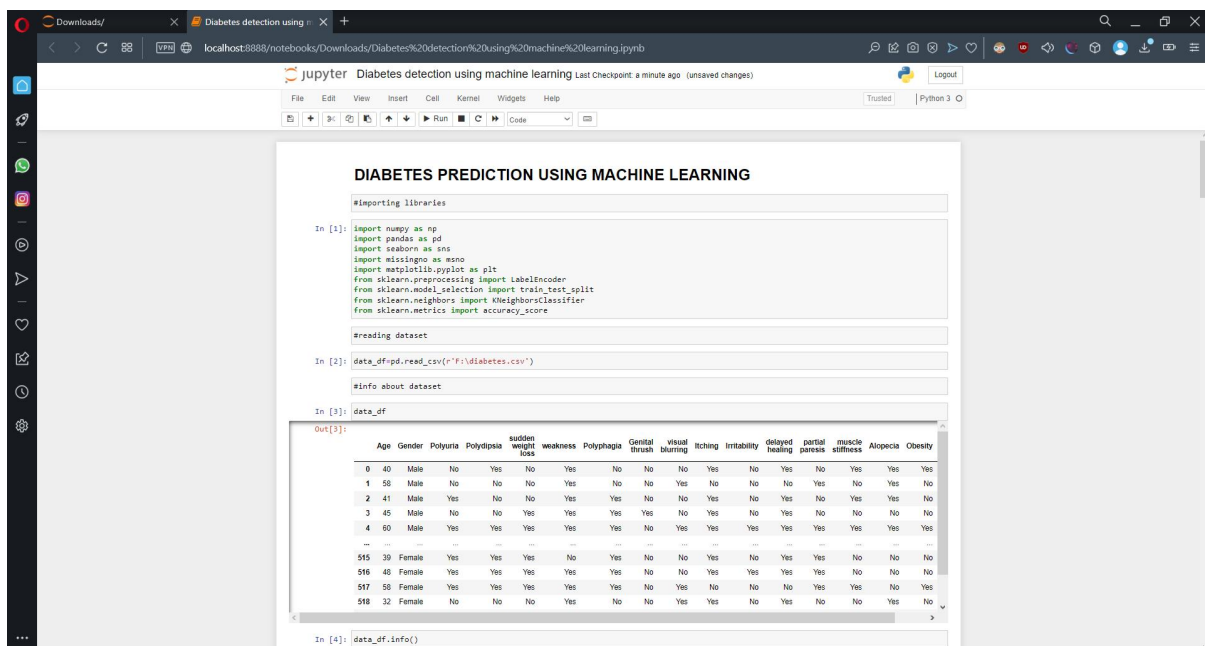
- [1] “Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus” Md Faisal Faruque, Asaduzzaman, Iqbal H. Sarker, IEEE 2019.
- [2] “A Comprehensive Exploration to the Machine Learning Techniques for Diabetes Identification” Sidong Wei1, Xuejiao Zhao, Chunyan Miao Shanghai Jiao Tong University, China.
- [3] “Association Rule Extraction from Medical Transcripts of Diabetic Patients” Lakshmi K S, G Santhosh Kumar, 2014.
- [4] “Diabetes Care Decision Support System” 2nd International Conference on Industrial and Information Systems IEEE 2010.
- [5] “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.
- [6] “Machine Learning Techniques for Classification of Diabetes and Cardiovascular Diseases” by BerinaAlic, Lejla Gurbeta, IEEE 2017.
- [7] “Performance Analysis of Classification Approaches for the Prediction of Type II Diabetes” by M. Durgadevi, M. Durgadevi, IEEE 2017.
- [8] “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
- [10] P. Suresh Kumar and V. Umatejaswi, "Diagnosing Diabetes using Data Mining Techniques", International Journal of Scientific and Research Publications, Volume 7, Issue 6, June 2017 705 ISSN 2250-3153.
- [11] "Clustering Medical Data to Predict the Likelihood of Diseases" by Razan Paul, Abu Sayed Md. Latiful Hoque, IEEE 2010.
- [12] "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.
- [13] Anjali C And Veena Vijayan V, Prediction and Diagnosis of Diabetes Mellitus, "A Machine Learning Approach" ,2015 IEEE in Intelligent Computational Systems (RAICS) | Trivandrum.
- [14] 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.
- [15] Dr. M. Renuka Devi and J. Maria Shyla, "Analysis of Various Data Mining Techniques to Predict Diabetes Mellitus", International Journal ISSN 0973-4562 Volume 11, Number 1 (2016) pp 727-730
- [16] 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.
- [17] 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.
- [18] B. Nithya and Dr. V. Ilango," Predictive Analytics in Health Care Using Machine Learning Tools and Techniques", International Conference on Intelligent Computing and Control Systems, 978-1-5386-2745-7, 2017.
- [19] 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.

- [20] Aiswarya Iyer, S. Jeyalatha and Ronak Sumbaly,” Diagnosis of Diabetes Using Classification Mining Techniques”, International Journal of Data Mining & Knowledge Management Process (IJDMP) Vol.5, No.1, January 2015.
- [21] 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.
- [22] 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.
- [23] 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 the Diabetes and Heart Disease”, Expert Systems with Applications: An International Journal, Volume 35 Issue 1-2, July, 2008.
- [25] 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.
- [26] Dost Muhammad Khan¹, Nawaz Mohamudally², “An Integration of K-means and Decision Tree (ID3) towards a more Efficient Data Mining Algorithm ”, Journal Of Computing, Volume 3, Issue 12, December 2011

Chapter 11

SCREENSHOTS



The screenshot shows a Jupyter Notebook interface with the title "Diabetes detection using machine learning". The notebook contains the following code cells:

```
#importing libraries

In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import missingno as msno
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

#reading dataset

In [2]: data_df=pd.read_csv(r"F:\diabetes.csv")

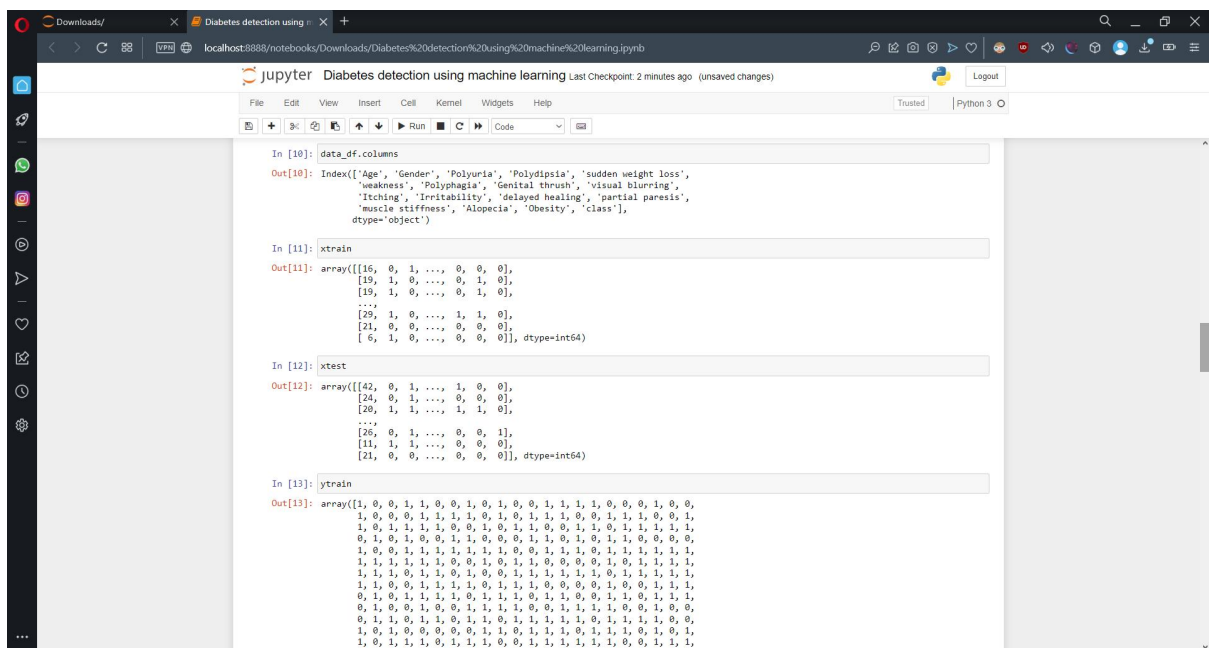
#info about dataset

In [3]: data_df

Out[3]:
```

	Age	Gender	Polyuria	Polydipsia	Sudden weight loss	Weakness	Polyphagia	Genital thrush	Visual blurring	Itching	Irritability	Delayed healing	Partial paresis	Muscle stiffness	Alopecia	Obesity
0	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes
1	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
...
515	39	Female	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No	No	No
516	48	Female	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No
517	58	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	Yes	No	Yes
518	32	Female	No	No	No	Yes	No	No	Yes	Yes	No	Yes	No	No	Yes	No

```
In [4]: data_df.info()
```



```

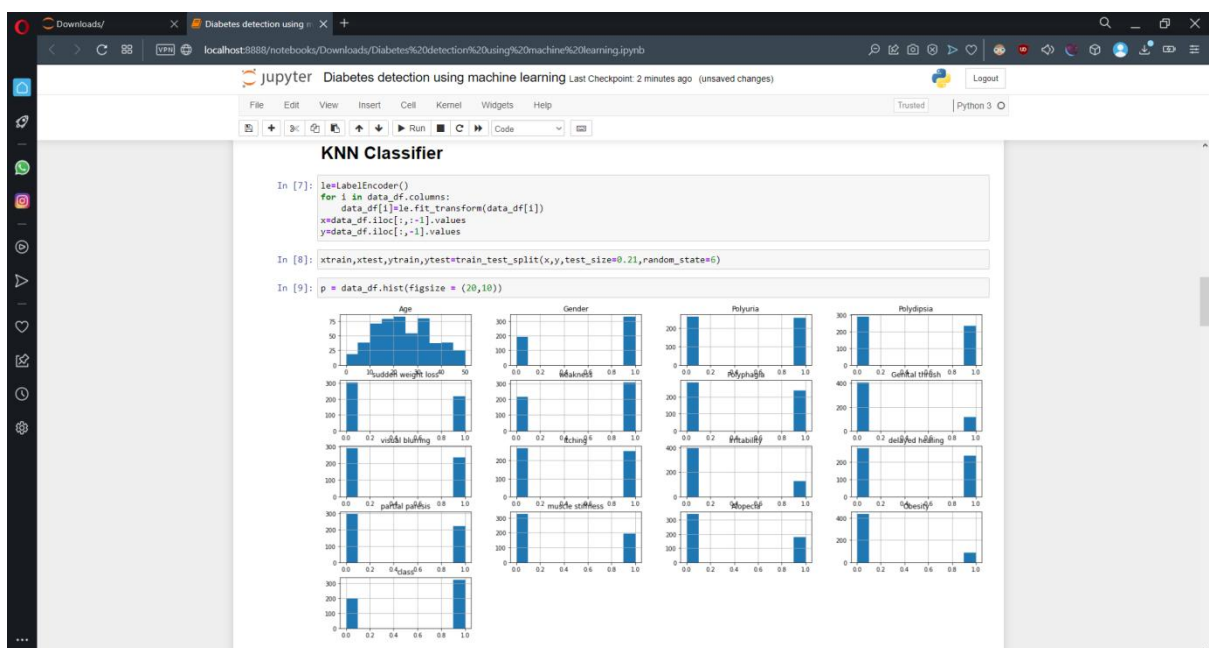
jupyter Diabetes detection using machine learning Last Checkpoint: 2 minutes ago (autosaved)

Out[12]: array([[42, 0, 1, ..., 1, 0, 0],
               [24, 0, 1, ..., 0, 0, 0],
               [20, 1, 1, ..., 1, 1, 0],
               ...,
               [26, 0, 1, ..., 0, 0, 1],
               [11, 1, 1, ..., 0, 0, 0],
               [21, 0, 0, ..., 0, 0, 0]]) dtype=int64

In [13]: ytrain
Out[13]: array([1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0,
               1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1,
               1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1,
               0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
               1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1,
               1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
               1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
               1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
               0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
               0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1,
               0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
               1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
               1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
               0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1,
               0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1,
               1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1])

In [14]: ytest
Out[14]: array([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
               0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1,
               0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
               1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
               0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0,
               0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1,
               0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1,
               1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1,
               0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1])

```



```

In [15]: modelknn=KNeighborsClassifier(n_neighbors=1)
          modelknn.fit(xtrain,ytrain)
          ypredknn=modelknn.predict(xtest)

In [16]: print('KNN = ',accuracy_score(ytest,ypredknn)*100)

KNN = 94.54545454545455

```


Classification report and confusion matrix of KNN model

```
In [19]: from sklearn.metrics import classification_report, confusion_matrix
```

```
print(confusion_matrix(ytest, predictions))
print(classification_report(ytest, predictions))
```

```
[[45  1]
 [ 0 64]]
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	46
1	0.98	1.00	0.99	64
accuracy			0.99	110
macro avg	0.99	0.99	0.99	110
weighted avg	0.99	0.99	0.99	110

Using Random Forest

```
In [17]: from sklearn.ensemble import RandomForestClassifier
```

```
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(xtrain, ytrain)
```

```
Out[17]: RandomForestClassifier
```

```
RandomForestClassifier(n_estimators=200)
```

```
In [18]: from sklearn import metrics
```

```
predictions = rfc.predict(xtest)
print("Accuracy Score =", format(metrics.accuracy_score(ytest, predictions)*100))
```

```
Accuracy Score = 99.0909090909091
```

Classification report and confusion matrix of random forest model

```
In [20]: from sklearn.metrics import classification_report, confusion_matrix
```

```
print(confusion_matrix(ytest, predictions))
print(classification_report(ytest, predictions))
```

```
[[45  1]
 [ 0 64]]
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	46
1	0.98	1.00	0.99	64
accuracy			0.99	110
macro avg	0.99	0.99	0.99	110
weighted avg	0.99	0.99	0.99	110

Downloads/ Diabetes detection using X +

localhost:8888/notebooks/Downloads/Diabetes%20detection%20using%20machine%20learning.ipynb

jupyter Diabetes detection using machine learning Last Checkpoint: 4 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Decision Tree

```
In [21]: from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()
dtree.fit(xtrain, ytrain)
```

Out[21]:

```
> DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [22]: from sklearn import metrics

predictions = dtree.predict(xtest)
print("Accuracy Score =", format(metrics.accuracy_score(ytest,predictions)*100))

Accuracy Score = 96.36363636363636
```

Classification report and confusion matrix of the decision tree model

```
In [23]: from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(ytest, predictions))
print(classification_report(ytest,predictions))
```

```
[[43  3]
 [ 1 63]]
```

	precision	recall	f1-score	support
0	0.98	0.93	0.96	46
1	0.95	0.98	0.97	64
accuracy			0.96	110
macro avg	0.97	0.96	0.96	110
weighted avg	0.96	0.96	0.96	110

Downloads/ Diabetes detection using X +

localhost:8888/notebooks/Downloads/Diabetes%20detection%20using%20machine%20learning.ipynb

jupyter Diabetes detection using machine learning Last Checkpoint: 4 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Decision Tree

```
In [21]: from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()
dtree.fit(xtrain, ytrain)
```

Out[21]:

```
> DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [22]: from sklearn import metrics

predictions = dtree.predict(xtest)
print("Accuracy Score =", format(metrics.accuracy_score(ytest,predictions)*100))

Accuracy Score = 96.36363636363636
```

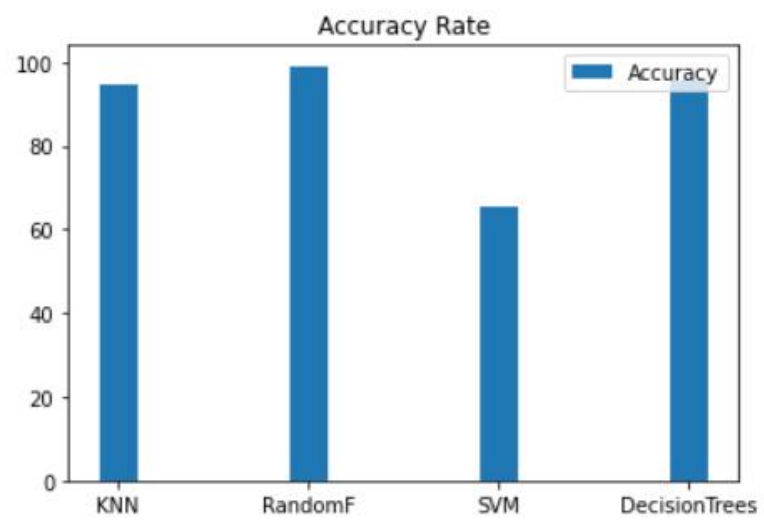
Classification report and confusion matrix of the decision tree model

```
In [23]: from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(ytest, predictions))
print(classification_report(ytest,predictions))
```

```
[[43  3]
 [ 1 63]]
```

	precision	recall	f1-score	support
0	0.98	0.93	0.96	46
1	0.95	0.98	0.97	64
accuracy			0.96	110
macro avg	0.97	0.96	0.96	110
weighted avg	0.96	0.96	0.96	110



Chapter 12

SOURCE CODE

```
#importing libraries
import numpy as np
import pandas as pd
import seaborn as sns
import missingno as msno
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

#reading dataset
data_df=pd.read_csv(r'F:\diabetes.csv')

#info about dataset
data_df
data_df.info()
data_df.describe()
data_df.isnull().sum()

#KNN CLASSIFIER
le=LabelEncoder()
for i in data_df.columns:
    data_df[i]=le.fit_transform(data_df[i])
x=data_df.iloc[:, :-1].values
y=data_df.iloc[:, -1].values
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.21,random_state=6)
p = data_df.hist(figsize = (20,10))
data_df.columns
xtrain
xtest
ytrain
ytest
modelknn=KNeighborsClassifier(n_neighbors=1)
modelknn.fit(xtrain,ytrain)
ypredknn=modelknn.predict(xtest)
```

```

print('KNN = ',accuracy_score(ytest,ypredknn))

#USING RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(xtrain, ytrain)
from sklearn import metrics
predictions = rfc.predict(xtest)
print("Accuracy Score =", format(metrics.accuracy_score(ytest, predictions)))

#Classification report and confusion matrix of KNN model
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(ytest, predictions))
print(classification_report(ytest,predictions))
#Classification report and confusion matrix of random forest model
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(ytest, predictions))
print(classification_report(ytest,predictions))

#DECISION TREE
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree.fit(xtrain, ytrain)
from sklearn import metrics
predictions = dtree.predict(xtest)
print("Accuracy Score =", format(metrics.accuracy_score(ytest,predictions)))

#Classification report and confusion matrix of the decision tree model
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(ytest, predictions))
print(classification_report(ytest,predictions))

# Support Vector Machine (SVM)
from sklearn.svm import SVC
svc_model = SVC()
svc_model.fit(xtrain, ytrain)
svc_pred = svc_model.predict(xtest)
from sklearn import metrics
print("Accuracy Score =", format(metrics.accuracy_score(ytest, svc_pred)))

#Classification report and confusion matrix of the SVM classifier
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(ytest, svc_pred))
print(classification_report(ytest,svc_pred))

# CONCLUSION
# Therefore Random forest is the best model for this prediction since it has an accuracy_score
of 0.98.

#feature Importance

```

```
rfc.feature_importances_  
import pickle  
  
# Firstly we will be using the dump() function to save the model using pickle  
saved_model = pickle.dumps(rfc)  
  
# Then we will be loading that saved model  
rfc_from_pickle = pickle.loads(saved_model)  
  
# lastly, after loading that model we will use this to make predictions  
rfc_from_pickle.predict(xtest)  
data_df.head()  
data_df.tail()
```

Chapter 13

PLAGIARISM CHECK

report

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

Chapter 14

PAPER SUBMISSION PROOF

2023 INTERNATIONAL CONFERENCE ON COMPUTER
COMMUNICATION AND INFORMATICS

23-25 Jan 2023

HELLO G. PARIMALA

[Change Password](#)

Paper ID	174
Paper Title	Diabetes Prediction using Machine Learning
Author Name	G. Parimala
Author Category	Academician
Author Dept	CS
Paper Category	Data Analytics
First Author Country	India
Initial Screening Status	Accepted
Plagiarism Status	
Technical Review Status	
Reviewer Comments	Your paper has been moved to Plag Check Desk
Registration Status	
Payment Status	

IMPORTANT DATES

Nov 30

30 Nov 2022

Full Paper Submission
Deadline

Dec 15

15 Dec 2022

Notification Of Paper
Acceptance

Dec 16

16 Dec 2022

Commencement Of Conference
Registration

Jan 10

10 Jan 2023

Conference Registration
Deadline

1573 PAPERS IN IEEE XPLORE