# IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR ANALYZING DIABETES DISEASE

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

## COMPUTER SCIENCE ENGINEERING

By:

Vedika Garg (151231)

under the supervision

of

Dr. Pradeep Kumar Gupta

To

Department of Computer Science Engineering and Information Technology

**Jaypee University of Information Technology, Waknaghat,**

**Solan, Himachal Pradesh-173234**

Certificate

# Candidate’s Declaration

I hereby declare that the work presented in this report entitled “Implementation of Machine learning algorithms for analyzing diabetes disease” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of **Dr. Pradeep Kumar Gupta, Associate Professor, Computer Science/ IT.**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Vedika Garg, 151231

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pradeep Kumar Gupta

Associate Professor

Computer Science/Information Technology

Dated:

# ACKNOWLEDGMENT

Working for the “diabetes analysis” project was interesting. We got to learn about Neural Networks, Bayesian Networks and Decision Trees in Machine Learning, and to compare the accuracies and outcomes of each of these.

Special thanks to our supervisor Dr. Pradeep Kumar Gupta for his guidance and advice on this project.

Also, we are very grateful to our college, lectures, and friends where they gave us enough time to complete this report and we would like to thank all other people who supported us in our project.

Thank you.

# TABLE OF CONTENTS

# LIST OF FIGURES

# List of Tables

# ABSTRACT

Diabetes has affected over 246 million people worldwide with a majority of them being women. According to the WHO report, by 2025 this number is expected to rise to over 380 million. The disease has been named the fifth deadliest disease in the United States with no imminent cure in sight. With the rise of information technology and its continued advent into the medical and healthcare sector, the cases of diabetes, as well as their symptoms, are well documented.

Our report focuses on quicker and efficient techniques for the diagnosis of diabetes disease. Traditional techniques like Random Forest Classifier had a few limitations in predicting the outcome of the disease. So we try to implement more advanced machine learning algorithms in our research like Naive Bayes, k-nearest neighbor and Decision Trees which give better predictions and increased accuracy. Along with these algorithms, we will be implementing many other algorithms too and be doing a comprehensive and a comparative study on these algorithms to get the best of all. This study can further help in determining which algorithms to use.

# Chapter 1
# Introduction

## 1.1 Introduction

Impacts of diabetes are accounted to have deadly and declining sway on ladies as compared to men on account of the lesser survival rate and lacking way of living. World Health Organization study expresses nearly about 1/3rd ladies who experience the ill effects of diabetes have got no information related to it. The impact of diabetes is one of a kind if there should be an occurrence of moms in light of the fact that the ailment is transferred to their yet to be born kids. Attacks, unnatural birth cycles, visual impairment, kidney disappointment, and removals are only a portion of the complexities that emerge through this ailment.

A human is viewed as experiencing diabetes when glucose levels are more than typically 4.4 - 6.1 millimole per liter. Pancreas existing in a person’s body secrete a hormone which is dependable to enable glucose to achieve every cell within the system. A person suffering from diabetes basically has lesser secretion of insulin or their system can’t utilize the insulin efficiently. The 3 primary kinds of diabetes are type 1, type 2 and gestational diabetes.

These days, an expansive measure of data is gathered as sufferer’s record by the healing centers. Learning revelation for prescient reasons for existing is conducted via information mining that is an examination system which aids in advising inductions. The technique aids in basic leadership via calculations through a lot of information created by the medicinal focuses. Bearing in mind the significance of initial medicinal conclusion of the illness, information mining systems are connected to support the ladies in the identification of diabetes at a beginning period as well as to conduct that might aid for maintaining a strategic distance from difficulties.

Figure 1 Diabetes

## About diabetes

Diabetes is an infection which happens when the insulin creation in the human system is insufficient or system can't utilize the secreted insulin in an efficient way, thus, it prompts increasing levels of sugar in the blood. The system units separate the sustenance into sugar but the sugar should be conveyed to each unit of the system. Insulin is a secretion which coordinates the sugar delivered by separating the sustenance within the system units. An adjustment for the creation of this hormone prompts an increment in the glucose intensities which can prompt harm in the external and internal body parts including the tissues.

## Diabetes classification:

Diabetes can be classified into 3 kinds as portrayed beneath:

a. Type I: Although just 10% of people suffering from diabetes are accounted to be having this kind of diabetes, as of late, there is an ascent in the number of instances of this sort in the US. This illness shows an immune system sickness happening at an extremely youthful period of beneath 20 yrs. consequently likewise known as adolescent beginning diabetes. Within this kind of diabetes, the units of the pancreas which secrete insulin tend to be demolished by the safeguard arrangement in the human system. Infusions of insulin alongside regular blood check-up and diet limitations need to be trained by people experiencing this kind of diabetes.

b. Type II: About 90% of individuals suffer from this kind of diabetes and are regularly known as the grown-up beginning diabetes, else non-insulin subordinate diabetes. For this situation, the different parts present in the human system move toward becoming insulin safe, which expands the interest of insulin. In such a case, pancreatic cells are unable to secrete an appropriate measure of insulin. For keeping such a sort of diabetes under control, these people suffering need to pursue a severe eating regimen, do physical exertion daily as well as monitor the level of sugar. Having more than what is normal weight or not doing any physical exercise are some of the causes of having this kind of illness.

c. Gestational diabetes: Pregnant ladies are susceptible to such a kind of diabetes because of the increasing level of glucose since pancreatic units aren’t able to create an adequate measure of insulin hormone. Poor management of healing this disease may prompt complexities while the delivery of the child. Taking precautions and proper care of the eating routine as well as consuming insulin hormone may reduce the symptoms of such a type of diabetes.

**Symptoms**: Some of the signs of diabetes include: the repeated urge to urinate, craving to eat very frequently, feeling thirsty at short intervals of time, excessive increase or decrease in weight, injuries are not healed quickly, an eyesight impaired, tired feeling, and many more.

**Diagnosis**: Regular check-up of urine samples and blood can be ways of diagnosis of diabetes since these can check the amount of sugar present in the body. Tests like OGTT, fasting sugar and AIC may also be carried out for detection of diabetes.

**Treatment**: Type I as well as Type II diabetes may never be completely healed but the effects and a further worsening of the disease might be reduced using measures like exercising daily, injecting or consuming insulin hormone and taking proper precautions in food plus drinks consumption. Worsening of the disease may lead to the removal of the foot, neural treatments, loss of vision, higher chances of kidney impairment, cardiac arrests and many more.

## 1.2 Problem statement

The issue which we would be addressing in this project report is how to analyze diabetes disease at an initial stage of development.

Given a dataset having more than 600 entries and 8 attributes namely
Pregnancies,
Glucose,
Blood pressure,
Skin thickness,
Insulin,
BMI,
Diabetes pedigree function, and
Age,
We need to analyze each attribute as well as implement various machine learning algorithms and approaches to this data.

## 1.3 Objective

Our aim is to implement various machine learning algorithms like k-Neural Networks, Naïve Bayes Classifier, Support Vector Machines, Random Forest Classifier, Logistic Regression, Gradient Boosting, and Decision Trees for the analysis of diabetes disease.

We focus on considering each attribute in our dataset and applying the above-mentioned algorithms for the prediction of the outcome. In this manner, we would be able to analyze which attribute mostly plays a significant role in the determination of diabetes. Also what minimum value of that attribute would result in a positive outcome?

Moreover, this project would help us acknowledge that using which algorithm we can predict the outcome with maximum accuracy and precision. By this, we can tell the pros and cons of each algorithm used.

We will be carrying out a comprehensive and comparative study by analysing the results of each algorithm to determine which algorithm can be the best fit for an initial prediction of the occurrence of Diabetes in a patient.

## 1.4 Methodologies

### Machine learning

Machine learning is utilization of man-made consciousness (computer-based intelligence) which gives frameworks the capacity to consequently take in and enhance for a fact by avoiding the unequivocal customization. ML centers around the improvement of PC programming which may get to information as well as utilize the data absorb from itself.

One of the ways toward knowledge starts through perceptions or information, for instance, precedents, coordinate involvement, or guidance, with the end goal to search for instances in information and settle on better choices later on dependent on the models which we give. The essential objective lies within permitting the PCs grasp naturally avoiding a person's mediation or help and alter activities in like manner.

Algorithms used for the analysis of diabetes are:

### 1.4.1 **K-Nearest Neighbor**:

The KNN technique refers to a way to deal with information order which gauges how possibly an information idea is to be an individual from one gathering or another relying upon which cluster the information guides closest toward it is present in.

This technique is referred to as a lazy approach, implying it won't assemble a structure utilizing the train set unless an inquiry of the informational collection has been accomplished.
The flowchart depicts the general approach in implementing the K- nearest neighbours algorithm.

It is also called a non-parametric approach. This implies that this technique doesn't propose any suppositions within hidden information dissemination that is the arrangement is resolved through the information. Subsequently, this scheme is utilized if we initially have no learning related to the data.

### 1.4.2 **Naïve Bayes Classifier**:

This classification technique utilizes likelihood hypothesis for characterization of information. This approach works on the basis of Bayes' hypothesis. The primary knowledge of Bayes' hypothesis states: the likelihood of an occasion may be balanced as fresh information is presented.

This classifier is called naïve because of its presumption that each characteristic of information under thought are autonomous of one another.

This is a group of ML schemes that utilize statistics autonomy. This technique is moderately simple to compose as wells as execute more proficient than other Bayes' techniques.

### 1.4.3 **Support Vector Machines**:

This scheme refers to a directed ML technique that may be utilized for both regression and classification algorithms. But, this is largely applied to classification algorithms. For SVM, our aim is to arrange every data as a dot with coordinates in an n-dimensional domain in which n refers to the number of characteristics in our data where each property belongs to a specific coordinate. After this, our objective is to implement grouping through searching a hyperplane which separates both classes exceptionally fine.

Support vectors refer to information which is near the hyperplane and impact the position as well as an introduction to the hyperplane. Utilizing the vectors, we expand the edge within the classifier. Erasing the support vectors will alter the place of the hyperplane.

1.4.4 **Decision Trees**: This algorithm comprises a tree structure that is represented as a flowchart. DT is utilized by a strategy for surmising and classification via portrayal utilizing hubs and internodes. The main and inner hubs constitute of the experiments which are utilized to isolate the occasions with various highlights.

The inner node itself is the consequence of a trait test case. Leaf hubs signify the instance variable. Figure 1.3 demonstrates an example of a DT.

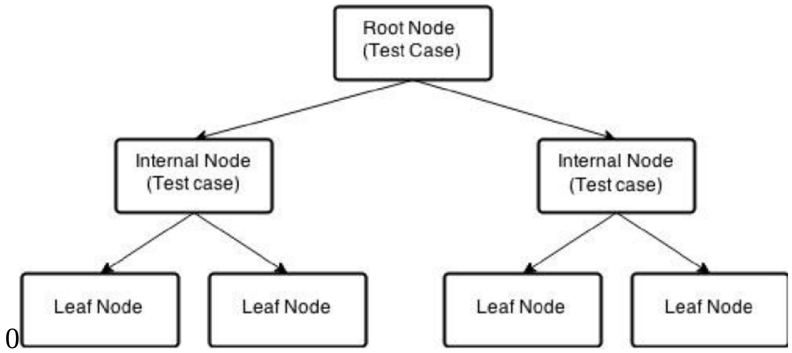


Figure 2 Sample decision tree

### 1.4.5 Random Forest Classifier:

Random forests, otherwise called random decision forests, are a famous troupe technique that can be utilized to fabricate prescient models for both arrangement and relapse issues.

Group strategies utilize different learning models to increase better prescient outcomes — on account of a random forest, the model makes a whole forest of random uncorrelated decision trees to land at the most ideal answer.

Random forest algorithm is a supervised learning classification algorithm. In this algorithm, a forest is created and somehow made random which can clearly be determined by its name. The “Forest” thus constructed is a collection of Decision Trees and mostly prepared and trained with the bagging methods. The basic thought of the bagging technique is that a mix of learning models builds the general outcome.

The greater is the number of trees in a random forest more is the robustness of the forest. And greater is the robustness if a random forest classifier, higher is the accuracy of the results.

The below figure shows the working of a Random forest classifer:

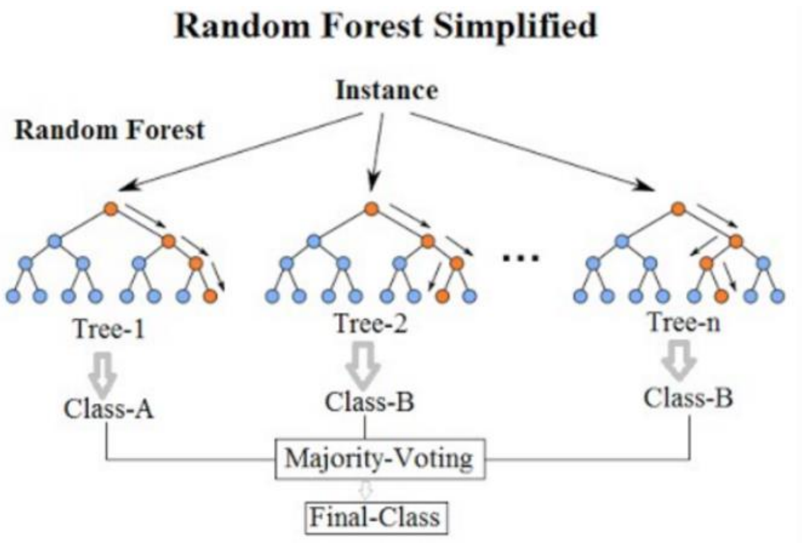


Figure 3 Random Forest

### 1.4.6 Logistic regression

Logistic regression is a factual ML algorithm that arranges the information by considering result factors on extraordinary closures and attempts makes a logarithmic line that recognizes them.

The expression “Logistic” is taken from Logit function that is utilized in this strategy for grouping or so Called classification. I prefer to use this algorithm in my research because it provides a solution to the classification problem which the main aim of this project ie. I have to classify whether or not the patient is diabetic based on the given parameters. The results obtained by this technique are in the form of a Yes or a No.

### 1.4.7 Gradient Boosting

“Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.”

The Gradient boosting algorithm and predict unknown mappings between the inputs. Such algorithms are usually trained on datasets where the input and output pairs are already given.

### 1.5 **Organization**

In Chapter 1, we have discussed what is diabetes disease and challenges faced by diabetic people. Also, we pointed out the schemes that we would be using to analyze this illness in our project report.

In Chapter 2, we would be discussing the research papers we have referred to in order to get a better understanding of our project. The papers mainly focus on techniques used in machine learning and other researches carried out in this field.

In Chapter 3, we would be citing the possible requirements that are the hardware and software system that what language we will be using and where are we going to implement it along with the libraries required along with details about the platform used.

In Chapter 4, we would be discussing in detail the algorithms and approaches used to predict the outcome and effectiveness of our result. Implementations and theresults of the outputs has been discussed.

In Chapter 5,we would be giving the conclusions that have been derived from this study and the future scope of this project.

# Chapter 2

## Literature Survey

In this section, we will be summarizing the research papers we have referred to in order to conduct a comprehensive study of related to our project, that is Implementation of Machine Learning Algorithms for analyzing Diabetes.

## A) Systematic Literature Survey

### 2.1 **"Machine learning applications in cancer prognosis and prediction"**

**Authors** : Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, Dimitrios I Fotiadis

Konstantina Kourou, Themis P. Exarchos along with three researchers conducted a survey to detect, predict and diagnose Cancer. They used various machine learning algorithms for their research including Artificial Neural Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM), Bayesian Networks (BN). A number of analysts from the Bioinformatics and Biomedical field studied the relevance of machine learning approaches, also used it to structure the success of medicating various cancerous growths. This project made use of various input properties plus data to be processed within the above-mentioned ML techniques.

Traditionally, scientists used the procedure of screening at an initial stage even before any of the symptoms were seen to identify a type of malignant growth. Lately, the presences of numerous introductions within the stream of medicine but most efficient outcomes, also predictions are produced by ML algorithms. These strategies can find and distinguish examples and connections within complex datasets while they adequately anticipate future results belonging to the disease type. Additionally, the advantages and disadvantages of each machine learning method incorporated have been illustrated.

The subject of Artificial Intelligence constitutes ML which connects with the issue of studying from statistical tests for the basic idea in surmising. Each procedure comprises the following stages: (1) Prediction of obscure conditions in a framework for particular data. (2) Utilization of evaluated conditions to anticipate new yields of the framework.

The first technique used is ANN. ANNs handle an assortment of characterization or the issue of accepting pattern. These are prepared for creating the yield as a result of blending between figures. The second one is the DT. A Decision Tree is a primary and profound approach used in classification. It comprises of a structure of a tree where a node signifies an input data, output data is represented by the leaf.

The third approach is the SVM. Support Vectors divide the input data into two groups on the basis of the individual data features through a hyperplane. The last one is BN. Bayesian Network follows an approach of finding probability instead of direct estimation.

In light of the examination of the outcome, the dataset they used along with the use of various systems or approaches to choosing attributes can point out evident instruments for the malignant growth space.

## 2.2 **Survey of Machine Learning Algorithms for Disease Diagnostic**

**Authors :** Meherwar Fatima
Maruf Pasha

Meherwar Fatima and Maruf Pasha discussed a few approaches of machine learning in the field of medicine in their research. These approaches have become essential for computerized analysis. Subsequent to utilizing simple condition organs might necessarily not be shown precisely. Thus, design acknowledgment in a general sense includes gaining from these, models. Within the stream of medicine, design acknowledgment and machine learning guarantee the enhanced precision for the analysis of ailment.

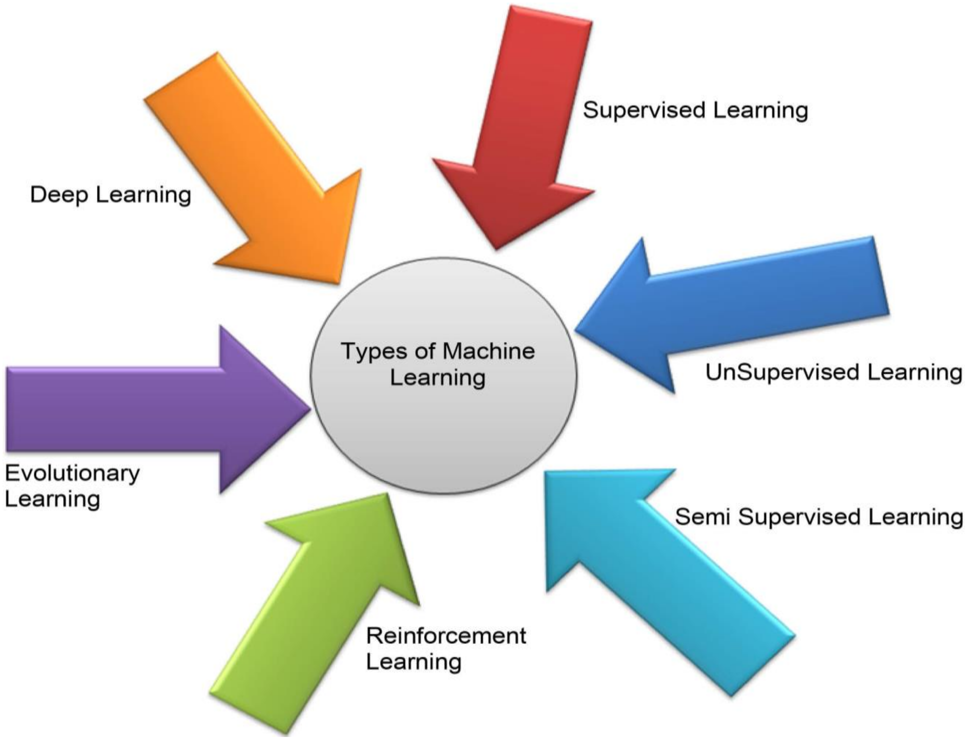


**Figure 4 Types of Machine Learning algorithms**

Numerous scientists have chipped away at various MLL calculations for ailment analysis. Specialists have acknowledged that ML calculations function admirably in the analysis of various illnesses. This paper illustrates the following ailments analyzed by machine learning: Liver, heart, diabetes, hepatitis, and dengue.

In the case of heart diseases, Support Vector Machine gives the most precise results that are of 94.6%. Additionally, support vectors make use of determining the most appropriate characteristics and this approach results in a precision of 85.1% which is lower compared to what is shown by SVM. Naïve Bayes algorithm and decision tree were used for the detection of diabetes.

For diabetes, the Naïve Bayes algorithm was found to be the most efficient with a success of 95%. In addition to better diagnosis, this algorithm points out reduced fallacies. One of the drawbacks of using this approach was that it required an enormously huge dataset. A number of analysts used various combinations of different ML techniques to detect the liver disease but the most precise outcomes were recorded by FT tree ( Feature selection tree). This algorithm took lesser time for its development in comparison to the others with the exactness of 97.1%.

Dengue illness is among the genuine infectious sicknesses. In its diagnosis, the Rough set theory proved to have the maximum correctness. It is skilled to oversee vulnerability, commotion as well as lacking values. Determination of credit engages the classifier to outperform alternate models. In the domain of research for hepatitis, a neural network specifically FFNN( feed forward neural network) illustrated maximum precision with the measurement of 98% . its drawbacks include an immense amount of computerized work. And moreover due to huge datasets it takes a lot of time for evaluation.

This review paper sets forth a number of examples of ML algorithms that have proved to produce the best outcomes due to their capability of differentiating the characteristics precisely. Moreover, it adduces a pool of tools which are shown to have progressed in the field of artificial intelligence.

## B) Algorithmic literature reviews

### 2.3 “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”

Authors: Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov

The group proposed methods for the prevention of neural networks from over-loadedness. One of the designs for acknowledging the above issue they used in their research is a dropout. They used the training dataset and followed ideologies to arbitrarily to abdicate the unit as well as its interconnections. It altogether lessens overloading and produces significant upgrades over the regular strategies. They demonstrated that dropout enhances the execution of neural systems on administered errands in perception, discourse acknowledgment, and science.

Profound neural systems comprise of various nonstraight shrouded layers which result in exceptionally expressive models which learn extremely entangled connections amongst the sources of info and yields. With restricted preparing information, a huge number of these convoluted connections will be the after effects of testing clamor, hence these would be present in the preparation set yet not in the genuine set Regardless of whether it is concluded from similar dissemination. This prompts overloading and numerous strategies are created for lessening which incorporates halting the preparation when execution on an approval set begins to deteriorate presenting weighing punishments of different sorts.

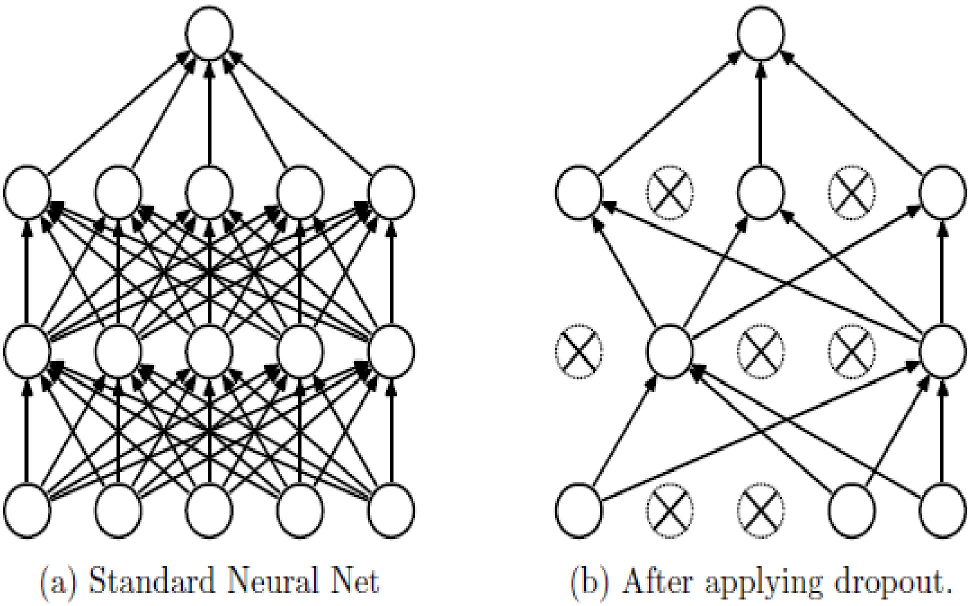


Figure 5 Graphs depicting Dropout

Back-propagation learning develops week co-adjustments which are related to the preparation information, however, don’t sum up to inconspicuous information. Arbitrary dropout separates the co-adjustments resulted by creating the nearness of a specific concealed unit untrustworthy. This technique tends to enhance the execution of neural networks in an enormous assortment of usage area consisting of protest grouping, digit acknowledgment, discourse acknowledgment, report characterization and investigation of computerized science information. This proposes that dropout is a common procedure, furthermore isn't explicit to some areas.

Dropout extensively enhances the execution of common neural networks on another information collection also. The boost in training time is a major disadvantage of dropout. This scheme regularly takes twice or thrice the time to prepare as compared to a common neural system of similar engineering.

A note-worthy reason for this expansion is that the characteristical refreshes are extremely loud. A preparation case successfully attempts to prepare an alternate irregular design. In this manner, the inclinations which are being registered aren't slopes of the last design that is destined to be utilized at the testing phase. Hence it isn't amazing that preparation requires quite a while. But all things considered, it anticipates overloading.

## 2.4 "Comparing the computational complexity and accuracy of classification algorithms"

**Authors**: Beatrice, S and Thirumahal, R and Raja

They conducted an evaluation to compare the complexity of classification algorithms. They compared two algorithms of data mining with the Pioneer classifier algorithm. Their paper illustrates the above-mentioned approach to set upon SLIQ and OIESLIQ. One of the vital assignments of data mining is a group of information.

The 3 algos discussed go for characterizing which decision tree is ideal as well as easy to comprehend and translate. The Supervised Learning in Quest technique can deal with numerical as well as qualitative characteristics. In this algorithm, a GINI Index is created where each numerical value has n attributes. In the latter approach, i.e.

The On Improving the efficiency of SLIQ was developed mainly to overthrow the drawbacks of SLIQ. This scheme is a faster approach as the lesser number of computations are carried out, c*m in the best case where c is the product of the number of classes whereas in SLIQ, m*(n-1) computations are done which is way more than the previous scheme.

The proposed technique, The Pioneer Classifier Algorithm efficiently segregates enormous sets of data rather than the little sets of data. In this method, the split value is evaluated by taking the aggregate of the SD (standard deviation of the group ) and the least value in the data.

A Gini Index is then calculated corresponding to each index. After the research was conducted, it was concluded that the OIESLIQ algo gave the reduced split values for all data records in comparison to the other two techniques.

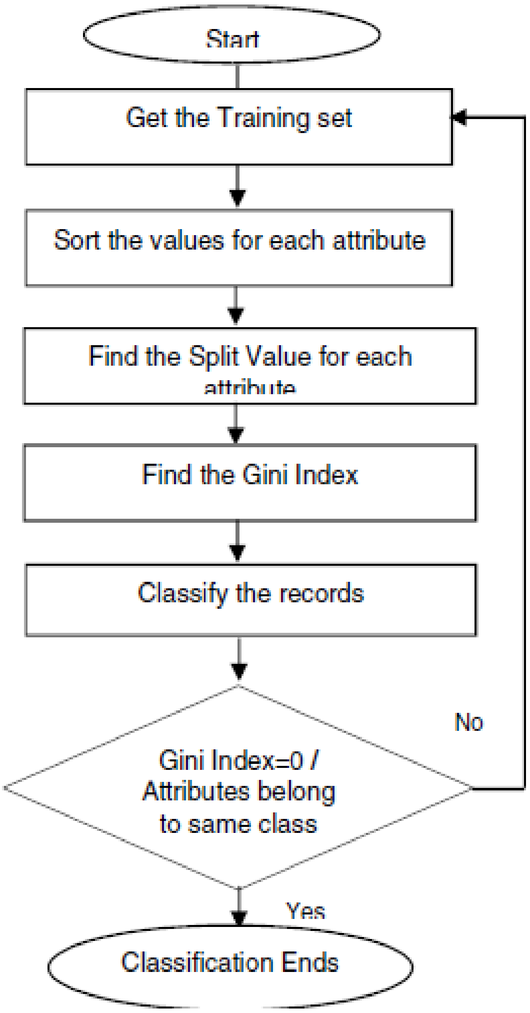

Start
Get the Training set
Sort the values for each attribute
Find the Split Value for each attribute
Find the Gini Index
Classify the records
Gini Index=0 / Attributes belong to same class
No
Yes
Classification Ends

Figure 2 flowchart of the three algorithms

## 2.5 **Do we Need Hundreds of Classifiers to Solve Real-World Classification Problems**?

**Authors:** Manuel Fern´andez-Delgado, Eva Cernadas, Sen´en Barro

The group members conducted a survey to solve a real-world problem which is faced by almost every analyst that is which classifier should be chosen amongst the hundreds of classifiers present. They conducted this research 179 classifiers coming from 17 different families in a number of languages like Weka, Matlab, C, and R. The prominent ones among these families were SVMs, BNN, Decisions trees, random forests and many more.

They used more than 100 sets of data to check the efficiency and accuracy of each of this classification technique. The random forests classifier was destined to be the most efficient and achieved 94.1% exactitude. The SVM with Gaussian Kernel was, however, the second best when performed with C using LibSVM and gave the accurateness of 92.3%.

Usually, the scientists tend to leave behind the techniques that they have never come across and often use the ones they have mastery in. In this way many highly efficient techniques are forgotten and never applied, however, these may result in better outcomes and higher accuracies.

The objectives of this research were to firstly, to choose the most suitable classifier for the set of data. Secondly, to arrange each one according to its exactness. Thirdly to evaluate the probabilities of every classifier and its family of scoring the maximum accuracy. Also to compare the obtained results with the best possible results.

Though this study came across certain issues, for example, it is not possible to tell what is the maximum accuracy that can be attained for a set of data, so it becomes quite difficult to measure the true nature of a scheme. It cannot be determined if the selected algorithms would efficiently work or not.

After the study, it was concluded that the best possible results were attained by the Parallel Random Forest classifier when it was implemented in R with a caret achieving an average of 94.1 % exactness.

This algorithm achieved an average accuracy of 82.0% when it was tested over other sets of data. However Random forest classifier tuned with the caret in R stood just next to the previous approach even though it gave better and more accurate results.

## 2.6 "An empirical study of the naive Bayes classifier"

**Authors:** Irina Rish

The Naive Bayes classifier significantly disentangles learning by accepting that the highlights are autonomous given class. In spite of the fact that autonomy is commonly a poor supposition, practically speaking Naive Bayes regularly contends well with progressively complex classifiers.

The goal of this research was to comprehend the information attributes which influence the execution of naive Bayes. Monte Carlo simulations have been utilized in this particular approach to allow a deliberate investigation of arrangement exactness for a few classes of haphazardly created issues.

They have dissected the effect of the appropriation entropy n the grouping blunder, appearing low-energy includes appropriations yields great precision for this classifier. Certainly, it has also been demonstrated that Naïve Bayes Classifier works best feature dependencies that are closely-functional.

This observation thus gives two good outputs in the independent features which were expected and surprisingly in the features that are functionally dependent. Another astonishing outcome is that the precision of Naive Bayes isn't legitimately connected with the degree of highlight dependencies estimated as the class contingent shared data between the elements.

The research concludes that in spite of its implausible autonomy supposition, the naive Bayes classification technique is shockingly powerful by and by since its characterization choice may frequently be right regardless of whether its likelihood gauges are wrong. This classifier gives the worst performance between the above two mentioned scheme.

Shockingly, the precision of this classifier isn't straightforwardly connected with the level of highlight dependencies estimated as the class-contingent common data between the highlights. Rather, a superior indicator of exactness is the loss of data that features contain about the class while considering the Bayesian model. Be that as it may, further experimental and hypothetical contemplate is required to all the more likely comprehend the connection between those data theoretic measurements and the conduct of naive Bayes.

Further headings additionally incorporate the investigation of naive Bayes on the pragmatic application that has nearly deterministic dependencies, describing different districts of naive Bayes optimality furthermore, concentrating the impact of different data features on the Bayes mistake.

At long last, a superior comprehension of the effect of independent assumption on characterization can be used to devise better estimation strategies for learning proficient Bayesian net classifiers, and for a probabilistic deduction, e.g., for discovering most extreme probability assignments.

### 2.7 **“Prediction of Outcome in Acute Lower Gastrointestinal Bleeding Using Gradient Boosting”**

**Authors:** Lakshmana Ayaru , Petros-Pavlos Ypsilantis , Abigail Nanapragasam, Ryan Chang-Ho Choi, Anish Thillanathan, Lee Min-Ho , Giovanni Montana

Most of ALGIB(Acute lower Gastrointestinal bleeding) introductions (80 to 85%) resolve suddenly with no unfavorable result and passing is unprecedented (2 to 4%). Anyway, a noteworthy extent (17 to 30%) experience remedial mediation to halt unstoppable bleeding. Because of worry about intermittent bleeding or requirement for mediation routine clinical practice for most by far of patients with ALGIB who present to crisis division is admission to a medical clinic for in-quiet perception for a variable number of days with an extent experiencing endoscopy or radiological examination.

Models based on MLR(multiple logistic regression) might be restricted in foreseeing result in ALGIB as they depend on the supposition that a straight mix of the watched highlights can be utilized to decide the likelihood of every specific result overlooking any factor cooperation which might be the key for the exact forecast.

SO to improve the results of MLR, and a more accurate prediction of bleeding, Gradient Boosting algorithm was suggested to reach a non-parameterized algorithm’s performance.

The point of this examination was to test whether the GB calculation had the capacity to precisely anticipate clinical results in patients showing to crisis offices with ALGIB utilizing non-endoscopic factors accessible to clinicians around then.

The results estimated were helpful mediation extreme bleeding and intermittent bleeding. These results were picked as they demonstrate the requirement for inpatient care. Restorative mediation to stop the source of a bleed was incorporated as this recommended the nearness of a continuous bleed that was not settling precipitously.

The current clinical practice incorporates colonoscopic-based triage which is obtrusive, satisfactory readiness hard to accomplish and treatable stigmata of rare hemorrhage. This examination has demonstrated that a GB calculation dependent on clinical and research facility factors was exact (greater than 80%) in foreseeing the clinical results of repetitive and continuous bleeding, therapeutic intercession and extreme bleeding.

The Gradient boosting algorithm predicted high negative values. This recommends these models might be valuable to triage patients into a generally safe gathering who could be made do with a shortened remain in emergency clinic maintaining a strategic distance from abnormal amounts of consideration or as outpatients.

A specific quality of this examination is the approval and great execution in an outside associated with a lower frequency of serious bleeding showing the calculation can keep up precision in an alternate setting.

This GB approach proved superior to the pre-existing MLR approaches and the other-other published approaches as well. Outfit AI models have been appeared to be more precise than ordinarily calculated relapse to arrange ailment or foresee result in an assortment of clinical settings.

# Chapter 3

# System Development

**Software Used**: Jupyter Notebook Using Anaconda

**Language Used**: Python

### 3.1 About Jupyter Notebook

The Jupyter Notebook provides the perfect environment to conduct researches and studies using computer codes in languages like python. They can read documents which have text figures like paragraphs, equations, links, etc. These notebooks can easily be read by any human as these are documents that have documents on which an analysis was conducted as well as the results in the form of figures or tables or plots. This application is a client-server application that enables you to edit as well as run your notebook in any web browser.

A standout amongst the most huge advances in the logical registering field is in progress with the blast of enthusiasm for Jupyter (in the past, IPython) Notebook innovation. The logical distribution Nature as of late included an article on the advantages of Jupyter Notebooks for logical research.

The explanation behind Jupyter's enormous achievement is it exceeds expectations in a type of programming called Literate programming. This kind of programming stresses a writing first methodology where piece with human-accommodating content is punctuated with code squares. It exceeds expectations at show, research, and showing goals particularly for science. Proficient programming enables clients to figure and portray their musings with composition, enhanced by numerical conditions, as they get ready to compose code squares. This mentality is the opposite we for the most part consider code. Throughout the years, we have seen shut source, revenue driven executions of "Literate programming" with programming bundles, for example, Mathematica, and Matlab.

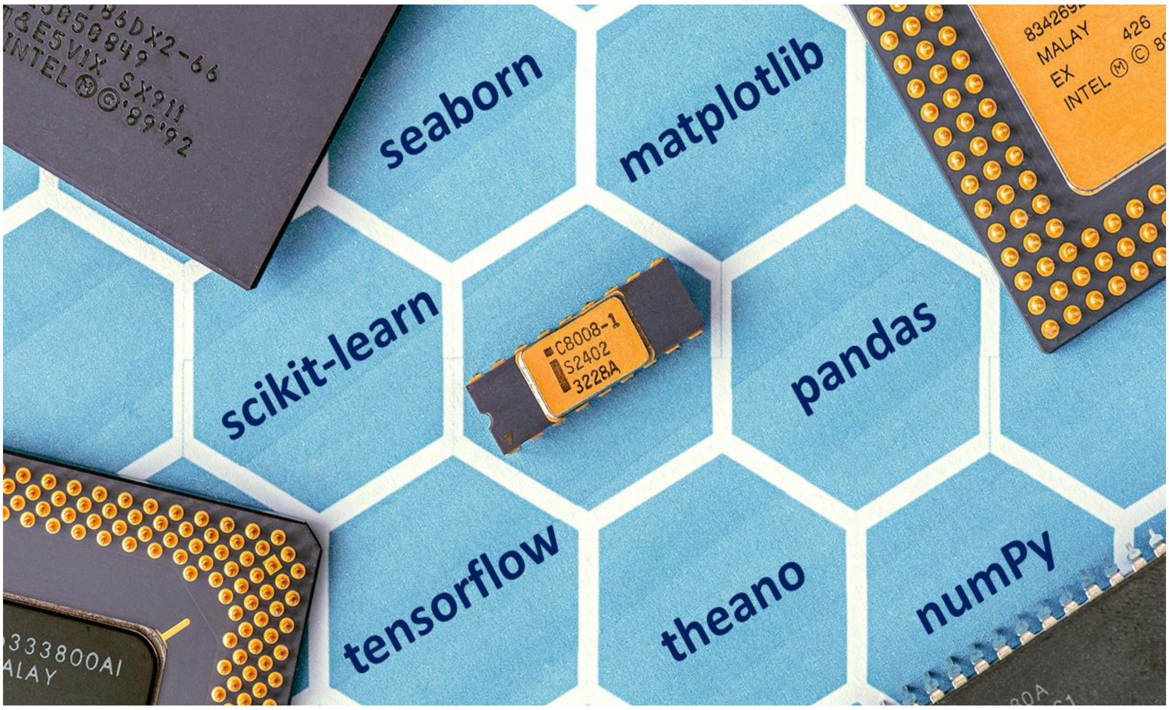


Figure 3 Python libraries used in the project

## 3.2 About Python Language

Python is a "batteries included" PC programming language. All the more solidly, Python is a programming language that, as opposed to other programming dialects, for example, C, Fortran, or Java, enables clients to all the more promptly center and take care of space issues as opposed to managing the multifaceted nature of how a PC works. Python accomplishes this objective by having the accompanying characteristics.

Python is a powerful coding language. It is not like R and is in itself a complete language as well as provides a platform that is useful for innovation and creating production systems. It is highly recommended by developers in this the field of data analysis as it has a consistent syntax and refined models. It can also be incorporated into other web applications and production conditions.

Python is an high-level language, implying that it abstracts fundamental PC related specialized subtleties. For instance, Python does not make its clients ponder PC memory the board or appropriate revelation of factors and uses safe suppositions about what the software engineer is endeavoring to pass on. Also, an abnormal state language can be communicated in a way nearer to English exposition or numerical conditions. Python is ideal for educated programming due to its lightweight, "low function" nature.

Python is a broadly useful language implying that it very well may be utilized for all issues that a PC is able to do as opposed to represent considerable authority in a particular region, for example, factual investigation.

Python is an interpreted language implying that assessment of code to acquire results can happen quickly as opposed to experiencing a tedious, arrange and run cycle, which in this manner accelerates the reasoning and experimentation forms. IPython is an intelligent type of the Python language additionally concocted by Fernando Pérez. These situations exceed expectations for fast model of code or brisk and straightforward experimentation with new thoughts.

This language provides you with a large number of libraries and modules which are its most valuable assets make our tasks easier and also facilitate us with many ways to perform each task.

The most repeatedly used and useful libraries that are used in this project are listed below:

1. Scikit-learn - This is the most famous library in python used for machine learning. It provided many unsupervised and supervised ML algos. Scikit-learn has a base of two main libraries, NumPy and SciPy.
   Scikit-learn gives a scope of administered and unsupervised learning algorithms by means of a steady interface in Python.
   It is authorized under a lenient streamlined BSD permit and is circulated under numerous Linux disseminations, empowering scholarly and business use.
   The library is based upon the SciPy that must be introduced before you can utilize scikit-learn.

2. NumPy - offers us high in performance array objects and functions and tools to work with these arrays. It helps in carrying out high-speed operations on multidimensional arrays containing data of similar type.

3. SciPy when combined with NumPy, gives us numerous easy to use and productive numerical practices, for example, schedules for numerical incorporation and advancement.

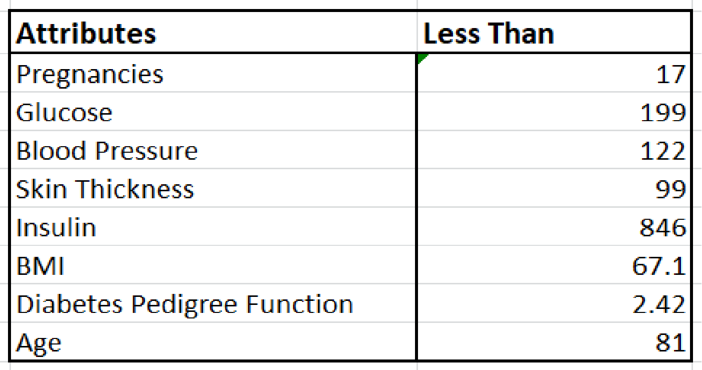
4. Pandas- This library offers intuitive and easy to use high-level data structures. It has inbuilt operations that can conduct combining, grouping as well as filtering of information.

5. Matplotlib- Machine learning works on the basic principle of analyzing information which is facilitated by the Matplotlib library. It can be used for representing the data graphically the data through 2D plots and graphs.

We are applying the machine learning algorithms which we are using for our project in the following dataset. This dataset has been sourced from kaggle.com(UCIML).

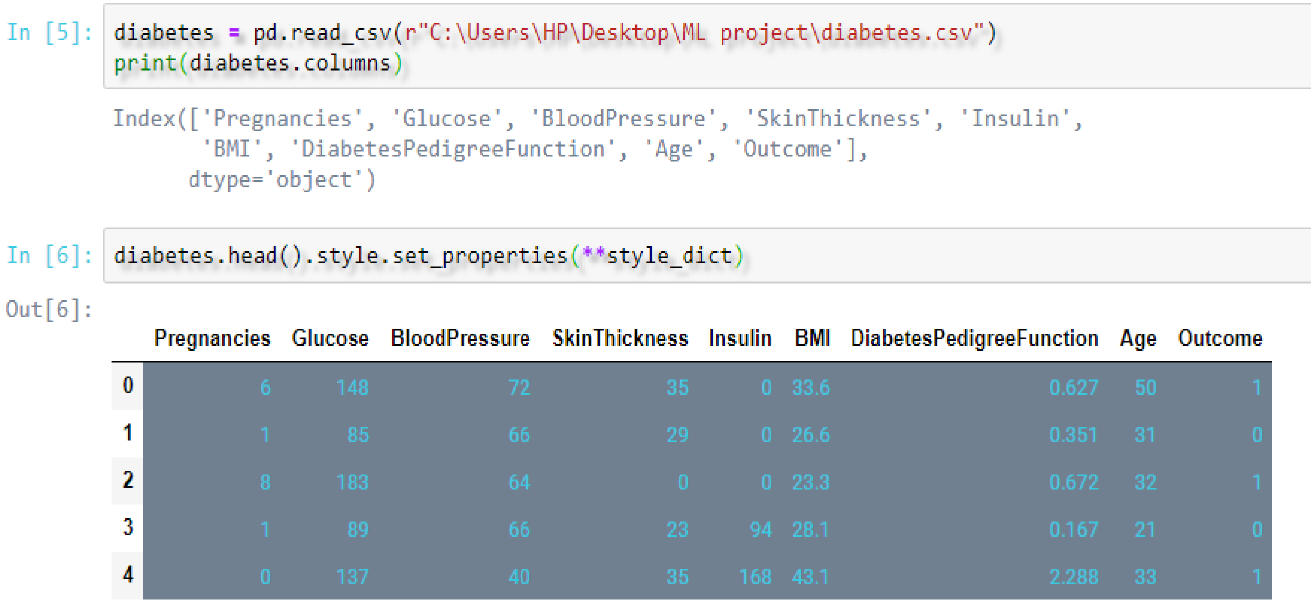
### 3.3 Description of Data:

Our data set belongs to Pima India Diabetes database of the National Institute of Diabetes and Digestive and Kidney Diseases. Our data set is of type.csv type. It comprises eight attributes which are the therapeutic indicators that are analyzed for the presence of diabetes in any patient and 768 instances. These indicators vary from patient to patient. The following table shows the highest ranges of the medical indicators in our dataset.



| Attributes | Less Than |
|---|---:|
| Pregnancies | 17 |
| Glucose | 199 |
| Blood Pressure | 122 |
| Skin Thickness | 99 |
| Insulin | 846 |
| BMI | 67.1 |
| Diabetes Pedigree Function | 2.42 |
| Age | 81 |

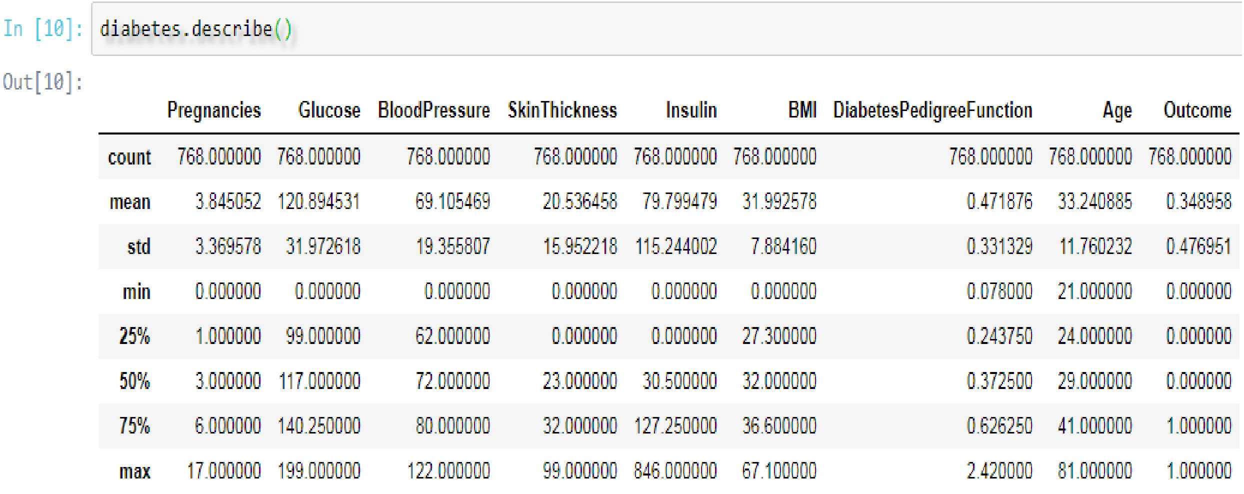
Figure 8 Attributes in the dataset.



```
In [5]: diabetes = pd.read_csv(r"C:\Users\HP\Desktop\ML project\diabetes.csv")
        print(diabetes.columns)

        Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
               'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
              dtype='object')

In [6]: diabetes.head().style.set_properties(**style_dict)

Out[6]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

Figure 4 First 5 entries in the dataset on Jupyter



```
In [10]: diabetes.describe()

Out[10]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| count | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| mean | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| std | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| 50% | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 | 0.000000 |
| 75% | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 | 1.000000 |
| max | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

Figure 5 Description of the diabetes dataset

# Chapter 4

# Performance Analysis

## 4.1 In depth Analysis

In this chapter we will be discussing all the algorithms that have been used in this research project. Along with the basic definitions and the ideas about the algorithms , the implementation of the same has been shown. Furthermore, I will be discussing the results of each of these algos and the changes that had to be made to each approach to make it suitable for achieving the desired best results.  Algorithms used for analyzing diabetes:

### 4.1.1 **Naïve Bayes Algorithm**:

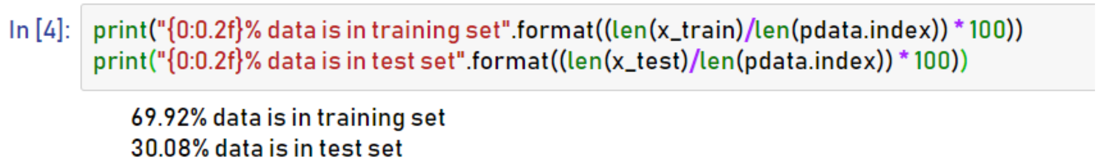
This technique works by probability measure which follows a specific sequence and order in its implementation. In order to determine relationships among the signs, diagnosis, and treatments of diabetes, various mining or ML approaches can be used, but they come with a number of disadvantages like repetitiveness and increasingly large time for calculating the outcomes. BN deals with these disadvantages in its own ways by removing an increasing number of repeated predictions. Moreover, this scheme might also be utilized for a huge data set in actual time.

This method is implemented using the following formula:

$$\begin{matrix}\text{Posterior Probability} \\ P(c|x)\end{matrix} = \frac{\begin{matrix}\text{Likelihood} \\ P(x|c)\end{matrix} \times \begin{matrix}\text{Class Prior Probability} \\ P(c)\end{matrix}}{\begin{matrix}\text{Predictor Prior Probability} \\ P(x)\end{matrix}}$$

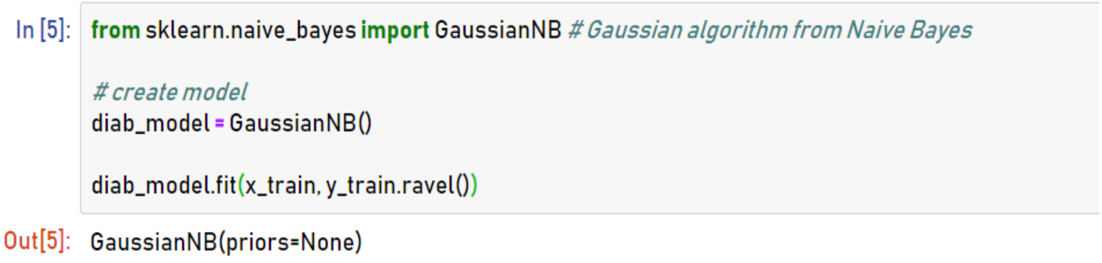
Implementation of Naïve Bayes:

For this implementation , I had split my data set into 70:30 ratio where training set is of size 70% approximately and the test set is of 30% approximate size.

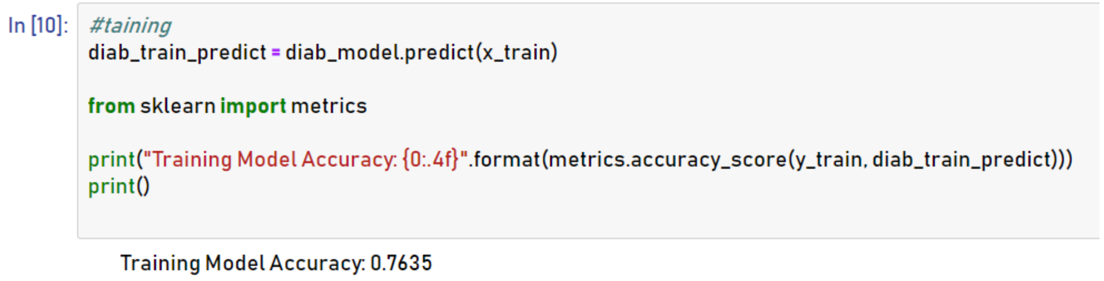


```
In [4]: print("{0:0.2f}% data is in training set".format((len(x_train)/len(pdata.index)) * 100))
        print("{0:0.2f}% data is in test set".format((len(x_test)/len(pdata.index)) * 100))

        69.92% data is in training set
        30.08% data is in test set
```

The image given underneath shows the creationof the Naïve Bayes classifier Model. I have opted for the Gaussian Algorithm which is the simplest classifier model.

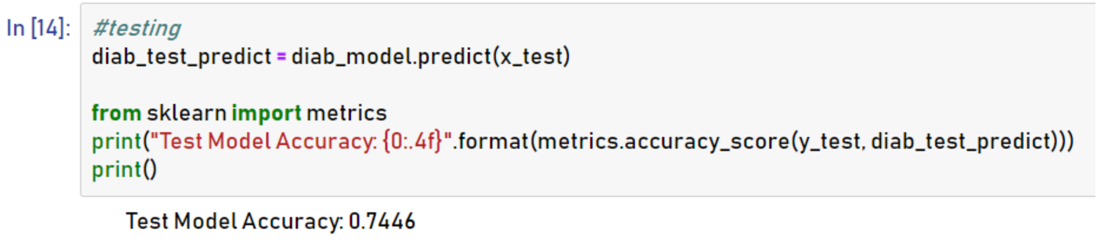


```
In [5]: from sklearn.naive_bayes import GaussianNB # Gaussian algorithm from Naive Bayes

        # create model
        diab_model = GaussianNB()

        diab_model.fit(x_train, y_train.ravel())

Out[5]: GaussianNB(priors=None)
```



```
In [10]: #taining
         diab_train_predict = diab_model.predict(x_train)

         from sklearn import metrics

         print("Training Model Accuracy: {0:.4f}".format(metrics.accuracy_score(y_train, diab_train_predict)))
         print()

         Training Model Accuracy: 0.7635
```

The above snippet shows the accuracy of the training model while the lower snippe shows the accuracy score we receive on the testing set.



```
In [14]:  #testing
          diab_test_predict = diab_model.predict(x_test)

          from sklearn import metrics
          print("Test Model Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test, diab_test_predict)))
          print()

          Test Model Accuracy: 0.7446
```

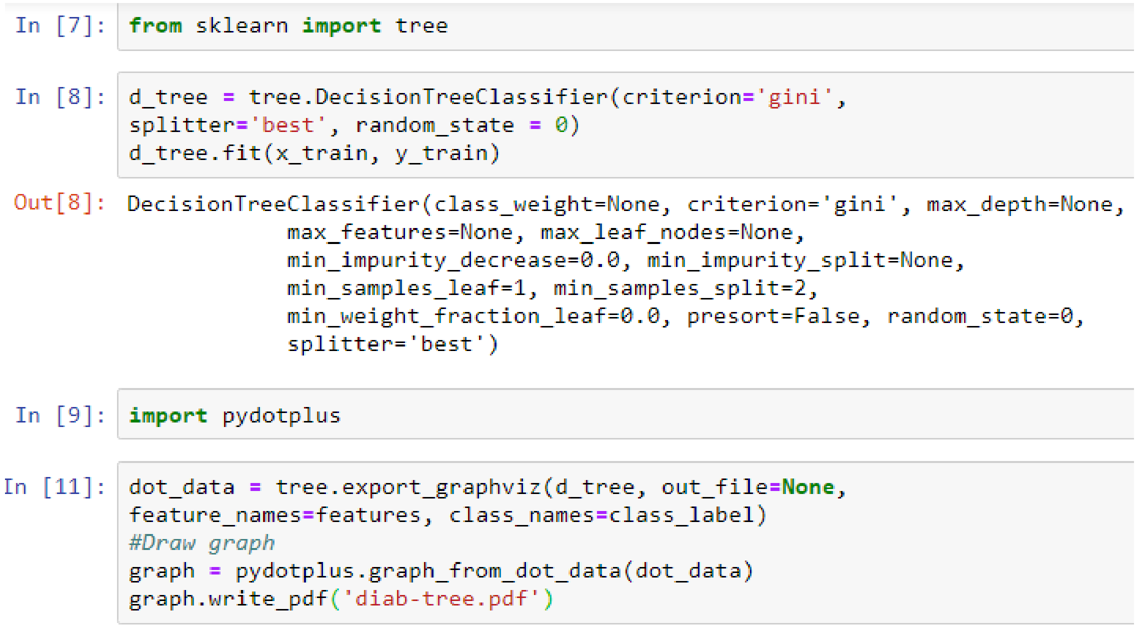
So we see that the Naïve Bayes classifier gives good results for our problem.

### 4.1.2 **Decision tree algorithm**:

This technique gives a strong approach to the estimation and classification of Diabetes disease. Every node of such a tree is given by computing the largest data increase within all the properties, but when a particular property produces an uncertain last outcome, its leading branches are ceased and the final value is allocated to that unit.

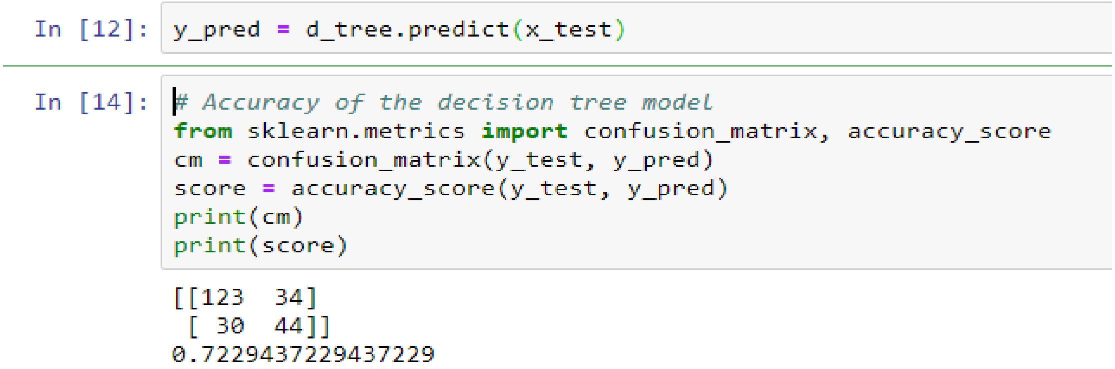
DT produced after the algorithm is implemented on a dataset helps decide if a person has been detected positive with diabetes or negative. The information is split into sub-modules and property is allotted to it. The root of the tree is given by the maximally normalized attribute in our dataset.

The below figure shows part of the implementation of the DT:



```
In [7]:  from sklearn import tree

In [8]:  d_tree = tree.DecisionTreeClassifier(criterion='gini',
         splitter='best', random_state = 0)
         d_tree.fit(x_train, y_train)

Out[8]:  DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                     max_features=None, max_leaf_nodes=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, presort=False, random_state=0,
                     splitter='best')

In [9]:  import pydotplus

In [11]: dot_data = tree.export_graphviz(d_tree, out_file=None,
         feature_names=features, class_names=class_label)
         #Draw graph
         graph = pydotplus.graph_from_dot_data(dot_data)
         graph.write_pdf('diab-tree.pdf')
```

The accuracy of this algorithm was predicted by:



```
In [12]: y_pred = d_tree.predict(x_test)
```

```
In [14]: # Accuracy of the decision tree model
         from sklearn.metrics import confusion_matrix, accuracy_score
         cm = confusion_matrix(y_test, y_pred)
         score = accuracy_score(y_test, y_pred)
         print(cm)
         print(score)
```

```
[[123  34]
 [ 30  44]]
0.7229437229437229
```

The root node in the outcome was:

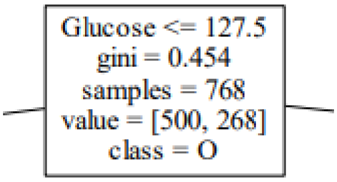


**Figure 6 DT root node**

The below figure is part of the DT emerging from the root node:

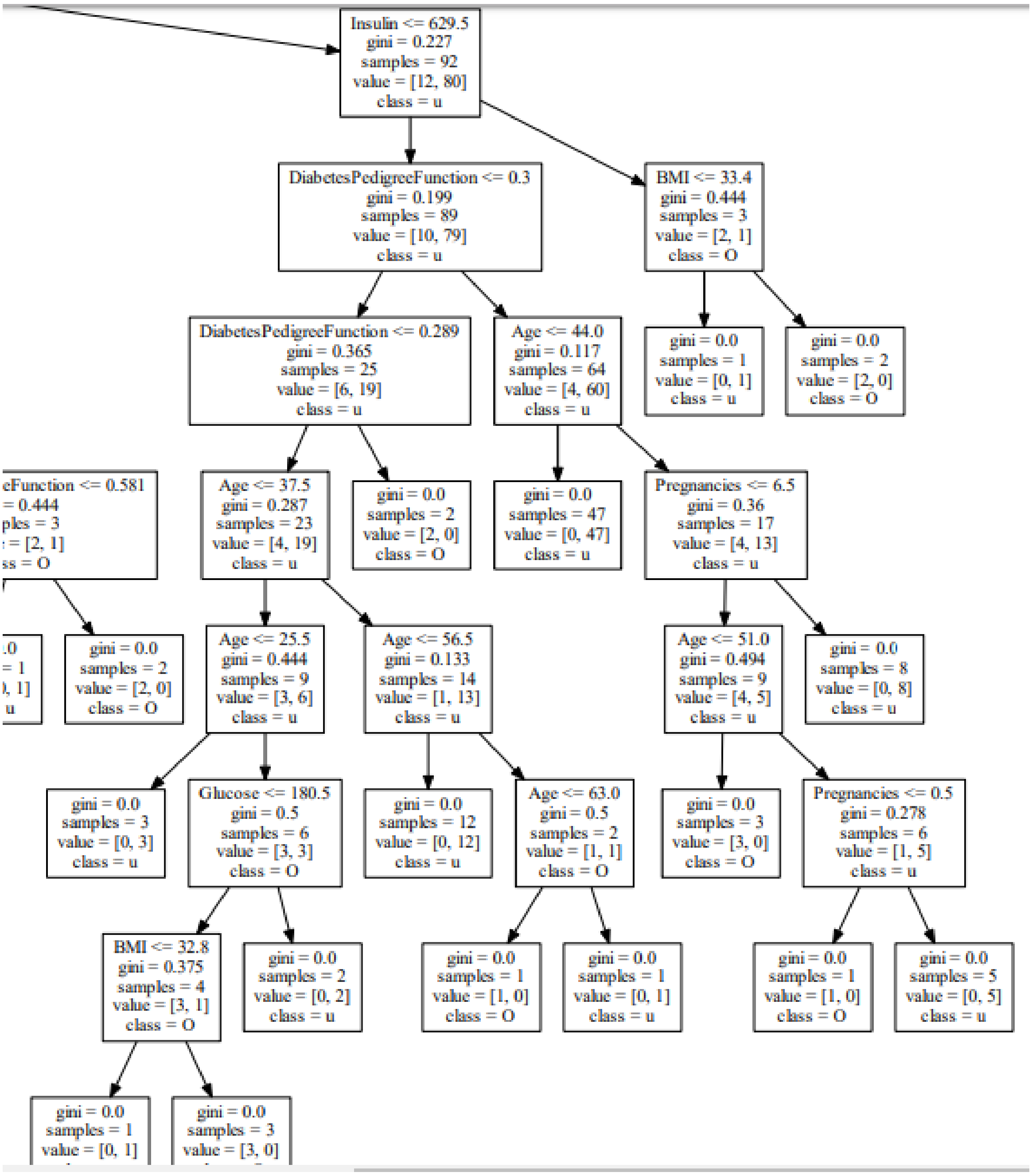


Figure 7 Analysis of Diabetes using DT

### 4.1.3 **K-Nearest Neighbor Algorithm**:

This algorithm involves inputting our data set of 9 columns and 769 rows where there are 8 input vectors and a single output vector. This approach computes the Euclidean distance amongst each attribute. Further, it decides an arbitrary value of k which is said to be the number of nearest neighbors. Then using the values of Euclidean Distance, it finds out that which value belongs to which attribute on the basis of which it computes the outcome.

To make a forecast for another information point, the calculation finds the nearest information focuses on the preparation information set—its "closest neighbors".

If values come out to be identical, then the person is said to be suffering from diabetes, else not. Furthermore, we can calculate the accuracy and precision of this algorithm.

It is also called a non-parametric approach. This implies that this technique doesn't propose any suppositions within a hidden information dissemination that is the arrangement is resolved through the information. Subsequently, this scheme is utilized if we initially have no learning related to the data.

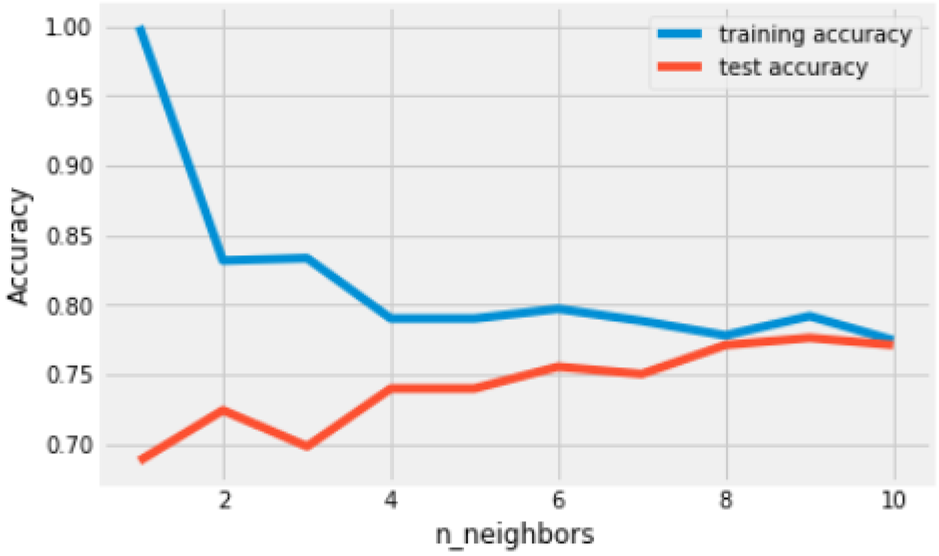
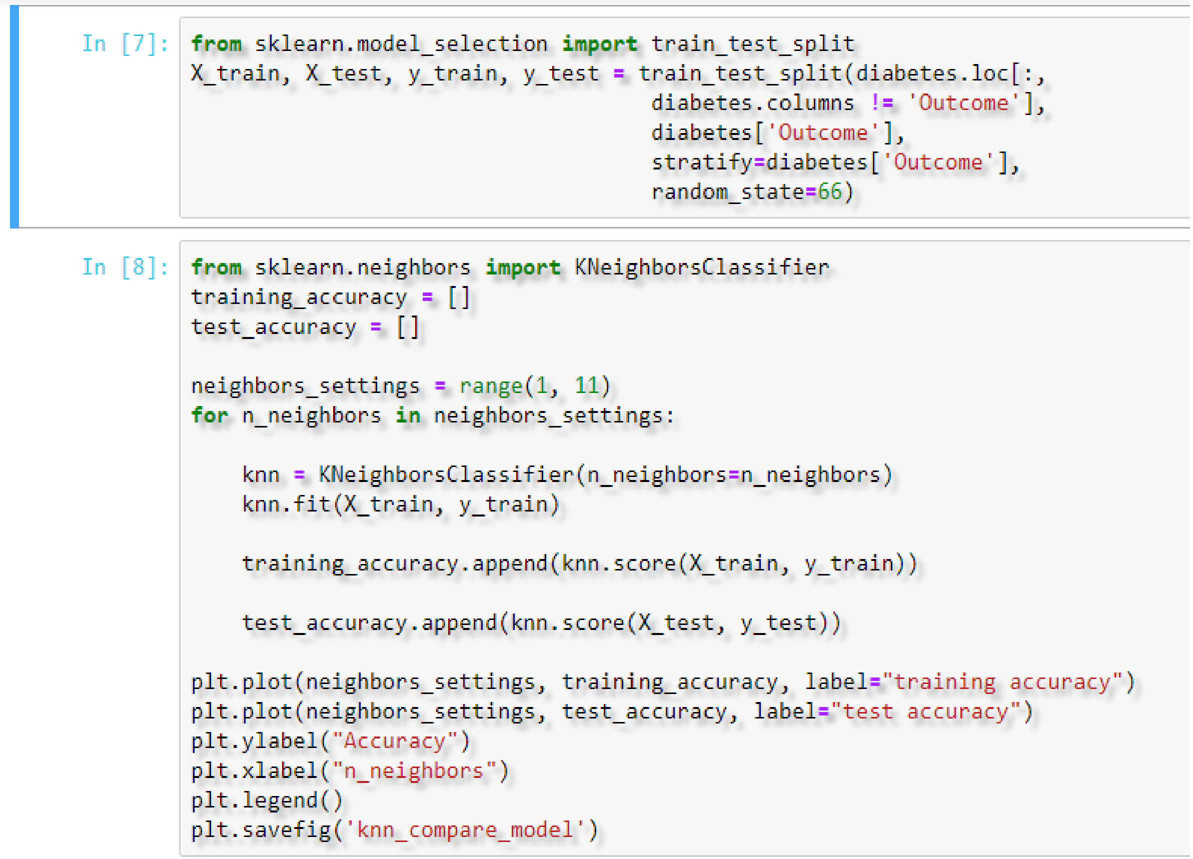


Figure 11 Training accuracy for KNN

The above plot indicates the relation of the number of nearest neighbors with the test and train accuracy. On the y-axis, we have the accuracy meter whereas on the x-axis is the n_neighbours i.e. the number of nearest neighbors.

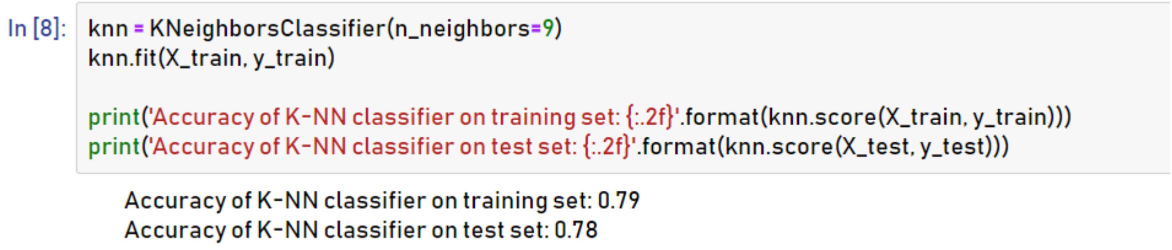
The graph very well indicates that we get the maximum training accuracy score for n=1 but in this case, the test accuracy score is the least.

Implementation of KNN:



```
In [7]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(diabetes.loc[:,
                                                    diabetes.columns != 'Outcome'],
                                                    diabetes['Outcome'],
                                                    stratify=diabetes['Outcome'],
                                                    random_state=66)
```

```
In [8]: from sklearn.neighbors import KNeighborsClassifier
        training_accuracy = []
        test_accuracy = []

        neighbors_settings = range(1, 11)
        for n_neighbors in neighbors_settings:

            knn = KNeighborsClassifier(n_neighbors=n_neighbors)
            knn.fit(X_train, y_train)

            training_accuracy.append(knn.score(X_train, y_train))

            test_accuracy.append(knn.score(X_test, y_test))

        plt.plot(neighbors_settings, training_accuracy, label="training accuracy")
        plt.plot(neighbors_settings, test_accuracy, label="test accuracy")
        plt.ylabel("Accuracy")
        plt.xlabel("n_neighbors")
        plt.legend()
        plt.savefig('knn_compare_model')
```



```
In [8]: knn = KNeighborsClassifier(n_neighbors=9)
        knn.fit(X_train, y_train)

        print('Accuracy of K-NN classifier on training set: {:.2f}'.format(knn.score(X_train, y_train)))
        print('Accuracy of K-NN classifier on test set: {:.2f}'.format(knn.score(X_test, y_test)))

            Accuracy of K-NN classifier on training set: 0.79
            Accuracy of K-NN classifier on test set: 0.78
```

We further and keep on increasing the number of nearest neighbors up to n=9 and we get the perfect score 79% and 78% on the training set and the test set respectively.

### 4.1.4 Random Forest Classifier

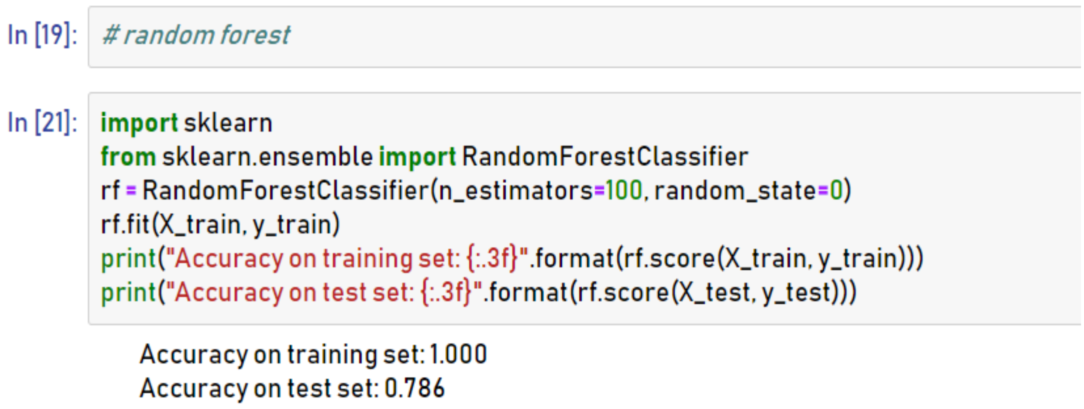
Random forest algorithm is a supervised learning classification algorithm. In this algorithm, a forest is created and somehow made random which can clearly be determined by its name. The “Forest” thus constructed is a collection of Decision Trees and mostly prepared and trained with the bagging methods. The basic thought of the bagging technique is that a mix of learning models builds the general outcome.

The greater is the number of trees in a random forest more is the robustness of the forest. And greater is the robustness if a random forest classifier, higher is the accuracy of the results.

In short, we can say that Random forest forms numerous DTs (Decision trees) and combines them to get an increasingly exact and stable expectation. The main plus point of this algo is that it can freely be used for classification problems as well as regression problems.

The essential parameters to RFC can be all out a number of trees to be produced and choice tree related parameters like split criteria, least split, Gini index, gain ratio, etc. The random forest classifier is less complex and all the more dominant when contrasted with the other non-linear classification.

Implementation of Random Forest Classifier:



```
In [19]: #random forest

In [21]: import sklearn
         from sklearn.ensemble import RandomForestClassifier
         rf = RandomForestClassifier(n_estimators=100, random_state=0)
         rf.fit(X_train, y_train)
         print("Accuracy on training set: {:.3f}".format(rf.score(X_train, y_train)))
         print("Accuracy on test set: {:.3f}".format(rf.score(X_test, y_test)))

         Accuracy on training set: 1.000
         Accuracy on test set: 0.786
```

This approach gives us 78.6% accuracy on our test set which is far better than the single decision tree and also better than the logistic regression model without the involvement of any parameters. Be that as it may, we can alter the max_features setting, to see whether the outcome can be improved.

### 4.1.5 Logistic Regression

Logistic Regression is a classification algorithm which can be used when a categorical response variable is required. It can be used to determine the bond between the characteristics and the probability of a specific output. The expression "Logistic" is taken from Logit function that is utilized in this strategy for grouping or so called classification. I prefer to use this algorithm in my research because it provides a solution to the classification problem which the main aim of this project ie. I have to classify whether or not the patient is diabetic based on the given parameters. The results obtained by this technique are in the form of a Yes or a No.

A clarification of the logistic function will give us the clarification of Logistic Regression.

A logistic function takes values between 0 and 1 which is the reason why it is a Sigmoid Function.

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

The plot of the above equation looks like:

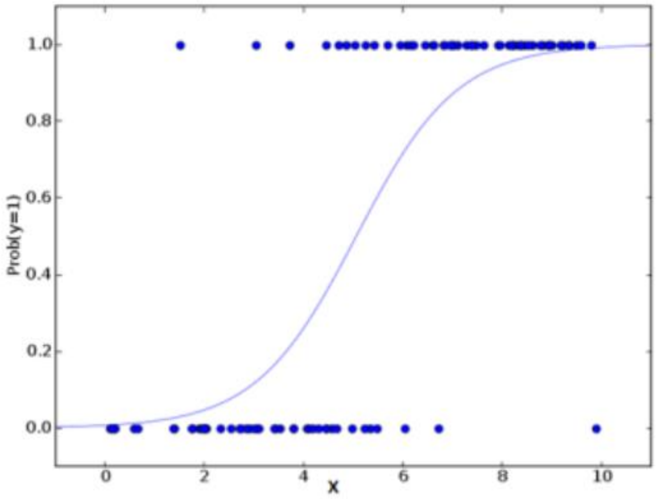


Figure 8 Logistic regrassion graph

Let's consider ***t*** as linear function in a univariate regression model.

$$t = \beta_0 + \beta_1 x$$

So the Logistic Equation will become

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Now, when logistic regression model come across an outlier, it will take care of it.

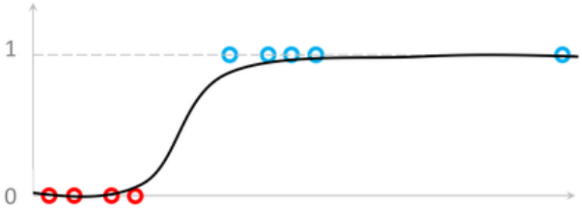
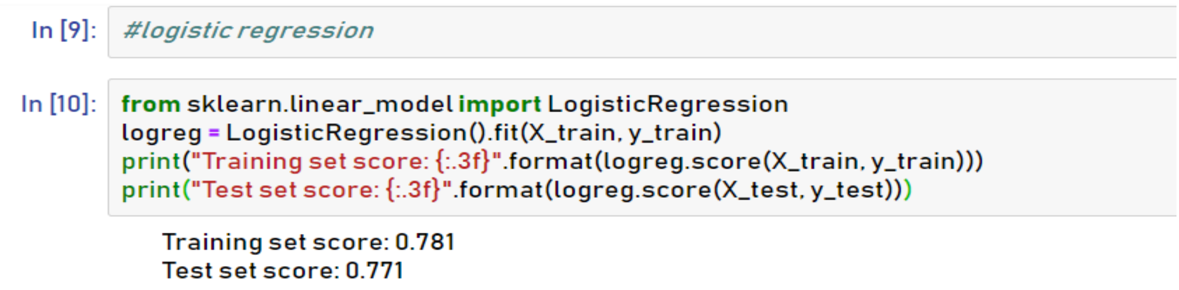


Figure 12 logistic regression model

Decision Boundaries separate probabilities into positive and negative classes.

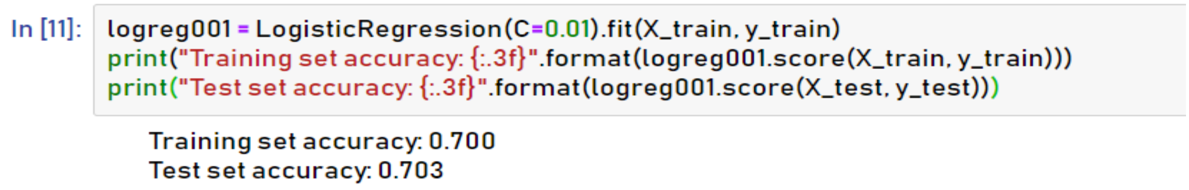
Implementation of Logistic Regression:



```
In [9]:  #logistic regression

In [10]: from sklearn.linear_model import LogisticRegression
         logreg = LogisticRegression().fit(X_train, y_train)
         print("Training set score: {:.3f}".format(logreg.score(X_train, y_train)))
         print("Test set score: {:.3f}".format(logreg.score(X_test, y_test)))

           Training set score: 0.781
           Test set score: 0.771
```

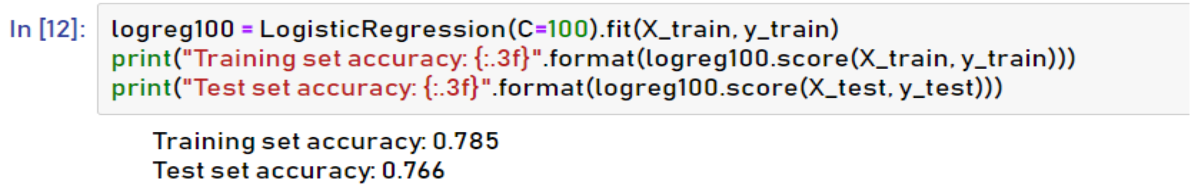
When the logistic regression algorithm is applied without the regularization parameter C, we get the training set accuracy as 78.1% and the test set score as 77.1%.

Next, the regularization parameter is considered: C=1 – the default value. Parameter C = 1/λ. Lambda (λ) controls the exchange off between enabling the model to build it's intricacy as much as it needs with endeavoring to keep it straightforward. The value of Lambda can determine overfitting and underfitting in our model. For instance, if Lambda is very low or equivalent to zero, the model will have enough capacity to expand its intricacy (overfit) by allocating enormous qualities to the loads for every parameter.



```
In [11]: logreg001 = LogisticRegression(C=0.01).fit(X_train, y_train)
         print("Training set accuracy: {:.3f}".format(logreg001.score(X_train, y_train)))
         print("Test set accuracy: {:.3f}".format(logreg001.score(X_test, y_test)))

           Training set accuracy: 0.700
           Test set accuracy: 0.703
```

In the above case, we receive lower scores for both the test case and the training set. So in the next attempt, we take the value to C=100.



```
In [12]: logreg100 = LogisticRegression(C=100).fit(X_train, y_train)
         print("Training set accuracy: {:.3f}".format(logreg100.score(X_train, y_train)))
         print("Test set accuracy: {:.3f}".format(logreg100.score(X_test, y_test)))

           Training set accuracy: 0.785
           Test set accuracy: 0.766
```

When we consider C=100, the test scores are quite commendable but training set accuracy is a bit lower than the test case accuracy. This affirms that less regularization and a more intricate model may not sum up superior to any default setting.  Hence we can conclude that we choose the default parameter value as the main calculating value.

We should envision the coefficients learned by the models with the three distinct settings of the regularization parameter C. More grounded regularization (C=0.001) pushes coefficients increasingly more toward 0. Assessing the plot all the more intently, we can likewise observe that highlight "DiabetesPedigreeFunction", for C=100, C=1 and C=0.001, the coefficient is certain. This shows high "DiabetesPedigreeFunction" include is identified with an example being "diabetes", notwithstanding which model we take a gander at.

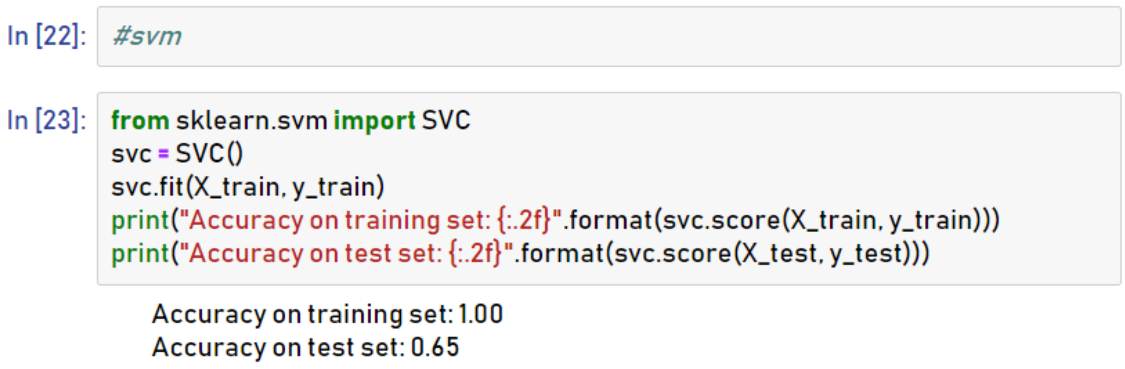
### 4.1.6 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally characterized by an isolating hyperplane. As such, given a supervised and labeled dataset for training (regulated learning), the technique yields an ideal hyperplane which arranges new precedents. In 2D space, this hyperplane is a line isolating a plane in two sections wherein each class lays in either side. This approach can again be used for both – classification and regression, though mostly for classification.

We plot every datum thing as a point in n-dimensional space (where n is a number of parameters in the dataset) with the estimation of each element being the estimation of a specific coordinate. Later classification can be performed by spotting the hyperplane that separates the given classes nicely.

The reasons why this algorithm has been considered for this field are: SVM works actually well with the clear edge of partition, is highly efficient in places where no. of dimensions is larger than data entries, and It utilizes a subset of preparing focuses in the decision function, hence it is likewise memory proficient.

Implementation of SVM:

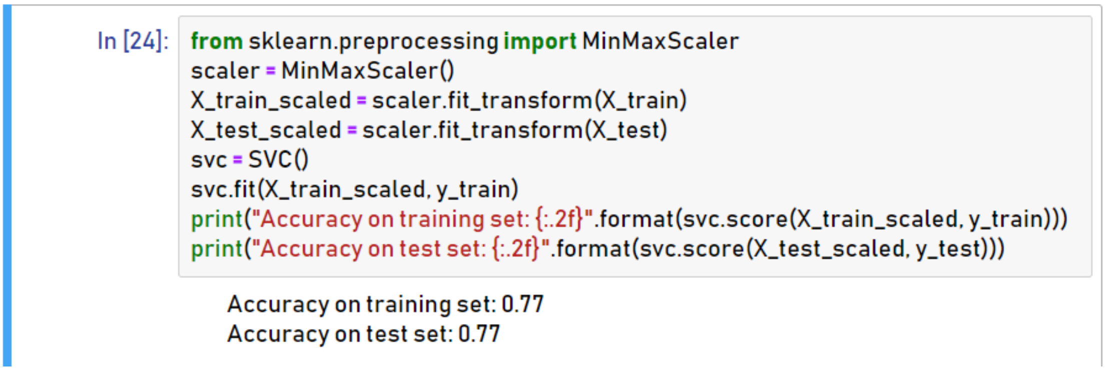


```
In [22]: #svm
```

```
In [23]: from sklearn.svm import SVC
         svc = SVC()
         svc.fit(X_train, y_train)
         print("Accuracy on training set: {:.2f}".format(svc.score(X_train, y_train)))
         print("Accuracy on test set: {:.2f}".format(svc.score(X_test, y_test)))

         Accuracy on training set: 1.00
         Accuracy on test set: 0.65
```

Here we observe that we obtain a perfect training set score whereas the test set gives only 65% accurate results.

From this, we can incur that our features are not aligned on a similar scale. Tuning parameters esteem for AI calculations viably improves the model execution. So we re-scale all these features:

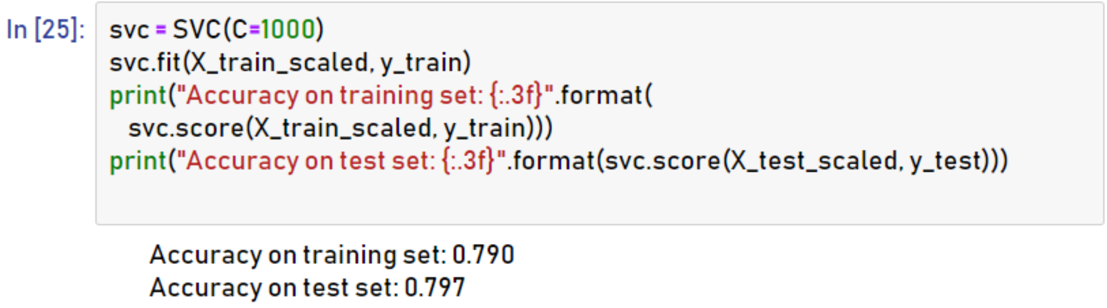


```
In [24]: from sklearn.preprocessing import MinMaxScaler
         scaler = MinMaxScaler()
         X_train_scaled = scaler.fit_transform(X_train)
         X_test_scaled = scaler.fit_transform(X_test)
         svc = SVC()
         svc.fit(X_train_scaled, y_train)
         print("Accuracy on training set: {:.2f}".format(svc.score(X_train_scaled, y_train)))
         print("Accuracy on test set: {:.2f}".format(svc.score(X_test_scaled, y_test)))

         Accuracy on training set: 0.77
         Accuracy on test set: 0.77
```

Since SVM streamlining happens by limiting the decision vector w, the ideal hyperplane is affected by the size of the information highlights and it's along these lines suggested that information be institutionalized preceding SVM model preparing.

Scaling the data parameters had an immense effect! Presently we are really underfitting, where preparing and test set execution are very comparable yet less near 100% precision. So now we will probably increment the value of gamma or C to set into a more intricate model.

Greater the estimated value of gamma will attempt to correct fit the according to pre the data set for example speculation mistake and cause an over-fitting issue.



```
In [25]: svc = SVC(C=1000)
         svc.fit(X_train_scaled, y_train)
         print("Accuracy on training set: {:.3f}".format(
           svc.score(X_train_scaled, y_train)))
         print("Accuracy on test set: {:.3f}".format(svc.score(X_test_scaled, y_test)))
```

```
Accuracy on training set: 0.790
Accuracy on test set: 0.797
```

When we set the value of C=100, the model improves and ends up in giving out 79.7% test set precision.

### 4.1.7 Gradient Boosting

“Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.”

The Gradient boosting algorithm and predict unknown mappings between the inputs. Such algorithms are usually trained on datasets where the input and output pairs are already given.

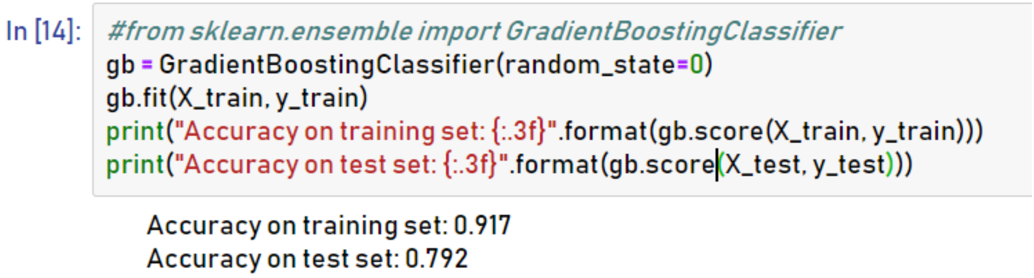
This algorithm follows the ensemble learning notion where various learners can be prepared/ trained to take care of a similar issue to get a better prediction that can possibly be attained by the use of any prevalent algorithm. Of clinical pertinence are a few reports exhibiting that group learning order models are exact in foreseeing result in an assortment of clinical settings.

Let us suppose that we have MSE(Mean squared error) as loss which can be defined as below:

$$Loss = MSE = \sum (y_i - y_i^p)^2$$

where, $y_i$ = ith target value, $y_i^p$ = ith prediction, $L(y_i, y_i^p)$ is Loss function

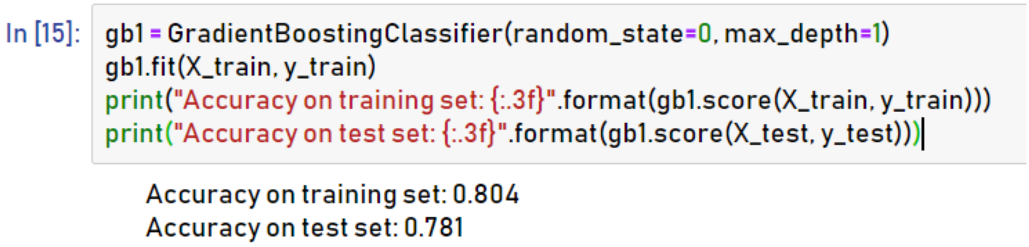
Implementation of Gardient Boosting Algorithm:



```
In [14]: #from sklearn.ensemble import GradientBoostingClassifier
         gb = GradientBoostingClassifier(random_state=0)
         gb.fit(X_train, y_train)
         print("Accuracy on training set: {:.3f}".format(gb.score(X_train, y_train)))
         print("Accuracy on test set: {:.3f}".format(gb.score(X_test, y_test)))
```

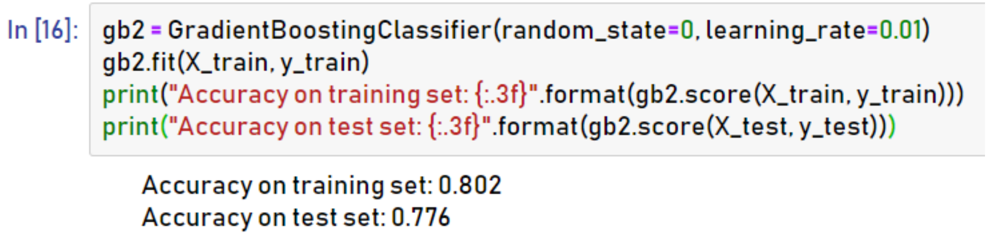
```
Accuracy on training set: 0.917
Accuracy on test set: 0.792
```

The above code gives us an accuracy of 91.7% on the training set and 79.2% on the test set. There is large difference between the two values which indicates that there is probably over-fitting in out model. To diminish overfitting, we could either apply more grounded pre-pruning by restricting the most extreme profundity or lower the learning rate:



```
In [15]: gb1 = GradientBoostingClassifier(random_state=0, max_depth=1)
         gb1.fit(X_train, y_train)
         print("Accuracy on training set: {:.3f}".format(gb1.score(X_train, y_train)))
         print("Accuracy on test set: {:.3f}".format(gb1.score(X_test, y_test)))
```

```
Accuracy on training set: 0.804
Accuracy on test set: 0.781
```



```
In [16]: gb2 = GradientBoostingClassifier(random_state=0, learning_rate=0.01)
         gb2.fit(X_train, y_train)
         print("Accuracy on training set: {:.3f}".format(gb2.score(X_train, y_train)))
         print("Accuracy on test set: {:.3f}".format(gb2.score(X_test, y_test)))
```

```
Accuracy on training set: 0.802
Accuracy on test set: 0.776
```

The two strategies for diminishing the model multifaceted nature decreased the preparation set precision, true to form. In any case, for this situation, none of these techniques expanded the speculation execution of the test set.

## 4.2 Results

The table below lists all the recorded values of the outputs that were obtained throughout the study. The first column contains the name of the algorithm. The Factors column lists the factors that were considered in that particular algorithm to vary the results. The last two columns list the training and test set accuracy scores respectively.

| **Algorithm** | **Factors** | **Training Set Accuracy** | **Test Set accuracy** |
|---|---|---|---|
| **Decision Trees** | | 1.00 | 0.722 |
| | Depth=3 | 0.773 | 0.740 |
| **Naïve Bayes** | | 0.7635 | 0.7466 |
| **KNN** | N=9 (neighbours) | 0.79 | 0.78 |
| **Random Forest** | | 1.00 | 0.786 |
| **Logistic Regression** | (Regularization Parameter) C=1 | 0.781 | 0.771 |
| | C=0.01 | 0.700 | 0.703 |
| | C=100 | 0.785 | 0.766 |
| **Support Vector Machine** | (Overfitting) | 1.00 | 0.65 |
| | (underfitting) | 0.77 | 0.77 |
| | C=1000 | 0.790 | 0.797 |
| **Gradient Boosting** | (Overfitting) | 0.917 | 0.792 |
| | Max_depth=1 | 0.804 | 0.781 |
| | Learnng_rate=0.01 | 0.802 | 0.776 |
| | | | |

Table 1 Results

# Chapter 5
# Conclusion

## 5.1 Conclusion

Our study reveals the effectiveness of each algorithm used. No doubt not one but many algorithms have proved to be equivalent in giving out the almost equivalent accuracies for our problem statement and the field of research. But the best outcomes have been portrayed by the Support Vector Machine Algorithm (SVM) which gives us the accuracy of 79.7% when we set C=1000. Next in the line we have the Gradient Boosting , Random Forest and the K-nearest neighbours algorithms which give us the prediction accuracies of 78.1%, 78.6% and 78% respectively which almost equals 78% in each case. We see that the Naïve Bayes classifier and the Decision tree approach did not prove to be much effective as they give is the outcomes of merely 74%(approx.). We rehearsed a wide cluster of AI models for classification and relapse, what their preferences and hindrances are, and how to control model intricacy for every one of them. We saw that for a large number of the calculations, setting the correct parameters is significant for good execution. If we plot the feature importance of each of the used algorithms we will observe that the importance of each feature differs. For some algorithm, glucose is the most important feature while for the other, it does not hold that level of importance.

## 5.2 Future Scope

The themes we investigated recommend that expectation and avoidance are as of now being rejuvenated and fortified by ML applications, while "security and disappointment location" has been less widely looked into. As I would see it, analysts in this field should keep on exploiting the most recent upgrades in ML and to join them with advancement. Different examinations announced exact expectation and discovery devices that guarantee to improve the executives assets for present and future treatments.

When all is said in done, the most striking advances in the utilization of AI systems originate from information driven techniques that gain from huge datasets. The capacity to gather data from individual diabetic patients has prompted a move in diabetes the board frameworks; likewise, frameworks that need access to profitable information will confront generous obstacles.

The expanded accessibility of digitized wellbeing information from diabetic populaces, alongside the developing utilizations of AI and research patterns, for example, the AP and customized medication, recommends that we are advancing toward another worldview for the board of diabetes. This new viewpoint proposes to accomplish custom conveyance of diabetes care while fitting proficient practices, therapeutic choices, and medicines to singular patients. Then again, the incorporation of clever calculations in basic leadership has moral ramifications that ought to be tended to by doctors and researchers.

The moral dangers related with the arrival of individual information ought to likewise be explored. For instance, the undeniably visit utilization of wellbeing applications and the potential utilization of devices dependent on AI by insurance agencies could prompt segregation or the rejection (or both) of certain residents from wellbeing administrations. Research in this field should proceed and should look to find the chances and preferences of applying AI techniques in diabetes the board that separate these methodologies from other established methodologies.

# References

1. Fatima, M. and Pasha, M., "Survey of Machine Learning Algorithms for Disease Diagnostic." Journal of Intelligent Learning Systems and Applications, (2017), 9, 1-16.

2. K. Kourou et al./Computational and Structural Biotechnology Journal 13 (2015), 8–17

3. Acknowledgment for the dataset: Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265).

4. Do we Need Hundreds of Classifiers to Solve Real-World Classification Problems? Journal of Machine Learning Research 15 (2014) 3133-3181

5. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdino, "Dropout: A Simple Way to Prevent Neural Networks from Overftting", Journal of Machine Learning Research 15 (2014) 1929-1958

6. I. Rish, "An empirical study of the naive Bayes classifier", T.J. Watson Research Center(2001)

7. Lakshmana Ayaru , Petros-Pavlos Ypsilantis , Abigail Nanapragasam, Ryan Chang-Ho Choi, Anish Thillanathan, Lee Min-Ho , Giovanni Montana. Prediction of Outcome in Acute Lower Gastrointestinal Bleeding Using Gradient Boosting
8. Clare Martin, Antonio Martinez-Millana, Andrew Stranieri, Klerisson Paixao, Maurice Mulvenna, and Francisco Nuñez-Benjumea. "Clare Martin, Antonio Martinez-Millana, Andrew Stranieri, Klerisson Paixao, Maurice Mulvenna, and Francisco Nuñez-Benjumea" (2018)
9. The Naive Bayes Algorithm in Python with Scikit-Learnstackabuse.com
10. What is K-Nearest Neighbor (K-NN)? - Definition from Techopediawww.techopedia.com
11. What is Machine Learning? A definition - Expert Systemwww.expertsystem.com
12. https://www.researchgate.net/publication/271850951
13. Introduction to Decision Trees (Titanic dataset) | Kagglewww.kaggle.com
14. How to explain gradient boosting https://explained.ai/gradient-boosting/
15. Logistic Regresson- https://towardsdatascience.com/tagged/logistic-regression
16. Support Vector Machine- https://towardsdatascience.com/tagged/svm