## CHAPTER -1

### INTRODUCTION

1.1 GENERAL

In underdeveloped nations, diabetes is one of the main causes of death. The government and private citizens are funding research projects in an effort to find a cure for the serious illness. Due to a shortage of insulin, which impacts blood sugar metabolism, diabetes is a condition in which blood sugar levels keep rising. Patients with diabetes are unable to efficiently convert the carbs they consume into the glucose sugar that provides energy for daily activities.

The most prevalent type of diabetes is type 1, which develops when the body does not create enough insulin. Low insulin production is a common occurrence in the diabetic population due to immune system attack and loss of pancreatic function. According to the research, both children and adults can get this type of diabetes. To ensure their continuing existence on this world, they must ingest enough insulin. The most important risk factors for type 1 diabetes are pancreatic disease, pancreatic infection, and family history. The next stage of diabetes, type 2, develops when the body's insulin is improperly utilised.

Different ML techniques can be used with various data structures. This study looks at predictive modelling in the healthcare industry. Healthcare data sets are analysed using ML algorithms. The focus of this experiment is gestational diabetes in the research. On the Pima Indian Diabetes Database (PIDD) data set, KNN, SVM, logistic regression, and random forest ML approaches are used to examine the prediction of diabetes. In order to obtain precision, this test is performed using a number of factors, including blood pressure, BMI, and glucose .

The most recent advancement in ML has improved the computer system's ability to recognise and categorise images, forecast diseases, and enhance decision-making through data analysis. The goal of machine learning applications is to teach the computer to perform tasks more effectively than a human. The model is trained using a supervised learning technique, and it is then evaluated using test data.

1.2 STATE OF THE ART

In the world of medicine, ML algorithms are well-known for their ability to predict disease. In an effort to produce the best and most accurate findings possible, many researchers have used ML approaches to predict diabetes .

Multiple classifiers, including SVM, J48, K-Nearest Neighbors (KNN), and Random Forest, were utilised by Kandhasamy and Balamurali [1]. A dataset from the UCI repository was used for the categorization (for more details see Table A4). Based on the accuracy, sensitivity, and specificity values, the classifiers' outputs were compared. Using 5-fold cross validation, the classification was carried out in two scenarios: with and without preprocessing of the dataset. The pre-processing procedure that was used on the dataset was not described by the authors; they merely said that the noise had been eliminated. According to their findings, KNN (k = 1) and Random Forest classifiers had the best accuracy rates of 100%, while the decision tree J48 classifier had the highest accuracy rate of 73.82% without pre-processing.

Furthermore, Deepti and Dilip [2] detected diabetes using Decision Tree, SVM, and Naive Bayes classifiers. Finding the classifier with the maximum accuracy was the goal. For this investigation, the Pima Indian dataset was utilised. Cross-validation on 10 folds is used to partition the dataset. The preprocessing of the data wasn't discussed by the authors. Accuracy, precision, recall, and the F-measure were used as metrics to assess performance. The Naive Bayes model had the highest level of accuracy (76.30%).

Negi and Jaiswal [3] sought to apply the SVM to predict diabetes in addition to the other research. The Diabetes 130-US and Pima Indians datasets were integrated to create a single dataset. Given that many other studies only used one dataset, the goal of this study was to confirm the accuracy of the findings. 64,419 of the 102,538 samples in the dataset were positive, while 38,115 were negative. The dataset consists of 49 attributes and 102,538 samples. The characteristics employed in this study were not discussed by the authors. The dataset is pre-processed by converting the non-numerical values to numerical values, replacing missing values and out-of-range data with zero, and normalising the data between 0 and 1. Before the SVM model was applied, various feature selection techniques were employed. While the Wrapper and Ranker methods (from the Weka Tool) picked nine and twenty characteristics, respectively, the Fselect script from the LIBSVM package only selected four attributes. The authors employed a 10-fold cross validation technique for the validation process. The diabetes prediction may be more accurate, at 72%, when using a combined dataset.

Six distinct classifiers were utilised by Mercaldo et al. [4]. J48, Multilayer Perceptron, Hoeffding Tree, JRip, BayesNet, and RandomForest are the classifiers. For this study, the Pima Indian dataset was also used. Although the authors didn't mention a pretreatment step, they used the GreedyStepwise and BestFirst algorithms to identify the discriminating characteristics that improve classification performance. Body mass index, plasma glucose concentration, diabetic pedigree function, and age are the four criteria that have been chosen. The dataset is validated via a 10 fold cross. Based on the importance of the precision, recall, and F-Measure, the classifiers were compared. The Hoeffding Tree algorithm produced a precision value of 0.757, recall value of 0.762, and F-measure value of 0.759.

1.3 OBJECTIVES

Objective is to create a machine learning model which may predict the diabetes using various machine learning techniques and gives the best result.

1.4 ORGANIZATION OF THE REPORT

The organization of the thesis is as follows. Chapter 1 gives an insight of our objective and Literature survey. In main content, Chapter 2 deals with the basics required to understand the concept of forecasting and its various methods. Chapter 3 is the description of the models used in the forecasting. Chapter 4 consists of the work done in forecasting, its results and outputs. Chapter 5 sums it all giving an overview of what is done, what is to be done and how the accuracy can be improved by considering economic factors.

## CHAPTER 2

### Data Pre-processing

2.1 Introduction

The adjustments made to our data prior to feeding it to the algorithm are referred to as pre-processing. Data pre-processing is a method for transforming unclean data into clean data sets. In other words, anytime data is acquired from various sources, it is done so in a raw manner that makes analysis impossible.

2.2 Need of Data Pre-processing

* The format of the data in machine learning projects must be correct in order to get better results from the applied model. For example, the Random Forest algorithm does not tolerate null values, hence null values must be handled from the original raw data set in order to execute the Random Forest algorithm. Some specific Machine Learning models require data in a specific format.
* Another consideration is that the data set should be organised so that many Machine Learning and Deep Learning algorithms can run simultaneously and the best one is selected.

2.2.1 Rescale Data

* Many machine learning methods can benefit from rescaling the attributes to have the same scale when our data contains attributes with different scales.
* This is helpful for optimization techniques like gradient descent, which are at the heart of machine learning methods.
* It is especially helpful for algorithms like regression and neural networks that weight inputs, as well as for algorithms like K-Nearest Neighbors that require distance measurements.
* Using the MinMaxScaler class from scikit-learn, we can rescale your data.

2.2.2 Binarize Data(Make Binary)

* Our data can be transformed using a binary threshold. The threshold is recorded as 1, and all values that are equal to or below it are marked as 0.
* This is referred to as binarizing or thresholding your data. When you wish to give clear values to probabilities, it can be handy. It is also helpful when adding new features that convey important information during feature engineering.
* Using scikit-Binarizer learn's class and new binary attributes, we can build new binary attributes in Python.

2.2.3 Standardize Data

* In order to convert attributes with a Gaussian distribution and varying means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1, standardisation is a useful technique.
* Scikit-StandardScaler Learn's class allows us to standardise data.

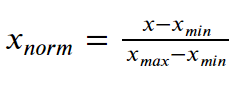
2.3 Processing techniques used in the model

2.3.1 Removing the null value

* By finding the null value in the data frame and replacing those value with the mean value of that particular attribute by using the function SimpleImputer from the scikit learn module of tensorflow.
* The scikit-learn class SimpleImputer is useful for handling the missing data in the dataset for the prediction model. It inserts a designated placeholder in lieu of the NaN values.
* The SimpleImputer() method, which accepts the following inputs, is used to implement it.

2.3.2 Normalising the data

* When the various characteristics (variables) are on a smaller scale, machine learning algorithms typically perform better or converge more quickly. Therefore, normalising the data before training machine learning models on it is standard practise.
* Additionally, normalisation reduces the sensitivity of the training process to the magnitude of the characteristics. As a result, following training, the coefficients improve.
* Feature scaling is the process of rescaling features to make them more training-friendly.
* The Normalization formula is provided below:



<https://journaldev.nyc3.digitaloceanspaces.com/2020/10/Normalization.png>

2.4 Splitting the dataset

The core of every ML problem is data. ML models are like bodies without a soul if they are not fed with the right data. But gathering data is no longer a significant issue in the 'big data' world of today. Every day, we consciously or unconsciously produce enormous datasets. Having an abundance of data available, however, does not make the issue go away. We need to feed massive amounts of high-quality data into ML models in order for them to produce accurate results.

The quality data is useless unless it is used effectively, even though extracting meaning from raw data is an art in and of itself and necessitates solid feature engineering skills and domain expertise (in special circumstances). How to divide the data for training and testing is the main issue that ML/DL practitioners encounter. Even though it initially appears to be a straightforward issue, only by delving deeply into it can its complexity be determined. Inaccurate training and testing sets may have unanticipated consequences on the model's output. It could result in the data being overfitted or underfitted, and our model could end up producing biassed results.

2.4.1 How to divide the data then?

Ideally, the data should be split into three sets: a train set, a test set, and a holdout cross-validation or development (dev) set. Let's first briefly discuss the meaning of these sets and the kinds of data they ought to contain.

* Train Set: The data that would be fed into the model would be in the train set. Simply put, our model would gain knowledge from this data. For instance, a regression model could uncover gradients to lower the cost function by using the instances in this data. These gradients will then be applied to lower costs and improve data prediction.
* Dev Set: The trained model is validated using the development set. As the foundation for our model evaluation, this setting is the most crucial. If there is a significant disparity between the error on the training set and the error on the development set, the model is over-fitting and has a high variance.
* Exam Set: The data used to test the trained and approved model are in the test set. It reveals the effectiveness of our entire model and the likelihood that it will forecast an illogical event. Numerous assessment criteria (such as precision, recall, accuracy, etc.) can be utilised to assess the effectiveness of our approach.

## CHAPTER 3

### Proposed method

3.1 **INTRODUCTION**

For data analysis, the suggested method has been utilised with the Anaconda tool (AEN 4.1 Version). The package versions for predictive analysis and data management are managed via the anaconda package management system [28]. those with diabetes disease—have been used as inputs. These datasets were chosen because they share characteristics with each other in real life and because there is some correlation between them. These data are loaded and examined to determine whether there are any missing values. Any missing values are replaced with null values if they are discovered. After that, it was determined if any of the data's columns were related to any other column in the data on their own. One of the columns is eliminated if any correlation between the two is discovered. If data has either a true or false value, they are replaced with 1 and 0, accordingly. The original data has been divided into training data, which contains 70% original data, and test data, which contains 30% original data.

To determine how many instances in the original, practise, and test data are true and false The Naive Bayes method has been trained on our different class datasets, and it has been used to calculate the accuracy of the classes individually using confusion matrix [29]. Fig. displays the proposed method's block diagram. 1. The accuracy calculation performance metrics are displayed separately for each kind of data in the performance report. In a similar manner, it trained the random forest algorithm with our three separate class data. The algorithm uses a confusion matrix to calculate the results accuracy and presents the three classifications separately. The accuracy calculation performance metrics are displayed separately for each kind of data in the performance report. Every epoch has one or more batches, which change internal model parameters. The epoch can be used to reduce dataset error rates.

For each class, a few example test results have been collected independently. We can determine whether or not the sample data are associated with the disease by applying them to each trained model of it. One can see that the model trained with Random forest provides the most accurate classification results when comparing the results of both models for each type of data. The trained data are compared individually with the suggested algorithms in order to determine the efficacy of the procedure, and the performance of the test data is also examined. The suggested approach can be used to test real-time disease data for classification and to determine whether a patient has a specific disease or not.

3.2 PROPOSED MODEL

Dataset

Pre-processing

1.Min-Max scaler

2. Replacing missing value with mean.

Classification algorithm such as SVM, Decision tree and Logistic Regression have been used

Result

2.2 PROPOSED ALGORITHM

2.1.1 Logistic Regression

Essentially, supervised categorization is what logistic regression does. For a specific collection of features (or inputs), X, the target variable (or output), y, can only take discrete values in a classification issue.

Contrary to popular assumption, a regression model is a logistic regression. In order to determine the likelihood that a specific data entry falls under the category designated by the number "1," the programme creates a regression model. Logistic regression models the data using the sigmoid function, just like linear regression assumes that the data follows a linear distribution.



https://media.geeksforgeeks.org/wp-content/uploads/20190522162153/sigmoid-function-300x138.png

2.1.2 Decision Tree

The most effective and well-liked technique for categorization and prediction is the decision tree. A decision tree is a type of tree structure that resembles a flowchart, where each internal node represents a test on an attribute, each branch a test result, and each leaf node (terminal node) a class label.

Glucose

low

High

Normal

Age

Yes

BMI

Old

Young

Normal

High

Yes

No

Yes

No

Decision Tree Representation: Decision trees classify instances by ordering them from the root of the tree to a leaf node, which then indicates the instance's categorization. As seen in the above diagram, to classify an instance, one tests the attribute given by the root node of the tree before continuing down the branch of the tree that corresponds to the attribute's value. The subtree rooted at the new node is then subjected to the same procedure once more.

According to whether a specific morning is ideal for diabetes prediction, the decision tree in the above diagram assigns a classification to each leaf and returns that classification (in this case Yes or No).

2.1.3 Support Vector Machine

One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues.

The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary.

SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. Consider the diagram below, where a decision boundary or hyperplane is used to categorise two distinct categories:

|  |
| --- |
| Support Vector Machine Algorithm |

https://static.javatpoint.com/tutorial/machine-learning/images/support-vector-machine-algorithm.png

## CHAPTER 4

RESULT AND DISCUSSION

3.1 INTRODUCTION

In this chapter we will discuss the working procedure of the model and at last we will see the accuracy gain from the use of different algorithm .

3.2 DATA SET

3.2.1 Data set description

The Pima Indian Diabetes Dataset, a UCI repository, is where the information came from. The collection contains a lot of information about 768 patients.

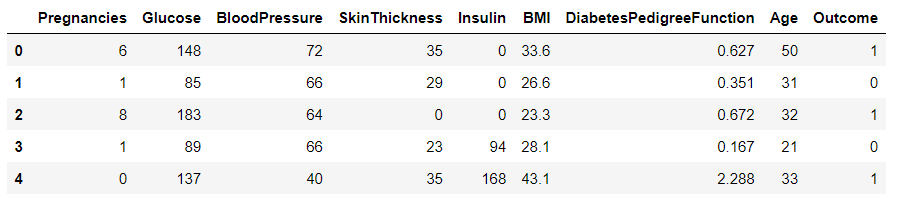
|  |  |
| --- | --- |
| S.NO | ATTRIBUTE |
| 1 | PREGNANCY |
| 2 | GLUCOSE |
| 3 | BLOOD PRESSURE |
| 4 | SKIN THICKNESS |
| 5 | INSULIN |
| 6 | BMI |
| 7 | DIABETES PEDIGREE FUNCTION |

3.2.2 Graphical representation of the dataset

Here the graphical representation of the dataset has been shown this consist of 500 negative dataset and 268 value of the positive value sample.

3.3.3 Representation in the form of DataFrame

Here the dataframe representation of the above dataset is being displayed using the pandas function.



3.3 FILLING IN THE MISSING VALUE

Any data that is missing a value indicates that the event did not occur, the data were unavailable, or the data were not applicable. Here, null values have been used to replace any missing or unavailable information.

3.3.1 Value consisting of value as 0 or null value

Below diagram shows all the attribute rows which consist of the value ‘0’ as well as value null .

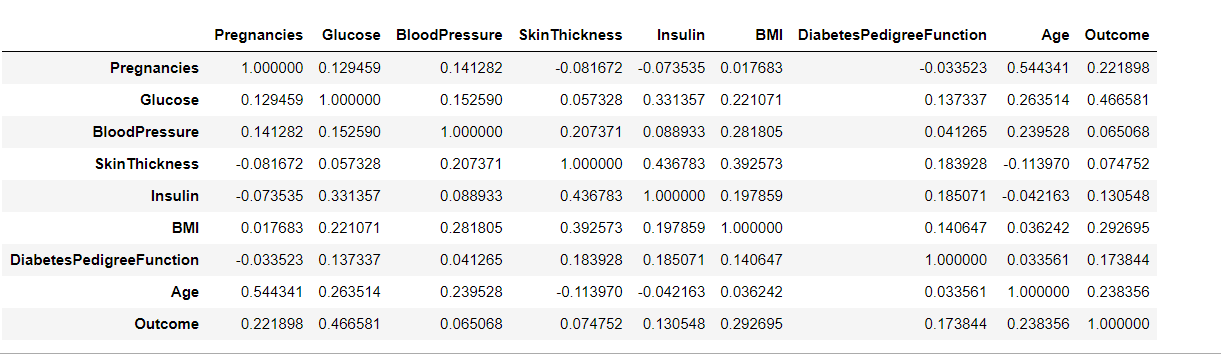
The above graph represents the attribute value which are either 0 or null so as shown in the graph we can see that the total no of rows in the dataset are 768 and that of column are 8 out of that insulin having the 374 column which is either 0 or null value ,pregnancies having 35 rows which consist of 0 or null value glucose attribute consist of 5 ‘0’ or null value and the blood pressure column consist of the 35 rows ,skinthickness consist of the 227 rows having the value as 0 or null , BMI (body mass index) this column consist of the 11 0 or null value ,Diabetes pedigree function and age having the 0 rows which consist of the ‘0’ or null value.

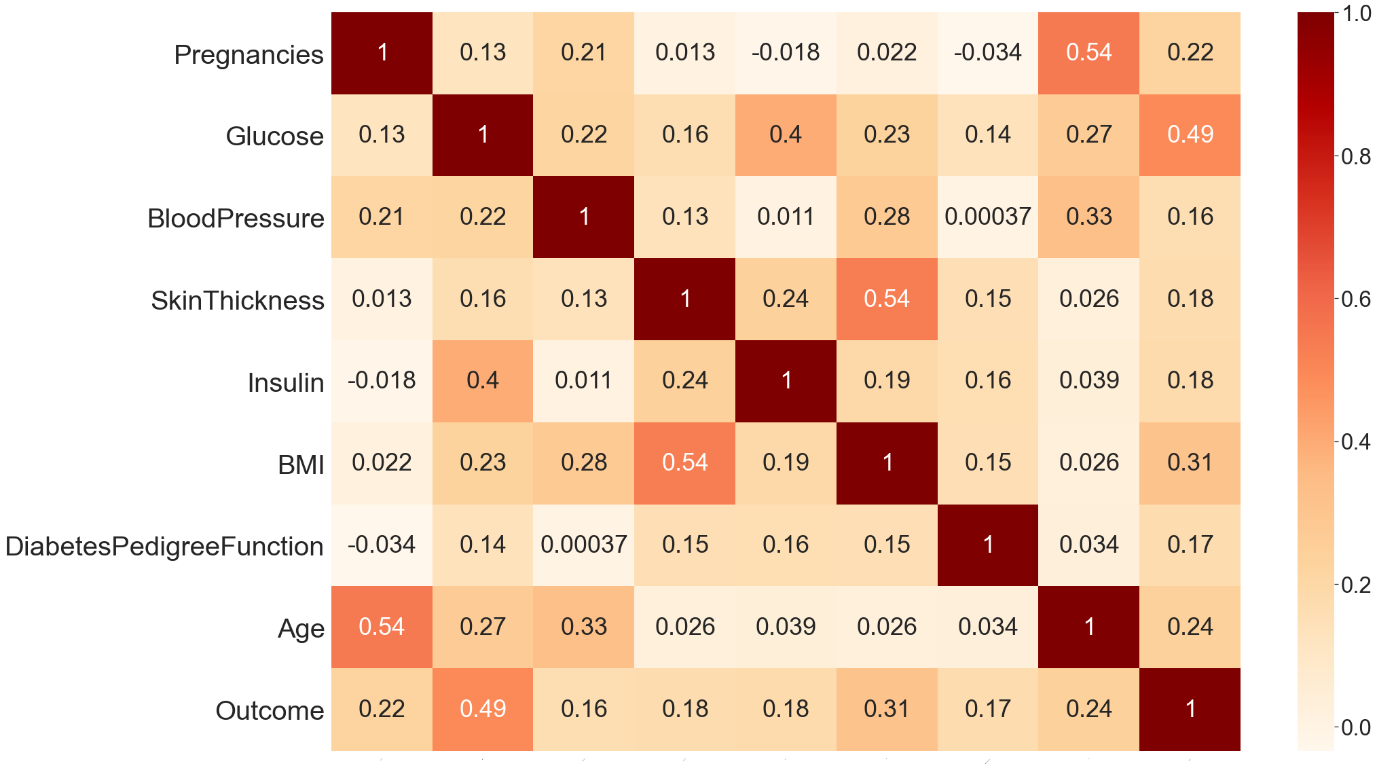
3.4 Correlation

A statistical measure called correlation shows how much two or more variables fluctuate in connection to one another. When two variables rise or decrease simultaneously, there is a positive correlation; when there is a negative correlation, one variable increases as the other falls.

3.4.1 correlation coefficient

In statistics, correlation coefficients are used to assess how closely two variables are related. It is a metric used in statistics to express the linear relationship between dependent and independent variables. It is symbolised by the letter "r" in lower case. Here, the correlation between each dataset's columns is determined to determine how closely related they are (Table 1). The outcomes provide the value of each column's correlation with every other column in the dataset. One of the columns in a dataset is eliminated to prevent value duplication if two columns in the dataset have identical associated values.





As stated in the above discussion the value of corrlation function is as close to 1 two attribute are highly corelated so from the diagram ploted above we can say that the corelation between BloodPressure and output is lowest i.e(0.16) and that of blood pressure and glucose is maximum i.e (0.49) except that of outcome and outcome i.e (1) because both the outcome and outcome value are same so they are highly corelated so from here we can say that the outcome is more dependent on the value of glucose concentration rather than that of Blood Pressure.

3.5 Result from the different algorithm

Now from here onwards we will see the accuracy achieved by different model and their f1 score as well as the precision and the recall of every model used in this classification.

3.5.1 Support Vector Machine

Using the support vector machine the accuracy gain on the training and testing data are as follows:

3.5.2 Decision Tree

Using the decision tree classifier accuracy gain on the training and testing dataset is:

3.5.3 Logistic Regression

Using the Logistic Regression classifier the accuracy of the model is being represented with the help of the bar chart shown below:

3.6 comparing different algorithm

3.7 Confusion Matrix

