IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 2, Ser. I (Mar. – Apr. 2021), PP 21-32
www.iosrjournals.org

Accuracy Comparison of Machine Learning Algorithms for Diabetes Detection

Barkha Mahajan[1], Nupur Achhara[2], Vibhasha Paun[3]

[1,2,3]*(Department of Computer Engineering, Gujarat Technological University, India)*

Abstract:

Background: *Diabetes mellitus (also referred to as "diabetes") falls under the category of metabolic disease elevating blood sugar levels, which over time may cause severe damage to the heart, blood vessels, eyes, kidneys, and nerves. A hormone, Insulin that is produced by pancreas tends to move sugar from the blood to cells in the body to store or use for energy. Diabetes is a medical condition that weakens the body's ability to process blood glucose (known as blood sugar). It happens by either inadequate insulin production (Type-1 diabetes) or Inadequate sensitivity of cells to the action of insulin (Type-2 diabetes). Gestational diabetes, a third type, is a kind of diabetes seen in pregnant women and is associated with both mother and child complications. It puts the mother and her child at a higher risk of developing type-2 diabetes later in their life. Diabetes mellitus (Type-2) sometimes develops to indicate other underlying diseases, such as a genetic syndrome, such as myotonic dystrophy, pancreatic disease, or drugs, such as glucocorticoids. Diabetes (type-2) has been a significant health issue in the past three decades and has risen dramatically in countries of all income levels. Approximately 422 million adults worldwide have diabetes, which corresponds to 1 person in 11 diagnosed with diabetes, and 1.6 million deaths are directly attributed to diabetes each year (data from World Health Organisation (WHO)). The International Diabetes Federation (IDF) predicts the figures to reach 628 million by 2045. Over that, around 79 million are pre- diabetic, a condition where the blood glucose is not sufficiently high to be diagnosed and declared as diabetic.*

With increased awareness with a timely and accurate diagnosis, diabetes could be treated more effectively or even be prevented using the already available medication. The same will help select the appropriate treatment and make necessary lifestyle changes as nothing, but early identification is the remedy to avoid complications. For several years now, multiple studies have been conducted for the same with the primary objective to predict what variables cause diabetes at a higher risk and provide a preventive action towards such an individual. Several parameters are considered as described later. To prevent or delay the onset of type-2 diabetes, one should have a healthy diet, maintain healthy body weight, do regular physical activity, and avoid tobacco use.

Materials and Methods: *The statistical and machine learning models come into use for the critical part of diabetes detection/ prediction. Several research pieces show that using various classification algorithms of machine learning approaches is widely successful in the accurate and early diagnosis of diabetes. Using Data Mining alongside Machine learning algorithms gives an extra because of the capability to handle large amounts of data, combine data from various sources, and integrate the study's background information. Medical healthcare systems are rich in information, and the wise use of this data can produce some predictive outcome. Pima Indian women in Arizona have participated in an intensive diabetes study. The observed result has been provided publicly, used in several research papers and studies related to diabetes. In this research, we have used the same (Pima Indian Diabetes Data) for performance analysis of the proposed algorithms. The dataset covers factors responsible for an individual female over the age of 21 years with a varying medical history, to be diabetic along with the received outcome (diabetic(1) or not diabetic(0)).*

Results: *In this work, Artificial Neural Network, Classification and Regression Tree (CART), Logistic Regression and Random Forest machine learning classification algorithms are used and evaluated on the Pima Indian dataset to check the accuracy and performance of prediction of diabetes in a patient. The features in data were used to train an Artificial Neural Network, Classification and Regression Tree (CART), Logistic Regression, and Random Forest algorithm achieving an accuracy of 76.5%, 66.2%, 79.2%, and 75.3%, respectively. The experimental performance of all four algorithms are compared and cross-checked on two software, namely, Rapid Miner and KNIME (version 4.3.0), and the results observed were very comparable.*

Conclusion: *The Logistic Regression Machine Learning algorithm gives the most accurate results, followed by the Artificial Neural Network, , Random Forest, and Decision Tree, respectively.*

Date of Submission: 01-03-2021 Date of Acceptance: 14-03-2021

DOI: 10.9790/0661-2302012132

I. Introduction

Diabetes mellitus(Diabetes) is considered one of the deadliest chronic diseases which cause an increase in sugar level, affecting millions of people worldwide. Many complications occur if diabetes remains unidentified and untreated. So, an early and accurate diagnosis of diabetes has crucial importance for the treatment of patients. The machine learning approach for diagnosing diabetes includes the design of several algorithms that can predict the possibility of diabetes in patients with maximum accuracy. In this study, four well-known classification algorithms that are Artificial Neural Network(ANN), Classification and Regression tree(CART), Logistic Regression(LR) and Random Forest(RF) - approaches are proposed to detect diabetes at an early stage. All four algorithms' performance is evaluated on the Pima Indian diabetes dataset and compared on two data mining platforms: KNIME and RapidMiner. The results obtained show that Logistic Regression outperforms by attaining the highest accuracy of 79.2% compared to other algorithms. In contrast, the accuracy of the other three are as follows:Artificial Neural Network (76.2%), Classification and Regression Tree (66.2%), and Random Forest (75.3%).

II. Material And Methods

The following subsections show how the diabetes prediction model has developed and the tools used for the same. The proposed procedure is summarised in the form of a model diagram, as in the figure below. It demonstrates the flow of the research conducted in constructing the model.

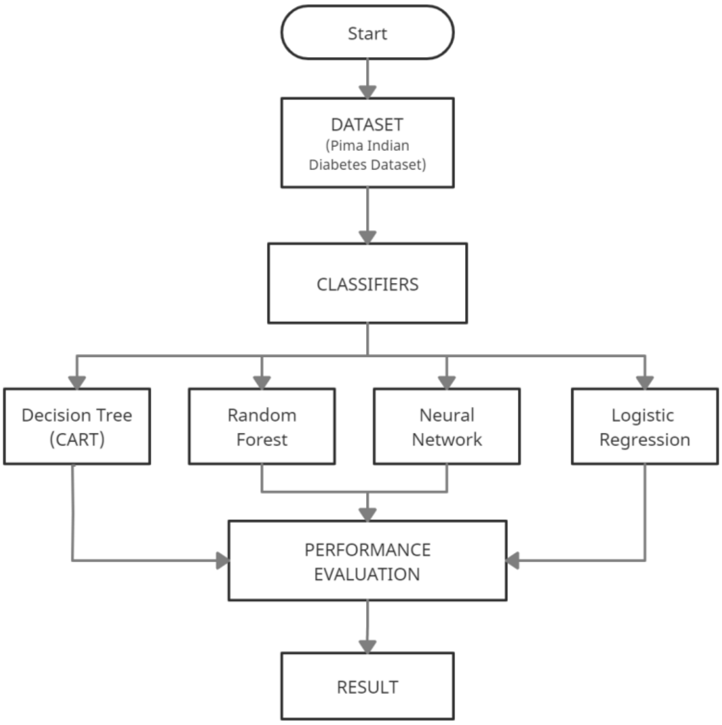


Fig. 1: Overview of the process to test the performance

Pima Indian Diabetes Dataset

The dataset we used for this research paper is obtained initially from the NIDDK (National Institute of Diabetes and Digestive and Kidney Diseases) of the United States. Pima Indian dataset aims to diagnostically predict whether a patient has diabetes, using the diagnostic measurements included in the dataset. During the selection of instances, multiple constraints were put into place. This dataset contains details of female patients aged 21 years and above from Pima Indian Heritage, coming from a population near Phoenix, Arizona, USA. The dataset has 768 records, out of which 268 instances are of 1 (diabetic), and 500 cases are of 0 (not diabetic). Each of the 768 instances has eight numerical attributes, which are medical predictor variables and the outcome(if 1, it is interpreted as "tested positive for diabetes"; if 0, it means "tested negative for diabetes").
The following are the predictor variables in the Pima Indian Diabetes Dataset:

1. Number of times pregnant (Pregnancies),
2. Plasma glucose concentration at two h in an oral glucose tolerance test (Glucose),
3. Diastolic blood pressure (mm Hg) (blood pressure),

4. Triceps skinfold thickness (mm) (SkinThickness),
5. 2-h serum insulin (mu U/ml) (Insulin),
6. Body mass index (BMI),
7. Diabetes pedigree function (DiabetesPedigreeFunction), and
8. Age (years) (Age).

This dataset also contains the Missing Attribute Values, which are handled using features in our selected software.

TABLE 1: Overview of the attributes in the Pima Indian dataset as used in this study

Attribute	**Type**
Pregnancies	Numeric
Glucose	Numeric
BloodPressure	Numeric
SkinThickness	Numeric
Insulin	Numeric
BMI	Numeric
DiabetesPedigreeFunction	Numeric
Age	Numeric
Outcome	0 or 1

Various machine learning models for prediction and classification ought to give accurate predictions to create real value for a given problem. The evaluation of a model should aim to maintain the delicate balance between experimentation, intuition and reckon the generalisation accuracy of a model on given data without being confused by the randomness. A model generalises to new, previously unseen data is an important factor along with training a model.

Evaluation of a machine learning model's performance can be broadly done by using the following methods:

III. Partitioning

Partitioning, also known as a splitting method, is used to estimate machine learning algorithms' performance. Such algorithms function by making data-driven predictions through the train-test split procedure. This procedure divides the original dataset into two most common subsets called training set and test set. The training set is employed to train the model, and the test set, which is independent of the training set, is used to evaluate a final model fit on the training dataset.

The training-test split procedure estimates machine learning algorithm performance on new data (that is not used for model training). If the model is evaluated with the same training data, it serves no purpose as it remembers the training data instead of generalising it. This method provides multiple ways to split the original dataset into training and test dataset and performs best on a sufficiently large dataset.

Working:

This procedure usually divides the whole data into two subsets, i.e., a training set and a test set.

The training set is a set of data taken from the original dataset used during the learning process and trains the model by making probability distributions.

The test set is provided with the remaining unseen data from the original dataset (not used in the training dataset) but follows the same probability distribution as the training set. The Test data elements are provided to the model to make the predictions to estimate the model's accuracy in classifying new data. These predictions are compared to the expected value, which is computed from training set data.

One main configuration parameter considered during splitting two subsets is the train and test sets' size. Different software has different configurations with no minimum split percentage.

The most common split percentages are:

- Train: 80%, Test: 20%
- Train: 67%, Test: 33%
- Train: 50%, Test: 50%

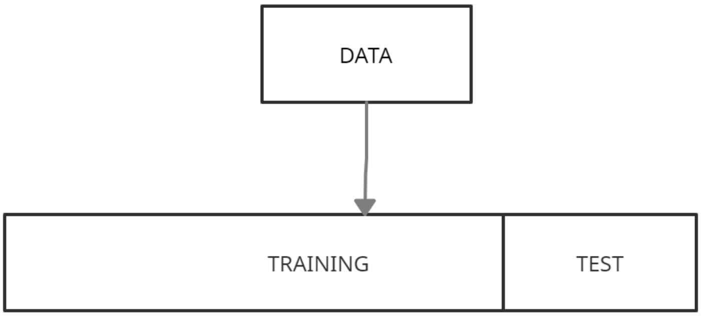


Fig. 2: Working of the partitioning method

IV. Cross-validation

Cross-validation is a kind of model validation method used for assessing how the results of a statistical analysis will generalise to a particular independent dataset. For prediction, a model is usually given a dataset divided into a training set and test set. The training set is given available data to train the model, and a dataset of unknown data(not used for training set) is given to the Test set (or validation set) against which the model is tested. The objective of cross-validation is to test the model's ability to predict the new data, which was not used in estimating it (training set).

There are generally two types of cross-validation techniques:

- Exhaustive: In Exhaustive cross-validation, the method tests on all possible combinations to partition the original dataset into training and test sets.
- Non-exhaustive: In Non-exhaustive, the original dataset is not separated into all possible combinations. It does not compute all ways of splitting the original dataset. E.g., k-fold cross-validation, which we have used to test our selected algorithms.

Working:

One iteration of cross-validation involves the division of the dataset into complementary subsets. The number of subsets is given by 'k-folds,' where k is any natural number. A value k=10 is very common for the prediction models in the field of machine learning.

Steps:

- The original set is partitioned into k equal size subsamples.
- Out of numerous k samples, only one subsample is retained as the validation data, which is utilised for testing the model, and the remaining k-1 subsamples are used as training data.
- For each iteration:
- A different test data set is chosen.
- The remaining subsamples are automatically considered as the training data set.
- Model is trained by a training dataset and evaluates it on the test dataset.
- The result is retained, and the model is discarded.

The same process is then repeated k times (the folds). The k results from the folds are then averaged or combined to produce a single estimation.

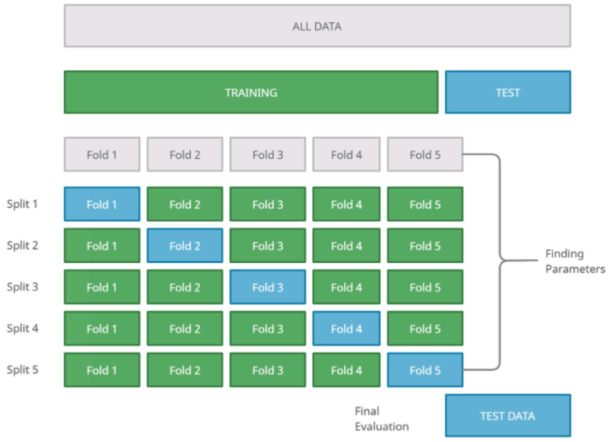


Fig. 3: Working of cross-validation method

Cross-validation performs better than the partitioning method with almost every machine learning algorithm because it allows the model to train on multiple train-test splits, which indicates how well the model will perform on unseen data giving more accurate predictions. Cross-validation enables us to make predictions utilising all the data. In complex machine learning models, using the same data as training sets (like in the partitioning method) in different steps can create problems, leading to the model's reduced prediction accuracy.

Softwares:

To predict the results and evaluate different machine learning algorithms' performances, we used two data mining cum analytics platforms that effectively depict how effectively the chosen algorithms have worked. With the detailed reports about the algorithms' operations provided by the chosen software, we could judge the algorithms' working and choose the best from the ones selected.

KNIME (Konstanz Information Miner)

KNIME (Konstanz Information Miner) is open-source software that helps its users with data analytics, reports, and integration at no cost. This software is based on Java and works on most operating systems. It consists of various machine learning and data mining components that can be exploited to extract insights from data, access and transform data using different tools. It provides a GUI based platform, making it easier to build the workflows and perform almost every kind of analytics. KNIME is considered one of the best ETL (extract, transform, load) tools as it is easy to perform tasks and interpret the results. KNIME has found its business intelligence applications, financial data analysis, text mining, and CRM customer data analysis.

We used KNIME (version 4.3.0) for our research. The simple drag and drop interface makes it easy to complete data flow through various Machine Learning algorithms. In this software, each node is configured and then executed, which implements your task's function. Every time a node is to be executed, all the predecessor nodes should have previously been successfully executed, and hence all data is available at the input ports.

- **Load the data file:** This is the first step, which allows the user to load the data they want to work on from their local device. Properties of the data set (here, Pima Indian Diabetes Dataset) are then specified, and later it is executed.
- **Select the method to train and test data:** Knime offers multiple ways to train the data, including the previously mentioned partitioning and cross-validation method. For our research, we have chosen X-Partitioner, which is a cross-validation model. All nodes passing through this node are executed as often as iterations should be performed (we specified ten iterations). This node was then executed.
- **Normaliser:** In this node, all the values of the selected columns are normalised. We set up two paths from normaliser onwards, one for learning and the other for predicting. This column is necessary as Random Forest, Decision Tree, and Logistic Regression require nominal values. From the multiple methods offered, we used the min-max normalisation on all the columns wherein the maximum value was 1, and the minimum value was 0. The node was then executed.
- **Number to a string:** At this node, the selected columns' numerical values are converted into a string data type. This node was necessary to add to the chain as Random Forest, Decision Tree, and Logistic Regression model required string values in the predicting column. Only the outcome column was converted, and then the node was run for both learning and predicting paths.
- **Learners:** Four learner nodes are added to the number to string (training) node, one for each method. At this node, the classification model for each algorithm is added to the main memory. Then the algorithm learns from the training data. We have the following nodes:
- Decision Tree Learner
- Logistic Regression Learner
- Neural Network (MLP) Learner
- Random Forest Learner

Each of these nodes is configured and then executed.

- **Predictors:** For each algorithm, four predictor nodes are added to the number to string (testing) node with one input from their respective learner. Each algorithm's existing model is used to predict the class value for new patterns at this node. We have the following nodes:
- Decision Tree Predictor
- Random Forest Predictor
- Logistic Regression Predictor
- Neural Network (MLP) Predictor

Each of these nodes is configured and then executed.

- **Scorer:** This is the final node that helps gauge the performance of each node. There is a different scorer for each predictor, which compares two columns by their attribute values and shows the confusion matrix with the number of matches in each cell and the accuracy results. After configuring, the note is executed, and we focussed on the p accuracy results.

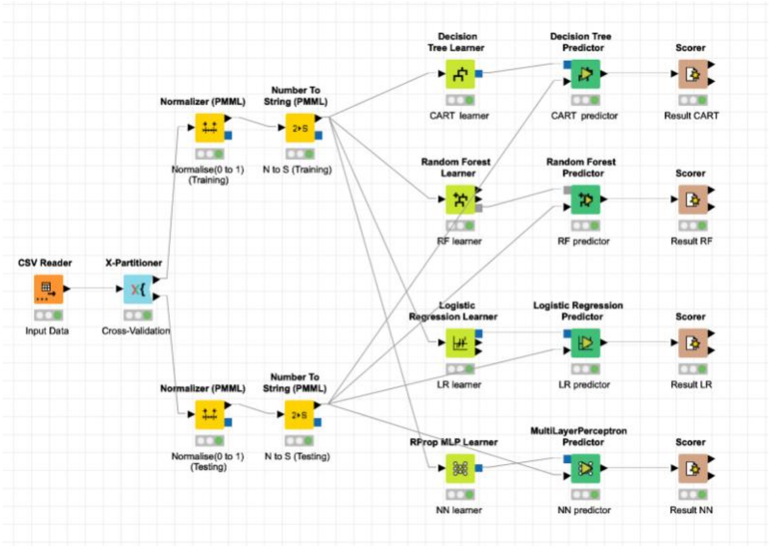


Fig.4: User interface and process overview of KNIME (version 4.3.0)

RapidMiner Studio (Version 9.8)

RapidMiner is a data science platform that can be exploited to perform data mining, text mining, data preparation, machine learning, and predictive analysis. This software package is cross-platform and functions on Java. Moreover, it is an all-in-one tool that offers its user work on raw data, which is then automatically analysed on a large scale. It provides hundreds of innovative data preparation and machine learning algorithms to assist different data mining projects. This platform is easy to use and includes a drag and drop process and the automatic building of different data mining models to compare and choose from the best. RapidMiner can be utilised for business, commercial, research, training, and education purposes due to its robust performance.

To support the results that we obtained from KNIME, we used RapidMiner Studio (Version 9.8). We executed three algorithms and compared their results using one of its extensions, AutoModel. It allows us to build the models and validate them. AutoModel lets us create our process and modify it according to our requirements. It mainly works on three important categories of the problems:

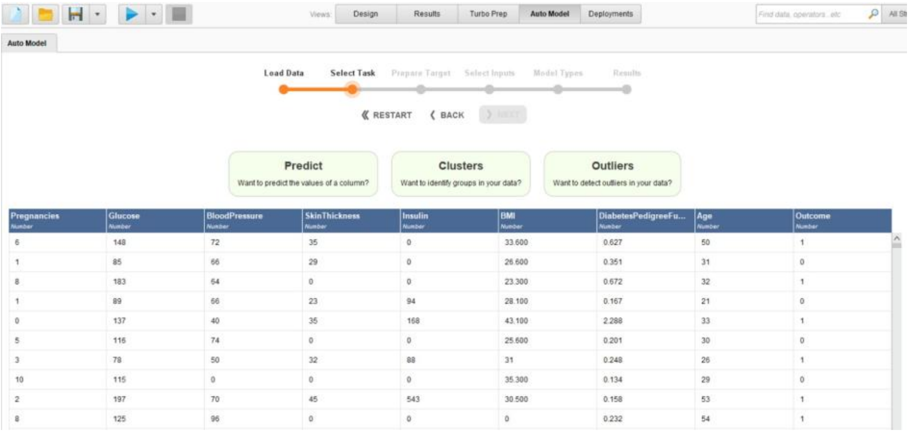
- Prediction: Prediction is used to predict the outcome of the given data.
- Clustering: Clustering helps you to group similar data from the dataset.
- Outliers: This can be selected when you want to detect the data's outliers, which means measuring the distances between the relative points and data densities.

The prediction section solves problems related to classification and Regression. To evaluate the performances of the algorithms, we have used the prediction section. It provided us with the various relevant models for the execution and the comparison of the results. We worked on three algorithms, namely, Decision Tree (CART), Random Forest, and Logistic Regression, for our research. AutoModel has a straightforward process, and it tends to make the task effortless. Below mentioned are the steps that can be followed to validate the models for solving classification and regression problems.

- **Load Data:** This lets the user load the data from the local devices that they want to work on and also provides its datasets as an option for a user to choose.
- **Selecting Task:** The next step involves selecting the category from the three options, i.e., Prediction, Clustering, and Outliers, where we opted to predict whether the patient has diabetes.
- **Preparing Target:** As the outcome has only two values, for the diabetic patient, it is "range 1." For non-diabetic, it is "range 2" while preparing the target, the "class of highest interest" for us is "range 2" as the precision and recall values depend on the positive result.
- **Selecting inputs:** Here, the user can select the data attributes to make the prediction. It also suggests to the user which attributes might help them get the precise results and gives them an option to discard the ones that will not be helpful.
- **Model Types:** In this section, we list all the classification models we want to execute. It gives us an option to select the required ones; here, we opted for Decision Tree (CART), Random Forest, and Logistic Regression.

- **Results:** This is the last step where the models' execution occurs, and the results are displayed according to the models' performance. The comparison part provides an overview of where the models are compared, and the bar chart represents the performance. We also get to view the results of the individual models that help us understand them more correctly.

Another added benefit of using AutoModel is that it gives the result and helps us comprehend the results, where for specific models, it gets hard to understand the inner logic at times that is when we can view the whole process and understand the algorithm. Under the section "Model Simulator," the" Open Process" displays the whole process and helps us validate the models.



Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu...	Age	Outcome
Number	Number	Number	Number	Number	Number	Number	Number	Number
6	148	72	35	0	33.600	0.627	50	1
1	85	66	29	0	26.600	0.351	31	0
8	183	64	0	0	23.300	0.672	32	1
1	89	66	23	94	28.100	0.167	21	0
0	137	40	35	168	43.100	2.288	33	1
5	116	74	0	0	25.600	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.300	0.134	29	0
2	197	70	45	543	30.500	0.158	53	1
8	125	96	0	0	0	0.232	54	1

Fig. 5: User interface and process overview of RapidMiner (version 9.8)

V. Literature Review

Diabetes is considered one of the fatal diseases, and its rapid spread has been one of the major concerns in today's medical world. Diabetes is caused due to either of the two mechanisms: inadequate production of insulin or insufficient sensitivity of cells to insulin action. The early detection of diabetes mellitus can help an individual take appropriate steps and further prevent health issues. Timely treatment and proper care can be useful to stop the spread of the disease. With advanced technology and abundant medical information, it is possible to develop effective detection methods. We have used the four algorithms of Machine Learning, namely Decision Tree (CART), Artificial Neural Network, Logistic Regression, and Random Forest, to exploit and enhance the disease's early detection process.
The working of the algorithms is described as follows:

Artificial Neural Network

Artificial Neural Networks is a set of models simulated by the biological neural network and are used for estimation or approximation of functions that can depend on many unknown inputs. Artificial neural networks are represented as systems of interconnected "neurons" that exchange messages with each other. The connections have multiple weights that tune themselves based on experience, making neural networks adaptive to inputs and capable of learning. ANN typically has three defining parameters:

- Pattern of Interconnection: Between neuron's various layers
- Learning Process: Updates the weights of the interconnection
- Activation Function: Converts neuron's weighted input to its output activation by determining whether each neuron's input is relevant for the model's prediction.

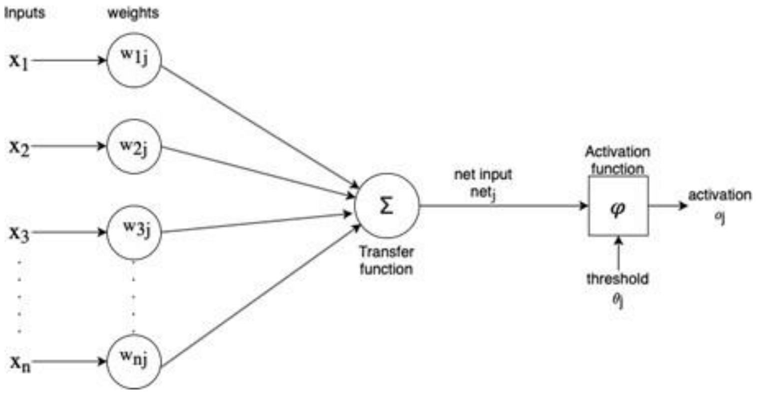


Fig. 6: Working of Artificial Neural Network (ANN)

Mathematical Definition

A neural network function f(x) is defined as a composition of another function gi(x), which can further complement other functions. The mentioned can easily be represented as a network structure, which depicts dependencies.

$$f(x) = K\left(\sum_{x=1} w_i g_i(x)\right)$$

where K (or the activation function) is some
predefined function, such as the hyperbolic tangent.

Rprop MLP

In our research, out of the multiple available options to implement Neural Networks, we used the Rprop MLP approach.

Resilient back-propagation (Rprop)

Rprop stands for Resilient back-propagation and is an algorithm that can train a neural network in a very similar way to the regular back-propagation. It differs from the regular back-propagation in two main ways:

- Training data using Rprop is often faster than training the same data using back-propagation.
- Rprop does require any free parameter values to be specified, as opposed to the back-propagation, which requires values for the learning rate.

Multilayer Perceptron (MLP)

Multilayer Perceptron or MLP is a deep artificial neural network that is composed of multiple perceptrons. It receives input signals from the input layer, and the output layer makes decisions or predictions on the input. Between the output and the input layer are an arbitrary number of hidden layers that are the real computational engine of the MLP. Each layer is capable of approximating any continuous function.

Multilayer perceptrons are mainly used for supervised learning problems. A set of input-output pairs are trained and then learn to model the correlations between inputs and outputs. The training aims to minimise errors by adjusting parameters/weights and biases using back-propagation.

Artificial Intelligence, along with its various tools and techniques, has helped detect diabetes early by extracting information from enormous amounts of data. The factors responsible for an individual to be diabetic, which have been well researched across people of various ages and lifestyles, are used for the prediction. Artificial Neural Network is a preferred technique for early diagnosis as classification and prediction are essential applications. It is necessary to do so based on risk factors in a disease like diabetes. The predictive capability within a fully trained dataset and on unseen data is analyzed.

CART

Decision Trees are vital for predictive modelling machine learning. Classification and Regression Trees also referred to as CART, is one such decision tree algorithm that can be exploited for both classification and Regression of predictive modelling problems.

CART Algorithm:

- It starts at the basis node using all the training instances
- An attribute is chosen based on the splitting criteria (Gini Index)

- Instances are then partitioned recursively consistent with the chosen attribute

CART utilises Gini Impurity, a measure of how randomly chosen elements from the given data set would be mislabeled if the label
distribution had randomly labelled it.

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

where n represents the total number of classes within the node and p is that the node's class distribution. After measuring impurity within the given data set, CART then generates a binary tree that splits the information in the decision tree.

CART is a resilient data mining tool that is mechanised to look for the crucial relationships and the behaviour of the data and automatically identifies the hidden patterns in the highly complex data. It is a reliable method. It does not require any advance assumptions about the elemental distribution of attribute values in the given data. Unlike other methods, CART continues to recursively split the data even after the best split is found until no more splitting is possible. Moreover, CART does not stop in between while the tree-growing process takes place. It drills down to the maximum levels by pruning away the maximal tree branches after it is generated not to miss any useful information.

Logistic regression

Logistic Regression is a classification algorithm often used to predict an outcome variable that is binary or multi-class dependent variables. It is used to model the probability of a particular class or event existing when a dependent variable is categorical such as pass/fail, win/lose, alive/dead, or healthy/sick.

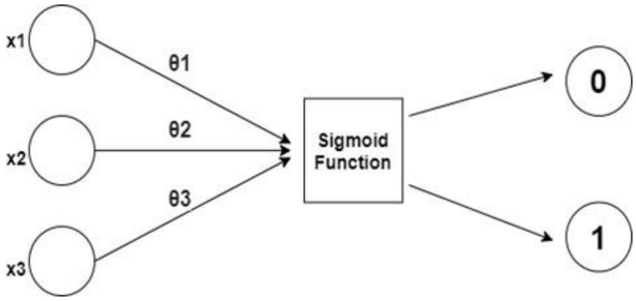


Fig. 7: Working of Logistic Regression

Logistic Regression uses a logistic function (also known as a sigmoid function) at the core of the method, which justifies the name' Logistic Regression.' In linear Regression, the output value being modelled is a binary value (0 or 1) rather than a numeric value. It builds the model to predict the peculiar occurrence instead of point estimate event

Mathematical Definition

The logistic Regression is represented using a logistic or sigmoid function which is a common S-shaped curve

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

with equation
Where e is the base of the natural logarithms, and x
is the actual numerical value detected.

Logistic Regression is a classification algorithm and also predicts a categorical or qualitative output variable. While predicting Diabetes Mellitus, certain various features are considered glucose, blood pressure, Insulin, BMI, and age. LR can be used to predict discrete functions. It is fast at classifying unknown records too. Based on the classification, the LR model functions all discrete features(x) and gives single-output(y) for the prediction – whether the patients have diabetes or not, Logistic Regression being more comfortable to implement and interpret, it is less inclined to overfitting.

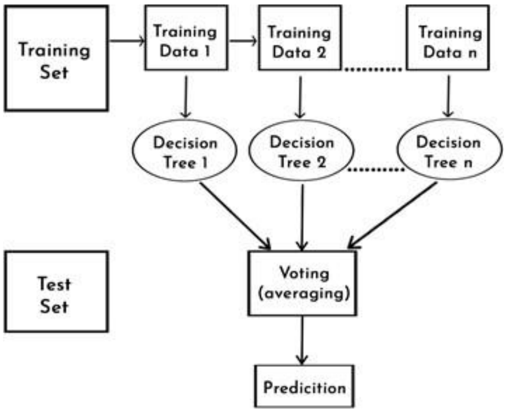
Random Forest

Random Forest is a commonly used Machine Learning algorithm that is mainly used to solve classification problems. This method is purely based on ensemble learning. The various algorithms are used to acquire more nuanced performance than those acquired from any other integral learning algorithms alone. Random Forest, works on generating various uncorrelated decision trees that together as a whole form a forest. Every tree in the random forest gives a class prediction, and the one with maximum votes is filtered out of the forest.

Understanding the working of the Random Forest algorithm:

- The first step involves selecting random samples from the particular dataset to be used.
- The algorithm will generate a decision tree for every sample that it selects and obtain the prediction result from each decision tree.
- This step involves voting for each predicted result.
- The most voted prediction result is selected as the final result.

Fig. 8: Working of Random Forest



The varied nature of this algorithm is the reason why it provides us with the most accurate results. The unrelated models tend to produce ensemble predictions that outperform the predictions made by the single model. It is known to eliminate overfitting by combining the multiple decision trees' results and coming up with the most precise result.

Random forest is one of those machine learning algorithms which is more adaptable towards the ensemble approach. It is a convenient and manageable technique that produces excellent outcomes, mostly without setting any super parameters. Some additional features that make it more functioning are its ability to estimate missing data. Weighted Random Forest (WRF) is one of its methods that balances error in imbalanced data and can handle missing values. Also, the importance of variables used in the classification is estimated by this method. Random Forest, being an ensemble learner, is highly capable of handling large datasets.

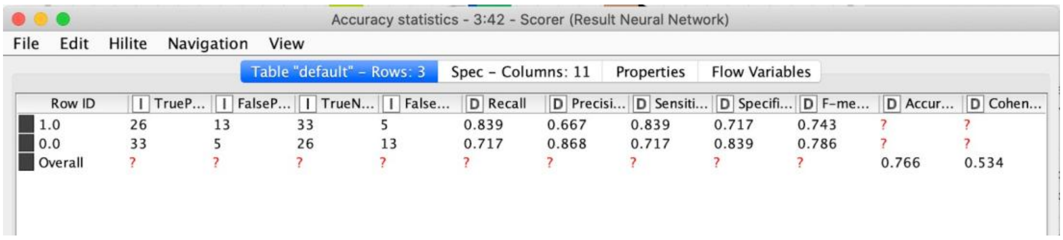
VI. Performance Evaluation

Performance of all the four algorithms with result screenshots of both software.

KNIME

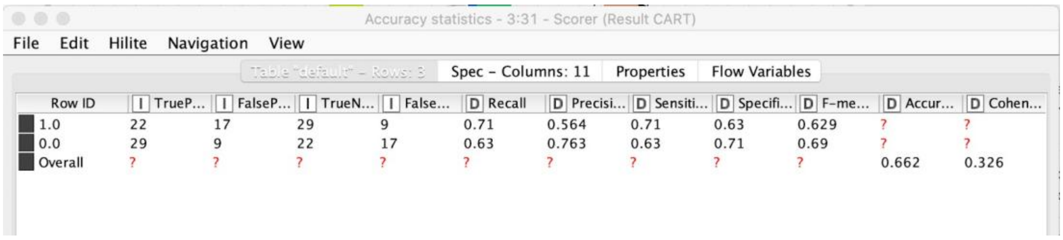
All the algorithms on KNIME were tested individually, as mentioned earlier. The dataset used to test these methods is Pima Indian Diabetes Dataset. When tested, the accuracies of the four algorithms that we got on KNIME in ascending order are as follows: Decision Tree (CART): 66.2%, Random Forest: 75.3%, Neural Network: 76.6%, and Logistic Regression: 79.2%. As the order suggests, Logistic Regression outperformed all the other three methods with the highest accuracy rate. In contrast, Decision Tree (CART) had the lowest accuracy among all the methods.

Fig. 9: Accuracy result on Knime using scorer for Artificial Neural Network (rProp MLP) Predictor on Pima Indian Dataset


Accuracy statistics - 3:42 - Scorer (Result Neural Network)
File
Edit
Hilite
Navigation
View
Table "default" – Rows: 3
Spec – Columns: 11
Properties
Flow Variables

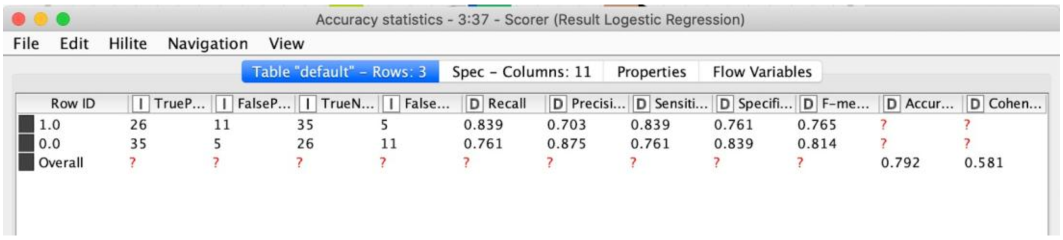
Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensiti...	D Specifi...	D F-me...	D Accur...	D Cohen...
1.0	26	13	33	5	0.839	0.667	0.839	0.717	0.743	?	?
0.0	33	5	26	13	0.717	0.868	0.717	0.839	0.786	?	?
Overall	?	?	?	?	?	?	?	?	?	0.766	0.534

Fig. 10: Accuracy result on Knime using scorer for Decision Tree Predictor on Pima Indian Dataset


Accuracy statistics - 3:31 - Scorer (Result CART)
File
Edit
Hilite
Navigation
View
Table "default" – Rows: 3
Spec – Columns: 11
Properties
Flow Variables

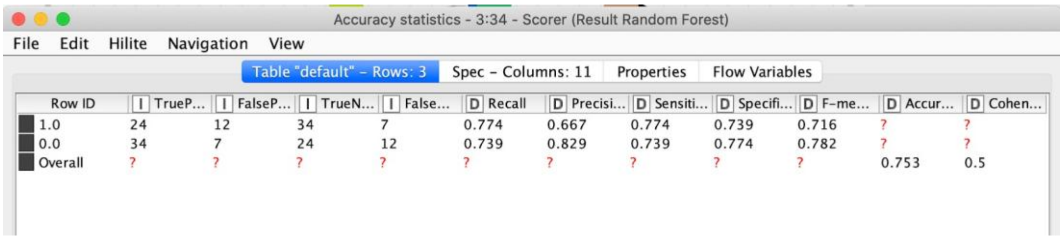
Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensiti...	D Specifi...	D F-me...	D Accur...	D Cohen...
1.0	22	17	29	9	0.71	0.564	0.71	0.63	0.629	?	?
0.0	29	9	22	17	0.63	0.763	0.63	0.71	0.69	?	?
Overall	?	?	?	?	?	?	?	?	?	0.662	0.326

Fig. 11: Accuracy result on Knime using scorer for Logistic Regression Predictor on Pima Indian Dataset


Accuracy statistics - 3:37 - Scorer (Result Logestic Regression)
File
Edit
Hilite
Navigation
View
Table "default" – Rows: 3
Spec – Columns: 11
Properties
Flow Variables

Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensiti...	D Specifi...	D F-me...	D Accur...	D Cohen...
1.0	26	11	35	5	0.839	0.703	0.839	0.761	0.765	?	?
0.0	35	5	26	11	0.761	0.875	0.761	0.839	0.814	?	?
Overall	?	?	?	?	?	?	?	?	?	0.792	0.581

Fig. 12: Accuracy result on Knime using scorer for Random Forest Predictor on Pima Indian Dataset

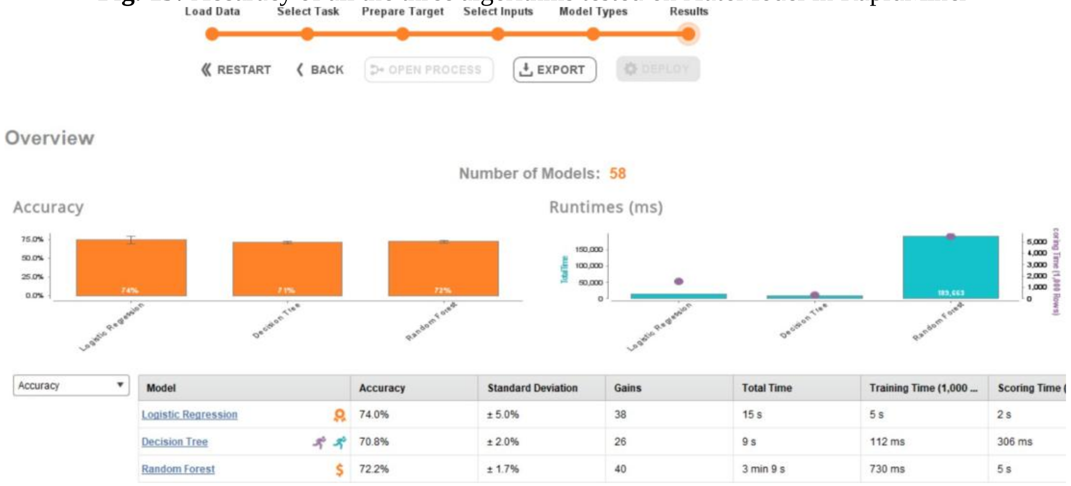

Accuracy statistics - 3:34 - Scorer (Result Random Forest)
File
Edit
Hilite
Navigation
View
Table "default" – Rows: 3
Spec – Columns: 11
Properties
Flow Variables

Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensiti...	D Specifi...	D F-me...	D Accur...	D Cohen...
1.0	24	12	34	7	0.774	0.667	0.774	0.739	0.716	?	?
0.0	34	7	24	12	0.739	0.829	0.739	0.774	0.782	?	?
Overall	?	?	?	?	?	?	?	?	?	0.753	0.5

RapidMiner

To test the algorithms on RapidMiner, we utilised the AutoModel that builds the model and compares the result of all the classifiers. The results were parallel to the ones that we obtained from KNIME. The algorithms that we tested are Decision Tree (CART), Logistic Regression, and Random Forest. These were tested on Pima Indian Diabetes Dataset. All the classifiers performed differently and gave different results. The classifiers' accuracy in ascending order is as follows: Decision Tree (CART): 70.8%, Random Forest: 72.2%, and Logistic Regression: 74%, where it showed that Logistic Regression had the best performance among the two. Simultaneously, the Decision Tree with the lowest accuracy had the Fastest Scoring Time that accounted for 112 milliseconds and the Fastest Total Time of 306 milliseconds. Random had the best gain of 40 compared to all the other algorithms.

Fig. 13: Accuracy of all the three algorithms tested on AutoModel in RapidMiner



Number of Models: 58

Model	Accuracy	Standard Deviation	Gains	Total Time	Training Time (1,000 ...	Scoring Time (
Logistic Regression	74.0%	± 5.0%	38	15 s	5 s	2 s
Decision Tree	70.8%	± 2.0%	26	9 s	112 ms	306 ms
Random Forest	72.2%	± 1.7%	40	3 min 9 s	730 ms	5 s

VII. Result

Running the Pima Indian Dataset on KNIME software and the RapidMiner software, the following results were obtained.

TABLE 2: Result of all the methods on data mining platform

Methods	KNIME (accuracy %)	RapidMiner (accuracy %)
Artificial Neural Network	76.5	-
Decision Tree	66.2	70.8
Logistic Regression	79.2	74.0
Random Forest	75.3	72.2

Running the Pima Indian Dataset on KNIME software resulted in the Logistic Regression Machine Learning Algorithm outperforming the other three algorithms (namely, Artificial Neural Network, Decision Tree and, Random Forest) with an accuracy of 79.2%. The Artificial Neural Network (Rprop MLP) Machine Learning Algorithm followed with an accuracy of 76.5%. Then is the Random Forest Machine Learning Algorithm followed by the Decision Tree method with an accuracy of 75.3% and 66.2%, respectively. The result is backed up by the RapidMiner software (with slightly different accuracy).

VIII. Conclusion

Inferring from the experiments we conducted, the Logistic Regression Machine Learning Algorithm yields the most accurate result in predicting true positive for diabetes. It is followed by Artificial Neural Network, Random Forest, and Decision Tree, respectively.

References

[1]. K. Sumangali, B. S. R. Geetika and H. Ambarkar, "A classifier based approach for early detection of diabetes mellitus," 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2016, pp. 389-392, doi: 10.1109/ICCICCT.2016.7987979.

[2]. Zou Q, Qu K, Luo Y, Yin D, Ju Y and Tang H (2018) Predicting Diabetes Mellitus With Machine Learning Techniques. Front. Genet. 9:515. doi: 10.3389/fgene.2018.00515

[3]. S. Benbelkacem and B. Atmani, "Random Forests for Diabetes Diagnosis," 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2019, pp. 1-4, doi: 10.1109/ICCISci.2019.8716405.

[4]. P. S. Kumar and S. Pranavi, "Performance analysis of machine learning algorithms on diabetes dataset using big data analytics," 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), Dubai, 2017, pp. 508-513, doi: 10.1109/ICTUS.2017.8286062.

[5]. S. Y. Rubaiat, M. M. Rahman and M. K. Hasan, "Important Feature Selection & Accuracy Comparisons of Different Machine Learning Models for Early Diabetes Detection," 2018 International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 2018, pp. 1-6, doi: 10.1109/CIET.2018.8660831.

[6]. Nongyao Nai-arun, Rungruttikarn Moungmai,Comparison of Classifiers for the Risk of Diabetes Prediction,Procedia Computer Science,Volume 69,2015,Pages 132-142, ISSN 1877-0509.

[7]. Using Neural Networks To Predict the Onset of Diabetes Mellitus, Murali S. ShankerJournal of Chemical Information and Computer Sciences 1996 36 (1), 35-41, DOI: 10.1021/ci950063e

[8]. Han Wu, Shengqi Yang, Zhangqin Huang, Jian He, Xiaoyi Wang, Type 2 diabetes mellitus prediction model based on data mining, Informatics in Medicine Unlocked, Volume 10, 2018, Pages 100-107, ISSN 2352-9148.

[9]. Srivastava, Suyash & Sharma, Lokesh & Sharma, Vijeta & Kumar, Ajai & Darbari, Hemant (2019). Prediction of Diabetes Using Artificial Neural Network Approach: ICoEVCI 2018, India. 10.1007/978-981-13-1642-5_59.

[10]. Lavery, Stephen and Jeremy Debattista. "Leveraging Pharmacy Medical Records To Predict Diabetes Using A Random Forest & Artificial Neural Network." AICS (2018).