“Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare” and Improvements of ML models.

Dixit Nandaniya (110122302)
Master of Applied Computing
University of Windsor
Windsor, Canada
nandanid@uwindsor.ca

Shashank Kannan (110122650)
Master of Applied Computing
University of Windsor
Windsor, Canada
kannan21@uwindsor.ca

Advisor: Dr. Adel Abusitta

ABSTRACT

This review paper critically analyses the study "Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare" written by Ayman Mir and Sudhir N. Dhage, which was published in 2017 and summarizes its advantages and disadvantages. In addition to implementing the algorithms from the research article, the paper evaluates several machine-learning techniques and provides a fresh strategy to address existing issues. By offering a comprehensive assessment, the objective is to aid in improving understanding of diabetes prediction and its advancements in the field of healthcare.

Keywords: Diabetes, Machine Learning, Strengths, Weaknesses

1. INTRODUCTION

This review paper provides a comprehensive analysis of the study "Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare". The essay's goal is to critically evaluate and list the benefits and drawbacks of the original work. The essay also implements the algorithms from the research article, compares several machine learning techniques, and recommends a novel strategy to address current issues. The goal of this study is to provide a complete survey that will help readers have a better grasp of diabetes prediction and its advances in the healthcare industry.

1.1 BACKGROUND AND SIGNIFICANCE

Millions of individuals throughout the world struggle with diabetes, a serious health issue. It is a chronic condition brought on by the body's inability to make enough insulin or use it appropriately, which raises blood sugar levels. Diabetes can have harmful effects on the body's organs, such as renal failure, blindness, and heart disease. Machine learning has the potential to help address the problem of diabetes by accurately predicting the disease and providing automated diagnosis under the validation of a professional doctor. Machine learning algorithms can be used to discover interesting patterns in disease data and improve diagnosis and prognosis.

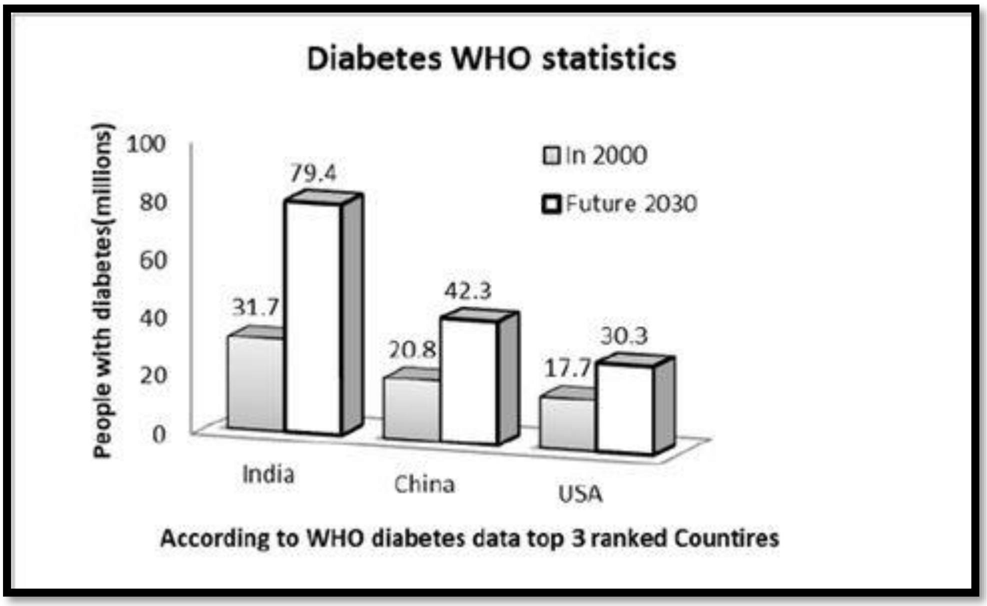

Figure 1

2. OVERVIEW OF "DIABETES DISEASE PREDICTION USING MACHINE LEARNING ON BIG DATA OF HEALTHCARE"

2.1 INTRODUCTION

In order to forecast diabetes, the study examines the use of machine learning algorithms in healthcare. The Naive Bayes, Support Vector Machine, Random Forest, and Simple CART algorithms were used by the authors to create a classifier model for predicting diabetes using the WEKA tool. The objective was to suggest the optimal algorithm for predicting diabetes based on effective performance outcomes. According to the trial findings, the Support Vector Machine was the most accurate at predicting the disease.

2.2 ADVANTAGES

This paper's comprehensive comparison of various machine learning methods for diabetes prediction is one of its benefits. The fact that the authors conducted their tests using a well-known dataset and a widely used machine learning technology (WEKA) lends weight to their findings.

2.3 DISADVANTAGES

However, applying machine learning in the healthcare industry could have significant drawbacks and difficulties. For instance, the Caliber and volume of the data used to train the models may affect how accurate the forecasts are. When dealing with sensitive medical information, data privacy, and security may also present problems.

2.4 SOLUTIONS

One proposal for future work on this study that may be made is to investigate the use of additional machine learning algorithms or methodologies to see if they can enhance prediction accuracy. Another approach may be to look into potential solutions for data security and privacy issues when applying machine learning to healthcare.

2.5 CONCLUSION

Overall, this work compares various diabetes prediction algorithms and makes a significant addition to the field of machine learning in healthcare. However, there is still opportunity for development and additional study in this field.

3. IMPLEMENTATION AND PERFORMANCE EVALUATION

In our study, we used the Dataspell IDE to develop several machine learning models for predicting diabetes which are:

- Fully Connected Neural Network
- Long Short-Term Memory
- K-Nearest Neighbours
- Logistic Regression
- Support Vector Machine
- Random Forest Classification
- Naive Bayes Classification.
- Simple CART
- Support Vector Machine
- XGBoost
- CATboost

We evaluated the performance of these models in terms of their Accuracy, True Positive Rate, False Positive Rate, Precision, Recall, and F-Measure.

3.1 Results

Here is a table that presents the accuracy, training time, and testing time of the different machine-learning models used in our study:

Model	Accuracy	True Positive Rate (Recall)	False Positive Rate	Precision	F-Measure
CNN	72.08	0.5818	0.2020	0.6154	0.5981
LSTM	74.03	0.7091	0.2424	0.6190	0.6610
KNN	70.78	0.5091	0.1818	0.6087	0.5545
Logistic Regression	75.32	0.6727	0.2020	0.6491	0.6607
SVM	75.97	0.6545	0.1818	0.6667	0.6606
Random Forest	72.08	0.6182	0.2222	0.6071	0.6126

Naive Bayes	76.62	0.7091	0.2020	0.6610	0.6842
Simple CART	74.68	0.7273	0.2424	0.6250	0.6723
XGBoost	68.83	0.6545	0.2929	0.5538	0.6000
CatBoost	75.32	0.6727	0.2020	0.6491	0.6607

Table 1

The accuracy ranges from a lowest of 68.83% (XGBoost) to a highest of 76.62% (Naive Bayes) among the models tested.

True Positive Rate (Recall): Simple CART achieved the highest recall rate of 72.73%, while K-Nearest Neighbours had the lowest at 50.91%.

False Positive Rate: Support Vector Machine showcased the lowest false positive rate at 18.18%, while XGBoost had the highest at 29.29%.

Precision: Support Vector Machine demonstrated the highest precision at 66.67%, while XGBoost had the lowest at 55.38%.

F-Measure: Naive Bayes attained the highest F-measure at 68.42%, and K-Nearest Neighbours achieved the lowest at 55.45%.

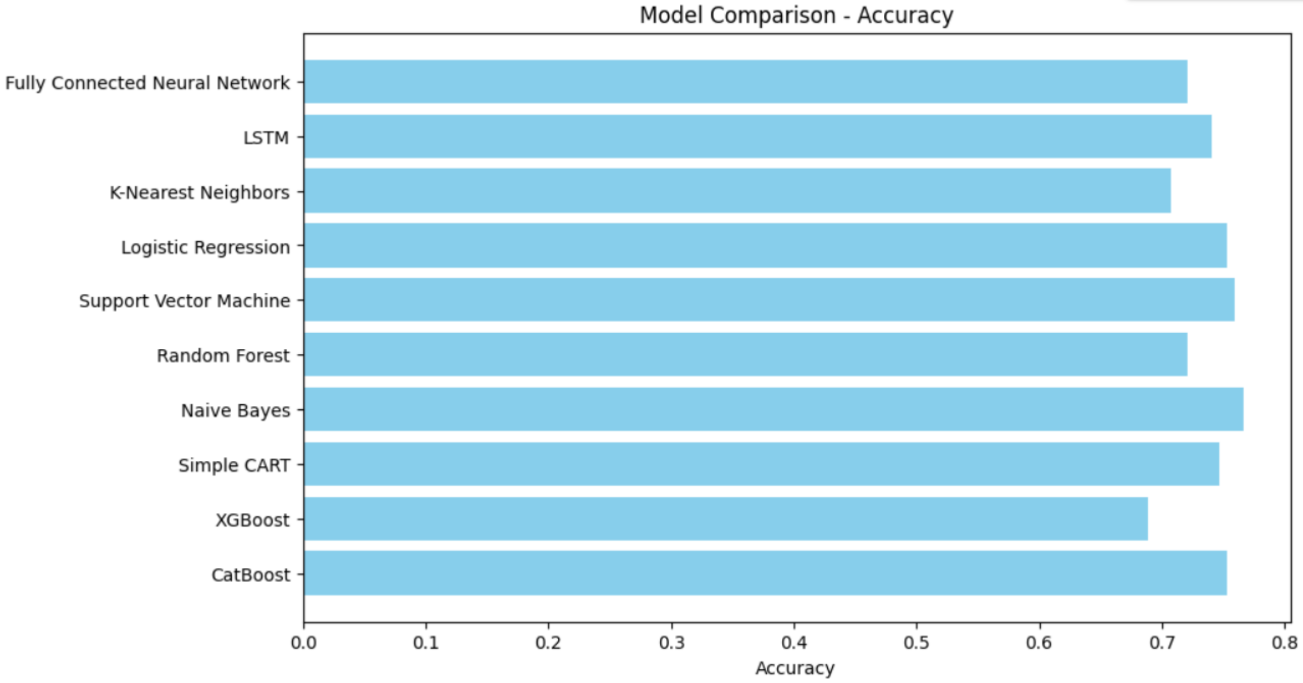

Figure 2

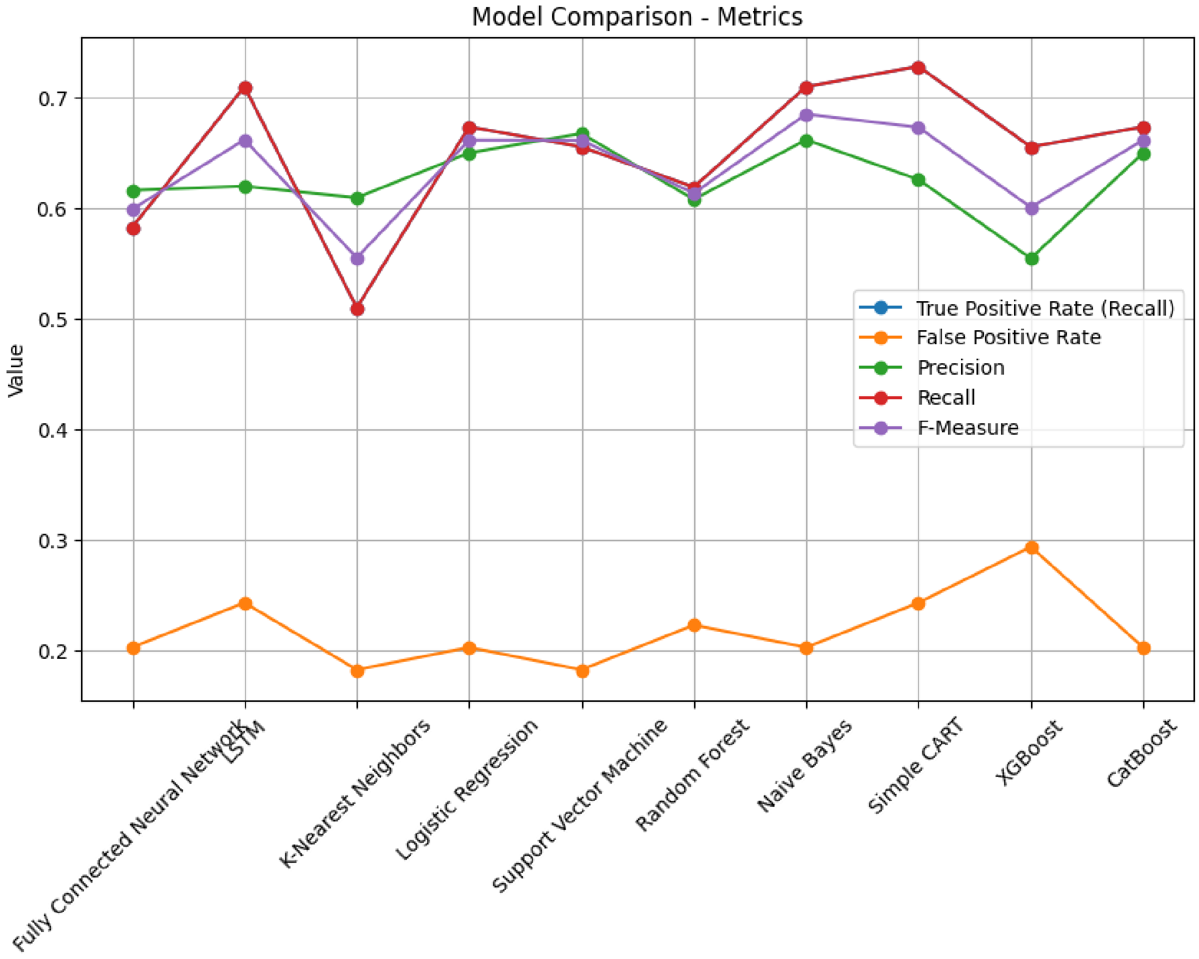

Figure 3

3.2 Model Descriptions

In our study, we compared the performance of several machine learning models for predicting diabetes using big data in healthcare.

3.2.1. Fully Connected Neural Network

A Fully Connected Neural Network, also known as a Multi-Layer Perceptron (MLP), is a type of artificial neural network. It consists of layers of interconnected nodes (neurons) that process input data through weighted connections. Each node applies an activation function to the weighted sum of its inputs, transforming the data and learning complex patterns in the process. In this scenario, the model processes the scaled diabetes dataset to learn relationships between input features and the diabetes outcome, aiming to make accurate predictions.

- Formula: The output of each neuron in a fully connected layer is calculated by applying an activation function to the weighted sum of its inputs.

- Activation Function Formula:
- Activation= (Weighted Sum)
- Activation= σ (Weighted Sum)
- Loss Function: Binary Cross-Entropy
- Optimizer: Stochastic Gradient Descent (SGD)
- The neural network iteratively adjusts weights to minimize the loss between predicted and actual outcomes.

3.2.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory is a type of recurrent neural network (RNN) designed for sequence data, like time series or text. Unlike traditional feedforward neural networks, LSTM includes memory cells and gates to capture long-term dependencies and avoid the vanishing gradient problem. In this scenario, LSTM processes the diabetes data as sequential data, learning temporal patterns that might contribute to diabetes outcomes.

- Formula: LSTM units include gates (input, forget, output) to control information flow and memory storage.
- LSTM Gate Formulas:
- Input Gate = (Weighted Sum of Inputs)
- Input Gate = σ (Weighted Sum of Inputs)
- Forget Gate = (Weighted Sum of Inputs)
- Forget Gate =σ (Weighted Sum of Inputs)
- Output Gate = (Weighted Sum of Inputs)
- Output Gate= σ (Weighted Sum of Inputs)
- Loss Function: Binary Cross-Entropy
- Optimizer: Adam Optimizer
- LSTM units capture long-term dependencies by adjusting gate weights and cell states.

3.2.3. K-Nearest Neighbours (KNN)

K-Nearest Neighbours is a simple classification algorithm that assigns a data point to the class most common among its k nearest neighbors. It measures the proximity between data points using distance metrics like Euclidean distance. In this case, the KNN model classifies diabetes outcomes based on the characteristics of similar data points in the training set.

- Formula: KNN classifies a data point based on the majority class of its k-nearest neighbours in the feature space.
- Distance Metric Formula (Euclidear $\text{Distance} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
- The model calculates distances to find the nearest neighbours and assigns the class based on the majority vote.

3.2.4. Logistic Regression

Logistic Regression is a statistical model used for binary classification. Despite its name, it's used for classification tasks. The model estimates the probability of a binary outcome using a logistic function, which maps input features to the probability of belonging to a certain class. In this scenario, logistic regression learns the relationships between features and diabetes outcomes to predict whether an individual has diabetes or not.

- Formula: Logistic Regression estimates the probability of the positive class using the logistic function.
- Logistic Function Formu$P(Y=1|X)=\frac{1}{1+e^{-z}}$
- Loss Function: Binary Cross-Entropy
- The model adjusts weights to minimize the difference between predicted probabilities.

3.2.5. Support Vector Machine (SVM)

A Support Vector Machine is a powerful classification algorithm that aims to find the optimal hyperplane that best separates data points of different classes. It does so by maximizing the margin between classes while minimizing misclassifications. In this case, SVM tries to find the hyperplane that best separates individuals with diabetes from those without, based on the input features.

- Formula: SVM finds the optimal hyperplane by maximizing the margin between classes while minimizing errors.
- Hyperplane Equatior$w \cdot x + b = 0$
- Margin Calculation: 2/{w}
- The model determines the hyperplane that best separates classes by maximizing the margin.

3.2.6. Random Forest Classification

Random Forest is an ensemble learning technique that builds multiple decision trees and combines their outputs for better accuracy and robustness. Each tree is trained on a subset of data with bootstrapping, and the model aggregates the predictions of individual trees. In this scenario, the Random Forest model constructs multiple decision trees to collectively predict diabetes outcomes based on various feature combinations.

- Formula: Random Forest combines predictions from multiple decision trees.
- Decision Tree Formula: A sequence of binary decisions based on feature conditions.
- Aggregation Formula:
- Ensemble Prediction = Mode (Predictions from Individual Trees)
- Ensemble Prediction=Mode (Predictions from Individual Trees)
- Each tree provides a prediction, and the mode of all predictions forms the final ensemble prediction.

3.3.7. Naive Bayes Classification

Naive Bayes is a probabilistic classification method based on Bayes' theorem. It assumes that features are conditionally independent given the class, simplifying computations. Despite its "naive" assumption, Naive Bayes performs well in practice, especially with text and categorical data. In this scenario, Naive Bayes calculates the probability of an individual having diabetes based on the probabilities of feature occurrences in each class.

- Formula: Naive Bayes calculates the probability of a class given the features using Bayes' theorem.
- Bayes' Theorem Formul$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$
- Conditional Independence Assumptic$P(X|Y) = \prod_{i=1}^{n} P(x_i|Y)$
- The model calculates probabilities to determine the class that maximizes the posterior probability.

3.2.8. Simple CART (Decision Tree) Classifier

A Decision Tree is a flowchart-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. The Simple CART model creates a binary tree where decisions are made at each node based on specific feature conditions. In this case, the Decision Tree is used to classify diabetes outcomes by traversing the tree based on input features.

- Formula: Decision Tree creates a binary tree structure to classify data.
- Decision Rule Formula: x ≤t→Left Child, Else→Right Child
- Gini Impurity Formula:
- Gini (p) = $1 - \sum_{i=1}^{c} p_i^2$ where pi is the probability of class i.
- The model splits data based on features using decision rules to minimize impurity.

3.2.9. XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced gradient boosting algorithm that combines multiple decision trees sequentially. It addresses shortcomings of traditional boosting, such as overfitting, by incorporating regularization and optimizing the model's hyperparameters. In this scenario, XGBoost constructs an ensemble of decision trees to predict diabetes outcomes while minimizing bias and variance.

- Formula: XGBoost combines predictions from boosting iterations.
- Objective Functior$\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{j=1}^{T} \Omega(f_j)$
- Regularization Term Formu$\Omega(f_j) = \gamma T + \frac{1}{2}\lambda \|w\|_2^2$
- The model iteratively fits decision trees and combines their predictions while considering regularization terms.

3.2.10. CatBoost

CatBoost stands for "Categorical Boosting" and is a gradient boosting algorithm specifically designed to handle categorical features effectively. It uses an ordered boosting technique and employs various strategies to process categorical data. In this case, CatBoost constructs an ensemble of decision trees to predict diabetes outcomes, taking into account the unique properties of categorical features in the dataset.

- Formula: CatBoost optimizes a loss function using ordered boosting.
- Loss Function: Customized based on application, often uses Logloss
- Gradient Calculation: Uses ordered boosting and gradient boosting techniques for optimization.
- The model iteratively fits decision trees while considering categorical features' unique properties.

Each of these models employs different techniques and strategies to learn patterns from the diabetes dataset and make accurate predictions about diabetes outcomes. The choice of model depends on the specific characteristics of the data and the desired trade-offs between accuracy, interpretability, and the ability to handle different types of features.

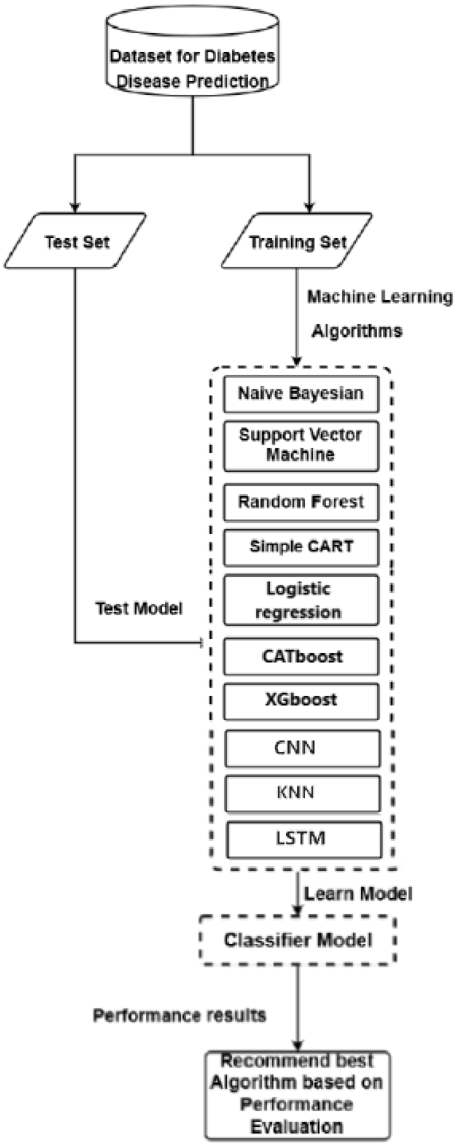

Figure 4

4. Conclusion

Overall, this paper offers useful information on how various machine learning models perform when used in healthcare big data to predict diabetes. It may be possible to find ways to increase the efficacy and accuracy of these models through further study.