# **A Comparison of Machine Learning Algorithms for Diabetes Prediction**

## **(i) Abstract**

Diabetes is a prevalent chronic disease that requires early detection for effective management. In this study, we explored the application of various machine learning (ML) algorithms to predict diabetes using the **Pima Indian Diabetes Dataset (PIDD)**. The dataset consists of 768 records with nine health-related attributes for female patients. Several machine learning models were implemented, including **Logistic Regression (LR)**, **Support Vector Machine (SVM)**, **Random Forest (RF)**, **K-Nearest Neighbors (KNN)**, **Naive Bayes (NB)**, **Decision Trees (DT)**, **AdaBoost (AB)**, and **Neural Networks (NN)**. Each model was evaluated using performance metrics such as accuracy, precision, recall, and F1-score.. The results indicate that the Support Vector Machine achieved the highest accuracy of 88.6%, significantly outperforming other models. This research highlights the potential of machine learning techniques in enhancing diabetes prediction and emphasizes the importance of specific health metrics in assessing diabetes risk.

## **(ii) Introduction**

### **Project Objectives**

The primary objective of this project is to evaluate the effectiveness of various machine learning algorithms in predicting diabetes using the Pima Indian Diabetes dataset. By analyzing the performance of these algorithms, we aim to identify the most accurate models and the key features that contribute to successful diabetes prediction. This study also seeks to provide insights into the applicability of machine learning in the healthcare domain, particularly in chronic disease management.

### **Problem Formulation**

Diabetes is a chronic condition that can lead to severe health complications, including cardiovascular diseases, kidney failure, and neuropathy, if not detected early. Traditional diagnostic methods may not always provide timely results, necessitating the exploration of advanced predictive techniques.

This study formulates the problem of diabetes prediction as a classification task, where the goal is to classify patients as diabetic or non-diabetic based on their health attributes. The challenge lies in selecting the appropriate machine learning models and features that can accurately predict the likelihood of diabetes. The research questions guiding this study include:

* Which machine learning algorithms provide the highest accuracy in predicting diabetes?
* What are the most significant features influencing diabetes prediction?

## **(iii) Methodology Used**

### **Dataset**

The Pima Indian Diabetes dataset (PIDD) was utilized for this study, sourced from the UCI Machine Learning Repository. The dataset contains 768 records of female patients, with nine attributes:

* Pregnancy: Number of pregnancies.
* Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test.
* Blood Pressure: Diastolic blood pressure (mm Hg).
* Skin Thickness: Triceps skin fold thickness (mm).
* Insulin: 2-Hour serum insulin (mu U/ml).
* BMI: Body mass index (weight in kg/(height in m)^2).
* Diabetes Pedigree Function: A function that scores the likelihood of diabetes based on family history.
* Age: Age (years).
* Outcome: Class variable (0 for non-diabetic, 1 for diabetic).

### **Data Preprocessing**

**Data preprocessing** is a crucial step to ensure that the data is in the correct format for training the models. The following steps were undertaken:

#### **Handling Missing Values**

The dataset contained several missing values, especially in features like **Glucose**, **Blood Pressure**, **Insulin**, and **BMI**, which were replaced with the mean of the respective attributes. This allowed us to retain the full dataset for modeling without dropping records.

#### **Normalization**

To ensure that all features contribute equally to the model and prevent bias due to feature magnitude, **MinMax Scaling** was applied to normalize the data between 0 and 1.

#### **Feature Selection**

Feature selection plays an important role in improving model performance by removing irrelevant or redundant features. I applied correlation-based **feature selection** to select the most important features. This reduced the dataset to the most significant predictors for diabetes. Based on the correlation values selected features are **'Pregnancies', 'Glucose', 'DiabetesPedigreeFunction', 'SkinThickness', 'Insulin', 'BMI', 'Age'** which results in higher accuracies in various models compared to other feature set

### **Machine Learning Models**

Implemented the following machine learning algorithms for diabetes prediction:

#### **1. Logistic Regression (LR)**

**Logistic Regression** is a statistical method for binary classification that estimates the probability of an outcome based on input variables. It uses the **sigmoid function** to predict the probability of a positive class (diabetes).

**Structure**: Logistic regression is a linear classifier that models the log-odds of the outcome as a linear combination of the input features.

* **Advantages**: Simple and interpretable.
* **Limitations**: Assumes linearity between features and the log-odds of the outcome.

#### **2. Support Vector Machine (SVM)**

**Support Vector Machine (SVM)** is a powerful classification algorithm that constructs an optimal hyperplane in a high-dimensional space to separate the classes.

**Structure**: SVM works by maximizing the margin between the support vectors and the decision boundary. The kernel trick is used to handle non-linear separations.

* **Advantages**: Effective in high-dimensional spaces.
* **Limitations**: Computationally expensive for large datasets.

#### **3. Naive Bayes (NB)**

**Naive Bayes** is a probabilistic classification model based on **Bayes' Theorem**, assuming that features are independent. Despite this assumption often being unrealistic, Naive Bayes can perform surprisingly well, especially on smaller datasets.

**Structure**: Based on **conditional probability** and assumes independence between features. It’s particularly useful for tasks with text classification but also performs well in medical prediction tasks.

* **Advantages**: Works well with small datasets and is computationally efficient.
* **Limitations**: Relies on the assumption of feature independence, which may not hold in practice.

#### **4. Random Forest (RF)**

**Random Forest** is an ensemble learning method that builds multiple decision trees and aggregates their predictions to improve accuracy and reduce overfitting. It is robust against noise and provides an inherent method for feature importance.

**Structure**: It constructs decision trees from bootstrapped samples of the dataset, and each tree is trained on a random subset of the features.

* **Advantages**: Reduces overfitting, handles large datasets well.
* **Limitations**: Less interpretable compared to single decision trees.

#### **5. K-Nearest Neighbors (KNN)**

**K-Nearest Neighbors (KNN)** is a simple, non-parametric method that classifies instances based on the majority class of their closest neighbors. The choice of k (the number of neighbors) is crucial to its performance.

**Structure**: The algorithm classifies based on the **majority vote** of the closest k neighbors. Distance metrics, such as **Euclidean distance**, are used to compute the proximity between data points.

* **Advantages**: Simple to implement and understand.
* **Limitations**: Computationally expensive for large datasets and sensitive to the choice of k.

#### **6. Decision Tree (DT)**

A **Decision Tree** splits the data recursively based on the most significant features to create a tree-like model for classification. Each internal node represents a decision based on a feature, and each leaf node represents a classification.

**Structure**: Decision Trees are built by recursively splitting the dataset at each node based on a feature that results in the most significant information gain.

* **Advantages**: Easy to interpret, works well with non-linear data.
* **Limitations**: Prone to overfitting if not controlled with pruning or setting a maximum depth.

#### **7. AdaBoost**

**AdaBoost** is an ensemble technique that combines multiple weak classifiers (often decision trees) into a strong classifier. It adjusts the weights of incorrectly classified instances and focuses more on difficult-to-classify samples in subsequent iterations.

**Structure**: AdaBoost boosts the performance of weak learners by iteratively improving their predictions and combining the results into a final strong prediction.

* **Advantages**: Improves accuracy by focusing on misclassified instances.
* **Limitations**: Sensitive to noise in the dataset.

**8. Neural Networks (NN)**

**Neural Networks** consist of layers of neurons that stimulate the human brain's structure. Each neuron takes inputs, applies weights, and passes the result through an activation function. In this experiment, we used a **feedforward neural network** with two hidden layers.

**Structure**: The neural network comprises an input layer, two hidden layers (with 12 and 8 neurons), and an output layer with a sigmoid activation for binary classification.

I have implemented this **neural network** with a simple architecture consisting of:

**Input Layer**: 12 neurons, taking 8 features as input.

**Hidden Layer**: 8 neurons, learning intermediate representations.

**Output Layer**: 1 neuron, producing a binary output for diabetes classification.

* **Advantages**: Captures non-linear relationships and patterns.
* **Limitations**: Requires large datasets and is computationally intensive.

### **Experimental Design**

The dataset was split into training and testing sets using a 80-20 ratio. Each model's performance was evaluated using accuracy, precision, recall, and F1-score metrics. The models were trained and tested using K-fold cross-validation (with K=10) to ensure robust evaluation and to mitigate overfitting. The following steps were taken:

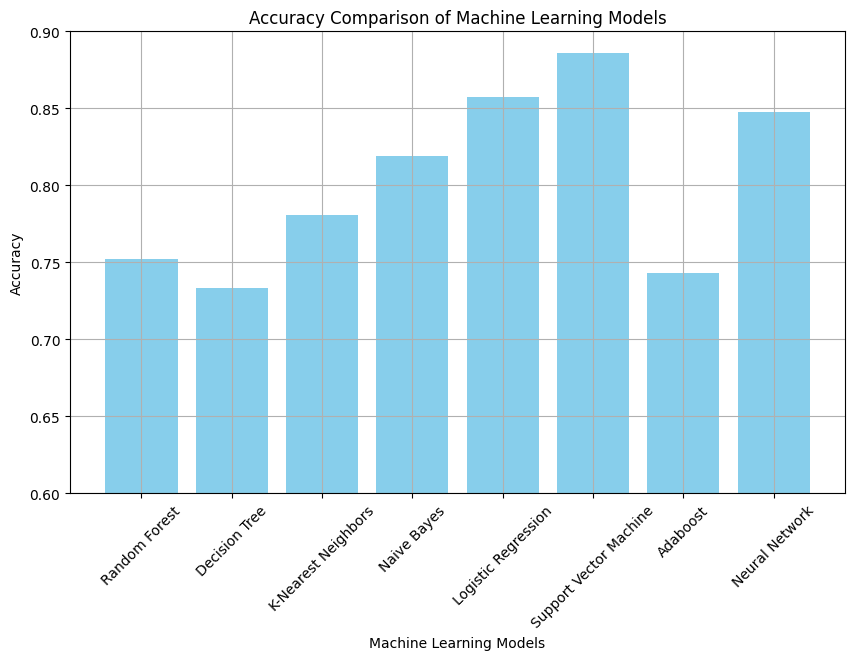
1. **Model Training:** Each algorithm was trained on the training dataset.
2. **Model Evaluation:** The trained models were evaluated on the test dataset, and performance metrics were calculated.
3. **Hyperparameter Tuning:** For the Neural Network, different configurations of hidden layers and epochs were tested to optimize performance.

### **(iv) Results**

#### **Model Performance**

The performance of the various machine learning models applied to the Pima Indian Diabetes (PID) dataset is summarized in Table and Graph. The accuracy of all classification methods used is above 70%. Notably, Logistic Regression (LR) and Support Vector Machine (SVM) demonstrated the best performance, achieving accuracies of 85.7% and 88.6% respectively when evaluated using the train/test splitting method. The Neural Network (NN) model, particularly with one hidden layer, achieved an accuracy of 85%, indicating a strong performance in predicting diabetes outcomes.

| **Model** | **Accuracy(%)** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 85.7 | 0.86 | 0..94 | 0.90 |
| Support Vector Machine | 88.6 | 0.88 | 0.97 | 0.92 |
| Naive Bayes | 81.9 | 0.86 | 0.88 | 0.87 |
| Random Forest | 75.23 | 0.79 | 0.86 | 0.83 |
| KNN | 78.09 | 0.83 | 0.86 | 0.84 |
| Decision Tree | 73.33 | 0.81 | 0.81 | 0.81 |
| AdaBoost | 74.28 | 0.83 | 0.79 | 0.81 |
| Neural Network  (1 Hidden layer ,100 Epochs) | 85 | 0.86 | 0.93 | 0.89 |



#### **Summary of Neural Network Performance:**

* **1 Hidden Layer - 200 Epochs: Accuracy - 0.7904762029647827**
* **1 Hidden Layer - 400 Epochs: Accuracy - 0.800000011920929**
* **1 Hidden Layer - 800 Epochs: Accuracy - 0.7333333492279053**
* **2 Hidden Layers - 200 Epochs: Accuracy - 0.761904776096344**
* **2 Hidden Layers - 400 Epochs: Accuracy - 0.7142857313156128**
* **2 Hidden Layers - 800 Epochs: Accuracy - 0.723809540271759**

#### **Performance on Chosen Dataset**

The models were evaluated on the PID dataset, which consists of 768 records with nine attributes. After preprocessing, including normalization and feature selection, the dataset was split into training and testing sets. The results indicated that all models performed satisfactorily, with accuracies above 70%. The confusion matrix for each classifier was also analyzed, providing insights into the true positive, false positive, true negative, and false negative rates.

The performance of the models varied, with the SVM showing the highest accuracy at 88.6%, followed by Logistic Regression and Neural Network. The results suggest that the chosen dataset is suitable for diabetes prediction, and the applied machine learning algorithms are effective in identifying patterns that correlate with diabetes outcomes. The detailed performance metrics for each model are illustrated in the results section, highlighting the strengths and weaknesses of each approach in the context of diabetes prediction.

### **Conclusion**

This study demonstrates the potential of machine learning algorithms in predicting diabetes using the Pima Indian Diabetes dataset. The findings indicate that while several models can achieve satisfactory accuracy, Neural Networks stand out as the most effective approach. Future work could explore the integration of additional datasets and advanced techniques, such as deep learning, to further improve prediction accuracy and generalizability.

### **References**

* **Codes:**

<https://github.com/Thanveerahamed-14/mlproject.git>

* **Dataset:** <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data>
* **Reference Paper:** <https://www.sciencedirect.com/science/article/pii/S2405959521000205>