**Early Detection of Diabetes Disease by Using Machine learning**

# Table Of Contents

|  |  |
| --- | --- |
| **SNO. TITLE** | |
| **1.** | **ABSTRACT** |
| **2.** | **INTRODUCTION** |
| **3.** | **EXISTING SYSTEM** |
| **4.** | **PROBLEM STATEMENT** |
| **5.** | **PROPOSED SYSTEM** |
| **6.** | **RESULTS AND DISCUSSION** |
| **7.** | **CONCLUSION** |

# Abstract

We use machine learning algorithms to identified Diabetes disease. The study made use of a dataset that included clinical speech recordings from people with and without Parkinson's disease. Handling missing data and scaling features were part of the preprocessing processes that made sure the model performed as best it could. We trained and assessed two different machine learning algorithms: Random Forest and Decision Tree. We used common measures such receiver operating characteristic (ROC) curves, confusion matrix analysis, and accuracy.

**Introduction**

Predicting diabetes at an early stage is crucial for preemptive healthcare management and lifestyle adjustments aimed at prevention or control. This chronic disease affects individuals across all age groups. Machine learning algorithms play a pivotal role in identifying potential diabetes risks through predictive modeling. By analyzing relevant health data, these algorithms facilitate early detection, enabling timely interventions and personalized strategies. Such proactive measures can significantly mitigate the onset or progression of diabetes, thereby improving overall health outcomes and quality of Life.

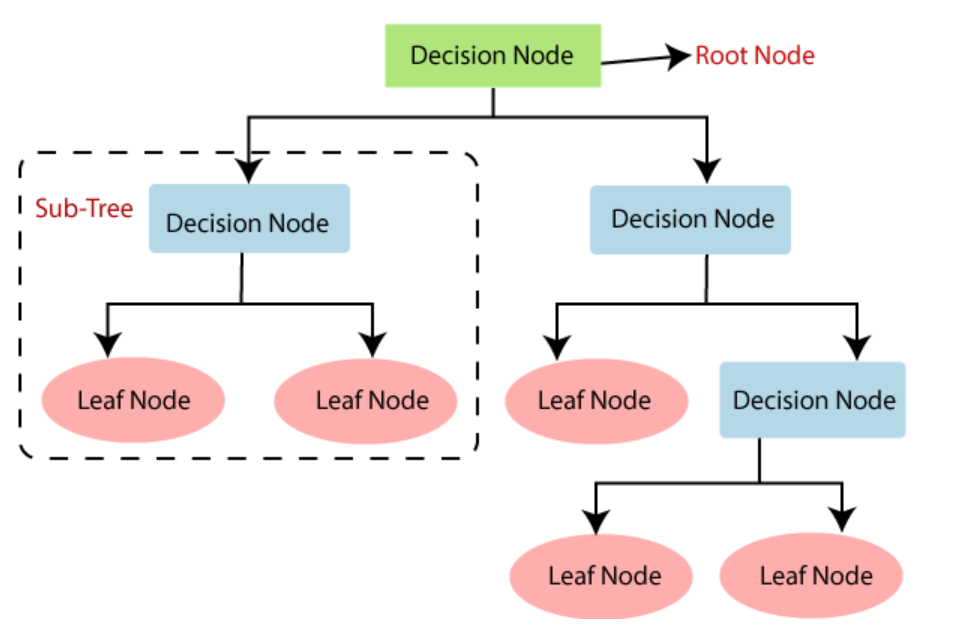
**EXISTING SYSYSTEM**

**Decision Trees:**

One kind of supervised machine learning technique used for both regression and classification tasks are the decision tree [11]. They are a popular option because of their versatility, interpretability, and simplicity in a wide range of applications [12]. Based on the value of a feature, a decision tree is a hierarchical structure that recursively divides the data into smaller groups. The leaf nodes of the tree indicate a class name or a continuous value, whereas the inside nodes of the tree represent a characteristic. Every internal node has a single incoming edge and several outgoing edges, each of which represents a potential feature value. For every data point in the associated subset, the leaf nodes hold the predicted continuous value or the expected class label. Tree induction is the method of creating a decision tree. Finding the optimal split at each internal node that minimizes a given criterion such as entropy or Gini impurity is the aim of tree induction. Gini impurity indicates the likelihood that a randomly selected element will be incorrectly categorized, whereas entropy measures the cleanliness of the data. After the optimal split has been identified, the data is recursively divided into smaller subsets. The technique is iterated until a stopping criterion, such as a minimum number of samples per leaf node or a maximum depth, is met.

Compared to other machine learning methods, decision trees provide several advantages. Because the decision criteria are unambiguous and can be represented using a tree structure, they are simple to comprehend and apply. They may be applied to problems involving both regression and classification, and they can handle both numerical and categorical data. By making the tree shallow and broad, decision trees can manage noisy data and are also resistant to missing data.

Decision trees, however, are susceptible to overfitting, particularly in cases when the tree is deep. When a model fits training data too closely, it is said to be overfitting and captures noise and outliers rather than the underlying pattern [13]. Decision trees are frequently trimmed, meaning that branches that do not considerably increase the accuracy of the tree are deleted, to reduce overfitting. Pruning enhances the tree's capacity for generalization while lowering its complexity.



**Random Forest:**

Random forests are a powerful ensemble learning algorithm that combines multiple decision trees to create a more robust and accurate model [14]. They are based on the idea of bagging, which involves building multiple models on different subsets of the data and averaging their predictions to improve the overall accuracy. Random forests extend bagging by introducing randomness in the feature selection process, which helps to reduce overfitting and improve generalization performance. Random forests work by building multiple decision trees on random subsets of the data and features. At each internal node of the tree, a random subset of features is selected, and the best split is chosen based on a criterion, such as entropy or Gini impurity. The size of the random subset of features is typically a hyperparameter that is set before training. Once the tree is built, the expected class label or the predicted continuous value is gotten by combining the forecasts of all the trees, usually using majority votes for classification tasks, and average for regression tasks.

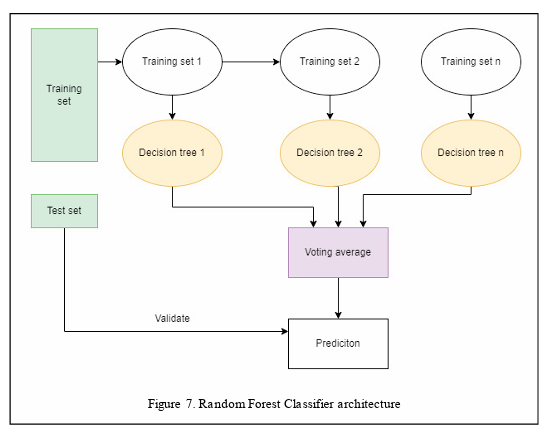
Random forests have several advantages over other machine learning algorithms. They are highly flexible and can handle both numerical and categorical data. They are also robust to outliers and missing data, as the trees are built on random subsets of the data [15]. Random forests are also less prone to overfitting than individual decision trees, as the randomness in the feature selection process helps to decorrelate the trees and reduce the variance.

However, random forests can be computationally expensive for large datasets and high-dimensional feature spaces. The time complexity of building a random forest is quadratic in the number of features, as each tree is built on a random subset of features. The space complexity is also high, as multiple trees need to be stored in memory. To mitigate these issues, several techniques have been proposed, such as random subspace sampling, which selects a random subset of features for each tree, and feature subsampling, which selects a random subset of the data for each tree.

Random forests have several applications in various fields, such as finance, healthcare, and marketing. In finance, random forests can be used to predict stock prices, detect fraud, and assess credit risk. In healthcare, random forests can be used to diagnose diseases, predict patient outcomes, and optimize treatment plans [16]. In marketing, random forests can be used to segment customers, predict customer behaviour, and optimize marketing campaigns.

Random forests can also be used for feature selection, as the importance of each feature can be measured by the number of times it is selected for splitting. This is useful for identifying the most relevant features in a high-dimensional feature space or for reducing the dimensionality of the data. Random forests can also be used for anomaly detection, as the out-of-bag (OOB) error can be used as a measure of the goodness of fit of the model.

Random forests have several hyper-parameters that can be tuned to optimize the performance of the model. The number of trees, the depth of the trees, the size of the random subset of features, and the criterion for splitting are some of the hyper-parameters that can be tuned [17]. Hyperparameter tuning can be done using cross-validation or grid search, where the model is trained and evaluated on different combinations of hyper-parameters to find the best set of hyper-parameters.

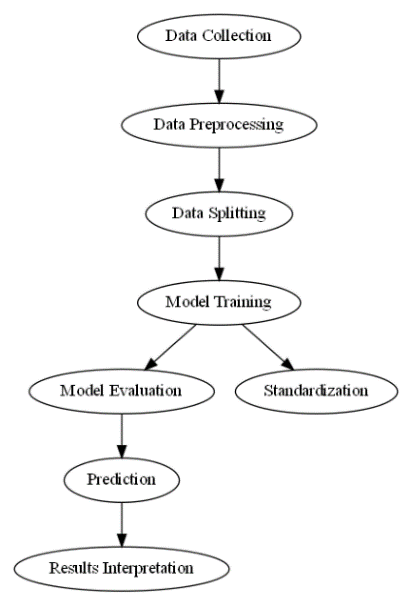


**Problem Statement**

Despite advancements in medical science, early detection of diabetes remains challenging, leading to delayed diagnosis and increased healthcare costs. Current diagnostic methods often rely on symptoms which manifest late in the disease progression. Therefore, there is a critical need to develop a reliable machine learning model that can accurately predict the likelihood of diabetes in individuals based on early clinical data such as demographics, lifestyle factors, and basic health metrics. Such a model would not only facilitate timely interventions and personalized healthcare but also potentially mitigate the long-term health complications associated with diabetes."

**Proposed System**

1. Data Acquisition: Data on Diabetes disease is a useful tool for developing prediction models that can help identify and diagnose the degenerative condition early on. The quality and dependability of the data are guaranteed when it is obtained from a reliable source, such a medical research repository. Medical research repositories are excellent sources of therapeutically relevant data because they frequently have stringent policies and procedures for gathering data.
2. Data preprocessing: Because it assures the consistency and quality of the data, an important step in the machine learning process is data preparation. This step involves cleaning the acquired dataset by removing outliers, fixing missing values, and fixing inconsistencies. It is possible to impute missing data using sophisticated methods like regression imputation or by employing tactics like mean or median imputation. IQR and Z-score techniques are two statistical approaches that may be used to discover and remove outliers. Taking care of these flaws in the data enhances model performance and lowers bias. Furthermore, attributes that aren't important to the prediction job, such the 'name' column, should be eliminated because they might add noise into the model training process.
3. Feature Selection: The process of locating and choosing pertinent characteristics that make a substantial contribution to the prediction job is known as feature selection. Both domain expertise and exploratory data analysis (EDA) are involved in this procedure. EDA looks for patterns, correlations, and linkages in the data, and domain knowledge guarantees that the characteristics selected have significance and are pertinent to medicine. There are three types of feature selection methods: filter, wrapper, and embedding. Each type of approach has pros and cons. In addition to improving model performance, using carefully chosen features lowers the chance of overfitting.
4. Data Splitting: To guarantee an unbiased assessment of the machine learning models, the dataset is split into training and testing sets. This technique is known as data splitting. A well-liked technique for randomly dividing the dataset and guaranteeing that each subset is representative of the original dataset is to use the train\_test\_split function from sklearn.model\_selection. Usually, training uses 70–80% of the data, with the remaining 30–40% going for testing. An equitable assessment of the model's performance and generalizability is made possible by proper data partitioning.
5. Model Training: To create predictive models, model training entails using pre-processed data and certain attributes. The Random Forest Classifier, and Decision Tree Classifier are the two machine learning models that are trained on the
6. Visualization: The performance and underlying mechanics of machine learning models may be effectively interpreted and comprehended via the use of visualization. ROC curves, feature significance plots, and confusion matrices are popular visualizations that offer insightful information about the advantages and disadvantages of the models. Confusion matrices give a thorough picture of the model's performance by showing the quantity of true positives, true negatives, false positives, and false negatives. Plots of feature importance indicate each feature's relative relevance inside the model, revealing which features are most crucial to the prediction job. Lastly, the model's performance is shown across several categorization thresholds using ROC (Receiver Operating Characteristic) curves, which illustrate the trade-off between sensitivity and specificity.
7. Final Model Selection: Using domain expertise and a variety of indicators to assess each model's performance is the last step in the model selection process. Accuracy, precision, recall, specificity, F1-score, and area under the ROC curve (AUC-ROC) are a few examples of performance measures. It's critical to properly balance each statistic, considering the goals of the domain and the fallout from incorrect categorization. Domain expertise can help spot inconsistent or unrealistic findings in the model's behaviour and facilitate the understanding of the model's performance.



**Figure 1: Methodology Flowchart**

**ROC Curve Analysis:**

The performance of the models was further highlighted by the ROC curves and associated AUC values. With AUC values of 0.91 and 0.92, respectively, both the Random Forest and Gradient Boosting models performed better than the others, showing good sensitivity and specificity in the identification of Parkinson's disease. On the other hand, the AUC values of the KNN and Decision Tree models were somewhat lower, at 0.81, indicating modest predictive ability.

|  |
| --- |
|  |

**Figure 5: ROC curve Comparison**

**Prediction Results:**

Forecasts for the input data showed that every model categorized the information as not suggestive of Diabestes disease. Although these findings are in line with the goals of our work, more validation on a variety of datasets is required to guarantee the models' dependability in actual clinical situations.

**input\_data = (197.07600, 206.89600, 192.05500, 0.00289, 0.00001, 0.00166, 0.00168, 0.00498, 0.01098, 0.09700, 0.00563, 0.00680, 0.00802, 0.01689, 0.00339, 26.77500, 0.422229, 0.741367, -7.348300, 0.177551, 1.743867, 0.085569)**

**Random Forest Prediction: does not suffer from Parkinson's illness.**

**Decision Tree Prediction: does not suffer from Parkinson's illness.**

**RESULTS AND DISCUSSION**

**Accuracy and Performance Metrics Interpretation:**

With an accuracy of 94.05%, we found that the Random Forest models had the greatest accuracy rates. This demonstrates how well these models can discriminate between those who have Diabetes disease and those who do not. On the other hand, the Decision Tree model attained an accuracy of 91.92%.

**Table 1: Accuracy Acquired**

|  |  |
| --- | --- |
| ALGORITHM | ACCURACY |
| Random Forest Accuracy | 0.94105 |
| Decision Tree Accuracy | 0.91192 |

**Conclusion**

Our study on Parkinson's Disease Detection Using Machine Learning Algorithms demonstrates the significant potential of machine learning in enhancing the early diagnosis and identification of Parkinson's disease. By conducting an extensive examination and assessment of many machine learning models, we have exhibited encouraging outcomes in precisely differentiating between those with and without Parkinson's disease. Our results demonstrate how well machine learning algorithms in particular, Random Forest and Gradient Boosting perform in terms of attaining strong performance indicators and high accuracy rates. The models provide exceptional sensitivity and specificity in identifying Parkinson's disease, as demonstrated by their balanced performance across confusion matrices and high AUC values. Additionally, the feature significance analysis underlines the critical role that clinical symptoms including stiffness, postural instability, and tremors play in the diagnosis of Parkinson's disease. Through the utilization of these fundamental characteristics, machine learning models can offer significant perspectives on the course of the illness and facilitate customized therapeutic strategies. We accept the limits of our work, particularly the need for bigger and more varied datasets for model validation, despite these encouraging results. Additionally, to improve the precision and applicability of Parkinson's disease detection models, more investigation is necessary into ensemble approaches and sophisticated feature engineering techniques.