A SURVEY ON PREDICTION OF DIABETIC LEVELS USING MACHINE LEARNING ALGORITHAM

Mrs. M. Vasuki[1], Dr. T. Amalraj Victoire[2], Suresh kumar S[3],

[1]Associate Professor, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Pondicherry, India

[2]Associate Professor, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Pondicherry, India

[3]PG Student, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Pondicherry, India

dheshna@gmail.com; amalrajvictoire@gmail.com; sk17kumar7@gmail.com

ABSTRACT

Diabetes is a serious, long-lasting illness that makes blood sugar levels high. It can lead to many health problems. By 2040, it's expected that 1 in 10 adults worldwide will have diabetes, reaching a staggering 642 million people. This is a big concern and needs attention. Scientists are using advanced technology like machine learning to predict diabetes. They studied data from hospital check-ups in Luzhou, China, which had 14 pieces of information about each person. They used two different methods like decision trees, random forests to make predictions. They tested these methods to see how well they worked, using a technique called five-fold cross-validation. To make sure their findings were reliable; they also did separate tests with a large group of healthy people and those with diabetes. Because the data wasn't evenly balanced between healthy and diabetic people, they adjusted it to make it fair. They found that random forests were the best at predicting diabetes accurately, with about 81% accuracy when using all the information available. They also used techniques like principal component analysis and minimum redundancy maximum relevance to make the data easier to work with.

KEYWORDS : Diabetes, blood sugar, predictions, hospital, China, methods, tests, data, balance, forests, accuracy, analysis.

1INTRODUCTION

Diabetes is a serious health problem where the sugar level in the blood is too high. It happens because the body either doesn't make enough insulin or can't use it properly. This can cause long-term damage to different parts of the body like the eyes, kidneys, heart, blood vessels, and nerves. There are two main types: type 1 and type 2. Type 1 usually affects younger people and causes symptoms like thirst, frequent urination, and high blood sugar levels. Treatment often involves insulin therapy. Type 2 is more common in middle-aged and older adults and is often linked to obesity, high blood pressure, high cholesterol, and other conditions.

As more people are getting diabetes, it's important to find ways to diagnose it quickly and accurately. Doctors usually diagnose diabetes by checking blood sugar levels after fasting, with glucose tolerance tests, or random blood sugar tests. The earlier we detect diabetes, the easier it is to manage. Machine learning can help by analyzing people's routine health data and giving an initial indication of whether they might have diabetes, which can assist doctors in their diagnosis.

There are many different algorithms used to predict diabetes. Some traditional methods include support vector machine (SVM), decision tree (DT), and logistic regression.

These methods help identify important features and choose the right classifier for prediction.

Decision trees, random forests, support vector machine are commonly used in predicting diabetes. Decision trees are good at classifying data, while random forests create many decision trees to improve accuracy.

Types of Diabetes

1. **Decision Trees:**

- Output: Decision trees produce a set of rules that can be easily interpreted. These rules form a tree-like structure where each node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or class label.
- Performance: Decision trees are prone to overfitting, especially when dealing with complex datasets with many features. They may not generalize well to unseen data if not appropriately pruned or regularized. However, they are fast to train and can handle both numerical and categorical data well.

Advantage:

- **Interpretability:** Decision trees provide a clear and understandable decision-making process. The tree-like structure represents a sequence of decisions based on features, which can be easily interpreted and explained to non-experts.
- **Handling Both Numerical and Categorical Data:** Decision trees can handle both numerical and categorical data without the need for extensive preprocessing. This makes them versatile for various types of datasets.
- **Non-Parametric:** Decision trees do not make any assumptions about the underlying distribution of the data. They are considered non-parametric models, which means they are flexible and can fit complex relationships between features and the target variable.
- **Feature Importance:** Decision trees can implicitly rank features based on their importance in predicting the target variable. Features that appear higher in the tree and closer to the root node are considered more important in making predictions.
- **Ease of Use:** Decision trees are relatively easy to implement and visualize. They do not require complex mathematical calculations, making them accessible to practitioners and beginners in machine learning.

Disadvantages:

- **Overfitting:** Decision trees are prone to overfitting, especially when the tree becomes too deep or complex. This occurs when the model captures noise or specific patterns in the training data that do not generalize well to unseen data.

- **High Variance:** Decision trees can be sensitive to small variations in the training data, leading to high variance in the model's predictions. This sensitivity can result in instability and inconsistency in the model's performance.
- **Instability:** Decision trees are sensitive to changes in the dataset, such as adding or removing data points or features. Small changes can lead to different tree structures, which may affect the model's performance and reliability.
- **Bias Towards Features with Many Levels:** Decision trees tend to favor features with a large number of levels or categories, as they can create more splits and potentially overfit the data. This bias can lead to imbalanced splits and suboptimal tree structures.
- **Difficulty Capturing Linear Relationships:** Decision trees are not well-suited for capturing linear relationships between features and the target variable. They may require additional preprocessing or transformations to handle such relationships effectively.

2. **Random Forests:**

    - Output: Random forests are an ensemble learning method that builds multiple decision trees and combines their predictions. The output is a consensus decision based on the predictions of all individual trees. This ensemble approach reduces overfitting and increases accuracy.
    - Performance: Random forests are highly robust and effective in handling large datasets with high dimensionality. They offer improved accuracy compared to individual decision trees by reducing variance. They are less prone to overfitting and provide better generalization to unseen data.

Advantages:

1. **High Accuracy:** Random forests generally provide higher accuracy compared to individual decision trees. By combining multiple decision trees and averaging their predictions, random forests reduce overfitting and variance, leading to improved performance on unseen data.
2. **Reduced Overfitting:** Random forests mitigate overfitting, a common issue with decision trees, by aggregating predictions from multiple trees. Each tree is trained on a subset of the data and features, resulting in diverse trees that capture different aspects of the data distribution.
3. **Feature Importance:** Random forests can measure the importance of features in predicting the target variable. By evaluating the decrease in impurity (e.g., Gini impurity or entropy) caused by each feature, random forests provide insights into which features are most relevant for making predictions.
4. **Robustness to Noise:** Random forests are robust to noisy data and outliers due to their ensemble nature. Outliers and noisy data points are less likely to influence the overall prediction, resulting in more stable and reliable predictions.
5. **Parallel Training:** Random forests can be trained in parallel, making them suitable for large datasets and distributed computing environments. Each tree in the forest can be trained independently, allowing for efficient scaling to high-dimensional data.

Disadvantages:

1. **Complexity and Computation:** Random forests can be computationally expensive and require more resources compared to individual decision trees. Training and evaluating a large number of decision trees can increase computational time and memory usage, especially for large datasets.
2. **Less Interpretability:** While random forests offer high accuracy, their ensemble nature makes them less interpretable compared to individual decision trees. Understanding the underlying decision-making process of a random forest model can be challenging due to the complexity of multiple trees.
3. **Parameter Tuning:** Random forests have several hyperparameters that need to be tuned for optimal performance, such as the number of trees (n_estimators), maximum depth of trees, and minimum number of samples required to split a node. Finding the right combination of hyperparameters can require extensive experimentation and computational resources.
4. **Biased Towards Features with Many Levels:** Random forests may exhibit a bias towards features with a large number of levels or categories. Features with more levels can result in more splits in the trees, potentially leading to overfitting and suboptimal model performance.
5. **Memory Usage:** Random forests require storing multiple decision trees in memory, which can consume a significant amount of memory, especially for large forests with many trees. This increased memory usage may limit the scalability of random forests on memory-constrained systems.

3. **Support Vector Machines (SVMs):**

- Output: SVMs are a supervised learning algorithm used for classification and regression tasks. The output of an SVM is a hyperplane that separates data points into different classes with the maximum margin of separation. In classification, the output is the predicted class label.
- Performance: SVMs work well in high-dimensional spaces and are effective when the number of features is larger than the number of samples. They are particularly useful when there is a clear margin of separation between classes. However, SVMs can be sensitive to the choice of kernel and parameters, and they may require careful tuning for optimal performance.

Advantage:

- Effective in High-Dimensional Spaces: SVMs perform well in high-dimensional spaces, making them suitable for tasks involving datasets with many features. They can effectively handle datasets where the number of features exceeds the number of samples.
- Versatility with Kernel Functions: SVMs can handle both linear and non-linear relationships between features and the target variable through the use of kernel functions. By mapping the input data into a higher-dimensional space, SVMs can find complex decision boundaries that separate different classes.
- Robust to Overfitting: SVMs are less prone to overfitting, especially in high-dimensional spaces, due to their ability to maximize the margin of separation between classes. The margin acts as a regularization parameter, preventing the model from fitting noise in the data.

- Global Optimization: SVMs solve a convex optimization problem, which guarantees that they find the global optimal solution rather than getting stuck in local optima. This property ensures that SVMs converge to the best possible decision boundary given the data.
- Works Well with Small to Medium-Sized Datasets: SVMs are particularly effective when dealing with small to medium-sized datasets, where they can find a clear margin of separation between classes. They perform well in scenarios where the data is well-structured and the classes are well-separated.

Disadvantages:

- Sensitivity to Kernel Choice: The performance of SVMs can be sensitive to the choice of kernel function and its parameters. Selecting the appropriate kernel and tuning its parameters requires domain knowledge and experimentation, which can be time-consuming and computationally expensive.
- Computationally Intensive: Training SVMs can be computationally intensive, especially for large datasets. As the size of the dataset increases, the time and memory requirements for training SVMs also increase, making them less scalable for big data applications.
- Limited Interpretability: SVMs provide little insight into the underlying decision-making process due to their black-box nature. While they produce accurate predictions, understanding how the model arrives at those predictions can be challenging, especially for non-linear kernels.
- Memory Usage: SVMs require storing support vectors in memory, which can consume a significant amount of memory, especially for large datasets with many support vectors. This increased memory usage may limit the scalability of SVMs on memory-constrained systems.
- Imbalanced Data: SVMs can be sensitive to class imbalance in the dataset, where one class has significantly fewer samples than the other(s). In such cases, the SVM may prioritize the majority class, leading to biased predictions and poor performance on the minority class.

Conclusion:

When determining the most suitable algorithm among Decision Trees, Random Forests, and Support Vector Machines (SVMs) to optimize accuracy, several pivotal factors must be considered. These include the inherent characteristics of the dataset, the intricacy of the problem at hand, and the computational resources available.

Decision Trees:

Decision Trees offer an intuitive approach to decision-making, making them attractive for scenarios prioritizing ease of interpretation. However, their susceptibility to overfitting, particularly in the presence of complex datasets, poses a significant challenge. Decision Trees tend to construct intricate models that can inadvertently capture noise, resulting in subpar performance when applied to unseen data.

Random Forests:

Random Forests tackle the issue of overfitting associated with Decision Trees by harnessing the power of ensemble learning. By amalgamating predictions from multiple decision trees, Random Forests enhance both accuracy and generalizability, especially when dealing with voluminous, multi-dimensional datasets featuring intricate inter-feature relationships. Moreover, they provide valuable insights into the importance of individual features, facilitating better interpretation.

Support Vector Machines (SVMs):

SVMs shine in scenarios characterized by high-dimensional data, adept at delineating complex decision boundaries. Their prowess is particularly evident in situations where clear class separation exists, leveraging their inherent robustness against overfitting by maximizing the margins of separation. However, their computational demands, especially when applied to large datasets, pose challenges in terms of scalability.

Additional Considerations:

Beyond accuracy and interpretability, several ancillary factors influence the selection of the most appropriate algorithm:

Data Imbalance: Algorithms may exhibit disparate performance when faced with datasets featuring imbalanced class distributions, with SVMs offering resilience through the implementation of appropriate class weighting techniques.

Feature Space: The dimensionality and structural intricacies of the features within the dataset play a crucial role in determining algorithmic suitability. Decision Trees and Random Forests demonstrate proficiency in handling both numerical and categorical data, whereas SVMs necessitate careful deliberation when selecting kernels to capture nonlinear relationships effectively.

Scalability: The algorithm's scalability with respect to dataset size and available computational resources is paramount. While Decision Trees and Random Forests demonstrate relative scalability, SVMs encounter hurdles due to their inherent computational complexity.

Optimal algorithm selection from among Decision Trees, Random Forests, and SVMs hinges on a comprehensive understanding of the dataset, the intricacies of the problem domain, and the specific requirements of the task at hand. Empirical evaluation across these algorithms serves as a cornerstone for gaining actionable insights into their performance and guiding informed decision-making.

Reference:

1. Gauri D. Kalyankar, Shivananda R. Poojara and Nagaraj V. Dharwadkar,” Predictive Analysis of Diabetic Patient Data Using Machine Learning and Hadoop”, International Conference On I-SMAC, 978-1-5090-3243-3, 2017.

2. Ayush Anand and Divya Shakti,” Prediction of Diabetes Based on Personal Lifestyle Indicators”, 1st International Conference on Next Generation Computing Technologies, 978-1-4673-6809-4, September 2015.

3. B. Nithya and Dr. V. Ilango,” Predictive Analytics in Health Care Using Machine Learning Tools and Techniques”, International Conference on Intelligent Computing and Control Systems, 978-1-5386-2745-7, 2017.

4. Dr Saravana kumar N M, Eswari T, Sampath P and Lavanya S,” Predictive Methodology for Diabetic Data Analysis in Big Data”, 2nd International Symposium on Big Data and Cloud Computing, 2015.

5. Aiswarya Iyer, S. Jeyalatha and Ronak Sumbaly Diagnosis of Diabetes Using Classification Mining Techniques”, International Journal of Data Mining & Knowledge Management Process (IJDKP), 5 (1) (January 2015)

6. P. Suresh Kumar and S. Pranavi “Performance Analysis of Machine Learning Algorithms on Diabetes Dataset using Big Data Analytics”, International Conference on Infocom Technologies and Unmanned Systems, 978-1-5386-0514-1, Dec. 18-20, 2017.

7. Mani Butwall, Shraddha Kumar A Data Mining Approach for the Diagnosis of Diabetes Mellitus using Random Forest Classifier”, International Journal of Computer Applications, 120 (Number 8, 2015)

8. K. Rajesh, V. Sangeetha ‚Application of Data Mining Methods and Techniques for Diabetes Diagnosis”International Journal of Engineering and Innovative Technology (IJEIT)