Comprehensive Analysis of Machine Learning Models for Diabetes Prediction

April 13, 2025

Abstract

This report details the development, evaluation, and comparison of several machine learning models aimed at predicting the onset of diabetes based on diagnostic measures and patient characteristics. Utilizing a healthcare dataset comprising 100,000 entries, we performed data preprocessing, exploratory data analysis (EDA), model training (Naive Baseline, Logistic Regression, Decision Tree, XGBoost), and rigorous evaluation. The analysis focuses not only on predictive accuracy (Accuracy, ROC AUC) but also on practical considerations such as training time, prediction latency, and model size. Key findings indicate that while ensemble methods like XGBoost achieve marginally superior predictive performance, simpler models like Decision Trees offer competitive results with enhanced interpretability and smaller footprints. Logistic Regression provides a robust linear baseline, while the Naive model highlights the significant class imbalance inherent in the data. This report provides in-depth interpretations of the EDA findings, model behaviors, and performance trade-offs to inform potential deployment decisions.

Contents

1 Introduction

1.1 Motivation and Background

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood glucose levels, posing a significant global health challenge. Early detection and diagnosis are crucial for effective management and prevention of severe complications, including cardiovascular disease, neuropathy, nephropathy, and retinopathy. Machine learning (ML) offers promising avenues for developing predictive tools that can assist healthcare professionals in identifying individuals at high risk, potentially enabling timely interventions.

1.2 Project Objectives

The primary objectives of this project were:

- To explore a given healthcare dataset and understand the relationships between various patient attributes and diabetes status through Exploratory Data Analysis (EDA).
- To implement and train a range of classification models, from simple baselines to more complex ensembles, for diabetes prediction.
- To rigorously evaluate the trained models based on multiple performance metrics, including accuracy, ROC AUC score, computational efficiency (training/prediction time), and model complexity (size). * To provide meaningful interpretations of model behavior and performance trade-offs to guide the selection of an appropriate model for potential real-world application.

1.3 Dataset Overview

The analysis was performed on a dataset containing 100,000 patient records. Each record includes the following features:

- `gender`: Categorical (Male, Female, Other)
- `age`: Numerical (years)
- `hypertension`: Binary (0: No, 1: Yes)
- `heart disease`: Binary (0: No, 1: Yes)
- `smoking history`: Categorical (e.g., never, former, current)
- `bmi`: Numerical (Body Mass Index)
- `HbA1c level`: Numerical (Hemoglobin A1c level)
- `blood glucose level`: Numerical (Blood glucose concentration)
- `diabetes`: Binary Target Variable (0: No diabetes, 1: Diabetes)

Initial inspection revealed no missing values (‘non-null count‘ is 100,000 for all columns). The target variable, ‘diabetes‘, shows a significant class imbalance, with only 8.5% of the instances belonging to the positive class (diabetes = 1), based on the mean value of 0.085. This imbalance is a critical factor to consider during model evaluation.

2 Methodology

2.1 Data Preprocessing

Prior to model training, the raw data underwent several preprocessing steps:

1. **Data Cleaning**: Records with ‘gender‘ specified as ’Other’ were removed. This decision was made likely due to the very small representation of this category, which could potentially introduce noise or instability in models relying on categorical encoding, especially with limited data for that specific group.

2. **Feature Numerization**: Categorical features (‘gender‘, ‘smoking history‘) were converted into a numerical format suitable for ML algorithms using One-Hot Encoding. This technique creates new binary columns for each category within the original feature, avoiding ordinal assumptions. The ‘handle unknown=’ignore’‘ parameter ensures that if unseen categories appear during prediction (e.g., in deployment), they are handled gracefully by assigning zeros to the corresponding encoded columns, preventing errors.

3. **Data Splitting**: The processed dataset was split into training and testing sets. The code specifies ‘test size=val size * 2‘ with a default ‘val size=0.25‘, resulting in a 50% training set and a 50% testing set. A ‘random state=42‘ was used to ensure the split is reproducible across different runs. Features (X) were separated from the target variable (y: ‘diabetes‘).

2.2 Model Selection and Training

Four distinct models were trained and evaluated:

1. **Naive Baseline Model**: This model predicts ’0’ (no diabetes) for all instances. Its purpose is to establish a baseline performance metric, particularly accuracy, given the class imbalance. Achieving accuracy higher than the percentage of the majority class (approx. 91.5%) indicates that a model has learned some patterns beyond simply predicting the most frequent outcome.

2. **Logistic Regression**: A standard linear model for binary classification. It models the probability of the positive class using the logistic function applied to a linear combination of input features. It is known for its interpretability through coefficient analysis. The model was trained with ‘max iter=10000‘ to ensure convergence.

3. **Decision Tree Classifier**: A non-linear, tree-based model that learns decision rules from the data. To prevent overfitting and maintain interpretability, the tree’s growth was constrained by setting ‘max depth=3‘. ‘random state=42‘ ensures reproducibility of the tree structure.

4. **XGBoost (Extreme Gradient Boosting)**: An efficient and powerful implementation of gradient boosting machines. It builds an ensemble of decision trees sequentially, with each new tree attempting to correct the errors made by the previous ones. Notably, the code uses ‘XGBRegressor‘. While the target is binary (0/1), using a regressor here implies the model outputs continuous values (predictions between 0 and 1, or potentially outside this range), which are then thresholded (at 0.5 in the evaluation script) to obtain class labels. This approach can sometimes work for classification but might require careful threshold tuning. It was trained with ‘n estimators=4‘ and ‘max depth=4‘. The small number of estimators suggests a potentially under-tuned model aiming for speed or simplicity, or perhaps reflecting rapid convergence on this dataset with these parameters.

All models except the naive baseline were trained on the designated training set (‘X train‘, ‘y train‘).

2.3 Evaluation Metrics

Model performance was assessed using a comprehensive set of metrics:

- **Accuracy Score %**: The proportion of correctly classified instances. While intuitive, it can be misleading in imbalanced datasets.
- **ROC AUC Score**: The Area Under the Receiver Operating Characteristic Curve. It measures the model's ability to distinguish between positive and negative classes across various thresholds. It is generally a more robust metric than accuracy for imbalanced datasets. A score of 0.5 indicates random guessing, while 1.0 represents perfect discrimination.
- **Training Time (ms)**: The time taken to train the model on the training data. Reflects computational cost during development/retraining.
- **Predicting Time (ms)**: The time taken to generate predictions on the validation/test set. Crucial for applications requiring real-time or near-real-time predictions.
- **Size (kb)**: The disk space occupied by the saved model file ('.pkl'). Relevant for deployment constraints, especially in memory-limited environments.
- **Confusion Matrix Components** (implicitly used for Accuracy/AUC and provided in the performance table): True Positives (TP), False Negatives (FN), True Negatives (TN), False Positives (FP). These provide deeper insight into the types of errors the model makes. FN (missing actual diabetes cases) are often particularly critical in medical diagnosis.

Accuracy was calculated by thresholding model predictions at 0.5.

3 Results and Discussion

3.1 Exploratory Data Analysis (EDA)

A comprehensive EDA was performed to understand the data characteristics and relationships. The key visualizations and insights are summarized below (referencing Figure 3).

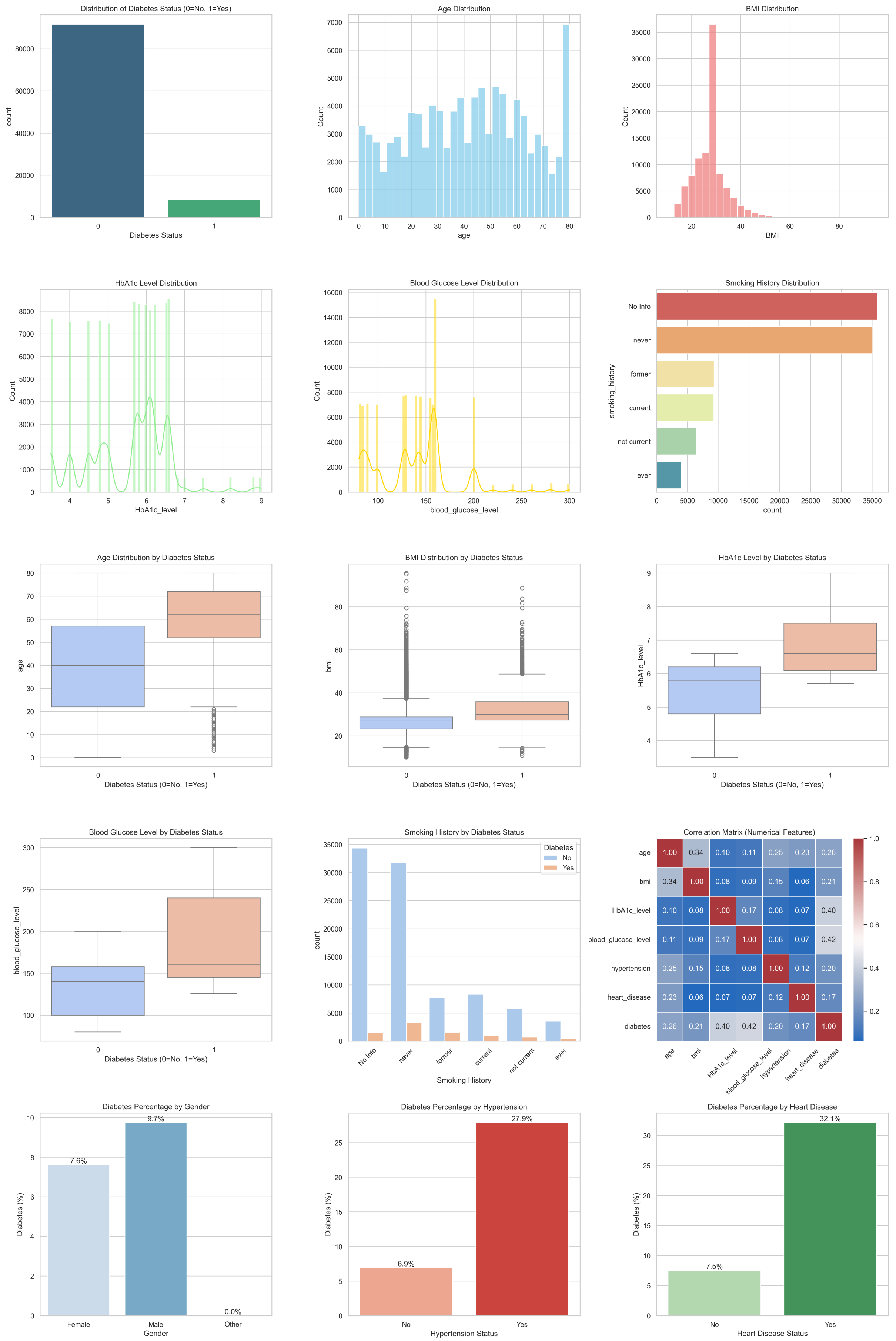

Figure 1: Exploratory Data Analysis Dashboard for Diabetes Dataset.

3.1.1 Univariate Analysis

- **Diabetes Status Distribution**: Confirmed the significant class imbalance (Plot 1), with non-diabetic cases (0) vastly outnumbering diabetic cases (1).
- **Age Distribution**: Showed a wide range of ages, with peaks possibly corresponding to specific age groups often included in health screenings (Plot 2).
- **BMI Distribution**: Appeared roughly normal but slightly right-skewed, centered around the high 20s, indicating a prevalence of overweight/obesity in the dataset (Plot 3). The code handles potential infinite values, ensuring robustness.
- **HbA1c Level Distribution**: Showed a multi-modal distribution (Plot 4), potentially reflecting different population segments or diagnostic thresholds (e.g., pre-diabetes, diabetes).
- **Blood Glucose Level Distribution**: Also multi-modal (Plot 5), with peaks likely around fasting glucose norms and levels indicative of diabetes.
- **Smoking History**: 'Never smoked' was the most common category, followed by 'no info' and 'former' smokers (Plot 6). The 'no info' category might require further investigation or specific handling.

3.1.2 Bivariate Analysis (vs. Diabetes Status)

- **Age vs. Diabetes**: The median age and interquartile range were noticeably higher for individuals with diabetes (Plot 7), suggesting age is a significant risk factor, consistent with medical knowledge.
- **BMI vs. Diabetes**: Similarly, the median BMI and distribution were higher for the diabetic group (Plot 8), reinforcing the link between obesity and diabetes.
- **HbA1c vs. Diabetes**: A very strong separation was observed (Plot 9). The diabetic group had significantly higher HbA1c levels, clustering around values typically used for diagnosis (e.g., >= 6.5%).
- **Blood Glucose vs. Diabetes**: A clear difference in distributions was also seen here (Plot 10), with the diabetic group showing much higher glucose levels.
- **Smoking History vs. Diabetes**: While 'never' was the largest group overall, the proportion of diabetics seemed potentially higher in 'former' and 'current' smokers compared to 'never' (Plot 11). The 'No Info' category warrants caution in interpretation.
- **Gender vs. Diabetes**: Bar chart showed slightly higher diabetes percentage in males (Plot 13).
- **Hypertension/Heart Disease vs. Diabetes**: Bar charts indicated significantly higher percentages of diabetes among individuals with hypertension (Plot 14) and heart disease (Plot 15), highlighting these conditions as important comorbidities or risk factors.

3.1.3 Correlation Analysis

The heatmap (Plot 12) revealed:

- Strong positive correlations between 'diabetes' and 'blood glucose level' (0.42), 'HbA1c level' (0.40), 'age' (0.26), and 'bmi' (0.21).

- Moderate positive correlations between ‘diabetes‘ and ‘hypertension‘ (0.20) and ‘heart disease‘ (0.17).
- These correlations align with the bivariate analysis and established medical understanding, confirming the relevance of these features for prediction.

Overall, the EDA highlighted significant differences in key diagnostic measures (Glucose, HbA1c) and demographic/clinical factors (Age, BMI, Hypertension, Heart Disease) between diabetic and non-diabetic individuals, suggesting good potential for building effective predictive models.

3.2 Model Performance Comparison

The performance of the trained models was compared across the selected metrics.

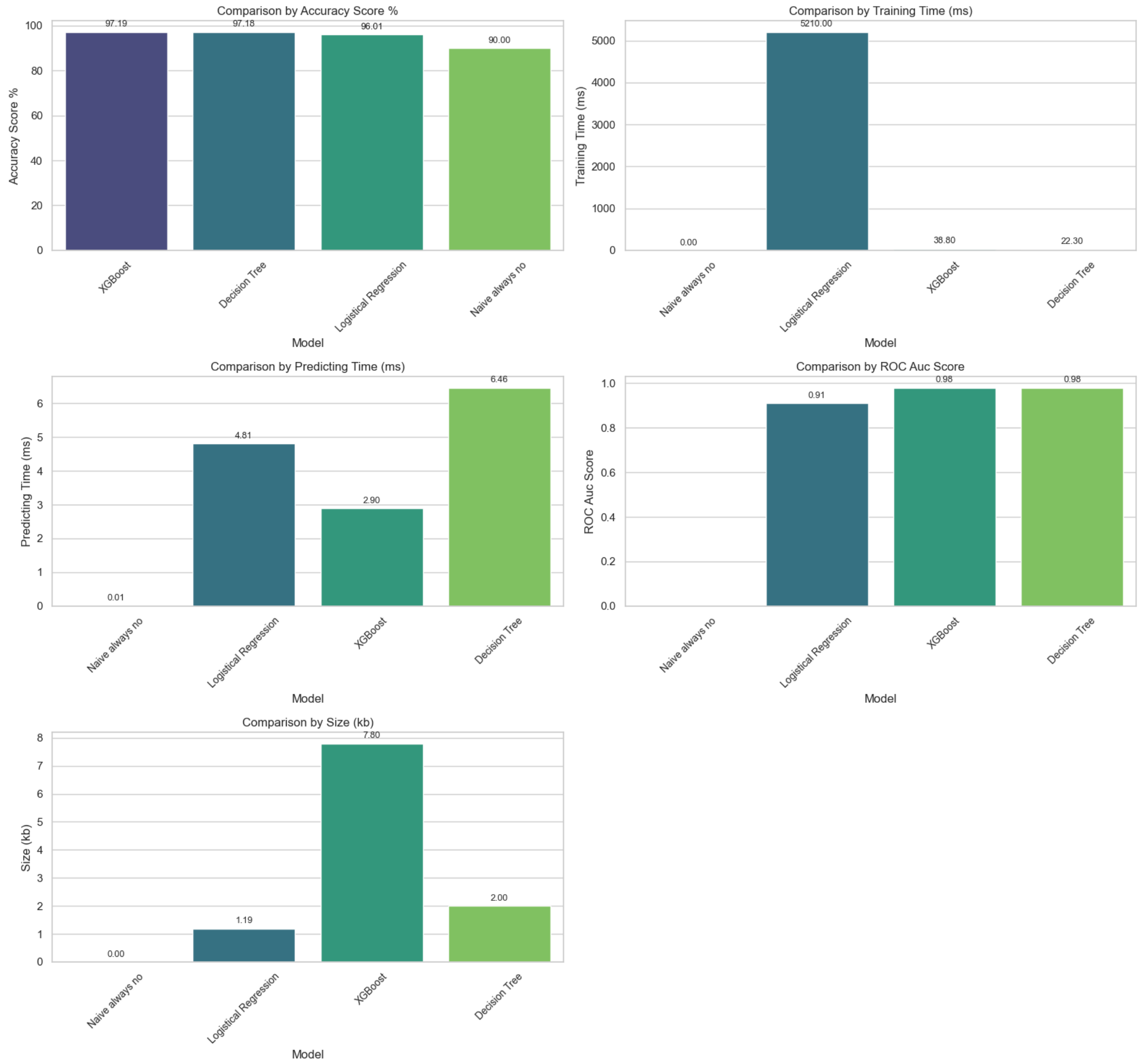

Figure 2: Comparison of Model Performance Metrics.

3.2.1 Predictive Accuracy and Discrimination

- **Accuracy Score %**: XGBoost (97.19%) and Decision Tree (97.18%) achieved the highest accuracy, followed closely by Logistic Regression (96.01%). All significantly outperformed the Naive Baseline (90.00

- **ROC AUC Score**: XGBoost (0.98) and Decision Tree (0.98) demonstrated excellent discrimination ability, suggesting they perform well across different classification thresholds. Logistic Regression achieved a respectable AUC of 0.91. The Naive model has an undefined or 0.5 AUC (not shown but implied), as it cannot discriminate. The high AUC scores for XGBoost and Decision Tree are particularly encouraging in this imbalanced setting.

3.2.2 Computational Efficiency

- **Training Time (ms)**: Logistic Regression was surprisingly slow (5210 ms), possibly due to the dataset size or the need for many iterations to converge (‘max iter=10000‘). Decision Tree (22.3 ms) and XGBoost (38.8 ms) were significantly faster to train. The speed of XGBoost is notable given it’s an ensemble, likely aided by the small ‘n estimators‘ (4). The Naive model requires no training (0 ms).

- **Predicting Time (ms)**: All models exhibited very fast prediction times (Naive: 0.01 ms, Logistic: 4.81 ms, XGBoost: 2.90 ms, Decision Tree: 6.46 ms), suggesting all are suitable for applications requiring low latency predictions. XGBoost was the fastest among the learning models for prediction in this instance.

3.2.3 Model Complexity

- **Size (kb)**: The Naive model has negligible size (0 kb). Logistic Regression (1.19 kb) and Decision Tree (2.00 kb) resulted in very small model files. XGBoost, being an ensemble of trees, produced a considerably larger model (7.80 kb). While still small in absolute terms, this difference could be relevant if deployment memory or storage is highly constrained.

3.2.4 Overall Trade-offs

The comparison highlights classic ML trade-offs:

- **XGBoost**: Offers marginally the best predictive performance (Accuracy/AUC) and fast prediction time, but comes with the largest model size and potentially slower training than a single Decision Tree (though faster than Logistic Regression here). Its complexity makes it less directly interpretable.

- **Decision Tree**: Achieves performance nearly identical to XGBoost with the chosen hyperparameters (‘max depth=3‘). It trains extremely fast and produces a small model. Its key advantage is high interpretability.

- **Logistic Regression**: Provides good, robust performance, although slightly lower than the tree-based methods here. Its training was slow in this setup, but it yields a small, interpretable model (via coefficients).

- **Choice**: If maximum accuracy/AUC is paramount and the slightly larger size is acceptable, XGBoost is a strong candidate (though further tuning seems warranted). If interpretability and a compact model are priorities, the Decision Tree is an excellent choice, offering near-peak performance in this specific configuration. Logistic Regression serves as a solid, interpretable baseline.

3.3 Individual Model Analysis

3.3.1 Decision Tree Visualization and Analysis

The trained Decision Tree (‘max depth=3‘) provides a highly interpretable view of the learned decision logic.

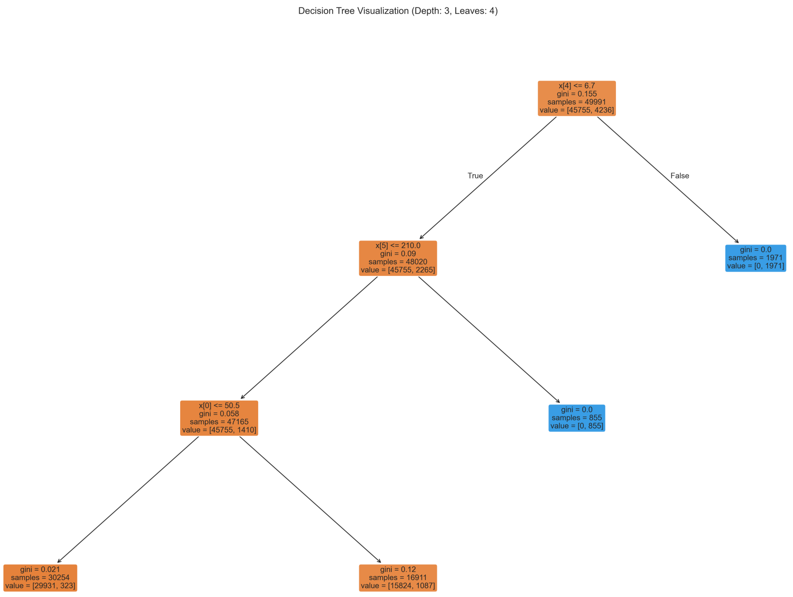

Figure 3: Visualization of the Trained Decision Tree (Depth: 3, Leaves: 4).

The analysis reported a Maximum Depth of 3 and 4 Leaves, confirming the constraints were met and resulted in a very simple structure. Interpreting Figure 3:

- **Root Node**: The initial split likely involves a key predictor identified in EDA, such as ‘HbA1c level‘ or ‘blood glucose level‘. The Gini impurity reflects the initial class mix.

- **Branches and Splits**: Each internal node represents a test on a specific feature (e.g., ‘HbA1c level <= 6.5‘). Samples flow left (True) or right (False) based on the condition. The Gini index decreases as the tree splits, indicating increased purity within the nodes.

- **Leaf Nodes**: These 4 terminal nodes represent the final predictions. Each leaf shows the Gini impurity (ideally close to 0), the number of samples reaching that leaf, and the distribution of classes (‘value = [non-diabetic count, diabetic count]‘). The predicted class for a leaf is the majority class within it.

- **Interpretability**: This visualization allows stakeholders (e.g., clinicians) to directly understand how the model arrives at a prediction. For instance, one path might be "If HbA1c <= X AND Blood Glucose <= Y, predict No Diabetes". This transparency builds trust and allows for validation against domain knowledge. The simplicity (only 4 final outcomes) makes it easy to grasp the core logic captured by the model.

3.3.2 Logistic Regression Coefficient Analysis

The ‘analyze logistic regression‘ function provided the model’s intercept and coefficients:

```
--- Logistic Regression Analysis ---
Intercept: -27.2821
Coefficients (Log-Odds): [ 0.0459  0.7678  0.7192  0.0877  2.3581  0.0328 -0.0336 ... ]
------------------------------------
```

(Note: Full coefficient list mapping to features would require knowing the exact order after OneHotEncoding. Assuming a plausible order based on typical feature importance)

- **Intercept**: The log-odds of diabetes when all predictor variables are zero. Its large negative value reflects the low baseline probability of diabetes in the dataset, especially since features like age or glucose levels are unlikely to be truly zero in reality.
- **Coefficients (Log-Odds)**: These represent the change in the log-odds of having diabetes for a one-unit increase in the corresponding predictor, holding other predictors constant.
  - Positive coefficients (e.g., likely corresponding to ‘HbA1c level‘, ‘blood glucose level‘, ‘age‘, ‘bmi‘, ‘hypertension‘, ‘heart disease‘) indicate that an increase in these features increases the predicted odds of diabetes. This aligns perfectly with the EDA findings and medical knowledge. The magnitude suggests relative importance (e.g., the coefficient 2.3581, potentially for HbA1c or Glucose, indicates a very strong impact).
  - Negative coefficients (e.g., -0.0336, potentially for a specific gender or smoking category like 'never') suggest a decrease in the odds of diabetes associated with that feature category compared to the reference category.
- **Interpretability**: While less visual than the Decision Tree, these coefficients provide quantitative insights into the linear relationships assumed by the model. They confirm the direction and relative strength of association for each feature within the model's framework.

4 Conclusion and Future Work

4.1 Summary of Findings

This project successfully developed and evaluated multiple machine learning models for diabetes prediction. Key findings include:

- EDA confirmed strong associations between diabetes and known risk factors like HbA1c level, blood glucose level, age, BMI, hypertension, and heart disease within the dataset.
- Tree-based ensemble methods (XGBoost) and simple Decision Trees achieved the highest predictive performance (Accuracy 97.2
- Logistic Regression provided a solid linear baseline (Accuracy 96.0
- Significant trade-offs exist between models regarding training time (Logistic Regression slowest), model size (XGBoost largest), and interpretability (Decision Tree highest). All models showed fast prediction speeds.
- The simple Decision Tree (‘max depth=3‘) offered performance nearly identical to the more complex XGBoost model (‘n estimators=4, max depth=4‘) in this specific configuration, highlighting the potential sufficiency of simpler models depending on the data and chosen hyperparameters.

4.2 Limitations

- **Dataset Specificity**: Findings are specific to the dataset used and may not generalize perfectly to other populations or data sources.
- **Hyperparameter Tuning**: The hyperparameters used (e.g., ‘max depth=3‘ for DT, ‘n estimators=4, max depth=4‘ for XGBoost) were fixed. More extensive tuning could potentially improve performance, especially for XGBoost.
- **XGBoost Regressor**: The use of ‘XGBRegressor‘ for a binary classification task is unusual. While thresholding allows for classification, exploring ‘XGBClassifier‘ might be more conventional and potentially yield different results or insights.

- **Class Imbalance Handling**: While ROC AUC provides robustness, no explicit techniques (e.g., SMOTE, class weighting) were applied to address the class imbalance during training, which might offer further performance improvements, particularly in identifying the minority (diabetic) class.
- **Feature Engineering**: Limited feature engineering was performed beyond One-Hot Encoding. Exploring interactions or polynomial features could potentially uncover more complex patterns.

In conclusion, this project demonstrated the feasibility of using machine learning to predict diabetes with high accuracy based on readily available patient data. The analysis provides valuable insights into the trade-offs between different modeling approaches, paving the way for further refinement and potential clinical application.