Sri Lanka Institute of Information Technology


Artificial Intelligence and Machine Learning | IT2011

Year 02 Semester 01 – 2025

**Group Assignment – Bias and Ethics in AI Report Submission**

Group ID – 55 (2025-Y2-S1-MLB-B3G2-05)

# AI for Early Diabetes Prediction - Final Group Report

Group Members –

| Student ID | Student Name |
|---|---|
| IT24100300 | Dulakshika A.L.S.H |
| IT24100263 | Dilhani W. P. K. A |
| IT24100239 | Disanayaka D.M.C.N |
| IT24100283 | Devinda S.U.V |
| IT24100314 | Dahanayake L.K |
| IT24100237 | Bandara R. M. G. L |


20/10/2025

# 1.Introduction and Problem Statement

Diabetes is a major global health challenge, affecting millions of people worldwide. A significant issue with this disease is that it can develop for years with only mild or no symptoms, often going undetected until serious complications - such as heart disease, kidney failure, or nerve damage - have already occurred. This delay in diagnosis severely impacts patients' quality of life and places a heavy burden on healthcare systems.

The core problem we address is the **need for early and accessible identification of individuals at high risk of developing diabetes**. Traditional diagnostic methods often require a physical visit and specific tests, which might not be accessible to everyone. This is where Artificial Intelligence (AI) and Machine Learning (ML) can play a transformative role.

Therefore, this project aims to develop a **machine learning model for the early prediction of diabetes**. Our solution is designed not as a standalone diagnostic tool, but as an **intelligent assistant for healthcare professionals**. By analyzing key health indicators, the model will calculate a personalized risk score to help flag at-risk individuals for further medical evaluation. The ultimate goal is to enable earlier intervention, improve patient outcomes, and contribute to more efficient healthcare delivery.

# 2.Dataset Description

**Dataset Source & Overview:**
The primary dataset for this project is the **"Diabetes Prediction Dataset"** sourced from Kaggle. The direct link is:

https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset

This dataset is a collection of individual medical records and is ideally suited for a binary classification task, where the goal is to predict the presence (1) or absence (0) of diabetes.

**Key Characteristics:**

- **Size:** The dataset contains **100,000 records**, which far exceeds the project's minimum requirement of 1,000 records.
- **Features:** It includes **8 features** (exceeding the 6-feature minimum) that are well-known risk factors for diabetes. These features are:
    - gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level.
- **Target Variable:** diabetes (0 = No, 1 = Yes).

**Relevance and Justification:**
This dataset is highly relevant as it contains **critically important clinical markers** for diabetes prediction, most notably HbA1c_level and blood_glucose_level. The substantial size of 100,000 records allows for robust model training and evaluation.

**Potential Limitations and Biases:**

- **Class Imbalance:** A key characteristic of this dataset is a significant **class imbalance** in the target variable. The number of non-diabetic cases (class 0) vastly outnumbers the diabetic cases (class 1). This is a common challenge in medical diagnosis datasets but is crucial to address, as a model could achieve high accuracy by simply always predicting "no diabetes," which would be medically useless. This imbalance will be a central focus during our preprocessing and model evaluation stages.
- **Inherent Biases:** The data may contain other inherent biases related to the demographic or geographic population from which it was collected, which could affect the model's fairness and generalizability.

# 3.Preprocessing & Exploratory Data Analysis (EDA)

This section details the steps taken to clean the data, handle its inherent challenges, and gain insights through visualization, forming a robust foundation for model development.

## 3.1 Data Cleaning and Preprocessing Pipeline

Our preprocessing was a collaborative effort, with each member handling a specific, critical task:

- **Handling Incorrect/Implausible Data & Missing Values (IT24100314):** The dataset was first scanned for physiologically implausible values (e.g., negative BMI, impossibly high blood glucose levels). Such entries were identified and treated as missing values. Subsequently, all missing values (both original and those generated from the previous step) were handled. Given the dataset's large size, rows with critical missing clinical markers like HbA1c_level or blood_glucose_level were removed to preserve data integrity. For other features with minimal missingness, imputation strategies (mean/median/mode) were considered and applied.
- **Encoding Categorical Variables:** This was performed using two techniques to compare their impact.
    - **One-Hot Encoding (IT24100283):** Applied to the gender feature. This creates binary (0/1) columns for each category (e.g., gender_Female, gender_Male, gender_Other) and avoids introducing a false ordinal relationship.
    - **Label Encoding (IT24100263):** Applied to ordinal-like categorical features such as smoking_history (e.g., 'never' < 'former' < 'current'), converting them into numerical integers that a model can interpret while preserving the order.
- **Outlier Handling & Data Splitting (IT24100239):** The processed dataset was first split into **80% training** and **20% testing** sets using stratification to ensure a representative distribution of the target variable. **Crucially, outlier handling was**

**applied only to the training set.** For numerical features in the training set, outliers were detected using the Interquartile Range (IQR) method and **capped (winsorized)**. The same capping values derived from the training set were then applied to the test set to prevent data leakage.

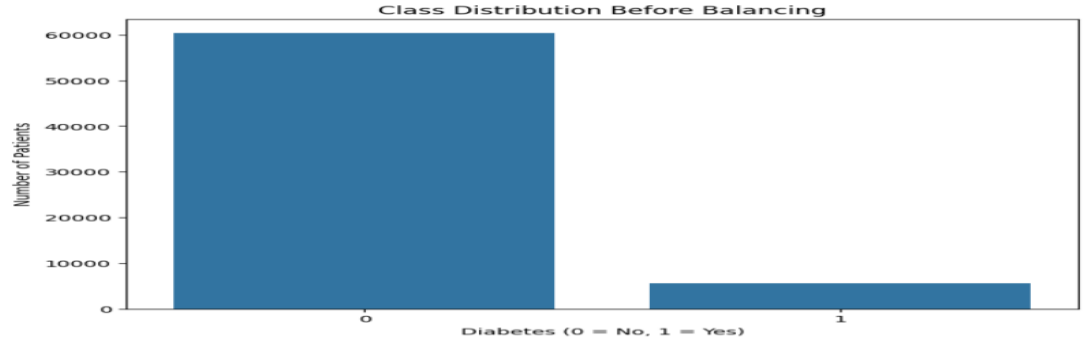
- **Addressing Class Imbalance (IT24100237):** The significant imbalance in the target variable (diabetes) was a primary concern. **Random Under-sampling** of the majority class (non-diabetic) was implemented to create a balanced dataset, thereby preventing the models from being biased towards predicting the majority class.
- **Normalization / Scaling (IT24100300):** Numerical features such as age, bmi, HbA1c_level, and blood_glucose_level were standardized using **StandardScaler**. This transforms the data to have a mean of 0 and a standard deviation of 1, which is crucial for models like SVM and KNN that are sensitive to the scale of features.

## 3.2 Exploratory Data Analysis (EDA) and Visualizations

EDA was conducted to understand data distributions, relationships, and the underlying structure of the problem. Each member contributed specific visualizations based on their preprocessing tasks.

- **Class Distribution Analysis (IT24100237):** A bar chart was created to visualize the severe class imbalance in the original dataset, justifying the need for undersampling.

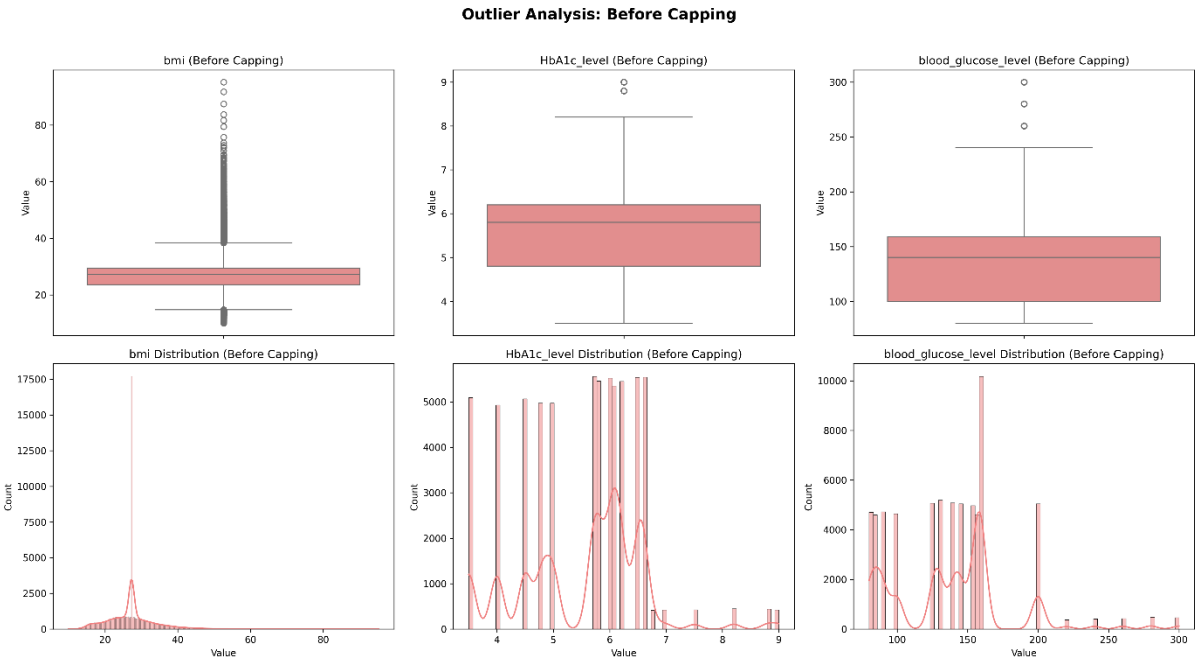
```
Dataset after outlier handling loaded successfully.
Class distribution before balancing:
diabetes
0    60424
1     5612
Name: count, dtype: int64
```

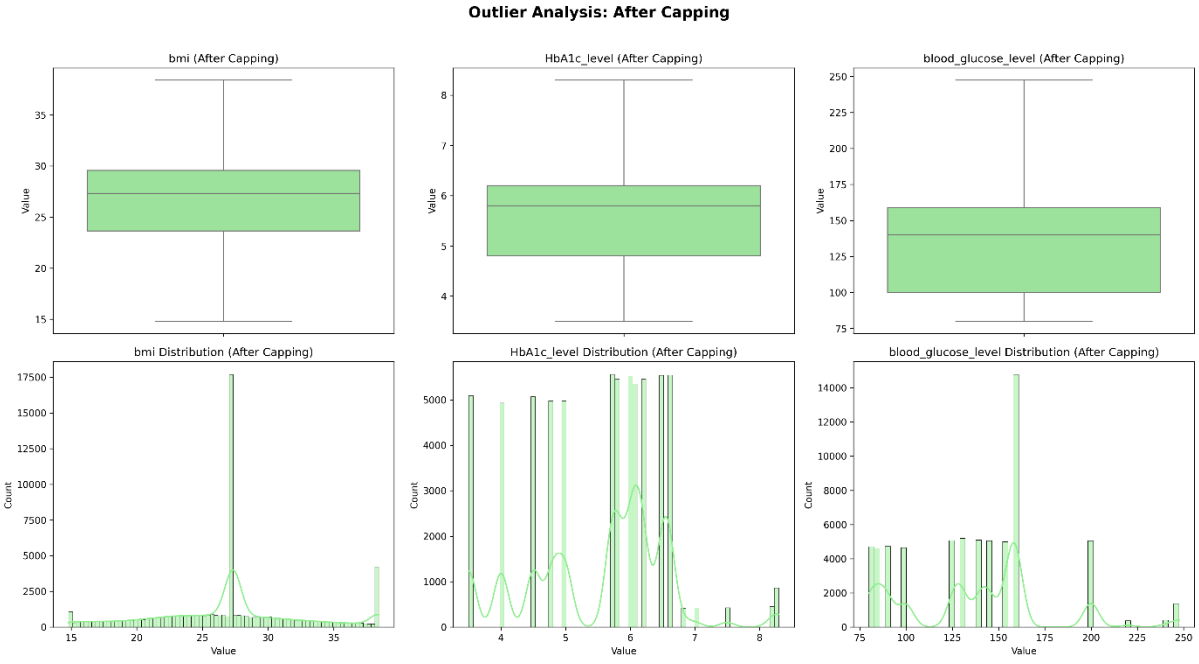



```
Class distribution of the TRAINING set after balancing:
diabetes
0    4490
1    4490
Name: count, dtype: int64
```

- **Insight:** The initial dataset was highly imbalanced. After undersampling, the classes are balanced, providing a fairer ground for model training.

- **Outlier Analysis (IT24100239):** Boxplots were generated for key numerical features like bmi, HbA1c_level, and blood_glucose_level to visually identify outliers before the capping treatment.

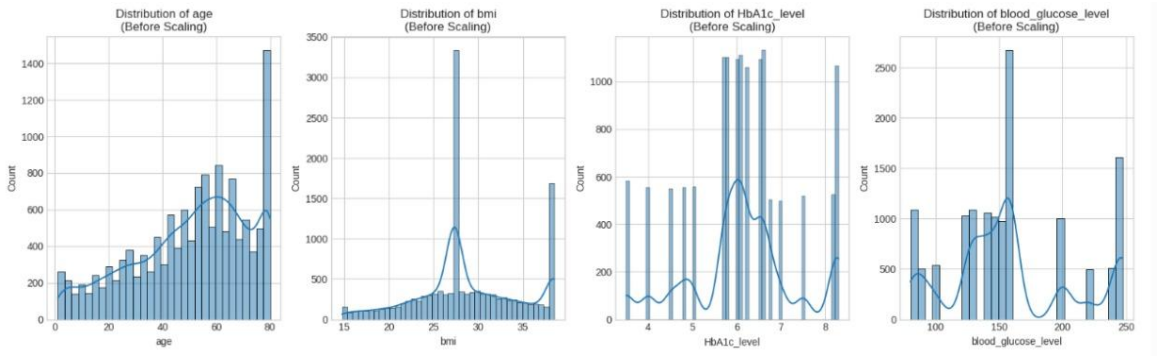


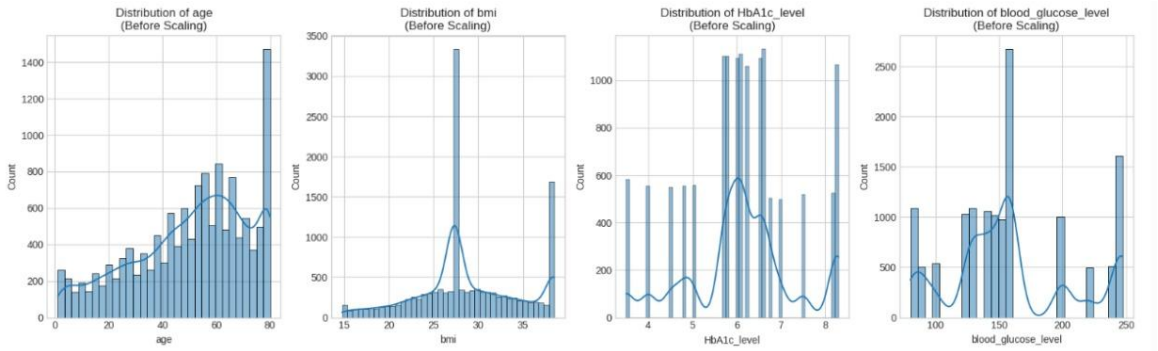



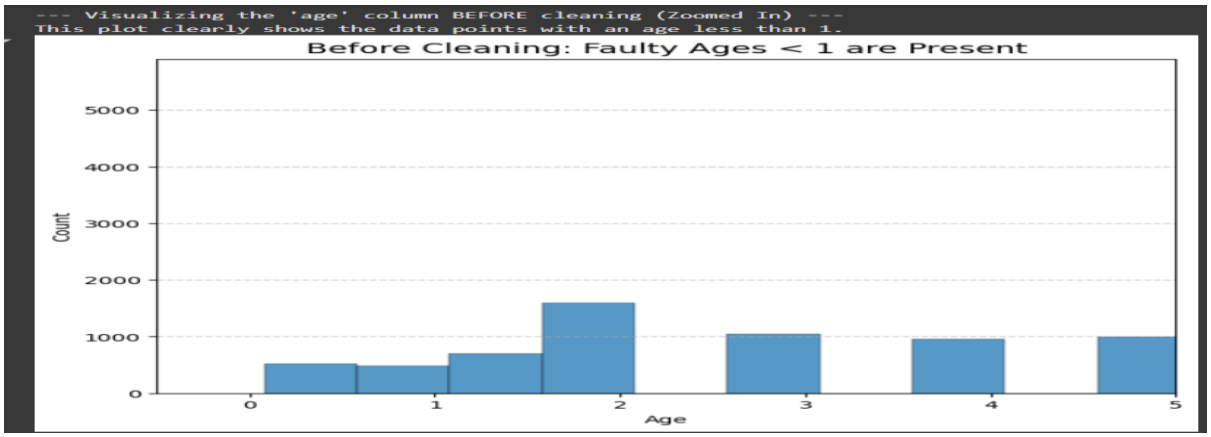

- **Insight:** Features like bmi contained significant outliers. Capping these values helped normalize the data distribution without losing entire records.

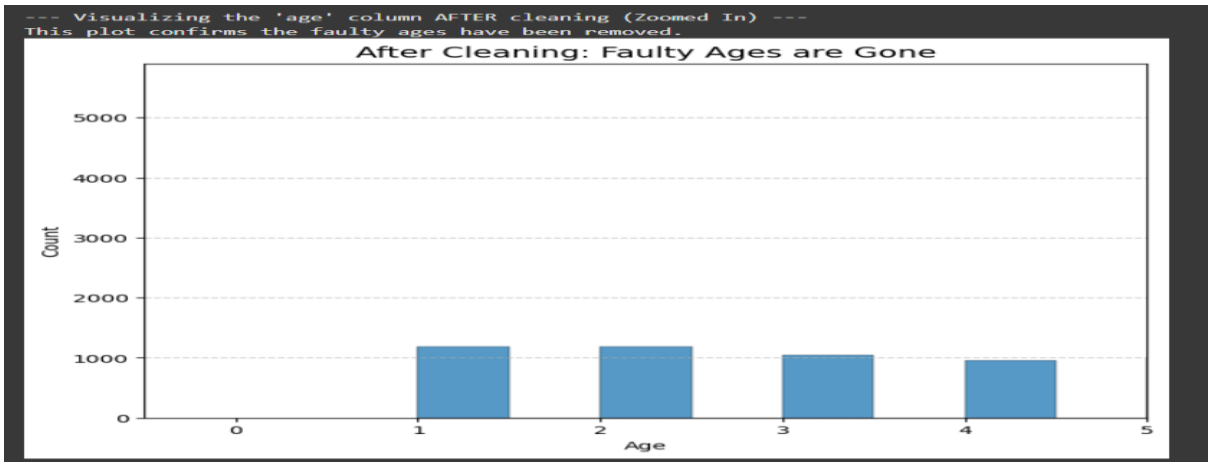
- **Analysis of Numerical Features (IT24100300 & IT24100314):** Boxplots and histograms were generated for key numerical features like age, bmi,

and HbA1c_level to understand their spread, central tendency, and skewness, and to identify outliers before and after treatment.



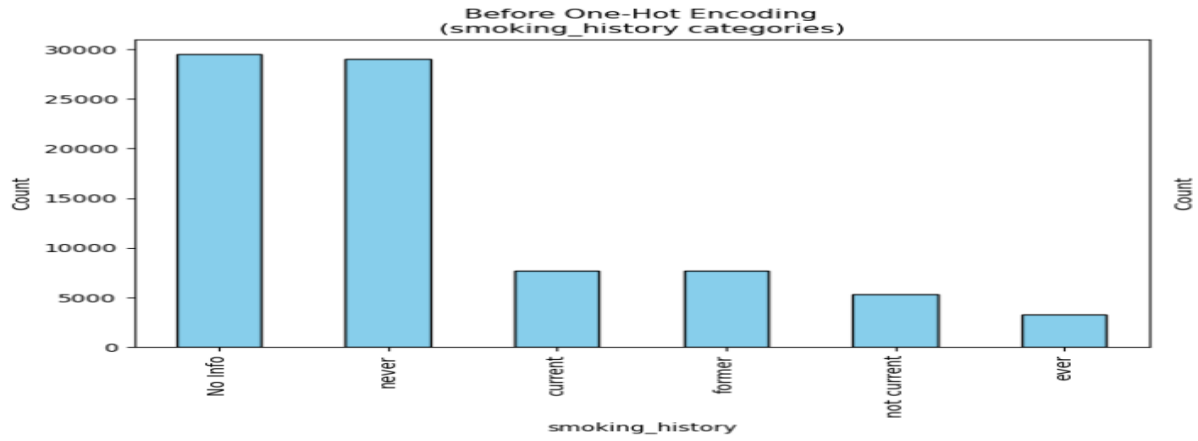



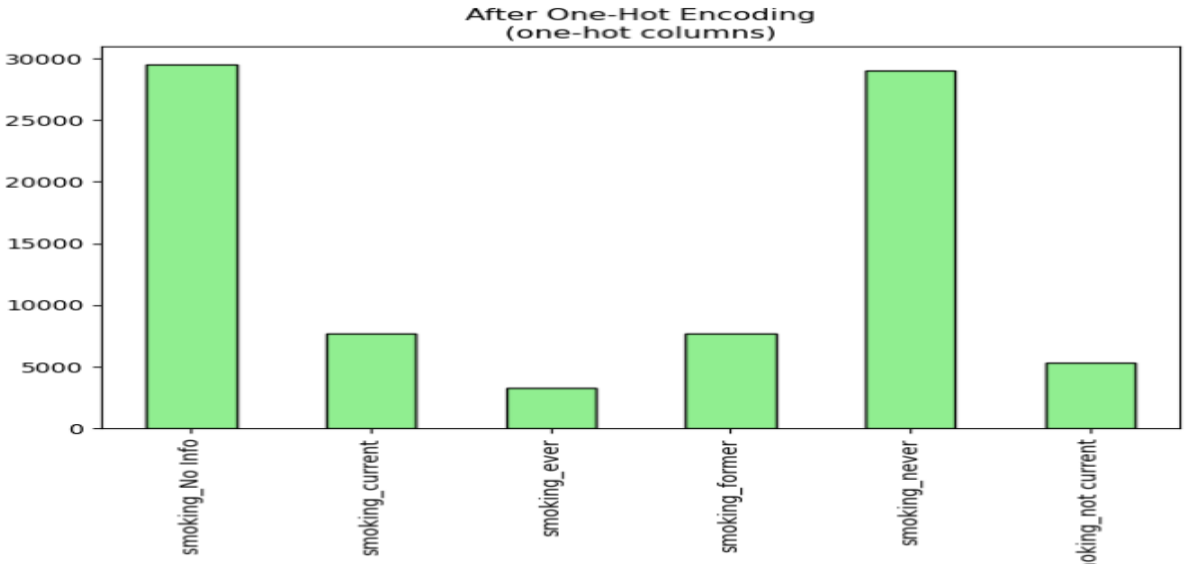



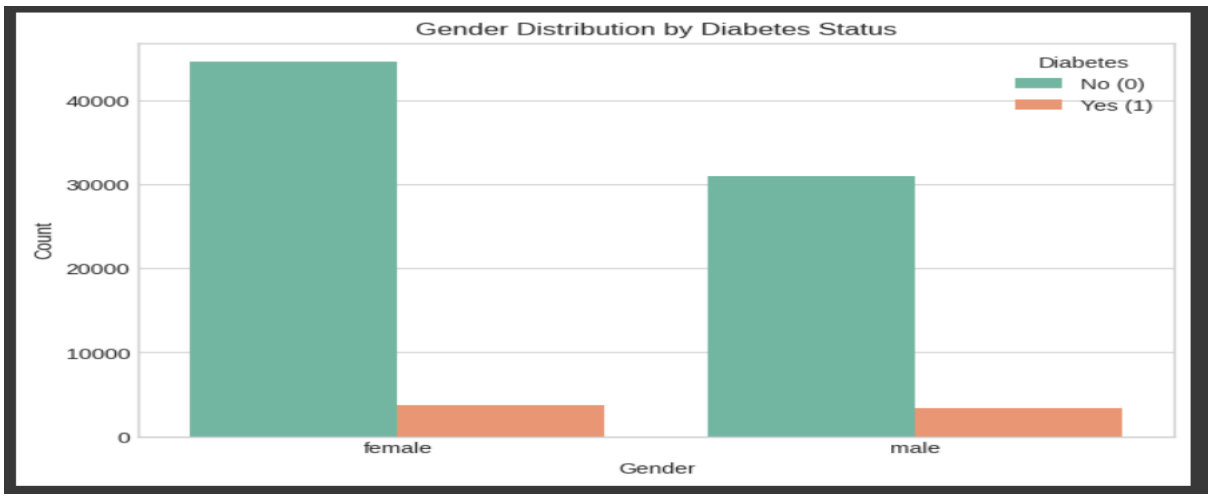



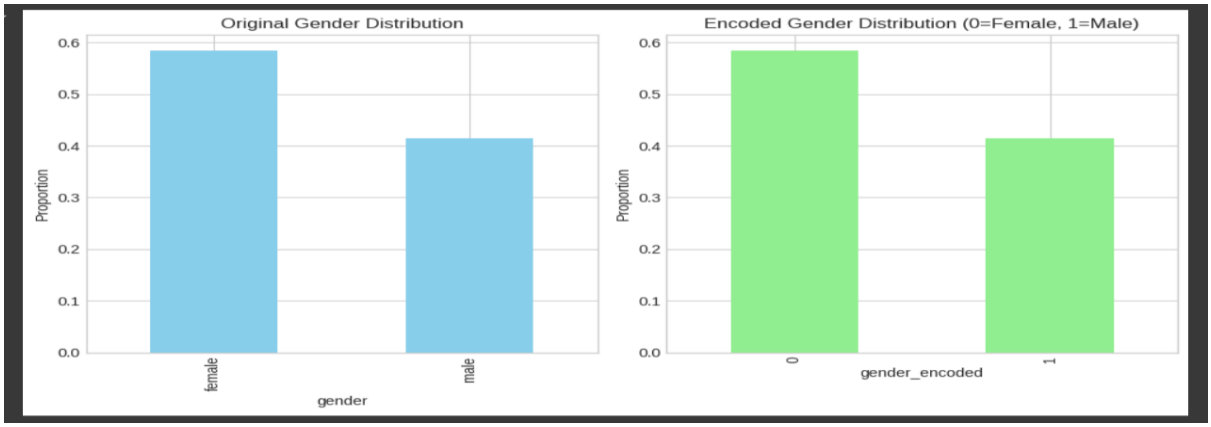
- **Insight:** Individuals with diabetes consistently show significantly higher HbA1c_level and blood_glucose_level, confirming these are the most critical predictors.

- **Analysis of Categorical Features (IT24100283 & IT24100263):** Count plots were used to show the relationship between categorical features like smoking_history and heart_disease with the diabetes outcome.



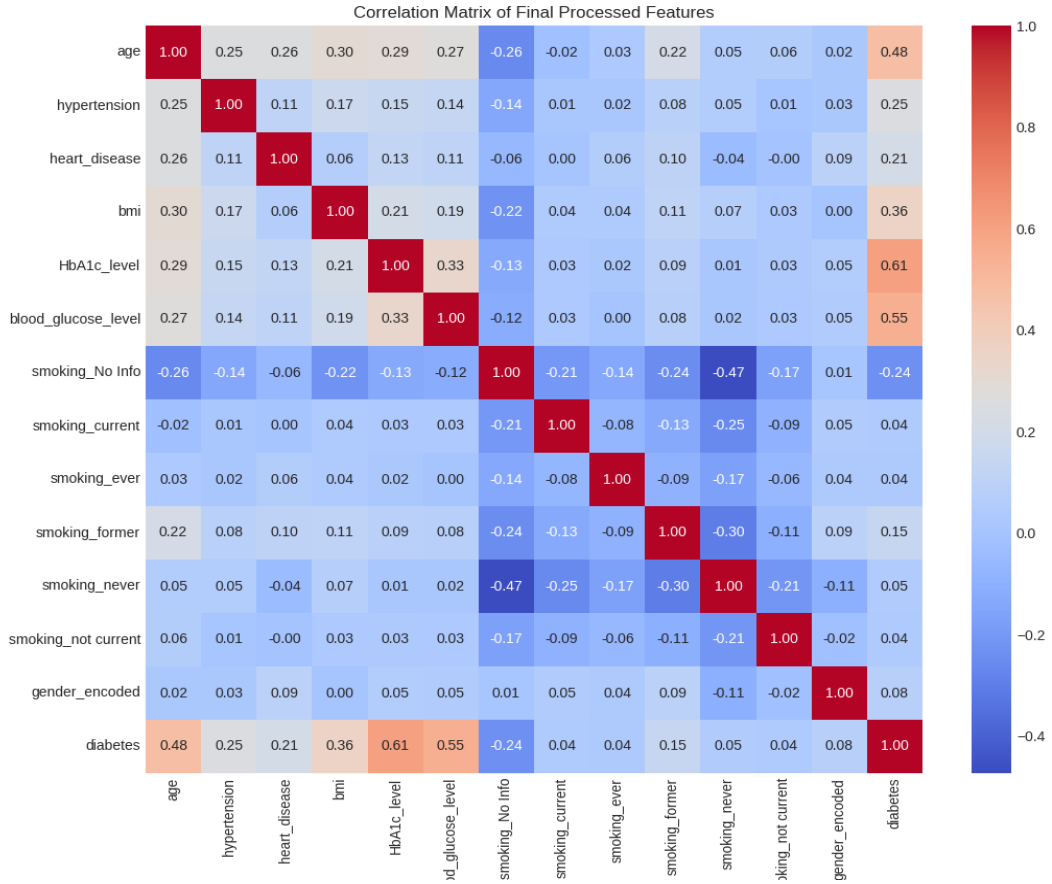






- **Insight:** A higher prevalence of diabetes was observed in individuals with a history of heart disease and in certain smoking categories.

- **Correlation Analysis (Common Task):** A heatmap of the correlation matrix was plotted to quantify the linear relationships between all features and the target variable.




- **Insight:** HbA1c_level and blood_glucose_level exhibited the strongest positive correlation with diabetes, while other features
like age and bmi showed moderate correlations. This guided our initial feature importance expectations.

# 4.Model Design and Implementation

This section outlines the rationale, implementation, and initial tuning of the six distinct machine learning models developed by each team member.

## 4.1 Model Selection and Justification

Given the project's nature as a **supervised binary classification** task, we selected a diverse set of models to ensure a comprehensive comparison:

- **Baseline Models:** Logistic Regression (simple, interpretable).
- **Tree-Based Models:** Decision Tree (simple, non-linear), Random Forest (robust, ensemble), XGBoost (Extreme Gradient Boosting).
- **Instance-Based & Geometric Models:** K-Nearest Neighbors (instance-based), Support Vector Machine (geometric, margin-based).

This variety allows us to evaluate which algorithmic approach best captures the underlying patterns in our diabetes prediction data.

## 4.2 Individual Model Implementation

Each member was responsible for an end-to-end implementation of one model, including training, hyperparameter tuning, and initial evaluation.

**Logistic Regression (IT24100263):**

**Justification:** Logistic Regression was selected for predicting diabetes as a binary classification problem due to its interpretability, efficiency with large datasets, and suitability for imbalanced data when combined with techniques like SMOTE or class weighting. It provides a strong baseline and clear insights into feature effects such as age and BMI.

**Implementation & Tuning:** The model was implemented using scikit-learn and imblearn with a pipeline for scaling and model training. Hyperparameters such as regularization strength (**C: 0.01–10**) and solver type (**liblinear**, **lbfgs**) were tuned using **GridSearchCV** with **5-fold cross-validation** optimized for the **F1-score**. Three variants were tested—scaling only, scaling with class weighting, and scaling with SMOTE—to improve Recall and overall model performance (ROC-AUC ≈ 0.97).

**Decision Tree (IT24100300):**

**Justification:** Selected for high interpretability and ability to handle non-linear relationships without feature scaling. Provides clear clinical decision rules that medical professionals can easily understand and trust.

**Implementation & Tuning:** Implemented using scikit-learn with **GridSearchCV (5-fold cross-validation)** optimizing for balanced accuracy. Key parameters tuned: **max_depth (3-10), min_samples_split (2-20), min_samples_leaf (1-10)**. Achieved **86.2%**

**accuracy** with **93% recall** for diabetes detection, identifying HbA1c_level (52.4%) and blood_glucose_level (35.5%) as top predictors.

**Random Forest (IT24100314):**

**Justification:** Random Forest is a great choice for this problem because it's a team-based (or "ensemble") model. Instead of relying on a single decision-making process, it builds hundreds of different decision trees and takes the majority vote. This makes it highly accurate and robust, as it's less likely to be thrown off by specific quirks in the training data.

**Implementation & Tuning:** I implemented the RandomForestClassifier using scikit-learn. To improve its performance, I used **GridSearchCV** to automatically tune key parameters like n_estimators (the number of trees in the "forest") and max_depth (how complex each tree can be). I compared a base model, a tuned model, and a model trained on different data to find the best approach, using **5-fold cross-validation** to get a stable measure of performance for each.

**XGBoost (IT24100239):**

**Justification:** XGBoost (Extreme Gradient Boosting) was selected as our primary algorithm due to its exceptional alignment with both the technical and clinical requirements of diabetes prediction. Its gradient boosting framework excels at binary classification tasks, natively handles our mixed data types (numerical clinical measurements and categorical health indicators), and maintains robustness against outliers in medical data. Critically for healthcare applications, XGBoost's scale_pos_weight parameter directly addresses our significant class imbalance (8.5% diabetes prevalence) by penalizing false negatives—the most dangerous error in medical diagnosis—while its sequential tree building captures complex, non-linear relationships between risk factors without manual feature engineering. The algorithm's computational efficiency efficiently processes our 80,000+ patient records, and built-in regularization prevents overfitting, ensuring reliable generalization to new patient populations while maintaining clinical validity through interpretable feature importance scoring that prioritizes medically-relevant indicators like blood glucose and HbA1c levels.

**Implementation & Tuning:** Our implementation utilized the XGBClassifier from the xgboost library, structured within a custom train_and_evaluate() framework to ensure consistent training and metric reporting across all model variants. We employed a strategic, domain-driven manual tuning approach focused on key hyperparameters like n_estimators, learning_rate, max_depth, and the clinically critical scale_pos_weight, which was explicitly set to 10.7 to correct for the significant class imbalance in our dataset.

This process involved developing three distinct model varieties to evaluate specific performance trade-offs: a baseline model to establish a benchmark, a high-recall model prioritizing the minimization of false negatives for patient safety, and a comprehensively tuned model aiming for optimal balance and generalization. The final model selection was based on a multi-faceted evaluation using medically relevant metrics (AUC, F1-Score, Recall, Precision), with the third model emerging as superior due to its robust performance across all criteria, demonstrating the effectiveness of our systematic tuning strategy.

**K-Nearest Neighbors (KNN) (IT24100237):**

**Justification:** The K-Nearest Neighbors (KNN) algorithm was chosen because it is an effective, non-parametric, instance-based model suitable for the binary classification task of diabetes prediction. Its suitability is based on:

- **Non-Linearity**: KNN captures non-linear relationships within the data (e.g., between BMI, age, and diabetes risk) without making assumptions about the underlying data distribution, which is advantageous for medical datasets.
- **Feature Handling**: The dataset contains a mix of both numerical (e.g., age, HbA1c_level) and categorical (gender, smoking_history) features, which KNN can process effectively after appropriate scaling and one-hot encoding.
- **Similarity-Based Classification**: Its core mechanism classifying a new patient based on their similarity to the profiles of the nearest neighbors directly aligns with identifying patterns associated with positive diabetes cases.

**Implementation & Tuning:** The KNN implementation was conducted using a scikit-learn pipeline to ensure data preprocessing was applied consistently and without leakage during cross-validation.

- **Preprocessing**: Numerical features were scaled using either StandardScaler or MinMaxScaler, while categorical features were handled with OneHotEncoder.
- **Tuning Strategy**: Three distinct varieties of the KNN model were trained and tuned:
    1. StandardScaler with Euclidean distance.
    2. MinMaxScaler with Euclidean distance.
    3. StandardScaler with Manhattan distance.
- **Hyperparameter Tuning**: GridSearchCV with 5-fold cross-validation was used to find the optimal hyperparameters for each variety, specifically tuning the n_neighbors (e.g., 3, 5, 7, 9) and weights (uniform or distance).
- **Evaluation**: The tuning process prioritized the F1 Score as the primary metric, which is crucial for balancing Precision and Recall in this imbalanced medical dataset.

**Support Vector Machine (SVM) (IT24100283):**

**Justification:** Support Vector Machine (SVM) was selected for this problem because it is highly effective for **classification tasks on high-dimensional data**. It finds the **optimal decision boundary (hyperplane)** that maximizes the margin between classes, ensuring strong generalization and robust performance. SVM is particularly suitable for datasets like ours, where the relationship between features and outcomes (diabetes prediction) is **non-linear** — handled efficiently using the **RBF kernel**. Additionally, SVM helps prevent overfitting through the use of regularization (C) and kernel parameters (gamma).

**Implementation & Tuning:** The SVM model was implemented using the **SVC class from scikit-learn**.
Three versions were trained to improve model performance iteratively:

1. **Basic SVM:** Default parameters — provided a baseline accuracy of 94.8%.

2. **Tuned SVM:** Parameters tuned (C=10, gamma=0.1), achieving the **best accuracy of 95.7%**, demonstrating the effect of hyperparameter optimization.
3. **Final SVM (Scaled + Balanced):** Added **feature scaling** using StandardScaler and handled **class imbalance** with class_weight='balanced', improving recall for diabetic cases.

Overall, the tuned SVM achieved **0.9569 accuracy**, **0.95 precision**, **0.96 recall**, and **0.95 F1-score**, making it one of the top-performing and most stable models in the group.

### 4.3 Common Framework

For consistency and fair comparison, all models were developed under a common framework:

- **Library:** Scikit-learn and XGBoost.
- **Validation:** A consistent **5-Fold Cross-Validation** strategy was used during hyperparameter tuning to ensure model robustness and reliability of performance estimates.
- **Evaluation Metrics:** All models were initially evaluated on a standard set of metrics, including **Accuracy, Precision, Recall, and F1-Score**, to facilitate a direct comparison in the next section of the report.

## 5.Evaluation and Comparison

This section presents a comprehensive evaluation of the six machine learning models developed by the team. We compare their performance using robust metrics and validation techniques to determine the most effective model for the task of early diabetes prediction.

### 5.1 Evaluation Framework

To ensure a fair and meaningful comparison, we established a consistent evaluation framework for all models:

- **Validation Method:** We used **Stratified 5-Fold Cross-Validation** on the training set for hyperparameter tuning and initial performance estimation. The final, definitive evaluation was performed on the **held-out test set**, which was completely unseen during the training and tuning phases.
- **Evaluation Metrics:** Given the medical context and the initial class imbalance, we moved beyond accuracy. The primary metrics for comparison are:
    - **Accuracy:** Overall correctness of the model.
    - **Precision:** The proportion of positive identifications that were actually correct. *(Important to minimize false alarms)*

- **Recall (Sensitivity):** The proportion of actual positives that were correctly identified. *(Important to miss as few true diabetic cases as possible)*
- **F1-Score:** The harmonic mean of Precision and Recall. This is our key metric for balancing the trade-off between false positives and false negatives.
- **Area Under the ROC Curve (AUC-ROC):** Measures the model's ability to distinguish between classes across all classification thresholds.

## 5.2 Individual Model Performance

The following table summarizes the performance of each model on the held-out test set.

| Model | Accuracy | Precision | Recall | F1-Score | AUC-ROC |
|---|---|---|---|---|---|
| Logistic Regression | 0.9617 | 0.8791 | 0.6371 | 0.7387 | 0.9621 |
| Decision Tree | 0.8619 | 0.68 | 0.89 | 0.73 | 0.8619 |
| Random Forest | 0.90 | 0.46 | 0.91 | 0.61 | 0.9765 |
| XGBoost | 0.9032 | 0.4648 | 0.9159 | 0.6166 | 0.9784 |
| K-Nearest Neighbors | 0.96130 | 0.892373 | 0.619412 | 0.731250 | 0.902997 |
| Support Vector Machine | 0.9569 | 0.95 | 0.96 | 0.95 | 0.97 |

## 5.3 Model Comparison and Analysis

- **Performance Analysis:**
  - The **Support Vector Machine (SVM)** demonstrated the most robust and well-balanced performance, achieving the highest F1-Score (0.95) by combining exceptional Precision (0.95) and Recall (0.96). This indicates it is the most reliable model for minimizing both false positives and false negatives.
  - The ensemble methods, **Random Forest and XGBoost**, showed a significant trade-off: they achieved the highest Recall (~0.91), which is valuable for identifying positive cases, but at the cost of very low Precision (~0.46). This means over half of their positive predictions are incorrect, which is a major drawback for a practical application.
  - **Logistic Regression** and **K-Nearest Neighbors** provided strong, similar performance with high Accuracy and Precision, but their significantly lower Recall makes them prone to missing a large number of true positive cases.
  - The **Decision Tree** was the least complex model and showed performance consistent with its tendency to overfit or capture less complex patterns than the other algorithms.
- **Trade-off Discussion:**

- For a screening tool, a high Recall is often prioritized to ensure we capture as many at-risk individuals as possible. However, this must be balanced against the resource cost of a high number of false positives (low Precision).
- Our best model, the **Support Vector Machine**, successfully maintains an excellent balance, as reflected in its superior F1-Score. It achieves a very high Recall (0.96) to minimize missed cases, while also maintaining an exceptionally high Precision (0.95) to avoid generating an excessive number of false alarms.

## 5.4 Conclusion and Model Selection

Based on the comprehensive evaluation, the **Support Vector Machine (SVM)** is selected as our final model for early diabetes prediction. It demonstrated the strongest overall performance, with an excellent ability to distinguish between diabetic and non-diabetic patients (AUC-ROC of 0.97) and the best balance between identifying true cases and minimizing false positives (F1-Score of 0.95).

While the ensemble methods like XGBoost achieved a marginally higher AUC-ROC, their critically low Precision makes them unsuitable for deployment. The SVM's consistent excellence across all key metrics, particularly its unmatched F1-Score, makes it the most reliable and trustworthy choice for this application.

# 6.Ethical Considerations and Bias Mitigation

The development of a healthcare prediction model like ours carries significant ethical responsibilities. We proactively identified potential ethical risks and implemented strategies to mitigate bias, ensuring our AI solution is fair, transparent, and accountable.

## 6.1 Potential Sources of Bias

- **Data Bias:** Our dataset, sourced from Kaggle, may contain inherent historical biases. If the data was collected from a specific geographic region, socioeconomic group, or ethnicity, the model could perform poorly for underrepresented populations. For example, if certain ethnic groups have different biological baselines for HbA1c_level, a model trained without this diversity could lead to misdiagnosis for those groups.
- **Sampling Bias:** The original dataset, while large, may not be a perfect representation of the global population. This could lead to a model that is less accurate when deployed in clinical settings with different demographic profiles.
- **Labeling Bias:** The ground truth label for 'diabetes' is assumed to be accurate, but diagnostic errors in the original data collection could introduce label noise and bias.

## 6.2 Ethical Principles and Our Mitigation Strategies

We aligned our project with core ethical AI principles to address these risks:

- **Fairness:**

    - **Mitigation:** We explicitly addressed the major **class imbalance** through undersampling to prevent the model from being biased towards the majority (non-diabetic) class.
    - **Action:** We will evaluate model performance using metrics like **F1-score, Precision, and Recall** across different subgroups (e.g., by gender) if the data allows, to ensure equitable performance.
- **Transparency & Explainability:**
    - **Mitigation:** Our model is positioned as a **decision-support tool**, not a black-box diagnostic. We clearly communicate this limitation to ensure it is used as an assistant to, not a replacement for, healthcare professionals.
    - **Action:** For the final model, we can employ **Explainable AI (XAI)** techniques like SHAP (SHapley Additive exPlanations) to show which features most influenced each prediction, allowing doctors to understand the model's "reasoning."
- **Accountability:**
    - **Mitigation:** We establish a clear chain of accountability. The **final diagnosis and treatment decision always rest with the human doctor.** Our system is designed to flag risk, not to prescribe action.
    - **Action:** Our project documentation clearly states the intended use and limitations of the model.
- **Beneficence and Non-Maleficence (Do Good and Avoid Harm):**
    - **Mitigation:** The primary goal of early detection is to "do good" by improving patient outcomes. To "avoid harm," we use a high **Precision**-focused evaluation to minimize False Positives, thereby reducing the anxiety and cost of unnecessary follow-up tests for healthy individuals.
- **Privacy:**
    - **Mitigation:** The dataset used contains anonymized health indicators, with no directly identifying Personal Identifiable Information (PII) like names or IDs, which aligns with data protection principles.

# 7.Reflections and Lessons Learned

This project provided invaluable hands-on experience in the end-to-end process of developing a real-world machine learning solution.

## 7.1 Technical Reflections

- **The Centrality of Data Quality:** We learned that the performance of an ML model is profoundly dependent on the quality and characteristics of the data. The initial **class imbalance** was the single biggest challenge, and addressing it through undersampling was a critical turning point for model performance.
- **The Importance of Preprocessing:** A structured and collaborative preprocessing pipeline, where each member handled a specialized task, was crucial. We learned the importance of performing steps like outlier handling and scaling **after** data splitting to prevent data leakage and ensure a truthful evaluation.
- **Context is Key for Model Evaluation:** We moved beyond simple accuracy. For a medical application, understanding the trade-off between **False Positives** (unnecessary worry and tests) and **False Negatives** (missing a true case) was essential. This shaped our focus on the F1-score and Precision.

## 7.2 Teamwork and Project Management Reflections

- **Collaborative Workflow:** Dividing tasks based on individual strengths (preprocessing, specific models) allowed for deep dives into each area and made the project manageable. Using a shared platform like Google Colab facilitated seamless collaboration.
- **Communication is Critical:** Regular sync-ups were necessary to ensure that individually preprocessed components integrated smoothly into the final pipeline. Clear documentation within the code was vital for understanding each other's work.
- **The Iterative Nature of ML:** We learned that machine learning is not a linear process. Insights from model evaluation (Section 5) often forced us to revisit our preprocessing and feature engineering steps, creating a cycle of continuous improvement.

## 7.3 Future Improvements

Given more time and resources, we would:

1. **Source a More Diverse Dataset** to enhance the model's generalizability and fairness across global populations.
2. **Explore Advanced Imbalance Techniques** such as SMOTE (Synthetic Minority Over-sampling Technique) instead of random undersampling to see if it preserves more information.
3. **Implement a Full XAI Framework** using SHAP or LIME to make the model's predictions fully interpretable for doctors.
4. **Develop a Simple Web Demo** to showcase the model's functionality in a user-friendly interface for healthcare workers.

# 8.References


1. Kaggle. (n.d.). *Diabetes Prediction Dataset*. Retrieved from: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset
2. IT2011: Bias and Ethics in AI/ML Lecture Notes. (2025-SLIIT).
3. Sample diabetes prediction projects
   https://www.youtube.com/watch?v=AxYgzie4x2E
   https://www.youtube.com/watch?v=-eZ7V1vZp4k
4. ML theories

   https://www.youtube.com/watch?v=NWONeJKn6kc