Predicting diabetes outcome for women

Andrei ROUMIANTSEV, March 2025

1 Introduction

The project focuses on **predicting diabetes outcomes** for women using a **machine learning pipeline**.

Designed, built, and deployed by DSTI students, this pipeline is developed as part of the Group Project for the Machine Learning with Python Labs course, 2025.

Students:
Mahamadou Ousmane Keita, Rosine Muna, Malathi Karuppasamy, Jean Daniel Houssou, Arthur Choisnet, Andrei Roumiantsev

Instructor: Christophe Bécavin, PhD, Université Côte d'Azur

Professor: Hanna Abi Akl, DSTI

Project dates: start December 2024 – end March 2025

Andrei Roumiantsev project GitHub link:
https://github.com/AndreiRRR/DSTI-Project-Diabeties-Prediction-In-Women/

The dataset provided contains information about **15,000 women** aged 20 to 80 who visited the **Taipei Municipal Medical Center** between 2018 and 2022. Each record includes **eight features**—such as **pregnancies**, **plasma glucose concentration**, and **body mass index (BMI)**—and a target variable indicating whether the individual was diagnosed with diabetes (**1 for diagnosed**, **0 for not diagnosed**).

The aim is to build a model that **predicts the likelihood of diabetes** based on these features. The task involves constructing and deploying a **complete machine learning pipeline**, from **exploratory data analysis** and **feature engineering** to **model selection** and **evaluation**. The ultimate goal is to integrate the model into a **web application** for user-friendly predictions.

2 Methodology

This section outlines the step-by-step methodology employed in this project to predict diabetes outcomes using machine learning. The workflow follows an end-to-end pipeline approach, covering data preparation, model building, and deployment.

The primary goal of this project is to develop the best possible prediction model. Every step in our methodology is designed to maximize the model's performance by analyzing and refining the data, ensuring the optimal choice of features, and selecting the most effective algorithms.

As a project rooted in the field of data science, we will follow the steps outlined below and detail the methodology in their respective subsections.

- **Exploratory Data Analysis :** In this step, we aim to familiarize ourselves with the dataset, identify notable characteristics, detect outliers, and handle missing values (*NaN* values). We will also clean the data as necessary to prepare it for subsequent analysis, ensuring a strong foundation for the project.
- **Feature Engineering and Selection :** After understanding the dataset, this step focuses on refining the data by removing unnecessary or redundant features. We will gain deeper insights into relationships between features and identify the most significant ones that contribute to accurate predictions.
- **Model Selection, Comparison, and Evaluation :** Here, we concentrate on evaluating the performance of different models and exploring ways to improve their effectiveness. This involves testing various algorithms, optimizing hyperparameters, and comparing metrics to select the best-performing model.

It is important to note that these steps are interconnected. For instance, additional exploratory data analysis may be conducted during the model selection and evaluation phase if required. This iterative approach ensures that we can refine each component of the pipeline to achieve the best results.

2.1 Exploratory Data Analysis

2.1.1 Dataset Overview

The dataset TAIPEI_diabetes.csv comprises health records of 15,000 female patients aged 20-80 who visited Taipei Municipal Medical Center between 2018-2022. This data set was collected for diabetes prediction research and contains 8 clinical characteristics plus a binary target variable. You can find the meaning of the variables in the following table.

Table 1 – Feature Description

Feature	Type	Description
PatientID	Integer	Unique ID to the patient
Pregnancies	Integer	Number of pregnancies
PlasmaGlucose	Integer	Plasma glucose concentration
DiastolicBP	Integer	Diastolic blood pressure
TricepsThickness	Integer	Skinfold thickness
SerumInsulin	Integer	2-Hour serum insulin
BMI	Float	Body mass index
DiabetesPedigree	Float	Diabetes likelihood
Age	Integer	Patient age
Diabetic	Binary	Diabetes diagnosis

2.1.2 Data Quality Assessment

Before proceeding with any analysis, we conducted a thorough quality assessment of our dataset, focusing on three key aspects : duplicate values, missing values, and outliers.

Handling Duplicate Values

The dataset contains a PatientID field which serves as a unique identifier for each patient. Our analysis revealed **210 duplicate PatientIDs** corresponding to **105 patients** with multiple entries. This indicates that some patients were registered multiple times in the system.

Before deciding whether to remove or retain these duplicates, we performed a comprehensive evaluation of their potential value. Our investigation showed that for the same PatientID, most recorded information varies significantly

This pattern suggests that these duplicate entries represent separate medical visits rather than data entry errors. Since each visit provides additional clinical information relevant to our diabetes prediction task, we made the deliberate decision to retain all entries in our dataset.

Handling Missing Values

The heatmap of the Figure 1 shows that the dataset contains no missing values, as evidenced by the absence of color variation in the visualization. We can therefore proceed with our analysis without needing to handle missing data.

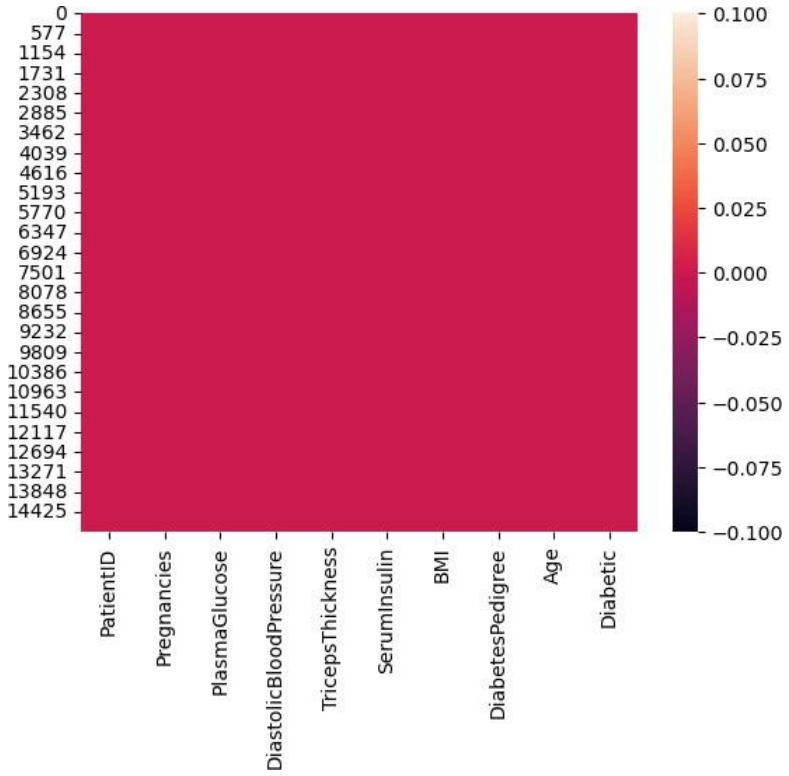

FIGURE 1 – Missing Values HeatMap

Handling Outliers Values

The boxplot visualization in Figure 2 reveals four variables containing significant outliers. We must now determine whether to retain or remove these anomalous data points based on their potential impact on our analysis.

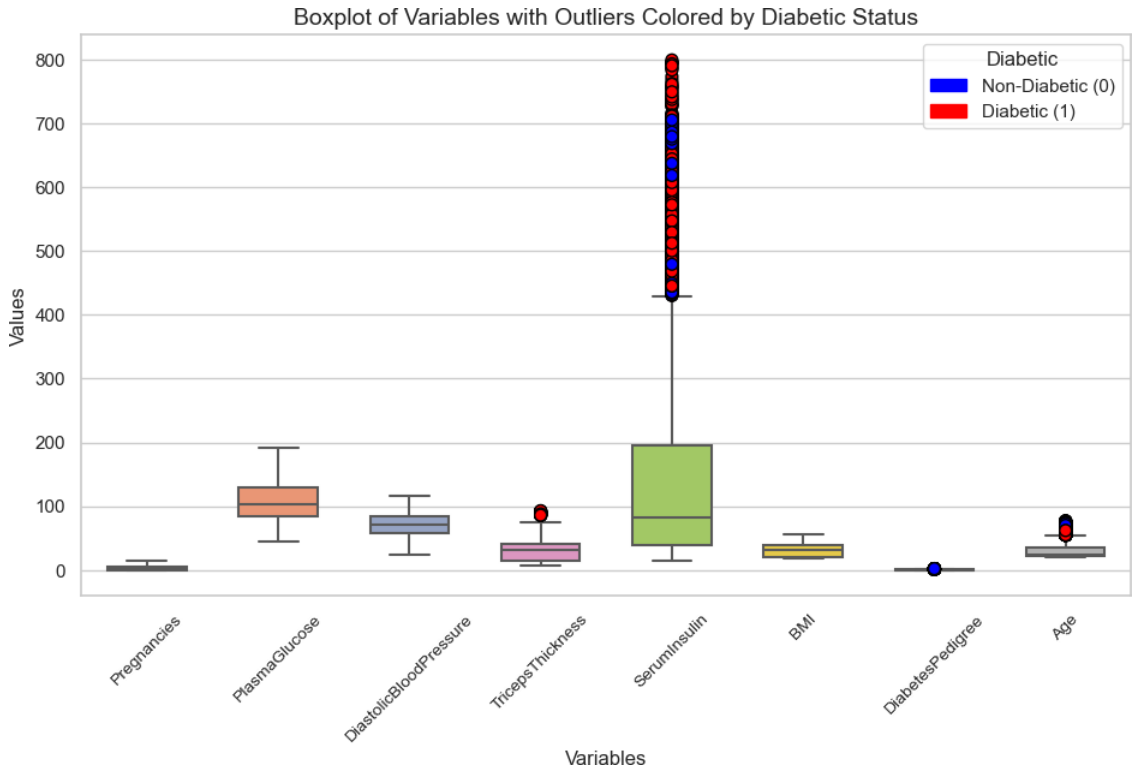

FIGURE 2 – Distribution des données diabétiques dans notre étude

When handling outliers in a classification context, we must verify that their removal does not disproportionately affect class balance. Figure 3 demonstrates the distribution of outliers across target classes, revealing whether certain classes are more impacted by extreme values.

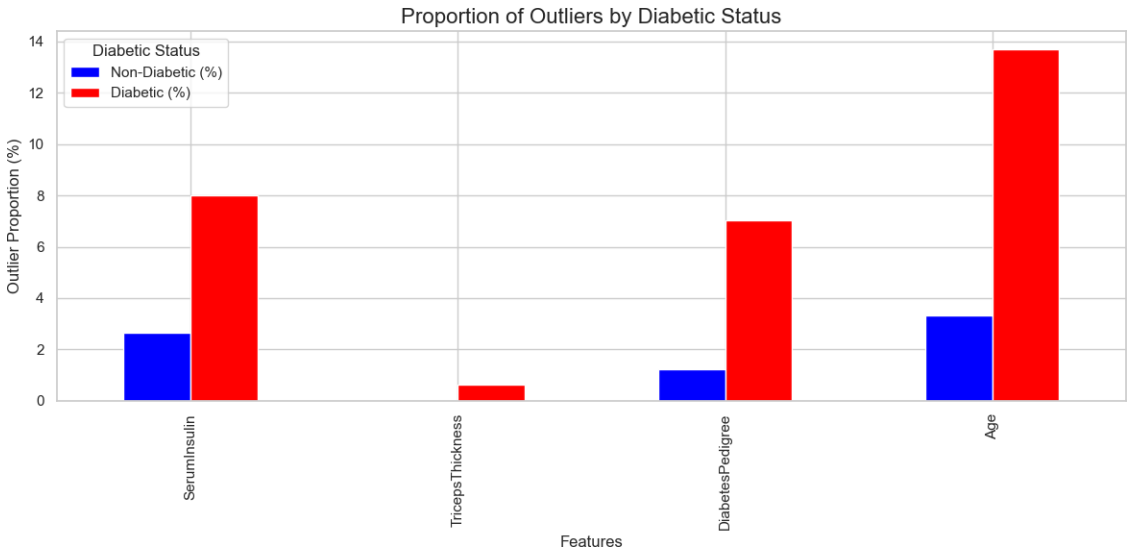

FIGURE 3 – Proportion of diabetic cases among outlier values (Class 0 : Non-Diabetic, Class 1 : Diabetic)

The barplot confirms that outliers are nearly evenly distributed between both classes, suggesting their removal would not significantly alter class proportions. This observation supports retaining these values, as they may contain meaningful clinical information rather than measurement errors.

2.2 Feature Engineering and Selection

With our dataset now cleaned, we conduct exploratory analysis to uncover patterns that will inform model selection and potential improvements.

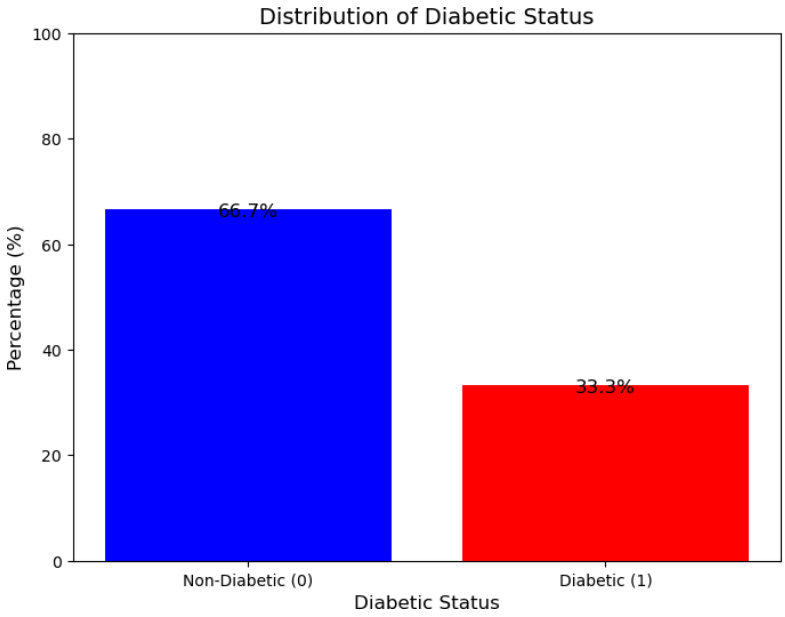

FIGURE 4 – Class distribution of diabetic cases (0 : Non-Diabetic, 1 : Diabetic)

Figure 4 reveals a significant class imbalance in our target variable. This imbalance suggests that certain models may develop prediction bias toward the majority class (non-diabetic cases).

Feature Significance Analysis

To identify which variables significantly differentiate diabetic cases, we performed Mann-Whitney U tests (non-parametric equivalent of t-test for non-normal distributions).

The results show that all examined variables show statistically significant differences ($p < 0.001$) between diabetic and non-diabetic groups. Notably, SerumInsulin shows the most extreme significance ($p = 3.98 \times 10^{-308}$), suggesting its particular importance for diabetes prediction.

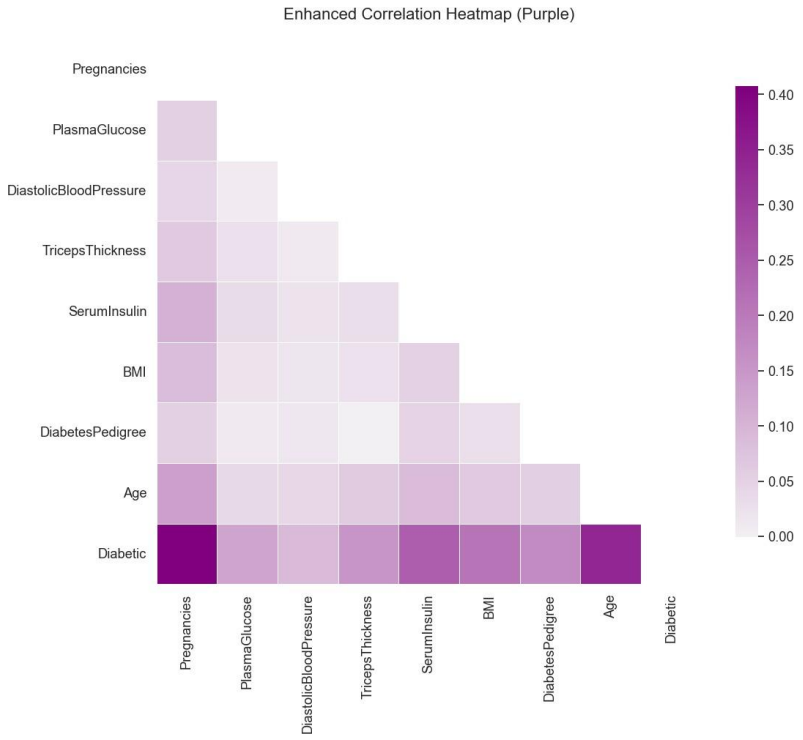

FIGURE 5 – Correlation matrix of predictor variables

The correlation matrix in Figure 5 shows :

— Low inter-feature correlations (all $|r| < 0.3$), indicating minimal multicollinearity

— Moderate target correlations (Glucose : 0.47, BMI : 0.31)

We have visualized the uneven predictive power across variables, corroborating our statistical test results. The combination of :

— Mann-Whitney U tests (statistical significance)

— Correlation analysis (linear relationships)

— Effect size visualization

provides comprehensive evidence for variable selection in our predictive models, and clearly, the have show to keep them all.

3 Model Selection, Comparison and Evaluation

3.1 Model Choose

Since we are dealing with a classification problem, we have selected the following models : Generalized Linear Models (GLM), Decision Trees, Random Forests, and XGBoost. The latter is particularly renowned in the data science community for its outstanding performance in machine learning competitions and should be our leader.

3.2 Cross-Validation Design

We implement **stratified 10-fold cross-validation** because :

— **Statistical robustness** :
  — 10 iterations balance variance and bias
  — Each test set contains 1,500 observations (10% of data)
— **Computational efficiency** :
  — More efficient than LOO (15,000 fits)
  — More stable than 5-fold (larger test sets)
— **Class imbalance preservation** :
  — Maintains original 66.7%/33.3% class ratio in all folds with the stratification method

3.3 Implementation Details

The validation process follows these steps :

1. Stratified data partitioning (10 folds)
2. Sequential training on 9 folds (13,500 obs)
3. Evaluation on holdout fold (1,500 obs)
4. Performance metric aggregation across folds

3.4 Evaluation Methodology

3.4.1 Metrics Framework

We employ a multi-metric assessment :

Table 2 – Evaluation Metrics

Metric	Purpose
Accuracy	Overall performance
AUC-ROC	Class separation quality
Confusion Matrix	Error type analysis

3.4.2 Initial Model Evaluation

All models must surpass the naive baseline accuracy of 66.7% (obtained by always predicting the majority class) to demonstrate their relevance over a simple heuristic. Subsequently, we will select the best-performing model.

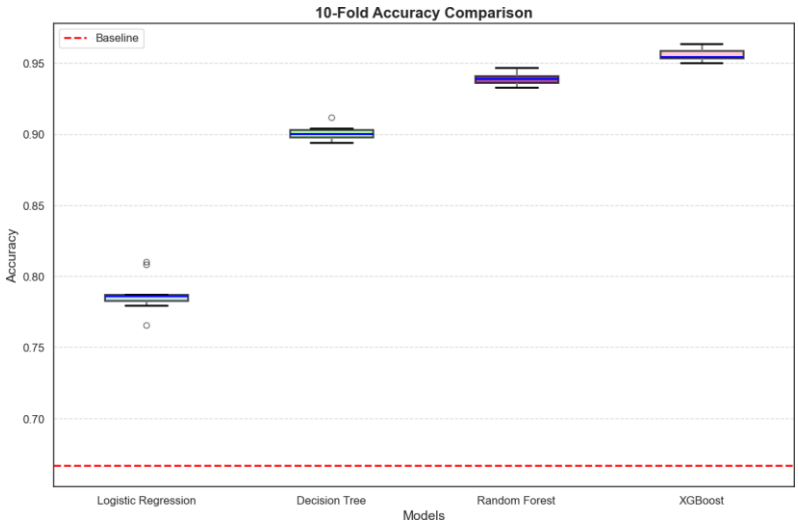

FIGURE 6 – Performance Comparison of Different Models

As shown in the boxplot in Figure 6, we can directly proceed with XGBoost, which outperforms the other methods as expected. While we could optimize the hyperparameters of all methods for a more thorough comparison, we focus on XGBoost because :

- It has demonstrated superior effectiveness in numerous cases
- It incorporates built-in tools like parameter regularization
- It offers advantages for feature engineering, which may be useful for future improvements

3.5 Outlier Treatment Analysis

To evaluate the impact of outliers on model performance, we implemented the following strategy :

- Created four new binary indicator features corresponding to the four variables containing outliers
- For each feature :
  - Assigned value 0 for observations within normal ranges
  - Preserved the original value for outlier observations
- Maintained all original features in the model

3.5.1 Results

The inclusion of these engineered outlier features showed a **slight impact (about 0.2%** on the XGBoost model's performance metrics. This suggests that :

- The XGBoost algorithm is inherently robust to the presence of outliers in our dataset
- The tree-based splitting mechanism effectively handles extreme values without requiring special treatment

We have decided to keep the outliers even if the gain is not huge because here, we are in competition with other teams on the same project and they do not add a lot in computational cost.

3.6 Model Optimization : XGBoost Hyperparameter Tuning

3.6.1 Motivation

Our systematic hyperparameter tuning serves three key objectives :

1. **Performance Maximization** XGBoost's effectiveness is highly dependent on proper parameter configuration, as default values rarely yield optimal results for specific datasets.
2. **Overfitting Prevention** Through careful regularization tuning (α, λ, and γ parameters), we maintain the model's generalization capability while capturing meaningful patterns.
3. **Computational Efficiency** We optimize the trade-off between *n_estimators* and *learning_rate* to achieve maximum accuracy with minimal computational overhead.

3.6.2 Implementation Strategy

The hyperparameter optimization focuses on four key aspects :

— **Tree Structure** (*max_depth*, *min_child_weight*) : Controls model complexity and prevents overfitting
— **Learning Process** (*learning_rate*, *n_estimators*) : Manages gradient descent convergence and computational efficiency
— **Robustness** (*subsample*, *colsample_bytree*) : Introduces stochasticity for better generalization
— **Class Imbalance** (*scale_pos_weight*) : Compensates for unequal class distribution, particularly relevant for our dataset

3.6.3 Optimization Method

We employ **Optuna**, a state-of-the-art hyperparameter optimization framework, which provides :

— **Bayesian optimization** using Tree-structured Parzen Estimator (TPE) algorithm
— **Automated trial pruning** of unpromising configurations
— **Distributed optimization** capabilities for efficient resource utilization
— **Visual analytics** for parameter importance analysis and optimization tracking

Contrary to our initial expectations, the engineered outlier features did not yield any measurable improvement in model accuracy. This null result led us to conclude that :

— The XGBoost model demonstrates inherent robustness to outliers in our dataset
— The additional complexity introduced by these features provided no predictive benefit
— The baseline feature representation already contained sufficient information for the model to handle edge cases effectively

Consequently, we removed these engineered features from our final model to maintain simplicity and computational efficiency.

3.6.4 Optimal Classification Threshold Analysis

In binary classification, while 0.5 serves as the conventional decision threshold, we investigated whether alternative thresholds might improve model performance :

— Computed mean optimal thresholds across **10-fold** cross-validation
— Identified **0.34** as a potential candidate threshold through ROC analysis

— Implemented comparative testing with both threshold values

The results demonstrated negligible differences in performance metrics, leading us to maintain the default 0.5 threshold for final model deployment.

4 Conclusion

This project developed a high-performance **XGBoost model** for diabetes prediction in women, achieving **95.6% accuracy** through rigorous feature analysis and hyperparameter tuning. Key findings include :

— **All selected variables** were significant predictors, contributing to model performance.
— **XGBoost outperformed alternative algorithms**, demonstrating robustness in handling class imbalance.
— **Medical outliers**, when retained and properly treated, enhanced predictive accuracy.

Deployment for Healthcare Accessibility

A user-friendly **web application** was deployed to facilitate real-time diabetes risk assessment, simulating healthcare decision-making :
Diabetes Prediction App

Open-Source Contribution

The complete **source code**, documentation, and methodology are publicly available for reproducibility and further research :
GitHub Repository

This **end-to-end solution** provides a practical framework for developing and deploying machine learning models in medical diagnostics, balancing interpretability and predictive power.