Project 1 - Neural Networks
PIMA Indian Diabetes Prediction

School of Information, Computer, and Communication Technology (ICT), Sirindhorn International Institute of Technology (SIIT), Thammasat University, Khlong Luang, Pathum Thani, Thailand

---

0. Team Contributions

This research project was a collaborative effort involving four dedicated team members, each contributing their unique skills and expertise towards achieving the project's objectives. The team members and their respective contributions are as follows:

1.1. Ms. Apinya Sriyota 6322771534

- Led the planning and set the direction for the project.
- Managed the data: from initial analysis, choosing features, fixing outliers, to preparing data for modeling.
- Handled the complexities of missing data, balanced the dataset, and divided data for training and testing.
- Built and fine-tuned the XGBoost model, ensuring it's up to standard.

1.2. Mr. Thanakit Mettarikanon 6322772045

- Collaborated on the early stages of the ANN model, ensuring it was on the right track.
- Led the development of both the RandomForest and Logistic Regression models.
- Offering insights and recommendations, helping refine our overall approach.

1.3. Ms. Isariya Kerdkla 6322773092

- Took charge of building and refining the Artificial Neural Network (ANN) model.
- Wrote the review of Part 1 of our report, laying a strong foundation.
- Reviewed other models, giving feedback and suggesting improvements.

1.4. Mr. Chanawong Karoon-ngampun 6322774025

- Worked with **Mr. Thanakit** to start the ANN model, setting a strong starting point.
- Built and fine-tuned the Support Vector Machine (SVM) model.
- Supported the team by solving challenges and providing expertise when needed.

1. Review Part 1

In the result of the loss graph, the loss values are considerably higher when compared to the validation dataset. As the number of epochs increases, both the test and validation loss trend to stabilize within a range roughly between 0.6 and 0.75. This suggests that the model may not be achieving a perfect fit to the training data but is generalizing reasonably well to validation data. Regarding the accuracy values, both the test and training datasets exhibit similar trends.Initially, there’s a significant spike in accuracy during the early epochs, and this is consistent up to around epoch 50. Subsequently, the accuracy figures tend to fluctuate within range from 0.7 to 0.8, starting from approximately epoch 60 and continuing up to 150. The accuracy result of about 76.62% on the test data should be taken with caution because it might not be a completely accurate reflection of how well the model can perform on new, unseen data. This is because the code uses the same data for both training and testing, with the testing part being a subset of the training data. Imagine if you had already seen the test questions before the exam; you'd likely do well, but it wouldn't show how well you can handle new questions. The loss and accuracy numbers help track how the model learns during training, but to truly evaluate its effectiveness, we need a separate set of new questions (test data) that the model has never seen before. So, it's important to use an entirely independent test dataset for a more accurate assessment.

2. Introduction and Data Analysis

Diabetes mellitus, a chronic metabolic disorder, has escalated to a global epidemic, impacting countless lives and presenting formidable healthcare challenges. Predictive modeling, particularly in the context of early diagnosis and intervention, offers the potential to significantly improve patient outcomes. Focusing on the PIMA Indian demographic, which exhibits a heightened susceptibility to the disease, this study aims to identify the most effective predictive model for diabetes prediction. The models under consideration, presented in sequence, are logistic regression, support vector machine (SVM), random forest, XGBoost, and an Artificial Neural Network (ANN). Through meticulous evaluation of these models, informed by data and insights sourced from reputable medical websites, our aim is to augment the domain of diabetes prediction and bolster our comprehensive understanding of the disease's management and prevention.

a. Column Analysis

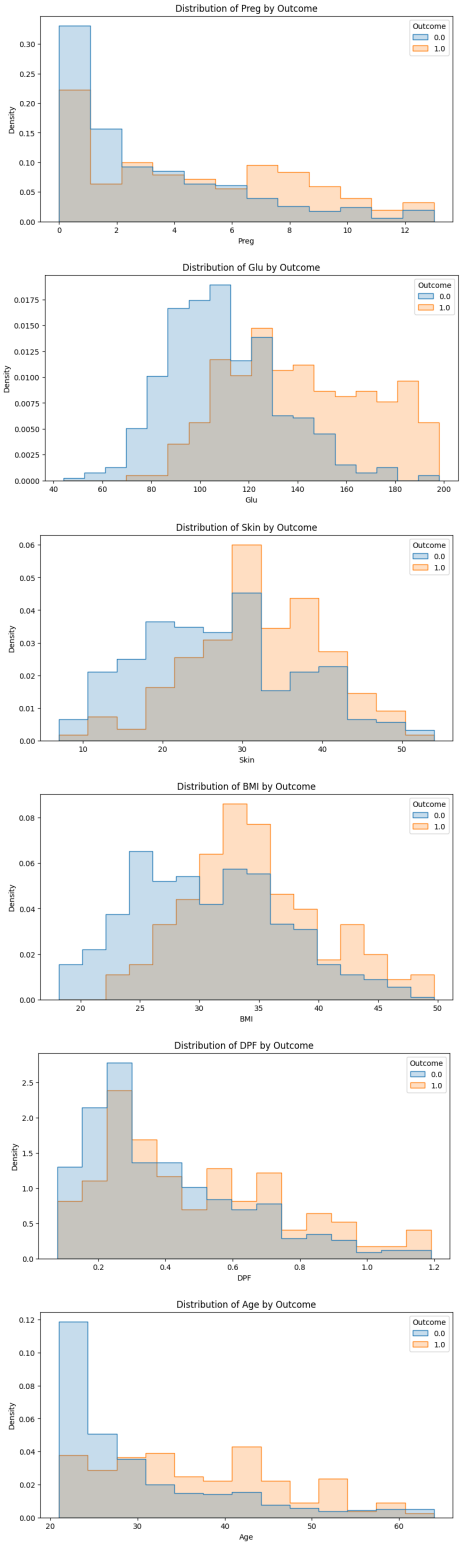

Fig. 1. The class distributions of each feature.

The dataset used in this work is the PIMA Indians Diabetes Dataset, sourced from the National Institute of Diabetes and Digestive and Kidney Diseases, consists of 768 instances, with 500 testing negative (class 0) and 268 testing positive (class 1) for diabetes.

Originating from a study on Pima Indian women over 21 years of age from Phoenix, Arizona, the dataset has eight key features: number of pregnancies, 2-hour plasma glucose concentration, diastolic blood pressure, triceps skinfold thickness, 2-hour serum insulin, body mass index (BMI), a diabetes pedigree function (indicating likelihood based on family history), and age. These metrics serve to predict the onset of diabetes, with specific challenges in the dataset including instances with biologically implausible zero values.

After imputing biologically implausible zero values in the PIMA Indians Diabetes Dataset, the data availability for each column is delineated as follows: 'Preg' (Number of times pregnant) has 768 entries; 'Glu' (Plasma glucose concentration) consists of 763 values; 'BP' (Diastolic blood pressure) comprises 733 records; 'Skin' (Triceps skinfold thickness) holds 541 entries; 'Ins' (2-Hour serum insulin) with only 394 records, is a candidate for removal due to its significant missing data. 'BMI' (Body mass index) has 757 values; 'DPF' (Diabetes pedigree function, Likelihood of diabetes based on family history) contains all 768 entries, and 'Age' (Age in years) is consistent with 768 records.

b. Multicollinearity Detection

Multicollinearity, where predictors closely interrelate, can hinder model interpretability and generalization in binary classification. To diagnose this, we utilized pair plots for initial visual insights on variable relationships and correlation matrices with heatmaps for quantification. Together, these techniques aid in pinpointing and addressing multicollinearity, optimizing our classification model's performance and clarity.

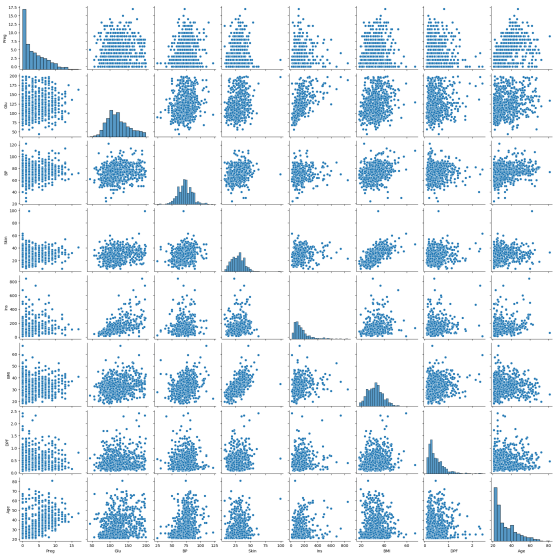

Figure 2. Pair plot displaying pairwise relationships among features.

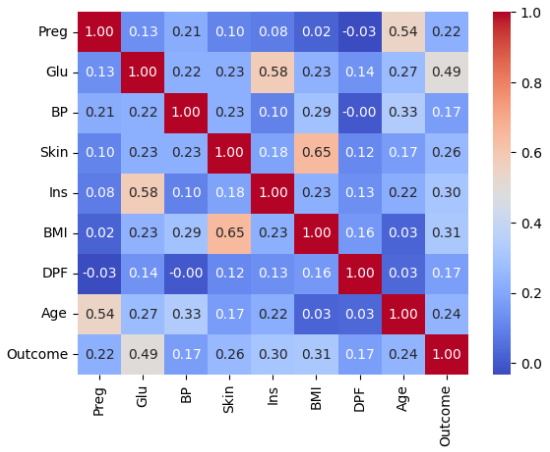

Figure 3. Heatmap of correlation matrix, visualizing the strength and direction of relationships among features.

After examining both the pair plot and the correlation heatmap, we observed that none of the predictors exhibit correlation values exceeding 0.7. In statistical analyses, a correlation coefficient above 0.7 is often regarded as indicating a strong correlation between variables. The absence of such strong correlations among our predictors suggests that multicollinearity is not a concern in this dataset.

3. Algorithm Design

Preprocessing the PIMA Diabetes dataset presents formidable challenges due to missing data and refusals from respondents, which can introduce biases. To mitigate these issues and enhance the efficacy of our algorithms, a multifaceted approach is necessary. This includes feature selection to identify the most informative predictors, feature engineering to create new relevant features, outlier detection and treatment for improved data integrity, imputation techniques to handle missing values, and strategies to address the class imbalance. By rigorously addressing these preprocessing challenges, our algorithm design becomes more resilient and capable of producing meaningful and unbiased results for binary classification.

a. Feature Selection

After a thorough column analysis, it became evident that the 'Ins' column (2-Hour serum insulin) was riddled with too many missing values, accounting for a significant 48.7% absence. Given the extensive gaps in this data and the challenges it would pose for imputation and model reliability, the decision was made to exclude the 'Ins' column from the dataset.

b. Feature Engineering

The refinement of our PIMA Diabetes dataset entailed focused attention on the 'Glu' (Glucose) and 'BMI' columns. Guided by WHO and ADA **[1]** criteria, we encoded 'Glu' () values into 'normal' (values < 140), 'pre-diabetes' (values between 140 and 199), and 'diabetes' (values ≥ 200). Simultaneously, leveraging insights from the "Obesity and Type 2 Diabetes" **[2]** research, BMI values were marked as 'obesity-related' for readings ≥ 27.5 kg/m^2, a threshold indicating obesity risks for the Asian population. This feature engineering process resulted in three datasets for our analysis: the original, one with encoded BMI and Glucose values, and another with scaled BMI and Glucose values, ensuring we can assess the impact of these modifications on model performance.

c. Outlier Detection

Outliers can significantly skew data analysis, underscoring the need for accurate detection mechanisms. We employ box plots for their visual clarity in representing data distribution, where points outside the whiskers (typically 1.5 times the Interquartile Range or IQR) are considered outliers. Complementing this, the IQR, representing the spread between the upper and lower quartiles, provides a quantitative method to precisely identify outliers. Together, the box plot and IQR ensure a comprehensive approach to preserving data integrity by effectively pinpointing and addressing anomalous values.

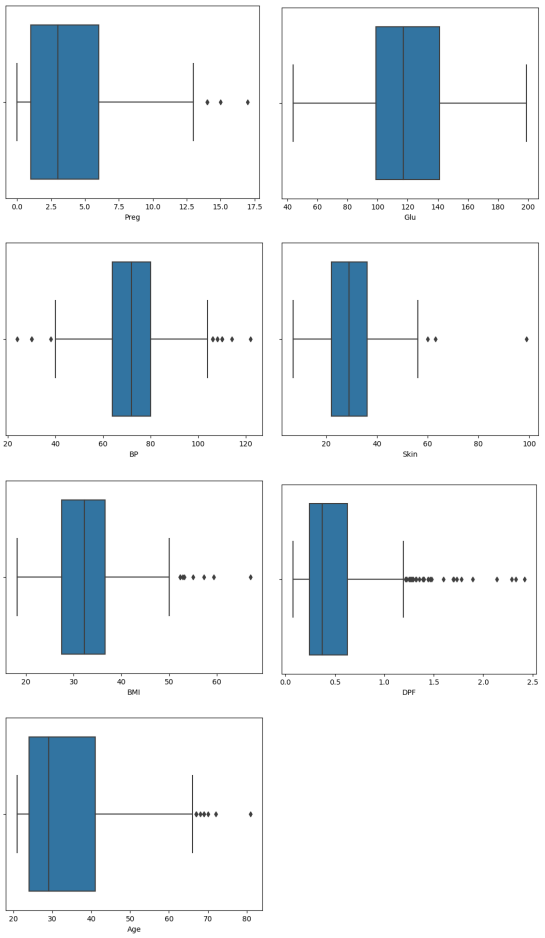

Figure 4. Boxplots represent the distribution of all features, illustrating the median, quartiles, and potential outliers.

d. Outlier Treatment

Outliers identified in our boxplots significantly deviate from the core data distribution. To address these anomalies, we adopted a standard approach using the Interquartile Range (IQR). Values lying outside the bounds defined by Q1 - 1.5 * IQR and Q3 + 1.5 * IQR are considered outliers. This method, grounded in statistical principles, ensures that extreme data points, which can skew results, are systematically excluded, enhancing the robustness of our analyses.

e. Imputation

Addressing missing data is pivotal for ensuring the quality and integrity of our analyses. In our quest to find the optimal imputation strategy, we selected three prominent methods: MissForest, KNN imputation, and IterativeImputer. Each of these methods offers advantages: MissForest leverages random forests and is non-parametric; KNN imputation considers similarity between data points; and IterativeImputer models each feature with missing values as a function of other features, iterating until a specified stopping criterion. To comprehensively assess the impact of imputation, we applied each technique to our three distinct datasets: the original, the one with encoded BMI and Glucose, and the scaled BMI and Glucose dataset. This rigorous approach ensures we can ascertain the most effective data-imputation combination for our subsequent binary classification tasks.

f. Handle Unbalanced Data

After eliminating the outliers and imputing missing values with the above 3 techniques, we were left with 698 records, 232 of which tested positive for diabetes and 466 of which tested negative for diabetes, which clearly demonstrates the unbalance of data. We turned to three renowned techniques: SMOTE (Synthetic Minority Over-sampling Technique), upsampling, and downsampling. SMOTE synthesizes new minority samples by interpolating between existing ones, enriching the under-represented class. Upsampling, or over-sampling, involves increasing the minority class by duplicating instances, while downsampling reduces the majority class to achieve balance. Deploying these methods across our distinct datasets—original, encoded BMI and Glucose, and scaled BMI and Glucose—and combined with the three imputation techniques, we've cultivated a rich matrix of 3x3x3 dataset combinations. This exhaustive approach ensures the exploration of multiple data landscapes, aiming to identify the best balance-imputation-dataset synergy for our binary classification challenge.

g. Train and Test Split

To ensure a robust model evaluation, we divided our dataset into training and testing subsets using Scikit-Learn's `train_test_split`. With an 80:20 split, the model is trained on a majority portion and then validated on the remaining unseen data. This random division ensures diverse data representation and prevents potential data leakage, guaranteeing genuine performance metrics.

i. Proposed Classification Model

In our pursuit to design a robust binary classification algorithm, we have selected 5 models, each with distinct strengths, theoretical underpinnings, and potential limitations. Our ensemble comprises logistic regression, support vector machine (SVM), random forest, XGBoost, and an Artificial Neural Network (ANN). These models have been chosen based on their historical performance, adaptability, and popularity in handling binary classification problems.

a. Comparison

1. **Logistic Regression:**
   Strength: Simplicity, probabilistic interpretation, and ease of implementation. Works well when relationships are approximately linear.
   Limitations: Assumes a linear relationship between dependent and independent variables. Can struggle with complex inter-variable relationships.
   Scaling: Generally benefits from feature scaling to converge faster during optimization.
2. **Support Vector Machine (SVM)**:
   Strength: Powerful for capturing complex relationships in data. Effective in high dimensional spaces.
   Limitations: Less effective on noisier datasets with overlapping classes. Computational inefficiency on large datasets.
   Scaling: Essential. SVMs are sensitive to the scale of the data, which can affect the decision boundary.
3. **Random Forest:**
   Strength: Handles large datasets well, can manage a mix of numerical and categorical features, and offers feature importance metrics.
   Limitations: Can overfit on noisy datasets. Model interpretability is not as straightforward as some other methods.
   Scaling: Generally not required, as decision trees are not sensitive to the scale of the data.
4. **XGBoost:**
   Strength: Regularized boosting technique, often outperforms other algorithms. Great for structured/tabular data.
   Limitations: Can overfit if not tuned properly. Takes longer to train on very large datasets.

Scaling: Like Random Forest, XGBoost (being tree-based) typically does not require feature scaling.

5. **Artificial Neural Network (ANN):**
   Strength: Highly flexible, can model complex, non-linear relationships.
   Limitations: Requires significant data and computational power. Black-box nature makes interpretation challenging.
   Scaling: Essential. ANNs require standardized input features to ensure consistent weight adjustments during backpropagation.

b. Hyperparameter tuning

1. **Logistic Regression:**
   We use optuna to find the best combination of hyperparameters by cross-validation.
2. **Support Vector Machine (SVM):**
   Kernel: For this task we experimentation with various kernels, including linear, sigmoid, polynomial, and RBF.
   Degree for polynomial kernel: For this task we first try default degree setting or 3 and try with other settings.
   Gamma for RBF kernel: For this task we using 'auto' and default settings allowed the algorithm to calculate or choose sensible values based on the dataset's characteristics.
3. **Random Forest:**
   This code utilizes the Optuna library to optimize hyperparameters for a Random Forest classifier. Hyperparameters being optimized:
   Number of trees in the forest: 100 to 1000 in steps of 100.
   Maximum depth of the trees: 2 to 32 in steps of 2.
   Minimum number of samples required to split an internal node: 0.1 to 1.0 in steps of 0.1.
   Minimum number of samples required to be at a leaf node: 0.1 to 0.5 in steps of 0.1.
4. **XGBoost:**
   These modifications are integral to achieving better predictive accuracy and model robustness within the context of our binary classification task.
   Objective ("binary:logistic"): The choice of objective function emphasizes the model's suitability for predicting probabilities in binary classification.
   Eval Metric ("logloss"): The use of "logloss" as the evaluation metric prioritizes optimizing probability estimates.
   Booster ("gbtree", "gblinear", "dart"): Making the booster parameter tunable allows for the selection of the best booster type during optimization. Options include tree-based models, linear models, and tree-based models with dropout regularization.
   Lambda ("lambda"), Alpha ("alpha"), Learning Rate ("learning_rate"): Tunable regularization terms and learning rate facilitate control over overfitting and model convergence.
   n_estimators ("n_estimators"): The tunable number of boosting rounds influences model complexity and the risk of overfitting.
   Max Depth ("max_depth"): If applicable, the max_depth parameter controls the depth of each decision tree, affecting model complexity.
   Gamma ("gamma"): Tunable gamma adds regularization by setting the minimum loss reduction for further leaf node partitioning.
   Grow Policy ("grow_policy"): Tunable grow_policy influences tree construction methodology, impacting model behavior.
5. **Artificial Neural Network (ANN):**
   Hyperparameter Tuning: we experiment with different values for dropout rates and regularization strengths to optimize the performance
   Learning Rate Schedule: We refine the learning rate schedule by adjusting parameters such as the initial learning rate and decay factor.
   Layer Modifying: We explore different layer configurations, including varying layer sizes and types.

4. Evaluation

The effectiveness of our proposed classification models demands precise assessment, and to achieve this, we have adopted a comprehensive evaluation framework. For each model—logistic regression, support vector machine (SVM), random forest, XGBoost, and Artificial Neural Network (ANN)—we employ a suite of evaluation metrics, including the confusion matrix, accuracy rate, and visualizations of model loss and accuracy across training and validation sets. These metrics and graphical representations provide a multi-faceted view of each model's performance, ensuring a thorough understanding of their strengths and areas of improvement.

Evaluation results with our distinct datasets—original, encoded BMI and Glucose, and scaled BMI and Glucose—combined with the three imputation techniques—and combined with the three unbalanced data handling techniques:

1. **Logistic Regression:**

a. Original BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:74 , FP:24 , FN:21 , TN:68 Accuracy rate: 0.76	Confusion matrix: TP: 75, FP: 26, FN: 15, TN: 71 Accuracy rate: 0.78	Confusion matrix: TP:33 , FP:14 , FN:14 , TN:32 Accuracy rate: 0.70
KNN imputation	Confusion matrix: TP:72 , FP:26 , FN:21 , TN:68 Accuracy rate: 0.75	Confusion matrix: TP: 69, FP: 32, FN: 19, TN: 67 Accuracy rate: 0.73	Confusion matrix: TP:31 , FP:16 , FN:12 , TN:34 Accuracy rate: 0.70
Iterative Imputer	Confusion matrix: TP:74, FP:24 , FN:19 , TN:70 Accuracy rate: 0.77	Confusion matrix: TP:75 , FP:26 , FN:16 , TN:70 Accuracy rate: 0.77	Confusion matrix: TP:32, FP:15 , FN:12 , TN:34 Accuracy rate: 0.71

b. Encoded BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:76 , FP:22 , FN:19 , TN:70 Accuracy rate: 0.78.	Confusion matrix: TP:75 , FP:26 , FN:26 , TN:60 Accuracy rate: 0.74	Confusion matrix: TP: 37, FP: 10, FN: 13, TN: 33 Accuracy rate: 0.75
KNN imputation	Confusion matrix: TP:77 , FP:21 , FN:18 , TN:71 Accuracy rate: 0.79	Confusion matrix: TP:76 , FP:25, FN:27, TN:59 Accuracy rate: 0.72	Confusion matrix: TP:35 , FP:12 , FN:11 , TN:35 Accuracy rate: 0.75
Iterative Imputer	Confusion matrix: TP:76 , FP:22 , FN:18 , TN:71 Accuracy rate: 0.78	Confusion matrix: TP:75 , FP:26, FN:25 , TN:61 Accuracy rate: 0.72	Confusion matrix: TP:38 , FP:9 , FN:13 , TN:33 Accuracy rate: 0.76

c. Scaled BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:76 , FP:22 , FN:22 , TN:67 Accuracy rate: 0.76	Confusion matrix: TP:75 , FP:26 , FN:15 , TN:71 Accuracy rate: 0.78	Confusion matrix: TP:33 , FP:14 , FN:14 , TN:32 Accuracy rate: 0.70
KNN imputation	Confusion matrix: TP:74 , FP:24 , FN:20 , TN:69 Accuracy rate: 0.76	Confusion matrix: TP:69 , FP:32 , FN:19 , TN:67 Accuracy rate: 0.73	Confusion matrix: TP:31 , FP:16 , FN:11 , TN:35 Accuracy rate: 0.71
Iterative Imputer	Confusion matrix: TP:74 , FP:24 , FN:21 , TN:68 Accuracy rate: 0.75	Confusion matrix: TP:74 , FP:27 , FN:16 , TN:70 Accuracy rate: 0.77	Confusion matrix: TP:32 , FP:15 , FN:12 , TN:34 Accuracy rate: 0.71

2. **Support Vector Machine (SVM)**

a. Original BMI and Glucose

	Upsampling
Iterative Imputer	Kernel='linear': Confusion matrix: [[74 27][24 62]] Accuracy rate: 0.73 Kernel='sigmoid': Confusion matrix: [[60 31][38 58]] Accuracy rate: 0.64 Kernel='poly': Confusion matrix: [[78 22][20 67]] Accuracy rate: 0.78 Kernel='poly',degree=2: Confusion matrix: [[68 38][30 51]] Accuracy rate: 0.67 Kernel='rbf': Confusion matrix: [[77 16][21 73]] Accuracy rate: 0.80 Kernel='rbf',gamma='auto': Confusion matrix: [[77 16][21 73]] Accuracy rate: 0.80 LinearSVC: Confusion matrix: [[73 21][25 68]] Accuracy rate: 0.75

*Example from the colab

b. Encoded BMI and Glucose

	Upsampling
MissForest	.Kernel='linear': Confusion matrix: [[80 22][18 67]]

	Accuracy rate: 0.79 Kernel='sigmoid': Confusion matrix: [[66 25][32 64]] Accuracy rate: 0.70 Kernel='poly': Confusion matrix: [[83 23][15 66]] Accuracy rate: 0.80 Kernel='poly',degree=2: Confusion matrix: [[85 31][13 58]] Accuracy rate: 0.76 Kernel='rbf': Confusion matrix: [[83 17][15 72]] Accuracy rate: 0.83 Kernel='rbf',gamma='auto': Confusion matrix: [[83 17][15 72]] Accuracy rate: 0.83 LinearSVC: Confusion matrix: [[78 19][20 70]] Accuracy rate: 0.79

*Example from the colab

c. Scaled BMI and Glucose

	Upsampling
Iterative Imputer	.Kernel='linear': Confusion matrix: [[74 27][24 62]] Accuracy rate: 0.73 Kernel='sigmoid': Confusion matrix: [[60 31][38 58]] Accuracy rate: 0.63 Kernel='poly': Confusion matrix: [[78 22][20 67]] Accuracy rate: 0.78 Kernel='poly',degree=2: Confusion matrix: [[68 38][30 51]] Accuracy rate: 0.64 Kernel='rbf': Confusion matrix: [[77 16][21 73]] Accuracy rate: 0.80 Kernel='rbf',gamma='auto': Confusion matrix: [[77 16][21 73]] Accuracy rate: 0.80 LinearSVC: Confusion matrix: [[73 21][25 68]] Accuracy rate: 0.75

*Example from the colab

3. Random Forest:

a. Original BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:80 , FP:18 , FN:22 , TN:67 Accuracy rate: 0.79	Confusion matrix: TP:74 , FP:27 , FN:17 , TN:69 Accuracy rate: 0.76	Confusion matrix: TP:31 , FP:16 , FN:10 , TN:36 Accuracy rate: 0.75
KNN imputation	Confusion matrix: TP:73 , FP:25 , FN:18 , TN:71 Accuracy rate: 0.77	Confusion matrix: TP:75 , FP:26 , FN:15 , TN:71 Accuracy rate: 0.78	Confusion matrix: TP:33 , FP:14 , FN:11 , TN:35 Accuracy rate: 0.73
Iterative Imputer	Confusion matrix: TP:76 , FP:22 , FN:13 , TN:76 Accuracy rate: 0.81	Confusion matrix: TP:73 , FP:28 , FN:13 , TN: 73 Accuracy rate: 0.78	Confusion matrix: TP:34 , FP:13 , FN:10 , TN:36 Accuracy rate: 0.75

b. Encoded BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:78 , FP:20 , FN:15 , TN:74 Accuracy rate: 0.81	Confusion matrix: TP:74 , FP:27 , FN:21 , TN:65 Accuracy rate: 0.74	Confusion matrix: TP:32 , FP:15 , FN:10 , TN:36 Accuracy rate: 0.73
KNN imputation	Confusion matrix: TP:77 , FP:21 , FN:15 , TN:74 Accuracy rate: 0.81	Confusion matrix: TP:75 , FP:26 , FN:25 , TN:61 Accuracy rate: 0.73	Confusion matrix: TP:31 , FP:16 , FN:10 , TN:36 Accuracy rate: 0.72
Iterative Imputer	Confusion matrix: TP: 76 FP: 22 FN: 13 TN: 76 Accuracy rate: 0. 81	Confusion matrix: TP: 75 FP: 26 FN: 14 TN: 72 Accuracy rate: 0.79	Confusion matrix: TP: 34 FP: 13 FN: 10 TN: 36 Accuracy rate: 0.75

c. Scaled BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP:76 , FP:22 , FN:18 , TN:71 Accuracy rate: 0.79	Confusion matrix: TP:74 , FP:27 , FN:17 , TN:69 Accuracy rate: 0.76	Confusion matrix: TP:34 , FP:13 , FN:10 , TN:36 Accuracy rate: 0.75
KNN imputation	Confusion matrix: TP:76 ,	Confusion matrix: TP:73 ,	Confusion matrix: TP:34 ,

	FP:22 , FN:21 , TN:68 Accuracy rate: 0.77	FP:28 , FN:15 , TN:71 Accuracy rate: 0.77	FP:13 , FN:10 , TN:36 Accuracy rate: 0.75
Iterative Imputer	Confusion matrix: TP:76 , FP:22 , FN:13 , TN:76 Accuracy rate: 0.81	Confusion matrix: TP:75 , FP:26 , FN:14 , TN:72 Accuracy rate: 0.79	Confusion matrix: TP:31 , FP:16 , FN:9 , TN:37 Accuracy rate: 0.73

4. XGBoost:

a. Original BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP: 86, FP: 12, FN: 3, TN: 86 Accuracy rate: 0.92	Confusion matrix: TP: 77, FP: 24, FN: 6, TN: 80 Accuracy rate: 0.84	Confusion matrix: TP: 0, FP: 47, FN: 0, TN: 46 Accuracy rate: 0.49
KNN imputation	Confusion matrix: TP: 80, FP: 18, FN: 10, TN: 79 Accuracy rate: 0.85	Confusion matrix: TP: 80, FP: 21, FN: 6, TN: 80 Accuracy rate: 0.86	Confusion matrix: TP: 0, FP: 47, FN: 0, TN: 46 Accuracy rate: 0.49
Iterative Imputer	Confusion matrix: TP: 85, FP: 13, FN: 3, TN: 86 Accuracy rate: 0.91	Confusion matrix: TP: 82, FP: 19, FN: 11, TN: 75 Accuracy rate: 0.84	Confusion matrix: TP: 34, FP: 13, FN: 12, TN: 34 Accuracy rate: 0.73

b. Encoded BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP: 80, FP: 18, FN: 6, TN: 83 Accuracy rate: 0.87	Confusion matrix: TP: 78, FP: 23, FN: 19, TN: 67 Accuracy rate: 0.78	Confusion matrix: TP: 35, FP: 12, FN: 10, TN: 36 Accuracy rate: 0.76
KNN imputation	Confusion matrix: TP: 81,	Confusion matrix: TP: 80,	Confusion matrix: TP: 0,
	FP: 17, FN: 4, TN: 85 Accuracy rate: 0.88	FP: 21, FN: 17, TN: 69 Accuracy rate: 0.80	FP: 47, FN: 0, TN: 46 Accuracy rate: 0.49
Iterative Imputer	Confusion matrix: TP: 80, FP: 18, FN: 3, TN: 86 Accuracy rate: 0.89	Confusion matrix: TP: 82, FP: 19, FN: 15, TN: 71 Accuracy rate: 0.82	Confusion matrix: TP: 0, FP: 47, FN: 0, TN: 46 Accuracy rate: 0.49

c. Scaled BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP: 87, FP: 11, FN: 4, TN: 85 Accuracy rate: 0.92	Confusion matrix: TP: 76, FP: 25, FN: 12, TN: 74 Accuracy rate: 0.80	Confusion matrix: TP: 34, FP: 13, FN: 12, TN: 34 Accuracy rate: 0.73
KNN imputation	Confusion matrix: TP: 86, FP: 12, FN: 4, TN: 85 Accuracy rate: 0.91	Confusion matrix: TP: 81, FP: 20, FN: 5, TN: 81 Accuracy rate: 0.87	Confusion matrix: TP: 34, FP: 13, FN: 13, TN: 33 Accuracy rate: 0.72
Iterative Imputer	Confusion matrix: TP: 88, FP: 10, FN: 3, TN: 86 Accuracy rate: 0.93	Confusion matrix: TP: 82, FP: 19, FN: 11, TN: 75 Accuracy rate: 0.84	Confusion matrix: TP: 34, FP: 13, FN: 12, TN: 34 Accuracy rate: 0.73

5. Artificial Neural Network (ANN):

a. Scaled BMI and Glucose

	Upsampling	SMOTE	Downsampling
MissForest	Confusion matrix: TP: 68 FP: 30 FN: 11 TN:78 Accuracy rate: 0. 77	Confusion matrix: TP: 63 FP: 38 FN: 11 TN: 75 Accuracy rate: 0.75	Confusion matrix: TP: 31 FP: 16 FN: 5 TN: 41 Accuracy rate: 0.77
KNN imputation	Confusion matrix: TP: 68	Confusion matrix: TP: 60	Confusion matrix: TP: 31

	FP: 30 FN: 9 TN: 80 Accuracy rate: 0. 79	FP: 41 FN: 8 TN: 78 Accuracy rate: 0.74	FP: 16 FN: 5 TN: 41 Accuracy rate: 0.77
Iterative Imputer	Confusion matrix: TP: 68 FP: 30 FN: ,9 TN:80 Accuracy rate: 0. 79	Confusion matrix: TP: 67 FP: 31 FN: 9 TN: 80 Accuracy rate: 0.75	Confusion matrix: TP: 31 FP: 16 FN: 5 TN: 41 Accuracy rate: 0.77

5. Conclusion

Among the five algorithms considered, our XGBoost model emerged as the standout performer, achieving an outstanding accuracy of 93%. This remarkable level of accuracy underscores the effectiveness of hyperparameter tuning and meticulous model training. It has positioned the XGBoost algorithm as a powerful tool for binary classification tasks, particularly in scenarios where precision and robust prediction are paramount.

Furthermore, it is noteworthy that we conducted our experiments using datasets scaled according to medical terms, enhancing the clinical relevance of our findings. This scaling strategy allowed us to align our models more closely with the domain-specific nuances of healthcare applications.

Our extensive exploration of hyperparameter optimization and model training demonstrated that, with careful parameter selection and optimization, the XGBoost algorithm can yield impressive results. The utilization of techniques such as early stopping and evaluation metric choice contributed to the model's overall excellence.

In conclusion, while each of the five algorithms explored in this study offered valuable insights and capabilities, the XGBoost model stands out as the clear choice for applications where high accuracy is a primary concern. Its ability to consistently deliver accurate binary classifications makes it a reliable and versatile tool in various domains, from healthcare to finance and beyond.

6. References

[1] National Center for Biotechnology Information. "Glucose Tolerance Test." U.S. National Library of Medicine, 23 April 2023, https://www.ncbi.nlm.nih.gov/books/NBK532915/

[2] National Center for Biotechnology Information. "Obesity and Type 2 Diabetes" U.S. National Library of Medicine, 19 June 2023, https://www.ncbi.nlm.nih.gov/books/NBK592412/