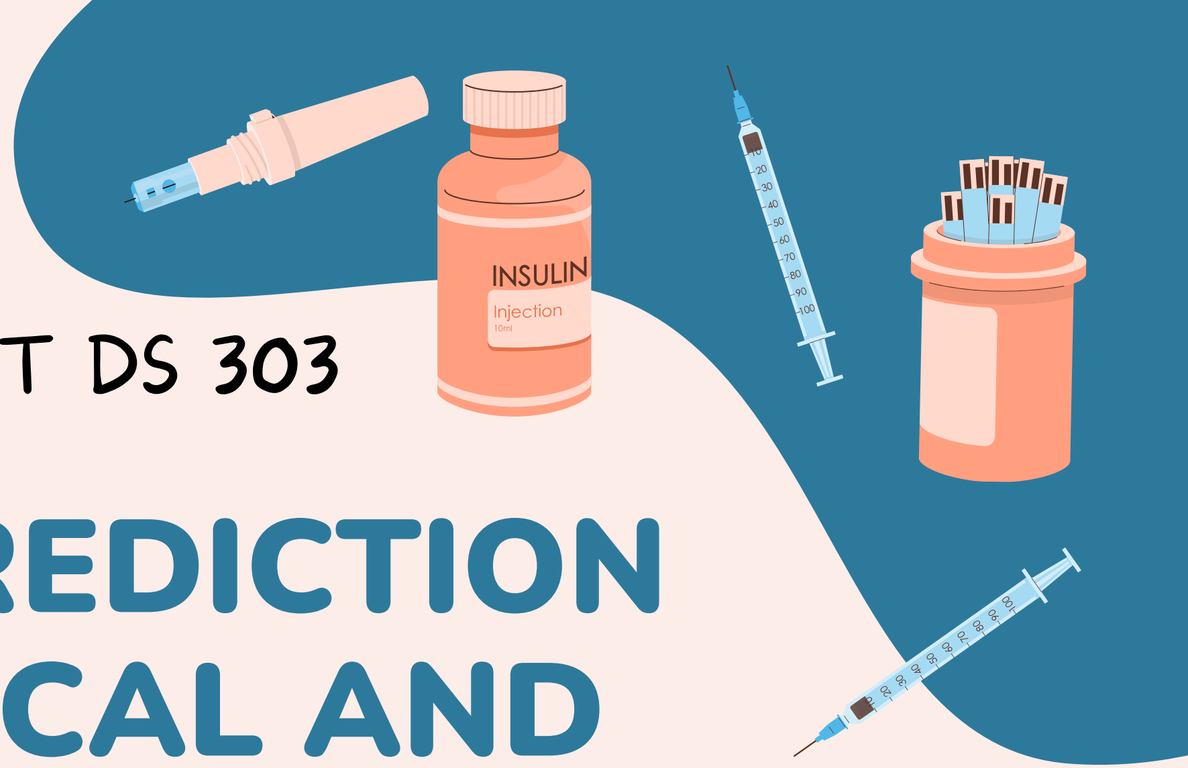
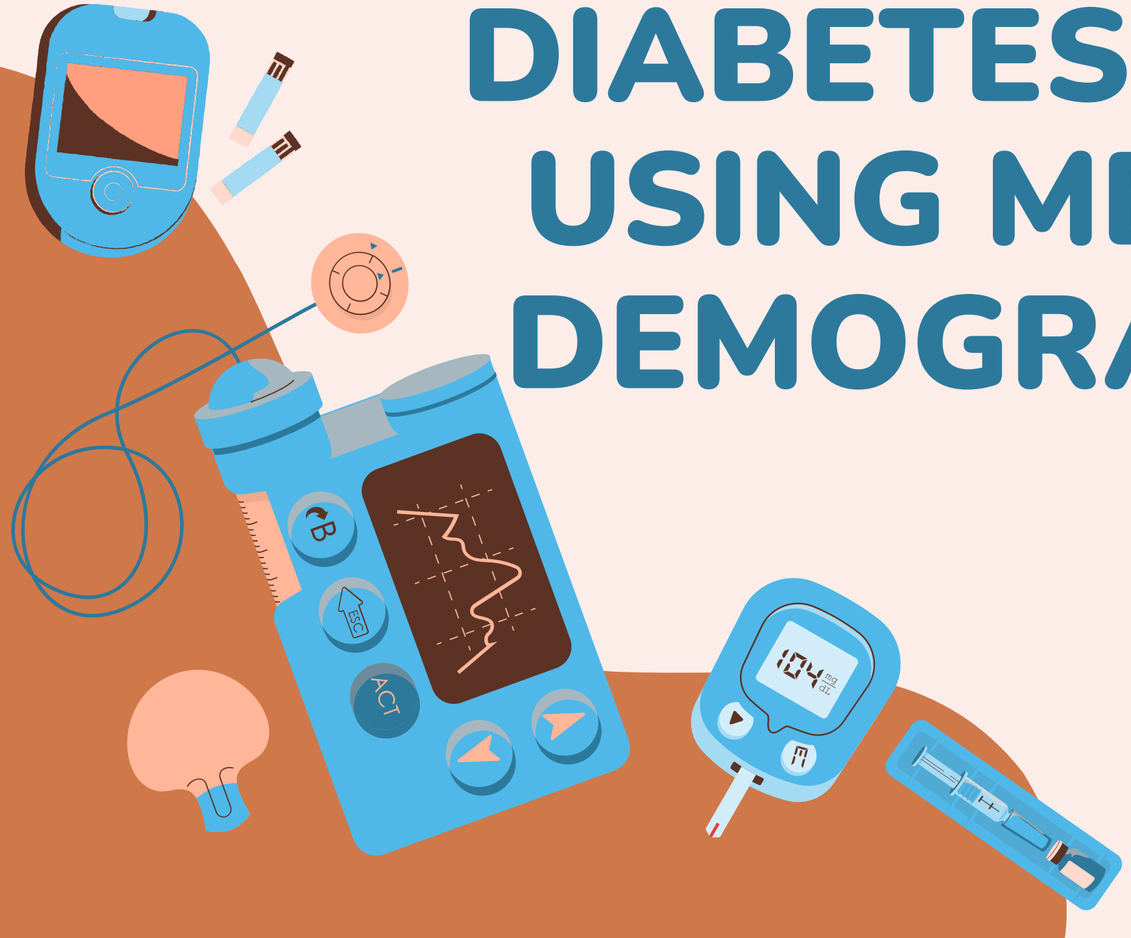


FINAL PROJECT DS 303

DIABETES PREDICTION USING MEDICAL AND DEMOGRAPHIC DATA

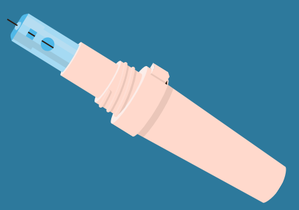


Team: Harsh Mehta (23b2453)
Hitesh Khiani (23b2415)
Neha Yadav(23b1807)
Yuvraj Singh Lodhi (23b0676)

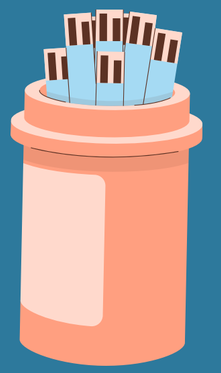
INTRODUCTION

This project utilizes a comprehensive dataset containing medical and demographic information to develop a machine learning model for predicting diabetes. By analyzing factors like age, BMI, blood glucose levels, and lifestyle habits, the model aims to assist healthcare professionals in early diagnosis and preventive care planning.

INTRODUCTION



Diabetes prediction falls under the broader domain of healthcare analytics and predictive modeling, where data-driven tools are used to support clinical decision-making. With rising global prevalence, diabetes presents significant health and economic challenges. Accurate prediction models can aid in early risk assessment, personalized treatment planning, and resource optimization in public health systems. Machine learning offers the advantage of uncovering complex, non-linear patterns in patient data, making it a valuable asset in modern medical diagnostics.



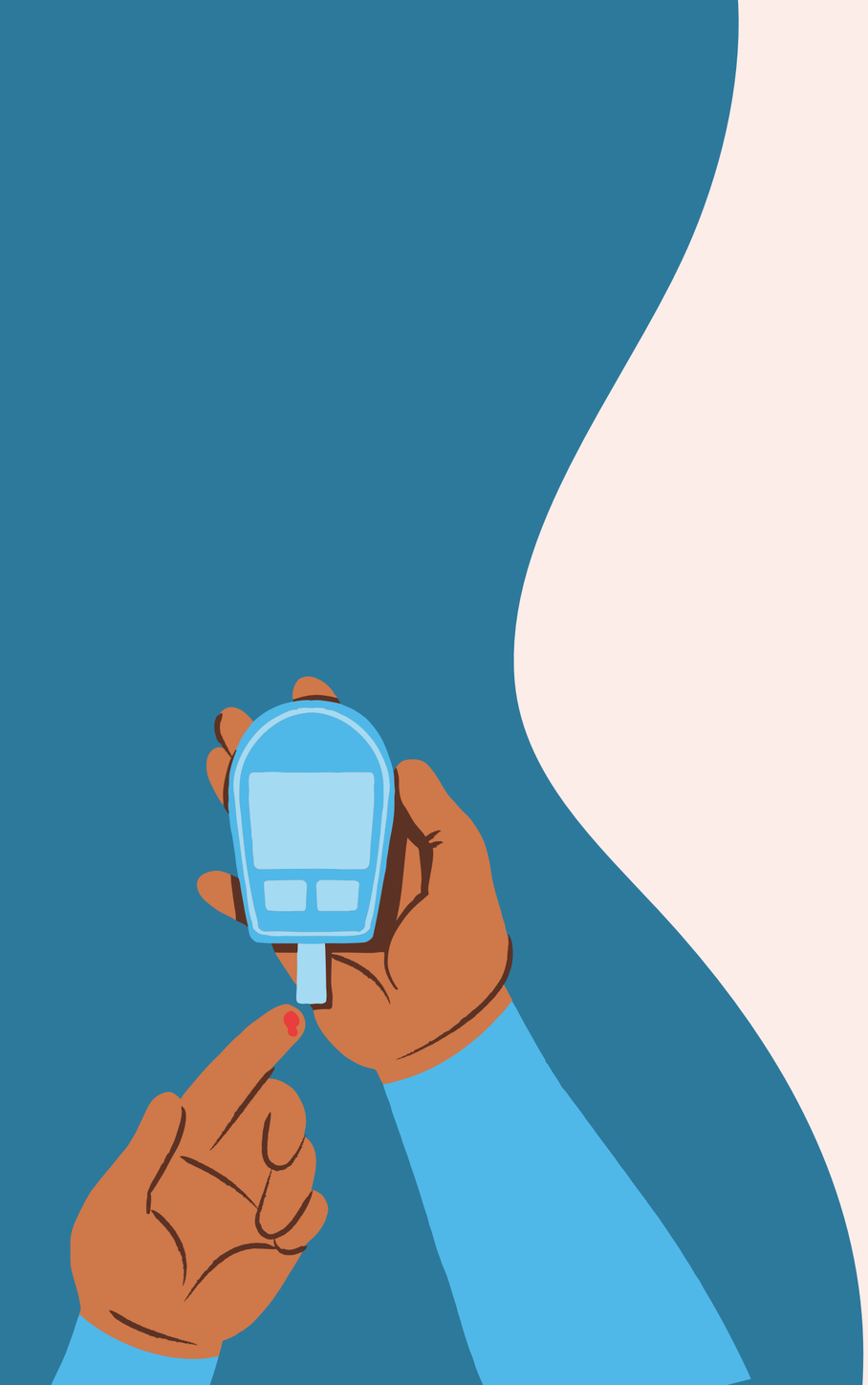
SOURCE OF DATA

www.kaggle.com

- The Diabetes Prediction Dataset is sourced from Electronic Health Records (EHRs) collected from multiple healthcare providers (hospitals, clinics).
- EHRs are digital patient records containing medical history, diagnoses, treatments, laboratory test results, and outcomes.
- Data was aggregated from different sources, then cleaned and preprocessed to remove inconsistencies and irrelevant information.
- Advantages of using EHRs:
  - Provides large, diverse patient data.
  - Offers long-term health tracking.
  - Ensures real-world healthcare relevance for predictive modeling.

How Data is Gathered?

- Data collected from patients diagnosed with or at risk of developing diabetes.
- Sources of Data:
  - Medical Records: Clinical diagnoses, treatments, disease history.
  - Surveys: Demographic details (e.g., age, gender, smoking habits).
  - Laboratory Tests: HbA1c levels, blood glucose levels.
- Processing Steps:
  - Data cleaning to remove errors and incomplete entries.
  - Standardization for consistency across different healthcare providers.
- Purpose: To support machine learning models and research in diabetes risk prediction.



FEATURES IN THE DATASET

Age – Patient’s age in years.

Gender – Biological sex (male/female/other).

Body Mass Index (BMI) – Weight relative to height.

Hypertension – High blood pressure (Yes/No).

Heart Disease – History of heart disease (Yes/No).

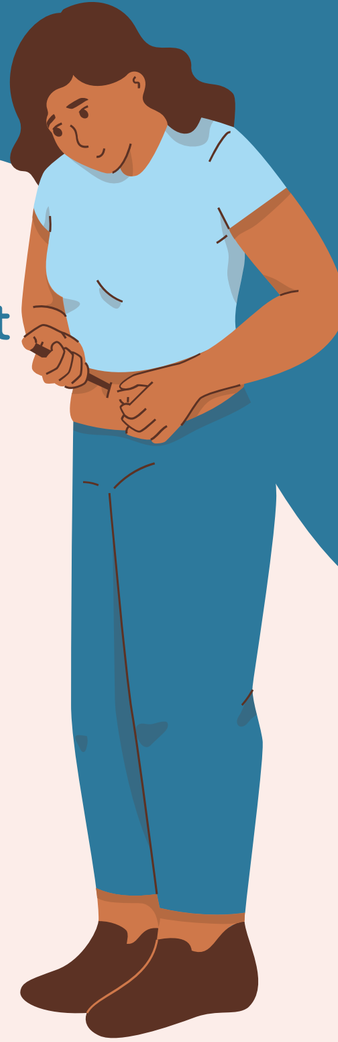
Smoking History – Smoking habits (never, former, current smoker).

HbA1c Level – Average blood sugar control over 2–3 months.

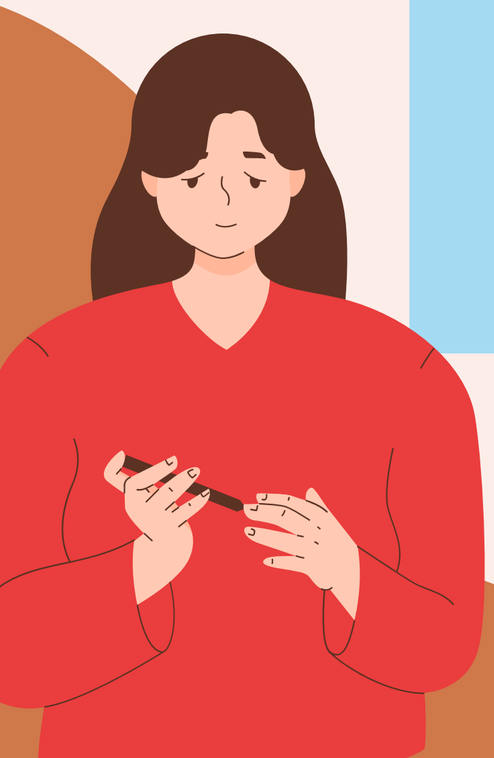
Blood Glucose Level – Current blood sugar level.

Diabetes Status – Target variable (positive/negative for diabetes).

RELATED WORK



PIMA Indian Diabetes dataset	NHANES Dataset (National Health and Nutrition Examination Survey)	MIMIC-III Database (Medical Information Mart for Intensive Care)
Logistic Regression ~75–78%	Logistic Regression ~78–81%	Logistic Regression: ~78–82%
Random Forest ~78–80%	Random Forest ~84–86%	Random Forest: ~81–85%
XGBoost ~79–82%	XGBoost ~86–88%	XGBoost ~83–87%

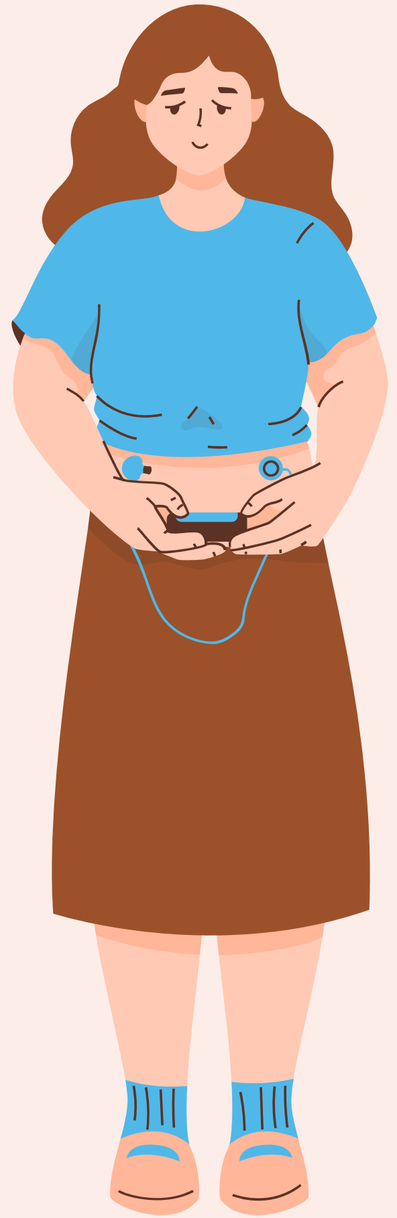


Applications

PIMA (UCI) [Diabetes classification]
NHANES (CDC) [Chronic disease prediction]
MIMIC-IIIICU [risk prediction, mortality]

WHAT WAS ACHIEVED?

- Accurate diabetes prediction using ML models on real-world datasets (PIMA, NHANES, MIMIC-III).
- XGBoost and Random Forest achieved 80–88% accuracy
- Predictive models enabled early diagnosis, risk stratification, and mortality forecasting



WHAT IT LACKED..

- Data imbalance and missing values affected reliability.
- Most datasets are region-specific(NHANES = US, MIMIC-III = one hospital)
- Diabetic cases are often a minority, causing models to favor majority (non-diabetic) class without proper balancing techniques

NOT REPRODUCING THE RESULTS THAT WE WERE AIMING

How We Aim to Improvise

- Compare multiple algorithms (Random Forest, Logistic, XGBoost) under same preprocessing pipeline
- Tune hyperparameters and address class imbalance using ensemble methods

Algorithm Used

We have tried the following algorithms:

1. Logistic Regression

- Simple and interpretable.
- Good baseline model for binary classification problems like diabetes prediction.

2. Random Forest

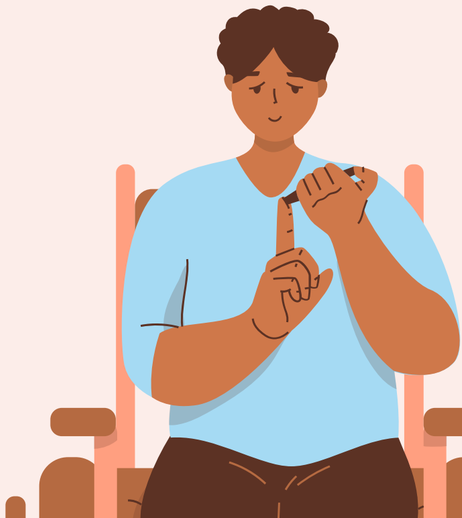
- Robust to noise and overfitting.
- Handles non-linear relationships well.
- Provides feature importance for explainability.

3. Gradient Boosting

- Builds strong learners sequentially.
- Often achieves higher accuracy on complex datasets like healthcare data.

4. SVM (Support Vector Machine)

- Works well in high-dimensional spaces.
- Effective for clearly separable classes.
- Kernels allow modeling complex decision boundaries.



5. KNN (K-Nearest Neighbors)

- Simple and intuitive.
- Makes predictions based on similarity with neighboring points.
- Useful for verifying consistency in predictions.

6. Naive Bayes

- Very fast and probabilistic.
- Performs surprisingly well when feature independence is assumed.
- Good baseline for comparison.

7. Decision Tree

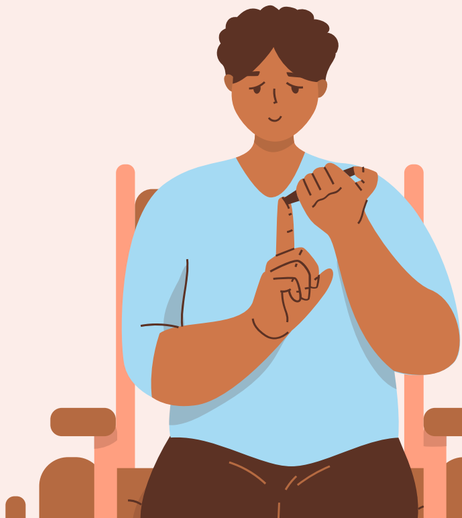
- Easy to interpret and visualize.
- Captures non-linear patterns.
- Serves as a base learner for many ensemble methods.

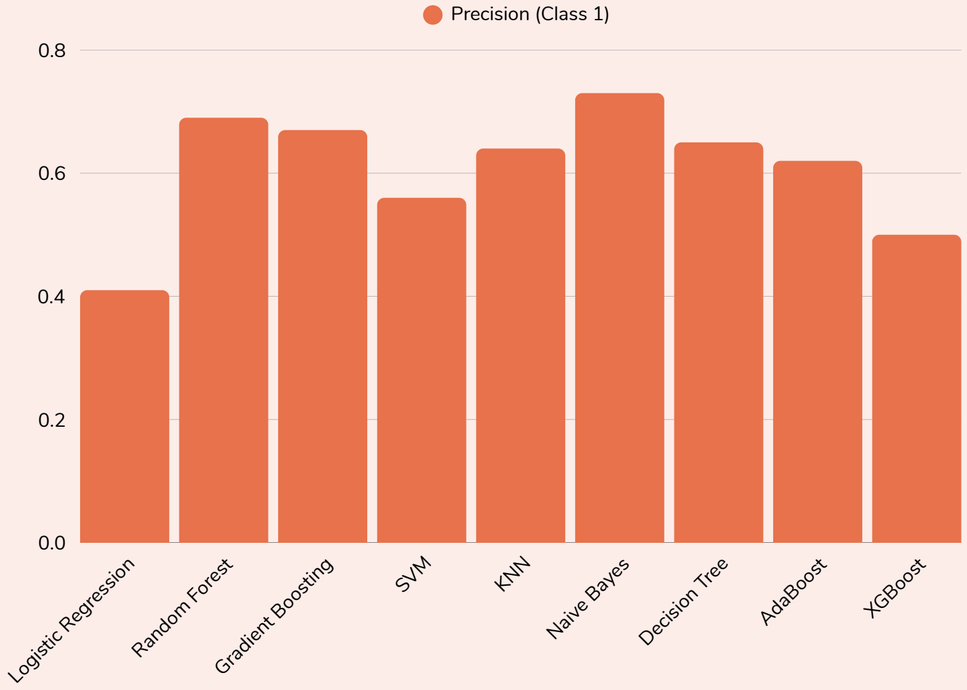
8. AdaBoost

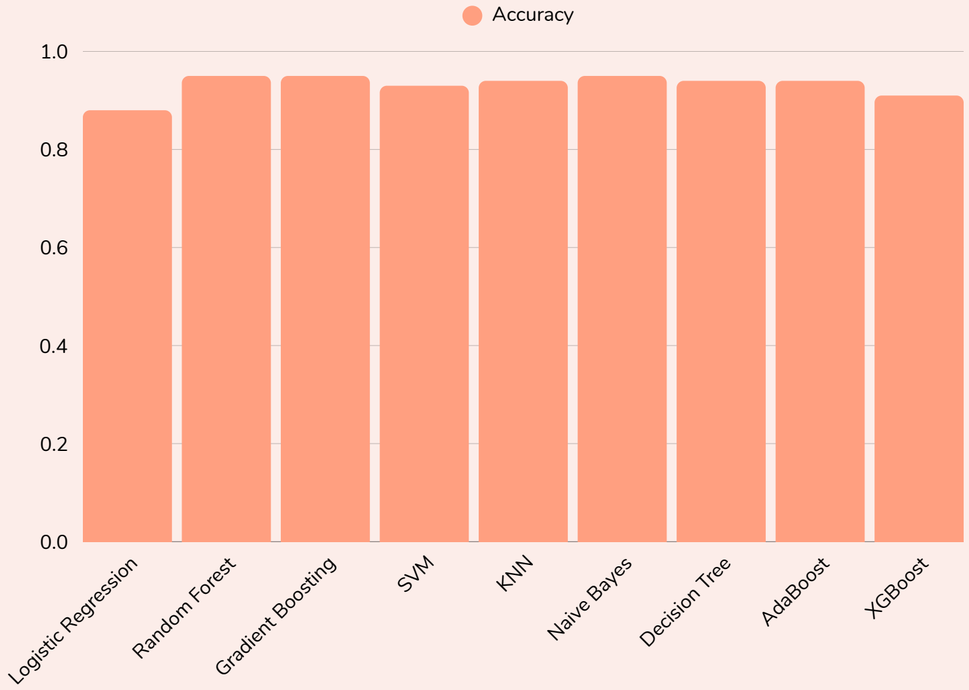
- Focuses on difficult-to-classify instances.
- Combines multiple weak learners into a strong one.
- Reduces bias in predictions.

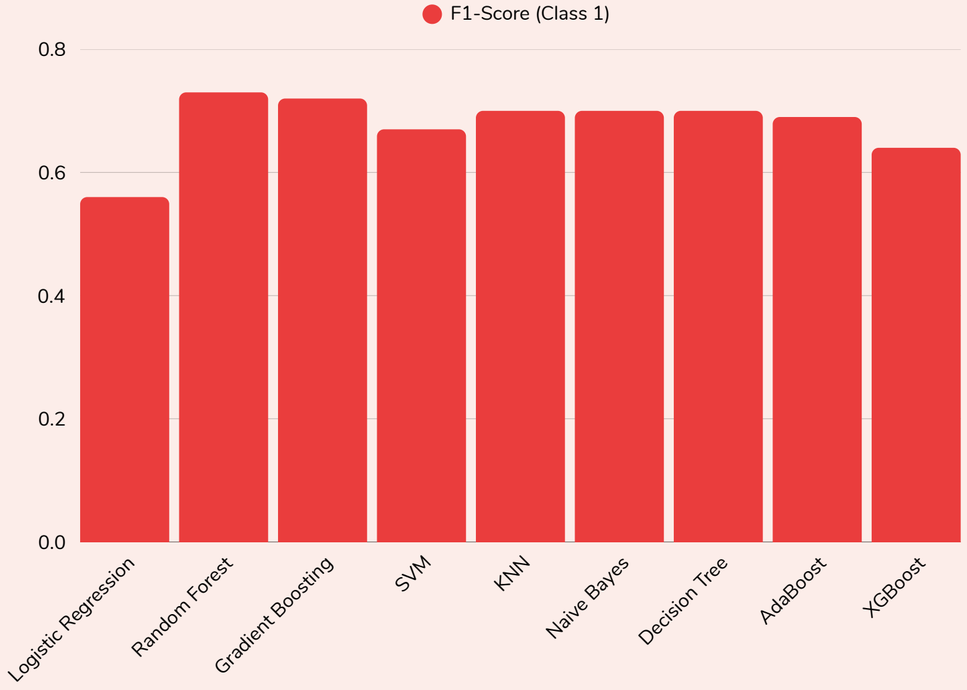
9. XGBoost

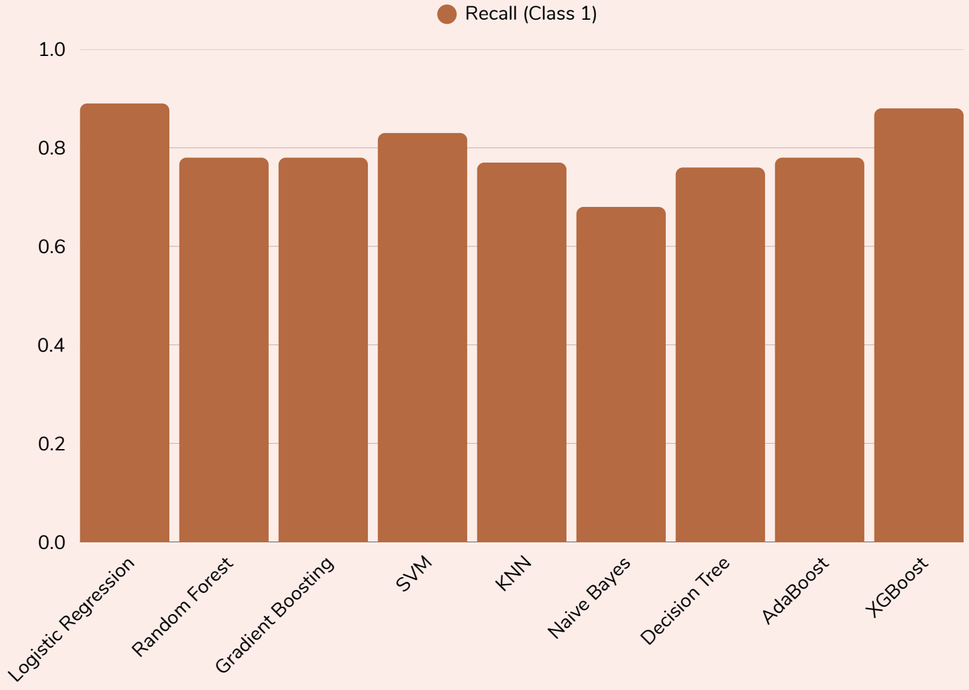
- Advanced gradient boosting algorithm.
- Known for high performance on structured/tabular data.
- Handles missing values and imbalanced data effectively.
- Used in many winning ML competitions.




Precision (Class 1)
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Gradient Boosting
SVM
KNN
Naive Bayes
Decision Tree
AdaBoost
XGBoost


Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Gradient Boosting
SVM
KNN
Naive Bayes
Decision Tree
AdaBoost
XGBoost


F1-Score (Class 1)
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Gradient Boosting
SVM
KNN
Naive Bayes
Decision Tree
AdaBoost
XGBoost


Recall (Class 1)
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Gradient Boosting
SVM
KNN
Naive Bayes
Decision Tree
AdaBoost
XGBoost

ALGORITHM USED

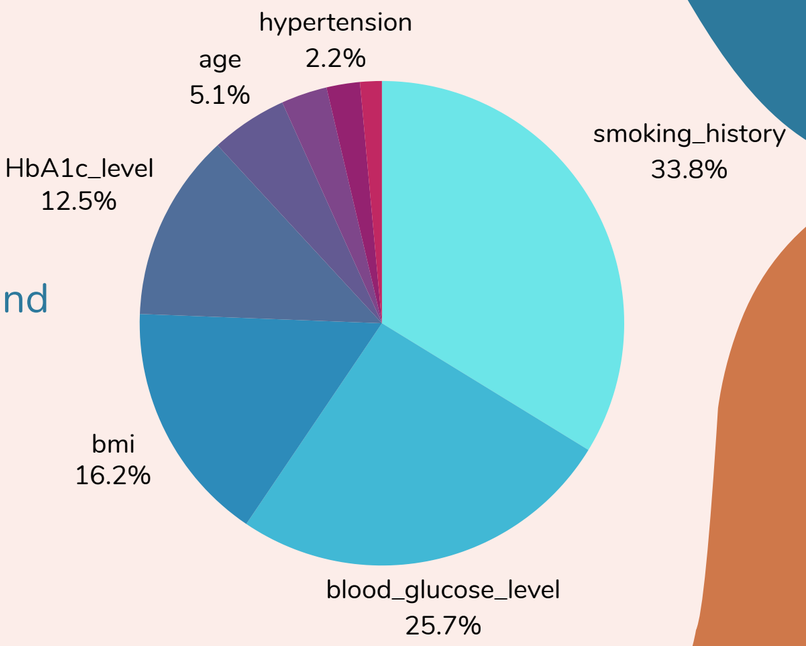
We selected the **Random Forest algorithm** for our final model as it achieved the highest F1-score and accuracy among all evaluated classifiers. It also demonstrated a strong balance between precision and recall, making it a robust choice for handling imbalanced medical datasets like ours. Its ensemble nature and resistance to overfitting further strengthen its reliability in real-world applications.

Approach

We explored several machine learning pipelines using a diabetes dataset. Each approach involved preprocessing, dimensionality reduction, and classification under varying sampling and modeling strategies, with careful consideration of class imbalance.

Common Preprocessing Steps (Applied in All Approaches)

- Dataset: CSV file db.csv.
- Label Encoding: Applied to categorical variables gender and smoking_history.
- Normalization: age, bmi, HbA1c_level, and blood_glucose_level scaled using StandardScaler.
- **Dimensionality Reduction: PCA with 6 components** used to reduce dimensionality and possibly improve model performance.
- Train-Test Split: Stratified **80/20 split** to maintain class balance in test data.



1. **Random Forest without Sampling (Baseline Model)**

- Goal: Establish baseline performance using original class distribution.

Method:

- Use RandomForestClassifier(class_weight='balanced').

Evaluation:

- Test set reflects original distribution (no oversampling).
- Performance metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix

Approach

2. Random Forest with SMOTE (Oversampling)

- Goal: Improve recall for minority class using **synthetic oversampling.**
- Method:
  - Apply **SMOTE only to training dat**a (after PCA).
  - Use RandomForestClassifier(class_weight='balanced').
- Evaluation:
  - Test set is never oversampled (**real-world simulation**).
  - Performance metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix

3. Multiple Classifier Ensemble with Undersampling

- Goal: Explore and compare different classifiers using ensemble learning.
- Undersampling Strategy:
  - **For 10 iterations**, sample **majority class to match 3.5× minority class** count.
- Models Used: Logistic Regression, Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Naive Bayes, k-Nearest Neighbors, Decision Tree, XGBoost.
- Training:
  - All models trained independently on each balanced sample.
  - Models are stored in per-type ensembles.
- Prediction:
  - Use soft voting (average predicted probabilities) across ensemble members.
- Evaluation:
  - Classification report (class-wise metrics + accuracy).
  - Confusion matrix.

Approach

4. Random Forest with Varying Undersampling Ratios

- Goal: Investigate how the class balance ratio affects performance.
- Ratios Tested: **1.0× to 6.0× (in 0.5 increments)** relative to minority class count.
- Method:
  - For each ratio, run 10 iterations of undersampling.
  - Train a RandomForestClassifier(class_weight='balanced') on each subset.
- Prediction:
  - Ensemble prediction by **averaging probabilities** of all 10 models.
- Evaluation:
  - Full classification report and confusion matrix printed for each ratio.
  - Used to identify optimal imbalance handling strategy.

5. Random Forest with Fixed Undersampling Ratio (3.5×)

- Goal: Use a mid-range fixed imbalance ratio to train an ensemble.
- Strategy:
  - Majority class undersampled to 3.5× minority class.
  - Trained 10 Random Forest models with class_weight='balanced}.
- Prediction:
  - Ensemble by averaging predicted probabilities.
- Evaluation:
  - Classification report and confusion matrix on fixed test set.

Evaluation and Testing

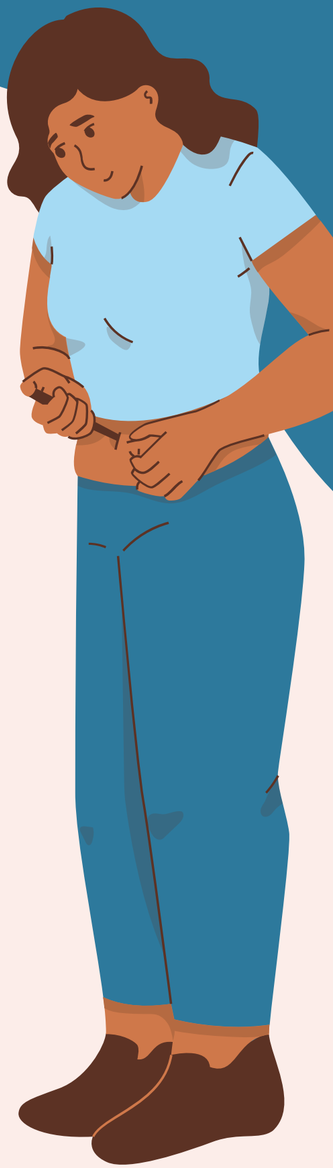
- Test Set Consistency: In all approaches, the test set is held fixed and never modified (no sampling).
- Performance Metrics:
  - Accuracy
  - Precision, Recall, F1-score (per class)
  - Support
  - Confusion Matrix
- Ensemble Evaluation: Used soft voting or average probabilities to reduce variance and improve stability.

Why this Approach?

We adopted a methodical, data-driven approach to build a reliable diabetes prediction model, with emphasis on dimensionality reduction, class imbalance handling, and performance stability.

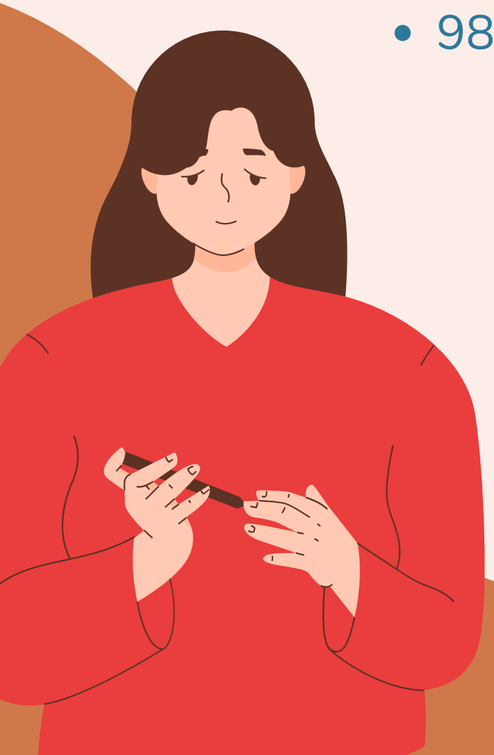
1. PCA for Dimensionality Reduction: **PCA reduced feature complexity** and noise, improving model efficiency while preserving predictive information. It also highlighted latent patterns among diabetic individuals.
2. Handling Class Imbalance: We evaluated both SMOTE and repeated random undersampling. SMOTE created synthetic minority samples for better generalization, while undersampling—paired with ensemble averaging—offered simplicity and stability.
3. Model Selection and Performance: Among nine classifiers, **Random Forest consistently performed best**, balancing precision and recall, and resisting overfitting—especially effective with resampling.
4. Ensemble Learning for Robustness: By averaging predictions across 10 training iterations, we **reduced variance** from random sampling and ensured more consistent evaluation.
5. Imbalance Ratio Analysis: Training with varying majority-minority ratios (1:1 to 6:1) helped us identify the sweet spot between **sensitivity and precision**—vital for medical use cases.

PERFORMANCE SUMMARY



Achievements:

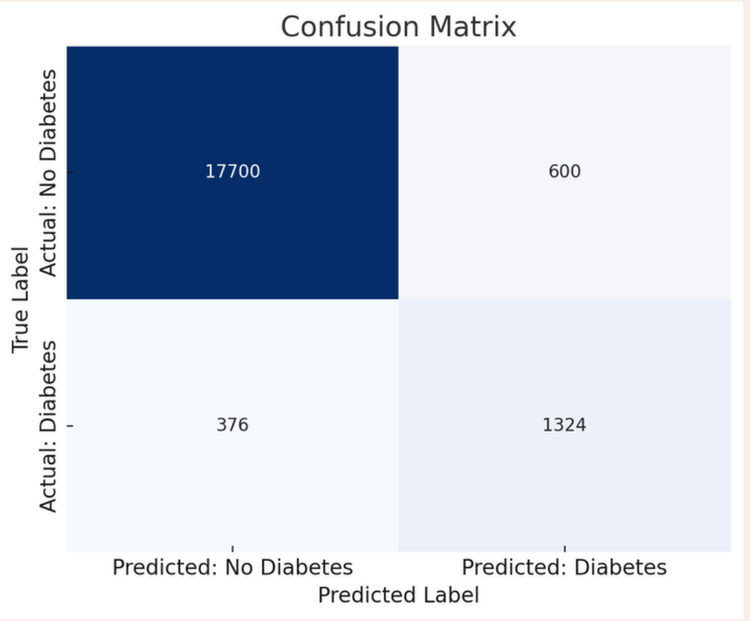
- After evaluating multiple sampling and modeling strategies, the best results were achieved using Random Forest with repeated undersampling (3.5x ratio).
- Achieved overall accuracy of 96%, demonstrating strong model reliability.
- Achieved 76% recall and 73% precision for the positive diabetic class.
- 98% precision and 97% recall for non-diabetic class.



Scope for Improvement:

- Future work can explore more on SMOTE and other advanced resampling methods to balance classes without losing majority class information.
- Introduce custom loss functions (like weighted loss or focal loss) to penalize misclassification of positive diabetics cases more heavily using boosting or deep learning models.

Ratio	Class 1 Precision	Class 1 Recall	Class 1 F1	Accuracy
1.0	0.44	0.92	0.59	0.89
1.5	0.52	0.87	0.65	0.92
2.0	0.58	0.82	0.68	0.94
2.5	0.64	0.79	0.71	0.94
3.0	0.69	0.78	0.73	0.95
3.5	0.76	0.74	0.75	0.96
4.5	0.79	0.73	0.76	0.96




```
Ensemble Model - Classification Report:

Class 0 - Precision: 0.98, Recall: 0.97, F1-score: 0.98, Support: 18300
Class 1 - Precision: 0.73, Recall: 0.76, F1-score: 0.74, Support: 1700

Accuracy: 0.96

Confusion Matrix:

[[17828   472]
 [  415  1285]]
```

THANK YOU