

# Multi-Class Diabetes Prediction

## Table of contents

1. Objective
  - Problem statements
  - Project Goals
  - Business Impact
2. Dataset
  - Dataset Overview
  - Class Distribution
  - Features and Descriptions
3. Experiment Methodology
  - Preprocessing
  - Feature Selection
  - Handling class Imbalance
4. Experiments
  - Classification Models
  - Clustering Models
  - Model Comparison
5. Error Analysis
6. Future Enhancements
7. Conclusion

# Objective

## 1.1 Problem statement

This project aims to analyze a multi-class diabetes dataset and develop machine learning models that can accurately predict diabetic status of individuals using both supervised classification and unsupervised clustering approaches. The dataset contains three target classes:

- N (Non-diabetic): Healthy individuals.
- P (Pre-diabetic): Individuals at risk of developing diabetes.
- Y (Diabetic): Individuals diagnosed with diabetes.

## 1.2 Project Goals

1. Build accurate classification models to predict diabetic status.
2. Apply clustering techniques to discover natural patient groupings.
3. Compare supervised vs unsupervised learning approaches.
4. Identify key biomarkers, risk factors and features that are most relevant for predicting diabetes.
5. Analyze misclassification and clustering errors to uncover limitations of the models.

## 1.3 Business Impact

This project aims to deliver tangible benefits to healthcare providers, patients and healthcare system:

- Early detection: Identify pre-diabetic patients at an early stage which reduce progression to diabetes.
- Clinical support: Assist healthcare professionals in risk assessments, patient stratifications, and decision-making based on data driven insights.

# 2. Dataset Analysis

## 2.1 Dataset Overview

The dataset used in this study is the Mendeley Diabetes Dataset, consisting of patient health records with clinical features relevant to diabetes prediction.

### Original Dataset

- Total Records: 1000 rows
- Total Features: 14 columns
- Target Variable: Diabetes status with three classes(N/Y/P)
- Data Quality:
  - No missing values
  - Contains duplicate patient ID's (200 duplicates)
  - Contains duplicate No\_Pation number (39 duplicates)
  - Whitespace in CLASS Column
    - The CLASS column contained leading/trailing whitespace (e.g., "Y " instead of "Y").

- This caused incorrect class separation, artificially splitting classes(Y and N appeared as two groups)
  - **Action Taken:** Applied `.str.strip()` to remove whitespace.

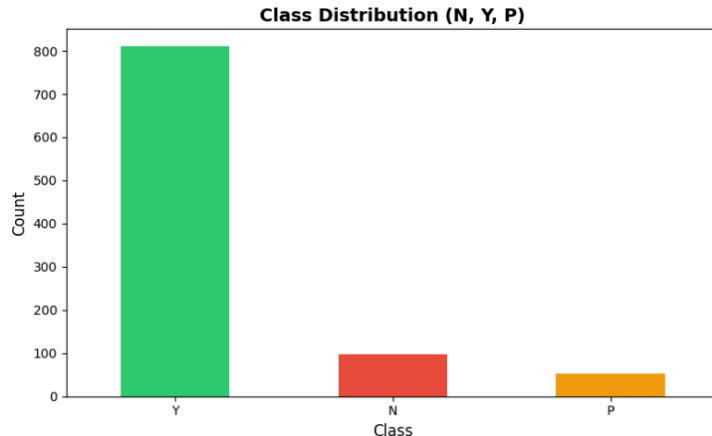
### After Data Cleaning

- Final Records: 961 rows
- Final Features: 13 columns
- Cleaning Steps Performed:
  - Removed 39 duplicate patient entries
  - Dropped ID column

## 2.2 Class Distribution

Here is the class distribution after clean the dataset

Class	Count	Percentage	Description
<b>N (Non-diabetic)</b>	<b>810</b>	<b>84%</b>	<b>Healthy individuals</b>
<b>P (Pre- diabetic)</b>	<b>53</b>	<b>5.5%</b>	<b>At risk individuals</b>
<b>Y (Diabetic)</b>	<b>98</b>	<b>10.2%</b>	<b>Diagnosed patients</b>



## 2.3 Feature Description

Here are the features with their description:

Feature Name	Description	Unit / Range	Notes / Medical Significance
Gender	Biological sex of the subject (F = Female, M = Male)	M or F	May influence metabolic parameters such as cholesterol and BMI

Feature Name	Description	Unit / Range	Notes / Medical Significance
AGE	Age in years	number (20 to 79)	Age is a major risk factor for Type 2 Diabetes
Urea	Urea level in blood	mg/dL (Normal: ~7–20)	Elevated levels may indicate kidney issues, common in diabetics
Cr (Creatinine)	Blood creatinine	mg/dL (Normal: ~0.6–1.3)	High values can suggest impaired kidney function
HbA1c	Glycated Hemoglobin percentage	% (Normal: <5.7%, Prediabetes: 5.7–6.4%, Diabetes: ≥6.5%)	Key diagnostic test for diabetes
Chol	Total cholesterol	mg/dL (Desirable: <200)	High cholesterol can indicate dyslipidemia, linked to diabetes
TG	Triglycerides	mg/dL (Normal: <150)	High TG often indicates insulin resistance
HDL	High-Density Lipoprotein	mg/dL (Optimal: >40 for men, >50 for women)	Low HDL is a risk marker for cardiovascular disease

Feature Name	Description	Unit / Range	Notes / Medical Significance
LDL	Low-Density Lipoprotein	mg/dL (Optimal: <100)	Elevated LDL increases heart disease risk
VLDL	Very-Low-Density Lipoprotein	mg/dL (Normal: 2–30)	High levels may indicate metabolic syndrome
BMI	Body Mass Index	kg/m <sup>2</sup> (Normal: 18.5–24.9; Overweight: 25–29.9; Obese: ≥30)	Strongly correlated with diabetes risk
CLASS	Target variable: N = Non-diabetic, Y = Diabetic, P = Predict-diabetic	Categorical	Labels for classification task

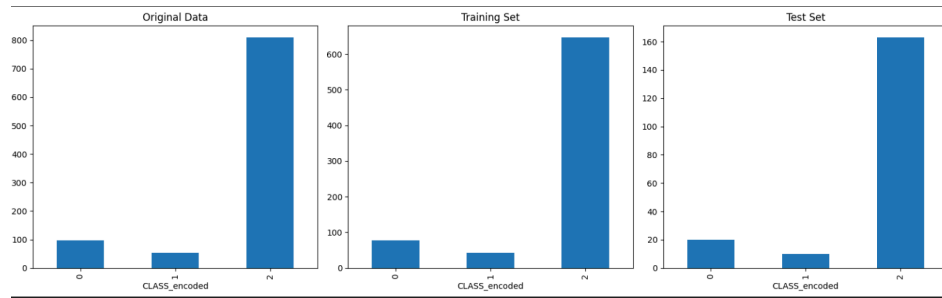
## 3. Experiment Methodology

### 3.1 Data Preprocessing

To prepare the dataset for machine learning experiments, I applied the following preprocessing steps.

- Feature Encoding
  - Categorical variables were converted into numerical form using Label Encoding to ensure compatibility with ML algorithms.
- Train-Test Split(Stratified Splitting):

- The dataset was split into 80% Training Data, 20% Testing Data using stratification, ensuring that class proportions (N, P, Y) remain consistent in both sets.



### 3. Feature Scaling:

- Applied StandardScaler to normalize numerical features
- Ensures that all features contribute equally and improves model performance

## 3.2 Feature Selection

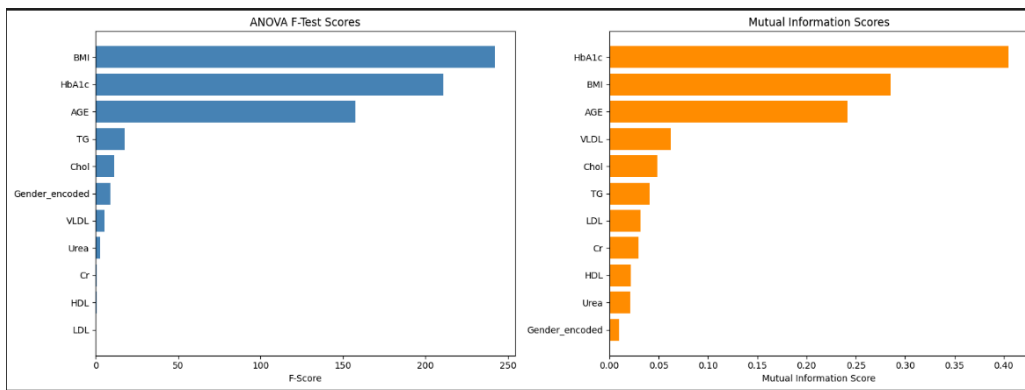
To identify the most informative features for predicting diabetic status, two feature selection techniques were applied

### 1. ANOVA F-Test

- Measures how well each feature separates the classes.

### 2. Mutual Information

- Measures the dependency between features and the target variable.



	Feature	F-Score_Norm	MI-Score_Norm	Combined_Score
3	HbA1c	0.870288	1.000000	0.935144
9	BMI	1.000000	0.697557	0.848779
0	AGE	0.650411	0.586591	0.618501
8	VLDL	0.021127	0.132573	0.076850
5	TG	0.070136	0.079086	0.074611
4	Chol	0.045656	0.097781	0.071718
7	LDL	0.000000	0.054867	0.027433
2	Cr	0.001335	0.050346	0.025841
1	Urea	0.008962	0.028623	0.018792
10	Gender_encoded	0.035068	0.000000	0.017534
6	HDL	0.000525	0.029903	0.015214

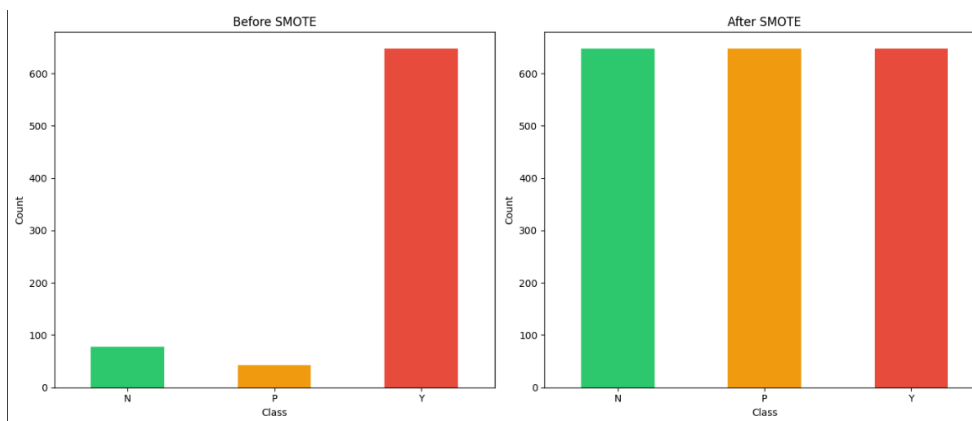
### 3.3 Class Imbalance Handling

The dataset exhibits a significant class imbalance with an imbalance ratio of 15.28, the majority class is over 15 times larger than the minority class. This imbalance can bias models toward predicting the majority class.

So, to prevent this I have implemented the technique called SMOTE(Synthetic Minority Oversampling Technique). This helps to generate synthetic samples for minority classes

- This is used only for supervised classification to ensure fair learning across all classes.
- It is applied to the training data only because test data should be real world data that can be imbalanced.

This ensured the classifier could learn equally from all classes and reduced bias toward the majority class.



For clustering models No SMOTE applied because Clustering aims to capture the natural structure of the dataset, and oversampling would introduce artificial patterns.

## 4. Experiments

### 4.1 Classification Models

#### 4.1.1 Model Selection:

There are many supervised machine learning algorithms, for this experiment I have selected four algorithms to evaluate their ability to predict diabetic status. Each model was chosen for its unique strengths.

1. Random Forest Classifier
  - I included Random forest because it works really well with tabular data . It uses many decision trees together, which makes it stable and less likely to overfit.
2. . XGBoost Classifier

- XGBoost is known for giving excellent performance, especially on datasets like this. It is fast, handles missing values internally, and has strong regularization, which helps prevent overfitting.

### 3. Support Vector Machine (SVM)

- It can capture complex, non-linear patterns in the data. Even though it's a bit slower on larger datasets, it usually generalizes well and is very reliable.

### 4. Logistic Regression

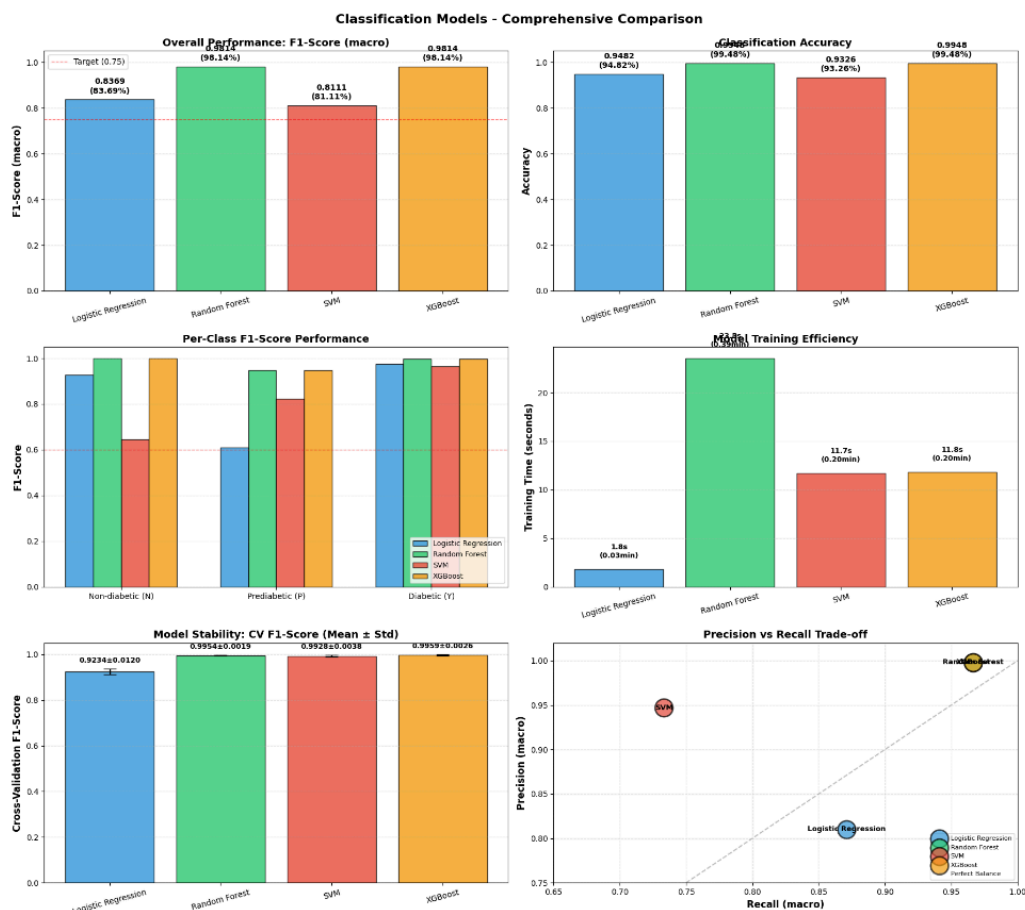
- This is the simplest model in the group, but still very useful. It trains quickly, gives interpretable coefficients, and serves as a good baseline to compare the other models against.

## 4.1.2 Classification Results

The comparison of different models with different metrics.

	Model	Accuracy	Precision (macro)	Recall (macro)	F1-Score (macro)	F1 (N)	F1 (P)	F1 (Y)	Training Time (s)	CV F1 Mean	CV F1 Std
0	Logistic Regression	0.948187	0.810215	0.871063	0.836893	0.926829	0.608696	0.975155	1.808946	0.923422	0.012045
1	Random Forest	0.994819	0.997967	0.966667	0.981437	1.000000	0.947368	0.996942	23.536123	0.995364	0.001925
2	SVM	0.932642	0.946840	0.733333	0.811063	0.645161	0.823529	0.964497	11.722343	0.992788	0.003782
3	XGBoost	0.994819	0.997967	0.966667	0.981437	1.000000	0.947368	0.996942	11.796157	0.995881	0.002626

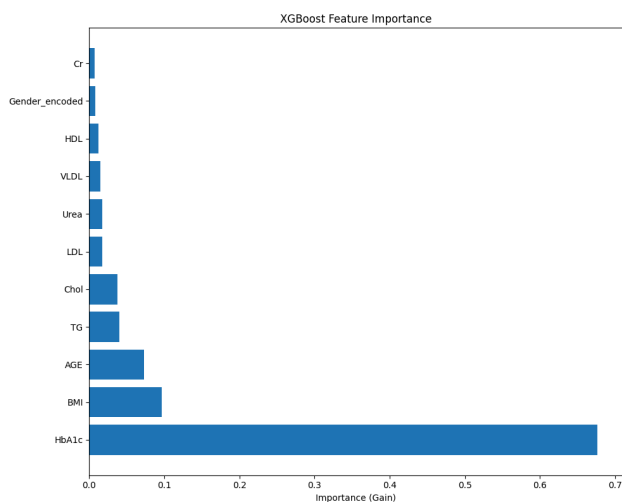
Here is the proper visualization chart:



The winner is XGBOOST.

#### 4.1.4 Feature Importance Analysis (XGBOOST)

After training the XGBOOST model, I looked at which features mattered the most for predicting diabetic status. And here is its chart



So here we can see that the HbA1 is the most important feature.

## 4.2 Clustering Models

### 4.2.1 Algorithm Comparison

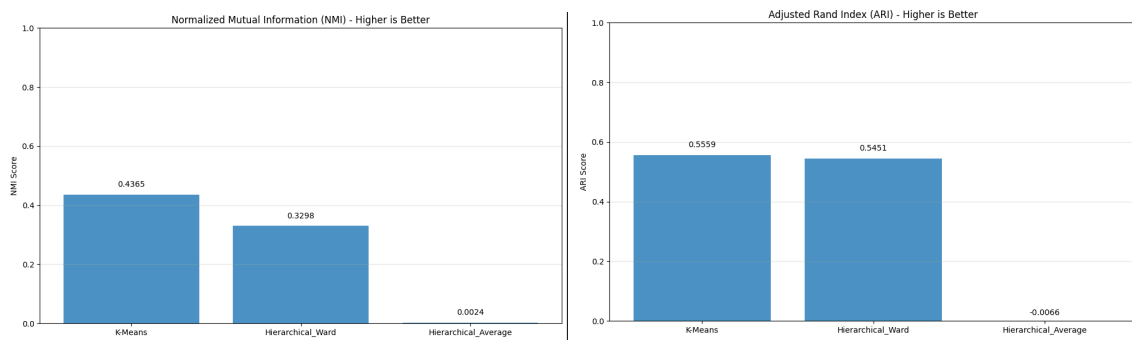
For clustering, I wanted to see if we could uncover natural groupings of patients without using the labels. I tested three unsupervised clustering methods.

1. K-Means Clustering
  - K-Means is a simple and fast algorithm. I set  $k = 3$  because we know there are three classes in the data. It works by minimizing the distances within each cluster, so patients in the same cluster are as similar as possible. It's very scalable and works well on large datasets.
2. Hierarchical Clustering (Ward Linkage)
  - This method builds clusters step by step by merging similar patients.
3. Hierarchical Clustering (Average Linkage)
  - This is another way to merge clusters, using the average distance between points.

### 4.2.2 Clustering Evaluation

After running the clustering algorithms, I evaluated their performance using several standard metrics. Here is the table showing its evaluation.

	Algorithm	ARI	Silhouette	Davies_Bouldin	NMI	Inertia
0	K-Means	0.555875	0.173068	1.586168	0.436468	6763.990175
1	Hierarchical_Ward	0.545081	0.129967	1.457560	0.329782	NaN
2	Hierarchical_Average	-0.006622	0.660185	0.229455	0.002371	NaN



Across various metrics k-means is performing better .

### 4.2.3 Model Comparison

Having identified XGBoost as the best-performing classification model and K-means as the optimal clustering algorithm, I now proceed to a comprehensive comparison between these two distinct machine learning approaches for diabetes prediction. To ensure a fair and unbiased evaluation, both models are tested on the same test dataset containing 193 data points, maintaining consistency in the evaluation framework.

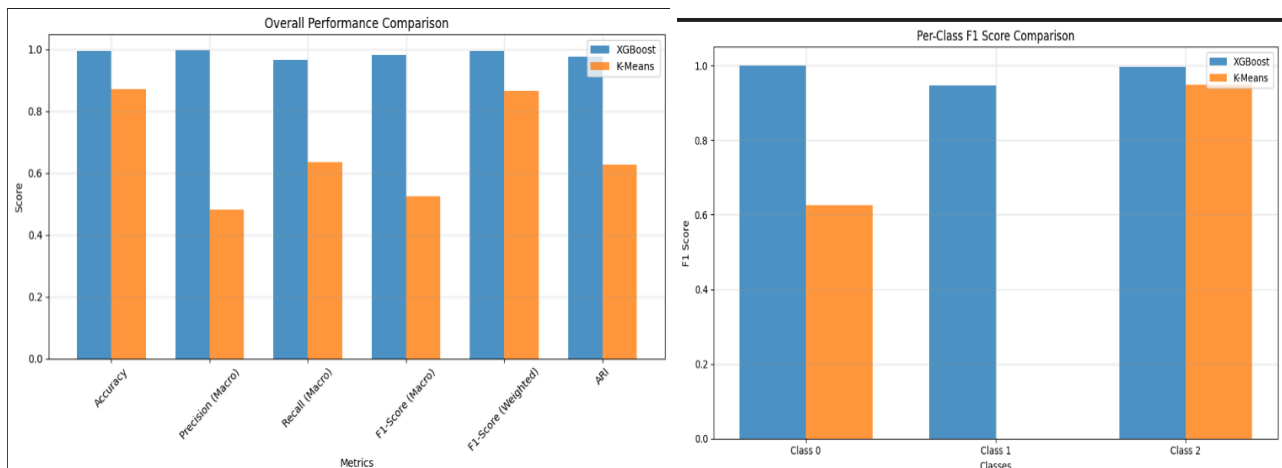
The K-means clustering model follows an unsupervised learning approach where we first train the model on the original training data (768 data points) to identify natural clusters, then create a cluster-to-label mapping based on the most frequent class within each cluster. This mapping is subsequently applied to the test set predictions to convert cluster assignments into class predictions. In contrast, XGBoost represents a supervised learning approach, directly trained on labeled data to learn the relationship between features and diabetes classes.

For this comparative analysis, I employ six key evaluation metrics that effectively capture different aspects of model performance: Accuracy (overall correctness), Precision (Macro) (average precision across all classes), Recall (Macro) (average sensitivity across classes), F1-Score (Macro) (harmonic mean of precision and recall), F1-Score (Weighted) (class-imbalance adjusted F1-score), and Adjusted Rand Index (ARI) (clustering quality measure that accounts for chance). These metrics provide a comprehensive assessment framework to determine which approach supervised classification or unsupervised clustering—is more effective for diabetes prediction in this experimental setting.

Here is the metric comparison table:

	Metric	XGBoost	K-Means Clustering	Difference
0	Accuracy	0.994819	0.870466	0.124352
1	Precision (Macro)	0.997967	0.482611	0.515356
2	Recall (Macro)	0.966667	0.635992	0.330675
3	F1-Score (Macro)	0.981437	0.524573	0.456864
4	F1-Score (Weighted)	0.994690	0.866016	0.128675
5	ARI	0.976483	0.628040	0.348443

Here is the overall performance comparison and F1 score per class comparison between two algorithms.



We can see that the clustering model struggles to form a distinct cluster for the Pre-diabetic (P) class. This makes sense because pre-diabetes is a transition phase between healthy and diabetic states, so the features of these patients overlap with both N and Y classes.

Overall, the classification model clearly outperforms the clustering approach for this dataset

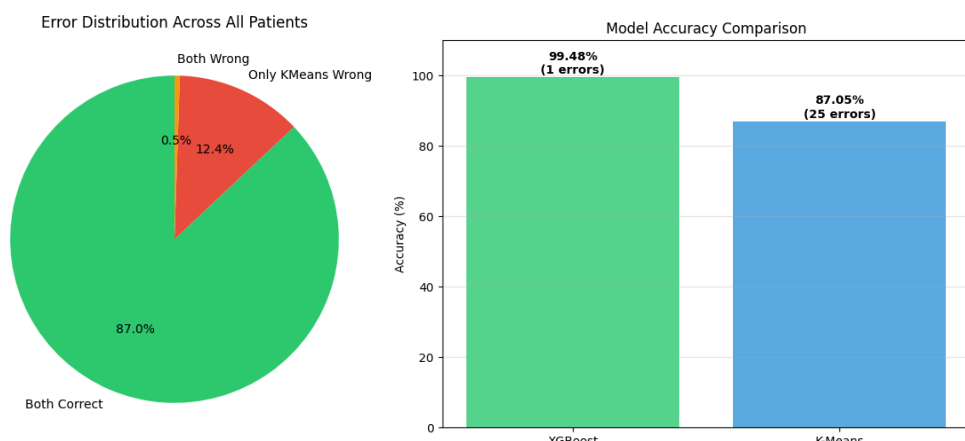
## Error Analysis

I performed a detailed error analysis to compare XGBoost and K-Means on the diabetes dataset. The goal was to understand where each model succeeds or fails and uncover patterns in their predictions.

- The analysis was conducted on 193 test cases.

Here are some visualizations:

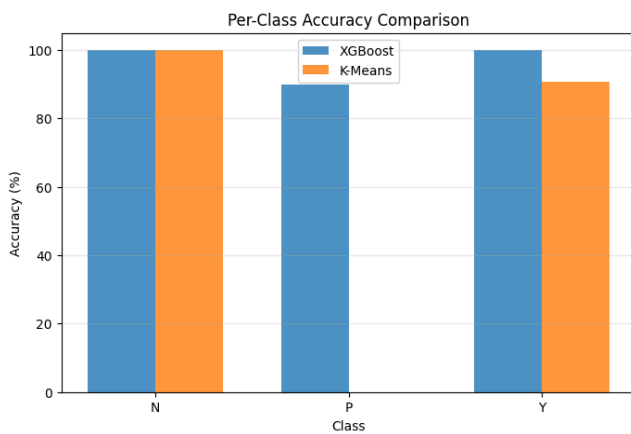
### Pie chart and Bar chart



By visualizing this plots and charts I observed

- XGBoost archives superior overall accuracy, handling most cases correctly.
- K-Means perform competitively in certain scenarios.

Per-class analysis:

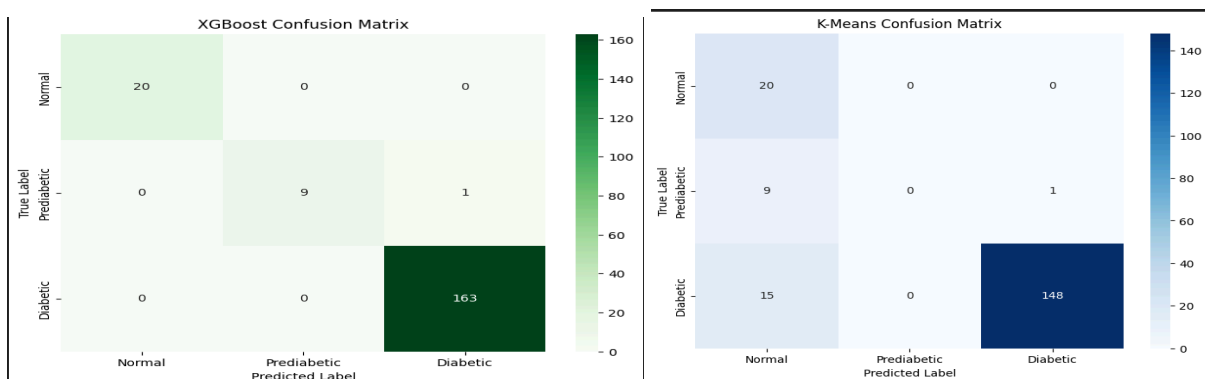


This shows model strengths vary across diabetes categories:

- Normal and diabetes classes are generally well predicted.
- Pre-diabetic(p) cases are more challenging, especially for clustering, due to their transitional nature .

Confusion metrics:

### XGBoost , k-Means



The confusion matrices give a **granular view of misclassifications**, showing which classes are commonly confused.

Overall, **XGBoost handles all three classes very well**, while **K-Means struggles with the Pre-diabetic class** due to its transitional nature

## Future Enhancements

### 1. Advanced Machine Learning

- Ensemble Methods: Combine XGBoost with Random Forest, SVM, or Neural Networks using stacking, voting, or blending to improve accuracy.
- Deep Learning: Build neural networks or transformer-based models to capture complex patterns in biomarkers.
- AutoML: Use AutoML tools (like AutoSklearn, H2O AutoML) for automated model selection and tuning.

## **2. Improve Data Quality**

- Expand Pre-diabetic Sample Size: Collect an additional 200–300 samples to improve model detection of at-risk individuals.
- Include More Patient Information: Record lifestyle data: exercise habits, diet, smoking status. Also include family history, parental or sibling diabetes occurrences.

## **Conclusion**

This experiment shows effective diabetes prediction using machine learning .Here I tested 4 supervised models and 3 unsupervised models.

The best supervised model XGBoost gives 99.48% accuracy while the best clustering model k-Means gives 87% accuracy .

Supervised models are best for accurate predictions , while clustering helps us understand patterns and groups in the data .