SC1015 MINI-PROJECT

ECDS Group 5

Xie Xiaomei U2430937E
Xie Xiaotian U2430437J

Table of contents

MORE THAN

800 million

PEOPLE ARE AFFECTED GLOBALLY

IN THE U.S. ALONE, UNDIAGNOSED DIABETES COSTS OVER

400 billion

USD ANNUALLY

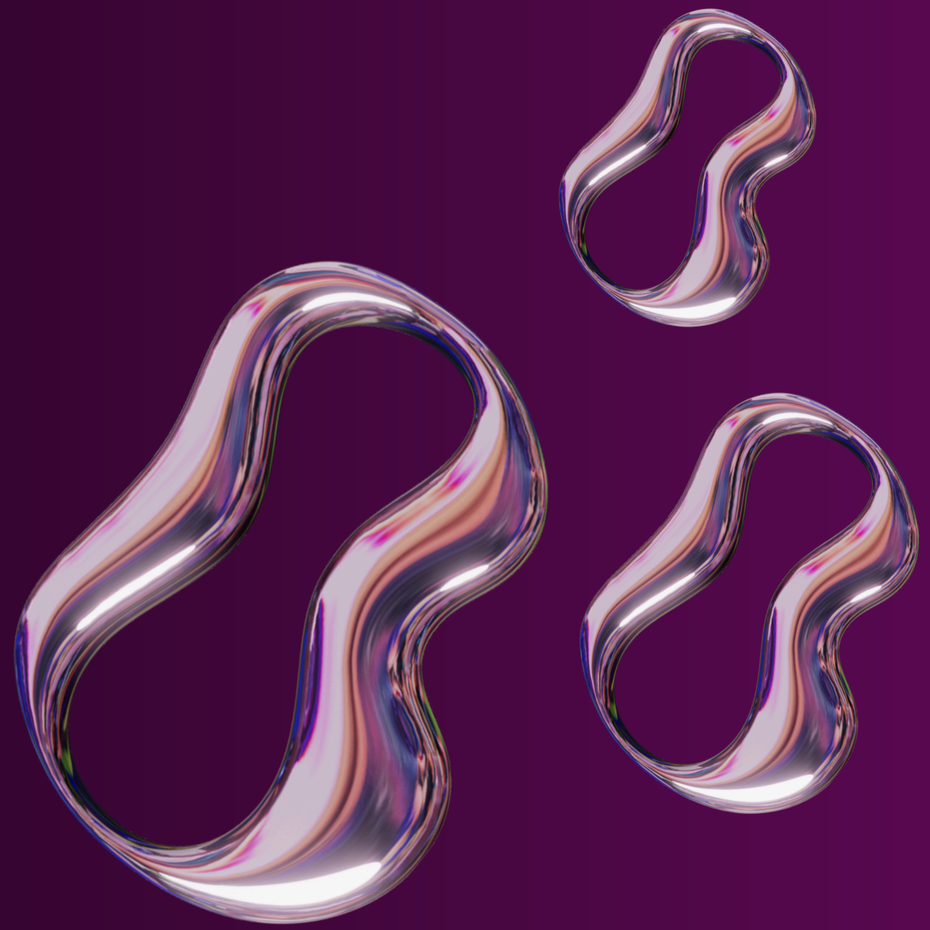
Problem

1. Despite medical advancements, millions—especially in underserved communities—don’t get screened in time.
2. Traditional healthcare systems lack the capacity to follow up with every at-risk individual.

But....

We realise that by identifying high-risk individuals sooner, we can:

- Prevent complications through timely treatment
- Reduce emergency care and long-term costs



Data Science Problem

Can we use machine learning to predict diabetes risk using routine clinical and lifestyle data?

Our Dataset

kaggle

Search

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Your Work

VIEWED

Diabetes prediction da...

Diabetes : EDA | ...

✅ Diabetes Predictio...

Diabetes+Hypertensio...

Global Cybersecuri...

MOHAMMED MUSTAFA · UPDATED 2 YEARS AGO

687 Code Download

Diabetes prediction dataset

A Comprehensive Dataset for Predicting Diabetes with Medical & Demographic Data

Data Card Code (309) Discussion (14) Suggestions (0)

About Dataset

The **Diabetes prediction dataset** is a collection of medical and demographic data from patients, along with their diabetes status (positive or negative). The data includes features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. This dataset can be used to build machine learning models to predict diabetes in patients based on their medical history and demographic information. This can be useful for healthcare professionals in identifying patients who may be at risk of developing diabetes and in developing personalized treatment plans. Additionally, the dataset can be used by researchers to explore the relationships between various medical and demographic factors and the likelihood of developing diabetes.

Usability
10.00

License
Data files © Original Authors

Expected update frequency
Never

Tags

Health Diabetes Classification Healthcare Binary Classification

Our Dataset

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0

Steps

- Drop duplicate rows
- Choose relevant columns
- Remove rows that had “Other” for gender
- One-hot encode categorical data (LEARNT!!)

Number of duplicate rows

```
duplicate_rows_data = diabetesData[diabetesData.duplicated()]
print("no. of duplicate rows: ", duplicate_rows_data.shape)
```

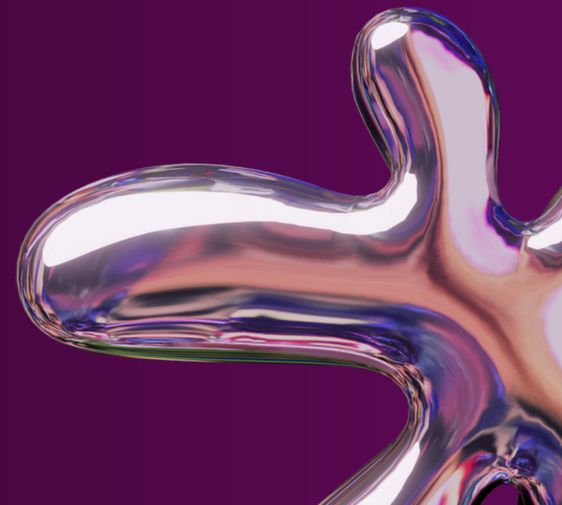
```
no. of duplicate rows:  (3854, 9)
```

(rows, columns) after dropping duplicate rows

```
diabetesData = diabetesData.drop_duplicates()
diabetesData.shape
```

```
(96146, 9)
```

- **Avoid Data Bias**: Duplicate entries can artificially inflate the importance of certain data points.
- **Better Evaluation Metrics**: Duplicates can inflate performance metrics (like accuracy or recall) by repeating easy-to-predict instances.



Drop smoking_history

smoking_history
never
No Info
never
current
current
never
never
No Info
never
never
never
former
former
never
No Info
No Info

```
diabetesData = diabetesData.drop("smoking_history",axis=1)
```

Checking for missing data

```
diabetesData.isnull().sum()
```

```
gender                 0
age                    0
hypertension           0
heart_disease          0
bmi                    0
HbA1c_level            0
blood_glucose_level    0
diabetes               0
dtype: int64
```

There are no missing data in this dataset.

Data Cleaning

Remove Rows in “Gender”

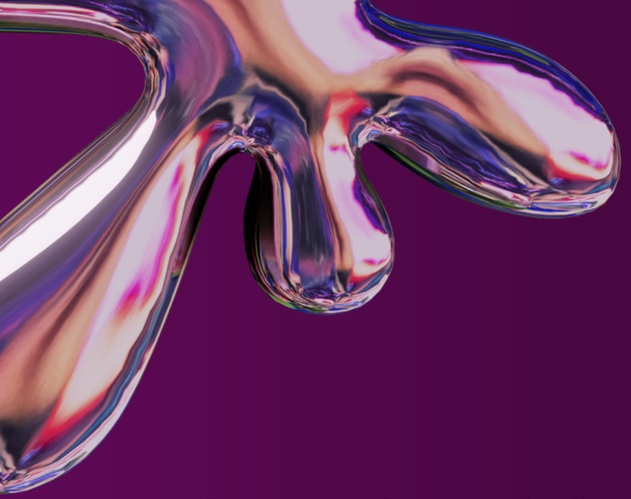
```
gender: 3 distinct values
```

```
diabetesData = diabetesData[diabetesData['gender'] != 'Other']
```

```
for column in diabetesData.columns:
    num_distinct_values = len(diabetesData[column].unique())
    print(f"{column}: {num_distinct_values} distinct values")
```

```
gender: 2 distinct values
age: 102 distinct values
hypertension: 2 distinct values
heart_disease: 2 distinct values
bmi: 4247 distinct values
HbA1c_level: 18 distinct values
blood_glucose_level: 18 distinct values
diabetes: 2 distinct values
```

Remove rows with gender entries labeled 'Other'



Data Cleaning One-Hot encoding

One-hot encode categorical variable

- One-hot encoding is a technique used to convert categorical variables into a numerical format without introducing unintended ordinal relationships

Dataframe with encoded categorical variables

```
# Check the updated dataframe
print(diabetesData.head())
```

```
    age  hypertension  heart_disease    bmi  HbA1c_level  blood_glucose_level  \
0  80.0             0              1  25.19          6.6                  140
1  54.0             0              0  27.32          6.6                   80
2  28.0             0              0  27.32          5.7                  158
3  36.0             0              0  23.45          5.0                  155
4  76.0             1              1  20.14          4.8                  155

   diabetes  gender_Male
0         0          0.0
1         0          0.0
2         0          1.0
3         0          0.0
4         0          1.0
```

Newly Learnt!!!

Exploratory Data Analysis

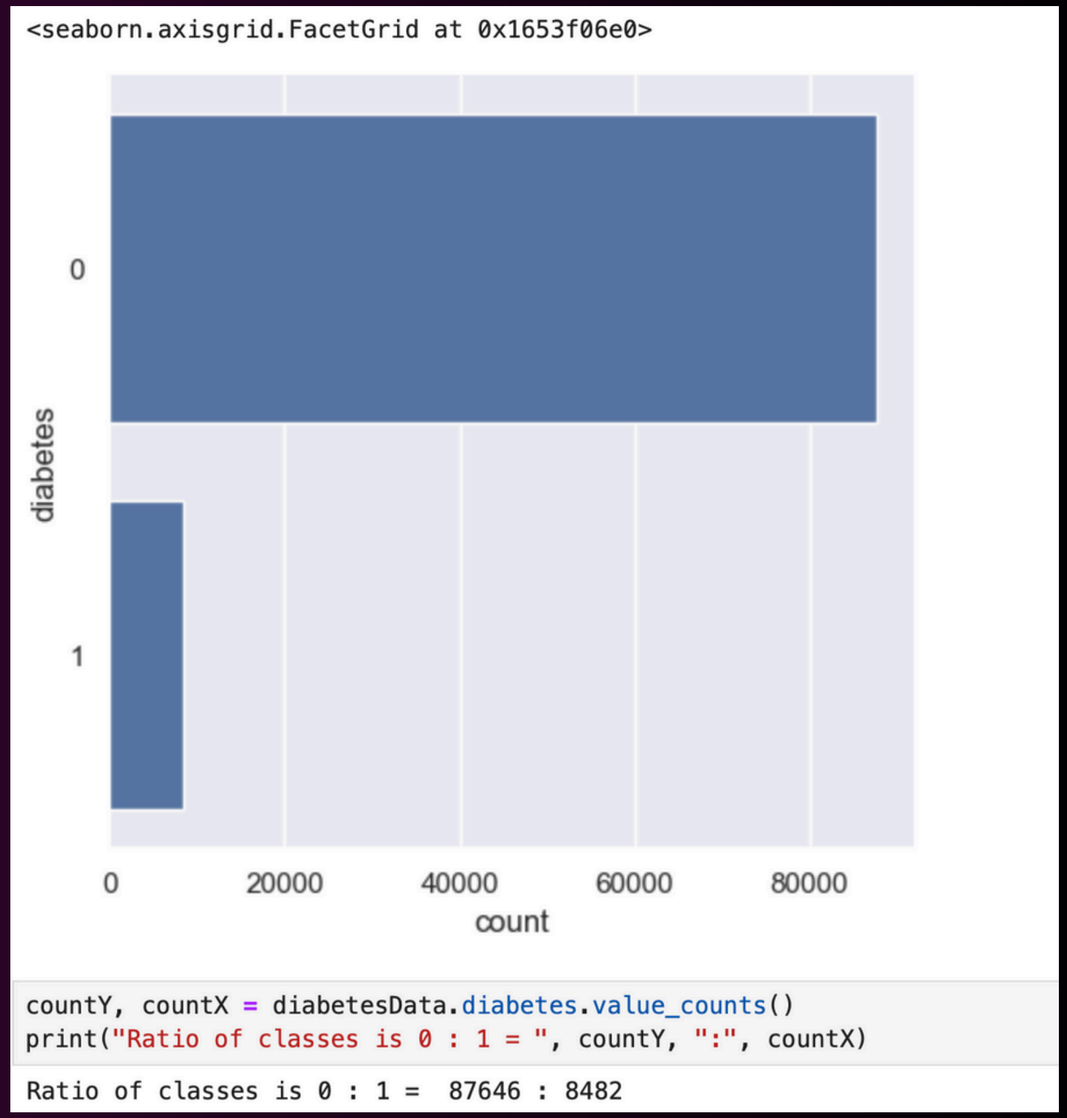
- Initial Observations
- Univariate Data Analysis
- Bivariate Data Analysis
- Multivariate Data Analysis

EDA

Initial Observations

ONLY

8.8% of patients have diabetes—highlighting **class imbalance**, hence we might require **resampling.**



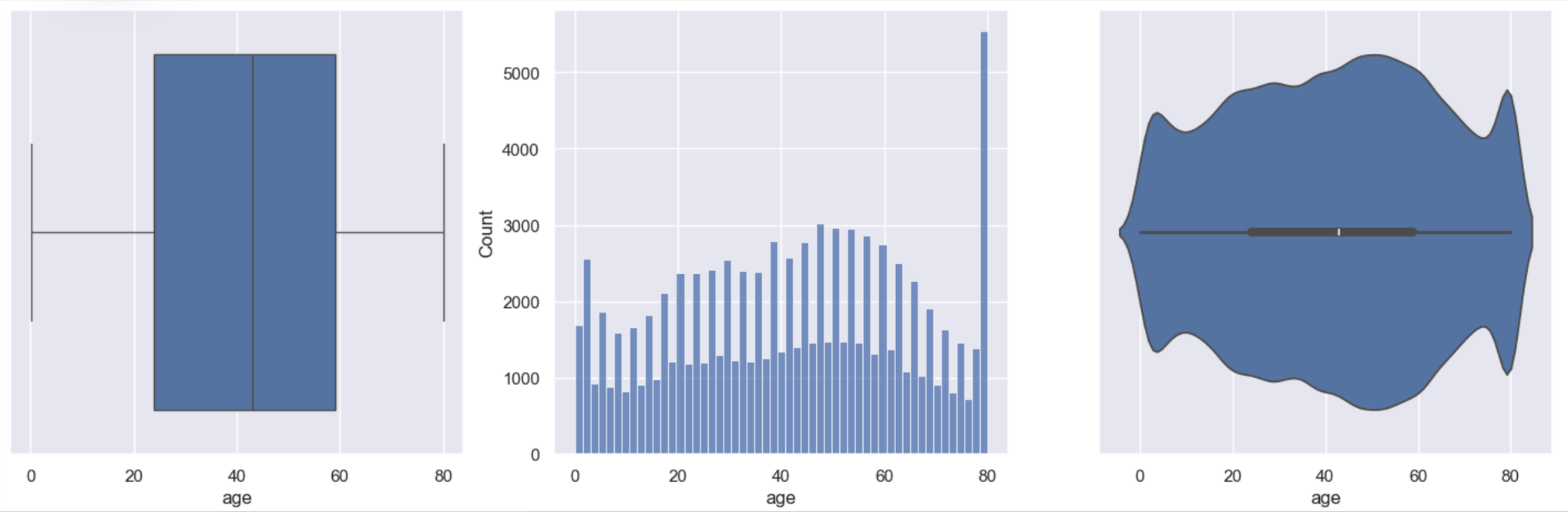

```
countY, countX = diabetesData.diabetes.value_counts()
print("Ratio of classes is 0 : 1 = ", countY, ":", countX)
```

```
Ratio of classes is 0 : 1 =  87646 : 8482
```

Because of this

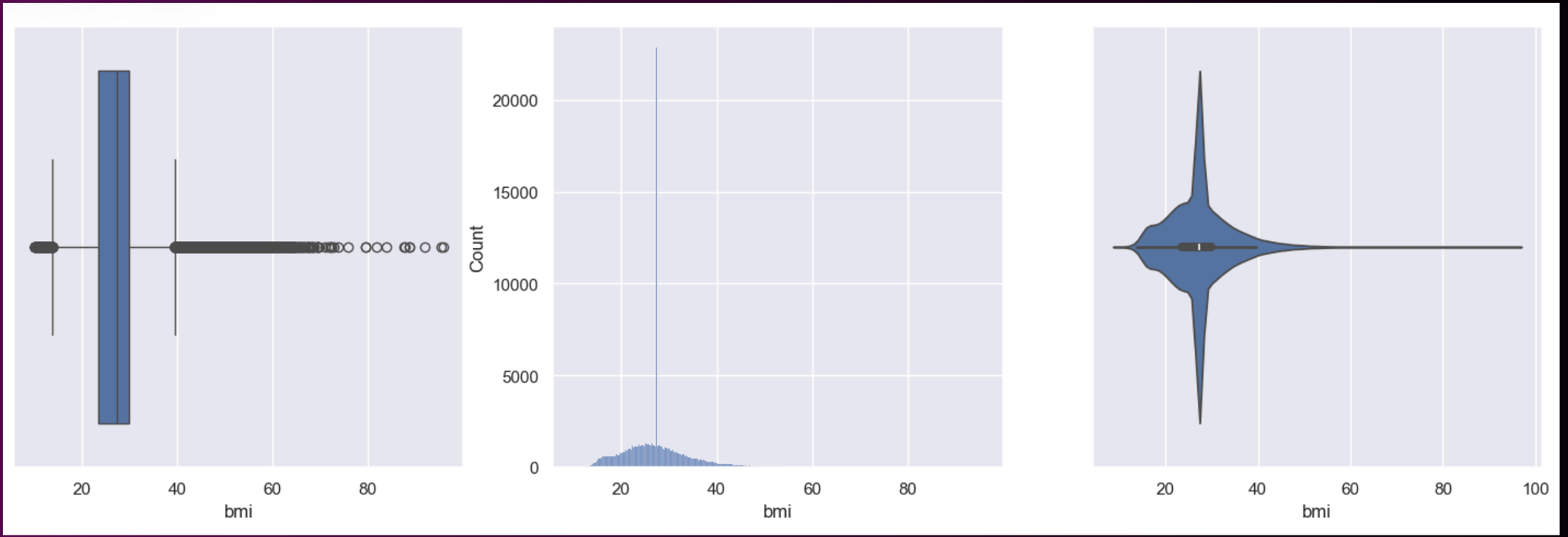
Plan to use metrics like **F1-score** , **precision** and **recall**, not just accuracy.

Age



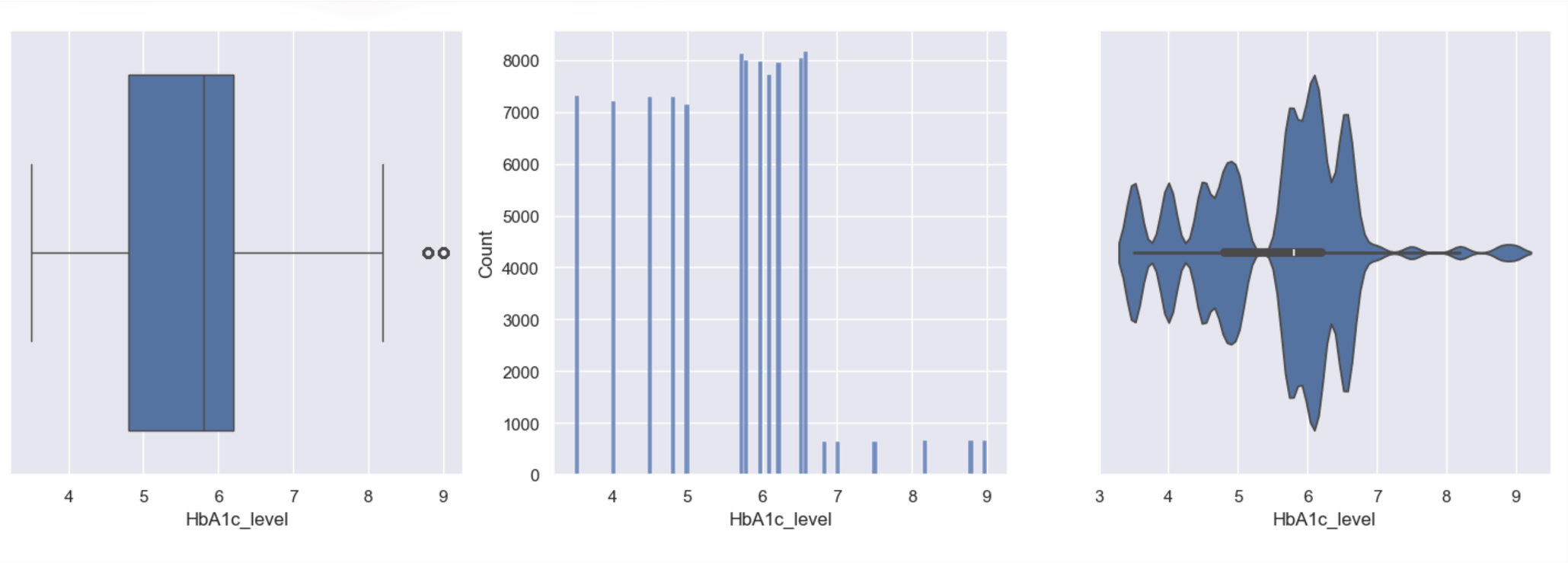
Age is symmetrically distributed around 40, suggesting a broad age range.

BMI



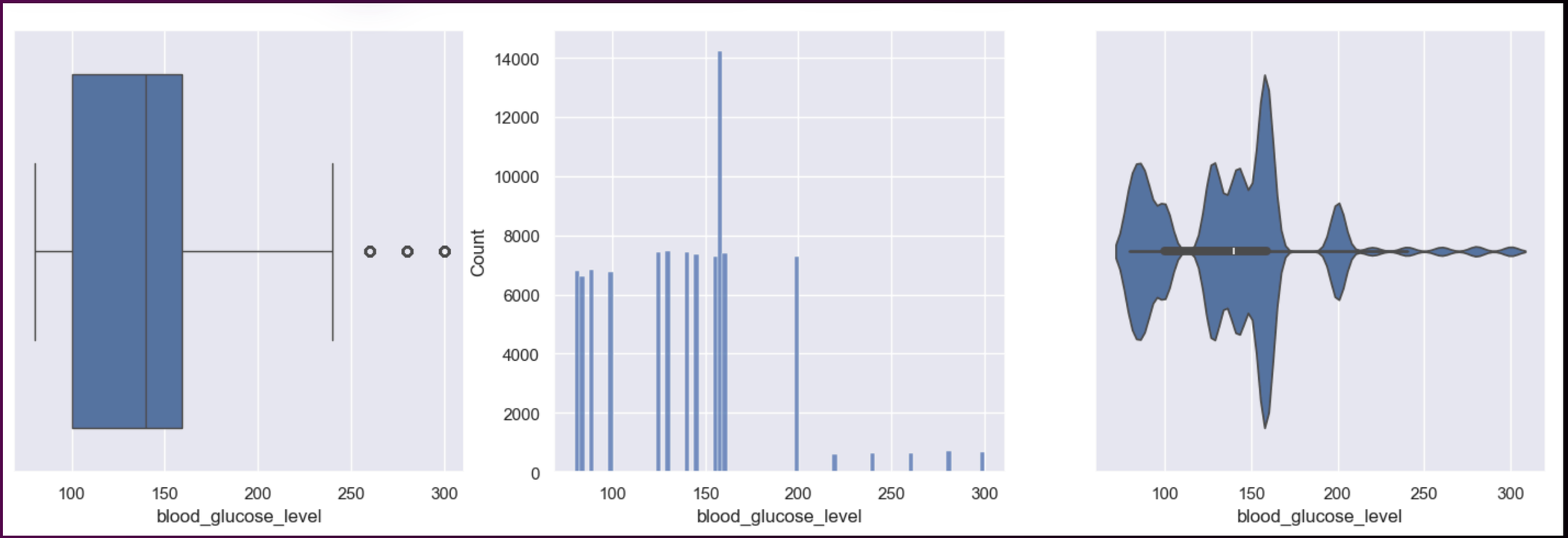
BMI is right-skewed, with many individuals in the overweight or obese range—and some extreme outliers.

HbA1c_level



HbA1c, a measure of long-term blood sugar, ranges mostly between 5.5 and 6.5, but a heavy tail beyond 8

Blood glucose level

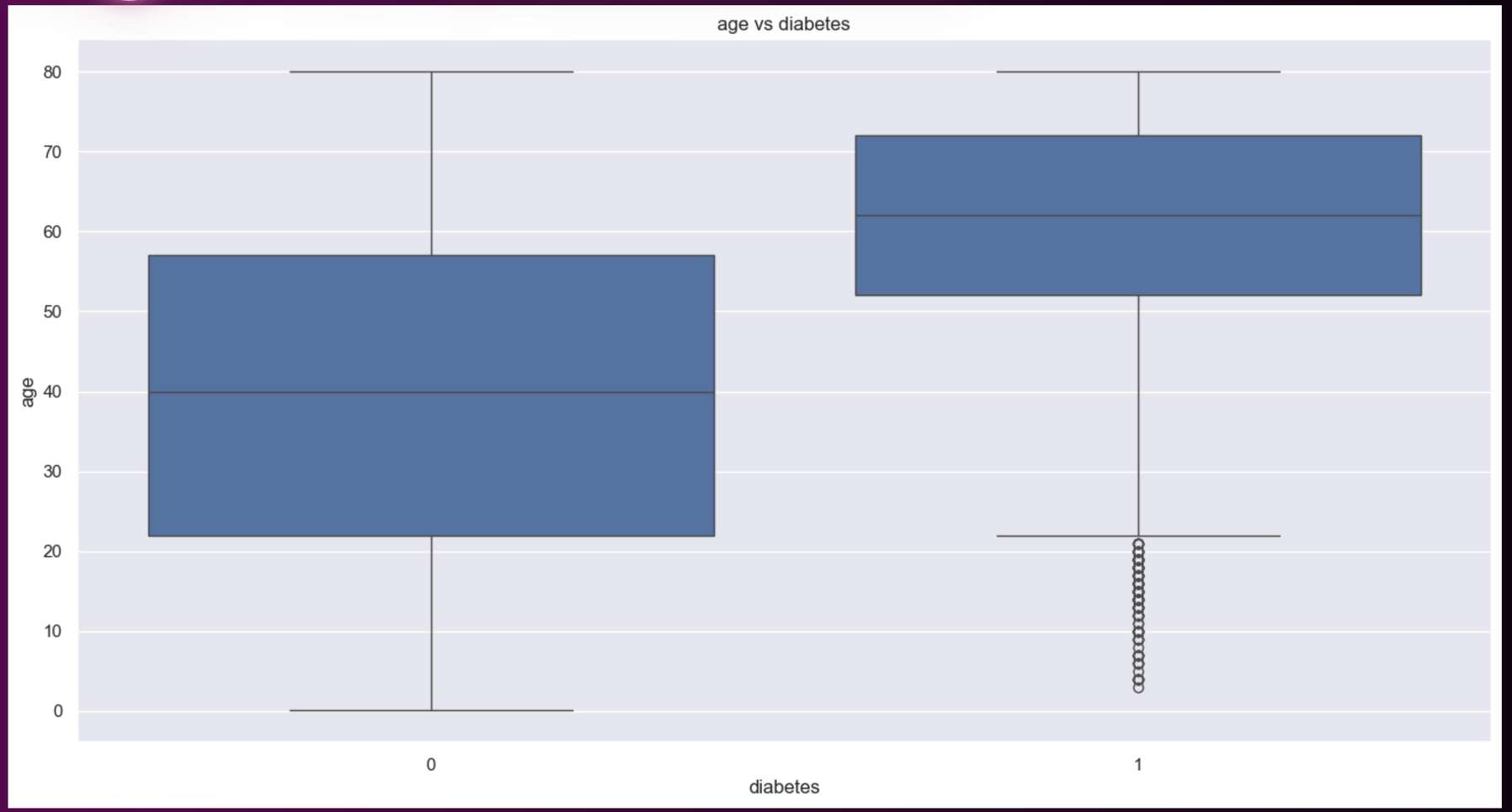


Blood glucose peaks around 150 mg/dL but includes high outliers.

Outliers?

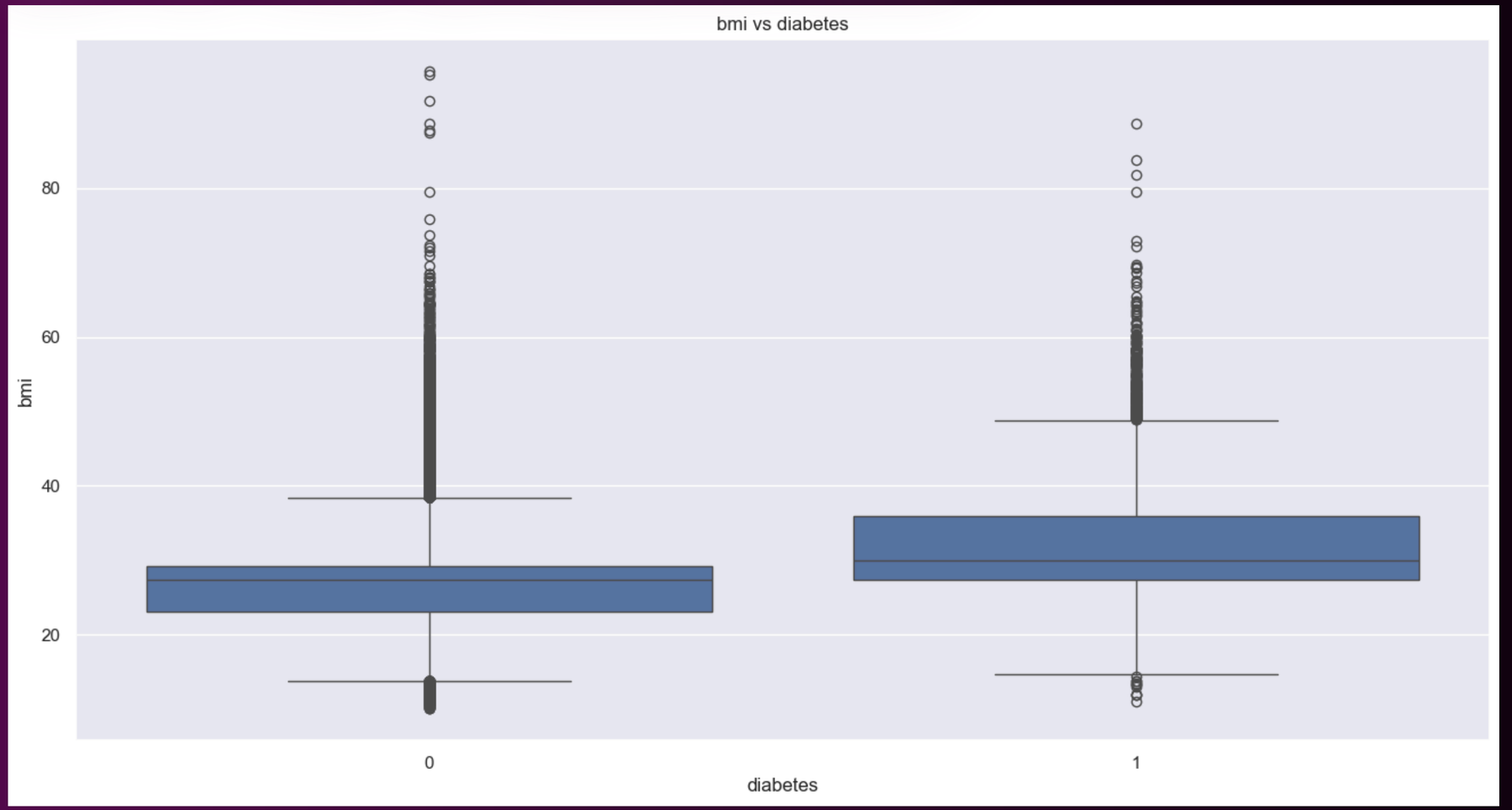
- In **medical data**, outliers reflect **important clinical conditions**
- Removing such values could reduce the model's ability to **detect severe case.**

Age vs Diabetes



The median age for diabetics is **63**, while for non-diabetics, it's around **40**.

BMI vs Diabetes

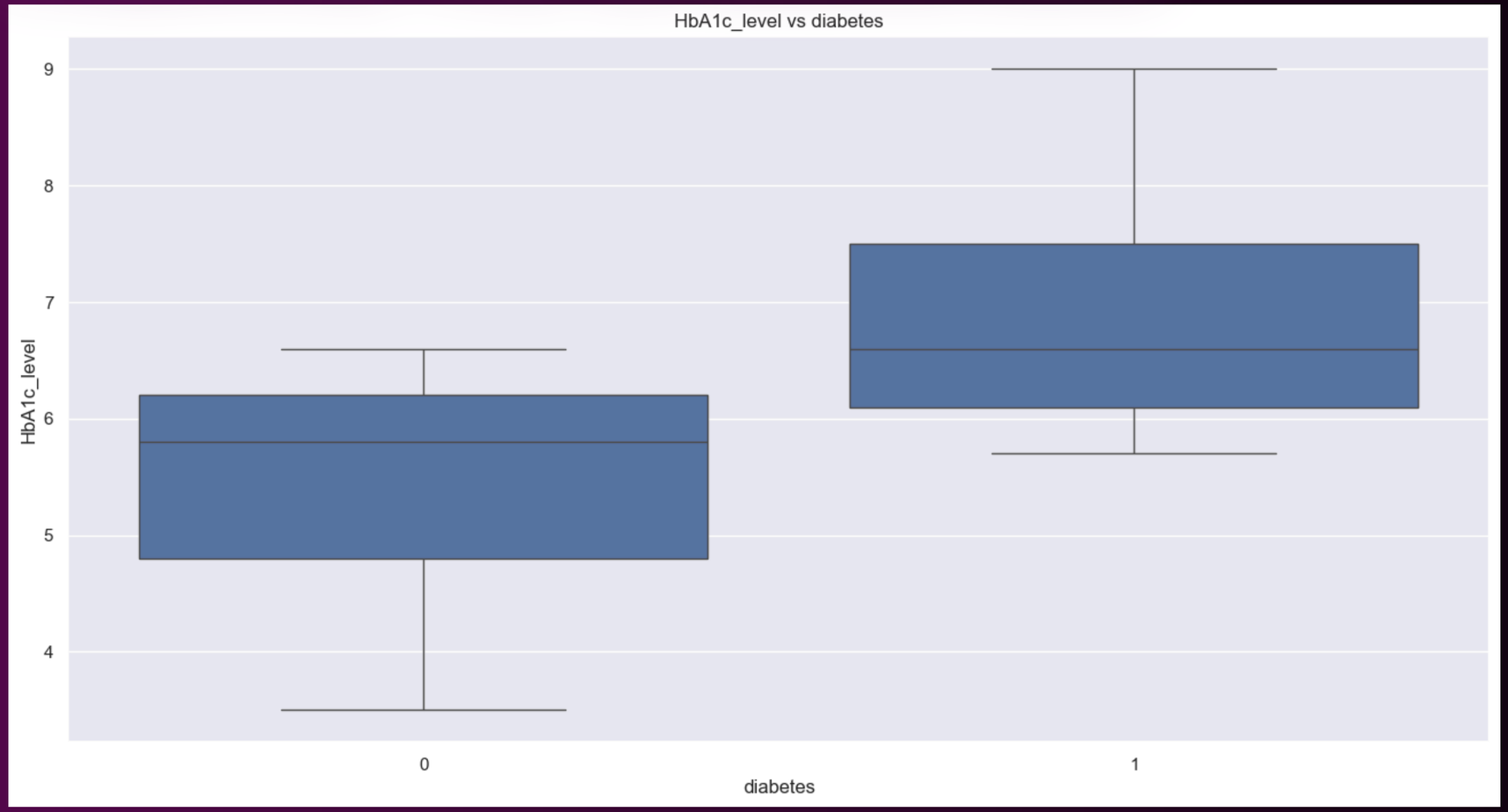


Diabetics tend to have **higher BMI**, but there's a lot of overlap between groups.

EDA

Bivariate Data Analysis

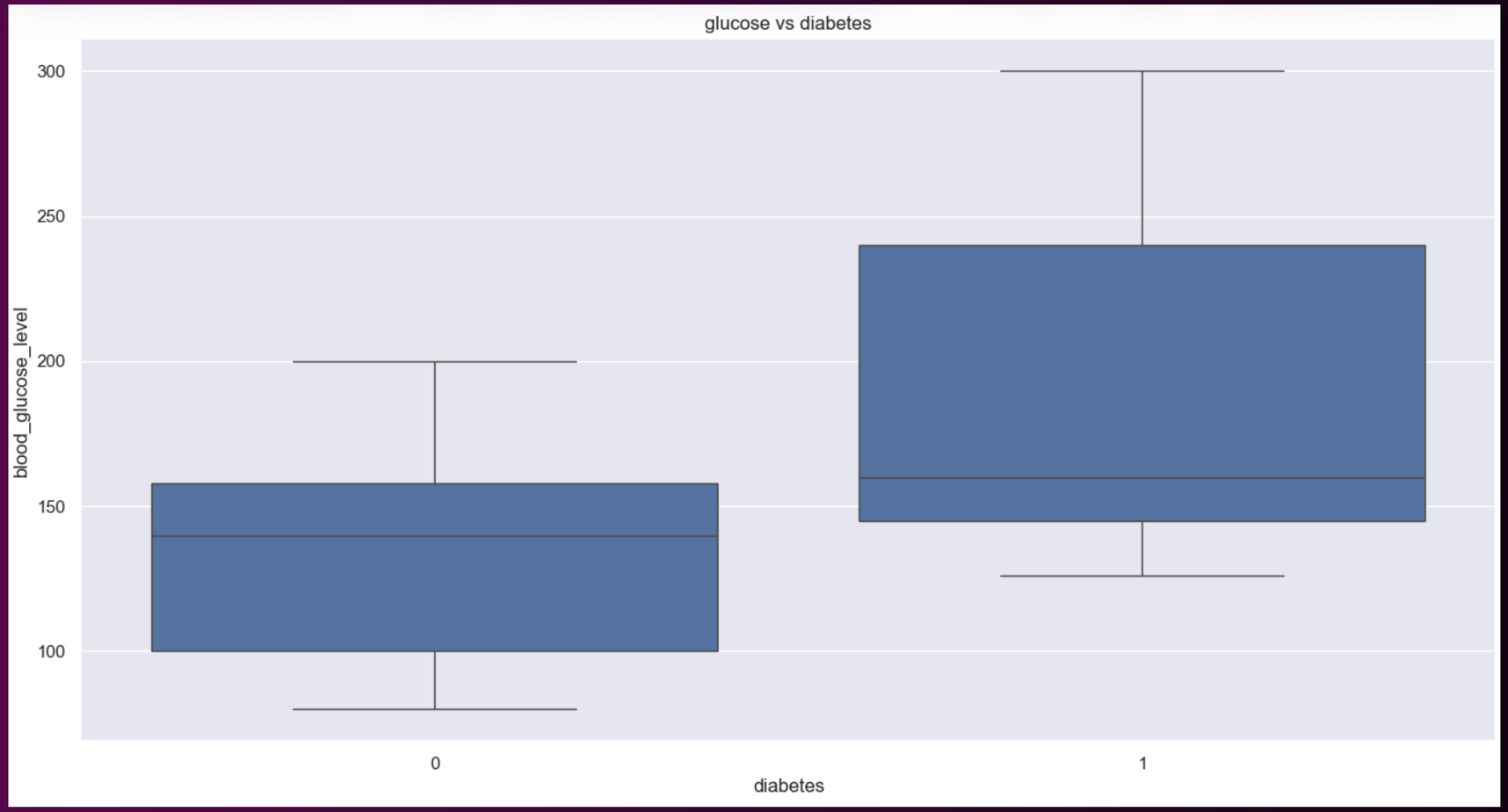
HbA1c vs Diabetes



Diabetics usually score well above 6.5

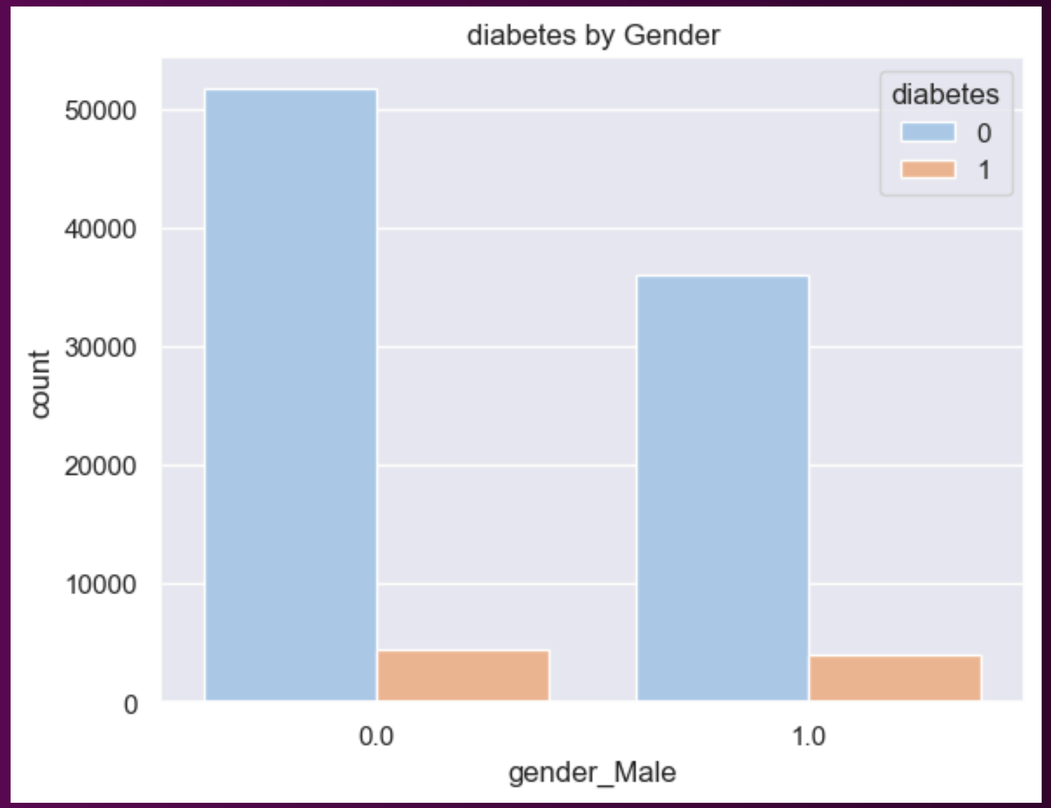
EDA Bivariate Data Analysis

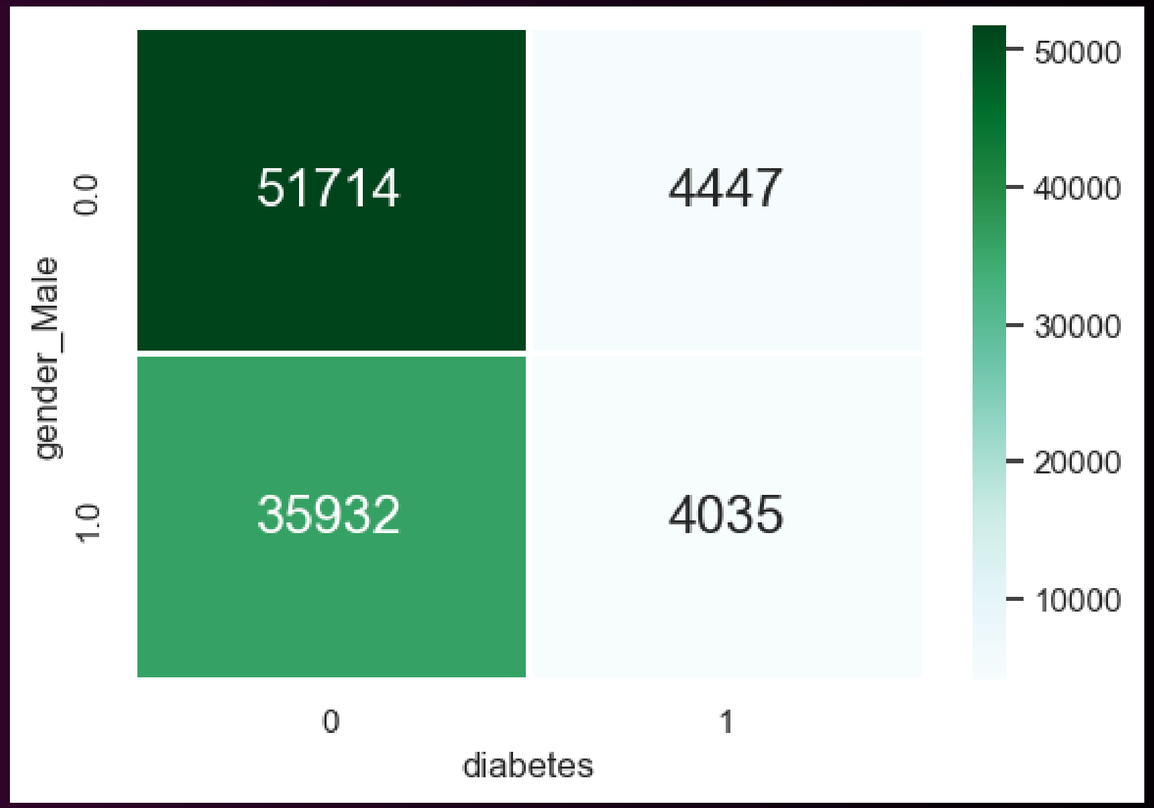
Blood Glucose vs Diabetes



Diabetics tend to have higher and more variable levels.

Gender vs Diabetes

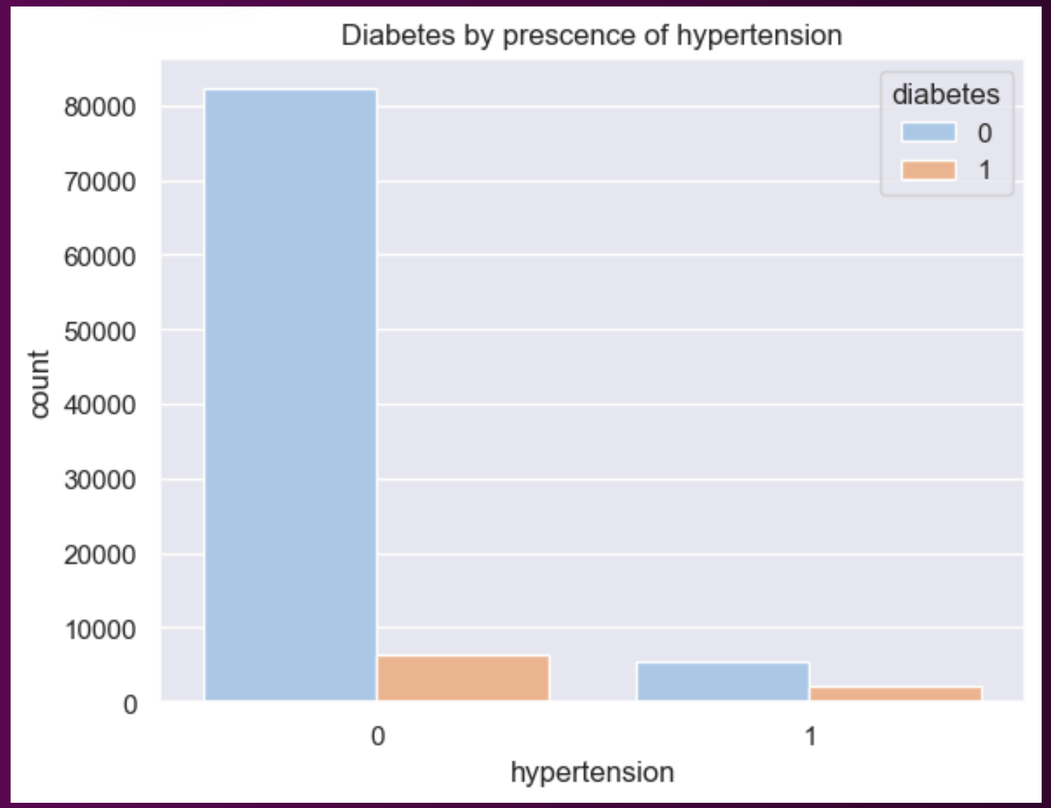


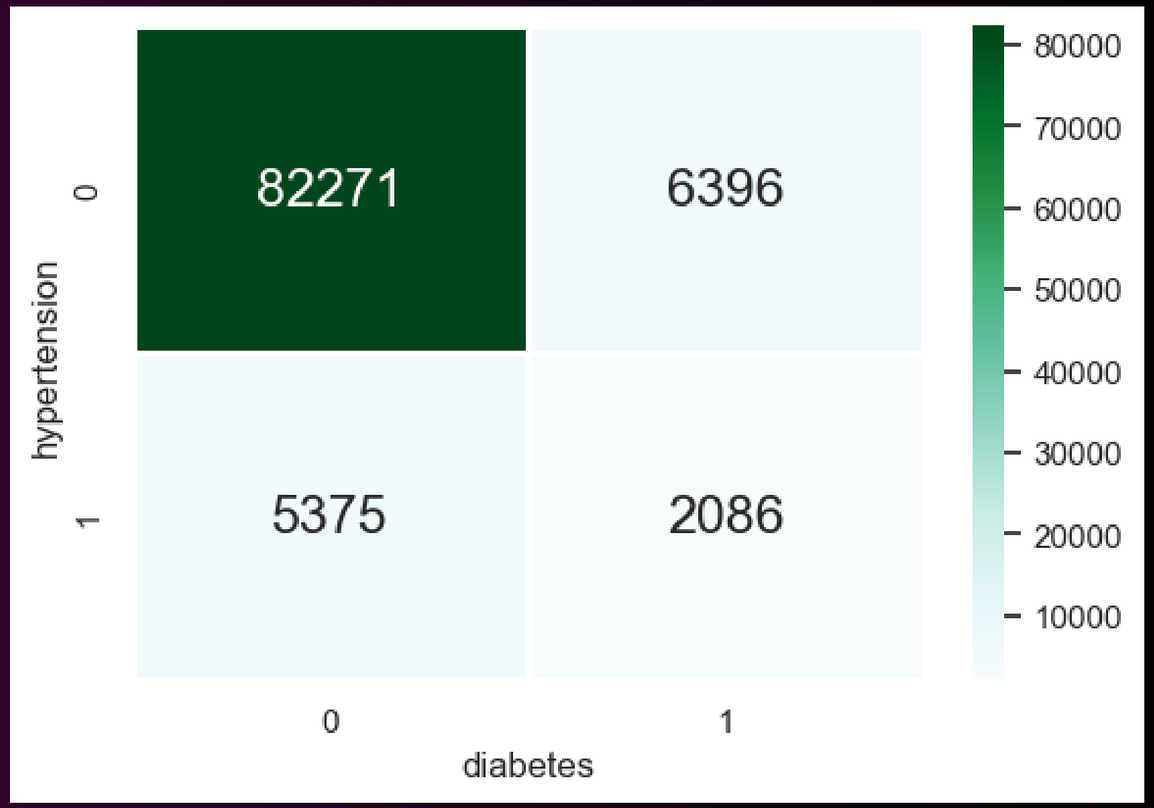


diabetes	0	1
gender_Male		
0.0	92.081694	7.918306
1.0	89.904171	10.095829

The diabetes rate is higher among males—10.1% compared to 7.9% for females.

Hypertension vs Diabetes

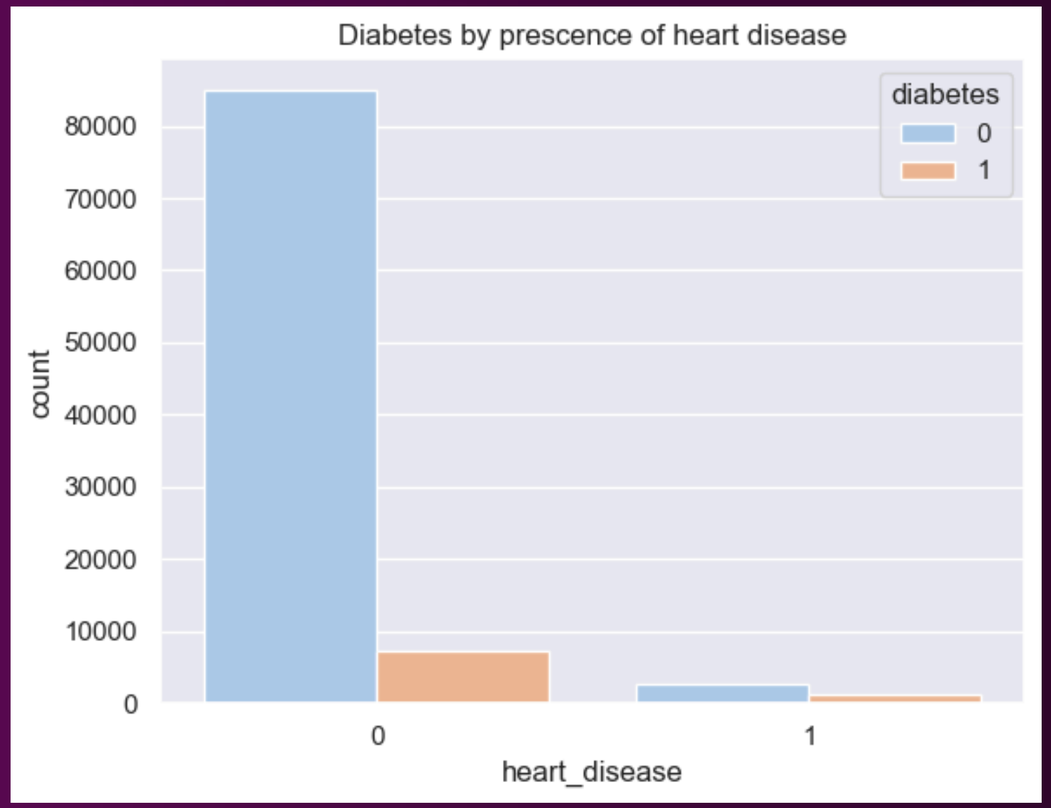


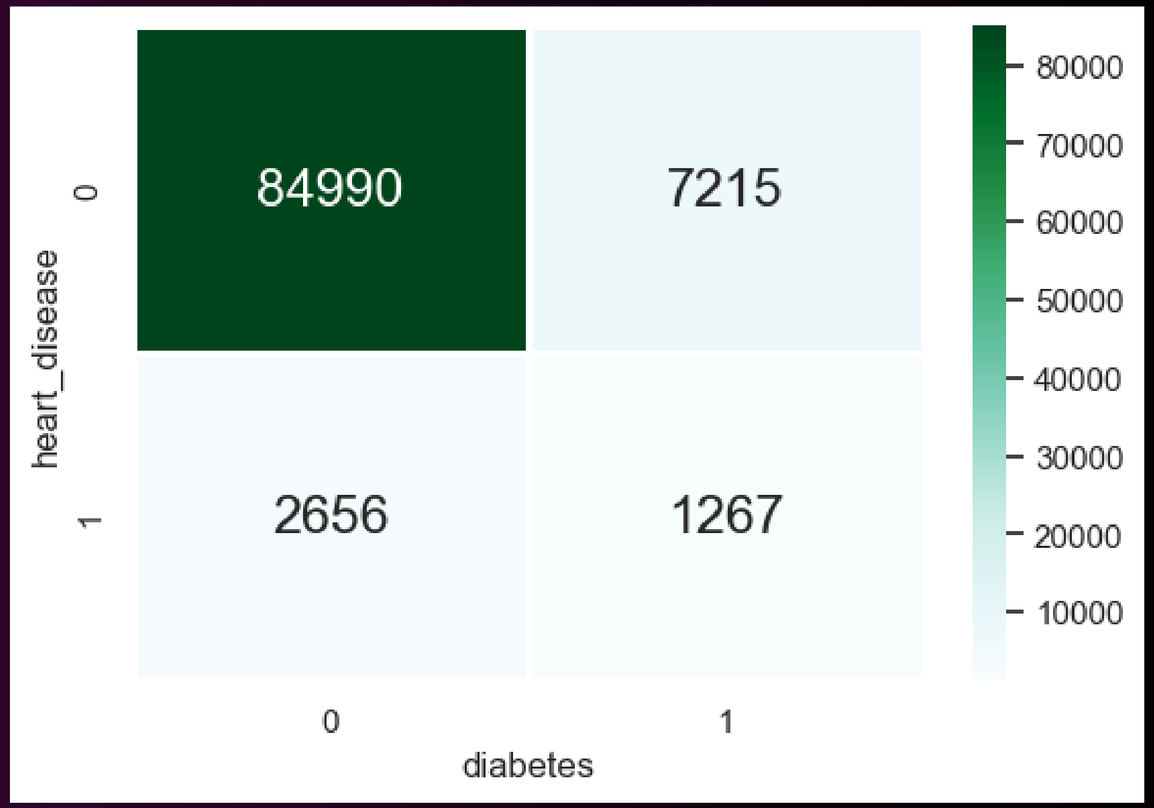


diabetes	0	1
hypertension		
0	92.786493	7.213507
1	72.041281	27.958719

Even though most diabetics don't have hypertension, having hypertension appears to significantly increase diabetes risk.

Heart disease vs Diabetes



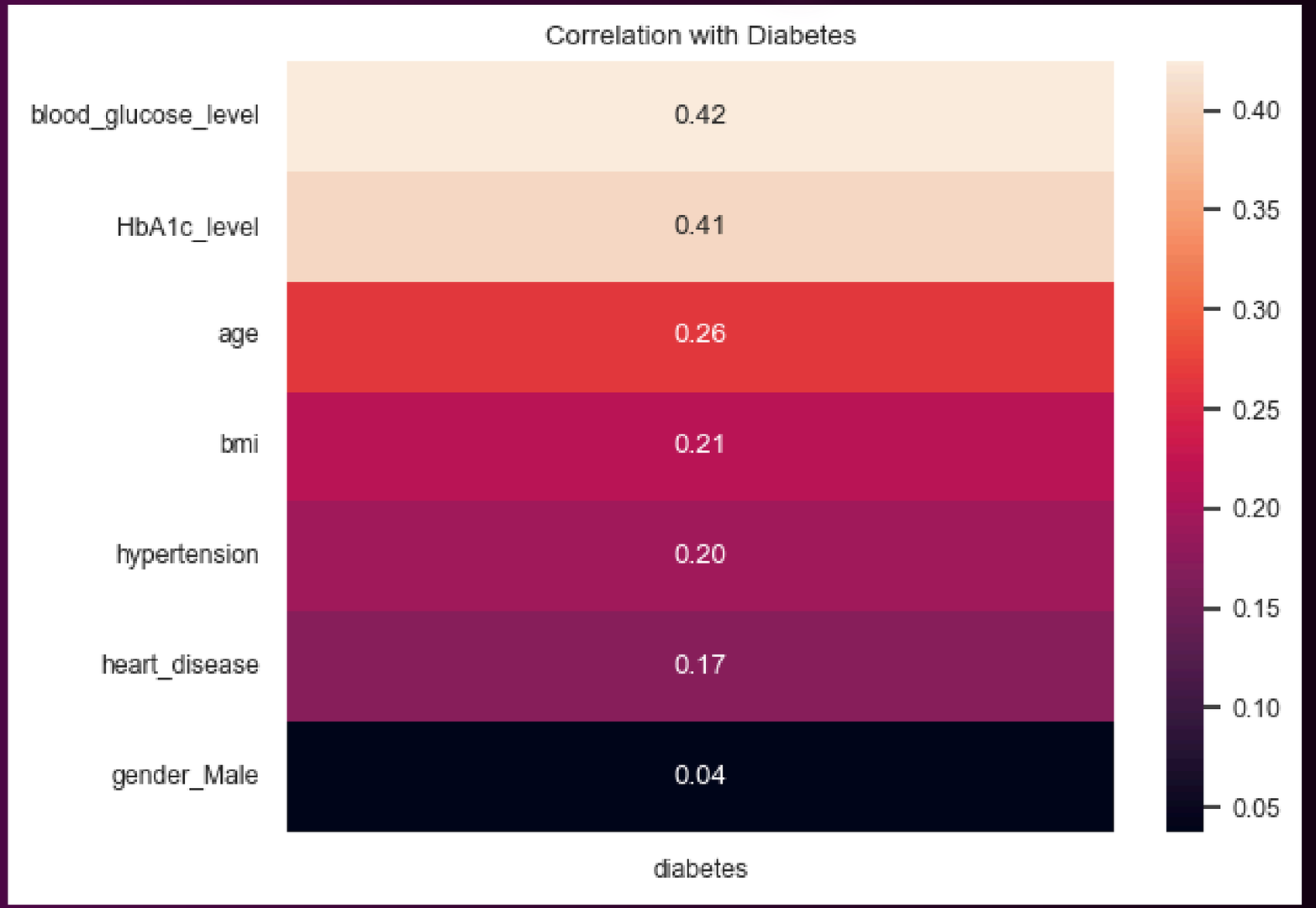


diabetes	0	1
heart_disease		
0	92.175045	7.824955
1	67.703288	32.296712

Those with heart disease show a **32.3% diabetes rate**, versus 7.8% for others

Correlation Matrix

Correlation Ranking



Data Preparation Resampling

```
Original class distribution in the training set:
diabetes
0    70121
1     6781
Name: count, dtype: int64
```

Resampling of train data

- SMOTE (Synthetic Minority Over-sampling Technique)
- Random Undersampler

```
Original training set size: 76902
Resampled training set size: 51087
Resampled class distribution:
diabetes
0    30051
1    21036
Name: count, dtype: int64
```

Newly Learnt!!!

Machine Learning

1. Multi-variate Decision Tree	course material
2. Logistic Regression	newly learnt
3. Random Forest (default hyperparameter)	newly learnt
4. Random Forest (tuned hyperparameter)	newly learnt

Machine Learning Evaluation metrics

Evaluation Metrics with Clinical Context

Recall

- important as missing actual diabetes cases could delay critical care

Precision

- important as false alarms waste resources and cause patient stress

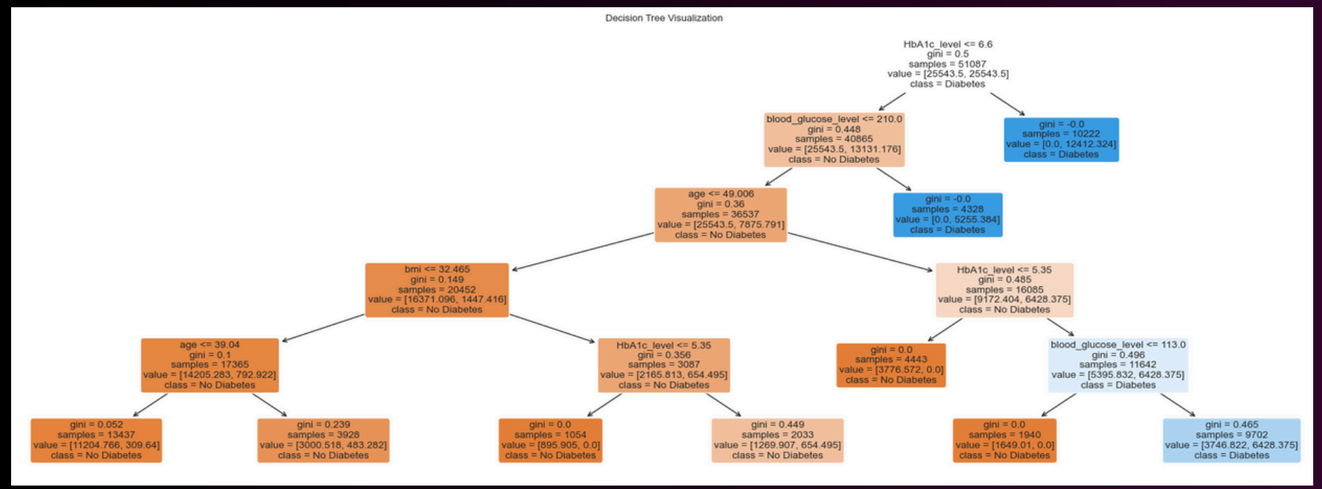
F1 score

- balances recall and precision

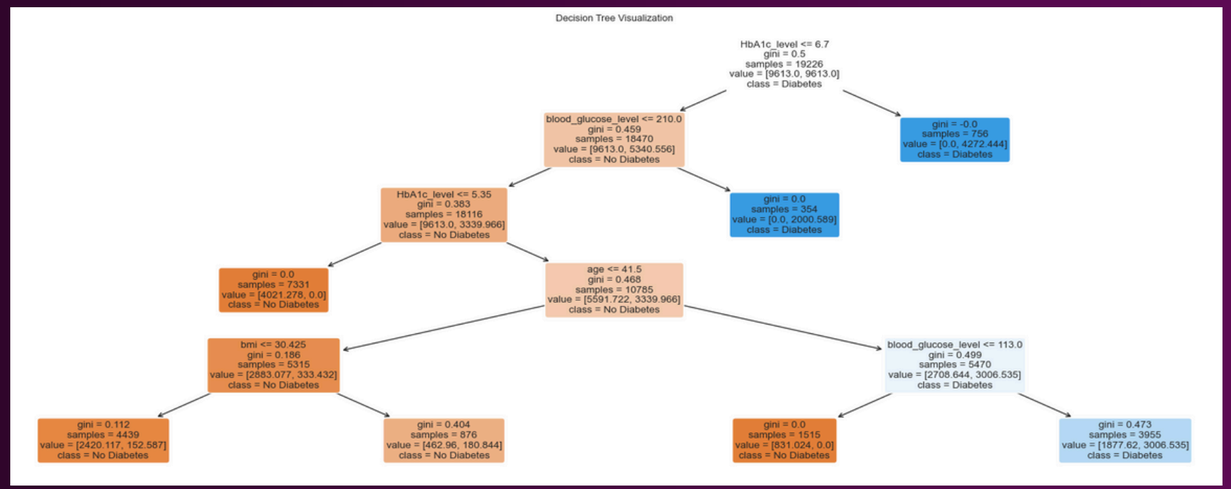
Machine Learning Classification Tree

Multi-variate Classification Tree

Train Data



Test Data

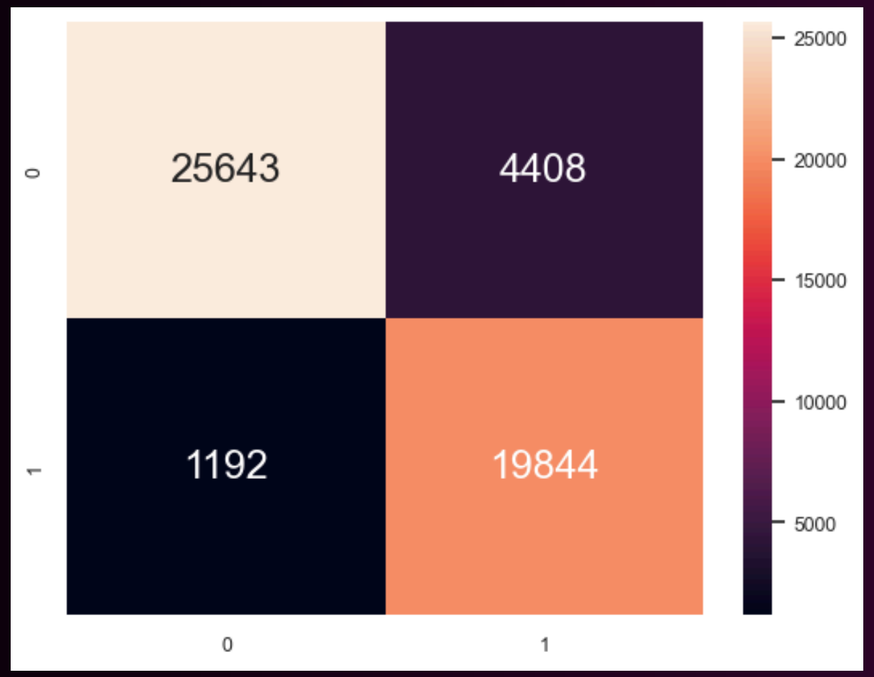


Multi-variate Classification Tree

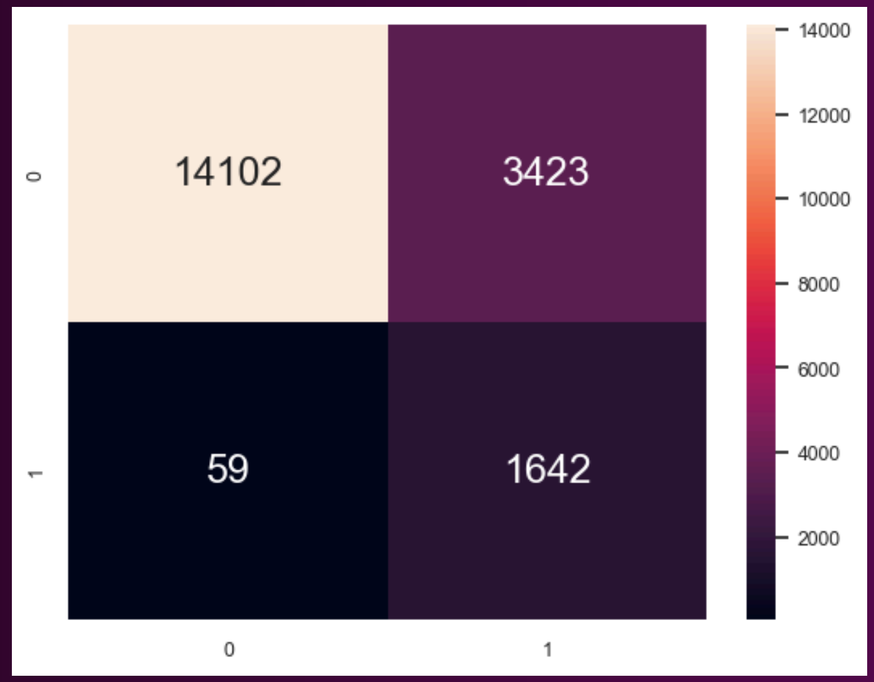
Train Data

Accuracy : 0.890
TPR Train : 0.943
TNR Train : 0.853
FPR Train : 0.147
FNR Train : 0.057

Precision : 0.818
Recall : 0.943
F1 Score : 0.876

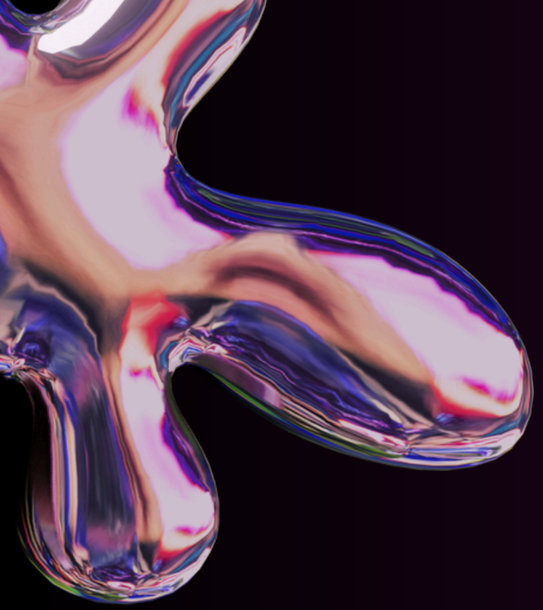


Test Data



Accuracy : 0.819
TPR Test : 0.965
TNR Test : 0.805
FPR Test : 0.195
FNR Test : 0.035

Precision : 0.324
Recall : 0.965
F1 Score : 0.485



Multi-variate Classification Tree

Train Data		Test Data	
Precision :	0.818241794491176	Precision :	0.32418558736426456
Recall :	0.9433352348355201	Recall :	0.9653145208700764
F1 Score :	0.8763469351704646	F1 Score :	0.485368016553355

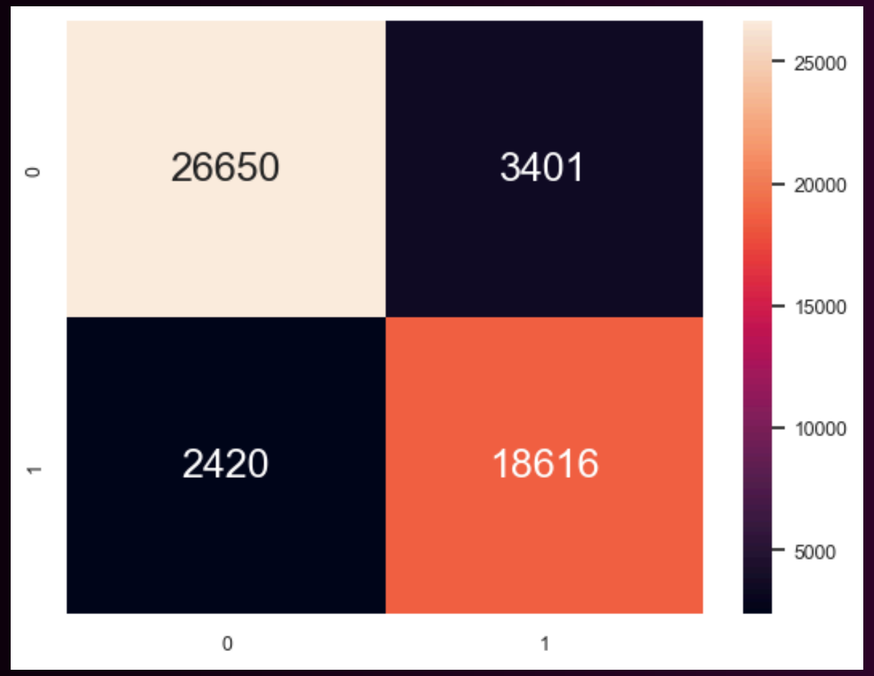
- It flagged nearly 70 percent of the patients as diabetic even though they are not
- Floods the system with **false alarms**, predicting diabetes in many healthy patients

Logistic Regression

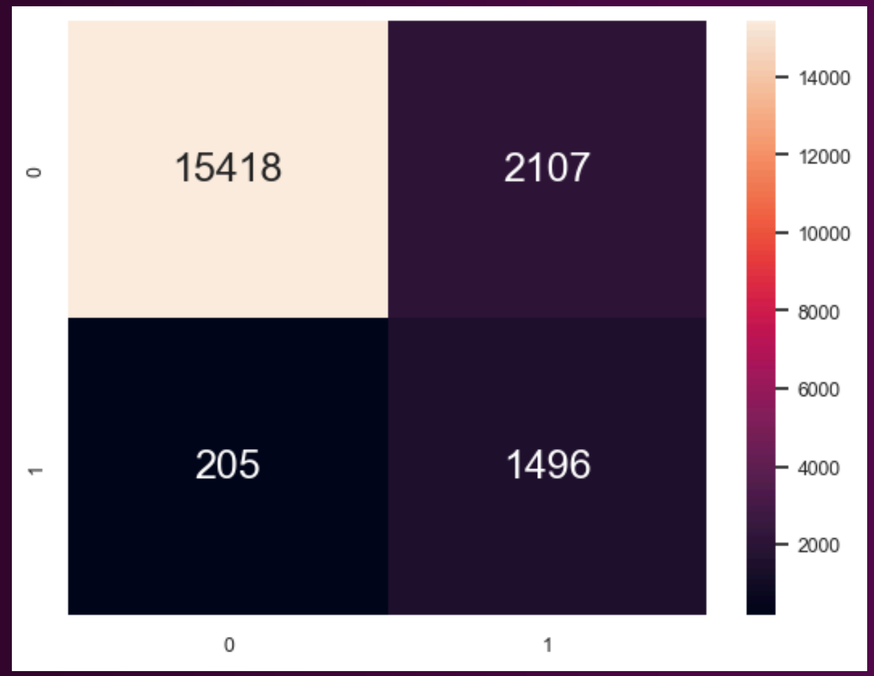
Train Data

Accuracy : 0.886
TPR Train : 0.885
TNR Train : 0.887
FPR Train : 0.113
FNR Train : 0.115

Precision : 0.846
Recall : 0.885
F1 Score : 0.865

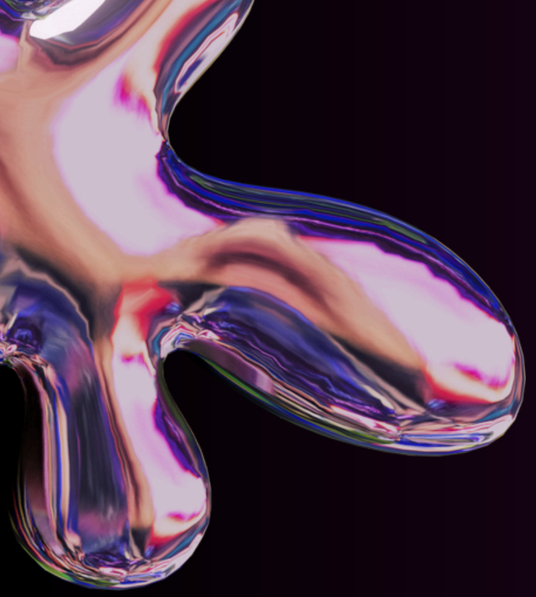


Test Data



Accuracy : 0.880
TPR Test : 0.879
TNR Test : 0.880
FPR Test : 0.120
FNR Test : 0.121

Precision : 0.415
Recall : 0.879
F1 Score : 0.564



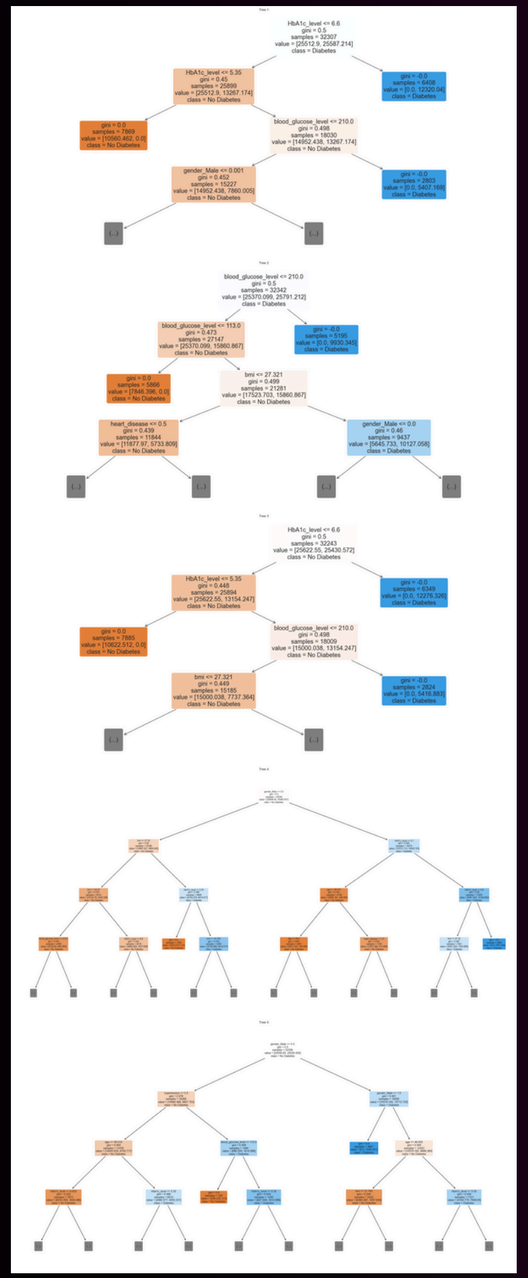
Logistic Regression

Train Data		Test Data	
Precision :	0.8455284552845529	Precision :	0.41520954759922285
Recall :	0.8849591177029854	Recall :	0.8794826572604351
F1 Score :	0.8647945555478131	F1 Score :	0.5641025641025641

- It flagged nearly 60 percent of the patients as diabetic even though they are not
- Floods the system with **false alarms**, predicting diabetes in many healthy patients

Machine Learning

Random Forest

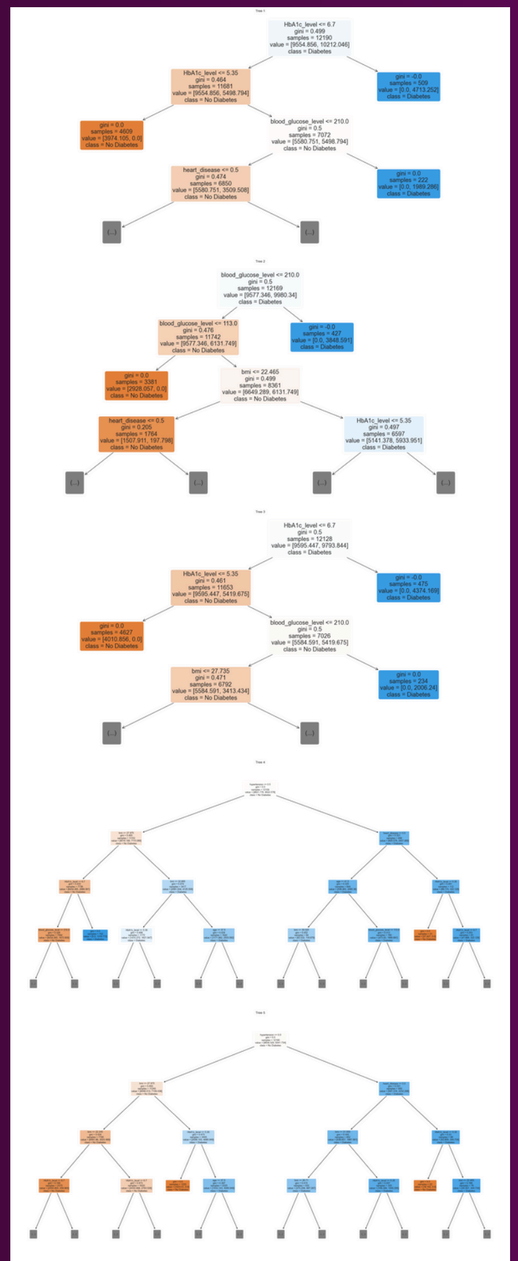


Random Forest

(before tuning)

Train Data

Test Data

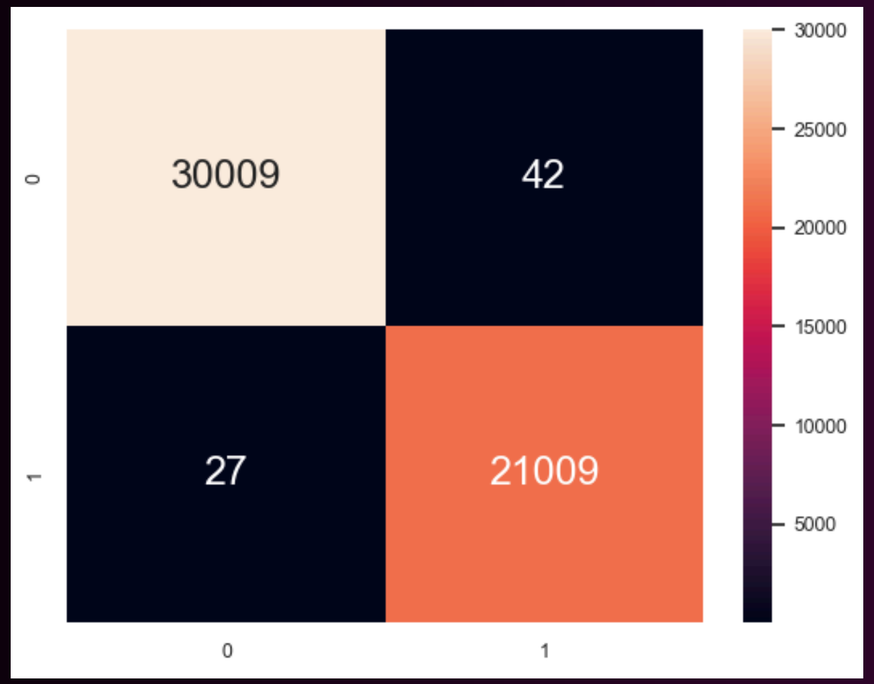


Random Forest
(before tuning)

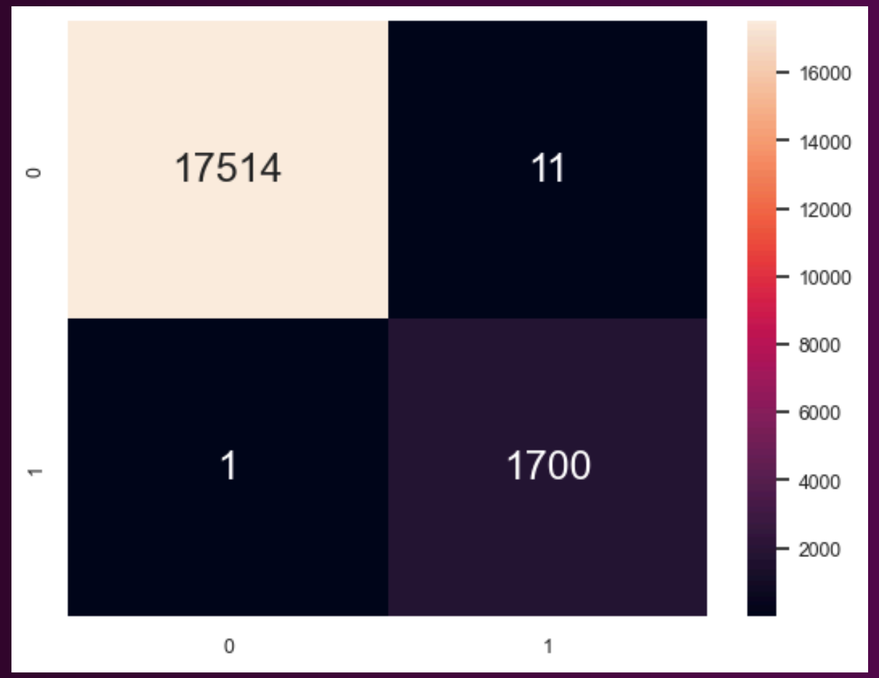
Train Data

Accuracy : 0.999
TPR Train : 0.999
TNR Train : 0.999
FPR Train : 0.001
FNR Train : 0.001

Precision : 0.998
Recall : 0.999
F1 Score : 0.998



Test Data



Accuracy : 0.999
TPR Test : 0.999
TNR Test : 0.999
FPR Test : 0.001
FNR Test : 0.001

Precision : 0.994
Recall : 0.999
F1 Score : 0.996

Machine Learning GridSearchCV

GridSearchCV

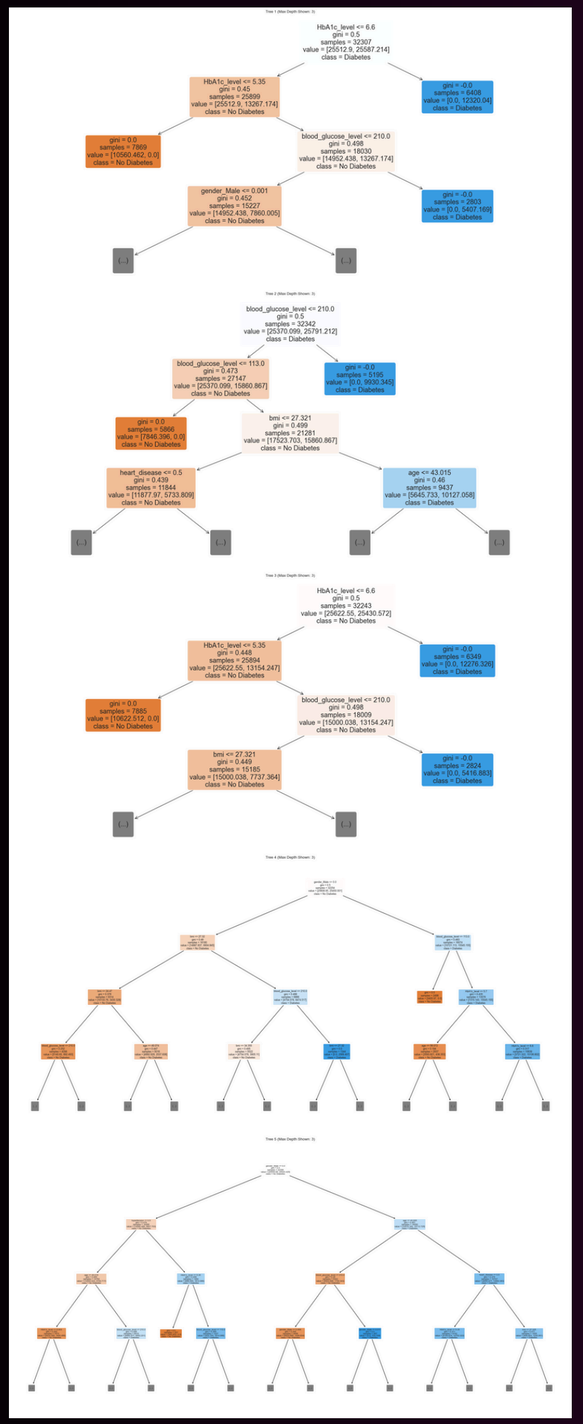
hyperparameter tuning

tested 216 different parameter combinations, using cross-validation to find the most reliable setting

Max Depth = 30

Min Samples Split = 5

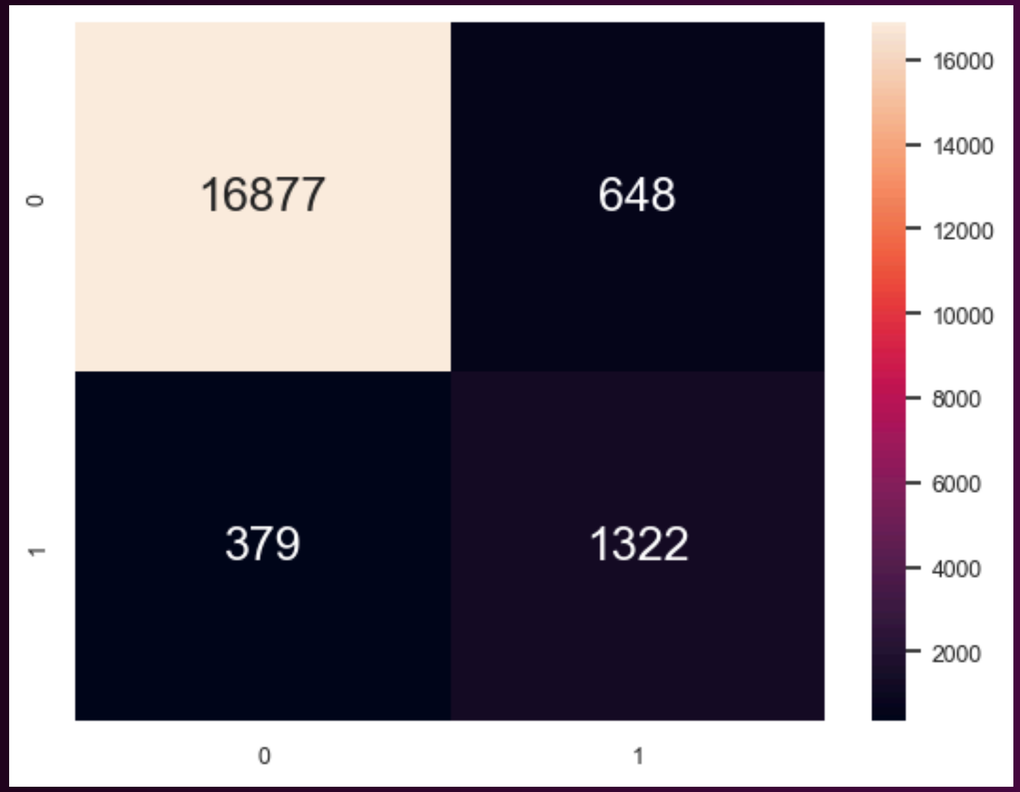
```
Fitting 5 folds for each of 216 candidates, totalling 1080 fits

Best Parameters: {'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}

Best rf: RandomForestClassifier(class_weight='balanced', max_depth=30,
                        min_samples_split=5, n_estimators=200, random_state=42)
```



Machine Learning Random Forest

Random Forest

(after tuning)

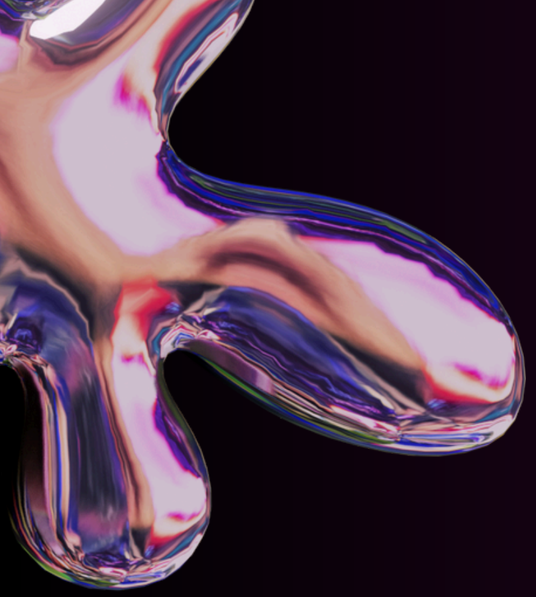


Test Data
Accuracy : 0.947

TPR Test : 0.777
TNR Test : 0.963

FPR Test : 0.037
FNR Test : 0.223

Precision : 0.671
Recall : 0.777
F1 Score : 0.720



Random Forest Comparison

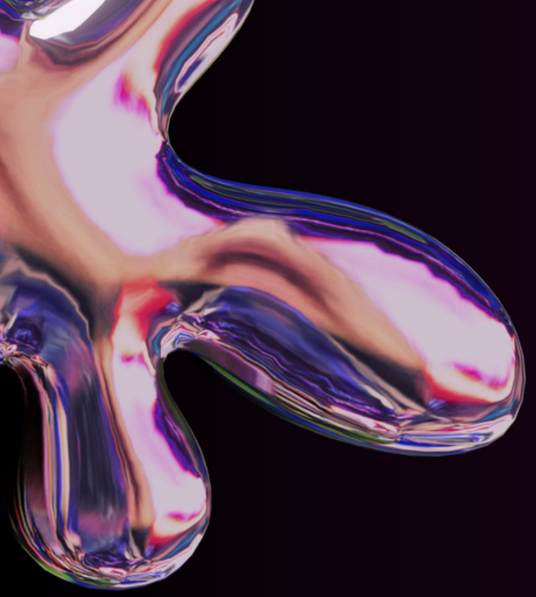
Before Tuning

```
Precision :     0.9935710111046172
Recall    :     0.9994121105232217
F1 Score  :     0.9964830011723329
```

After Tuning

```
Precision :     0.6710659898477157
Recall    :     0.7771898883009994
F1 Score  :     0.7202397166984473
```

- **Precision: 0.994 to 0.671**
- **Recall: 0.999 to 0.777**
- **F1 Score: 0.996 to 0.720**



Random Forest Comparison

- These numbers might seem worse, but they are actually **more trustworthy**.
- It catches slightly fewer true cases but is **more reliable when it does flag someone as high-risk.**

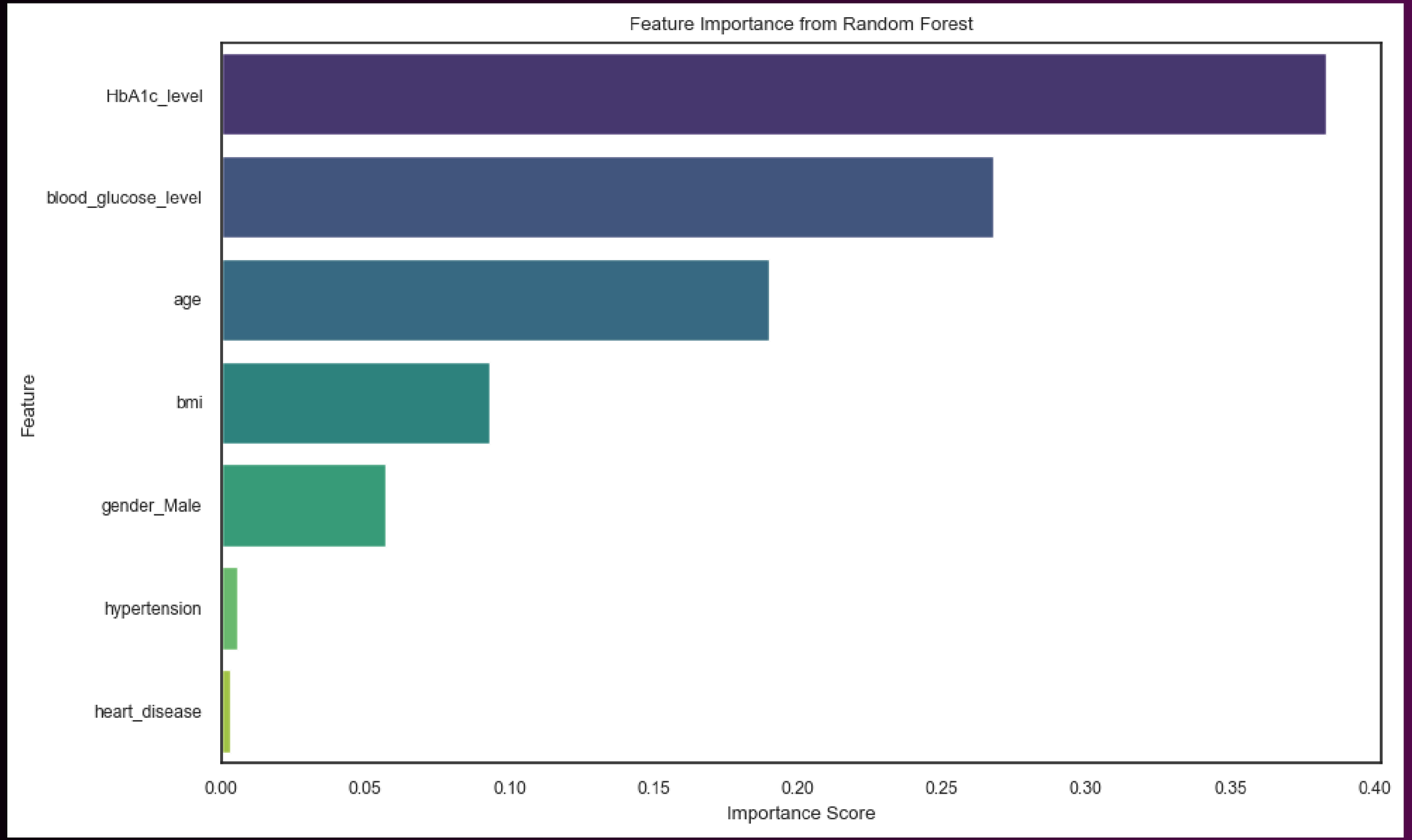
Conclusion

Outcome and Solution Effectiveness

- Captures around 3 out of 4 actual diabetes cases
- Keep false alarms at clinically manageable levels

```
Precision :      0.6710659898477157
Recall    :      0.7771898883009994
F1 Score  :      0.7202397166984473
```

Using our ML, healthcare providers can use it to **predict whether a patient is at risk of diabetes** and allow more **efficient use of resources** in healthcare sectors


Feature Importance from Random Forest
HbA1c_level
blood_glucose_level
age
bmi
gender_Male
hypertension
heart_disease
Feature
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Importance Score

Data Driven Insights

- Always include HbA1c and glucose tests as they are the most important predictors
- Adjust thresholds based on age
- Less emphasis on conditions like Hypertension and Heart Disease

Surprising finding

Hypertension and heart disease added almost no predictive value according to the ranking

BUT

It could be a result of class imbalance within these features. Our dataset contained a very small proportion of individuals with hypertension and heart disease

Actionable Recommendations

- Prioritize HbA1c testing for all patients over the age of 40, regardless of other risk factors
- Confirmatory testing should be focused on cases where HbA1c exceeds 5.7 and blood glucose is over 100
- Confirmatory testing should be focused on patients aged 50+ with BMI greater than 25

THANK YOU

Xie Xiaomei

- Data Cleaning
- Univariate Data Analysis (categorical data)
- Bivariate Data Analysis (categorical data)
- Resampling
- Machine Learning (Multivariate Classification Tree, Random Forest)

Xie Xiaotian

- Data Cleaning
- Univariate Data Analysis (numerical data)
- Bivariate Data Analysis (numerical data)
- Multivariate Data Analysis
- Machine Learning (Random Forest, Logistic Regression, Hyperparameter Tuning)