Predicting Diabetes: A Machine Learning Approach

Leveraging Machine Learning for Early Diabetes Detection in Pima Indians

Diabetes Prediction Using Machine Learning: Using the Pima Indians Dataset

1. Introduction and Objective

Early detection of diabetes is crucial in enhancing patient outcomes and reducing healthcare costs. This project develops a classification model to predict diabetes using the Pima Indians Diabetes Dataset, focusing on:

- Effectively handling missing values and outliers.
- Exploring feature correlations to understand their relationship with diabetes.
- Applying quantile transformation to address feature skewness.
- Testing and evaluating classification models for performance.

The target variable "Outcome" signifies whether a patient is diabetic (1) or not (0), making this a binary classification task.

2. Dataset Overview

- **Source:** Kaggle.
- **Shape:** 768 rows × 9 columns.
- **Target Variable:** Outcome (1 = diabetic, 0 = non-diabetic).
- **Features:**
  - Pregnancies
  - Glucose
  - BloodPressure
  - SkinThickness
  - Insulin
  - BMI
  - DiabetesPedigreeFunction
  - Age

3. Key Data Processing Steps

A. Handling Invalid Values

Several columns contained zeros, which were treated as missing values:

df[cols_to_fix] = df[cols_to_fix].replace(0, np.nan)

B. Missing Value Imputation

A KNN Imputer (n_neighbors=10) was used to impute missing values, preserving data patterns. Alternatives like mean imputation may be explored in future work.

C. Feature Selection

Features were selected based on their correlation with the target:

- **Dropped Features:** BloodPressure, DiabetesPedigreeFunction (low correlation with Outcome).

Note: Correlation does not imply causation. Future iterations could use methods like mutual information or recursive feature elimination.

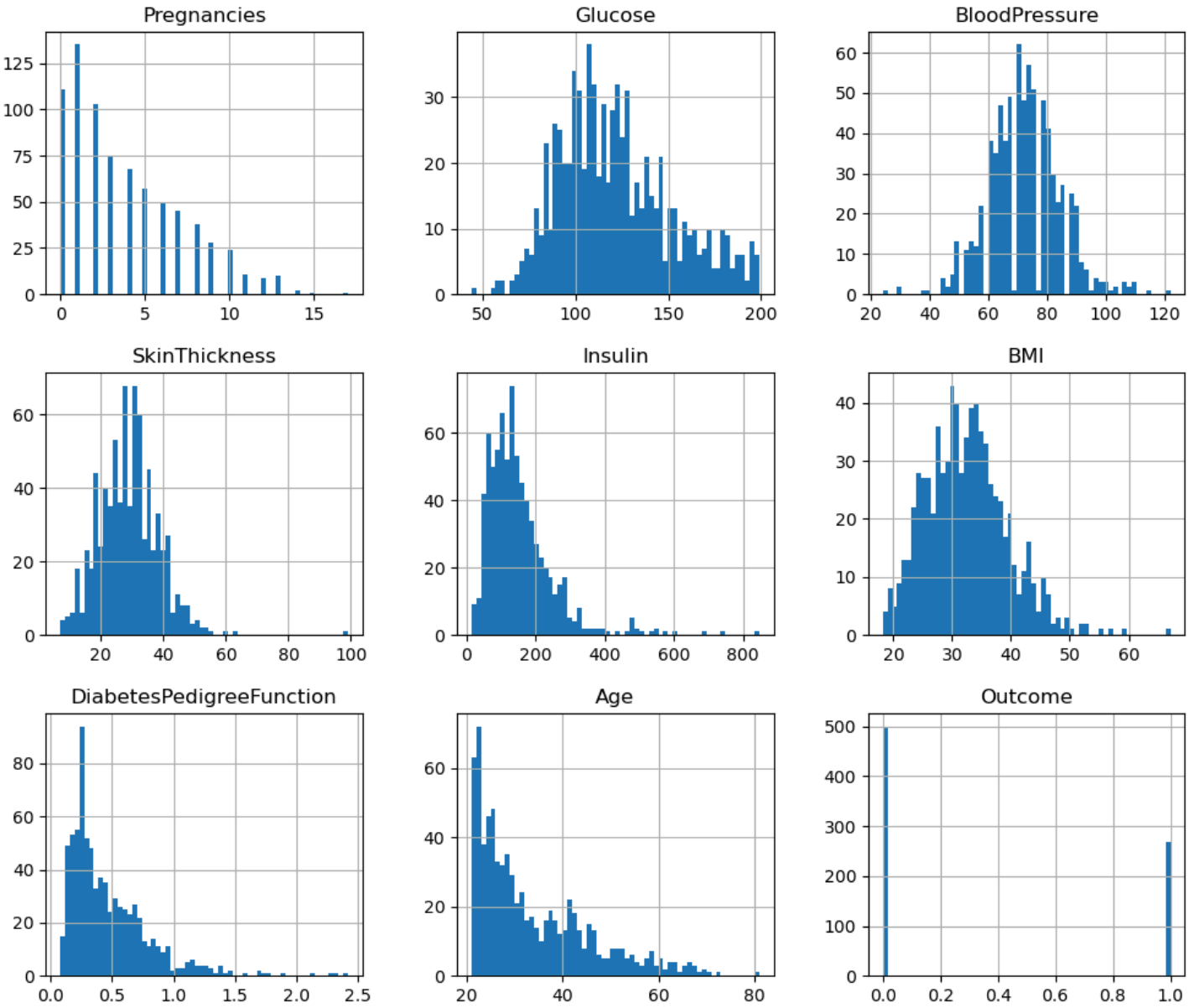
4. Exploratory Data Analysis (EDA)

A. Class Distribution

- Diabetic: 35%
- Healthy: 65%

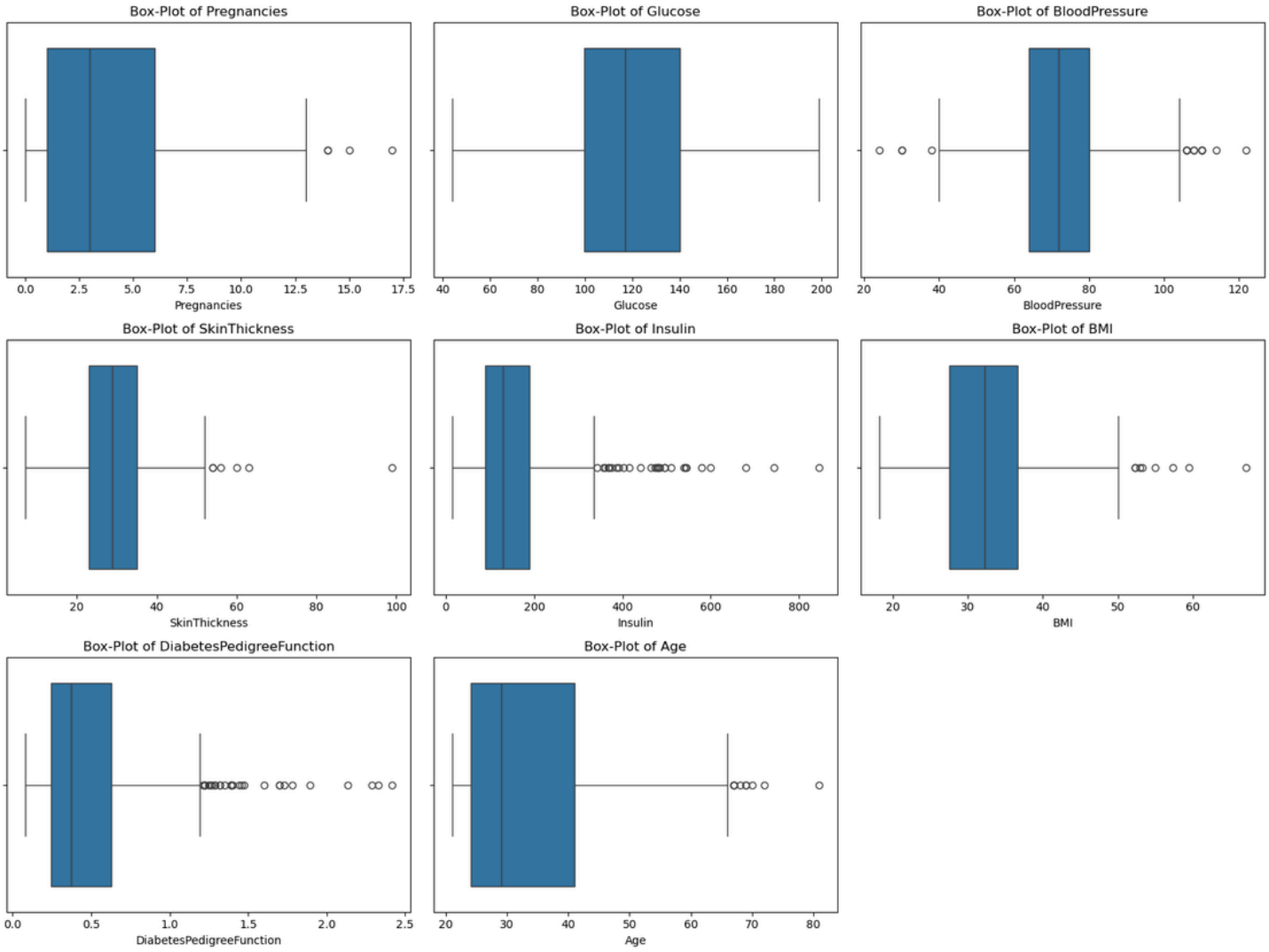
The dataset is imbalanced, with more non-diabetic cases, influencing model performance.

B. Skewness



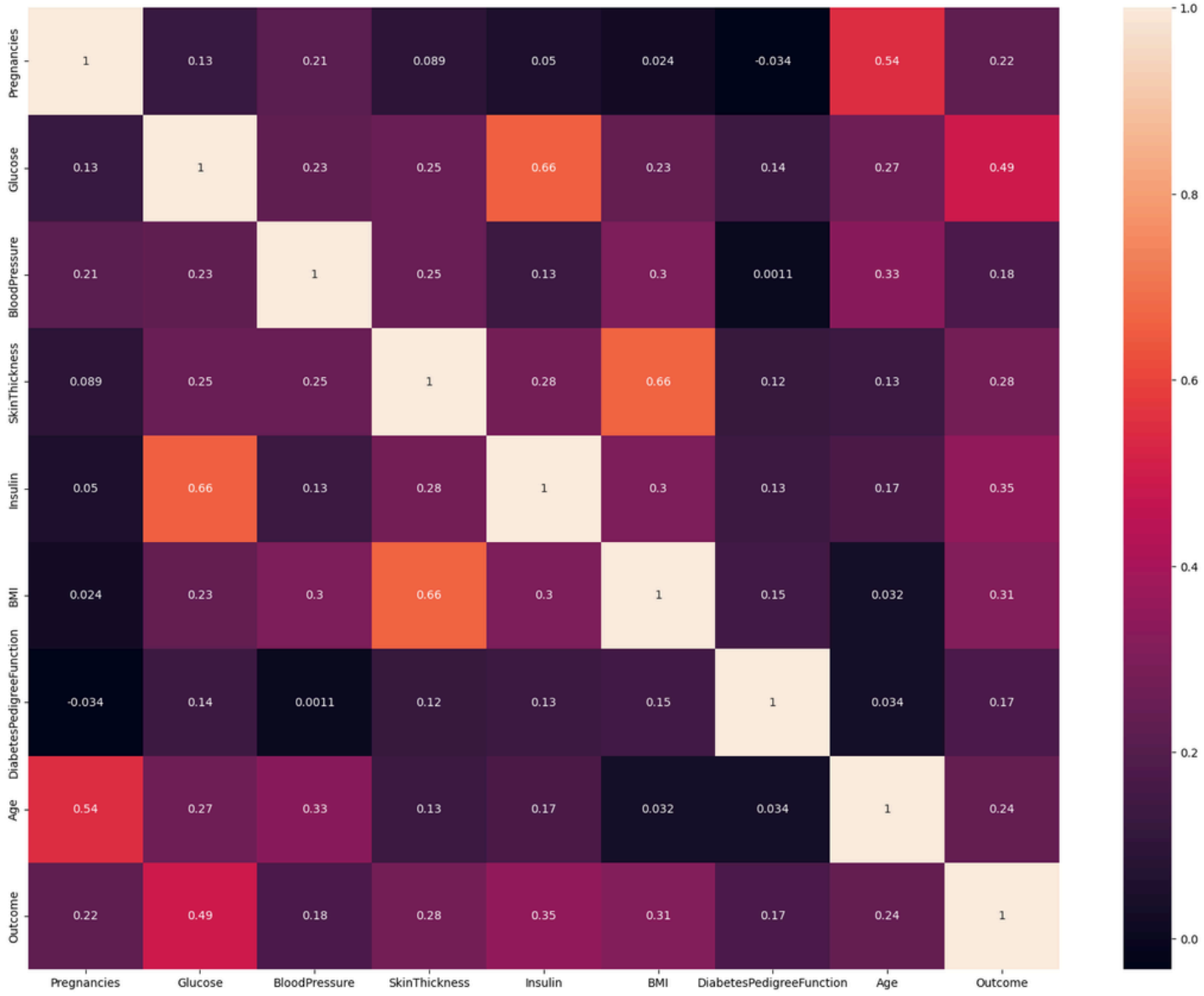
Most features were skewed, except Glucose and BloodPressure, which were approximately normal.

C. Outlier Detection



Boxplots and skewness metrics identified significant outliers in Insulin, SkinThickness, and BMI, necessitating preprocessing.

D. Feature Correlation



Key features correlated with diabetes:

- Glucose (strongest correlation)
- BMI
- Age
- Pregnancies

A correlation heatmap visually supported the decision to drop low-correlation features.

5. Feature Engineering

A. Quantile Transformation

A QuantileTransformer was applied to normalize skewed features, and outlier removal, enhancing model performance:

```
quantile = QuantileTransformer()
X = quantile.fit_transform(df_selected)
```

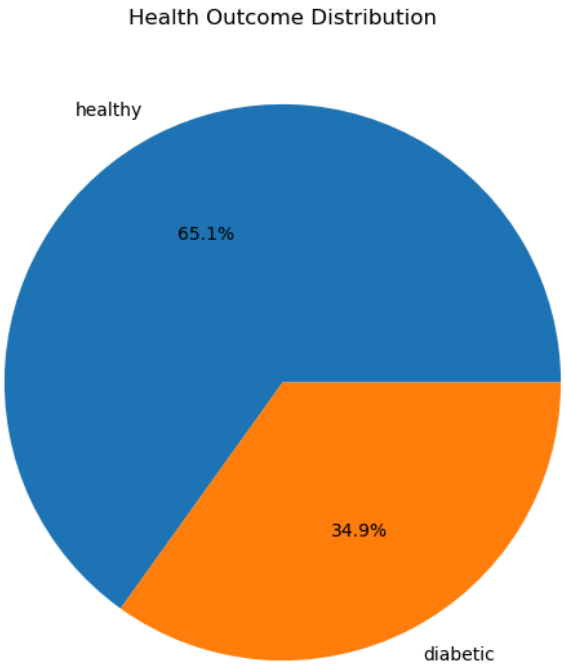
Post-transformation distributions were more normal-like.

B. Dataset After Transformation

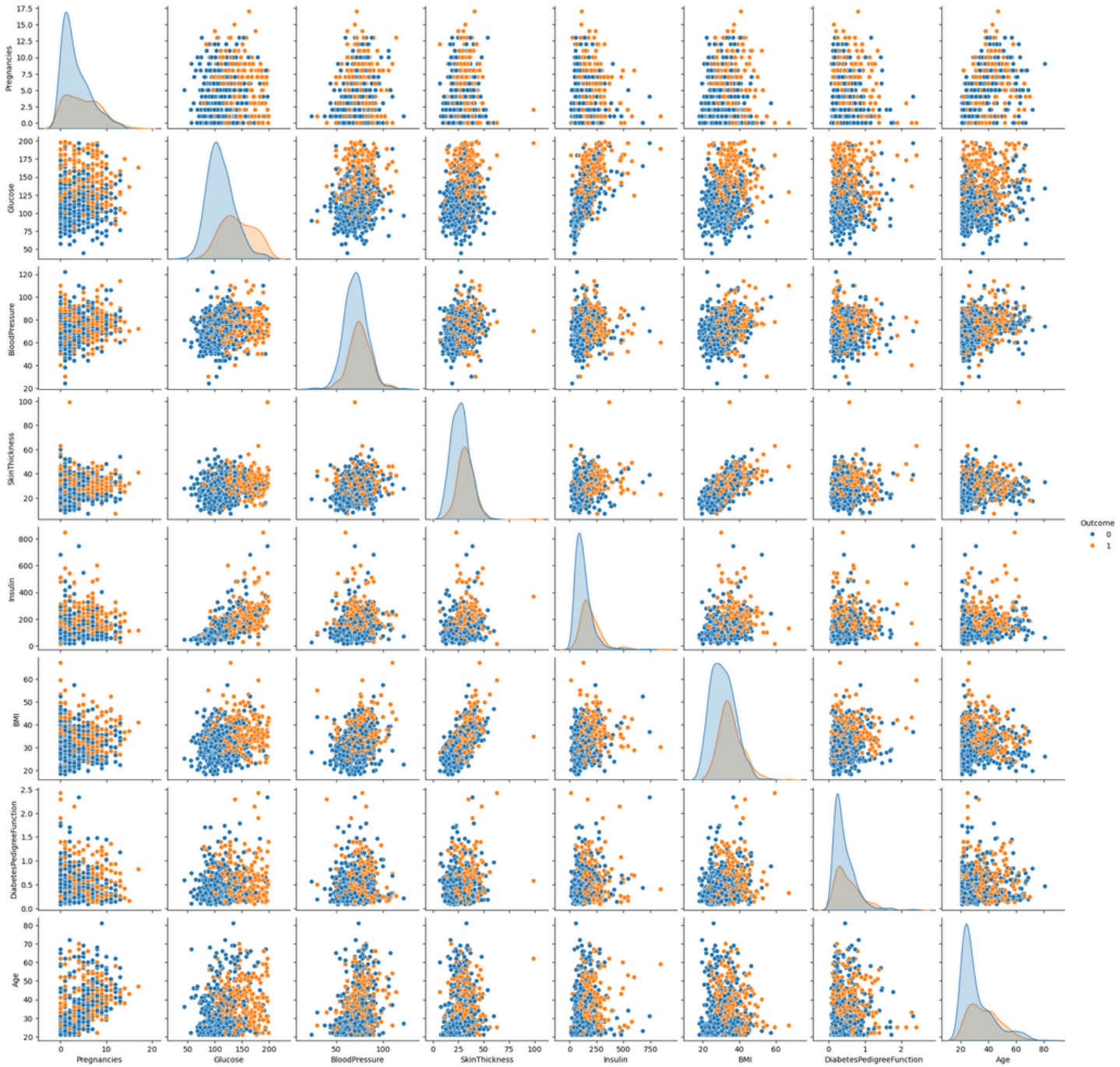
A new DataFrame was created with transformed features and removed Outliers, preserving the Outcome column for modeling.

6. Visualization Insights

- **Pie Chart & Barplot:** Illustrated class imbalance (65% healthy, 35% diabetic).



- **Boxplots:** Highlighted outliers before and after quantile transformation.
- **Pairplot:** Showed class separation, especially for Glucose, BMI, and Age.



- **Heatmap:** Confirmed feature correlations and justified feature selection.

Recommendation: Include these visualizations in the final report for clarity and impact.

7. Model Training and Evaluation

Three ensemble models were evaluated: Random Forest Classifier (RFC), Gradient Boosting Classifier (GBM), and XGBoost Classifier (XGB). Performance was assessed before and after hyperparameter tuning.

7.1 Random Forest Classifier

Without Hyperparameter Tuning

Metric	Class 0	Class 1
Precision	0.83	0.68
Recall	0.87	0.61
F1-Score	0.85	0.65
Accuracy	0.79	

With Hyperparameter Tuning

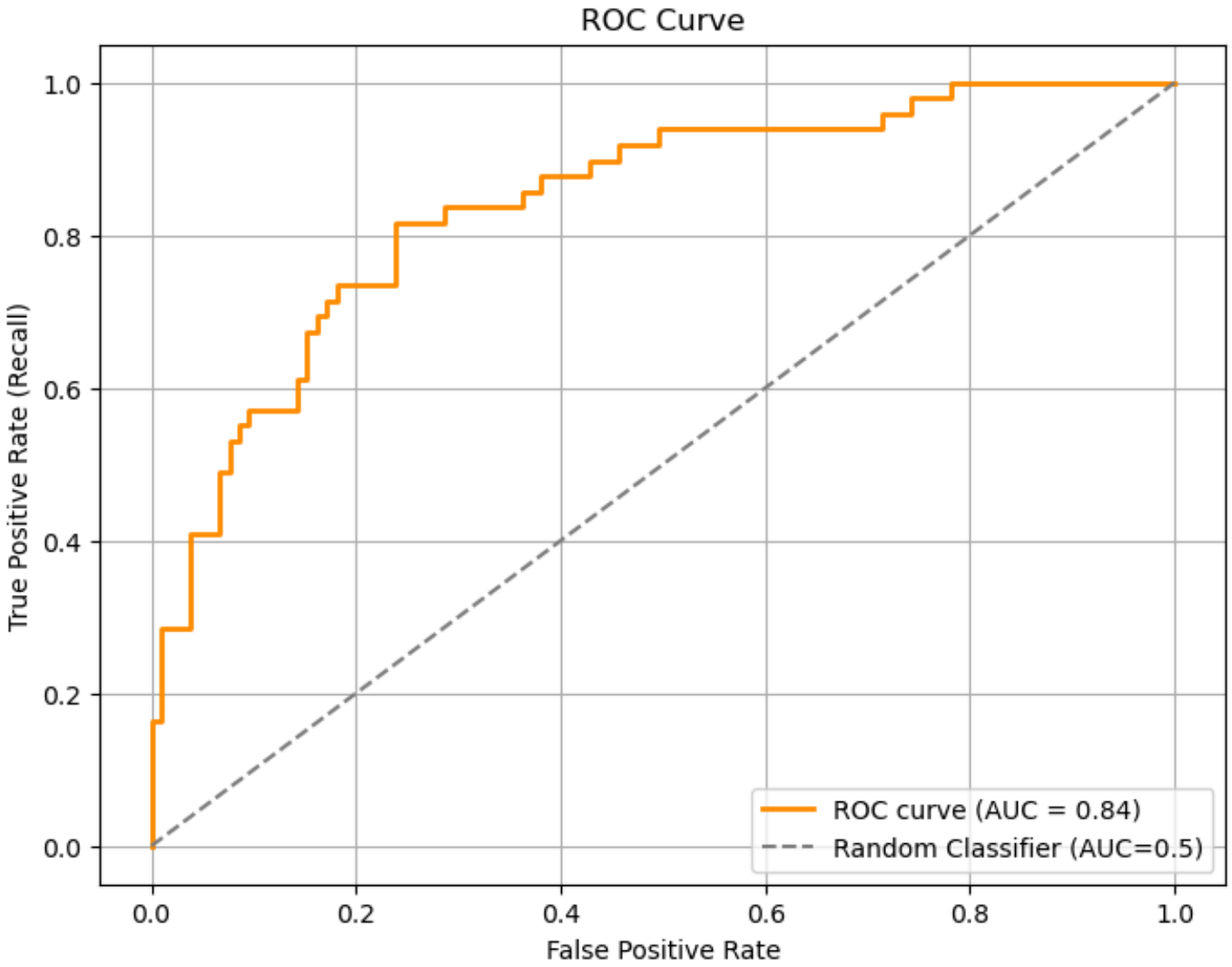
Metric	Class 0	Class 1
Precision	0.85	0.67
Recall	0.84	0.69
F1-Score	0.85	0.68
Accuracy	0.79	
AUC	0.84	

Best Parameters:

'bootstrap': True, 'class_weight': 'balanced', 'max_depth': 10,

'max_features': 'log2', 'min_samples_leaf': 1,'min_samples_split': 5,

'n_estimators': 200



Insight: Tuning improved recall for the diabetic class, critical in medical contexts to reduce false negatives.

7.2 Gradient Boosting Classifier

Without Hyperparameter Tuning

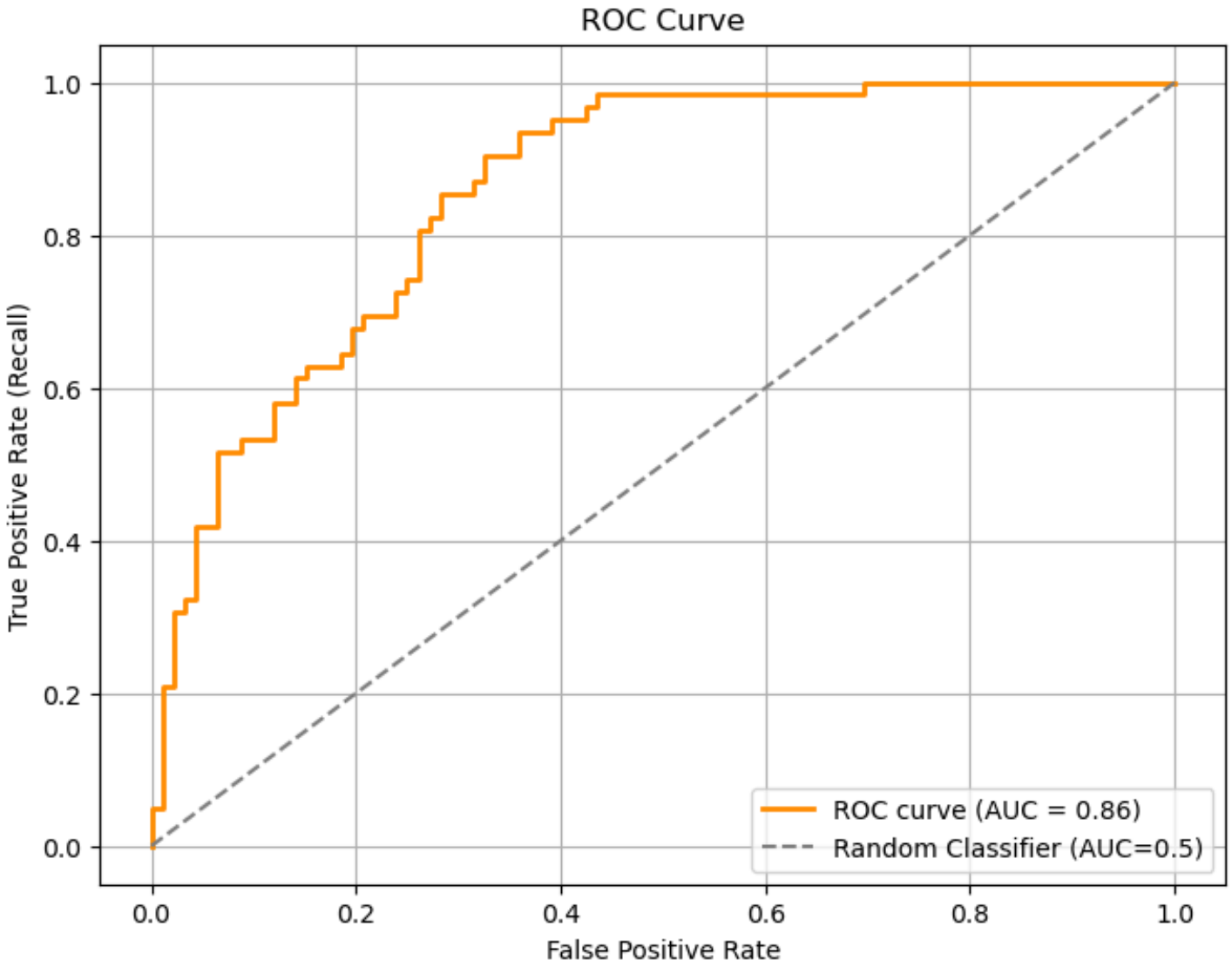
Metric	Class 0	Class 1
Precision	0.78	0.74
Recall	0.85	0.65
F1-Score	0.81	0.69
Accuracy	0.77	

With Hyperparameter Tuning

Metric	Class 0	Class 1
Precision	0.77	0.71
Recall	0.83	0.63
F1-Score	0.80	0.67
Accuracy	0.75	
AUC	0.86	

- **Best Parameters:**

{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 200}



Insight: Accuracy decreased slightly, but AUC improved, suggesting better class separation. The drop in diabetic class recall may reflect a trade-off favoring overall performance.

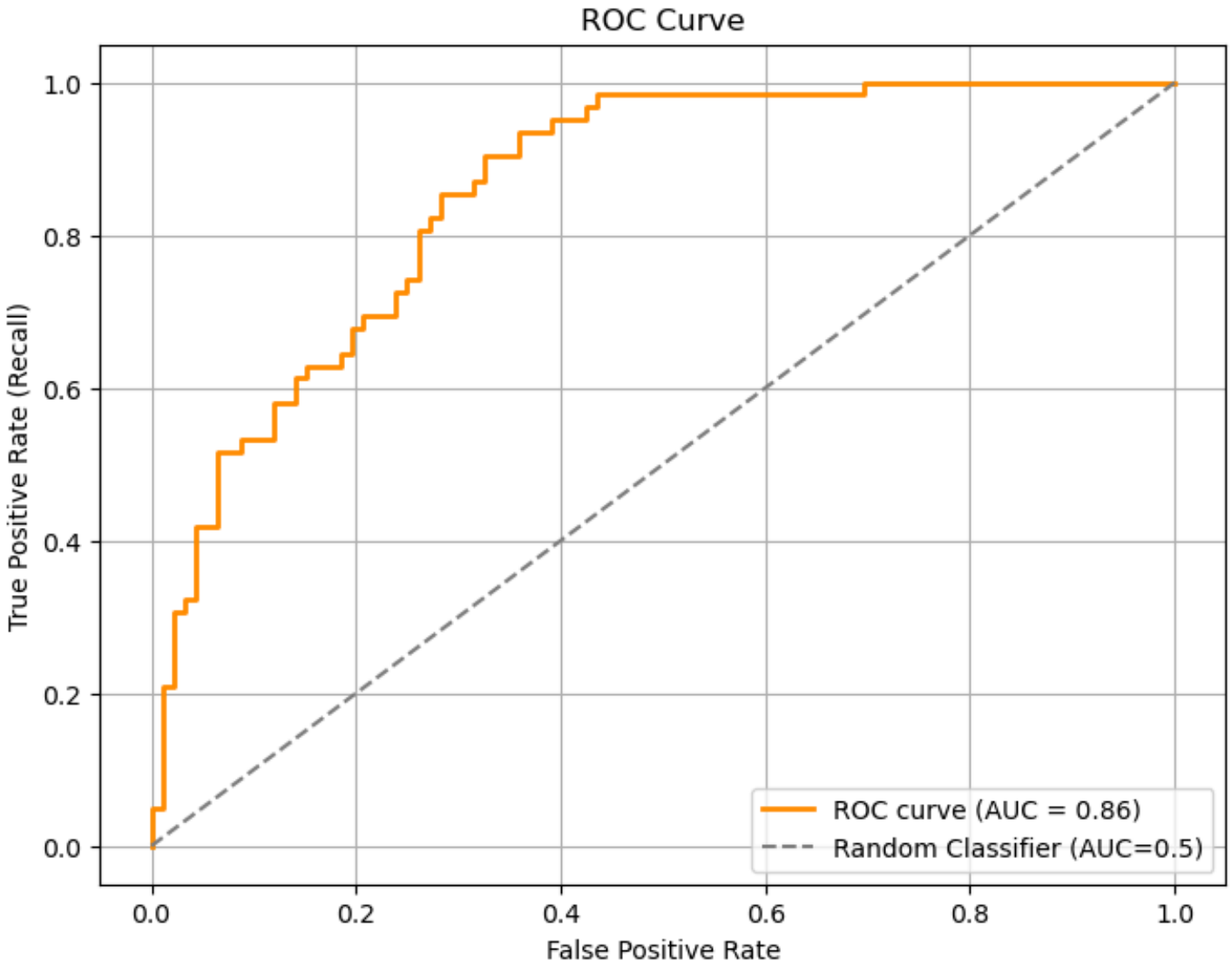
7.3 XGBoost Classifier

Without Hyperparameter Tuning

Metric	Class 0	Class 1
Precision	0.78	0.72
Recall	0.83	0.66
F1-Score	0.80	0.69
Accuracy	0.76	
AUC	—	

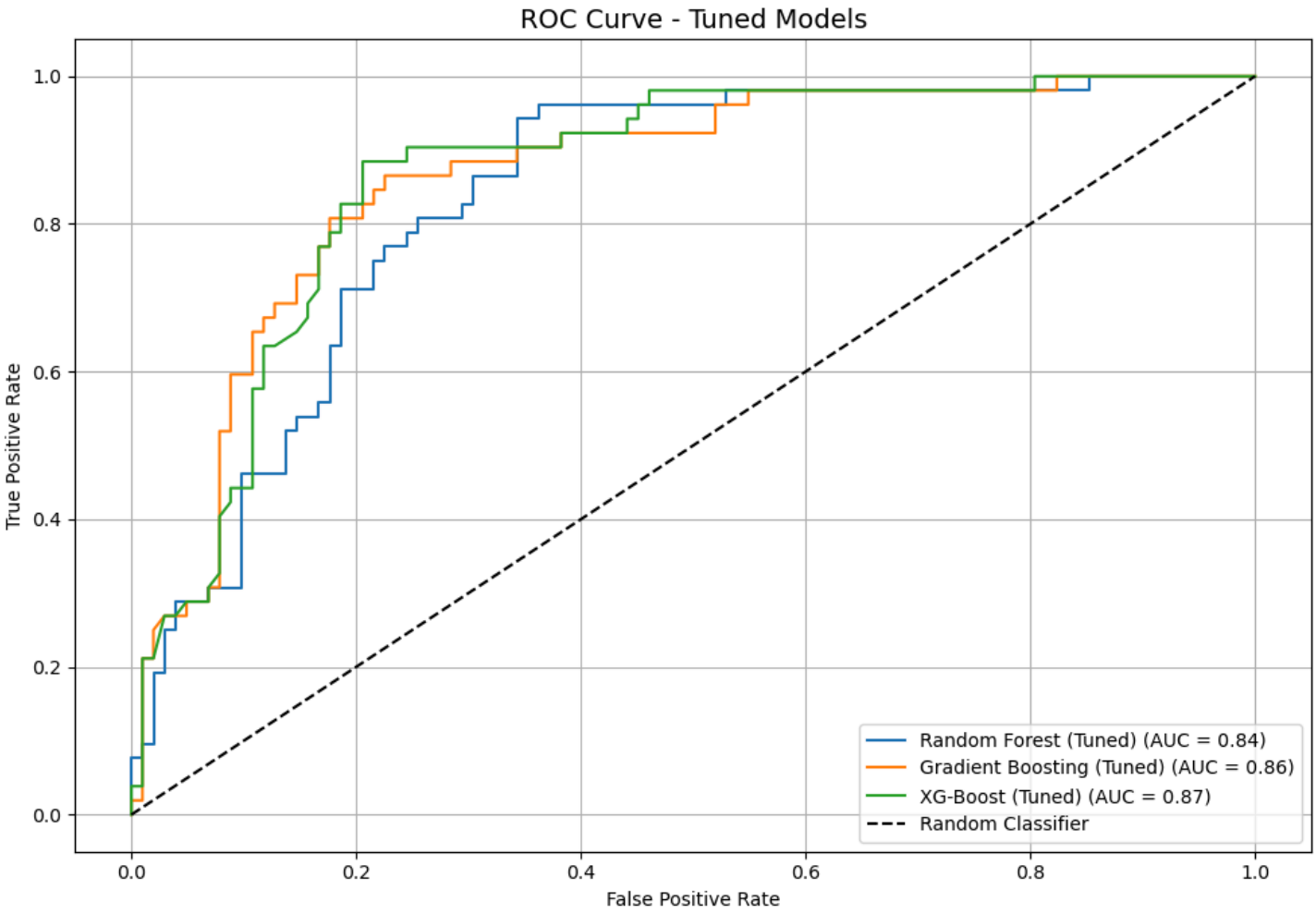
With Hyperparameter Tuning

Metric	Class 0	Class 1
Precision	0.78	0.72
Recall	0.83	0.66
F1-Score	0.80	0.69
Accuracy	0.76	
AUC	0.88	



Insight: XGBoost achieved the highest AUC, indicating superior class separation, though other metrics remained stable post-tuning.

8. Comparative Summary



Model	Accuracy	F1 (Diabetic)	Recall (Diabetic)	AUC
RFC (Tuned)	0.75	0.68	0.69	0.84
GBM (Tuned)	0.75	0.73	0.75	0.86
XGB (Tuned)	0.76	0.76	0.76	0.88

- **Best Balanced Model:** Random Forest (post-tuning) — Highest recall and solid F1-score for the diabetic class, ideal for minimizing false negatives.
- **Best AUC Score:** XGBoost — Excels in class separation, suitable for probability-based predictions.
- **GBM:** Competitive AUC but lower recall post-tuning.

Recommendation: For medical applications prioritizing diabetic case detection, the tuned Random Forest is recommended. For prediction confidence and class separation, XGBoost's higher AUC is preferable.

9. Conclusion and Future Work

This project successfully built and evaluated machine learning models to predict diabetes using the Pima Indians dataset. Key steps included robust preprocessing, feature engineering, and model comparison. Random Forest balanced recall and F1-score effectively, while XGBoost offered the best class separation.