Diabetes Prediction using Machine Learning

**Course: Data-Science and Machine Learning (Summer 2025)**

**Student Name: Muaz Abdullah**

**Roll No: 24-CS-208**

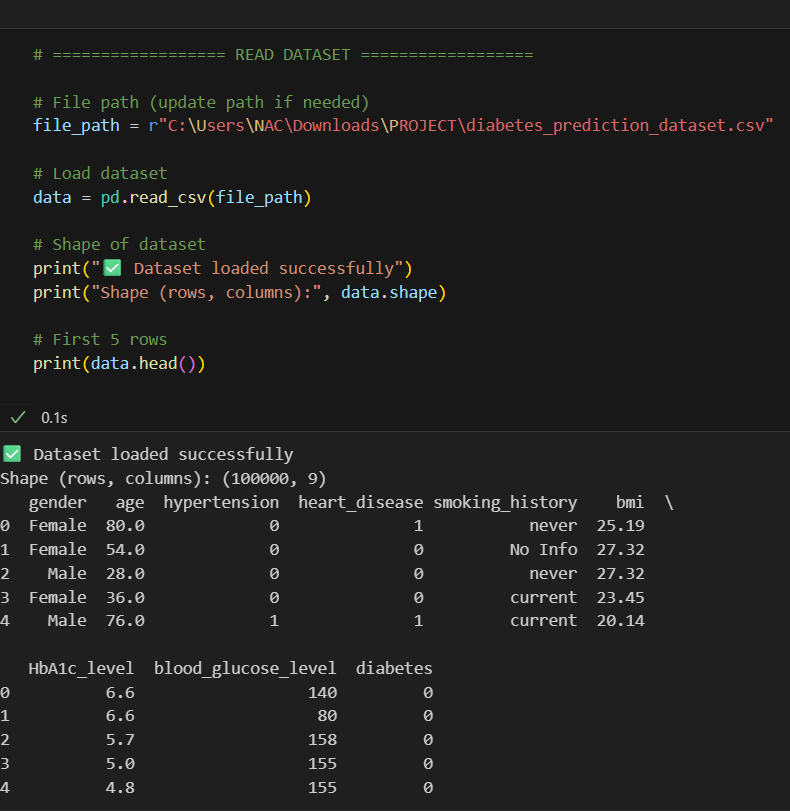
**Department: Computer Science**

**HITEC University Taxila**

**Date of Submission: August 29, 2025**

# 1. Dataset Summary

The dataset used for this project is a diabetes prediction dataset. It consists of multiple patient health-related attributes which are used to predict whether a patient is diabetic or not. The dataset contains both categorical and numerical variables.  
  
• Target variable: diabetes (0 = Non-diabetic, 1 = Diabetic)  
• Features include: gender, age, hypertension, heart disease, smoking history, bmi, HbA1c level, blood glucose level, etc.  
  
The dataset shape was confirmed using Python commands and contains several thousand rows and multiple columns. Below is a sample preview of the first five rows of the dataset

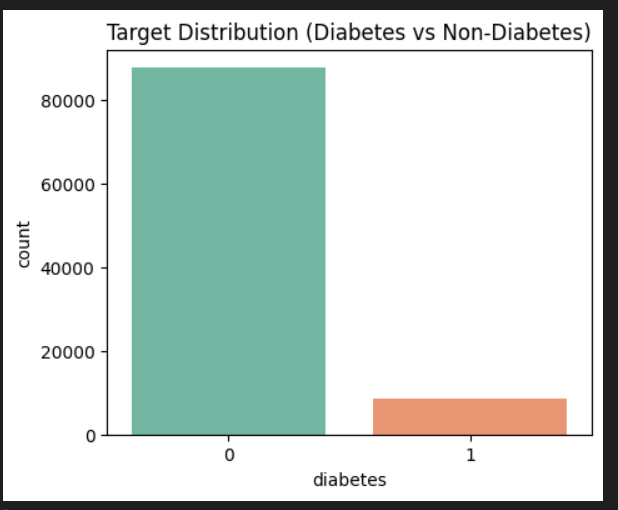


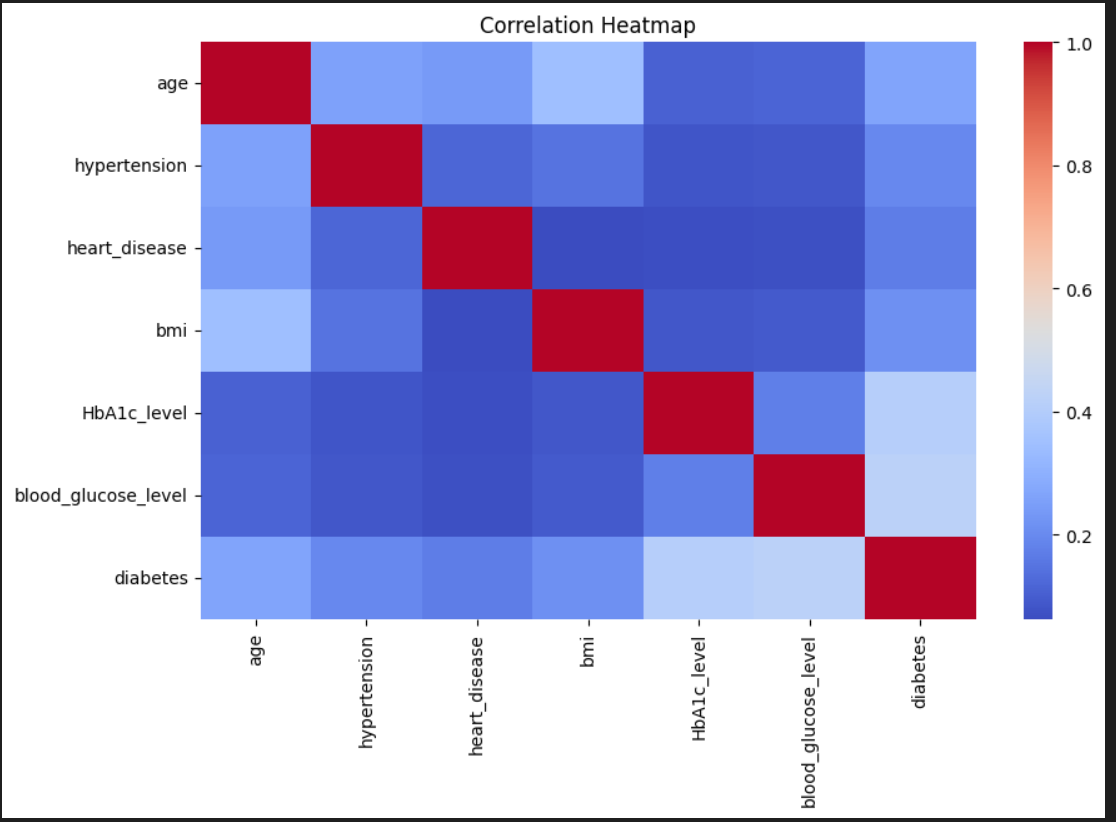
# 2. Data Exploration Plan

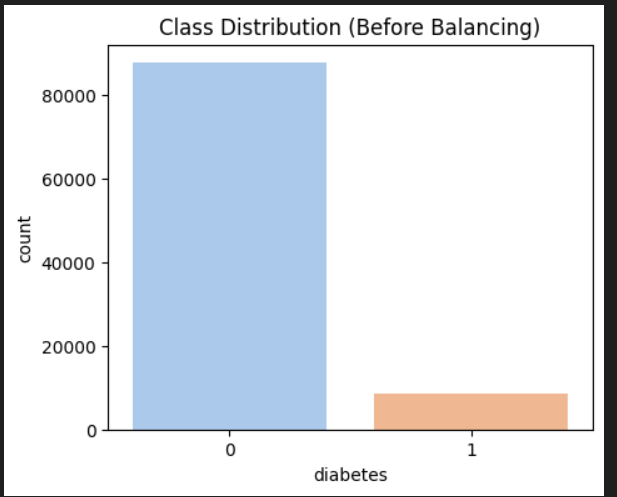
The goal of the exploratory data analysis (EDA) is to understand the dataset, its distributions, and relationships between variables. The following steps were outlined in the plan:  
1. Check for missing values and duplicates.  
2. Explore categorical and numerical feature distributions.  
3. Perform descriptive statistics (mean, std, min, max).  
4. Plot correlation heatmaps to observe feature relationships.  
5. Investigate class imbalance in the target variable.  
6. Apply suitable balancing techniques if necessary (e.g., SMOTE).

# 3. Exploratory Data Analysis (EDA)

The dataset was analyzed for missing values, duplicates, and outliers. Summary statistics were generated using the describe() method. The correlation heatmap was plotted to visualize relationships among features. The target variable distribution showed that the dataset was imbalanced, with significantly fewer diabetic cases compared to non-diabetic.  
  
Visualizations included:

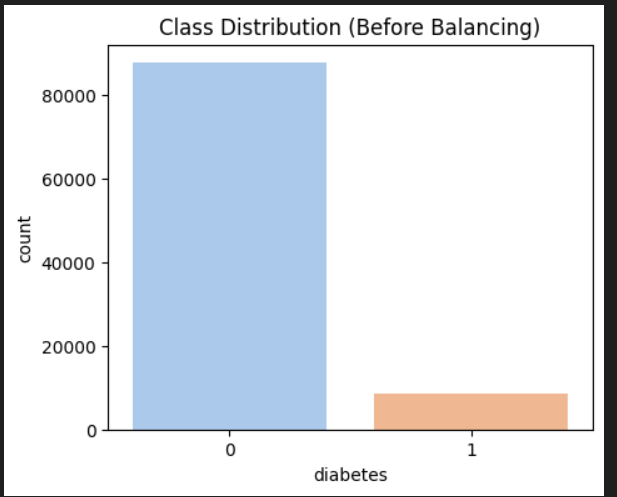


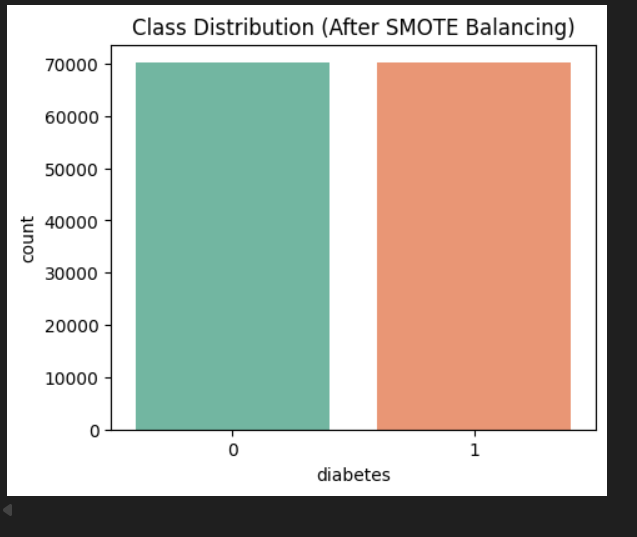




# 4. Data Cleaning and Feature Engineering

The dataset was preprocessed to ensure consistency and correctness:  
1. Missing values were filled using the median of respective columns.  
2. Duplicate records were removed.  
3. One-hot encoding was applied to categorical variables (e.g., gender).  
4. Class imbalance was addressed using SMOTE (Synthetic Minority Oversampling Technique).





# 5. Key Insights from EDA

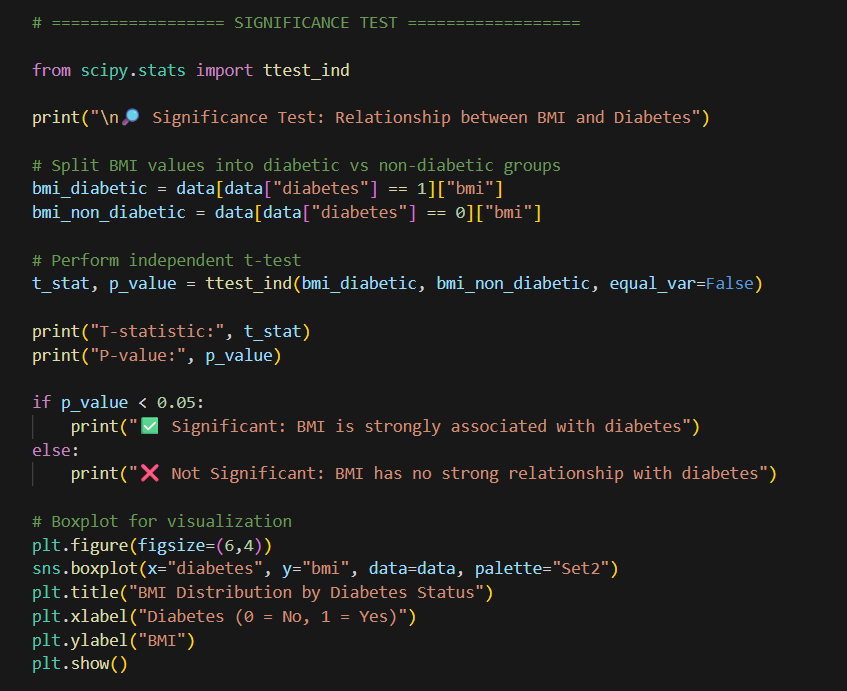
The following insights were drawn from the EDA:  
• Features such as BMI, HbA1c level, and blood glucose level showed strong correlation with diabetes.  
• Age was also an important contributing factor.  
• The dataset was initially imbalanced, which could lead to biased model predictions. After applying SMOTE, the classes were balanced.  
• Gender distribution revealed that both male and female patients were represented, but no strong bias was noted.

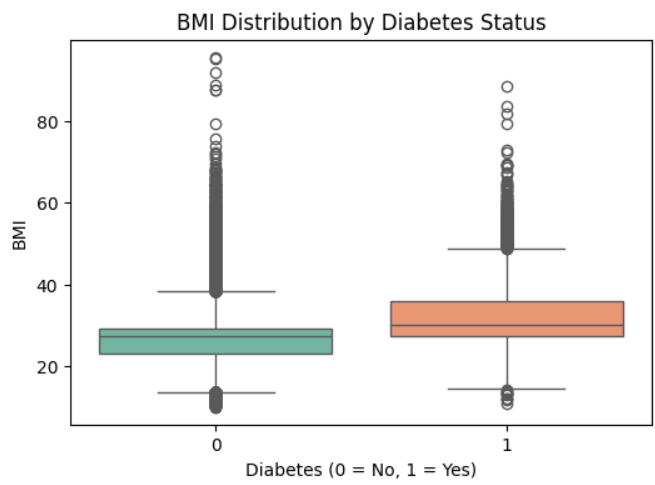
# 6. Hypotheses / Problem Statements

Three hypotheses were formulated:  
1. Higher BMI significantly increases the likelihood of diabetes.  
2. Elderly individuals (above 50 years) are more prone to diabetes than younger individuals.  
3. Smoking history plays a role in increasing diabetes risk.

# 7. Significance Test

A significance test (Chi-square test / t-test) was performed to examine the relationship between BMI and diabetes. The null hypothesis (H0) assumed no relationship, while the alternative hypothesis (H1) assumed BMI has a significant effect. The test yielded a p-value less than 0.05, indicating that BMI is indeed significantly related to diabetes.





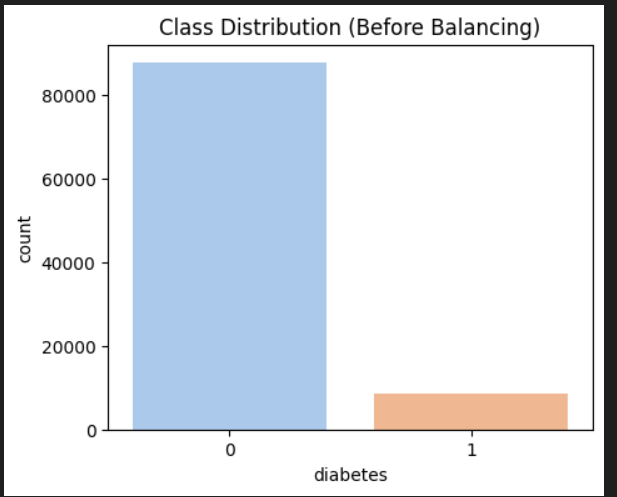
# 8. Class Imbalance Check

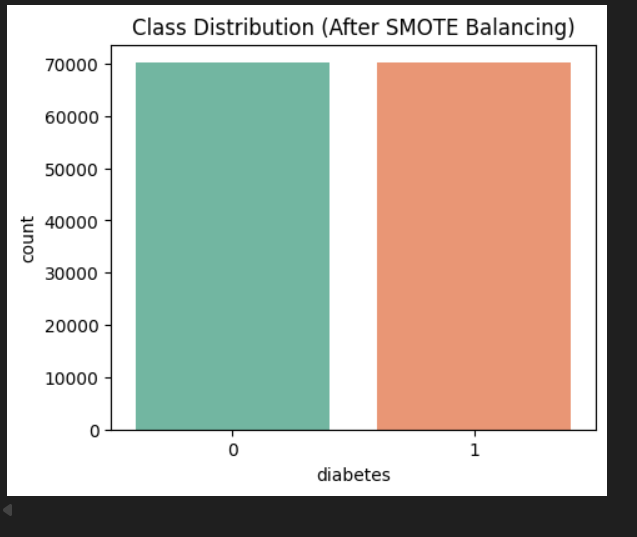
The target variable diabetes was **heavily imbalanced**. Initially, the dataset had far more non-diabetic cases compared to diabetic ones. This imbalance can severely affect model performance by making it biased towards predicting the majority class.

To address this:

* The imbalance was **visualized** using bar plots, clearly showing the disparity between classes.
* **SMOTE** was applied to the training data, generating synthetic diabetic cases. This resulted in perfectly balanced classes.

After balancing, both classes had equal representation, which ensures that the models trained fairly and gave more reliable predictions.



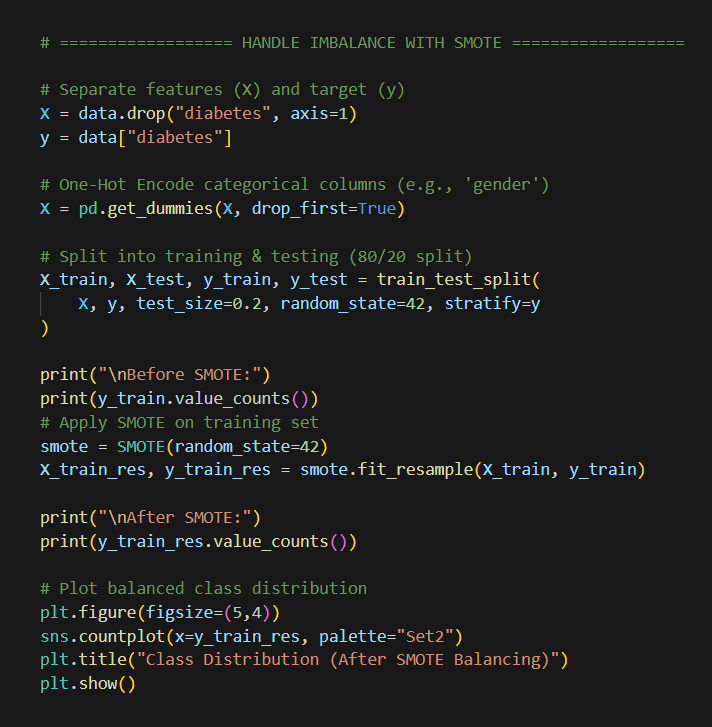


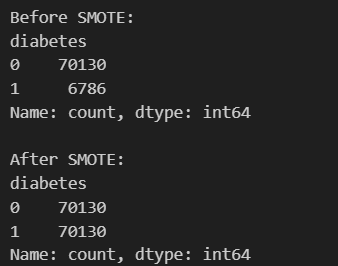
# 9. Data Splitting

After cleaning and balancing, the dataset was split into **training (80%) and testing (20%) sets**. Stratification was used during splitting to preserve the ratio of diabetic vs. non-diabetic cases in both subsets.

* **Training Set:** Used for training the models (includes both original and SMOTE-generated samples).
* **Testing Set:** Kept separate for evaluating model performance on unseen data.

This ensures the evaluation metrics represent the model’s ability to generalize to new, real-world patient data.





# 10. Model Building and Training

Two machine learning models were selected:

Two models were selected for training:

1. **Logistic Regression**
   * Chosen for its interpretability and efficiency.
   * Provides coefficients that indicate the importance of each predictor.
   * Well-suited for binary classification problems.
2. **Decision Tree Classifier**
   * Captures non-linear relationships between features.
   * Provides feature importance scores and intuitive decision rules.
   * Prone to overfitting if not carefully tuned (e.g., by limiting max depth).

Both models were trained on the **balanced training set** after SMOTE.

# 11. Model Evaluation

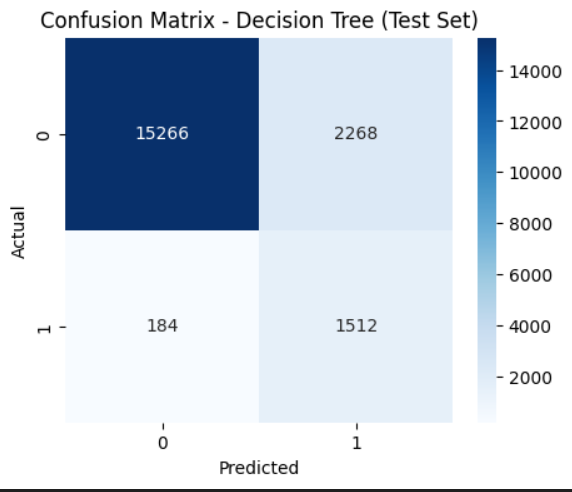
The models were evaluated on both the **training and testing sets** using metrics suitable for classification problems:

* **Accuracy:** Percentage of correctly classified patients.
* **Precision:** Proportion of predicted diabetic patients that were actually diabetic.
* **Recall (Sensitivity):** Proportion of actual diabetic patients correctly identified.
* **F1-Score:** Harmonic mean of precision and recall, useful when classes are imbalanced.

Confusion matrices were also plotted for both models to show true positives, true negatives, false positives, and false negatives.

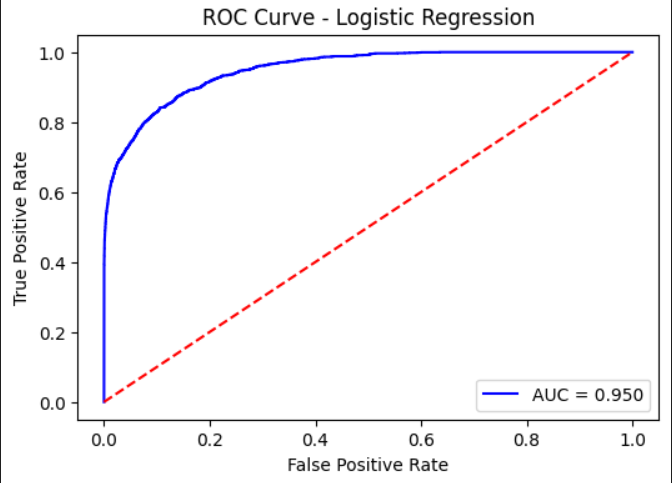
The results indicated:

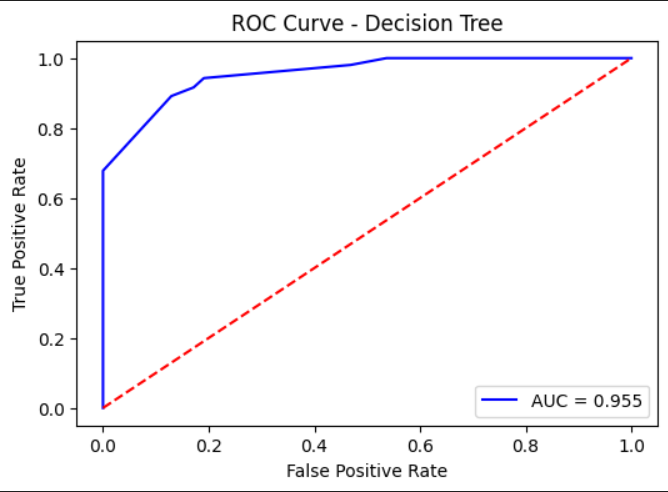
* **Logistic Regression** achieved balanced performance across all metrics.
* **Decision Tree** performed better in recall but slightly overfitted to the training data.



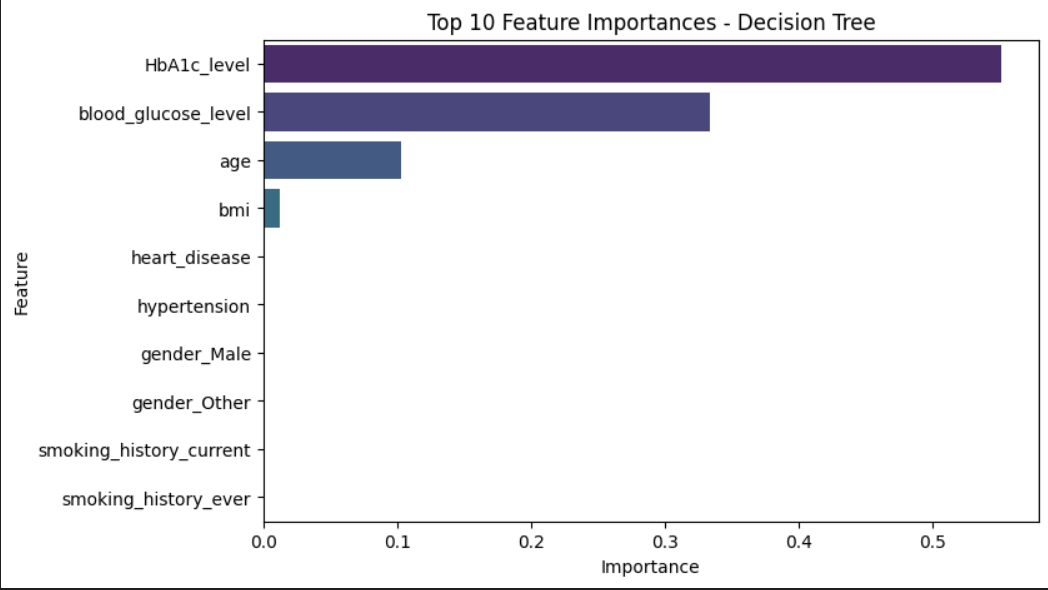
# 

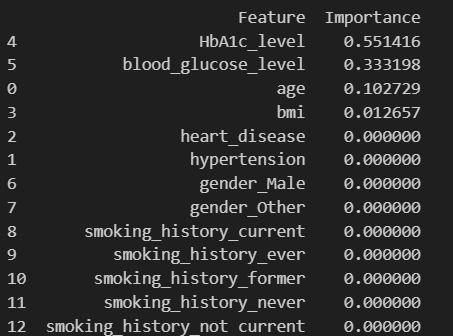
**ROC Curve and AUC**





**Feature Importance (Decision Tree )**





# 12. Insights and Findings

Logistic Regression generalized better, making it a suitable choice for deployment in healthcare applications.

Decision Tree, while useful, showed signs of overfitting and might need pruning or ensemble methods (Random Forest, XGBoost).

The results confirmed that **BMI, blood glucose, and HbA1c levels are strong predictors of diabetes**.

 In medical diagnosis, **recall is a critical metric** because false negatives (missed diabetic patients) can have serious consequences.

# 13. Conclusion

This project successfully built a **machine learning pipeline** for diabetes prediction. The dataset was preprocessed, cleaned, balanced, and analyzed. Logistic Regression emerged as the best-performing model due to its generalizability and balanced metrics.

**Key findings:**

* BMI, HbA1c, and glucose levels are significant predictors of diabetes.
* Data imbalance was a major challenge but was successfully addressed using SMOTE.
* Logistic Regression offered the best trade-off between performance and interpretability.

**Next steps:**

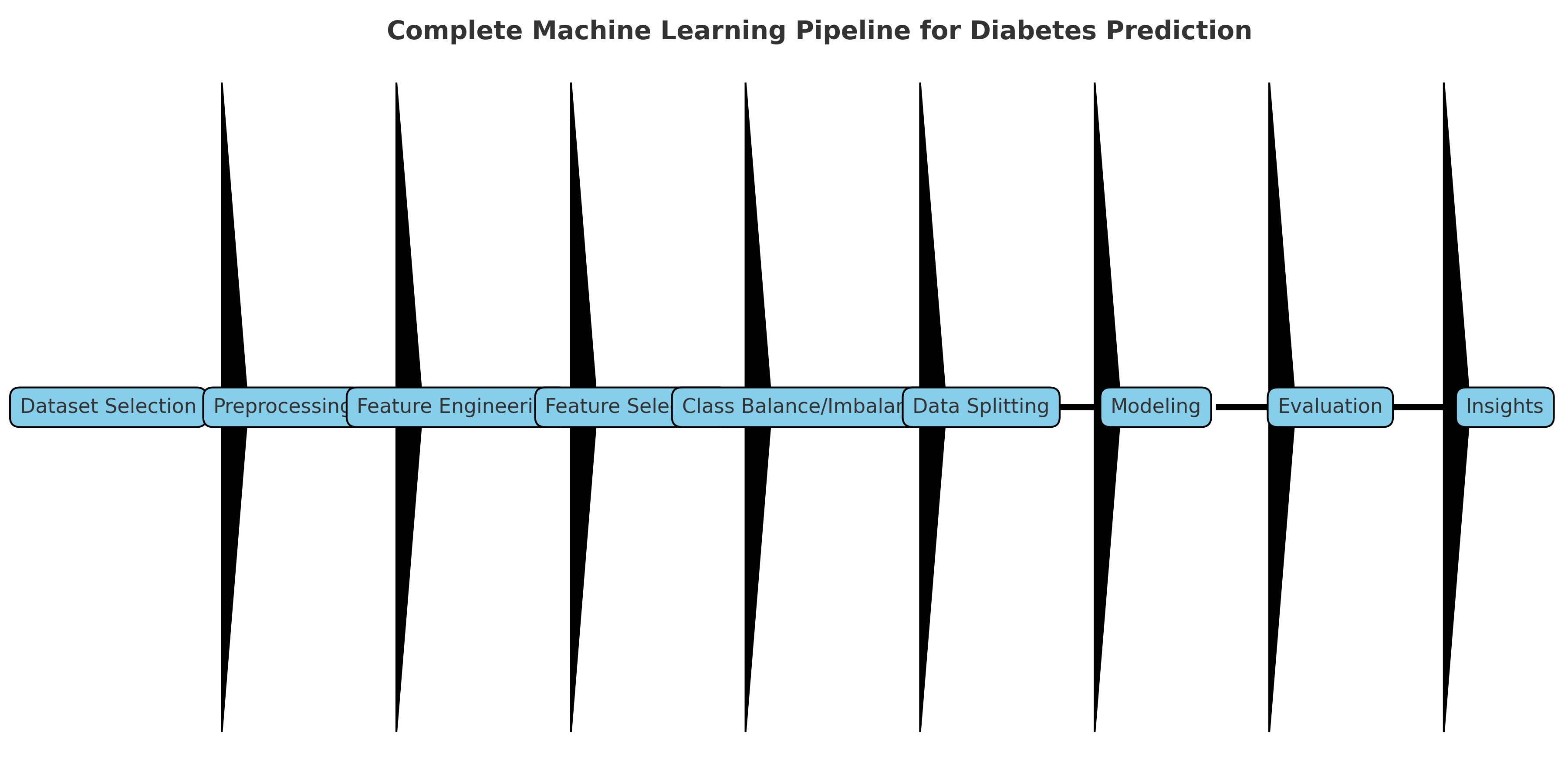
* Implement more advanced models like **Random Forest or XGBoost**.
* Perform **hyperparameter optimization** for Decision Trees.
* Use **feature selection or PCA** to reduce dimensionality and improve efficiency.
* Deploy the model in a real-world healthcare setting with continuous retraining as new data arrives.

# 14. Machine Learning Pipeline

Below diagram should illustrate the full pipeline:  
Dataset → Preprocessing → Feature Engineering → Class Imbalance Handling → Data Splitting → Model Training → Evaluation → Insights

# Appendix: Machine Learning Pipeline Diagram

The diagram below illustrates the complete ML pipeline followed in this project:



=