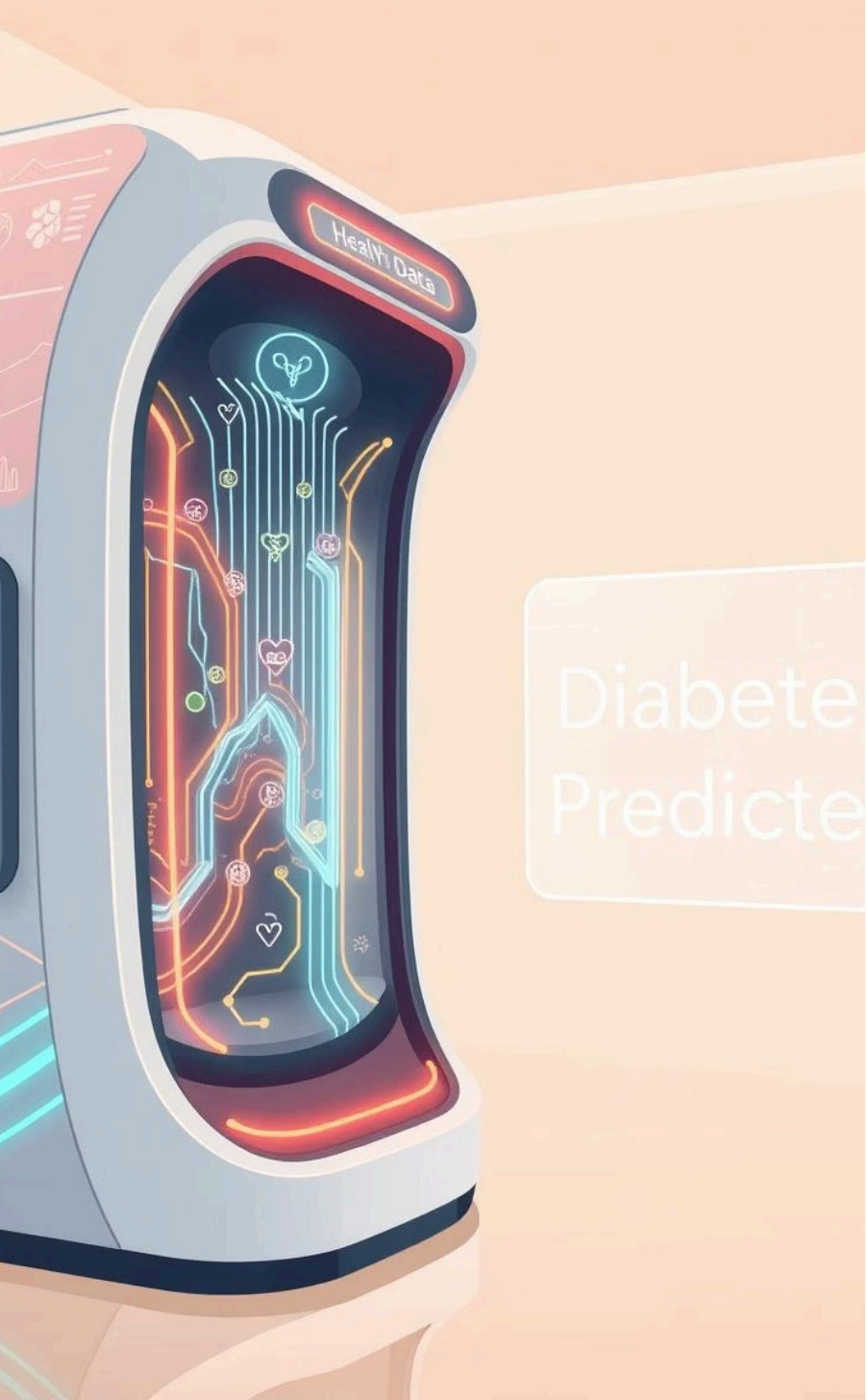


Diabetes Prediction Project

This project aims to develop a classification model for predicting diabetes likelihood using health data. We will explore the dataset, perform necessary cleaning and visualization, and build various machine learning models to achieve high accuracy in prediction.

Made by : Kavya Tripathi

Jahnavi Nagar

Krish Sandhu

Krish Gupta

Understanding Diabetes and Our Goal

Diabetes is a chronic condition where the body either doesn't produce enough insulin or can't effectively use the insulin it produces. Our objective is to classify whether an individual has diabetes (1) or not (0) based on a set of medical variables.

The dataset includes independent variables such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age, with 'Outcome' as the dependent variable.

Key Variables

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration
- BloodPressure: Diastolic blood pressure
- SkinThickness: Triceps skin fold thickness

Key Variables (Cont.)

- Insulin: 2-hour serum insulin
- BMI: Body Mass Index
- DiabetesPedigreeFunction: Diabetes likelihood score based on family history
- Age: Age in years

Data Exploration and Cleaning

Our initial steps involve understanding the dataset's structure, including its head, shape, column types, and a summary of statistical measures. This helps us identify potential issues like zero values in medically impossible fields.

Data cleaning focuses on dropping duplicate rows and checking for null values. Crucially, we replace medically impossible zero values in columns like Glucose, BloodPressure, SkinThickness, Insulin, and BMI with their respective mean or median values, depending on their distribution.

1 Understanding the Dataset

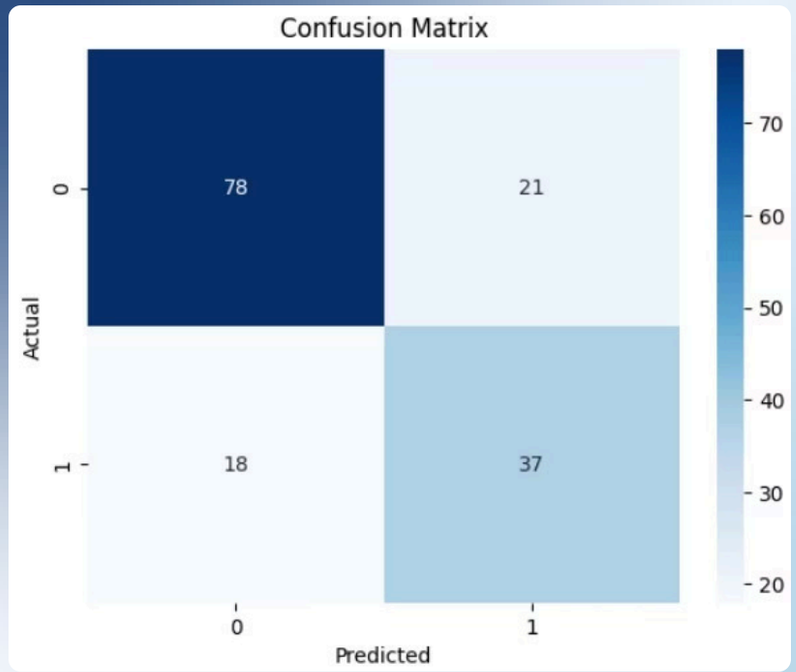
Reviewing dataset head, shape, column types, and summary statistics.

2 Data Cleaning

Dropping duplicates, checking for nulls, and replacing zero values in critical medical fields.

3 Addressing Imbalances

Recognizing and preparing to handle the imbalance where non-diabetic cases outnumber diabetic ones.



Visualizing Data Distributions

Data visualization is crucial for understanding the dataset's characteristics. We use count plots to assess data balance, histograms to check for normal or skewed distributions, and box plots to identify outliers.

Scatter plots and pair plots help us understand relationships between variables. For instance, histograms revealed that only Glucose and Blood Pressure are normally distributed, while others are skewed with outliers. Box plots further confirmed the presence of outliers, which need careful handling.

Count Plot

To check dataset balance.

Histograms

To analyze data distribution.

Box Plot

To identify outliers.

Scatter Plots

To understand variable relationships.

Feature Selection and Outlier Handling

Feature selection, guided by Pearson's Correlation Coefficient, helps identify the most relevant variables. A heatmap visualization shows that Glucose, BMI, and Age are most correlated with the 'Outcome' variable, while BloodPressure, Insulin, and DiabetesPedigreeFunction are less correlated and can be dropped.

Outliers are addressed using Quantile Transformation. This method spreads out frequent values and reduces outlier impact by transforming features to follow a uniform or normal distribution, ensuring better model performance without data loss.

Feature Selection

Using Pearson's Correlation to identify key variables like Glucose, BMI, and Age.

Heatmap Analysis

Visualizing correlation scores to determine feature importance.

Outlier Treatment

Applying Quantile Transformer to normalize distributions and reduce outlier impact.

Data Splitting for Model Training

After feature selection and outlier handling, the dataset is split into independent (X) and dependent (y) variables. 'Outcome' becomes our target (y), while all other selected columns form X.

The data is then divided into training and testing sets using an 80% train and 20% test split. This standard practice ensures that the model is trained on a significant portion of the data and evaluated on unseen data to assess its generalization capability.

1

Independent Variables (X)

Features used for prediction: Pregnancies, Glucose, SkinThickness, BMI, Age.

2

Dependent Variable (y)

Target variable: Outcome (0 or 1 for diabetes).

3

Train-Test Split

80% for training the model, 20% for evaluating its performance.

Classification Algorithm Overview

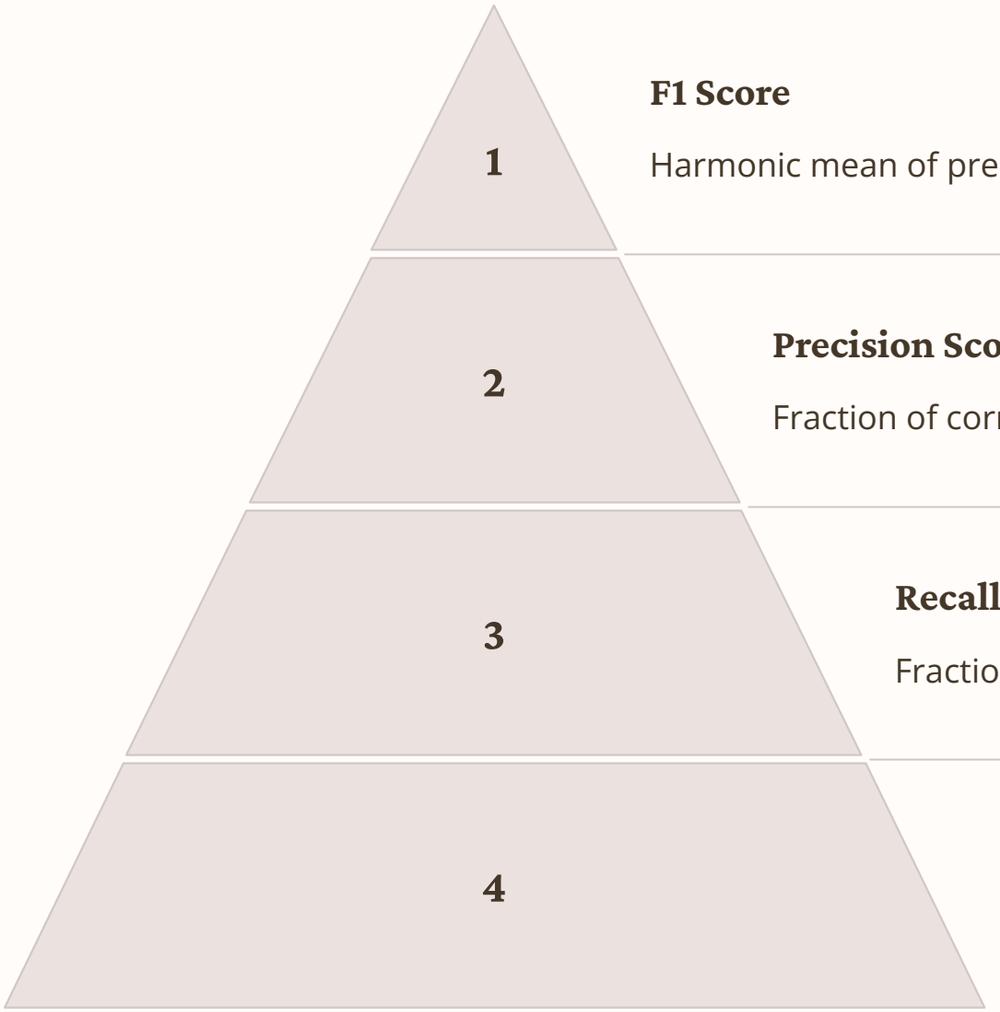
Logistic Regression

- Logistic Regression is a **supervised machine learning algorithm** used for **binary classification** problems, such as predicting whether a person has diabetes (1) or not (0).
- It models the relationship between the **independent variables** (medical features like Glucose, BMI, Age, Blood Pressure) and the **probability** of the dependent outcome (diabetes).
- The model uses the **sigmoid function**, which converts any real-valued number into a value between 0 and 1, representing the probability of belonging to a class.
- Based on a **decision threshold** (commonly 0.5), the model classifies the input as diabetic or non-diabetic.
- Logistic Regression is **easy to interpret** and provides insights into which features contribute most to the prediction.
- In this project, it serves as a **strong baseline model** to compare with other classifiers.

Model Performance Metrics

To evaluate our models, we use several key performance metrics: Confusion Matrix, F1 Score, Precision Score, and Recall Score. The Confusion Matrix provides a tabular visualization of predictions versus actual labels.

F1 Score is preferred due to the imbalanced nature of our dataset, as it offers a harmonic mean between precision and recall, providing a balanced measure of the model's accuracy, especially for the smaller positive class.



F1 Score

Harmonic mean of precision and recall, ideal for imbalanced classes.

Precision Score

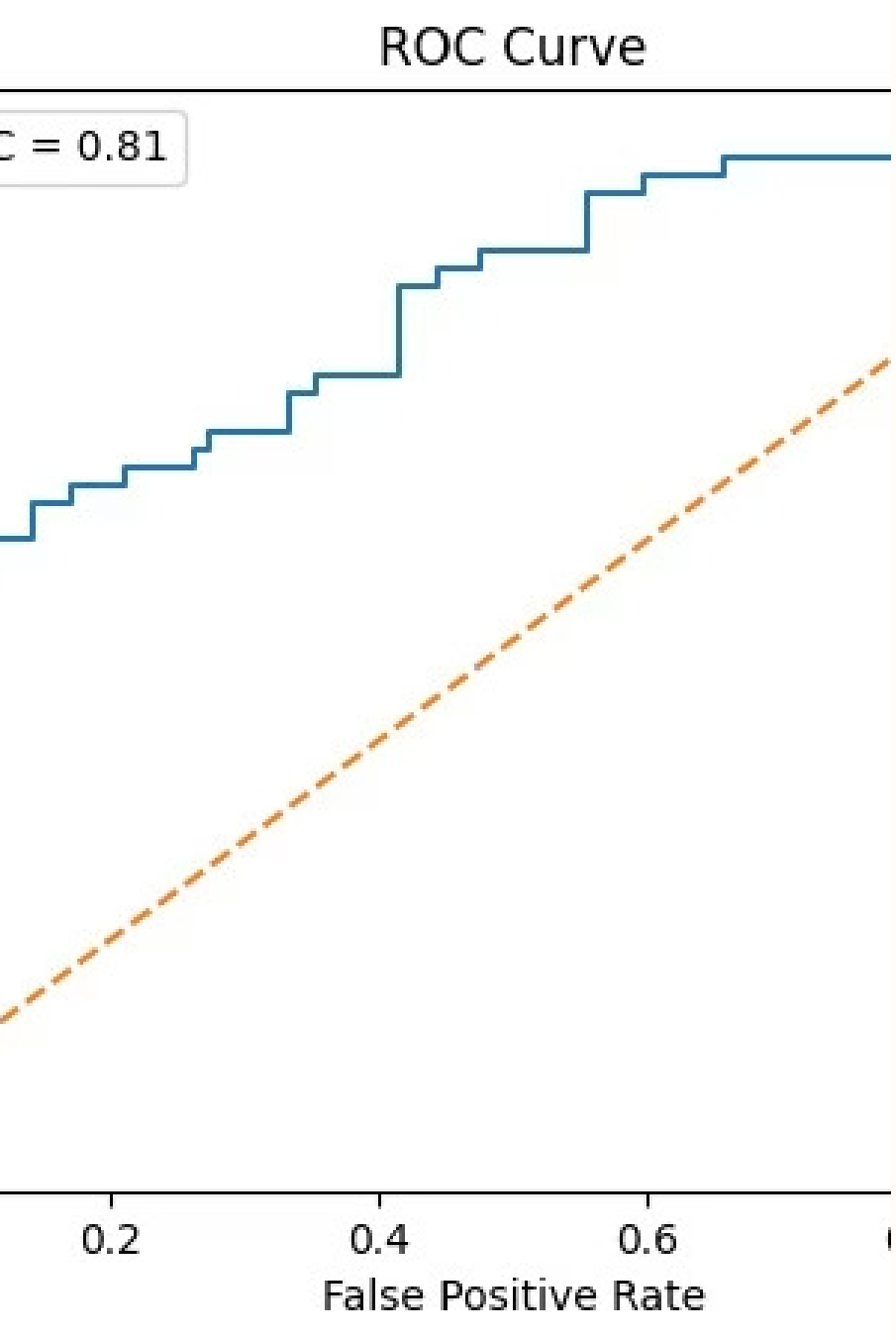
Fraction of correctly predicted positives among all predicted positives.

Recall Score

Fraction of correctly predicted positives among all actual positives.

Confusion Matrix

Visualizes true positives, true negatives, false positives, and false negatives.



Logistic Regression Results

For Logistic Regression, hyperparameter tuning was not extensively applied due to minimal accuracy improvements observed. The model was fitted on the training data, and predictions were made on the test set.

Our Website