Predicting Diabetes

By Abdelrahman Gamal

GET STARTED

Introduction

This presentation outlines our project focused on using machine learning to predict diabetes risk. By analyzing key health indicators, we developed a model that offers valuable insights for early detection and management of diabetes.

Goal of the Predictive Model

The primary goal is to create an accurate, user-friendly model for predicting diabetes risk based on critical health metrics. By identifying at-risk individuals early, we aim to provide actionable insights that can lead to improved health outcomes and informed decisions.

Dataset

Dataset Link: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

Number of samples : 100k

features: age - gender- hypertension-heart disease -smoking history-bmi-HbA1c level blood_glucose_level

Target: diabetes (0,1)

blood glucose level

The amount of sugar (glucose) in the blood at a specific time. It shows how well the body manages blood sugar and is important for diagnosing and managing diabetes.

bmi

(Body Mass Index): A number calculated from a person's weight and height. It's used to classify underweight, normal weight, overweight, or obesity.

HbA1c_level

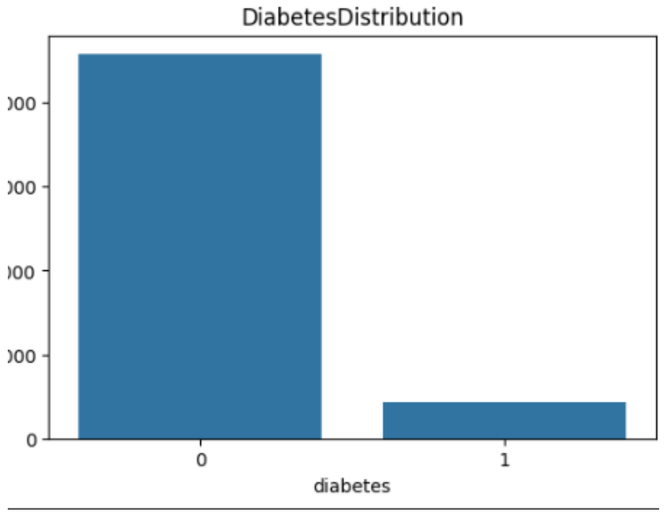
A blood test that shows the average blood sugar level over the past 2–3 months. It's used to diagnose and monitor diabetes.

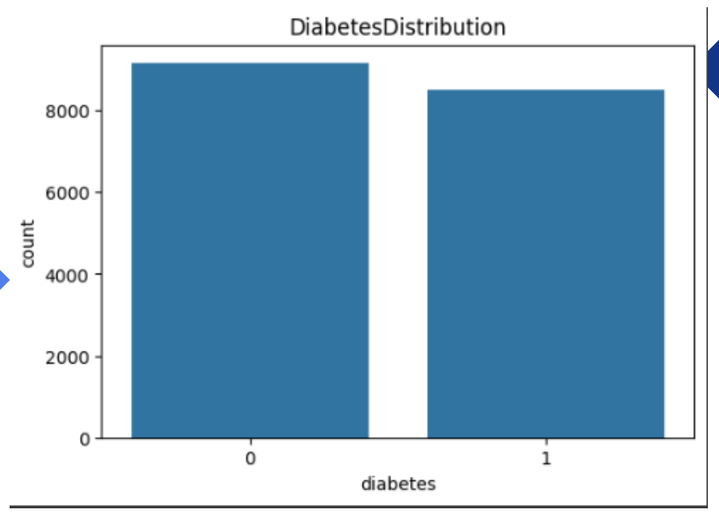
Challenge we faced

We had an issue with the data: 95 out of 100 samples had no sugar, which raises concerns about potential bias.

how we fix it

We solved it by taking 10% of the no-sugar samples and all the sugar samples, balancing the data.





Methods and techniques used

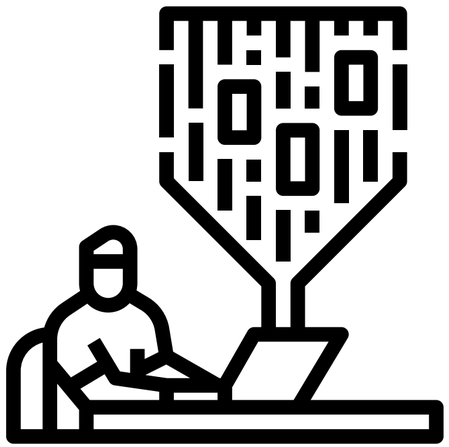
1.Label Encoder

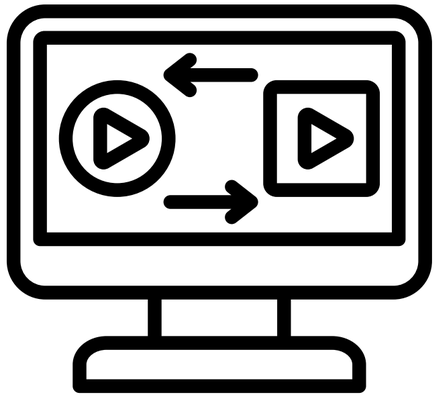
Label Encoder is a tool in machine learning used to convert categorical data (like text labels) into numbers so models can understand them. For example, if you have a column with categories like "red," "blue," and "green," Label Encoder assigns each a unique number: red=0, blue=1, green=2. This makes it easier for algorithms to process the data, but it assumes no order or importance between categories unless specified. It's simple but can introduce bias if the categories have no natural numerical relationship.

Methods and techniques used

2.OneHotEncoder

One-Hot Encoding is a technique to convert categorical data into a numerical format suitable for machine learning models. It creates binary (0 or 1) columns for each category, avoiding the ordinal bias of Label Encoding.





3. Scaling

We use Min-Max Scaling is a technique to transform numerical data into a specific range, usually [0, 1]. It works by taking each value in your dataset, subtracting the minimum value of the dataset, and dividing by the range (maximum value minus minimum value).Formula:

$$X_{scaled} = (X - X_{min}) / (X_{max} - X_{min})$$

Algorithms used

K-Nearest Neighbors (KNN)

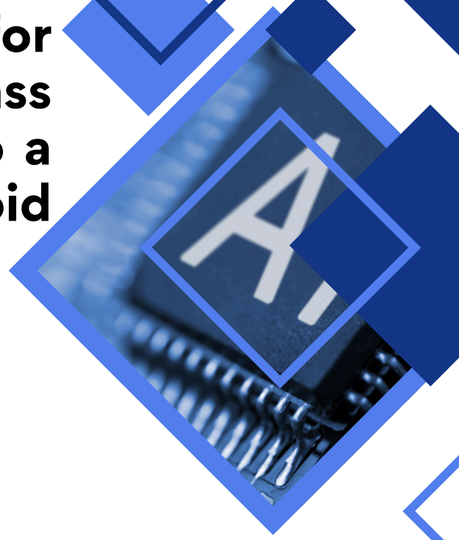
KNN is a simple, supervised machine learning algorithm used for classification and regression. It predicts a data point's label by finding the K closest data points (neighbors) in the training set based on a distance metric (e.g., Euclidean distance). For classification, it assigns the most common label among the K neighbors; for regression, it averages their values.

How it works

1. Choose K (number of neighbors).
2. Calculate distances from the new data point to all training points.
3. Select the K nearest points.
4. Predict based on majority vote (classification) or average (regression).

Logistic regression

Logistic Regression is a supervised machine learning algorithm used for binary classification (e.g., spam vs. not spam) or multi-class classification. It predicts the probability that a data point belongs to a specific class, outputting values between 0 and 1 using the sigmoid function.



How it works

1. Takes input features (e.g., age, income) and assigns weights to them.
2. Computes a linear combination: $z = w_1x_1 + w_2x_2 + \ldots + b$.
3. Applies the sigmoid function: $p = 1 / (1 + e^{-z})$ to get a probability.
4. Classifies based on a threshold (e.g., $p \geq 0.5 \rightarrow$ class 1, else class 0).

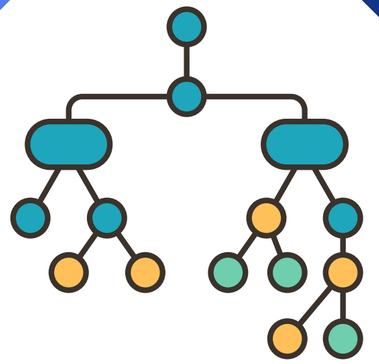
- Handles Linear Data Well
- Efficient
- Low Overfitting Risk

Decision Tree

A Decision Tree is a supervised machine learning algorithm used for classification and regression. It splits data into branches based on feature values, creating a tree-like structure to make decisions or predictions.

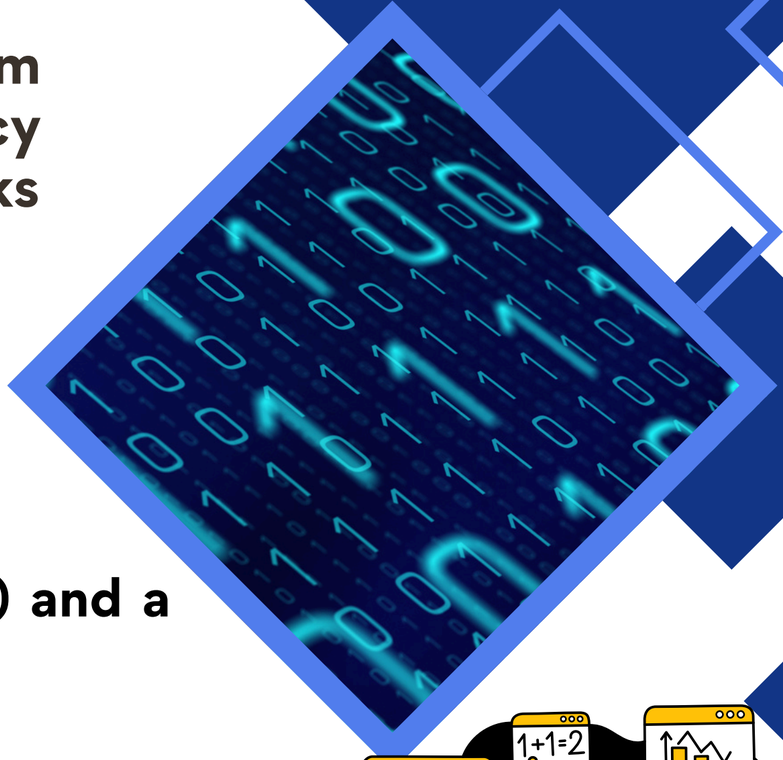
How it works

1. Start at the *root* node (entire dataset).
2. Split data based on the feature that best separates classes (e.g., using information gain).
3. Create *branches* for each possible value or range of the feature.
4. Repeat splitting at each node until reaching *leaf nodes* (final predictions).
5. Predict: For classification, assign the majority class; for regression, use the average value.



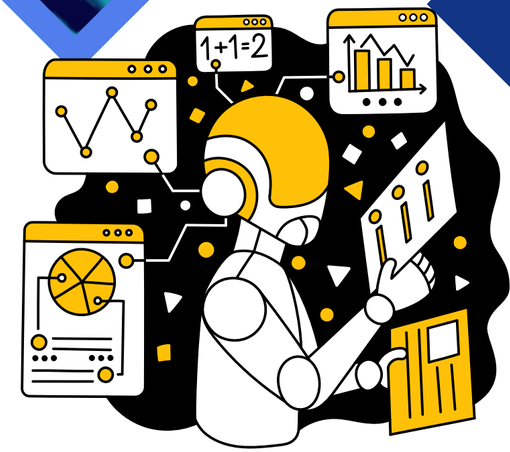
Random forest (best for my project)

Random Forest is a supervised machine learning algorithm that combines multiple Decision Trees to improve accuracy and robustness. It's used for classification and regression tasks



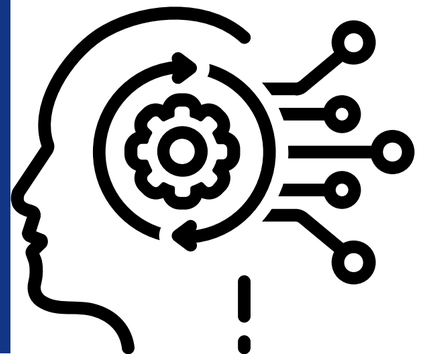
How it works

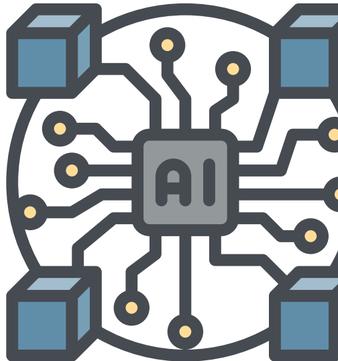
1. Creates a "forest" of many decision trees (e.g., 100 trees).
2. Each tree is trained on a random subset of the data (bagging) and a random subset of features.
3. For predictions:
-Classification: Majority vote from all trees.
-Regression: Average of all tree predictions.
4. Combines results to reduce overfitting and improve accuracy.



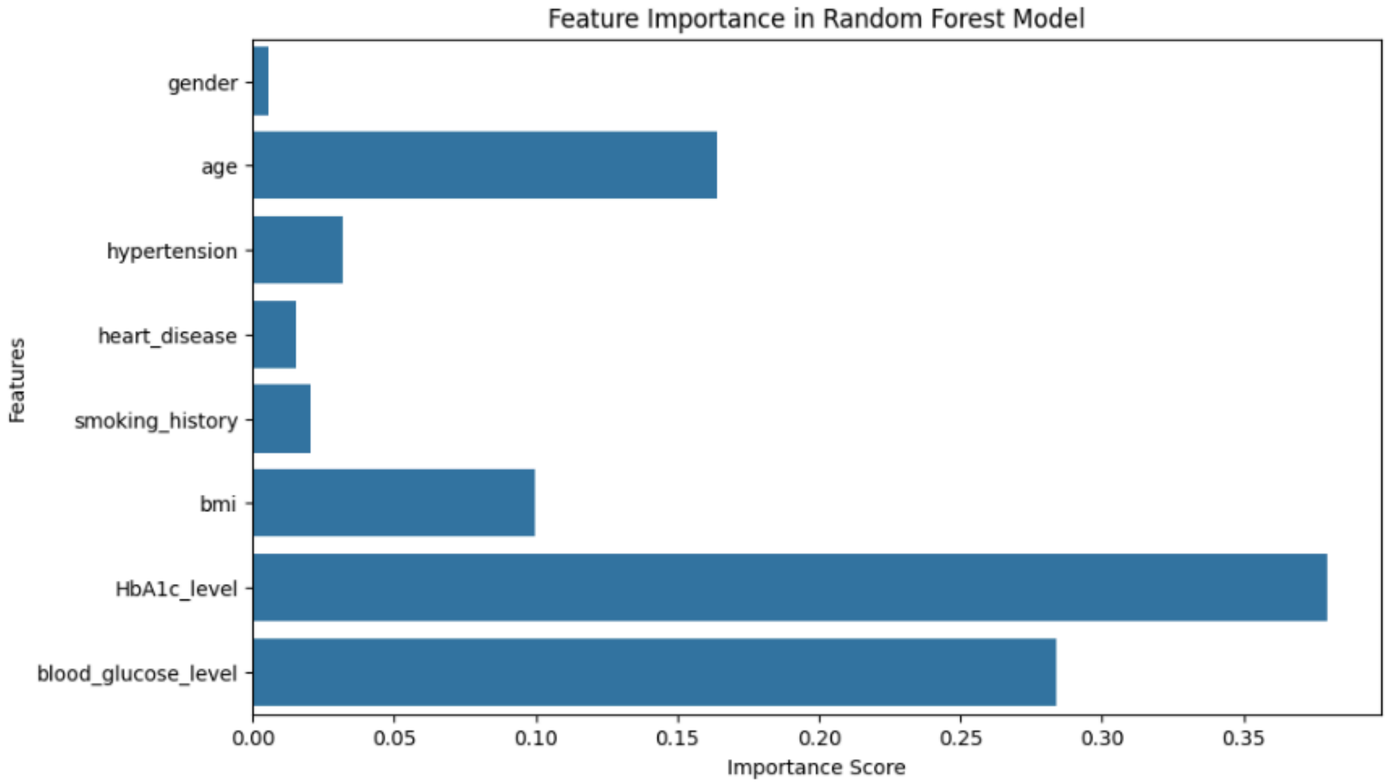
WHY Random forest

- Highly Accurate: Outperforms single decision trees by reducing variance.
- Robust to Overfitting: Randomization prevents overfitting compared to a single tree.
- Handles Non-Linear Data: No need for data scaling (unlike Logistic Regression or KNN).
- Feature Importance: Identifies which features matter most.
- Versatile: Works with categorical and numerical data, small or large datasets.





Feature Importance in Random Forest



we see in this image the most important features , It directly and highly affects our results, such as HbA1c ,Blood glucose level

Hyperparameter to Random forest

Hyperparameters are settings configured before training a machine learning model, like Random Forest, to control its behavior and performance. They are not learned from data.

Advantages

- Improves model accuracy and reduces overfitting.
- Balances speed and performance (e.g., fewer trees for faster training).
- Use techniques like Grid Search or Random Search to find optimal values.

Model Evaluation

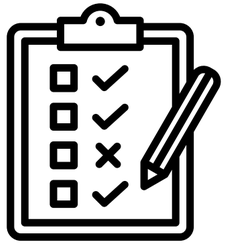
Accuarcy: % of correct predictions (good for balanced data)

Precision: Correct positive predictions / Total positive predictions.

Recall: Correct positive predictions / Total actual positives.

F1-Score: Balances precision and recall

Confusion Matrix: Shows true positives, false positives, true negatives, false negatives.



Thank You

For your attention! If you have any questions or would like to discuss further, please feel free to reach out.

+201552519691

abdelrahman.ga12@gmail.com

Egypt, Ismailia