**Aim**

The primary aim of this project is to develop a predictive model using the Random Forest classifier to determine the likelihood of diabetes in patients based on various health metrics. The goal is to provide healthcare professionals and institutions with a reliable tool that can assist in early diagnosis and intervention strategies for diabetes management.

**Problem Statement**

Diabetes is a significant global health issue that affects millions of people. Early diagnosis is crucial for effective management and treatment, reducing complications, and improving patient outcomes. The problem addressed in this project is to predict the presence of diabetes in patients using various health-related attributes, enabling timely medical intervention. The model will analyze key indicators such as glucose levels, blood pressure, and body mass index (BMI) to provide actionable insights.

h bullet points for clarity:

**Procedure**

* **Data Collection**:
  + The dataset used for this project is the Pima Indians Diabetes Database, which includes various health metrics for female patients of Pima Indian heritage.
* **Data Preprocessing**:
  + **Duplicate Removal**: The dataset is scanned for duplicates to ensure data integrity.
  + **Handling Missing Values**: Any missing values in the dataset are dropped to maintain a complete dataset for analysis.
* **Feature Selection**:
  + Features selected include:
    - Pregnancies
    - Glucose
    - BloodPressure
    - SkinThickness
    - Insulin
    - BMI
    - DiabetesPedigreeFunction
    - Age
  + The target variable is the presence of diabetes (0 = No Diabetes, 1 = Diabetes).
* **Data Splitting**:
  + The dataset is divided into training (70%) and testing (30%) sets to evaluate the model's performance.
* **Model Selection**:
  + A Random Forest Classifier is chosen for its robustness and ability to handle complex relationships in the data.
* **Hyperparameter Tuning**:
  + **Grid Search**: A grid search is conducted to find the optimal hyperparameters for the model, focusing on parameters such as:
    - Number of trees (n\_estimators)
    - Maximum depth of the trees (max\_depth)
    - Minimum samples required to split an internal node (min\_samples\_split)
    - Minimum samples required at a leaf node (min\_samples\_leaf)
    - Whether bootstrap samples are used (bootstrap)
* **Model Training**:
  + The Random Forest model is trained using the optimal hyperparameters identified through grid search.
* **Model Evaluation**:
  + The model’s performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix on the test dataset.
* **User Prediction**:
  + A predefined user input is used to demonstrate the model's predictive capabilities, allowing for immediate assessment of diabetes risk based on user health metrics.

This format should help in clearly outlining the procedure step by step

SOURCE CODE

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Load the dataset

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

df = pd.read\_csv(url, header=None, names=columns)

# Data preprocessing

df = df.drop\_duplicates()

df = df.dropna()

# Split features and target

X = df.drop(columns=['Outcome'])

y = df['Outcome']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Define the parameter grid for Random Forest

param\_grid = {

    'n\_estimators': [50, 100, 150, 200],

    'max\_depth': [10, 20, 30, None],

    'min\_samples\_split': [2, 5, 10],

    'min\_samples\_leaf': [1, 2, 4],

    'bootstrap': [True, False]

}

# Create a GridSearchCV object

grid\_search = GridSearchCV(estimator=RandomForestClassifier(random\_state=42), param\_grid=param\_grid, cv=5, scoring='accuracy', n\_jobs=-1)

# Fit the grid search to the data

grid\_search.fit(X\_train, y\_train)

# Get the best parameters and the best score

best\_params = grid\_search.best\_params\_

best\_score = grid\_search.best\_score\_

print("Best Parameters:", best\_params)

print("Best Cross-Validation Accuracy:", best\_score)

# Train the Random Forest with the best parameters

rf\_clf = RandomForestClassifier(\*\*best\_params, random\_state=42)

rf\_clf.fit(X\_train, y\_train)

# Predict on the test set

y\_pred\_rf = rf\_clf.predict(X\_test)

# Calculate the accuracy score

accuracy = accuracy\_score(y\_test, y\_pred\_rf)

print(f"Accuracy: {accuracy:.2f}")

# Print the classification report

report = classification\_report(y\_test, y\_pred\_rf, target\_names=["No Diabetes", "Diabetes"])

print("Classification Report:")

print(report)

# Print the confusion matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred\_rf)

print("Confusion Matrix:")

print(conf\_matrix)

# Predefined user input for prediction

user\_input = {

    'Pregnancies': 2,

    'Glucose': 85,

    'BloodPressure': 75,

    'SkinThickness': 30,

    'Insulin': 90,

    'BMI': 28.1,

    'DiabetesPedigreeFunction': 0.5,

    'Age': 25

}

user\_df = pd.DataFrame([user\_input])

user\_df = user\_df.reindex(columns=X.columns, fill\_value=0)

# Predict based on user input

user\_prediction\_rf = rf\_clf.predict(user\_df)

print("Random Forest Classifier Prediction:", "Diabetes" if user\_prediction\_rf[0] == 1 else "No Diabetes")

OUTPUT

Best Parameters: {'bootstrap': False, 'max\_depth': 10, 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100}

Best Cross-Validation Accuracy: 0.7802007615091726

Accuracy: 0.75

Classification Report:

precision recall f1-score support

No Diabetes 0.81 0.80 0.81 151

Diabetes 0.63 0.65 0.64 80

accuracy 0.75 231

macro avg 0.72 0.73 0.72 231

weighted avg 0.75 0.75 0.75 231

Confusion Matrix:

[[121 30]

[ 28 52]]

Random Forest Classifier Prediction: No Diabetes

**Inference**

From a business perspective, the development of this predictive model for diabetes risk assessment can significantly enhance healthcare delivery. By enabling early detection of diabetes, healthcare providers can implement preventive strategies, thereby improving patient outcomes and reducing long-term treatment costs associated with complications. This proactive approach allows for better allocation of resources by focusing on high-risk individuals, ultimately leading to more efficient healthcare operations. Furthermore, personalized health assessments can increase patient engagement, encouraging lifestyle modifications that may prevent the onset of diabetes. Overall, this model serves as a valuable tool for healthcare professionals, aiding in timely interventions and supporting ongoing research into diabetes risk factors.