Step 1: Import Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Step 2: Load the Dataset

```
from google.colab import files

uploaded = files.upload()
```

Choose Files diabetes.csv

- **diabetes.csv**(text/csv) - 23873 bytes, last modified: 9/23/2024 - 100% done

```
Saving diabetes.csv to diabetes (2).csv
```

```
# Load dataset
data = pd.read_csv('diabetes.csv')
print(data.head())
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

Step 3: Data Preprocessing

```
print(data.isnull().sum())

X = data.drop('Outcome', axis=1)
y = data['Outcome']
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Step 4: Train-Test Split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 5: Feature Scaling

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Step 6: Model Training

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Step 7: Model Evaluation

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

print(classification_report(y_test, y_pred))

conf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion Matrix:\n', conf_matrix)
```

```
Accuracy: 0.72
                precision    recall  f1-score   support

             0       0.79      0.78      0.78        99
             1       0.61      0.62      0.61        55

      accuracy                           0.72       154
     macro avg       0.70      0.70      0.70       154
  weighted avg       0.72      0.72      0.72       154

  Confusion Matrix:
   [[77 22]
   [21 34]]
```

Step 8: Making Predictions

```
def predict_diabetes(input_data):
    input_data = pd.DataFrame([input_data], columns=X.columns)  # Use the same column names as the training data
    input_data = scaler.transform(input_data)  # Scale the input data
    prediction = model.predict(input_data)
    return 'Diabetes' if prediction[0] == 1 else 'No Diabetes'

sample_input = [5, 166, 72, 19, 175, 25.8, 0.587, 51]  # Replace with actual values
result = predict_diabetes(sample_input)
print(result)
```

```
Diabetes
```