## Phase 3 Overview This Python code is designed to create an AI-based diabetes prediction system using a dataset with the following columns: 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', and 'Outcome'. It employs the Random Forest Classifier for prediction. Prerequisites Before using this code, ensure that you have the following libraries installed: - pandas - numpy - scikit-learn # import necessary libraries import pandas as pd import numpy as np

import pandas as pd
import numpy as np
from sklearn.model\_selection import train\_test\_split
from sklearn.ensemble import RandomForestClassifier
from wassklearn.metrics import accuracy\_score

dataset file. Load your dataset

1.Load Dataset

The code begins by loading your dataset from a CSV file.

Replace ""your\_dataset.csv" with the path to your

2. Define Feature and Target Columns

Specify the feature columns (X) and the target column

# Define feature columns (X) and target column (y)

The data is split into training and testing sets using

`train\_test\_split`. Adjust the `test\_size` and

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,

A Random Forest Classifier is used for building the

# Create a Random Forest Classifier (you can choose

The model is trained using the training data with

learning algorithm of your choice.

another algorithm if you prefer)

clf = RandomForestClassifier()

5. Model Training

`clf.fit(X\_train, y\_train)`.

prediction model. You can replace it with another machine

`random\_state` parameters as needed.

data = pd.read\_csv("your\_dataset.csv")

(y) based on your dataset's column names.

X = data[['Pregnancies', 'Glucose', 'BloodPressure',
'SkinThickness', 'Insulin', 'BMI',
'DiabetesPedigreeFunction', 'Age']]
y = data['Outcome']

3.Split Data

y, test\_size=0.2, random\_state=42)

4. Model Creation

# Fit the model to the training data clf.fit(X\_train, y\_train)

6. Make Predictions

# Make predictions on the test data

clf.predict(X\_test)

Predictions are made on the test data with

7. Evaluate Model Performance

8. Prediction for New Data

You can use the trained model to predict diabetes for new

data samples by calling `clf.predict(new\_data)` where

`new\_data` is a DataFrame with the same column names

The accuracy of the model is calculated and displayed. You can add more evaluation metrics as needed.

 $y_pred = clf.predict(X_test)$ 

for new data samples

12:17 ◎ 

Yether 5.31 4G fill □ 36%

import necessary iteraties
import pandas as pd
import numpy as np
from sklearn.model\_selection import train\_test\_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy\_score

data = pd.read\_csv("/content/drive/MyDrive/Colab Notebooks/diabetes.csv")

# Define feature columns (X) and target column (y)
X = data[{'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedi
y = data['Outcome']

# Now you can use this trained model to predict diabetes for new data samples # For example, you can use clf.predict(new\_data) where new\_data is a DataFrame with the same column n

# Split the data into training and testing sets
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, randon\_state=42)

# Create a Random Forest Classifier (you can choose another algorithm if you prefer)
clf = RandomForestClassifier()

Untitled1.ipynb

▶ # Import necessary libraries

clf.fit(X\_train, y\_train)

Accuracy: 74.68%

# Wake predictions on the test data y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)
print("Accuracy: {:.2f}%".format(accuracy \* 100))

+ Code + Text

 $\{x\}$ 

File Edit View Insert Runtime Tools Help Saving.

○ colab.research.google.com/

# Calculate the accuracy of the model

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

print("Accuracy: {:.2f}%".format(accuracy \* 100))

# Now you can use this trained model to predict diabetes

14

😃 Share 🌼 🥵

00 **4** • 1

Comment

≥ 2s completed at 12:16 PM

2023-10-28