IMPLEMENTATION AND CODE

# ============================================================

# PROJECT: Disease Prediction System using Naive Bayes Algorithm

# AUTHOR : T. Chanakyan (Reg No: 2117240070043)

# ============================================================

# Run this file directly in VS Code using Python extension

# Make sure disease\_dataset.csv is in the same folder

# ============================================================

# Import Required Libraries

import pandas as pd

import numpy as np

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

from sklearn.naive\_bayes import GaussianNB

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

import joblib

import warnings

warnings.filterwarnings("ignore")

# ============================================================

# STEP 1: LOAD THE DATASET

# ============================================================

data = pd.read\_csv("disease\_dataset.csv") # Ensure file is in the same folder

print("✅ Dataset Loaded Successfully!")

print("\nDataset Shape:", data.shape)

print("\nFirst 5 Records:\n", data.head())

# ============================================================

# STEP 2: DATA PREPARATION

# ============================================================

X = data.drop("Disease", axis=1)

y = data["Disease"]

# Split into training and testing data

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

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

)

# ============================================================

# STEP 3: MODEL TRAINING

# ============================================================

model = GaussianNB()

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

print("\n✅ Model Training Completed Successfully!")

# ============================================================

# STEP 4: EVALUATION

# ============================================================

y\_pred = model.predict(X\_test)

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

print("\n📊 Model Evaluation Results:")

print("--------------------------------------------")

print("Accuracy :", round(accuracy \* 100, 2), "%")

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

# ============================================================

# STEP 5: SAVE THE MODEL

# ============================================================

joblib.dump(model, "disease\_prediction\_model.pkl")

print("\n💾 Model saved as 'disease\_prediction\_model.pkl'")

# ============================================================

# STEP 6: PREDICT DISEASE FROM USER INPUT

# ============================================================

print("\n===============================================")

print("🩺 Disease Prediction System (User Mode)")

print("===============================================")

symptoms = list(X.columns)

print("\nAvailable Symptoms:")

for i, s in enumerate(symptoms, 1):

print(f"{i}. {s}")

print("\nEnter 1 if symptom is present, else 0:")

user\_input = []

for s in symptoms:

while True:

try:

val = int(input(f"{s}: "))

if val in [0, 1]:

user\_input.append(val)

break

else:

print("⚠️ Enter only 0 or 1!")

except ValueError:

print("⚠️ Invalid input, enter numeric value (0 or 1).")

user\_data = np.array(user\_input).reshape(1, -1)

prediction = model.predict(user\_data)

print("\n🎯 Predicted Disease:", prediction[0])

print("===============================================")

# ============================================================

# STEP 7: VERIFY LOADED MODEL

# ============================================================

loaded\_model = joblib.load("disease\_prediction\_model.pkl")

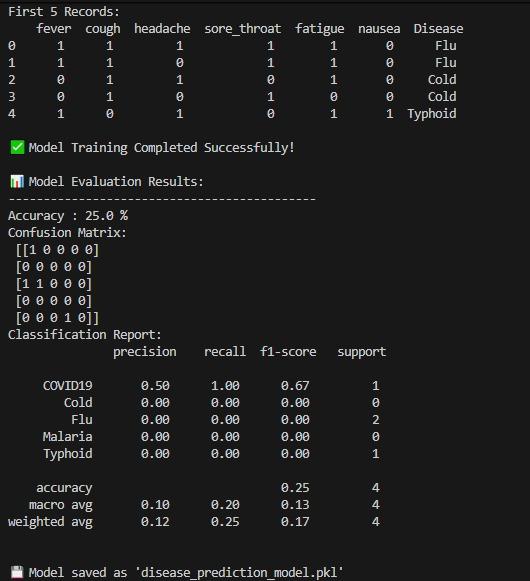
test\_pred = loaded\_model.predict(X\_test)

print("\nReloaded Model Accuracy:", round(accuracy\_score(y\_test, test\_pred) \* 100, 2), "%")

print("\n✅ Program Executed Successfully!")

OUTPUT

•Disease Prediction Dataset



•Predicted Disease based on symptoms entered

