Predict Disease Outcome Based on Genetic and Clinical Data

Name: Vansh Agarwal
Roll No: 2024011003002 73
Class: CSEAI-D
Course: Introduction to AI
Date: 4/18/2025

Introduction

The early and accurate prediction of diseases, especially cancer, is one of the key applications of Artificial Intelligence in healthcare. This project focuses on predicting the outcome of breast cancer—whether it is benign or malignant—using a dataset that combines genetic and clinical data.

The dataset used in this assessment is derived from the Wisconsin Diagnostic Breast Cancer (WDBC) dataset and includes features computed from digitized images of fine needle aspirate (FNA) of breast masses. Each sample is labeled either 'M' (Malignant) or 'B' (Benign), based on diagnosis.

Using machine learning algorithms, we aim to develop a model that can classify new patient data accurately, supporting medical professionals in early diagnosis and treatment planning.

Methodology

The following steps were taken to approach the problem:

1. Data Cleaning and Preprocessing

- **The dataset initially contained 33 columns and 569 records.**
- **Columns like `id` and `Unnamed: 32` were removed as they held no predictive value.**
- **The target column `diagnosis` was encoded into binary values:**
  - **M (Malignant) → 1**
  - **B (Benign) → 0**

2. Feature Scaling

- **Since the dataset contains features with different ranges, StandardScaler was applied to normalize the features. This ensures all input data is on the same scale, which helps improve the performance of machine learning models.**

3. Data Splitting

- **The dataset was split into:**
    - **Training set: 80%**
    - **Testing set: 20%**
- **The random seed was fixed for reproducibility (`random_state = 42`).**

4. Model Training

- **A Random Forest Classifier was selected because of its robustness, performance on tabular data, and ability to handle feature importance.**

- **The model was trained on the processed training data using default parameters.**

5. Model Evaluation

- **Predictions were made on the test set.**
- **Model performance was evaluated using:**
    - **Classification report (precision, recall, f1-score, support)**
    - **Confusion matrix for visual performance understanding**

Code

```
# Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.ensemble import RandomForestClassifier


# Load the dataset

df = pd.read_csv("/content/3. Predict Disease Outcome Based on Genetic and Clinical Data.csv")


# Drop unnecessary column

df.drop(columns=["Unnamed: 32", "id"], inplace=True)


# Encode the target column (M = 1, B = 0)

df['diagnosis'] = df['diagnosis'].map({'M': 1, 'B': 0})


# Split features and target
```

```
X = df.drop("diagnosis", axis=1)

y = df["diagnosis"]


# Standardize the features

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


# Split into train and test sets

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


# Train a Random Forest Classifier

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)


# Make predictions

y_pred = model.predict(X_test)


# Print performance report

print("Classification Report:\n", classification_report(y_test, y_pred))

print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))


# Plot confusion matrix
```

```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()
```

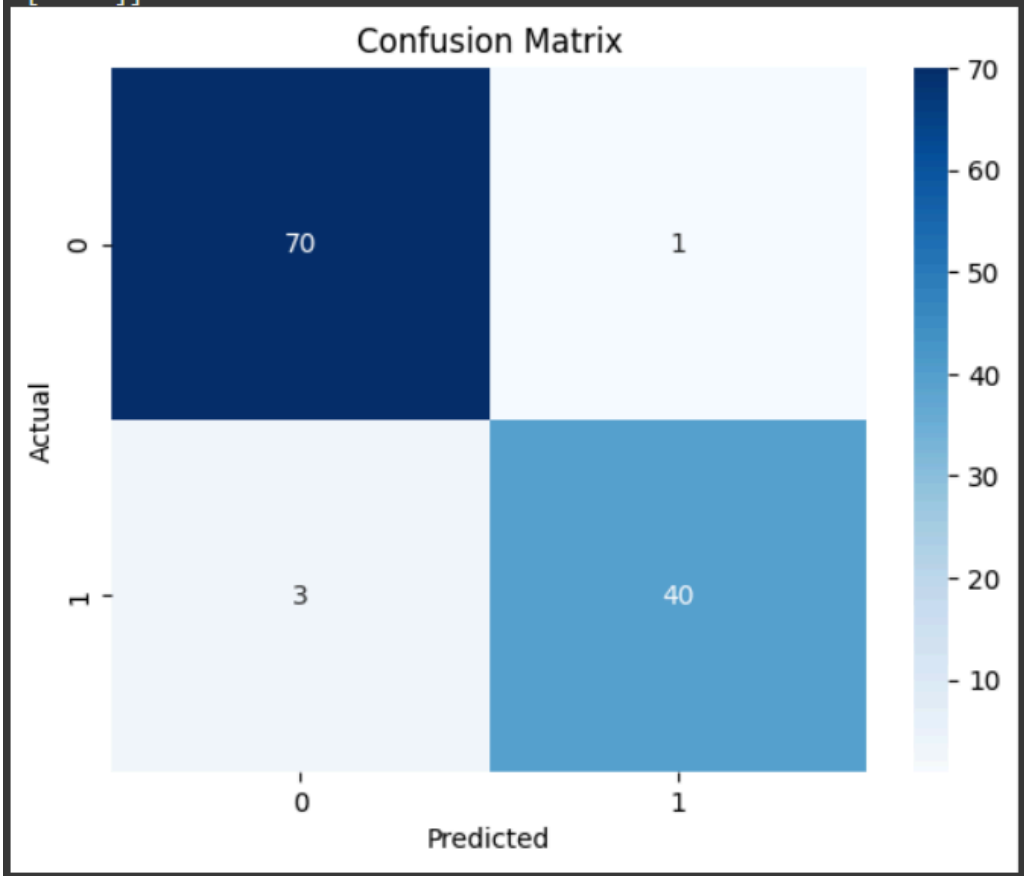
Output

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97        71
           1       0.98      0.93      0.95        43

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114


Confusion Matrix:
 [[70  1]
 [ 3 40]]
```



References / Credits

- **Dataset Source: UCI Machine Learning Repository – Breast Cancer Wisconsin Dataset**
- **Python Libraries Used:**
  - `pandas`, `numpy` **for data manipulation**
  - `matplotlib`, `seaborn` **for visualization**
  - `scikit-learn` **for machine learning models and evaluation**
- **Development Environment: Google Colab**
- **Model Used: Random Forest Classifier (from `sklearn.ensemble`)**