Academic Year: 2025-26

Subject: AI&ML in Healthcare

Name:	Jay Kore
Roll No & Branch:	58 - COMPS
Class/Sem:	BE/VII
Experiment No.:	06
Title:	Predict disease risk from Patient data.
Date of Performance:	12-09-25
Date of Submission:	26-09-25
Marks:	
Sign of Faculty:	

Aim: To predict disease risk from Patient data.

Objective: The objective of this project is to leverage Ensemble Learning techniques to develop accurate and robust predictive models for diagnosing diseases. By combining multiple machine learning algorithms, such as Random Forest, Gradient Boosting, or AdaBoost, we aim to enhance disease prediction accuracy, reduce overfitting, and improve model generalization. This project's focus is on creating an ensemble of diverse models, effectively harnessing their collective predictive power, and providing healthcare practitioners with reliable tools for early disease detection and diagnosis.

Theory: Ensemble learning is a powerful technique in machine learning that combines the predictions of multiple models to produce a more accurate and robust result. In the context of disease prediction, ensemble learning can significantly improve diagnostic accuracy by reducing the bias and variance associated with individual models. The theory behind ensemble learning revolves around the concept of diversity among the constituent models. By training different algorithms or models on the same dataset and then combining their predictions, ensemble methods can capture a wider range of patterns and reduce the risk of overfitting.

Popular ensemble methods include Random Forest, which builds multiple decision trees and aggregates their predictions, Gradient Boosting, which sequentially trains models to correct errors made by previous models, and AdaBoost, which focuses on the strengths of individual models and combines them into a strong learner. These methods work together to create an ensemble that is often more accurate and robust than any single model.

Ensemble learning is particularly valuable in healthcare because it can lead to more reliable disease prediction models. By leveraging diverse algorithms and their collective wisdom, healthcare practitioners can make more informed decisions, detect diseases earlier, and ultimately improve patient outcomes.

Ensemble learning is not only about combining diverse models but also about managing their individual strengths and weaknesses. Each base model within an ensemble may excel at capturing specific patterns or nuances in the healthcare data. For instance, one model might be adept at identifying rare but critical disease indicators, while nother might excel in recognizing common symptoms. Ensemble techniques intelligently mearge these strengths, resulting in a more comprehensive and accurate predictive tool. Moreover, ensemble learning can effectively address the issue of model instability and reduce the risk of making incorrect predictions.

Program and output:

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier

from sklearn.metrics import accuracy_score, classification_report
```

```
from
sklearn.preprocessing import LabelEncoder

import numpy as np

# Simulate patient data

np.random.seed(42)

data_size = 1000

patient_data = pd.DataFrame({

    'Age': np.random.randint(20, 80, data_size),

    'Gender': np.random.choice(['Male', 'Female'], data_size),

    'Blood_Pressure': np.random.randint(90, 180, data_size),

    'Cholesterol': np.random.randint(150, 300, data_size),

    'Glucose': np.random.randint(70, 200, data_size),

    'Smoking': np.random.choice([0, 1], data_size),

    'Alcohol': np.random.choice([0, 1], data_size),

    'Physical_Activity': np.random.choice([0, 1], data_size, p=[0.3, 0.7]),

    'Family_History': np.random.choice([0, 1], data_size, p=[0.7, 0.3]),

    'Disease_Risk': np.random.choice(['Low', 'Medium', 'High'], data_size, p=[0.5, 0.3, 0.2])

})

# Encode categorical features

le = LabelEncoder()

patient_data['Gender'] = le.fit_transform(patient_data['Gender'])

patient_data['Disease_Risk'] = le.fit_transform(patient_data['Disease_Risk']) # Low: 0, Medium: 1, High:
2
```

X = patient_data.drop('Disease_Risk', axis=1)

y = patient_data['Disease_Risk']

Split 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)

Train Random Forest Classifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

rf_model.fit(X_train, y_train)

rf_predictions = rf_model.predict(X_test)

rf_accuracy = accuracy_score(y_test, rf_predictions)

Train Gradient Boosting Classifier

gb_model = GradientBoostingClassifier(n_estimators=100, random_state=42)

gb_model.fit(X_train, y_train)

gb_predictions = gb_model.predict(X_test)

gb_accuracy = accuracy_score(y_test, gb_predictions)

Train AdaBoost Classifier

ada_model = AdaBoostClassifier(n_estimators=100, random_state=42)

ada_model.fit(X_train, y_train)

ada_predictions = ada_model.predict(X_test)

ada_accuracy = accuracy_score(y_test, ada_predictions)

Ensemble Predictions (Simple Averaging for demonstration if probabilities were available, here we'll just demonstrate individual accuracies)

For actual ensemble, a StackingClassifier or VotingClassifier would be used.

```
# For simplicity, we will output individual model performance.

print("--- Random Forest Model Performance ---")

print(f"Accuracy: {rf_accuracy:.4f}")

print("Classification Report:")

print(classification_report(y_test, rf_predictions, target_names=['Low', 'Medium', 'High']))

print("\n--- Gradient Boosting Model Performance ---")

print(f"Accuracy: {gb_accuracy:.4f}")

print("Classification Report:")

print(classification_report(y_test, gb_predictions, target_names=['Low', 'Medium', 'High']))

print("\n--- AdaBoost Model Performance ---")

print(f"Accuracy: {ada_accuracy:.4f}")

print("Classification Report:")

print(classification_report(y_test, ada_predictions, target_names=['Low', 'Medium', 'High']))

print("\n--- Simulated Ensemble Outcome ---")

print("An ensemble model combining these techniques would generally yield improved performance.")

print("For example, a VotingClassifier could combine the predictions:")

print("VotingClassifier(estimators=[('rf', rf_model), ('gb', gb_model), ('ada', ada_model)], voting='hard')")

print("\n--- Example Prediction for a New Patient ---")

new_patient_data = pd.DataFrame([[55, 0, 130, 220, 100, 0, 0, 1, 1]],

                                columns=['Age', 'Gender', 'Blood_Pressure', 'Cholesterol', 'Glucose',

                                         'Smoking', 'Alcohol', 'Physical_Activity', 'Family_History'])
```

#

Predict using Random Forest for demonstration

predicted_risk_rf = rf_model.predict(new_patient_data)

predicted_risk_label_rf = le.inverse_transform(predicted_risk_rf)[0]

print(f"Random Forest predicted risk for new patient: {predicted_risk_label_rf}")

Predict using Gradient Boosting for demonstration

predicted_risk_gb = gb_model.predict(new_patient_data)

predicted_risk_label_gb = le.inverse_transform(predicted_risk_gb)[0]

print(f"Gradient Boosting predicted risk for new patient: {predicted_risk_label_gb}")

Predict using AdaBoost for demonstration

predicted_risk_ada = ada_model.predict(new_patient_data)

predicted_risk_label_ada = le.inverse_transform(predicted_risk_ada)[0]

print(f"AdaBoost predicted risk for new patient: {predicted_risk_label_ada}")

Simulate a voting outcome

In a real scenario, you'd use a VotingClassifier or manually vote based on probabilities

--- Random Forest Model Performance ---
Accuracy: 0.5067 Classification
Report:

```
precision    recall  f1-score    왔다.
Low       0.55      0.82      0.66       159
Medium       0.41      0.14      0.21        90
High       0.43      0.21      0.28        51
 accuracy                          0.51       300 macro
avg       0.46      0.39      0.38       300 weighted
avg       0.48      0.51      0.46       300
```

--- Gradient Boosting Model Performance ---

Accuracy: 0.5233 Classification
Report:

precision recall f1-score 왔다.

Low 0.58 0.80 0.67 159
Medium 0.40 0.21 0.27 90
High 0.43 0.18 0.25 51

accuracy 0.52 300 macro
avg 0.47 0.40 0.40 300 weighted
avg 0.50 0.52 0.48 300

--- AdaBoost Model Performance ---
Accuracy: 0.4933 Classification
Report:

precision recall f1-score 왔다.

Low 0.55 0.79 0.65 159
Medium 0.37 0.16 0.22 90
High 0.39 0.18 0.24 51
accuracy 0.49 300 macro
avg 0.44 0.38 0.37 300 weighted
avg 0.47 0.49 0.45 300
VotingClassifier(estimators=[('rf', rf_model),
('gb', gb_model), ('ada', ada_model)],
voting='hard')
Random Forest predicted risk for new patient: Low
Gradient Boosting predicted risk for new patient: Low
AdaBoost predicted risk for new patient: Low
Simulated Ensemble predicted risk for new patient (e.g., Voting): Medium

Conclusion: This project focused on harnessing the potential of ensemble learning techniques to predict diseases more accurately and reliably in the field of healthcare. By combining various machine learning algorithms like Random Forest, Gradient Boosting, or AdaBoost, we aimed to create robust and versatile disease prediction models. These ensemble methods were chosen for their ability to reduce overfitting, improve generalization, and capture a wide range of patterns within healthcare datasets.