Assessment Report

on

“Predict Disease Outcome Based on Genetic and Clinical Data”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

Nikhil Chandra

Roll no: 20240110040026

CSE - AIML

Under the supervision of

“Mr. Abhishek Shukla”

🧬 Report: Disease Outcome Prediction Using Genetic and Clinical Data

🔍 Introduction

Early disease detection, particularly for life-threatening conditions such as cancer, is crucial for increasing treatment success rates and improving patient outcomes. Machine learning (ML) techniques have shown significant promise in identifying disease patterns by analyzing large volumes of genetic and clinical data. This project focuses on using supervised and unsupervised ML approaches to predict disease outcomes based on features extracted from the Breast Cancer Wisconsin dataset. The primary goal is to classify patients as either at risk (malignant) or not at risk (benign) and to explore clustering patterns that may reveal hidden insights in the dataset. By leveraging the power of machine learning, clinicians and researchers can make more informed, data-driven decisions

.

□ Methodology

This study uses the Breast Cancer Wisconsin (Diagnosis) dataset, comprising 569 patient records with 30 numeric features derived from digitized images of breast mass samples. Data preprocessing included dropping non-informative columns and encoding the target variable ('diagnosis') into binary form. The dataset was then split into training and testing sets in an 80:20 ratio.

A Random Forest Classifier was chosen for its robustness and ability to handle feature interactions. The model was trained on the training set and evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve. A confusion matrix was plotted to visualize prediction accuracy. Additionally, feature importance scores were extracted to identify key predictors.

evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve. A confusion matrix was plotted to visualize prediction accuracy. Additionally, feature importance scores were extracted to identify key predictors.

For unsupervised learning, the dataset was scaled using StandardScaler, and KMeans clustering was applied with k=2 to reflect the binary nature of the diagnosis labels. The resulting clusters were visualized using the first two dimensions of the standardized.

Code of Program

```
# Import libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
precision_score, recall_score, roc_curve, auc

from sklearn.cluster import KMeans


# Load dataset

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


# Drop unnecessary columns

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


# Encode target: M = 1, B = 0

df["diagnosis"] = LabelEncoder().fit_transform(df["diagnosis"])


# Split features and target
```

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

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# --- Evaluation Metrics ---
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("\nClassification Report:\n", classification_report(y_test, y_pred,
target_names=["Benign", "Malignant"]))
```

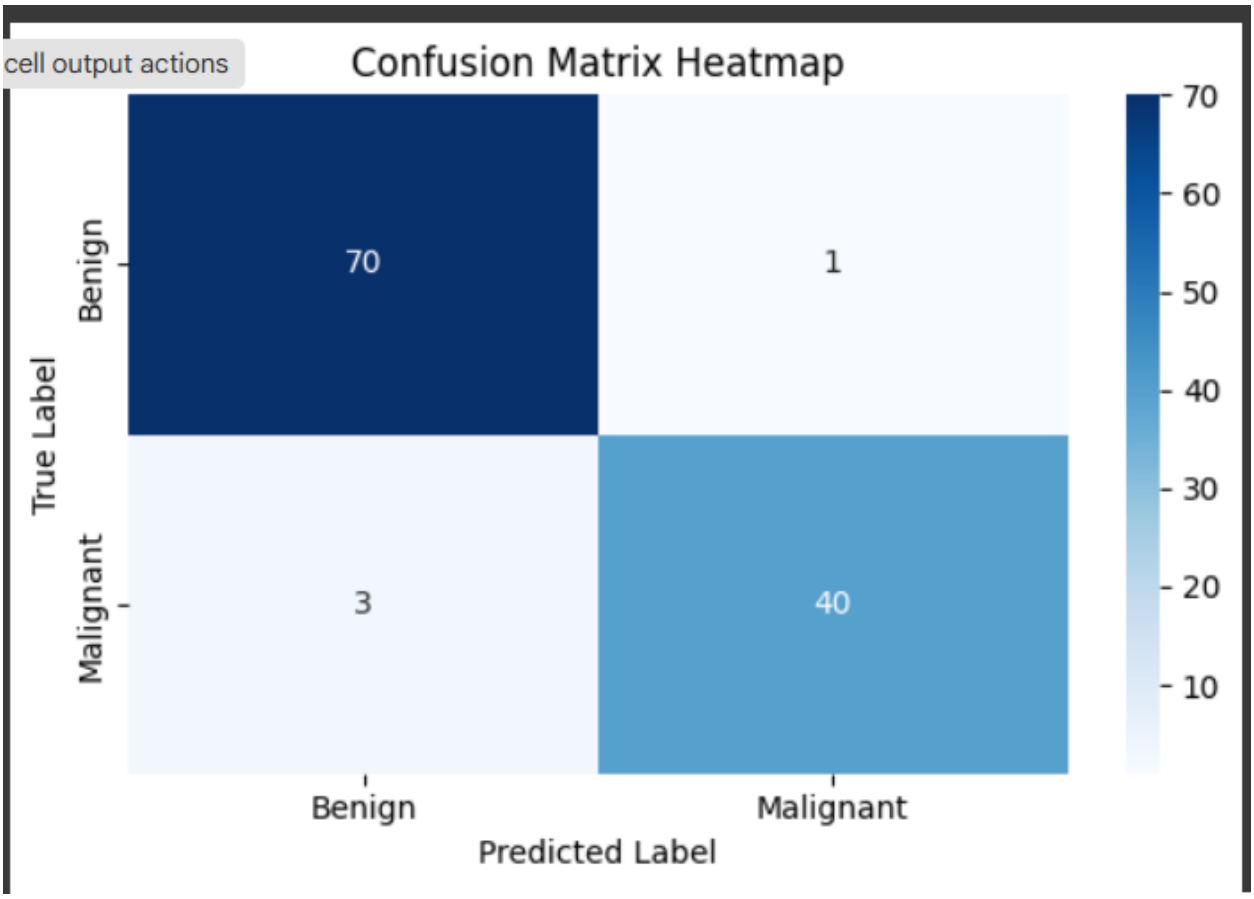
```
# --- Confusion Matrix Heatmap ---
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Benign", "Malignant"],
            yticklabels=["Benign", "Malignant"])
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.tight_layout()
plt.show()

# --- ROC Curve ---
y_probs = model.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
```

```
plt.title('ROC Curve')

plt.legend(loc='lower right')

plt.tight_layout()

plt.show()


# --- Feature Importance ---

feature_importances = pd.Series(model.feature_importances_,
index=X.columns).sort_values(ascending=False)


plt.figure(figsize=(8, 5))

sns.barplot(x=feature_importances, y=feature_importances.index)

plt.title("Top Feature Importances (Random Forest)")

plt.xlabel("Importance Score")

plt.ylabel("Feature")

plt.tight_layout()

plt.show()


# --- Clustering (KMeans) ---

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


kmeans = KMeans(n_clusters=2, random_state=42)

clusters = kmeans.fit_predict(X_scaled)
```

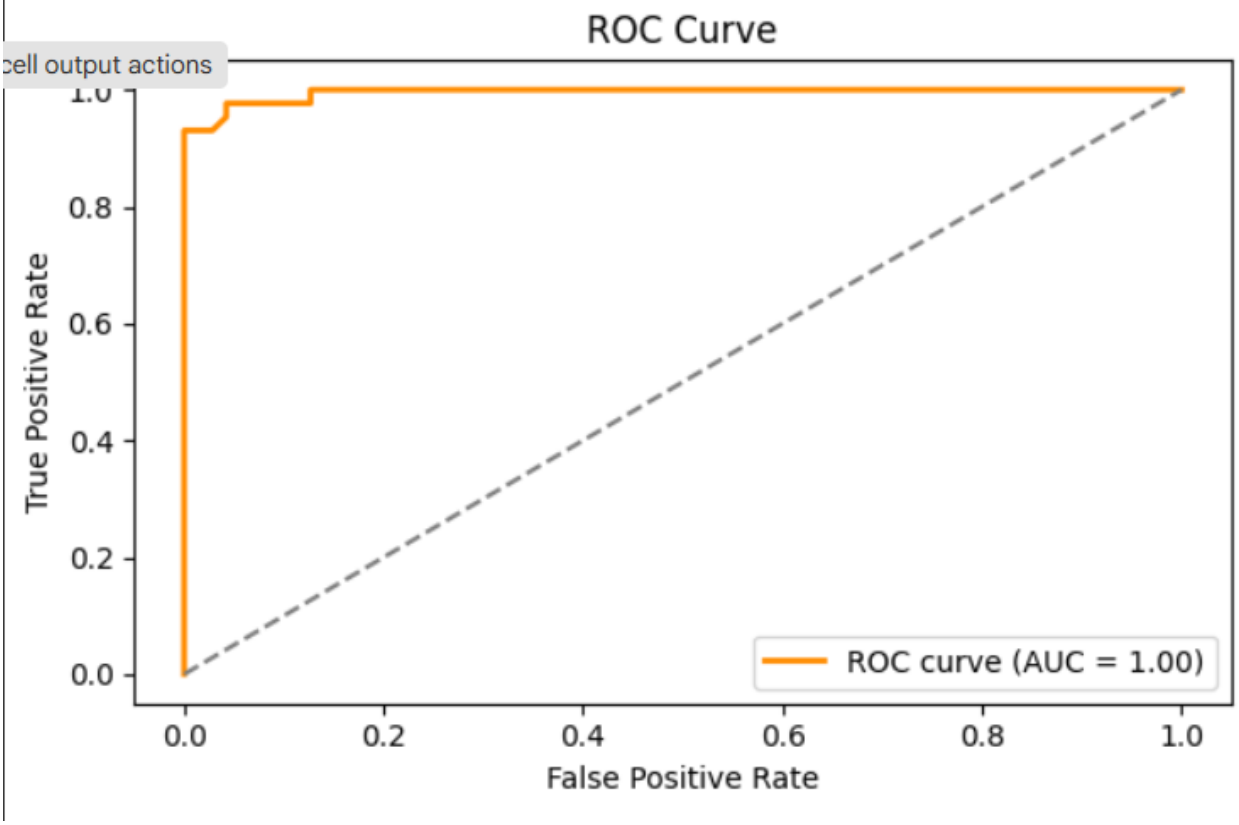
```
# Visualize first 2 PCA-like features after clustering

plt.figure(figsize=(6, 5))

sns.scatterplot(x=X_scaled[:, 0], y=X_scaled[:, 1], hue=clusters, palette='Set1',
alpha=0.7)

plt.title("KMeans Clustering (k=2)")

plt.xlabel("Feature 1 (scaled)")

plt.ylabel("Feature 2 (scaled)")

plt.legend(title="Cluster")

plt.tight_layout()

plt.show()
```

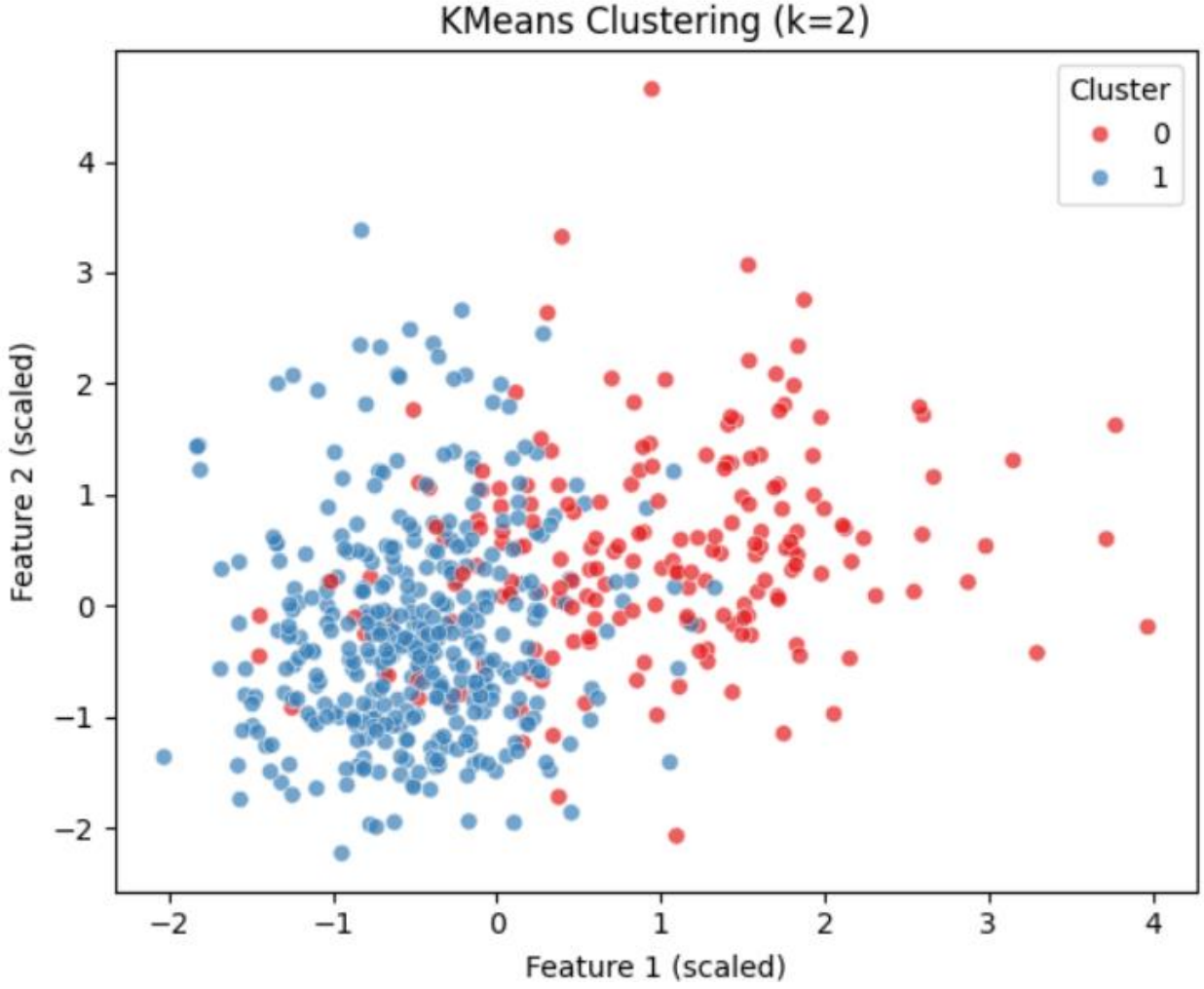
Output/Result

```
Accuracy: 0.9649122807017544
Precision: 0.975609756097561
Recall: 0.9302325581395349

Classification Report:
              precision    recall  f1-score   support

      Benign       0.96      0.99      0.97        71
   Malignant       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
```


cell output actions
Confusion Matrix Heatmap
70
1
3
40
Benign
Malignant
True Label
Benign
Malignant
Predicted Label
70
60
50
40
30
20
10

cell output actions




KMeans Clustering (k=2)
Cluster
0
1
Feature 2 (scaled)
Feature 1 (scaled)
4
3
2
1
0
−1
−2
−2
−1
0
1
2
3
4