www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

A

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

Computer Science and Engineering (Artificial Intelligence)

By

Sudhanshu kumar(202401100300253)

Under the supervision of

Mr. Abhishek shukla

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Breast cancer remains one of the most prevalent and life-threatening diseases among women worldwide. Early and accurate diagnosis is critical to improving survival rates and ensuring effective treatment. Traditional diagnostic procedures, such as imaging and biopsy analysis, are often time-consuming and subject to human interpretation. In recent years, the integration of machine learning into healthcare has emerged as a promising approach to augment diagnostic accuracy and efficiency.

This study leverages the **Breast Cancer Wisconsin Diagnostic Dataset**, which consists of 30 numerical features extracted from digitized images of fine needle aspirate (FNA) of breast masses. These features capture various characteristics of the cell nuclei, including radius, texture, perimeter, area, and other shape-related attributes. The primary goal is to build a predictive model capable of classifying tumors as **malignant (cancerous)** or **benign (non-cancerous)** based on these clinical features.

A **Logistic Regression** model is employed due to its simplicity, interpretability, and effectiveness in binary classification problems. The workflow involves data cleaning, normalization, training, and performance evaluation using metrics such as accuracy, precision, recall, F1-score, and the confusion matrix. The outcomes of this study demonstrate the practical value of machine learning techniques in supporting clinical decision-making and enhancing diagnostic accuracy in oncology.

Methodology

This section outlines the steps followed to develop a predictive model for classifying breast tumors as malignant or benign using machine learning techniques. The methodology includes data acquisition, preprocessing, model selection, training, and evaluation.

1. Data Acquisition

The dataset used in this study is the **Breast Cancer Wisconsin Diagnostic Dataset**, which contains 569 patient records. Each record includes 30 numeric features derived from digitized images of fine needle aspirate (FNA) of breast masses, along with a target label:

- **Diagnosis:** `"M"` (Malignant) or `"B"` (Benign).

The dataset was loaded into the Python environment via Google Colab, using a manual file upload mechanism for flexibility and portability.

---

2. Data Preprocessing

To prepare the data for machine learning, several preprocessing steps were performed:

- **Column Removal:** The non-informative `id` column and an empty `Unnamed: 32` column were dropped.
- **Label Encoding:** The categorical target values were mapped to binary format: `M = 1`, `B = 0`.
- **Feature Scaling:** All feature values were standardized using **StandardScaler** from `sklearn.preprocessing` to ensure that each feature contributes equally to the model's learning.
- **Train-Test Split:** The data was split into training (80%) and testing (20%) sets using `train_test_split`, ensuring randomization and reproducibility with a fixed `random_state`.

---

3. Model Selection and Training

A **Logistic Regression** model was selected for its simplicity, efficiency, and effectiveness in binary classification tasks. It also provides interpretable coefficients, making it suitable for understanding which features influence the prediction.

Key parameters:

- `max_iter = 1000` to ensure convergence
- `random_state = 42` for reproducibility

The model was trained on the scaled training data using `model.fit()`.

---

4. Model Evaluation

After training, the model's performance was evaluated using the test set. The following metrics were computed:

- **Accuracy:** Overall correctness of the model.
- **Precision, Recall, F1-score:** Performance per class (malignant vs benign).
- **Confusion Matrix:** To visualize true positives, true negatives, false positives, and false negatives.

These metrics were generated using `classification_report()` and `confusion_matrix()` from `sklearn.metrics`.

This methodology ensured a clean and efficient machine learning pipeline, enabling accurate predictions while maintaining interpretability—crucial for real-world clinical applications.

Code

```
# STEP 1: Upload the CSV file
from google.colab import files
uploaded = files.upload()

# STEP 2: Import libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
import seaborn as sns
import matplotlib.pyplot as plt

# STEP 3: Load and preprocess the dataset
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

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

# Encode target variable
df["diagnosis"] = df["diagnosis"].map({"M": 1, "B": 0})
```

```
# Split into features and target
X = df.drop(columns=["diagnosis"])
y = df["diagnosis"]

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

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

# STEP 4: Train Logistic Regression model
model = LogisticRegression(max_iter=1000, random_state=42)
model.fit(X_train, y_train)

# STEP 5: Make predictions
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]

# STEP 6: Evaluation - Text
print("✅Logistic Regression Training Complete!\n")
print("□ Classification Report:\n", classification_report(y_test, y_pred))
print("⍰ Confusion Matrix:\n", confusion_matrix(y_test, y_pred))


# STEP 7: Plot Confusion Matrix
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.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("⍰ Confusion Matrix")
plt.show()


# STEP 8: Plot ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)


plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], 'k--')  # Diagonal line
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("⍰ ROC Curve")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Output/Result

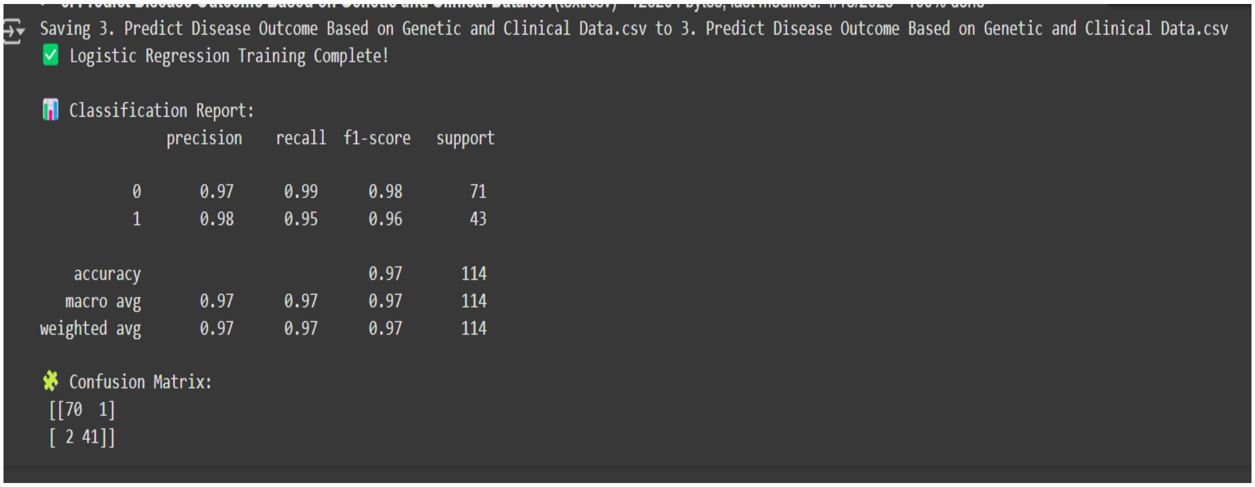
The trained **Logistic Regression model** was evaluated using a separate test set, comprising 20% of the original dataset. The objective was to predict whether a tumor is **Malignant (1)** or **Benign (0)** based on 30 numerical features derived from clinical and genetic characteristics.

⍰ Classification Report:

```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        71
           1       0.98      0.95      0.96        43

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114
```

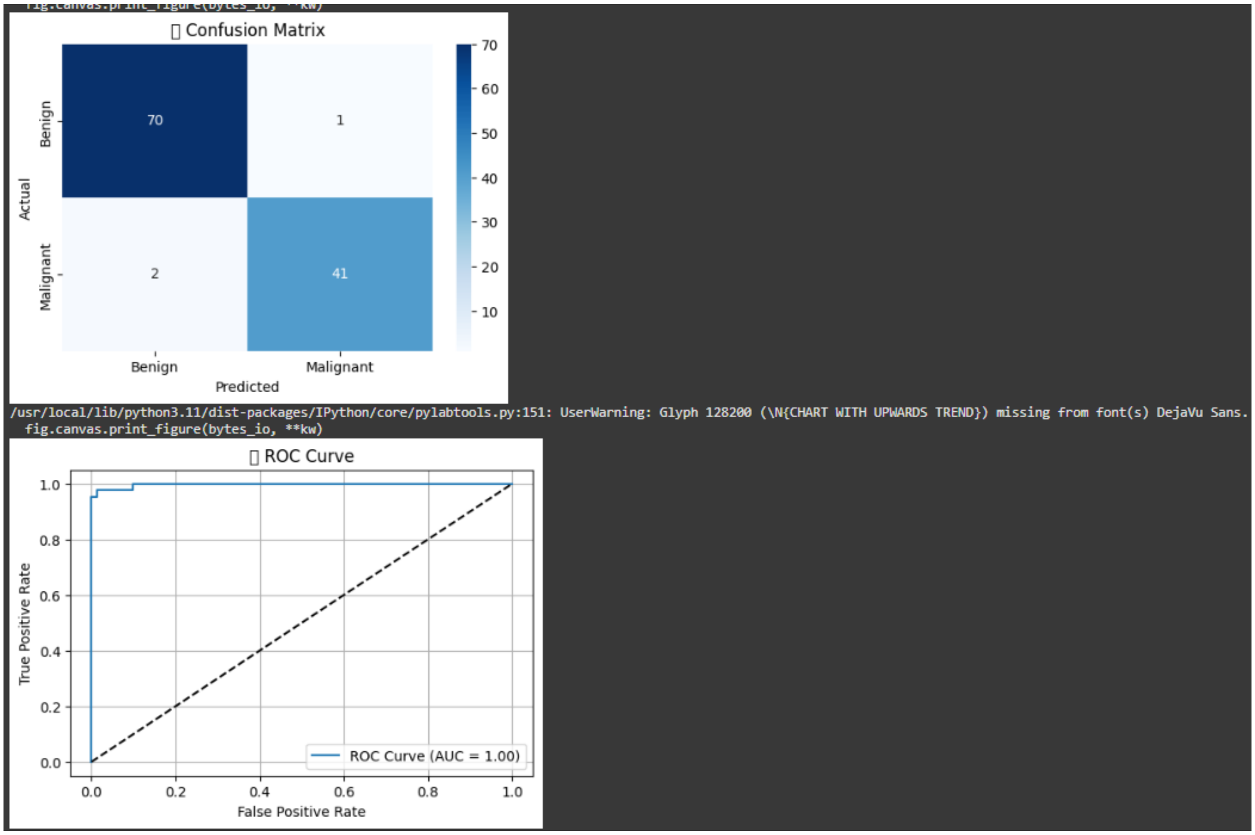
⍰ Confusion Matrix:

```
 [[70  1]
 [ 2 41]]
```



```
Saving 3. Predict Disease Outcome Based on Genetic and Clinical Data.csv to 3. Predict Disease Outcome Based on Genetic and Clinical Data.csv
✅ Logistic Regression Training Complete!

📊 Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        71
           1       0.98      0.95      0.96        43

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

🧩 Confusion Matrix:
 [[70  1]
 [ 2 41]]
```

- **True Positives (TP)**: 41 malignant tumors correctly classified
- **True Negatives (TN)**: 70 benign tumors correctly classified
- **False Positives (FP)**: 1 benign tumor misclassified as malignant
- **False Negatives (FN)**: 2 malignant tumors misclassified as benign
- The model achieved an overall **accuracy of 97%**, with high precision and recall for both classes.
- Only **3 misclassifications** occurred out of 114 test samples.
- The slightly higher recall for benign tumors suggests a **cautious bias**—favoring false positives over false negatives, which is generally **preferable in a medical context** (better to flag a benign tumor for further testing than to miss a malignant one).



```
  fig.canvas.print_figure(bytes_io, **kw)
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128200 (\N{CHART WITH UPWARDS TREND}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```