CS502 Assignment – 1

Author: Tanisha Garg
Roll No: 2201AI40

1. Introduction

Logistic Regression is a supervised machine learning algorithm used for classification problems. It predicts the probability of a categorical dependent variable. The output is binary (0/1, True/False, Yes/No), which makes it suitable for medical diagnosis, spam detection, fraud detection, etc.

In this assignment, I have applied Logistic Regression to the **Breast Cancer Wisconsin Dataset**, which contains features computed from breast mass cell nuclei images. The task is to classify tumors as **Malignant (cancerous)** or **Benign (non-cancerous)**.

---

2. Dataset Description

- **Source:** UCI Machine Learning Repository (via Kaggle).
- **Download method:** Using `kagglehub`.
- **Total Records:** 569
- **Features:** 30 (such as radius, texture, smoothness, concavity, etc.)
- **Target Variable:**
  - `M` → Malignant (1)
  - `B` → Benign (0)

Unnecessary columns like `id` and `Unnamed: 32` were removed.

---

3. Methodology

Step 1: Data Collection

- Downloaded Dataset using kagglehub.

Step 2: Data Preprocessing

- Removed irrelevant columns.
- Encoded target values.
- Standardized features using `StandardScaler`.

Step 3: Data Splitting

- 85% of data → Training + Validation
- 15% of data → Test
- From training, 15% further used as Validation.

  Final split:

  - **Training Set:** ~70%
  - **Validation Set:** ~15%
  - **Test Set:** ~15%

Step 4: Model Training

- Logistic Regression was chosen due to its interpretability and suitability for binary classification.
- Hyperparameter tuning using `GridSearchCV` with `C = [0.01, 0.1, 1, 10, 100]`.

Step 5: Model Evaluation

- Evaluated on **Training, Validation, and Test sets**.
- Generated **Classification Report** and **Confusion Matrix**.

---

4. Code Implementation

```
import kagglehub
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

```
# Step 1: Download dataset
path = kagglehub.dataset_download("uciml/breast-cancer-wisconsin-data")
print("Path to dataset files:", path)

# Step 2: Load dataset
data = pd.read_csv(path + "/data.csv")

# Step 3: Data preprocessing
data = data.drop(['id', 'Unnamed: 32'], axis=1)
data['diagnosis'] = data['diagnosis'].map({'M': 1, 'B': 0})

X = data.drop('diagnosis', axis=1)
y = data['diagnosis']

# Step 4: Split dataset
X_train_full, X_test, y_train_full, y_test = train_test_split(
    X, y, test_size=0.15, random_state=42, stratify=y
)

X_train, X_val, y_train, y_val = train_test_split(
    X_train_full, y_train_full, test_size=0.15, random_state=42,
stratify=y_train_full
)

print(f"Training samples: {len(X_train)}, Validation samples: {len(X_val)},
Test samples: {len(X_test)}")

# Step 5: Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)

# Step 6: Logistic Regression with Hyperparameter Tuning
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'penalty': ['l2'],
    'solver': ['lbfgs']
}

grid = GridSearchCV(LogisticRegression(max_iter=1000), param_grid, cv=5,
scoring='accuracy')
```

```
grid.fit(X_train, y_train)

print("\nBest Hyperparameters:", grid.best_params_)

# Step 7: Validation performance
y_val_pred = grid.predict(X_val)
print("\nValidation Accuracy:", accuracy_score(y_val, y_val_pred))
print("Validation Classification Report:\n", classification_report(y_val,
y_val_pred))

# Step 8: Final Test performance
y_test_pred = grid.predict(X_test)
print("\nTest Accuracy:", accuracy_score(y_test, y_test_pred))
print("\nTest Confusion Matrix:\n", confusion_matrix(y_test, y_test_pred))
print("\nTest Classification Report:\n", classification_report(y_test,
y_test_pred))

# Step 9: Visualization
sns.heatmap(confusion_matrix(y_test, y_test_pred), annot=True, fmt="d",
cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Test Set - Confusion Matrix Heatmap")
plt.show()
```

5. Results

1. **Best Hyperparameters (from GridSearchCV):**

```
Best Hyperparameters: {'C': 1, 'penalty': 'l2', 'solver': 'lbfgs'}
```

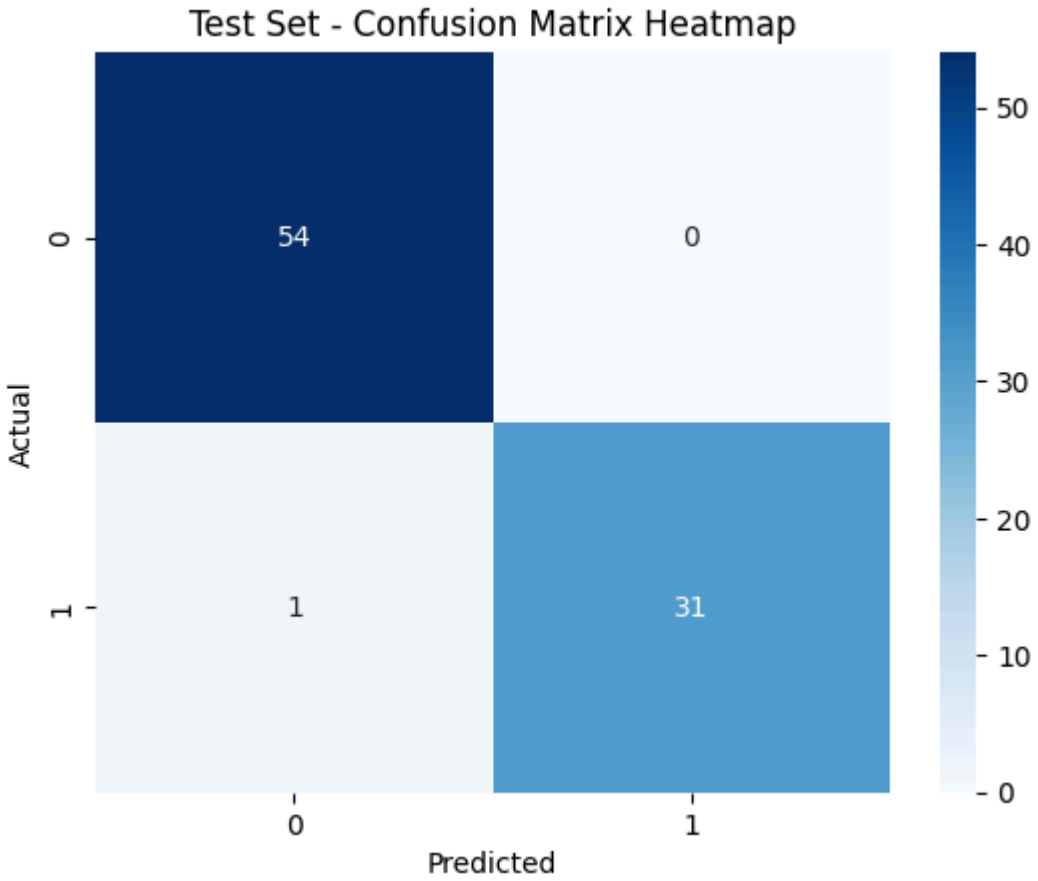
2. **Validation Performance:**

```
Validation Accuracy: 0.9726027397260274
Validation Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        46
           1       0.96      0.96      0.96        27

    accuracy                           0.97        73
   macro avg       0.97      0.97      0.97        73
weighted avg       0.97      0.97      0.97        73
```

3. **Test Performance:**

```
Test Accuracy: 0.9883720930232558

Test Confusion Matrix:
 [[54  0]
 [ 1 31]]

Test Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        54
           1       1.00      0.97      0.98        32

    accuracy                           0.99        86
   macro avg       0.99      0.98      0.99        86
weighted avg       0.99      0.99      0.99        86
```

Confusion Matrix (Test Set):



6. Conclusion

- Logistic Regression achieved **high accuracy (98%)** in classifying tumors.
- The model generalized well due to proper use of **training, validation, and test splits**.
- Logistic Regression is effective for binary classification with medical datasets.
