EXPERIMENT NUMBER : 03

Name: RITIK SHARMA

SAP ID: 590017886

Logistic Regression and Sigmoid Function for Classification-Based Problems

1. **Logistic Regression:** Logistic Regression is a supervised learning algorithm used for binary classification problems. Unlike linear regression, it uses a logistic function (sigmoid function) to map predicted values to probabilities between 0 and 1.

```
diabetes = load_diabetes()
X, y = diabetes.data, diabetes.target

y_binary = (y > np.median(y)).astype(int)
```

```
X_train, X_test, y_train, y_test = train_test_split(
  X, y_binary, test_size=0.2, random_state=42)
```

```
[4] scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
```

```
[6] model = LogisticRegression()
    model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression()
```

2. Sigmoid Function: The sigmoid function is given by:

where is the linear combination of input features and their respective weights:

Measures the proportion of actual positives correctly identified.

- **False Positive Rate (FPR):**

Measures the proportion of actual negatives incorrectly classified as positives.

- **True Negative Rate (TNR) or Specificity:**

Measures the proportion of actual negatives correctly identified.

- **False Negative Rate (FNR):**

Measures the proportion of actual positives incorrectly classified as negatives.

- **Precision (Positive Predictive Value):**

Measures the proportion of positive predictions that are actually correct.

- **F1 Score:**

The harmonic mean of precision and recall, balancing both metrics.

```
[7] y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy: {:.2f}%".format(accuracy * 100))
```

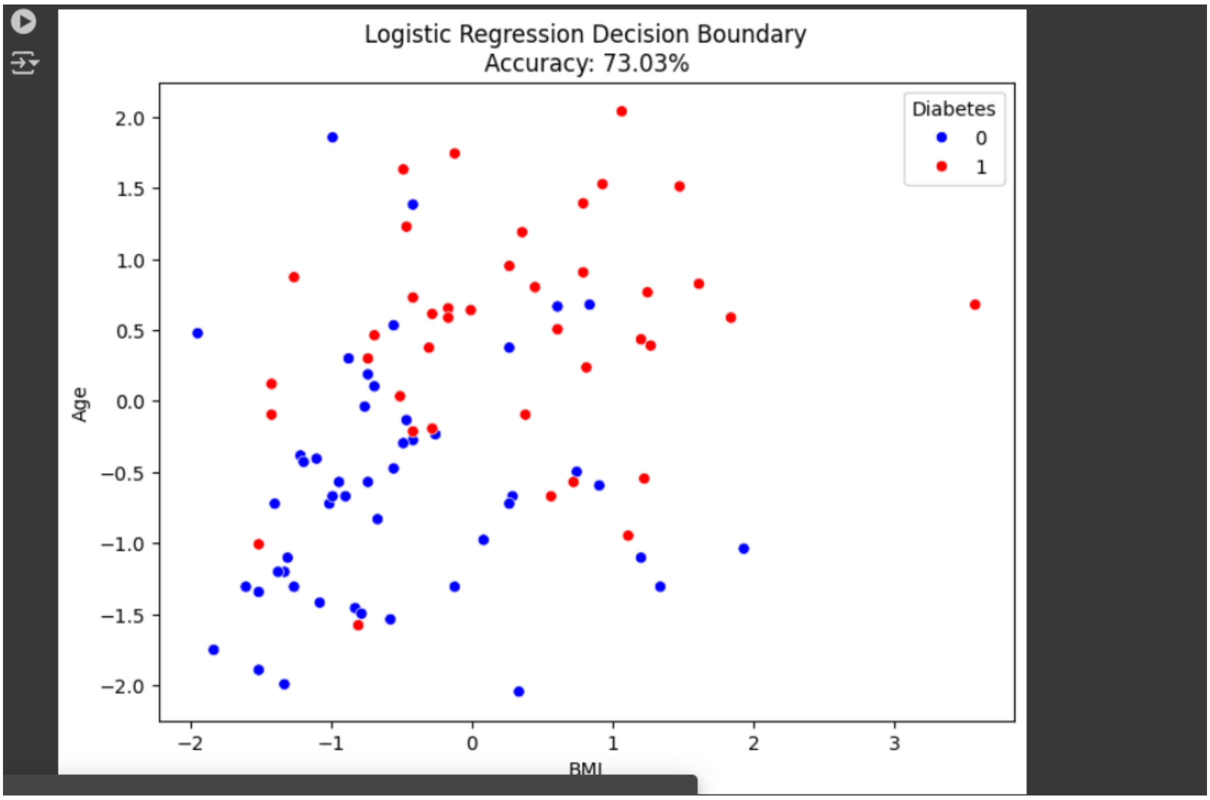
```
Accuracy: 73.03%
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Confusion Matrix:
 [[36 13]
 [11 29]]

Classification Report:
               precision    recall  f1-score   support

           0       0.77      0.73      0.75        49
           1       0.69      0.72      0.71        40

    accuracy                           0.73        89
   macro avg       0.73      0.73      0.73        89
weighted avg       0.73      0.73      0.73        89
```

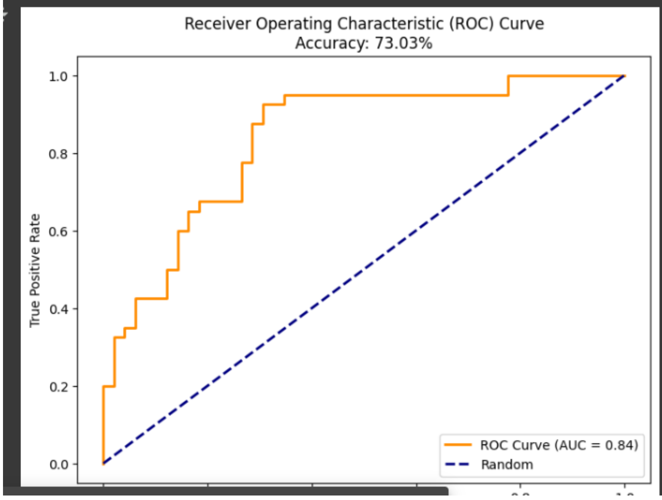
4. ROC-AUC Curve Analysis:

- The **Receiver Operating Characteristic (ROC) curve** is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold values.
- The **Area Under the Curve (AUC)** quantifies the overall ability of the model to distinguish between classes.
  - AUC = 1: Perfect classifier.
  - AUC = 0.5: Random guess.
  - AUC < 0.5: Worse than random guessing.
- A higher AUC indicates better model performance.



AUC ROC CURVE~

```
    label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve\nAccuracy: {:.2f}%'.format(
  accuracy * 100))
plt.legend(loc="lower right")
plt.show()
```



Understanding the ROC Curve:

- The **ROC curve** (orange line) plots the **True Positive Rate (TPR)** against the **False Positive Rate (FPR)** at different classification thresholds.
- The **diagonal dashed line** represents a **random classifier** (i.e., a model that randomly classifies outcomes). A perfect classifier would have a curve that reaches the top-left corner (TPR = 1, FPR = 0).

Interpreting AUC (Area Under the Curve):

- **AUC = 0.84**: This indicates that the model has a good ability to distinguish between positive and negative classes.
    - AUC = 1.0 → Perfect classifier.
    - AUC = 0.5 → Random guessing.
    - AUC < 0.5 → Worse than random.

Accuracy Annotation:

- The title mentions **Accuracy: 73.03%**, which means that the model correctly classified approximately **73% of the instances**.

- However, accuracy alone is not always the best metric, especially if there is class imbalance.

Key Takeaways:

- The model performs **significantly better than random guessing (AUC = 0.84)**.
- The ROC curve shows a strong ability to **separate the classes**, but further evaluation (e.g., precision-recall analysis) may be necessary for better assessment, especially in imbalanced datasets.