Logistic Regression - Conceptual Explanation

Logistic Regression - A Deep Dive into the Project Concepts

1. Logistic Regression:

Logistic Regression is a linear model used for binary classification problems. Unlike linear regression, which outputs a continuous value, logistic regression outputs a probability score between 0 and 1 using the sigmoid function.

The model calculates a linear combination of the input features, then applies the sigmoid function:

sigmoid(z) = 1 / (1 + exp(-z))

This squashes any real-valued number into the range (0, 1), making it interpretable as a probability. Predictions are then made by choosing a threshold (commonly 0.5).

2. Label Encoding:

Many machine learning models require the target variable to be numeric. Since the diagnosis is categorical (Malignant = 'M', Benign = 'B'), we use LabelEncoder to convert:

'M' -> 1 (malignant)

'B' -> 0 (benign)

This transformation enables numeric computation for classification algorithms.

3. fit_transform vs transform:

- fit_transform: Learns parameters from the training data and applies the transformation.

- transform: Applies the already-learned transformation on new data.

In this project, `fit_transform` is used on the training set to compute scaling parameters, and `transform` is used on the test set to ensure consistent scaling.

4. StandardScaler and Feature Standardization:

Logistic regression is sensitive to feature scales. StandardScaler ensures that each feature has:

- Mean = 0

- Standard deviation = 1

This helps the model converge faster and makes the optimization process more stable.

5. Accuracy Score:

Accuracy is the ratio of correctly predicted observations to the total observations. While it's easy to understand, it's not reliable for imbalanced datasets where one class dominates.

6. Classification Report:

Provides a detailed breakdown:

- Precision: TP / (TP + FP) - How many predicted positives are actually positive?

- Recall: TP / (TP + FN) - How many actual positives did we catch?

- F1 Score: Harmonic mean of precision and recall.

7. Confusion Matrix:

A table showing:

- True Positives (TP)

- True Negatives (TN)

- False Positives (FP)

- False Negatives (FN)

This helps in visualizing misclassifications and model performance per class.

8. ROC-AUC:

ROC (Receiver Operating Characteristic) curve plots TPR vs FPR across thresholds.

AUC (Area Under Curve) measures the model's ability to distinguish between classes. A value close

to 1.0 means excellent separability.

9. Sigmoid Function in Thresholding:

Sigmoid is used in logistic regression to compute class probabilities.

By default, predictions are made with threshold = 0.5. However, depending on business context, we can move this threshold to favor recall or precision.

10. Threshold Tuning:

- By sweeping across thresholds (e.g., from 0.0 to 1.0), we can analyze how precision, recall, and F1 score change.
- In this project, the optimal threshold (based on max F1 score) was found to be ~0.38.
- At this point, the model achieved better balance between recall and precision.

11. ROC-AUC Before vs After Threshold Tuning:

ROC-AUC remains unchanged because it's threshold-independent.

However, precision, recall, and F1 score can improve significantly at the new threshold.

Conclusion:

This project demonstrates the end-to-end process of building a binary classifier with logistic regression, handling preprocessing, standardization, threshold tuning, and evaluating using robust metrics. It is a strong foundation for more advanced classification workflows in medical data analysis.