AMIT

Machine Learning

Session 3

Agenda:

1	Logistic regression
2	Classifcation metrics

Logistic regression

Logistic regression

Logistic Regression is a supervised learning **classification algorithm** used to predict the **probability of a categorical target variable**. It deals with discrete classes like **Yes or No, 0 or 1, True or False**.

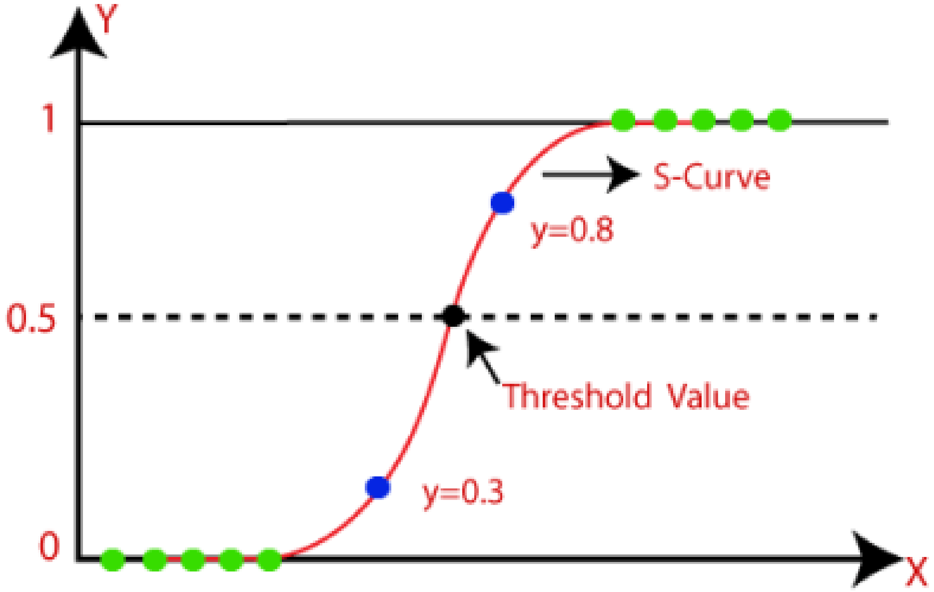
Probabilistic Output

- Unlike linear regression, logistic regression outputs a **probability value between 0 and 1**, indicating the likelihood that a given input belongs to a certain class. Based on a threshold (usually 0.5), logistic regression then classifies the input as either **class 0** or **class 1**.

Logistic regression

S-Curve

- Instead of fitting a straight line (as in linear regression), logistic regression fits an **S-shaped logistic curve**. This sigmoid curve maps the outputs to probabilities between 0 and 1.



Logistic regression

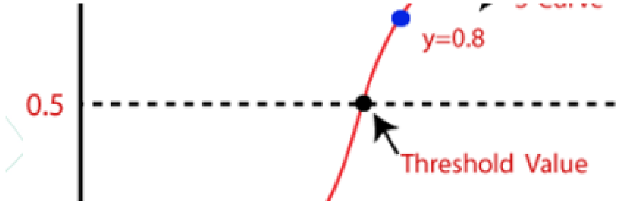
Logistic Function

1. Logistic (Sigmoid) Function

- The logistic function, also known as the **sigmoid function**, is a mathematical function that maps the predicted values to probabilities. The formula is:

$$g(z) = \frac{1}{1 + e^{-z}}$$

- This function produces values between **0 and 1**, making it ideal for classification problems. The curve's shape indicates the decision boundary for classification.



Logistic regression

Logistic Regression Equation

1. Hypothesis for Linear Regression

- The hypothesis of linear regression is given by

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n = \theta^T x$$

- Here, x represents the input features, and θ represents the coefficients or weights learned by the model.

Logistic regression

Logistic Regression Equation

2. Extension for Logistic Regression

- Logistic regression uses the **same linear hypothesis** but transforms the output using the **sigmoid function** to convert the values into probabilities.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_j x_j \quad = \theta^T x \quad = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_j \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \ldots \\ x_j \end{bmatrix}$$

$$Where\ x_0 = 1$$

Logistic regression

Logistic Regression Equation with Sigmoid Function

1. Logistic Hypothesis

- The hypothesis for logistic regression is represented as : $\hat{y} = h_\theta(x) = \sigma(\theta^T x)$

  Here, σ is the sigmoid function, which converts the linear equation's output into a probability.

Logistic regression

Logistic Regression Equation with Sigmoid Function

2. Probabilistic Interpretation

In this equation, y^ represents the predicted probability that the instance x belongs to class 1. Based on this probability, logistic regression makes a decision about the class.

Logistic regression

Sigmoid Function Explanation

1. Application of Sigmoid Function

- The **sigmoid function** is applied to the linear model's raw output to map it to a value between **0 and 1**. This makes it possible to compare the predicted output with a threshold value (e.g., 0.5) and classify the input accordingly.

Logistic regression

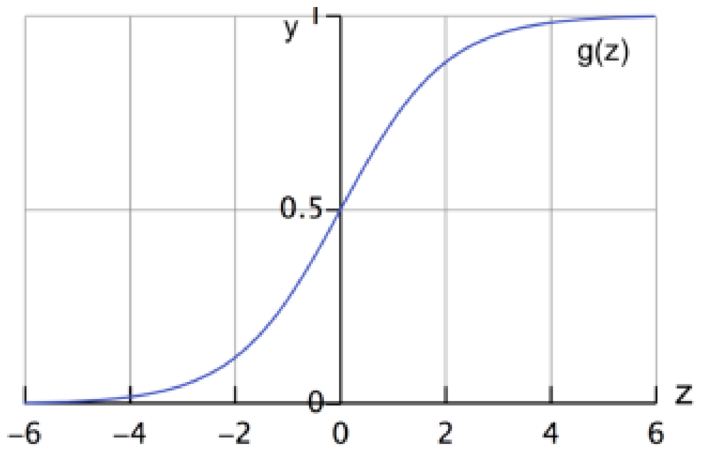
Sigmoid Function Explanation

2. Graph of Sigmoid Function

- The sigmoid function graph shows the curve with values transitioning from **0** to **1**, with the steepest part of the curve near 0.5, indicating the threshold for classification.

$$\textit{Sigmoid Function}: \quad g(z) = \frac{1}{1 + e^{(-z)}}$$

$$\textit{Hypothesis}: \quad h_\theta(x) = \frac{1}{1 + e^{(-\theta^T x)}}$$



Logistic regression

Logistic Regression Decision Boundary

1. Hypothesis for Class 1

- The hypothesis function returns the probability that y=1, given x, parameterized by θ. The formula is:

$$h(x) = P(y = 1|x; \theta)$$

2. Decision Boundary

- Logistic regression predicts class 1 if $\theta^T x \geq 0$ or equivalently if $h(x) \geq 0.5$.
- If $\theta^T x < 0$, logistic regression predicts class 0, indicating a probability less than 0.5.

Classifcation metrics

Classifcation metrics

When working with classification problems, especially binary or multiclass classification, we need to evaluate how well our model is performing. For this, we use a set of metrics called **classification metrics**.

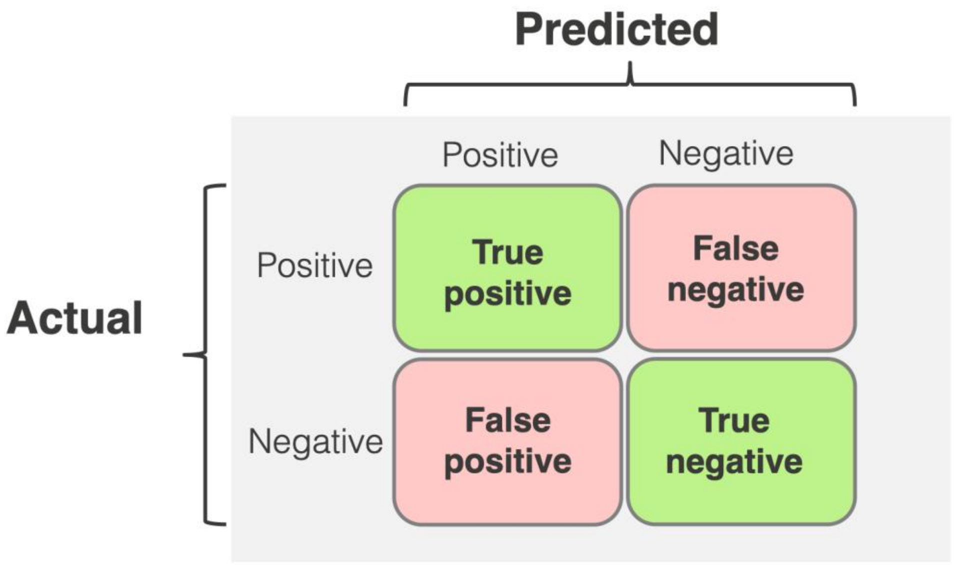
Classifcation metrics

1. Confusion Matrix

- The **confusion matrix** is the foundation of most classification metrics. It is a 2x2 table (for binary classification) that shows the **actual vs predicted** values in four categories:
  - **True Positives (TP)**: Correctly predicted positive classes.
  - **True Negatives (TN)**: Correctly predicted negative classes.
  - **False Positives (FP)**: Incorrectly predicted positive classes (also called **Type I Error**).
  - **False Negatives (FN)**: Incorrectly predicted negative classes (also called **Type II Error**).

Classifcation metrics

1. Confusion Matrix



Classifcation metrics

2. Accuracy

- Accuracy is the most intuitive and simple classification metric. It measures the percentage of correctly predicted instances out of all instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Imagine you have 100 predictions

- 80 are correct (either TP or TN)
- 20 are incorrect (either FP or FN)

Your accuracy would be $\frac{80}{100} = 80\%$

Classifcation metrics

2. Accuracy

Key Note

- Accuracy is a good metric when the data is **balanced**, meaning there are roughly equal numbers of positive and negative examples.

- However, for **imbalanced datasets**, accuracy can be misleading because the model could simply predict the majority class and achieve high accuracy.

Classifcation metrics

3. Precision

- **Precision** is the ratio of correctly predicted positive observations to the total predicted positives. It tells us how many of the predicted positives are actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

If your model predicts 30 positives, and 25 of them are actually correct (TP), while 5 are incorrect (FP), the precision would be :

$$\frac{25}{30} = 83.3\%$$

Classifcation metrics

3. Precision

Key Insight

- Precision is important in situations where **false positives** are costly, such as in email spam classification (false spam predictions are problematic).

Classifcation metrics

4. Recall (Sensitivity or True Positive Rate)

- **Recall** measures the proportion of actual positives that were correctly identified. It answers the question: **Of all the actual positives, how many did we correctly identify?**

$$\text{Recall} = \frac{TP}{TP + FN}$$

If there are 50 actual positives, and your model correctly predicted 40 of them, but missed 10, your recall would be:

$$\frac{40}{50} = 80\%$$

Classifcation metrics

4. Recall (Sensitivity or True Positive Rate)

Key Insight

- Recall is crucial in situations where **false negatives** are costly, such as disease detection. Missing a positive case (false negative) is more harmful than incorrectly predicting one.

Classifcation metrics

5. F1-Score

- The **F1-score** combines precision and recall into one metric by calculating the **harmonic mean** of both. It's particularly useful when you want to balance both precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

If your model has a precision of 80% and a recall of 70%, the F1-score would be

$$F1 = 2 \times \frac{0.8 \times 0.7}{0.8 + 0.7} = 74.6\%$$

Classifcation metrics

5. F1-Score

Key Insight

- The F1-score is a great metric when you need a balance between precision and recall. It’s especially useful in imbalanced datasets where accuracy can be misleading.

Classifcation metrics

6. ROC Curve and AUC (Area Under the Curve)

- The **Receiver Operating Characteristic (ROC) curve** is a graphical representation that shows the performance of a binary classifier as the discrimination threshold is varied.

  - True Positive Rate (TPR) = Recall = $\frac{TP}{TP+FN}$
  - False Positive Rate (FPR) = $\frac{FP}{FP+TN}$

The **ROC curve** plots the TPR against the FPR at different threshold levels. The closer the curve is to the top-left corner, the better the model.

Classifcation metrics

6. ROC Curve and AUC (Area Under the Curve)

AUC (Area Under the Curve)

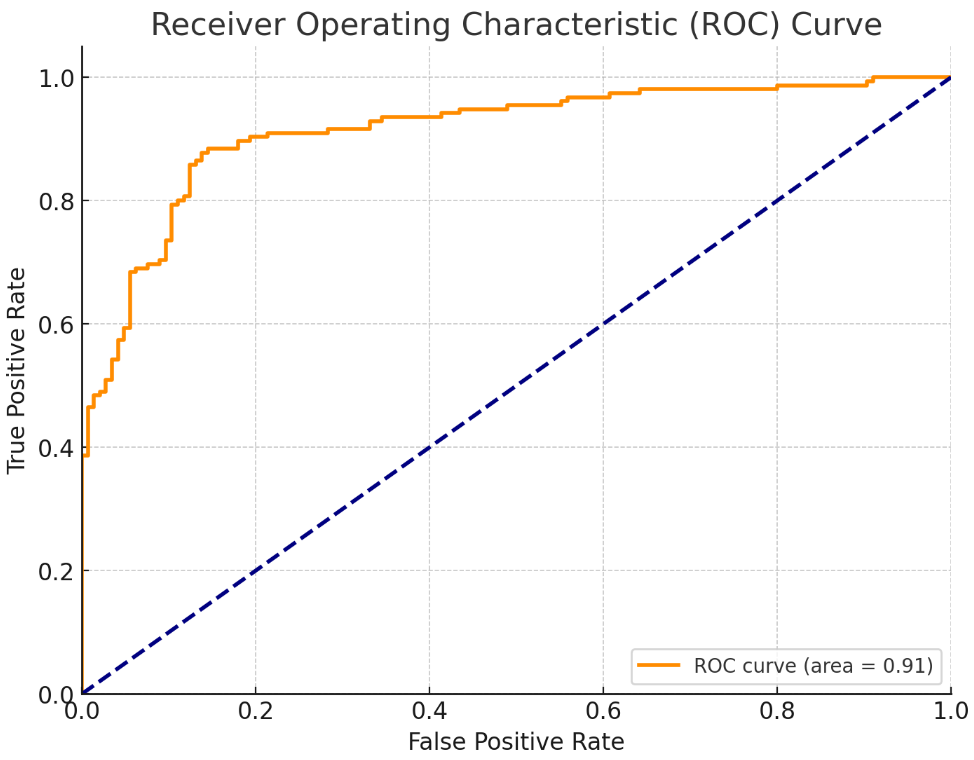
- The **AUC** value represents the degree of separability. Higher AUC means the model is better at distinguishing between positive and negative classes.

AUC ranges from 0 to 1

- 1 = Perfect classifier
- 0.5 = Random guess
- 0 = Completely wrong classification

Classifcation metrics

AUC (Area Under the Curve)



- ROC Curve is used to evaluate a classification model's performance.
- It plots True Positive Rate (TPR) against False Positive Rate (FPR) at different thresholds.
- The diagonal line represents random guessing.
- The orange line shows the model's performance with an AUC of 0.91, indicating strong performance.
- The closer the curve is to the top left corner, the better the model distinguishes between classes.

Classifcation metrics

6. ROC Curve and AUC (Area Under the Curve)

Key Insight

- **ROC-AUC** is ideal when you care about the ranking of predictions rather than the exact predicted class.

- It helps to understand how well the model distinguishes between classes across all thresholds.

Quiz

Why is Logistic Regression preferred for binary classification problems over Linear Regression?

Quiz

Expected Answer

- **Logistic Regression** is specifically designed for binary classification problems, as it outputs probabilities that range between 0 and 1 using a sigmoid function. The decision boundary (threshold) is then set (commonly at 0.5) to classify the outputs into one of two classes.

- **Linear Regression**, on the other hand, predicts continuous numeric values. When used for binary classification, it can produce values that go beyond the [0, 1] interval, which cannot be directly interpreted as probabilities. This makes it less suitable and less reliable for binary classification tasks.