Explainable Artificial Intelligence

XAI

Definition of XAI:

Explainable Artificial Intelligence (XAI) is a set of methods, techniques, and tools that enable humans to understand, interpret, and trust the decisions or predictions made by AI, machine learning, or deep learning models, by providing clear, human-understandable explanations of how and why the model reached a particular outcome.

How XAI is Different from AI, ML, DL

AI (Artificial Intelligence)

Goal: Make machines act smart (decision-making like a human).

Example: A chess-playing program that can defeat humans.

ML (Machine Learning)

Subset of AI: Machine learns from data instead of hardcoding rules.

Example: Spam filter learns patterns from emails → classifies as spam or not spam.

DL (Deep Learning)

Subset of ML: Uses neural networks with many layers to solve very complex problems.

Example: Self-driving car identifying pedestrians using a CNN.

Problem:

ML/DL models often act as black boxes.

They give predictions but don’t explain “why”.

If your loan is rejected by a bank’s AI system, you want to know:

Was it because of income? Age? Credit history?

XAI (Explainable AI)

Not about making predictions, but about explaining predictions.

Adds trust, transparency, and accountability.

Example:

Normal AI: “Loan rejected.”

XAI: “Loan rejected because your income is below $30k and your credit score is under 600. If your score was above 700, the loan would likely be approved.”

Types of Explanations in XAI (with Examples)

1. Global vs Local Explanations

Global → Explain the whole model’s logic.

Local → Explain one particular prediction.

Example: Loan Approval Model

Global: “Overall, the most important factors are income (40%), credit score (35%), and loan history (25%).”

Local: “For your case, income contributed -0.2, credit score contributed -0.5 → that’s why the decision was rejection.”

2. Model-Specific vs Model-Agnostic

Model-specific → Works only for certain models.

Example: Decision trees (you can follow the rule path).

Model-agnostic → Works for any black-box model.

Example: SHAP, LIME (can explain neural networks, XGBoost, etc.).

Example:

Model-specific: Decision Tree → “If income < 30k AND credit score < 600 → Reject loan.”

Model-agnostic: Use SHAP on XGBoost → shows feature contributions for each prediction.

3. Post-hoc Explanations (After Training)

Instead of changing the model, we explain it afterward.

Examples:

LIME: Builds a small interpretable model (like linear regression) near the prediction to explain it.

Loan rejection → LIME says: “Low credit score contributed -0.45, low income contributed -0.35.”

SHAP: Uses Shapley values from game theory → fair distribution of feature contributions.

Same example → shows exact percentage impact of features.

4. Visualization-Based Explanations

Used especially in deep learning (images, text).

CNN (image): Grad-CAM highlights which pixels influenced the classification.

Example: In a dog vs cat classifier → heatmap shows model focused on the dog’s ears.

NLP (text): Attention visualization shows which words influenced sentiment.

Example: In “This movie was amazing but a bit long” → model highlights “amazing” as positive.

AI/ML/DL = decision-making

XAI = explaining decisions in a human-friendly way

What is SHAP?

SHAP = SHapley Additive exPlanations

It is a popular XAI algorithm used to explain predictions of any ML or DL model (black-box models).

Goal: Break down a model’s prediction into contributions of each feature in a fair and mathematically sound way.

Based on: Shapley values from cooperative game theory.

Think of the prediction of a model as a “reward” in a game:

Each feature (like Age, Income, Credit Score) is a “player” in the game.

The prediction (e.g., probability of loan approval) is the “payout” or “reward”.

SHAP calculates how much each feature contributed to the final prediction, fairly distributing the reward among all features.

You trained a loan approval model using a black-box ML algorithm. For a specific applicant, the predicted probability of loan approval is 0.25. Using SHAP values, the contribution of each feature is given below:

1. Verify that the sum of SHAP values plus the base value equals the model output. Assume the base value (expected prediction) is 0.25.
2. Based on the SHAP values, determine which features pushed the prediction up and which pushed it down.
3. If the applicant’s income increases, and the SHAP value for income changes to +0.08, calculate the new predicted probability.

Feature	SHAP Value
Credit Score	-0.10
Income	-0.05
Age	+0.02
Loan History	+0.03
Employment	+0.10

1. Verify prediction using SHAP values

$$\text{Predicted probability} = \text{Base value} + \sum \text{SHAP values}$$

2. Identify which features pushed prediction up or down

Negative SHAP values → push prediction down:

Credit Score (-0.10) → decreased probability

Income (-0.05) → decreased probability

Positive SHAP values → push prediction up:

Age (+0.02) → increased probability

Loan History (+0.03) → increased probability

Employment (+0.10) → increased probability

3. New predicted probability if Income SHAP changes to +0.08

New predicted probability = 0.38