**Mini Project 3**

Author:

|  |  |
| --- | --- |
| **Abhirup Ranjan** | **110091866** |

Submitted To:

|  |
| --- |
| **Professor** |
| **Dr. Adel Abusitta** |

**Table of Contents**

[1. Problem Statement 1 2](#_Toc140837546)

[1.1 Explainable XAI 2](#_Toc140837547)

[1.2 Interpretable AI 4](#_Toc140837548)

[2. Problem Statement 2 5](#_Toc140837549)

[2.1 Explanation 5](#_Toc140837550)

[2.2 Code 9](#_Toc140837551)

# **Problem Statement 1**

Explain difference between and Explainable (XAI) and Interpretable AI.

# **Explainable XAI**

Explainable AI helps usto understand why the model made certain predictions hence it provides a details explanation of the model prediction. This helps us to understand the model's decision-making processes.

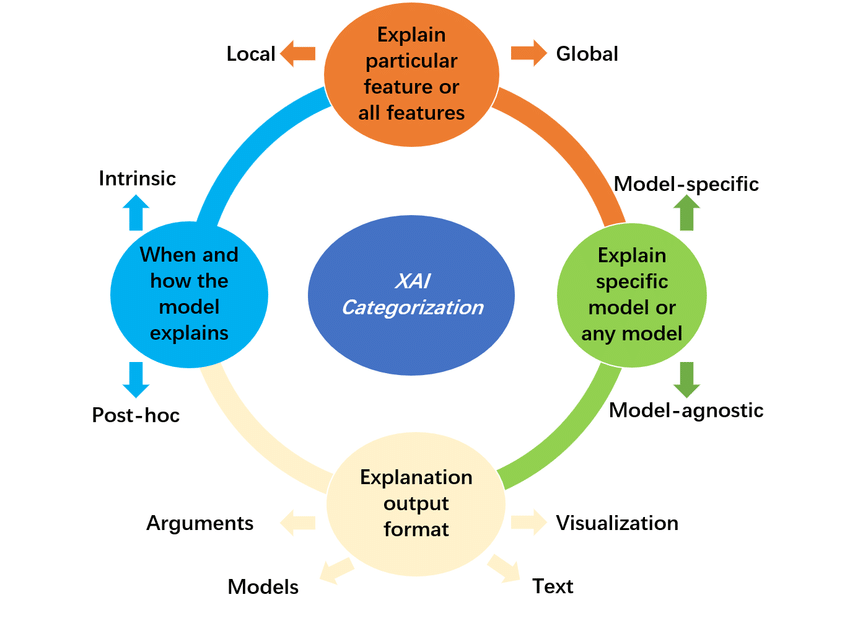
Explainable Artificial Intelligence (XAI) refers to the set of techniques and methodologies that aim to make machine learning models more transparent and interpretable to humans. In traditional machine learning models, such as deep neural networks, the decision-making process can be complex and challenging to understand. XAI aims to bridge this gap by providing insights into how these models arrive at their predictions, allowing users to comprehend and trust the system's output.

The need for XAI arises from the growing use of machine learning algorithms in critical domains such as healthcare, finance, autonomous vehicles, and legal systems. When using AI in these areas, stakeholders often require a clear explanation for the decisions made by the models to ensure fairness, accountability, and safety.

Here are some machine learning models and techniques that can be categorized under Explainable Artificial Intelligence (XAI):

* **Decision Trees:** Decision trees are inherently interpretable as they consist of a series of if-then rules. They are used for classification and regression tasks and provide a transparent way to make predictions.
* **Rule-based Models:** Rule-based models, such as rule lists or production rules, are highly interpretable. They are composed of a set of human-readable rules that guide the decision-making process.
* **Linear Regression:** Linear regression models have a simple, interpretable form, making them easy to understand. The coefficients of the linear regression equation represent the feature importance.

These models and techniques are just a few examples of the wide range of approaches used in Explainable AI. Depending on the specific problem and domain, different XAI techniques can be employed to make machine learning models more transparent and interpretable.



**Figure 1.1**

Consider a simple example of Explainable Machine Learning (XAI) using a decision tree for a loan approval system.

Assuming a bank wants to automate its loan approval process using machine learning. The bank has historical data on past loan applicants, including their credit scores, income, and employment status, along with whether their loan was approved or not (the target variable). They want to build a model to predict whether a new loan applicant should be approved or denied based on their information.

For the above case, we will use a decision tree as the machine learning model. Decision trees are interpretable because they consist of a series of if-then rules that can be easily visualized and understood. Hence the decision tree will use features such as credit score, income, and employment status to make its prediction.

Therefore, the goal of XAI is to strike a balance between model performance and interpretability. While highly complex models may achieve state-of-the-art accuracy, they often lack interpretability, which can be crucial in critical applications. XAI techniques enable us to build more transparent and accountable AI systems, which is increasingly important as AI technology becomes more prevalent in various aspects of our lives.

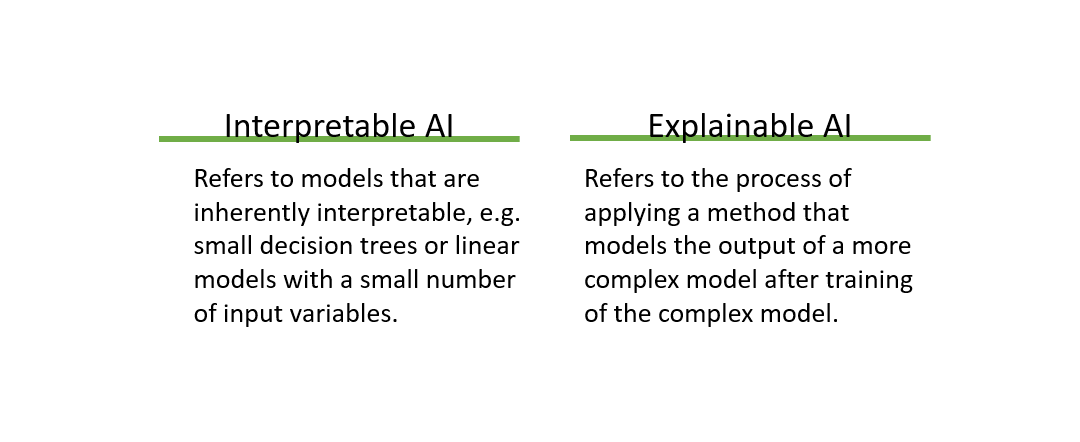
# **1.2 Interpretable AI**

Interpretable Machine Learning focuses specifically on making machine learning models more interpretable. It is a subset of XAI that deals with the interpretability of machine learning models. It focuses on improving model interpretability, making it easier for humans to understand how the model uses input features to arrive at its predictions.

Consider a simple example of Interpretable Machine Learning (ML) using a linear regression model to predict housing prices. Suppose we have a dataset containing information about houses, including their sizes (in square feet) and the corresponding prices at which they were sold. We want to build a machine learning model to predict the price of a new house based on its size.

For this case, we will use a linear regression model. With this simple linear regression model, we can now predict the price of a new house based on its size using the learned parameters.

In summary, XAI is a broader field that deals with explainability in various AI systems, while Interpretable ML specifically addresses the interpretability of machine learning models. Interpretable ML is a subset of XAI, focusing on techniques and methods that make machine learning models more transparent and easier to understand for humans. Both XAI and Interpretable ML are essential in ensuring the responsible and ethical use of AI technologies. Also, note that if the model is complex then for that case it is not interpretable.



**Figure 1.2**

# **Problem Statement 2**

Explain and Implement GAN’s. (Try to train data, what is the input and output. Can use any Machine learning model and any dataset, show the result when you train the model with training data and along with training data and augmented data).

# **2.1 Explanation**

**Implementation of a Generative Adversarial Network (GAN) for the MNIST dataset.**

Implementing a Generative Adversarial Network (GAN) for the MNIST dataset involves using deep learning techniques to generate realistic images of handwritten digits (0 to 9). GANs consist of two neural networks, a generator, and a discriminator, which are trained together in a competitive process.

A Generative Adversarial Network (GAN) is a type of deep learning model that consists of two main components: a generator and a discriminator. The generator's task is to generate realistic-looking images, while the discriminator's task is to distinguish between real images from the MNIST dataset and the fake images generated by the generator.

Here's a brief overview of the steps involved in implementing a GAN for the MNIST dataset:

1. **Data Preparation:**

* Download the MNIST dataset, which consists of 28x28 grayscale images of handwritten digits.
* Normalize the pixel values to a range between -1 and 1, which is typically done by dividing the pixel values by 127.5 and subtracting 1.

1. **Generator Model:**

* The generator takes random noise as input and generates fake images of handwritten digits.
* It typically consists of a series of fully connected or convolutional layers followed by activation functions like ReLU.
* The output layer usually employs a tanh activation function to map the pixel values between -1 and 1.

1. **Discriminator Model:**

* The discriminator is responsible for distinguishing between real images from the MNIST dataset and fake images generated by the generator.
* It also consists of a series of fully connected or convolutional layers with activation functions like LeakyReLU.
* The output layer employs a sigmoid activation function to output a probability score (0 to 1) indicating the likelihood of the input image being real.

1. **Training Loop:**

* In each training iteration, a batch of real MNIST images is selected.
* Another batch of random noise is generated as input for the generator to produce fake images.
* The discriminator is then trained with both real and fake images, aiming to correctly classify them as real or fake.
* The generator is trained to trick the discriminator by generating realistic-looking images that the discriminator misclassifies as real.

1. **Loss Functions:**

* The discriminator uses binary cross-entropy loss to distinguish between real and fake images.
* The generator uses binary cross-entropy loss to encourage the generation of fake images that the discriminator incorrectly classifies as real.

1. **Training Process:**

* During training, the generator and discriminator update their weights in an adversarial manner. The generator tries to generate better fake images to fool the discriminator, and the discriminator gets better at distinguishing between real and fake images.
* This competitive process continues until the generator produces realistic-looking images, and the discriminator becomes more accurate in distinguishing real from fake.

1. **Evaluation and Use:**

* Once training is complete, the generator can be used to generate new realistic images of handwritten digits by providing random noise as input.

Real Dataset:

A real dataset consists of real-world data collected from observations or measurements. It represents actual instances or samples that are relevant to the problem domain.

In the context of Generative Adversarial Networks (GANs), a real dataset refers to the collection of real images, texts, or other data that serve as the training set for the discriminator.

For example, in the case of image generation, a real dataset may consist of thousands of real images of handwritten digits from the MNIST dataset or real photos of cats and dogs from a dataset of animal images.

Fake Dataset:

A fake dataset, also known as a synthetic or generated dataset, is artificially created by a machine learning model, often by using a generator in GANs.

In GANs, the generator's role is to create fake data that resembles the real data as closely as possible. It generates samples from random noise with the goal of producing data that is indistinguishable from real data by the discriminator.

The fake dataset is generated by passing random noise as input to the trained generator, which produces synthetic samples that mimic the patterns and characteristics of the real dataset.

In the GAN training process, the fake dataset is used to train the discriminator to accurately differentiate between real and fake samples.

In summary, the main difference between real and fake datasets for GANs lies in their origin and content. Real datasets contain actual observations from the problem domain, while fake datasets are artificially generated by the GAN's generator to mimic the patterns present in the real data. The interplay between the real and fake datasets is what enables GANs to learn and generate realistic samples during the adversarial training process.

# **2.2 Code**

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with green and white text

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated