# Decision Tree Classifier for Biopsy Prediction

## Overview

This repository contains code for training and evaluating a Decision Tree classifier on a biomedical dataset (Dataset 1) to predict the likelihood of biopsy based on features such as 'Age' and 'Smokes (years)'. Additionally, it includes cross-validation metrics and an ROC curve analysis for a comprehensive model evaluation.

## Files

* decision\_tree\_classifier.py: Python script implementing the Decision Tree classifier and visualization for Dataset 1.
* cross\_val\_metrics.py: Python script performing 5-fold cross-validation and plotting the ROC curve for model evaluation.

## Usage

* Ensure you have the necessary Python libraries installed (pandas, numpy, scikit-learn, and matplotlib).
* Update the data loading logic in both scripts to point to your Dataset 1 (replace pd.read\_csv('path\_to\_dataset1.csv')).
* Run decision\_tree\_classifier.py to train the Decision Tree model and visualize decision boundaries.
* Run cross\_val\_metrics.py to perform cross-validation and generate evaluation metrics along with an ROC curve.

## Notes

* Handle non-numeric values and impute missing values with the mean of the column before training the model.
* The visualization script creates a decision boundaries plot showcasing how the model distinguishes between different classes based on selected features.

# Decision Tree Classifier for Fetal Health Prediction

## Overview

This repository contains code for training and evaluating a Decision Tree classifier on a fetal health dataset (Dataset 2). The implementation includes both cross-validated evaluation metrics and a visualization of decision boundaries for selected features.

## Files

* cross\_val\_metrics.py: Python script performing 5-fold cross-validation and printing cross-validated evaluation metrics.
* decision\_boundaries\_visualization.py: Python script training a Decision Tree classifier on selected features and visualizing decision boundaries.

## Usage

* Ensure you have the necessary Python libraries installed (pandas, numpy, scikit-learn, seaborn, and matplotlib).
* Update the data loading logic in both scripts to point to your Dataset 2 (replace pd.read\_csv('path\_to\_dataset2.csv')).
* Run cross\_val\_metrics.py to perform cross-validation and print evaluation metrics.
* Run decision\_boundaries\_visualization.py to train the Decision Tree model, evaluate accuracy, and visualize decision boundaries.

## Notes

* Handle non-numeric values and impute missing values with the mean of the column before training the model.
* The visualization script creates a decision boundaries plot showcasing how the model distinguishes between different classes based on selected features.

# Decision Tree Classifier with AUC-ROC Curve and Decision Boundaries Visualization

## Introduction

This repository contains code for a Decision Tree classifier implemented on Dataset 2. The classifier employs the One-vs-Rest (OvR) strategy for multiclass classification, and its performance is evaluated through an AUC-ROC curve for each class. Additionally, the decision boundaries of the model are visualized in a 2D plot.

## Code Overview

### 1. Data Preprocessing

* The dataset (dataset2) is loaded and preprocessed to handle non-numeric values and missing data.
* The target variable fetal\_health is binarized for multiclass classification.

### 2. Train-Test Split

* The data is split into training and testing sets.

### 3. Decision Tree Classification (One vs Rest)

* A Decision Tree classifier is trained using the One-vs-Rest strategy.

### 4. AUC-ROC Curve

* An AUC-ROC curve is plotted for each class, providing insights into the model's discrimination ability.

### 5. Decision Boundaries Visualization

* The decision boundaries of the trained classifier are visualized in a 2D plot.

### 6. Debugging Information

* Minimum and maximum feature values are printed for debugging purposes.
* Adjustments are made for better visualization of decision boundaries.

## Usage

* Ensure the required libraries (e.g., pandas, numpy, sklearn) are installed.
* Replace path\_to\_dataset2.csv with the actual path to your dataset.
* Run the code to train the Decision Tree classifier, plot the AUC-ROC curve, and visualize decision boundaries.

## Results

* The AUC-ROC curve provides insights into the model's discrimination performance.
* Decision boundaries visualization helps understand how the model separates different classes based on selected features.

## Notes

* This code assumes Dataset 2 is formatted appropriately for the task.

# XGBoost Classifier with AUC-ROC Curve and Decision Boundaries Visualization

## Introduction

This repository contains code for an XGBoost classifier implemented on Dataset 1. The classifier's performance is evaluated through an AUC-ROC curve for each class. Additionally, the decision boundaries of the model are visualized in a 2D plot.

## Code Overview

### 1. Data Preprocessing

* The dataset (dataset1) is loaded and preprocessed to handle non-numeric values and missing data.
* The target variable Biopsy is formatted appropriately for the task.

### 2. Train-Test Split

* The data is split into training and testing sets.

### 3. XGBoost Classification

* An XGBoost classifier is trained.

### 4. AUC-ROC Curve

* An AUC-ROC curve is plotted for each class, providing insights into the model's discrimination ability.

### 5. Decision Boundaries Visualization

* The decision boundaries of the trained classifier are visualized in a 2D plot.

### 6. Debugging Information

* Minimum and maximum feature values are printed for debugging purposes.
* Adjustments are made for better visualization of decision boundaries.

## Usage

* Ensure the required libraries (e.g., pandas, numpy, xgboost) are installed.
* Replace path\_to\_dataset1.csv with the actual path to your dataset.
* Run the code to train the XGBoost classifier, plot the AUC-ROC curve, and visualize decision boundaries.

## Results

* The AUC-ROC curve provides insights into the model's discrimination performance.
* Decision boundaries visualization helps understand how the model separates different classes based on selected features.

## Notes

* This code assumes Dataset 1 is formatted appropriately for the task.

## Adaboost Classifier

# Adaboost Classifier with AUC-ROC Curve and Decision Boundaries Visualization

## Introduction

This repository contains code for an Adaboost classifier implemented on Dataset 1. The classifier's performance is evaluated through an AUC-ROC curve for each class. Additionally, the decision boundaries of the model are visualized in a 2D plot.

## Code Overview

### 1. Data Preprocessing

* The dataset (dataset1) is loaded and preprocessed to handle non-numeric values and missing data.
* The target variable Biopsy is formatted appropriately for the task.

### 2. Train-Test Split

* The data is split into training and testing sets.

### 3. Adaboost Classification

* An Adaboost classifier is trained.

### 4. AUC-ROC Curve

* An AUC-ROC curve is plotted for each class, providing insights into the model's discrimination ability.

### 5. Decision Boundaries Visualization

* The decision boundaries of the trained classifier are visualized in a 2D plot.

### 6. Debugging Information

* Minimum and maximum feature values are printed for debugging purposes.
* Adjustments are made for better visualization of decision boundaries.

## Usage

* Ensure the required libraries (e.g., pandas, numpy, sklearn) are installed.
* Replace path\_to\_dataset1.csv with the actual path to your dataset.
* Run the code to train the Adaboost classifier, plot the AUC-ROC curve, and visualize decision boundaries.

## Results

* The AUC-ROC curve provides insights into the model's discrimination performance.
* Decision boundaries visualization helps understand how the model separates different classes based on selected features.

## Notes

* This code assumes Dataset 1 is formatted appropriately for the task.