Permutation Decision Tree (PDT)

`Under the Guidance of` `Professor Harikrishnan N. B.`

This project implements a
Permutation Decision Tree (PDT) for classification tasks. It includes functions to build, visualize, and predict using decision trees based on Effort To Compress (ETC) values.

Features

- **ETC Calculation**: Computes the effort needed to compress a sequence.
- **ETC Gain**: Determines the optimal split for a feature based on ETC gain.
- **Tree Construction**: Builds a PDT recursively to classify data.
- **Prediction**: Classifies new data points based on the decision tree.
- **Visualization**: Generates and displays a visual representation of the tree.

Dependencies

- `numpy`
- `graphviz`
- `matplotlib`

Usage

1. **Build the PDT**:

```
tree = build_pdt(data, labels, max_depth=10, resolution=2)
```

2. **Predict using the PDT**:

```
prediction = predict(tree, data_point)
```

3. **Visualize the PDT**:

```
plot(tree, name="PDT_Visualization")
```

Example

To build a decision tree and make predictions:

```
data = [[2.5, 3.1], [1.2, 3.3], [4.0, 2.8]]
labels = [0, 1, 0]
tree = build_pdt(data, labels)
predictions = prediction(data, tree)
plot(tree)
```