Assignment 4

Course: EE708, Fundamentals of Data Science and Machine Intelligence

Nikhil Jain

Dept: Electrical Engineering

Roll No: 220709

1 .

Step 1: Compute Gini Index Before Splitting

The Gini index is given by:

$$G = 1 - \sum p_i^2 \tag{1}$$

where p_i is the proportion of each class.

For the original dataset:

- Positive samples = 120
- Negative samples = 80
- Total samples = 200

Class probabilities:

$$p_{\text{positive}} = \frac{120}{200} = 0.6, \quad p_{\text{negative}} = \frac{80}{200} = 0.4 \tag{2}$$

Gini index before splitting:

$$G_{\text{before}} = 1 - (0.6^2 + 0.4^2) \tag{3}$$

$$G_{\text{before}} = 1 - (0.36 + 0.16) = 1 - 0.52 = 0.48 \tag{4}$$

Step 2: Compute Weighted Gini Index After Splitting

For the **left subset**:

- Positive samples = 50
- Negative samples = 10
- Total = 60

$$p_{\text{positive}} = \frac{50}{60} \approx 0.833, \quad p_{\text{negative}} = \frac{10}{60} \approx 0.167 \tag{5}$$

$$G_{\text{left}} = 1 - (0.833^2 + 0.167^2) \tag{6}$$

$$G_{\text{left}} = 1 - (0.694 + 0.028) = 1 - 0.722 = 0.278 \tag{7}$$

For the **right subset**:

- Positive samples = 70
- Negative samples = 70
- Total = 140

$$p_{\text{positive}} = \frac{70}{140} = 0.5, \quad p_{\text{negative}} = \frac{70}{140} = 0.5 \tag{8}$$

$$G_{\text{right}} = 1 - (0.5^2 + 0.5^2) \tag{9}$$

$$G_{\text{right}} = 1 - (0.25 + 0.25) = 1 - 0.5 = 0.5 \tag{10}$$

Step 3: Compute Weighted Gini Index
The weighted Gini index after splitting is:

$$G_{\text{after}} = \frac{60}{200} G_{\text{left}} + \frac{140}{200} G_{\text{right}} \tag{11}$$

$$G_{\text{after}} = \frac{60}{200} \times 0.278 + \frac{140}{200} \times 0.5 \tag{12}$$

$$G_{\text{after}} = 0.0834 + 0.35 = 0.4334 \tag{13}$$

Step 4: Check if Purity Improves
Since the Gini index decreased:

$$G_{\text{after}} = 0.4334 < G_{\text{before}} = 0.48 \tag{14}$$

The split improves purity as it reduces impurity.

2 .

Solution:

Given the dataset with two independent variables (x_1, x_2) and one dependent variable (y), we aim to determine the best splitting point for x_1 using the sum of squared errors (SSE) and construct the first split of a regression tree.

Step 1: Compute SSE for Different Splits on x_1

For each possible split s on x_1, we divide the dataset into two groups:

- **Left group:** $x_1 \leq s$
- **Right group:** $x_1 > s$

For each group, we compute:

1. The mean of y in that group.
2. The SSE for the group:

$$SSE = \sum (y_i - \bar{y})^2 \tag{15}$$

   where $\bar{y}$ is the mean of y in that group.

Total SSE for the split is the sum of SSEs for both groups. The split with the lowest SSE is the best.

Step 2: Compute the Best Split

The possible split points for x_1 are:

$$s = \{1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5\}$$

Computing SSE for each split, we find that the best split occurs at $x_1 = 4.5$, resulting in the minimum SSE of:

$$SSE_{\text{min}} = 82.75$$

Step 3: Constructing the First Split

Using $x_1 = 4.5$ as the split:

- **Left node** ($x_1 \leq 4.5$): Data points $(1, 10), (2, 12), (3, 15), (4, 18)$

$$\bar{y}_{\text{left}} = \frac{10 + 12 + 15 + 18}{4} = 13.75$$

- **Right node** ($x_1 > 4.5$): Data points $(5, 21), (6, 25), (7, 28), (8, 30)$

$$\bar{y}_{\text{right}} = \frac{21 + 25 + 28 + 30}{4} = 26$$

Thus, the regression tree's first split is:

$$\text{If } x_1 \leq 4.5, \text{ predict } y = 13.75$$

$$\text{If } x_1 > 4.5, \text{ predict } y = 26$$

This minimizes the total sum of squared errors (SSE).

3 .

Solution:

Step 1: Compute Squared Euclidean Distance

The squared Euclidean distance between a point (x, y) and a centroid (c_x, c_y) is given by:

$$d^2 = (x - c_x)^2 + (y - c_y)^2$$

Given initial centroids:

$$C_1 = (2, 3), \quad C_2 = (5, 8), \quad C_3 = (9, 4)$$

Data points:

$$(1, 2), \quad (3, 4), \quad (6, 7), \quad (8, 3), \quad (5, 5)$$

For each point, we compute:

$$
\begin{aligned}
d^2((1,2),C_1) &= (1-2)^2 + (2-3)^2 = 1+1 = 2\\
d^2((1,2),C_2) &= (1-5)^2 + (2-8)^2 = 16+36 = 52\\
d^2((1,2),C_3) &= (1-9)^2 + (2-4)^2 = 64+4 = 68\\
d^2((3,4),C_1) &= (3-2)^2 + (4-3)^2 = 1+1 = 2\\
d^2((3,4),C_2) &= (3-5)^2 + (4-8)^2 = 4+16 = 20\\
d^2((3,4),C_3) &= (3-9)^2 + (4-4)^2 = 36+0 = 36\\
d^2((6,7),C_1) &= (6-2)^2 + (7-3)^2 = 16+16 = 32\\
d^2((6,7),C_2) &= (6-5)^2 + (7-8)^2 = 1+1 = 2\\
d^2((6,7),C_3) &= (6-9)^2 + (7-4)^2 = 9+9 = 18\\
d^2((8,3),C_1) &= (8-2)^2 + (3-3)^2 = 36+0 = 36\\
d^2((8,3),C_2) &= (8-5)^2 + (3-8)^2 = 9+25 = 34\\
d^2((8,3),C_3) &= (8-9)^2 + (3-4)^2 = 1+1 = 2\\
d^2((5,5),C_1) &= (5-2)^2 + (5-3)^2 = 9+4 = 13\\
d^2((5,5),C_2) &= (5-5)^2 + (5-8)^2 = 0+9 = 9\\
d^2((5,5),C_3) &= (5-9)^2 + (5-4)^2 = 16+1 = 17
\end{aligned}
$$

Point	$d^2(C_1)$	$d^2(C_2)$	$d^2(C_3)$	Assigned Cluster
(1,2)	2	52	68	C_1
(3,4)	2	20	36	C_1
(6,7)	32	2	18	C_2
(8,3)	36	34	2	C_3
(5,5)	13	9	17	C_2

Step 2: Compute New Centroids

$$
\begin{aligned}
C_1' &= \left(\frac{1+3}{2}, \frac{2+4}{2}\right) = (2,3)\\
C_2' &= \left(\frac{6+5}{2}, \frac{7+5}{2}\right) = (5.5,6)\\
C_3' &= (8,3)
\end{aligned}
$$

Cluster	Assigned Points	New Centroid
C_1	(1,2), (3,4)	$(2,3)$
C_2	(6,7), (5,5)	$(5.5,6)$
C_3	(8,3)	$(8,3)$

Step 3: Compute Distortion

$$D_{\text{initial}} = 2+2+2+9+2+2+9 = 28$$

New distortion:

$$D_{\text{new}} = 2+2+1.25+1.25+0 = 6.5$$

Since $D_{\text{new}} = 6.5$ is less than $D_{\text{initial}} = 28$, the distortion has decreased, indicating improved clustering.

Final Answer:

- New centroids after one iteration: $C_1' = (2, 3), \quad C_2' = (5.5, 6), \quad C_3' = (8, 3)$.
- The distortion decreases from 28 to 6.5, confirming an improved clustering arrangement.

4 .

Solution:

We need to maximize the expectation of the complete log-likelihood function with respect to Σ_k and π_k, while keeping the responsibilities $\gamma(z_{nk})$ fixed.

Expected Log-Likelihood Function

The expectation of the complete log-likelihood function is given by:

$$E_Z[\ln p(X, Z|\mu, \Sigma, \pi)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \left(\ln \pi_k + \ln \mathcal{N}(x_n|\mu_k, \Sigma_k)\right). \tag{16}$$

Here, $\gamma(z_{nk})$ represents the posterior responsibility for data point x_n belonging to cluster k.

Maximization with respect to π_k

The constraint on mixing coefficients π_k is:

$$\sum_{k=1}^{K} \pi_k = 1. \tag{17}$$

Using a Lagrange multiplier λ, we maximize:

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln \pi_k + \lambda \left(1 - \sum_{k=1}^{K} \pi_k\right). \tag{18}$$

Differentiating w.r.t. π_k:

$$\sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} - \lambda = 0. \tag{19}$$

Solving for π_k:

$$\pi_k = \frac{N_k}{N}, \quad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{20}$$

Maximization with respect to Σ_k

The normal distribution term in the log-likelihood:

$$\ln \mathcal{N}(x_n|\mu_k, \Sigma_k) = -\frac{1}{2}\left(d\ln(2\pi) + \ln|\Sigma_k| + (x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right). \tag{21}$$

Differentiating w.r.t. Σ_k:

$$\frac{\partial}{\partial \Sigma_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(-\frac{1}{2}\ln|\Sigma_k| - \frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right) = 0. \tag{22}$$

Solving for Σ_k:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T. \tag{23}$$

Final Solutions

Thus, the closed-form solutions are:

$$\pi_k = \frac{N_k}{N}, \quad \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T. \tag{24}$$

5 .

Solution:

Consider the given mixture model:

$$p(x) = \sum_{k=1}^{K} \pi_k p(x \mid k) \tag{25}$$

where:

- x is the full data vector.
- π_k are the mixing coefficients with $\sum_{k=1}^{K} \pi_k = 1$.
- $p(x \mid k)$ are the component densities.
- The sum is taken over K mixture components.

Partitioning the Vector

We partition x into two parts: $x = (x_a, x_b)$. Then, each component density can be written as:

$$p(x \mid k) = p(x_a, x_b \mid k) = p(x_b \mid x_a, k)p(x_a \mid k). \tag{26}$$

Finding $p(x_b \mid x_a)$

Using the law of total probability:

$$p(x_b \mid x_a) = \frac{p(x_a, x_b)}{p(x_a)}. \tag{27}$$

Substituting the mixture model:

$$p(x_a, x_b) = \sum_{k=1}^{K} \pi_k p(x_a, x_b \mid k), \tag{28}$$

and marginalizing over x_b:

$$p(x_a) = \sum_{k=1}^{K} \pi_k p(x_a \mid k). \tag{29}$$

Thus, we obtain:

$$p(x_b \mid x_a) = \frac{\sum_{k=1}^{K} \pi_k p(x_a \mid k) p(x_b \mid x_a, k)}{\sum_{k=1}^{K} \pi_k p(x_a \mid k)}. \tag{30}$$

Identifying the Mixture Structure

The above equation is in the form:

$$p(x_b \mid x_a) = \sum_{k=1}^{K} \tilde{\pi}_k p(x_b \mid x_a, k), \tag{31}$$

where:

- The new mixing coefficients are given by:

$$\tilde{\pi}_k = \frac{\pi_k p(x_a \mid k)}{\sum_{j=1}^{K} \pi_j p(x_a \mid j)}. \tag{32}$$

- The new component densities remain:

$$p(x_b \mid x_a, k). \tag{33}$$

Conclusion

Thus, the conditional density $p(x_b \mid x_a)$ is itself a mixture distribution with updated mixing coefficients $\tilde{\pi}_k$ and component densities $p(x_b \mid x_a, k)$.

6 .

Solution:

(a) Log-Likelihood Function

Given a dataset $\{x_1, x_2, \ldots, x_N\}$ where each x_i is independently drawn from the Gaussian Mixture Model (GMM):

$$p(x|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{34}$$

The likelihood of the dataset is:

$$L(\Theta) = \prod_{i=1}^{N} p(x_i|\Theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \tag{35}$$

Taking the logarithm:

$$\log L(\Theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right) \tag{36}$$

This is the complete log-likelihood function.

(b) Maximum Likelihood Estimation (MLE) Update Rules

If the component $z_i \in \{1, \ldots, K\}$ generating each data point is known (i.e., we know which Gaussian component each data point belongs to), we define an indicator variable:

$$z_{ik} = \begin{cases} 1, & \text{if } x_i \text{ is generated by component } k \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

The complete-data log-likelihood (assuming known z_i) is:

$$\log L(\Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \left(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k) \right) \tag{38}$$

Maximizing with respect to the parameters:

1. Mixing Coefficients π_k

$$\pi_k = \frac{\sum_{i=1}^{N} z_{ik}}{N} \tag{39}$$

2. Mean μ_k

$$\mu_k = \frac{\sum_{i=1}^{N} z_{ik} x_i}{\sum_{i=1}^{N} z_{ik}} \tag{40}$$

3. Covariance Matrix Σ_k

$$\Sigma_k = \frac{\sum_{i=1}^{N} z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} z_{ik}} \tag{41}$$

These are the MLE update rules for a Gaussian Mixture Model when the component generating each data point is known.

7 .

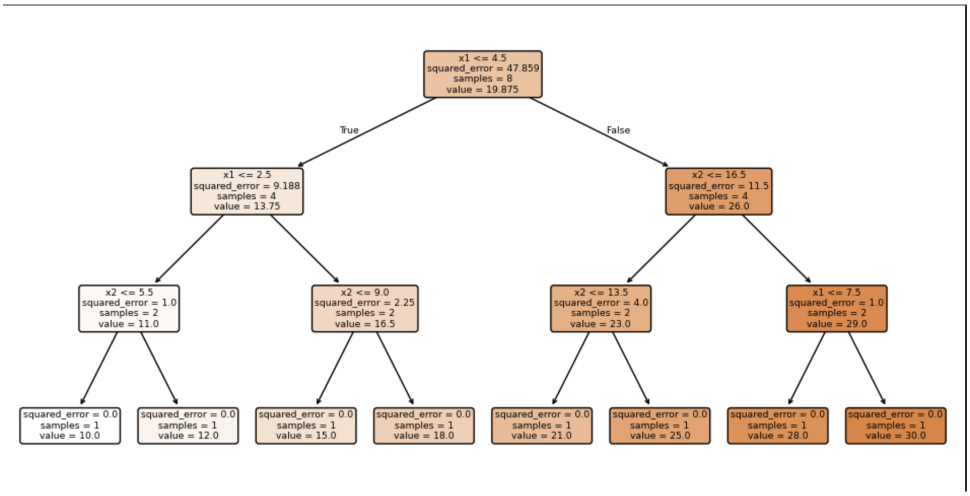
x1 <= 4.5 squared_error = 47.859 samples = 8 value = 19.875
True
False
x1 <= 2.5 squared_error = 9.188 samples = 4 value = 13.75
x2 <= 16.5 squared_error = 11.5 samples = 4 value = 26.0
x2 <= 5.5 squared_error = 1.0 samples = 2 value = 11.0
x2 <= 9.0 squared_error = 2.25 samples = 2 value = 16.5
x2 <= 13.5 squared_error = 4.0 samples = 2 value = 23.0
x1 <= 7.5 squared_error = 1.0 samples = 2 value = 29.0
squared_error = 0.0 samples = 1 value = 10.0
squared_error = 0.0 samples = 1 value = 12.0
squared_error = 0.0 samples = 1 value = 15.0
squared_error = 0.0 samples = 1 value = 18.0
squared_error = 0.0 samples = 1 value = 21.0
squared_error = 0.0 samples = 1 value = 25.0
squared_error = 0.0 samples = 1 value = 28.0
squared_error = 0.0 samples = 1 value = 30.0

Figure 1: Regression plot

8 .

a.

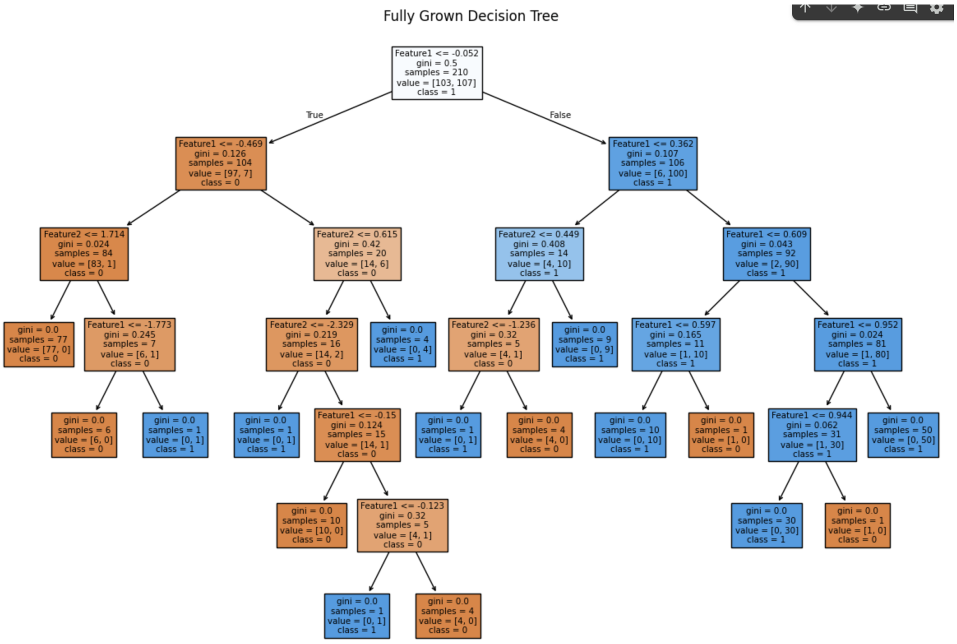
Fully Grown Decision Tree
Feature1 <= -0.052 gini = 0.5 samples = 210 value = [103, 107] class = 1
True
False
Feature1 <= -0.469 gini = 0.126 samples = 104 value = [97, 7] class = 0
Feature1 <= 0.362 gini = 0.107 samples = 106 value = [6, 100] class = 1
Feature2 <= 1.714 gini = 0.024 samples = 84 value = [83, 1] class = 0
Feature2 <= 0.615 gini = 0.42 samples = 20 value = [14, 6] class = 0
Feature2 <= 0.449 gini = 0.408 samples = 14 value = [4, 10] class = 1
Feature1 <= 0.609 gini = 0.043 samples = 92 value = [2, 90] class = 1
gini = 0.0 samples = 77 value = [77, 0] class = 0
Feature1 <= -1.773 gini = 0.245 samples = 7 value = [6, 1] class = 0
Feature2 <= -2.329 gini = 0.219 samples = 16 value = [14, 2] class = 0
gini = 0.0 samples = 4 value = [0, 4] class = 1
Feature2 <= -1.236 gini = 0.32 samples = 5 value = [4, 1] class = 0
gini = 0.0 samples = 9 value = [0, 9] class = 1
Feature1 <= 0.597 gini = 0.165 samples = 11 value = [1, 10] class = 1
Feature1 <= 0.952 gini = 0.024 samples = 81 value = [1, 80] class = 1
gini = 0.0 samples = 6 value = [6, 0] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 1
gini = 0.0 samples = 1 value = [0, 1] class = 1
Feature1 <= -0.15 gini = 0.124 samples = 15 value = [14, 1] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 1
gini = 0.0 samples = 4 value = [4, 0] class = 0
gini = 0.0 samples = 10 value = [0, 10] class = 1
gini = 0.0 samples = 1 value = [1, 0] class = 0
Feature1 <= 0.944 gini = 0.062 samples = 31 value = [1, 30] class = 1
gini = 0.0 samples = 50 value = [0, 50] class = 1
gini = 0.0 samples = 10 value = [10, 0] class = 0
Feature1 <= -0.123 gini = 0.32 samples = 5 value = [4, 1] class = 0
gini = 0.0 samples = 30 value = [0, 30] class = 1
gini = 0.0 samples = 1 value = [1, 0] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 1
gini = 0.0 samples = 4 value = [4, 0] class = 0

Figure 2: DT

b.

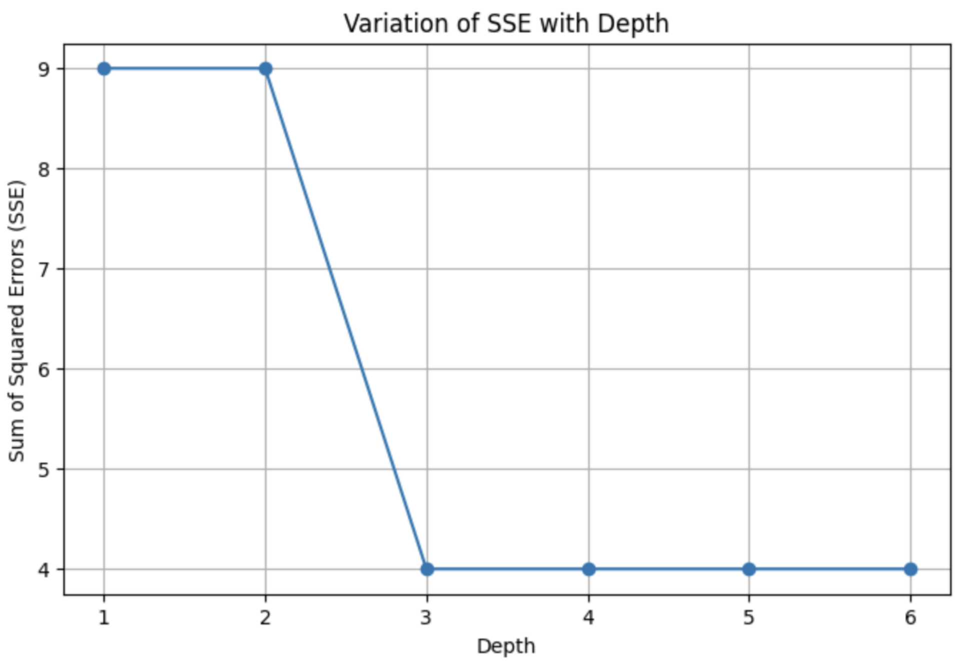

Figure 3: Variatoin plot

c. Optimal Pruning depth = 1

d.

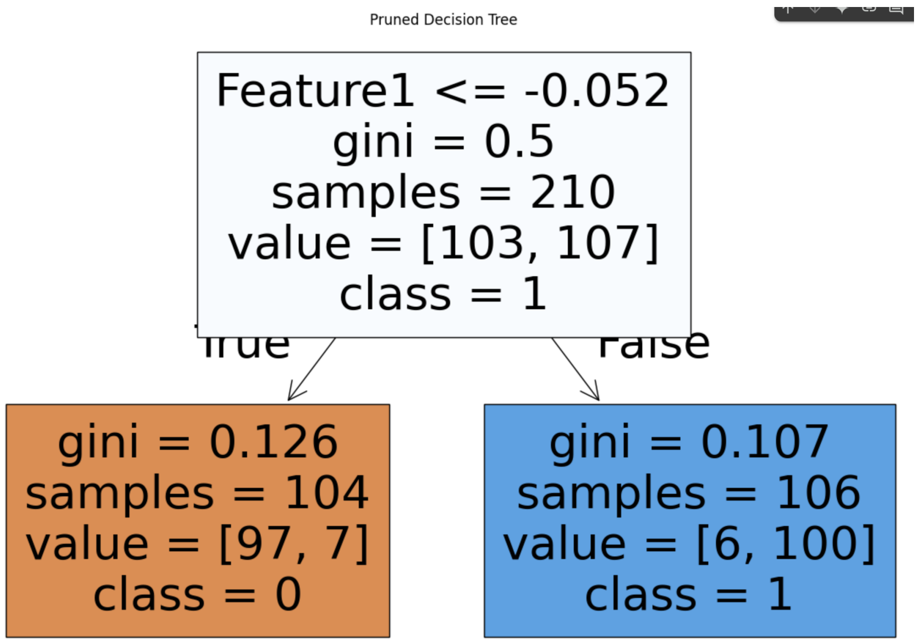

Figure 4: Pruned DT