Let's walk through how a decision tree is built using Gini impurity for this dataset to predict whether a game is played ("Play" = Yes or No). We will manually calculate the Gini impurity at each step and use it to determine the splits.

Step 1: Calculate Gini Impurity for the Root Node

Root Node (All Data)

We have 12 instances, with the target (Play) distributed as:

5 "No"

7 "Yes"

Gini impurity for the root node:

𝐺

𝑖

𝑛

𝑖

𝑟

𝑜

𝑜

𝑡

=

1

−

(

5

12

)

2

−

(

7

12

)

2

=

1

−

0.1736

−

0.3403

=

0.4861

Gini

root

​

=1−(

12

5

​

)

2

−(

12

7

​

)

2

=1−0.1736−0.3403=0.4861

Now, we need to determine the best feature to split the data. We'll calculate the Gini impurity for splits based on each feature: "Whether", "Temperature", and "Wind".

Step 2: Splitting Based on "Whether"

Split by "Whether"

Sunny (5 instances): 3 "No", 2 "Yes"

Gini for "Sunny":

𝐺

𝑖

𝑛

𝑖

𝑆

𝑢

𝑛

𝑛

𝑦

=

1

−

(

3

5

)

2

−

(

2

5

)

2

=

1

−

0.36

−

0.16

=

0.48

Gini

Sunny

​

=1−(

5

3

​

)

2

−(

5

2

​

)

2

=1−0.36−0.16=0.48

Rain (4 instances): 1 "No", 3 "Yes"

Gini for "Rain":

𝐺

𝑖

𝑛

𝑖

𝑅

𝑎

𝑖

𝑛

=

1

−

(

1

4

)

2

−

(

3

4

)

2

=

1

−

0.0625

−

0.5625

=

0.375

Gini

Rain

​

=1−(

4

1

​

)

2

−(

4

3

​

)

2

=1−0.0625−0.5625=0.375

Cloudy (3 instances): 1 "No", 2 "Yes"

Gini for "Cloudy":

𝐺

𝑖

𝑛

𝑖

𝐶

𝑙

𝑜

𝑢

𝑑

𝑦

=

1

−

(

1

3

)

2

−

(

2

3

)

2

=

1

−

0.111

−

0.444

=

0.444

Gini

Cloudy

​

=1−(

3

1

​

)

2

−(

3

2

​

)

2

=1−0.111−0.444=0.444

Now, we calculate the weighted Gini for the "Whether" split:

𝐺

𝑖

𝑛

𝑖

𝑊

ℎ

𝑒

𝑡

ℎ

𝑒

𝑟

=

5

12

×

0.48

+

4

12

×

0.375

+

3

12

×

0.444

=

0.45

Gini

Whether

​

=

12

5

​

×0.48+

12

4

​

×0.375+

12

3

​

×0.444=0.45

Step 3: Splitting Based on "Temperature"

Split by "Temperature"

High (5 instances): 3 "No", 2 "Yes"

Gini for "High":

𝐺

𝑖

𝑛

𝑖

𝐻

𝑖

𝑔

ℎ

=

1

−

(

3

5

)

2

−

(

2

5

)

2

=

0.48

Gini

High

​

=1−(

5

3

​

)

2

−(

5

2

​

)

2

=0.48

Normal (4 instances): 2 "No", 2 "Yes"

Gini for "Normal":

𝐺

𝑖

𝑛

𝑖

𝑁

𝑜

𝑟

𝑚

𝑎

𝑙

=

1

−

(

2

4

)

2

−

(

2

4

)

2

=

0.5

Gini

Normal

​

=1−(

4

2

​

)

2

−(

4

2

​

)

2

=0.5

Low (3 instances): 0 "No", 3 "Yes"

Gini for "Low":

𝐺

𝑖

𝑛

𝑖

𝐿

𝑜

𝑤

=

1

−

(

3

3

)

2

−

(

0

3

)

2

=

0

Gini

Low

​

=1−(

3

3

​

)

2

−(

3

0

​

)

2

=0

Now, the weighted Gini for "Temperature":

𝐺

𝑖

𝑛

𝑖

𝑇

𝑒

𝑚

𝑝

𝑒

𝑟

𝑎

𝑡

𝑢

𝑟

𝑒

=

5

12

×

0.48

+

4

12

×

0.5

+

3

12

×

0

=

0.42

Gini

Temperature

​

=

12

5

​

×0.48+

12

4

​

×0.5+

12

3

​

×0=0.42

Step 4: Splitting Based on "Wind"

Split by "Wind"

High (6 instances): 3 "No", 3 "Yes"

Gini for "High":

𝐺

𝑖

𝑛

𝑖

𝐻

𝑖

𝑔

ℎ

=

1

−

(

3

6

)

2

−

(

3

6

)

2

=

0.5

Gini

High

​

=1−(

6

3

​

)

2

−(

6

3

​

)

2

=0.5

Normal (6 instances): 2 "No", 4 "Yes"

Gini for "Normal":

𝐺

𝑖

𝑛

𝑖

𝑁

𝑜

𝑟

𝑚

𝑎

𝑙

=

1

−

(

2

6

)

2

−

(

4

6

)

2

=

0.444

Gini

Normal

​

=1−(

6

2

​

)

2

−(

6

4

​

)

2

=0.444

Now, the weighted Gini for "Wind":

𝐺

𝑖

𝑛

𝑖

𝑊

𝑖

𝑛

𝑑

=

6

12

×

0.5

+

6

12

×

0.444

=

0.472

Gini

Wind

​

=

12

6

​

×0.5+

12

6

​

×0.444=0.472

Step 5: Choosing the Best Split

Comparing the Gini impurities of the different splits:

Gini for "Whether" = 0.45

Gini for "Temperature" = 0.42

Gini for "Wind" = 0.472

The split based on "Temperature" gives the lowest Gini impurity (0.42), so we split the root node using "Temperature".

Step 6: Recursively Build the Tree

First Split: Temperature

Low → All "Yes" (pure node, Gini = 0)

High and Normal need further splitting.

For the "High" and "Normal" branches, we can repeat the process of calculating Gini impurity for further splits based on "Whether" and "Wind".

Summary of the Process:

Start with the entire dataset and calculate the Gini impurity of the root node.

For each feature (Whether, Temperature, Wind), calculate the weighted Gini impurity of the resulting splits.

Choose the feature that results in the lowest weighted Gini impurity to make the first split.

Recursively apply the process to each child node until all nodes are pure (or until other stopping criteria are met).

The resulting decision tree will continue branching, splitting based on the lowest Gini impurity until each leaf contains only instances of one class (pure nodes).