Building a Decision Tree from Scratch with ID3 Algorithm

August 28, 2025

1 Introduction

Decision trees are a powerful machine learning tool used for classification and regression tasks, representing decisions in a flowchart-like structure. This tutorial provides a comprehensive guide to building a decision tree using the ID3 (Iterative Dichotomiser 3) algorithm, focusing on classification. We'll use the "Play Tennis" dataset to demonstrate the process, including calculations for Entropy and Information Gain, and show how to construct and use the tree for predictions. This guide is beginner-friendly yet detailed, with a complete example dataset.

2 Why Decision Trees?

Decision trees are popular because:

- They are intuitive and easy to visualize.
- They handle both categorical and numerical data.
- No data normalization is required.
- They capture non-linear relationships.

Applications include spam detection, customer churn prediction, and medical diagnosis. We'll use the ID3 algorithm, which selects splits based on Information Gain.

3 Key Concepts

3.1 Entropy

Entropy measures the impurity or uncertainty in a dataset. For a binary classification (e.g., Yes/No), it is calculated as:

$$S = -\sum_{i=1}^{c} p_i \log_2 p_i$$

where c is the number of classes, and p_i is the proportion of examples in class i. A lower entropy indicates a purer dataset (e.g., all examples in one class: $S = 0$).

3.2 Information Gain

Information Gain (IG) measures how much entropy is reduced by splitting on an attribute:

$$IG(A) = S_{\text{parent}} - \sum_{v \in \text{values}(A)} \frac{|D_v|}{|D|} S_v$$

where A is the attribute, D_v is the subset for value v, and $|D|$ is the total dataset size. ID3 selects the attribute with the highest IG for each node.

4 Example Dataset: Play Tennis

We'll use the "Play Tennis" dataset to decide whether to play tennis based on weather conditions. It has 14 instances and 4 attributes, plus the target class (Play Tennis: Yes or No).

Table 1: Play Tennis Dataset

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Attributes:

- **Outlook**: Sunny, Overcast, Rain
- **Temperature**: Hot, Mild, Cool
- **Humidity**: High, Normal
- **Wind**: Weak, Strong
- **Target**: Play Tennis (9 Yes, 5 No)

5 Step-by-Step: Building the Decision Tree

5.1 Step 1: Entropy of the Entire Dataset

Total instances: 14 (9 Yes, 5 No).

$$p_{\text{Yes}} = \frac{9}{14}, \quad p_{\text{No}} = \frac{5}{14}$$

$$S = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) \approx 0.940$$

5.2 Step 2: Information Gain for Each Attribute

We compute IG for Outlook, Temperature, Humidity, and Wind.

5.2.1 Outlook

Values: Sunny (5: 2 Yes, 3 No), Overcast (4: 4 Yes, 0 No), Rain (5: 3 Yes, 2 No).

Entropy for Sunny:

$$S_{\text{Sunny}} = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) \approx 0.971$$

Entropy for Overcast:

$$S_{\text{Overcast}} = -\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right) = 0$$

Entropy for Rain:

$$S_{\text{Rain}} = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) \approx 0.971$$

$$IG_{\text{Outlook}} = 0.940 - \left(\frac{5}{14}\times 0.971 + \frac{4}{14}\times 0 + \frac{5}{14}\times 0.971\right) \approx 0.247$$

5.2.2 Temperature

Values: Hot (4: 2 Yes, 2 No), Mild (6: 4 Yes, 2 No), Cool (4: 3 Yes, 1 No). Entropies: Hot ≈ 1.000, Mild ≈ 0.918, Cool ≈ 0.811.

$$IG_{\text{Temp}} = 0.940 - \left(\frac{4}{14}\times 1.000 + \frac{6}{14}\times 0.918 + \frac{4}{14}\times 0.811\right) \approx 0.029$$

5.2.3 Humidity

Values: High (7: 3 Yes, 4 No), Normal (7: 6 Yes, 1 No). Entropies: High ≈ 0.985, Normal ≈ 0.592.

$$IG_{\text{Humidity}} = 0.940 - \left(\frac{7}{14}\times 0.985 + \frac{7}{14}\times 0.592\right) \approx 0.151$$

5.2.4 Wind

Values: Weak (8: 6 Yes, 2 No), Strong (6: 3 Yes, 3 No). Entropies: Weak ≈ 0.811, Strong ≈ 1.000.

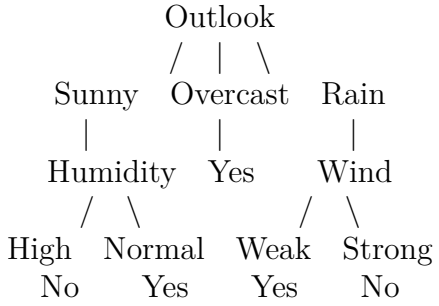
$$IG_{\text{Wind}} = 0.940 - \left(\frac{8}{14}\times 0.811 + \frac{6}{14}\times 1.000\right) \approx 0.048$$

Highest IG: Outlook (0.247) $\rightarrow$ Root node.

5.3 Step 3: Split and Recurse

- **Overcast**: All 4 Yes → Leaf node: Yes.
- **Sunny** (5 instances, entropy 0.971): Highest IG is Humidity (1.000).
  - High (3: 0 Yes, 3 No) → Leaf: No.
  - Normal (2: 2 Yes, 0 No) → Leaf: Yes.
- **Rain** (5 instances, entropy 0.971): Highest IG is Wind (0.971).
  - Weak (3: 3 Yes, 0 No) → Leaf: Yes.
  - Strong (2: 0 Yes, 2 No) → Leaf: No.

5.4 Step 4: Final Decision Tree



6 Making Predictions

For a new instance (Sunny, Mild, High, Strong):

- Outlook = Sunny → Humidity.
- Humidity = High → No (Don't play).

For Rain, Cool, Normal, Weak:

- Outlook = Rain → Wind.
- Wind = Weak → Yes (Play).

7 Python Implementation

Here's a Python script using scikit-learn to automate the process:

```
from sklearn import tree
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Load dataset
data = pd.DataFrame({
    'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain', 'Overcast',
```

```
                'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overcast', 'Rain'],
    'Temperature': ['Hot', 'Hot', 'Hot', 'Mild', 'Cool', 'Cool', 'Cool', 'Mild',
                    'Cool', 'Mild', 'Mild', 'Mild', 'Hot', 'Mild'],
    'Humidity': ['High', 'High', 'High', 'High', 'Normal', 'Normal', 'Normal',
                 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 'High'],
    'Wind': ['Weak', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Strong', 'Weak',
             'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Strong'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes',
             'Yes', 'Yes', 'Yes', 'No']
})

# Encode categorical data
le = LabelEncoder()
for col in data.columns:
    data[col] = le.fit_transform(data[col])

X = data.drop('Play', axis=1)
y = data['Play']

# Train decision tree
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf.fit(X, y)
```

8 Limitations and Tips

- **Overfitting**: Prune trees or use ensembles like Random Forests.
- **Bias**: ID3 favors attributes with many values.
- **Continuous Data**: Use C4.5 or CART for numerical data.
- **Tip**: Use a calculator for entropy calculations.

9 Conclusion

This tutorial demonstrated how to build a decision tree using the ID3 algorithm with the Play Tennis dataset. By calculating Entropy and Information Gain, we constructed a tree to predict whether to play tennis. Practice with new datasets, explore libraries like scikit-learn, or dive into Random Forests for advanced learning.

For more AI resources, visit x.ai. Happy learning!