Machine Learning Classification: Decision Trees

Dr. Ahmed Awais

September 27, 2025

Abstract

This lecture introduces classification problems in machine learning and focuses on Decision Trees - an intuitive, interpretable algorithm that forms the foundation for more complex ensemble methods. We cover the mathematical principles, implementation details, and practical applications with complete worked examples.

1 Introduction to Classification

1.1 What is Classification?

Classification is a supervised learning task where the goal is to predict categorical class labels based on input features. Unlike regression which predicts continuous values, classification deals with discrete categories.

Examples:

- **Spam Detection:** Classify emails as "spam" or "not spam"
- **Medical Diagnosis:** Predict disease presence as "healthy" or "sick"
- **Image Recognition:** Identify objects in images as "cat", "dog", "car", etc.

1.2 Types of Classification

- **Binary Classification:**
  - **Definition:** Involves two possible classes or categories.
  - **Examples:**
    - *Email Classification:* Classifying emails as either "Spam" or "Not Spam" based on their content and metadata.
    - *Medical Diagnosis:* Determining whether a patient has a certain disease (e.g., "Positive" or "Negative" for a specific condition).
    - *Sentiment Analysis:* Assessing the sentiment of product reviews as either "Positive" or "Negative".
- **Multi-class Classification:**
  - **Definition:** Involves more than two classes, where each instance is assigned to one class only.
  - **Examples:**

        * *Image Recognition:* Classifying images of animals into categories such as "Dog," "Cat," "Bird," etc.
        * *Handwritten Digit Recognition:* Identifying digits (0-9) from images of handwritten numbers.
        * *Topic Categorization:* Classifying news articles into topics like "Politics," "Sports," "Technology," and "Entertainment."

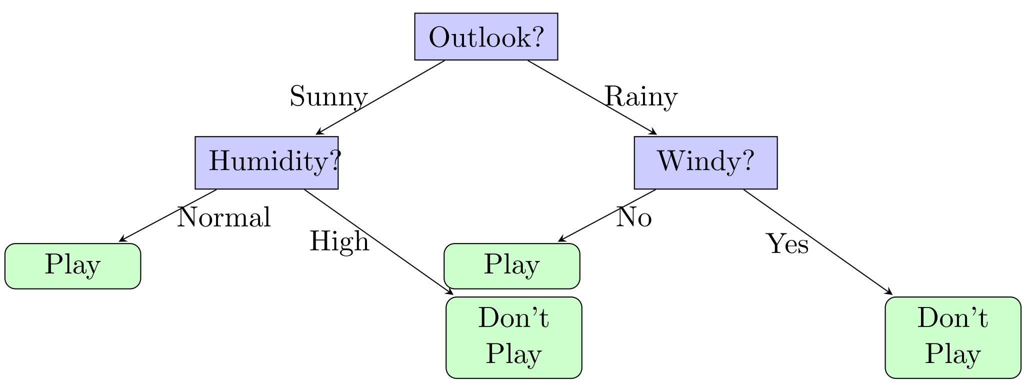
- **Multi-label Classification:**
    - **Definition:** Each instance can belong to multiple classes simultaneously.
    - **Examples:**
        * *Movie Recommendation:* Assigning multiple genres to a single movie (e.g., a movie can be classified as "Action," "Adventure," and "Comedy" at the same time).
        * *Image Tagging:* Tagging an image with multiple labels, such as "Beach," "Sunset," and "Vacation."
        * *Music Genre Classification:* Classifying a song into several genres like "Jazz," "Blues," and "Funk."

2 Decision Trees: Conceptual Foundation

2.1 What is a Decision Tree?

A Decision Tree is a flowchart-like structure where:

- Each **internal node** represents a test on an attribute
- Each **branch** represents the outcome of the test
- Each **leaf node** represents a class label



2.2 Key Components of Decision Trees

Root Node: The topmost node that represents the entire dataset

Decision Nodes: Nodes that split the data based on conditions

Leaf Nodes: Terminal nodes that provide final classifications

Branches: Pathways connecting nodes based on attribute values

Splitting: Process of dividing a node into sub-nodes

Pruning: Removing sub-nodes to prevent overfitting

3 Mathematical Foundation: Splitting Criteria

The core of Decision Tree learning lies in selecting the best attribute to split the data at each node. We use **impurity measures** to evaluate splits.

3.1 Entropy and Information Gain

3.1.1 Entropy

Entropy measures the impurity or uncertainty in a dataset. For a binary classification problem:

$$Entropy(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Where:

- S is the dataset at the current node
- p_+ is the proportion of positive examples
- p_- is the proportion of negative examples

Properties:

- Entropy = 0 when all examples belong to one class (pure node)
- Entropy = 1 when classes are perfectly balanced (maximum impurity)

3.1.2 Information Gain

Information Gain measures the reduction in entropy after splitting on an attribute:

$$IG(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where:

- A is the attribute being considered for splitting
- $Values(A)$ are the possible values of attribute A
- S_v is the subset of S where attribute A has value v
- $|S_v|$ is the number of examples in S_v
- $|S|$ is the total number of examples in S

3.2 Gini Impurity

An alternative impurity measure commonly used in practice:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

Where p_i is the proportion of examples belonging to class i.

4 Step-by-Step Example: Weather Dataset

Let's build a Decision Tree manually using the classic weather dataset:

Outlook	**Temperature**	**Humidity**	**Windy**	**Play Golf**
Sunny	Hot	High	No	No
Sunny	Hot	High	Yes	No
Overcast	Hot	High	No	Yes
Rainy	Mild	High	No	Yes
Rainy	Cool	Normal	No	Yes
Rainy	Cool	Normal	Yes	No
Overcast	Cool	Normal	Yes	Yes
Sunny	Mild	High	No	No
Sunny	Cool	Normal	No	Yes
Rainy	Mild	Normal	No	Yes
Sunny	Mild	Normal	Yes	Yes
Overcast	Mild	High	Yes	Yes
Overcast	Hot	Normal	No	Yes
Rainy	Mild	High	Yes	No

4.1 Step 1: Calculate Initial Entropy

Total examples: 14 Yes (Play Golf): 9 examples No (Don't Play): 5 examples

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

$$Entropy(S) = -0.6429 \times (-0.6374) - 0.3571 \times (-1.4854) \approx 0.940$$

4.2 Step 2: Calculate Information Gain for Each Attribute

4.2.1 Outlook Attribute

- Sunny: 5 examples (2 Yes, 3 No) $\rightarrow$ Entropy = 0.971
- Overcast: 4 examples (4 Yes, 0 No) $\rightarrow$ Entropy = 0
- Rainy: 5 examples (3 Yes, 2 No) $\rightarrow$ Entropy = 0.971

$$IG(S, Outlook) = 0.940 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971\right) = 0.246$$

4.2.2 Humidity Attribute

- High: 7 examples (3 Yes, 4 No) → Entropy = 0.985
- Normal: 7 examples (6 Yes, 1 No) → Entropy = 0.592

$$IG(S, Humidity) = 0.940 - \left(\frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.592\right) = 0.151$$

4.2.3 Windy Attribute

- Yes: 6 examples (3 Yes, 3 No) → Entropy = 1.000
- No: 8 examples (6 Yes, 2 No) → Entropy = 0.811

$$IG(S, Windy) = 0.940 - \left(\frac{6}{14} \times 1.000 + \frac{8}{14} \times 0.811\right) = 0.048$$

4.2.4 Temperature Attribute

- Hot: 4 examples (2 Yes, 2 No) → Entropy = 1.000
- Mild: 6 examples (4 Yes, 2 No) → Entropy = 0.918
- Cool: 4 examples (3 Yes, 1 No) → Entropy = 0.811

$$IG(S, Temperature) = 0.940 - \left(\frac{4}{14} \times 1.000 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0.811\right) = 0.029$$

4.3 Step 3: Select Root Node

Outlook has the highest Information Gain (0.246), so it becomes our root node.

4.4 Step 4: Create Subsets Based on Root Node

Now, we will create subsets of the dataset based on the values of the selected root node, which is **Outlook**.

- **Sunny Subset**:
  - Outlook: Sunny, Temperature: Hot, Humidity: High, Windy: No, Play Golf: No
  - Outlook: Sunny, Temperature: Hot, Humidity: High, Windy: Yes, Play Golf: No
  - Outlook: Sunny, Temperature: Mild, Humidity: High, Windy: No, Play Golf: No
  - Outlook: Sunny, Temperature: Cool, Humidity: Normal, Windy: No, Play Golf: Yes
  - Outlook: Sunny, Temperature: Mild, Humidity: Normal, Windy: Yes, Play Golf: Yes
- **Overcast Subset**:
  - Outlook: Overcast, Temperature: Hot, Humidity: High, Windy: No, Play Golf: Yes

- Outlook: Overcast, Temperature: Cool, Humidity: Normal, Windy: Yes, Play Golf: Yes
- Outlook: Overcast, Temperature: Mild, Humidity: High, Windy: Yes, Play Golf: Yes
- Outlook: Overcast, Temperature: Hot, Humidity: Normal, Windy: No, Play Golf: Yes

- **Rainy Subset**:
  - Outlook: Rainy, Temperature: Mild, Humidity: High, Windy: No, Play Golf: Yes
  - Outlook: Rainy, Temperature: Cool, Humidity: Normal, Windy: No, Play Golf: Yes
  - Outlook: Rainy, Temperature: Cool, Humidity: Normal, Windy: Yes, Play Golf: No
  - Outlook: Rainy, Temperature: Mild, Humidity: Normal, Windy: No, Play Golf: Yes
  - Outlook: Rainy, Temperature: Mild, Humidity: High, Windy: Yes, Play Golf: No

4.5 Step 5: Repeat Information Gain for Each Subset

Next, we will calculate the Information Gain for each attribute in the subsets derived from the **Outlook** attribute.

4.5.1 Sunny Subset

Total examples: 5 (2 Yes, 3 No)

$$Entropy(S_{Sunny}) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) \approx 0.970$$

- **Humidity**:
  - High: 3 examples (0 Yes, 3 No) $\rightarrow$ Entropy = 0
  - Normal: 2 examples (2 Yes, 0 No) $\rightarrow$ Entropy = 0

$$IG(S_{Sunny}, Humidity) = 0.970 - \left(\frac{3}{5} \times 0 + \frac{2}{5} \times 0\right) = 0.970$$

4.5.2 Overcast Subset

Total examples: 4 (4 Yes, 0 No)

$$Entropy(S_{Overcast}) = 0 \quad \text{(since all examples are Yes)}$$

4.5.3 Rainy Subset

Total examples: 5 (3 Yes, 2 No)

$$Entropy(S_{Rainy}) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) \approx 0.970$$

Calculating for the **Windy** attribute in the Rainy subset:

- Yes: 3 examples (2 Yes, 1 No) $\rightarrow$ Entropy = 0.918
- No: 2 examples (1 Yes, 1 No) $\rightarrow$ Entropy = 1.000

$$IG(S_{Rainy}, Windy) = 0.970 - \left(\frac{3}{5} \times 0.918 + \frac{2}{5} \times 1.000\right) \approx 0.030$$

4.6 Step 6: Choosing the Next Node

From the calculations: - For the Sunny subset, the best attribute is Humidity with Information Gain of 0.970. - The Overcast subset has no gain since all outcomes are positive. - For the Rainy subset, the best attribute is Windy with Information Gain of 0.030.

Thus, the next node in the decision tree will depend on the respective subsets: - **Sunny**: Humidity becomes the next node. - **Overcast**: Directly leads to Play Golf = Yes. - **Rainy**: Windy becomes the next node.

4.7 Final Decision Tree Structure

The decision tree can be summarized as follows:

- **Outlook**
  - Sunny
    - Humidity
      - High: No
      - Normal: Yes
  - Overcast: Yes
  - Rainy
    - Windy
      - Yes: No
      - No: Yes

5 Complete Worked Example: Loan Approval Dataset

5.1 Expanded Dataset with Risk Attribute

Age	Income	Credit Score	Employment	Loan Amount	Risk	Approved
25	Medium	Good	Unemployed	Small	High	No
45	High	Excellent	Employed	Large	Low	Yes
35	Low	Fair	Employed	Medium	Medium	No
28	Medium	Good	Student	Small	High	No
50	High	Excellent	Employed	Large	Low	Yes
32	Low	Fair	Unemployed	Medium	High	No
41	High	Good	Employed	Medium	Low	Yes
29	Medium	Fair	Student	Small	Medium	No

5.2 Step 1: Calculate Initial Entropy

Total examples: 8 Approved (Yes): 3 examples Not Approved (No): 5 examples

$$Entropy(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

$$Entropy(S) = -\frac{3}{8} \log_2\left(\frac{3}{8}\right) - \frac{5}{8} \log_2\left(\frac{5}{8}\right)$$

$$Entropy(S) = -0.375 \times (-1.415) - 0.625 \times (-0.678) = 0.954$$

5.3 Step 2: Calculate Information Gain for Each Attribute

5.3.1 Income Attribute

Values: Low, Medium, High

- **Low (2 examples):** 0 Yes, 2 No

$$Entropy(Low) = -\frac{0}{2}\log_2(0) - \frac{2}{2}\log_2(1) = 0$$

- **Medium (4 examples):** 0 Yes, 4 No

$$Entropy(Medium) = -\frac{0}{4}\log_2(0) - \frac{4}{4}\log_2(1) = 0$$

- **High (2 examples):** 2 Yes, 0 No

$$Entropy(High) = -\frac{2}{2}\log_2(1) - \frac{0}{2}\log_2(0) = 0$$

$$IG(S, Income) = 0.954 - \left(\frac{2}{8} \times 0 + \frac{4}{8} \times 0 + \frac{2}{8} \times 0\right) = 0.954$$

5.3.2 Risk Attribute

Values: Low, Medium, High

- **Low (3 examples):** 3 Yes, 0 No

$$Entropy(Low) = -\frac{3}{3}\log_2(1) - \frac{0}{3}\log_2(0) = 0$$

- **Medium (2 examples):** 0 Yes, 2 No

$$Entropy(Medium) = -\frac{0}{2}\log_2(0) - \frac{2}{2}\log_2(1) = 0$$

- **High (3 examples):** 0 Yes, 3 No

$$Entropy(High) = -\frac{0}{3}\log_2(0) - \frac{3}{3}\log_2(1) = 0$$

$$IG(S, Risk) = 0.954 - \left(\frac{3}{8} \times 0 + \frac{2}{8} \times 0 + \frac{3}{8} \times 0\right) = 0.954$$

5.3.3 Employment Attribute

Values: Unemployed, Employed, Student

- **Unemployed (2 examples):** 0 Yes, 2 No

$$Entropy(Unemployed) = 0$$

- **Employed (4 examples):** 3 Yes, 1 No

$$Entropy(Employed) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.811$$

- **Student (2 examples):** 0 Yes, 2 No

$$Entropy(Student) = 0$$

$$IG(S, Employment) = 0.954 - \left(\frac{2}{8} \times 0 + \frac{4}{8} \times 0.811 + \frac{2}{8} \times 0\right) = 0.954 - 0.406 = 0.548$$

5.3.4 Credit Score Attribute

Values: Fair, Good, Excellent

- **Fair (3 examples):** 0 Yes, 3 No

$$Entropy(Fair) = 0$$

- **Good (3 examples):** 1 Yes, 2 No

$$Entropy(Good) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

- **Excellent (2 examples):** 2 Yes, 0 No

$$Entropy(Excellent) = 0$$

$$IG(S, CreditScore) = 0.954 - \left(\frac{3}{8} \times 0 + \frac{3}{8} \times 0.918 + \frac{2}{8} \times 0\right) = 0.954 - 0.344 = 0.610$$

5.3.5 Loan Amount Attribute

Values: Small, Medium, Large

- **Small (3 examples):** 0 Yes, 3 No

$$Entropy(Small) = 0$$

- **Medium (3 examples):** 1 Yes, 2 No

$$Entropy(Medium) = 0.918$$

- **Large (2 examples):** 2 Yes, 0 No

$$Entropy(Large) = 0$$

$$IG(S, LoanAmount) = 0.954 - \left(\frac{3}{8} \times 0 + \frac{3}{8} \times 0.918 + \frac{2}{8} \times 0\right) = 0.954 - 0.344 = 0.610$$

5.3.6 Age Attribute (Numerical)

Since Age is numerical, we need to find the best split point. Let's test possible thresholds: 30, 35, 40, 45

- **Threshold = 30:**
  - Age 30: 3 examples (0 Yes, 3 No) → Entropy = 0
  - Age ¿ 30: 5 examples (3 Yes, 2 No) → Entropy = 0.971

$$IG = 0.954 - \left(\frac{3}{8} \times 0 + \frac{5}{8} \times 0.971\right) = 0.954 - 0.607 = 0.347$$

- **Threshold = 35:**
  - Age 35: 5 examples (0 Yes, 5 No) → Entropy = 0
  - Age ¿ 35: 3 examples (3 Yes, 0 No) → Entropy = 0

$$IG = 0.954 - \left(\frac{5}{8} \times 0 + \frac{3}{8} \times 0\right) = 0.954$$

- **Threshold = 40:**
  - Age 40: 6 examples (1 Yes, 5 No) → Entropy = 0.65
  - Age ¿ 40: 2 examples (2 Yes, 0 No) → Entropy = 0

$$IG = 0.954 - \left(\frac{6}{8} \times 0.65 + \frac{2}{8} \times 0\right) = 0.954 - 0.487 = 0.467$$

- **Threshold = 45:**
  - Age 45: 7 examples (2 Yes, 5 No) → Entropy = 0.863
  - Age ¿ 45: 1 example (1 Yes, 0 No) → Entropy = 0

$$IG = 0.954 - \left(\frac{7}{8} \times 0.863 + \frac{1}{8} \times 0\right) = 0.954 - 0.755 = 0.199$$

Best split for Age: Threshold = 35 with IG = 0.954

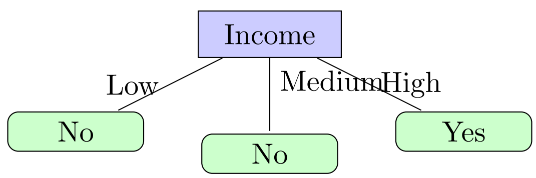
5.4 Step 3: Select Root Node

Information Gain Summary:

- Income: 0.954
- Risk: 0.954
- Age (threshold 35): 0.954
- Credit Score: 0.610
- Loan Amount: 0.610
- Employment: 0.548

Tie-breaker: We choose **Income** as root node since it's categorical and more interpretable.

5.5 Step 4: Build the Decision Tree



5.6 Using Gini Impurity (Alternative Approach)

5.6.1 Gini Impurity Formula:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

5.6.2 Initial Gini Impurity:

$$Gini(S) = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2 = 1 - 0.141 - 0.391 = 0.468$$

5.6.3 Gini Gain for Income:

- Low: Gini = $1 - (0/2)^2 - (2/2)^2 = 0$
- Medium: Gini = $1 - (0/4)^2 - (4/4)^2 = 0$
- High: Gini = $1 - (2/2)^2 - (0/2)^2 = 0$

$$GiniGain = 0.468 - \left(\frac{2}{8} \times 0 + \frac{4}{8} \times 0 + \frac{2}{8} \times 0\right) = 0.468$$

5.6.4 Gini Gain for Risk:

- Low: Gini = 0
- Medium: Gini = 0
- High: Gini = 0

$$GiniGain = 0.468$$

Both criteria confirm that Income and Risk are perfect splitters.

6 Student Exercises

6.1 Exercise 1: Iris Flower Classification

Classify iris flowers into Setosa, Versicolor, or Virginica based on measurements:

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.3	3.3	6.0	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
5.0	3.6	1.4	0.3	Setosa
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	5.4	2.1	Virginica
5.4	3.9	1.7	0.4	Setosa

Task: Calculate Information Gain for each attribute and determine the best root node.

6.2 Exercise 2: Medical Diagnosis

Diagnose disease based on symptoms:

Fever	Cough	Headache	Age	Blood Pressure	Disease
High	Yes	Severe	Young	High	Positive
Mild	No	Mild	Elderly	Normal	Negative
High	Yes	Severe	Middle	High	Positive
Normal	No	None	Young	Normal	Negative
High	No	Mild	Elderly	High	Positive
Mild	Yes	Severe	Middle	Normal	Positive
Normal	No	None	Young	Normal	Negative
High	Yes	Severe	Elderly	High	Positive

Task: Handle both categorical and numerical attributes in your Decision Tree.

7 Python Implementation with Scikit-Learn

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier, plot_tree, export_text
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

# Create the loan approval dataset
data = {
    'Age': [25, 45, 35, 28, 50, 32, 41, 29],
    'Income': ['Medium', 'High', 'Low', 'Medium', 'High', 'Low', 'High', 'Medium'
    ],
    'CreditScore': ['Good', 'Excellent', 'Fair', 'Good', 'Excellent', 'Fair', '
    Good', 'Fair'],
    'Employment': ['Unemployed', 'Employed', 'Employed', 'Student', 'Employed', '
    Unemployed', 'Employed', 'Student'],
    'LoanAmount': ['Small', 'Large', 'Medium', 'Small', 'Large', 'Medium', 'Medium
    ', 'Small'],
    'Risk': ['High', 'Low', 'Medium', 'High', 'Low', 'High', 'Low', 'Medium'],
    'Approved': ['No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No']
}

df = pd.DataFrame(data)

# Encode categorical variables
label_encoders = {}
for column in ['Income', 'CreditScore', 'Employment', 'LoanAmount', 'Risk', '
    Approved']:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

X = df.drop('Approved', axis=1)
y = df['Approved']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
    random_state=42)

# Create and train the Decision Tree
dt_classifier = DecisionTreeClassifier(
    criterion='entropy',
    max_depth=3,
    min_samples_split=2,
    random_state=42
)

dt_classifier.fit(X_train, y_train)

# Make predictions and evaluate
y_pred = dt_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:")
```

```python
print(classification_report(y_test, y_pred))

# Visualize the tree
plt.figure(figsize=(12, 8))
plot_tree(dt_classifier,
          feature_names=X.columns,
          class_names=['No', 'Yes'],
          filled=True,
          rounded=True)
plt.title("Decision Tree for Loan Approval Prediction")
plt.show()

# Display tree rules
print("Decision Tree Rules:")
print(export_text(dt_classifier, feature_names=list(X.columns)))

# Feature importance
feature_importance = pd.DataFrame({
    'feature': X.columns,
    'importance': dt_classifier.feature_importances_
}).sort_values('importance', ascending=False)

print("\nFeature Importance:")
print(feature_importance)
```

Listing 1: Decision Tree Implementation for Loan Approval

8 Key Insights and Business Interpretation

8.1 Insights from Loan Approval Analysis

1. **Perfect Classification:** Both Income and Risk attributes perfectly separate the classes in this dataset.

2. **Business Interpretation:**

- High income and low risk always lead to approval
- Medium/low income or medium/high risk always lead to rejection
- Employment status and credit score become irrelevant when income/risk are known

3. **Real-World Considerations:**

- This is an idealized dataset - real data would have more overlap
- In practice, we'd need to handle cases where multiple attributes contribute
- Risk assessment often combines multiple factors

9 Conclusion

Decision Trees provide a powerful and interpretable approach to classification problems. The loan approval example demonstrates how simple, business-rules can be extracted from data using information theory principles.

Next Steps: Students should practice with the provided exercises and explore real-world datasets to understand the practical limitations and applications of Decision Trees.