Module-04, Python for Machine Learning

Classification Algorithms (Decision Trees)

Dostdar Ali
Instructor

Data science and Artificial Intelligence
3-Months Course
at
Karakaroum international Univrsity

February 4, 2024

Table of Contents

Decision Trees

- **Definition:** Decision Trees are tree-like models that make decisions based on features. They split the dataset into subsets based on the most significant attribute.
- **History:** Originated in decision theory and statistics. Widely adopted in machine learning for its simplicity and interpretability.
- **Working Principle:** Each internal node represents a feature, each branch a decision rule, and each leaf node an outcome.
- **Examples:**
  - Customer churn prediction.
  - Credit risk assessment.

Decision Trees

- **Definition:** Decision Trees are tree-like models that make decisions based on features. They split the dataset into subsets based on the most significant attribute.
- **History:** Originated in decision theory and statistics. Widely adopted in machine learning for its simplicity and interpretability.
- **Working Principle:** Each internal node represents a feature, each branch a decision rule, and each leaf node an outcome.
- **Examples:**
  - Customer churn prediction.
  - Credit risk assessment.

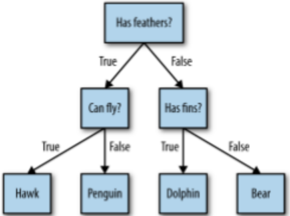
Decision Trees

- **Definition:** Decision Trees are tree-like models that make decisions based on features. They split the dataset into subsets based on the most significant attribute.
- **History:** Originated in decision theory and statistics. Widely adopted in machine learning for its simplicity and interpretability.
- **Working Principle:** Each internal node represents a feature, each branch a decision rule, and each leaf node an outcome.
- **Examples:**
  - Customer churn prediction.
  - Credit risk assessment.

Decision Trees

- **Definition:** Decision Trees are tree-like models that make decisions based on features. They split the dataset into subsets based on the most significant attribute.
- **History:** Originated in decision theory and statistics. Widely adopted in machine learning for its simplicity and interpretability.
- **Working Principle:** Each internal node represents a feature, each branch a decision rule, and each leaf node an outcome.
- **Examples:**
  - Customer churn prediction.
  - Credit risk assessment.

- Imagine you want to distinguish between the following four animals: bears, hawks, penguins, and dolphins. Your goal is to get to the right answer by asking as few if/else questions as possible. You might start off by asking whether the animal has feathers, a question that narrows down your possible animals to just two.
- If the answer is "yes," you can ask another question that could help you distinguish between hawks and penguins. For example, you could ask whether the animal can fly. If the animal doesn't have feathers, your possible animal choices are dolphins and bears, and you will need to ask a question to distinguish between these two animals—for example, asking whether the animal has fins.



Decision Trees: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Decision Rule:**

$$f(X) = \begin{cases} \text{Class 0, if } X_i \leq \text{threshold} \\ \text{Class 1, otherwise} \end{cases}$$

- **Training:** Find optimal thresholds for feature X_i maximizing information gain.

Decision Trees: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Decision Rule:**

$$f(X) = \begin{cases} \text{Class 0, if } X_i \leq \text{threshold} \\ \text{Class 1, otherwise} \end{cases}$$

- **Training:** Find optimal thresholds for feature X_i maximizing information gain.

Decision Trees: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Decision Rule:**

$$f(X) = \begin{cases} \text{Class 0, if } X_i \leq \text{threshold} \\ \text{Class 1, otherwise} \end{cases}$$

- **Training:** Find optimal thresholds for feature X_i maximizing information gain.

Decision Trees Example

- **Problem:** Classify whether a customer will purchase a product based on age and income.
- **Working Principle:** Recursive partitioning of data based on feature splits.
- **Example:**
  - Split on age ≤ 30: Predict "No Purchase"
  - Split on age > 30 and income $\leq 50,000$: Predict "Purchase"
  - Split on age > 30 and income $> 50,000$: Predict "No Purchase"
- **Prediction:** Traverse the tree based on feature values to reach a leaf node and predict the class.
- **Advantages:** Simple, interpretable model.

Decision Trees Example

- **Problem:** Classify whether a customer will purchase a product based on age and income.
- **Working Principle:** Recursive partitioning of data based on feature splits.
- **Example:**
  - Split on age ≤ 30: Predict "No Purchase"
  - Split on age > 30 and income $\leq 50,000$: Predict "Purchase"
  - Split on age > 30 and income $> 50,000$: Predict "No Purchase"
- **Prediction:** Traverse the tree based on feature values to reach a leaf node and predict the class.
- **Advantages:** Simple, interpretable model.

Decision Trees Example

- **Problem:** Classify whether a customer will purchase a product based on age and income.
- **Working Principle:** Recursive partitioning of data based on feature splits.
- **Example:**
  - Split on age ≤ 30: Predict "No Purchase"
  - Split on age > 30 and income $\leq 50,000$: Predict "Purchase"
  - Split on age > 30 and income $> 50,000$: Predict "No Purchase"
- **Prediction:** Traverse the tree based on feature values to reach a leaf node and predict the class.
- **Advantages:** Simple, interpretable model.

Decision Trees Example

- **Problem:** Classify whether a customer will purchase a product based on age and income.
- **Working Principle:** Recursive partitioning of data based on feature splits.
- **Example:**
  - Split on age ≤ 30: Predict "No Purchase"
  - Split on age > 30 and income $\leq 50,000$: Predict "Purchase"
  - Split on age > 30 and income $> 50,000$: Predict "No Purchase"
- **Prediction:** Traverse the tree based on feature values to reach a leaf node and predict the class.
- **Advantages:** Simple, interpretable model.

Decision Trees Example

- **Problem:** Classify whether a customer will purchase a product based on age and income.
- **Working Principle:** Recursive partitioning of data based on feature splits.
- **Example:**
  - Split on age ≤ 30: Predict "No Purchase"
  - Split on age > 30 and income $\leq 50,000$: Predict "Purchase"
  - Split on age > 30 and income $> 50,000$: Predict "No Purchase"
- **Prediction:** Traverse the tree based on feature values to reach a leaf node and predict the class.
- **Advantages:** Simple, interpretable model.

Decision Trees: Hand Calculations

- **Problem:** Binary classification based on two features (X_1 and X_2).
- **Data:**

Example	Feature X_1	Feature X_2
1	3	2
2	1	4
3	2	3

- **Decision Tree:**

Decision Node	Prediction
$X_1 \leq 2$	0
$X_1 > 2$ and $X_2 \leq 3$	1
$X_1 > 2$ and $X_2 > 3$	0

- **Prediction for Example 1:** $X_1 = 3, X_2 = 2 \Rightarrow$ Predict 0

- **Problem:** Binary classification based on two features (X_1 and X_2).
- **Data:**

Example	Feature X_1	Feature X_2
1	3	2
2	1	4
3	2	3

- **Decision Tree:**

Decision Node	Prediction
$X_1 \leq 2$	0
$X_1 > 2$ and $X_2 \leq 3$	1
$X_1 > 2$ and $X_2 > 3$	0

- **Prediction for Example 1:** $X_1 = 3, X_2 = 2 \Rightarrow$ Predict 0

- **Problem:** Binary classification based on two features (X_1 and X_2).
- **Data:**

Example	Feature X_1	Feature X_2
1	3	2
2	1	4
3	2	3

- **Decision Tree:**

Decision Node	Prediction
$X_1 \leq 2$	0
$X_1 > 2$ and $X_2 \leq 3$	1
$X_1 > 2$ and $X_2 > 3$	0

- **Prediction for Example 1:** $X_1 = 3, X_2 = 2 \Rightarrow$ Predict 0

Decision Trees: Hand Calculations

- **Problem:** Binary classification based on two features (X_1 and X_2).
- **Data:**

Example	Feature X_1	Feature X_2
1	3	2
2	1	4
3	2	3

- **Decision Tree:**

Decision Node	Prediction
$X_1 \leq 2$	0
$X_1 > 2$ and $X_2 \leq 3$	1
$X_1 > 2$ and $X_2 > 3$	0

- **Prediction for Example 1:** $X_1 = 3, X_2 = 2 \Rightarrow$ Predict 0

Python Code: Decision Trees

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assuming 'X' is feature matrix and 'y' is target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, te

# Creating and training the model
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Making predictions
predictions = model.predict(X_test)

# Evaluating accuracy
accuracy = accuracy_score(y_test, predictions)
```

Great Job
Thank yo