Machine Learning

| How to Split based on Continuous Feature

Prof. Dr. Mostafa Elhosseini

https://youtube.com/ElhosseiniAcademy

Agenda

- Continuous feature
- Example
- Coding using Python

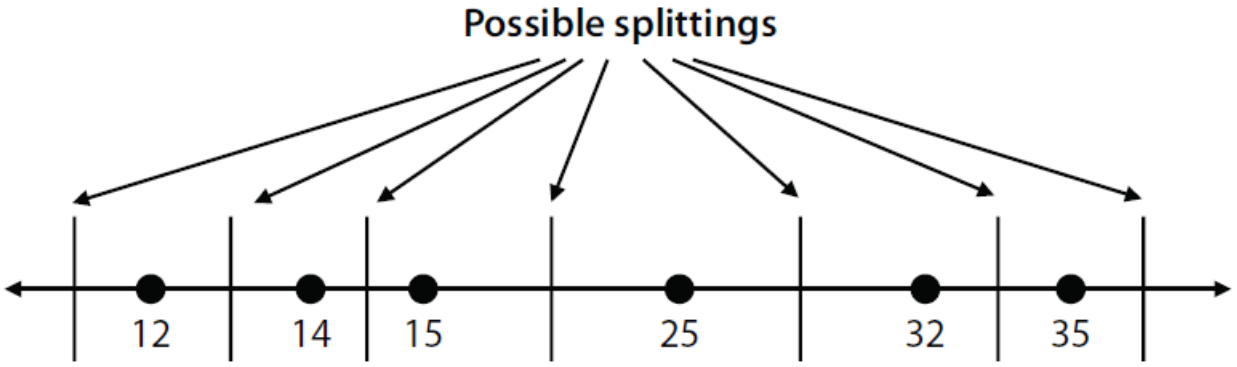
Splitting the data using continuous features

- The **age** feature contained numbers

Platform	Age	App
iPhone	15	Atom Count
iPhone	25	Check Mate Mate
Android	32	Beehive Finder
iPhone	35	Check Mate Mate
Android	12	Atom Count
Android	14	Atom Count

Splitting the data using continuous features

- only seven splits are possible



Platform	Age	App
iPhone	15	Atom Count
iPhone	25	Check Mate Mate
Android	32	Beehive Finder
iPhone	35	Check Mate Mate
Android	12	Atom Count
Android	14	Atom Count

Splitting the data using continuous features

- For clarity, let's carry out the calculations of **accuracy**, **weighted Gini** impurity index, and **weighted entropy** for the third question

Platform	Age	App
iPhone	15	Atom Count
iPhone	25	Check Mate Mate
Android	32	Beehive Finder
iPhone	35	Check Mate Mate
Android	12	Atom Count
Android	14	Atom Count

- Set 1 (younger than 14.5)
  - Ages: 12, 14
  - Labels: {A, A}
- Set 2 (14.5 and older):
  - Ages: 15, 25, 32, 25
  - Labels: {A, C, B, C}

Splitting the data using continuous features

Accuracy

- Set 1 (younger than 14.5)
  - Ages: 12, 14
  - Labels: {A, A}
  - The most common label is “A” → Every element is predicted correctly
- Set 2 (14.5 and older):
  - Ages: 15, 25, 32, 25
  - Labels: {A, C, B, C}
  - The most common label is “C” → Two elements are predicted correctly
- four out of the six data points, for an **accuracy of 4/6 = 0.667**

Splitting the data using continuous features

Gini Index

- Set 1 (younger than 14.5)
  - Labels: {A, A}
  - Set 1 is pure → Gini impurity index is 0
- Set 2 (14.5 and older):
  - Labels: {A, C, B, C}
  - $1 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.625$
- weighted average $= \frac{2}{6} \times 0 + \frac{4}{6} \times 0.625 = 0.417$

Splitting the data using continuous features

Entropy

- Set 1 (younger than 14.5)
  - Labels: {A, A}
  - Set 1 is pure → Entropy is 0
- Set 2 (14.5 and older):
  - Labels: {A, C, B, C}
  - $-\frac{1}{4}\log_2\frac{1}{4}-\frac{2}{4}\log_2\frac{2}{4}-\frac{1}{4}\log_2\frac{1}{4}=1.5$
- weighted average $=\frac{2}{6}\times 0+\frac{4}{6}\times 1.5=1$

Splitting the data using continuous features

Question	First set (yes)	Second set (no)	Labels	Weighted accuracy	Weighted Gini impurity index	Weighted entropy
Is the user younger than 7?	empty	12, 14, 15, 25, 32, 35	{}, {A,A,A,C,B,C}	3/6	0.611	1.459
Is the user younger than 13?	12	14, 15, 25, 32, 35	{A}, {A,A,C,B,C}	3/6	0.533	1.268
Is the user younger than 14.5?	12, 14	15, 25, 32, 35	{A,A} {A,C,B,C}	4/6	0.417	1.0
Is the user younger than 20?	**12, 14, 15**	**25, 32, 35**	**{A,A,A}, {C,B,C}**	**5/6**	**0.222**	**0.459**
Is the user younger than 28.5?	12, 14, 15, 25	32, 35	{A,A,A,C}, {B,C}	4/6	0.416	0.874
Is the user younger than 33.5?	12, 14, 15, 25, 32	35	{A,A,A,C,B}, {C}	4/6	0.467	1.145
Is the user younger than 100?	12, 14, 15, 25, 32, 35	empty	{A,A,A,C,B,C}, {}	3/6	0.611	1/459

Using Scikit-Learn to build a decision tree

- Dataset

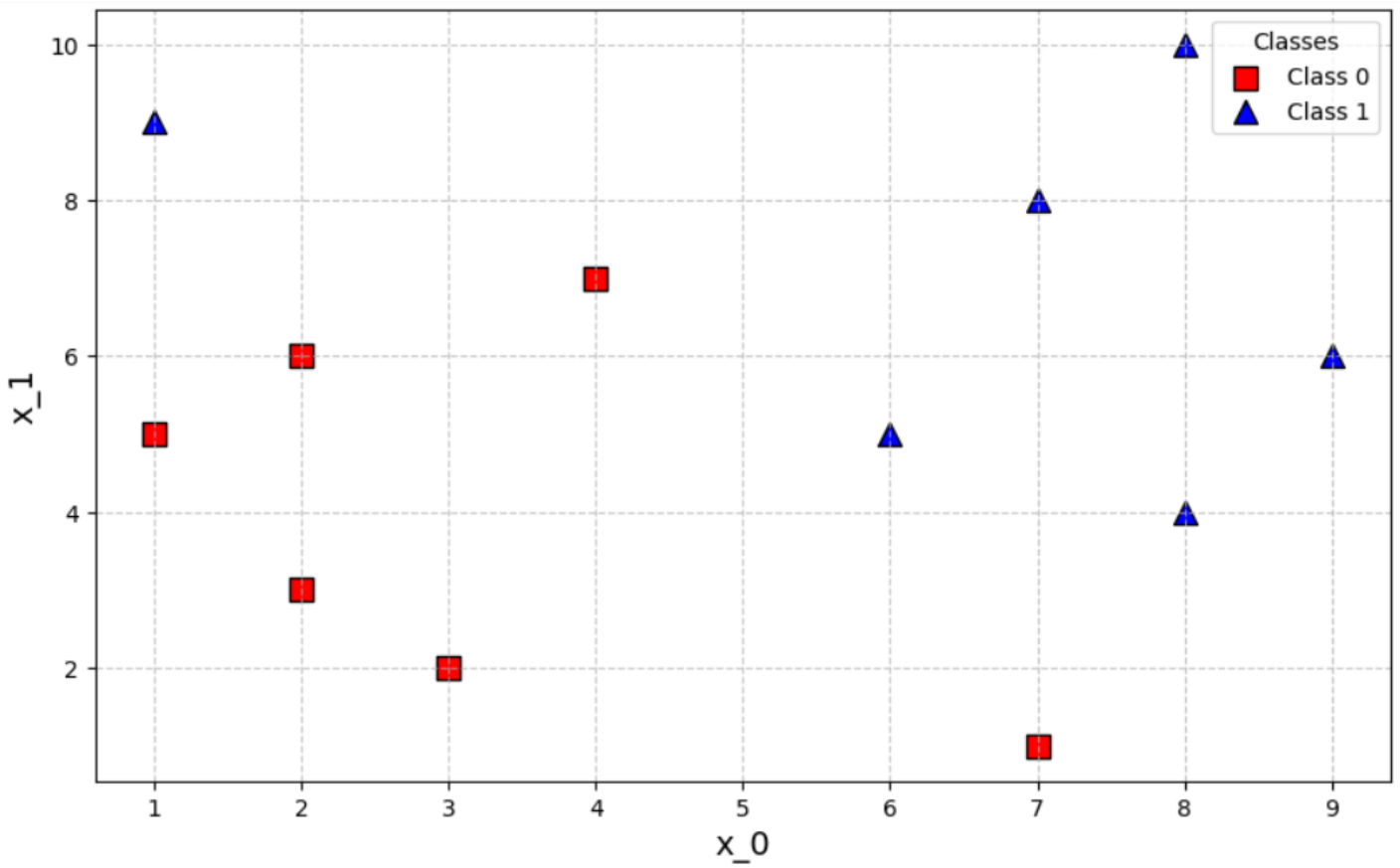
```
# Create the dataset
dataset = pd.DataFrame({
    'x_0': [7, 3, 2, 1, 2, 4, 1, 8, 6, 7, 8, 9],
    'x_1': [1, 2, 3, 5, 6, 7, 9, 10, 5, 8, 4, 6],
    'y': [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
})

# Display the dataset
print(dataset)
```

Using Scikit-Learn to build a decision tree

- Dataset

	x_0	x_1	y
0	7	1	0
1	3	2	0
2	2	3	0
3	1	5	0
4	2	6	0
5	4	7	0
6	1	9	1
7	8	10	1
8	6	5	1
9	7	8	1
10	8	4	1
11	9	6	1



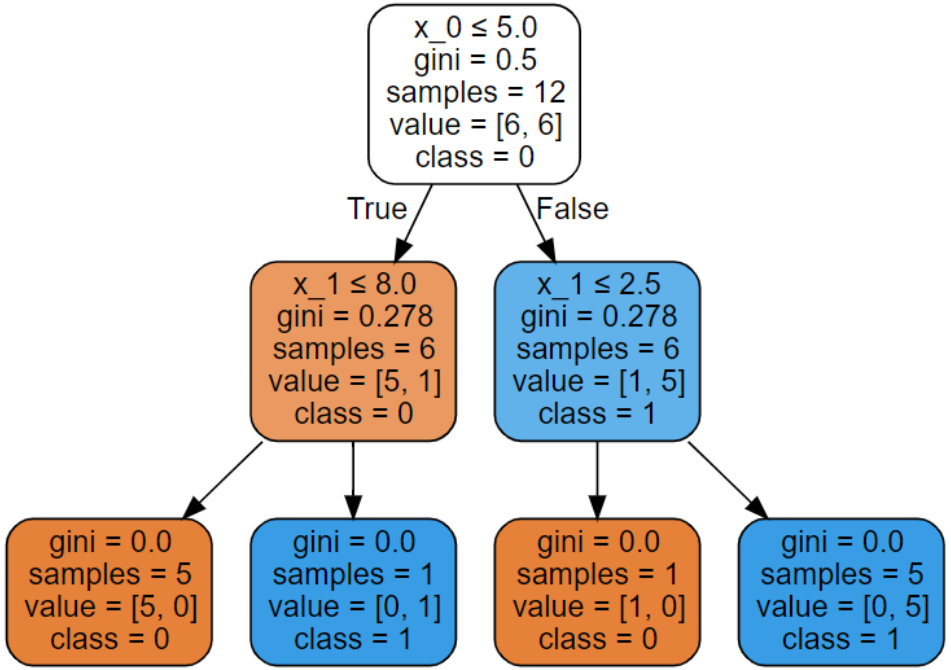
Using Scikit-Learn to build a decision tree

- Train the Decision Tree Classifier
  - **Samples**: the number of data points (samples) corresponding to that node
  - **Value**: the number of data points of each of the two labels at that node

```
# Features and target variable
X = dataset[['x_0', 'x_1']]
y = dataset['y']
```

```
# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)

# Train the model
clf.fit(X, y)
```



Using Scikit-Learn to build a decision tree

- Train the Decision Tree Classifier