# Classical Machine Learning: Classification and Regression (I)

## Learning Objectives

- Learn some techniques to understand your data and prepare your data for ML.
- Learn the basic concepts of a few interesting classifiers.

# Techniques to Understand Your Data

# Understand Your Data

- Machine learning is all about the data.
- If data quality is poor, even the most sophisticated analysis would generate only lackluster (乏善可陳) results.
- A tale (see Know_Your_Data.pdf)

# Understand Your Data with Descriptive Statistics

Data_understand.ipynb

- Take a peek at your raw data.
- Review the dimensions of your dataset.
- Review the data types of attributes in your data.
- Summarize the distribution of instances across classes in your dataset.
- Summarize your data using descriptive statistics.
- Understand the relationships in your data using correlations.
- Review the skew of the distributions of each attribute.

# Understand Your Data with Visualization


- **Histograms.**
- **Density Plots.**
- **Box and Whisker Plots.**

jupyterlab

Data_understand.ipynb

# Prepare your data for machine learning

# Data Preparation

Data_prepare.ipynb

- Rescale data.
- Standardize data.
- Normalize data.
- Binarize data.

**Scikit-Learn Recipe**

- Load the data.
- Split the dataset into the input feature matrix and output target vector for machine learning.
- Apply a pre-processing transform to the input variables.
- Summarize the data to show the change.

# Classification algorithm walkthrough

# Fun Time

**Which learning problems below is likely NOT a classification problem?**

1. **Given an image, try to predict whether it is dog or cat.**
2. **Given an applicant information, try to predict whether we should issue a credit card to her/him.**
3. **Given a rainfall, try to predict the water level of a dam.**
4. **Given a X-ray, try to predict whether it is a cancer.**

# Classification

**Classification uses models called classifiers to predict categorical (discrete, unordered) class labels.**

| Task | Feature set, **x** (or attribute set) | Class label, *y* |
|---|---|---|
| Spam filtering | Features extracted from email message header and content | spam or non-spam |
| Tumor identification | Features extracted from MRI scans | malignant or benign |
| Bridge warning | Features extracted from river velocity and depth | danger or safe |

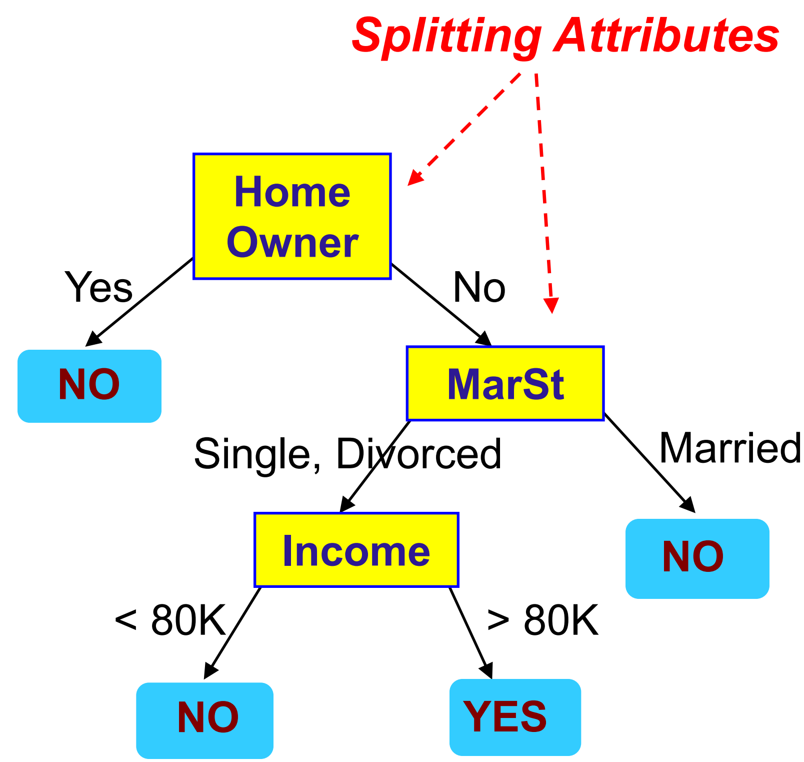
# Classification Algorithm Walkthrough: Decision Tree

# Example of a Decision Tree

categorical categorical continuous class

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

**Training Data**

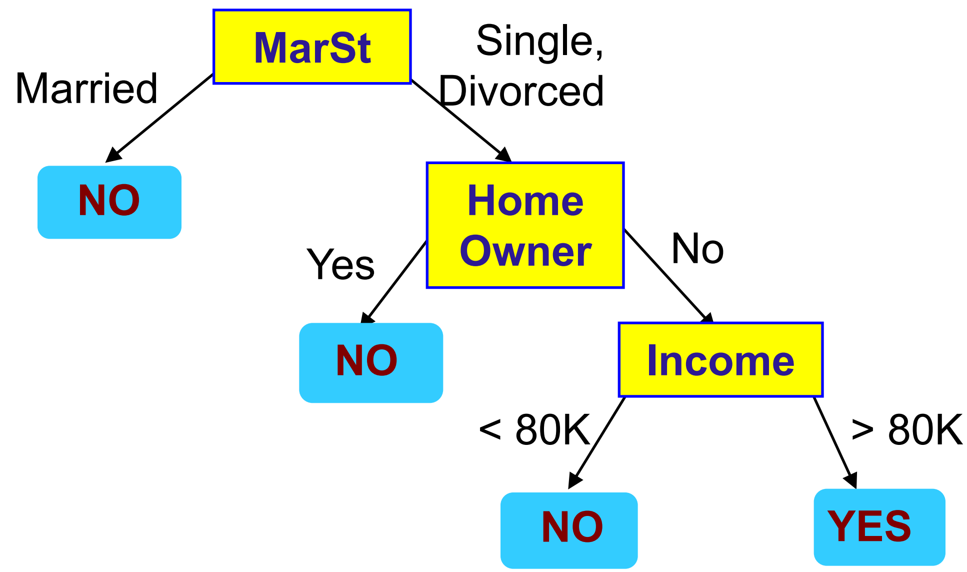



**Model: Decision Tree**

# Another Example of Decision Tree

categorical categorical continuous class

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |




There could be more than one tree that fits the same data!

# Decision Tree Induction

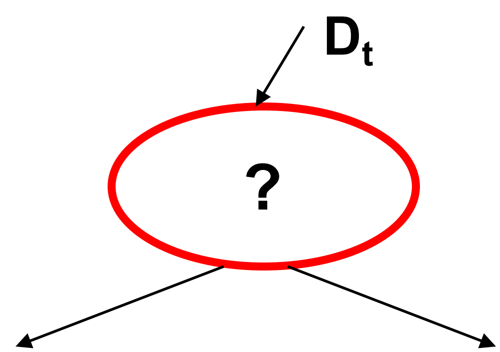
- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT

# General Structure of Hunt's Algorithm

- Let $D_t$ be the set of training records that reach a node t

- General Procedure:
  - If $D_t$ contains records that belong the same class $y_t$, then t is a leaf node labeled as $y_t$
  - If $D_t$ contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |



# Hunt’s Algorithm

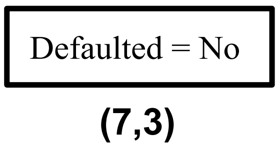
Defaulted = No

**(7,3)**

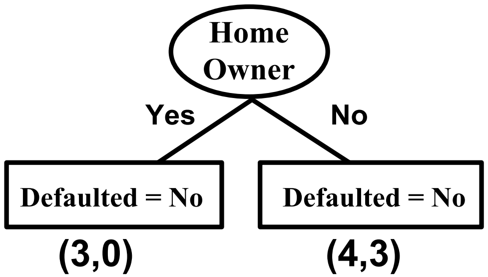
(a)

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

# Hunt's Algorithm



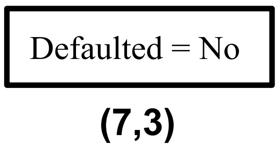

(a)



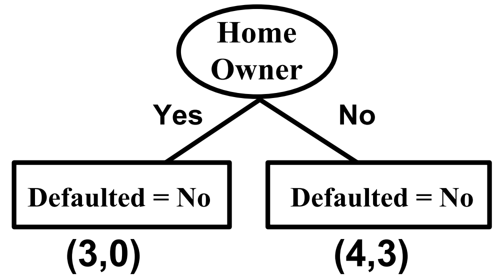

(b)

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

# Hunt's Algorithm

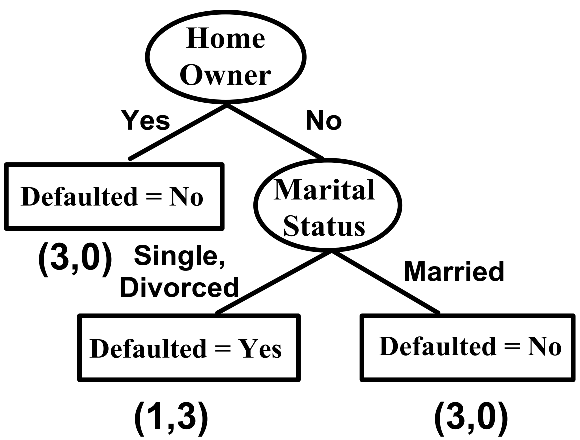



(a)



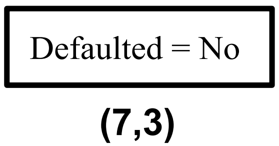

(b)

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

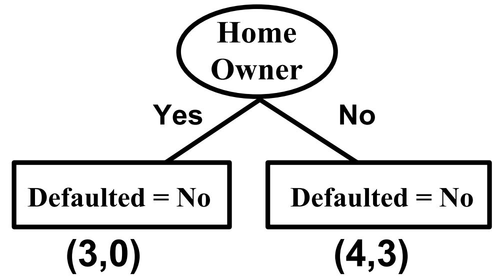



(c)

# Hunt's Algorithm

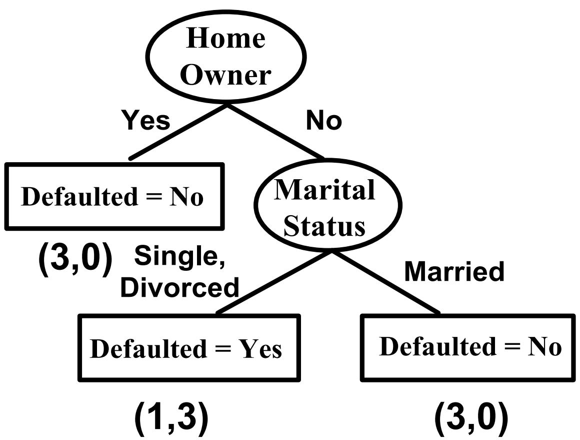



(a)

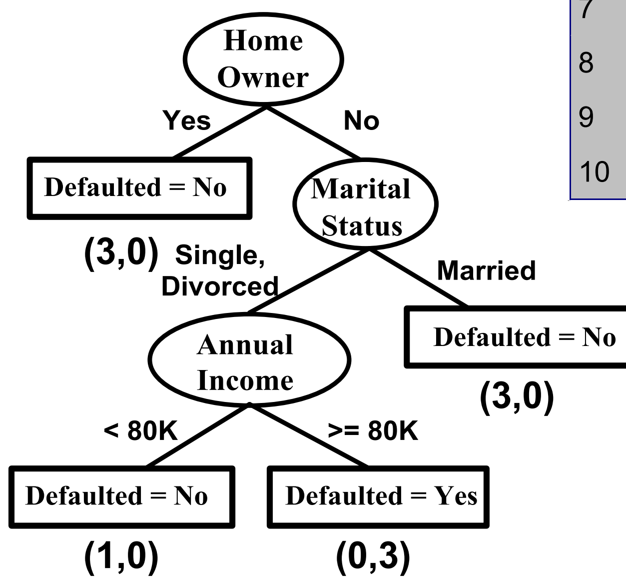



(b)

| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |




(c)




(d)

# Decision Tree: theoretical minimum and example

- The phrase "theoretical minimum" is taken from a very successful book series written by Leonard Susskind, a great physicist at Stanford University.

- "Theoretical minimum" means <u>just the minimum theories and equations you need to know in order to proceed to the next level.</u>

- See Decision_Tree.pdf

# Summary

# Classification Algorithm Walkthrough: Decision Tree

- Decision tree is simple and useful for interpretation.
- Decision tree uses a greedy algorithm with a best-split attribute to recursively split the tree.
- The “Gini” criteria, or the “Entropy” criteria is the most commonly used index to determine the best split.
- Shallow decision trees are weak learners and are not competitive in terms of prediction accuracy
- Deep decision trees tend to overfit data.
- An ensemble of randomized decision trees such as random forests is a powerful algorithm for classification. This will be covered in the sequel.