Classification:

Basic Concepts & Decision Trees

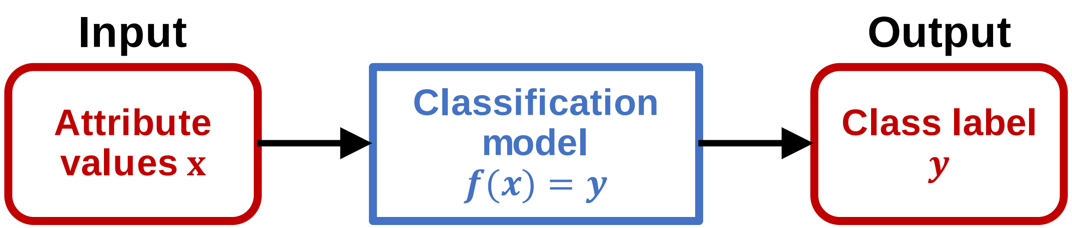
CS 418. Introduction to Data Science

Classification
Basic Concepts (I)

- **Classification** is the task of **assigning class labels** to **unlabeled** observations in a dataset.
  - Each observation in the dataset is characterized by a tuple $(\mathbf{x}, y)$, where $\mathbf{x}$ are the values for a set of **attributes** (also called **variables** or **features**) and y is a **class label**.
  - ***Example*:**
    - Classifying emails as **spam** or **non-spam**.
    - Classifying credit card transactions as **fraudulent** or **legitimate**.
    - Classifying vertebrates as **mammals**, **reptiles**, **birds**, **fishes**, and **amphibians**.
- A **classification** task involves two steps:
  - **Induction:** applying a **learning algorithm** to **labeled** observations in a **training** dataset to build a **classification model** (or **classifier**).
  - **Deduction:** applying the **classifier** to **unlabeled** observations in a **test** dataset to predict their **class labels**.

Classification
Basic Concepts (II)

- A **classifier** is used to perform a **classification** task.
  - A **classifier** is a function $\boldsymbol{f}$ that takes as input the set of attribute values $\mathbf{x}$ and produces as output the corresponding class label $\boldsymbol{y}$.



  - A **classifier** is used as a **predictive model** to classify **unlabeled** observations.
  - A **classifier** is used as a **descriptive model** to identify the characteristics that distinguish observations from different classes.
- A **classification technique** is a general approach to **classification** (e.g., **decision tree**, **logistic regression**).

Classification Evaluation (I)

- The performance of a **classifier** can be evaluated by comparing the **predicted class labels** against the **true class labels** of the observations.
- This information can be tabulated in a **confusion matrix**.

		Predicted class	
		P $(y = 1)$	***N*** $(y = 0)$
True class	***P*** $(y = 1)$	***TP***	***FN***
	N $(y = 0)$	***FP***	***TN***

where:

- ***TP*:** number of **true** positives (positive observations classified correctly).
- ***FP*:** number of **false** positives (positive observations classified incorrectly).
- ***TN*:** number of **true** negatives (negative observations classified correctly).
- ***FN*:** number of **false** negatives (negative observations classified incorrectly).

Classification Evaluation (II)

- The information in the **confusion matrix** can be summarized as **evaluation metrics**.
  - **Accuracy:** measures how many observations were classified correctly.

$$\boldsymbol{Accuracy = \frac{TP+TN}{TP+FP+TN+FN}}$$

  - **Error:** measures how many observations were classified incorrectly.

$$\boldsymbol{Error = 1 - Accuracy = \frac{FP+FN}{TP+FP+TN+FN}}$$

  - **Precision:** measures how many observations classified as positive were indeed positive.

$$\boldsymbol{Precision = \frac{TP}{TP+FP}}$$

  - **Recall:** measures how many positive observations were classified correctly.

$$\boldsymbol{Recall = \frac{TP}{TP+FN}}$$

Classification Evaluation (III)

- The information in the **confusion matrix** can be summarized as **evaluation metrics**.
  - The goal of a **classifier** is to increase the **recall** without decreasing the **precision**.
  - **F-Measure:** measures the trade-off between **precision** and **recall**.

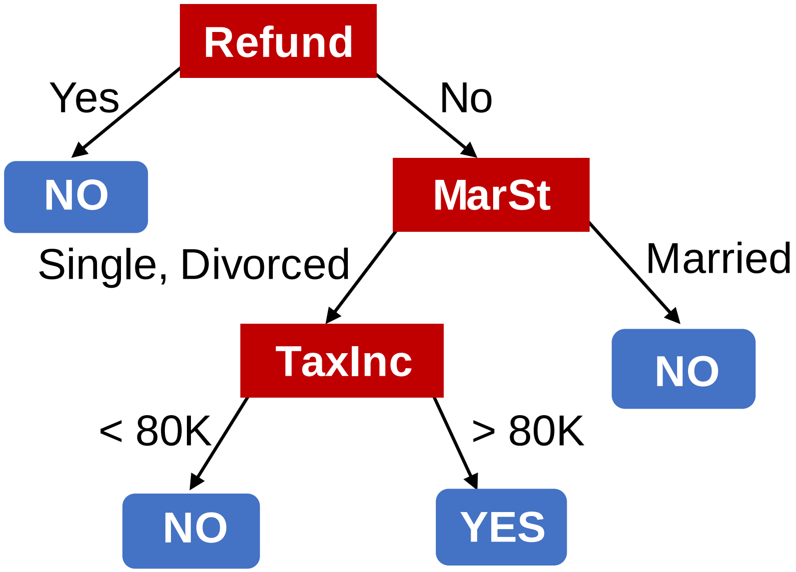
    $$F\text{-}Measure = \frac{(1+\beta^2)\cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision}$$

    where is a coefficient that specifies the relative importance of **precision** versus **recall**.
  - The coefficient of the **F-Measure** is usually set to 1 and the resulting metric is known as the **F1 Score**.

$$F1\ Score = \frac{2\cdot Recall \cdot Precision}{Recall + Precision}$$

Decision Trees
Introduction

- A **decision tree** uses a tree structure to represent a number of possible **decision paths** and an outcome for each path.
- ***Example***:



Decision tree to predict if someone will cheat on their taxes given their refund status, marital status, and taxable income

- A **decision tree** has three types of nodes: a **root node**, **internal nodes**, and **leaf** (or **terminal**) **nodes**.
- Every **leaf node** is associated with a **class label**.

Decision Trees
Algorithm

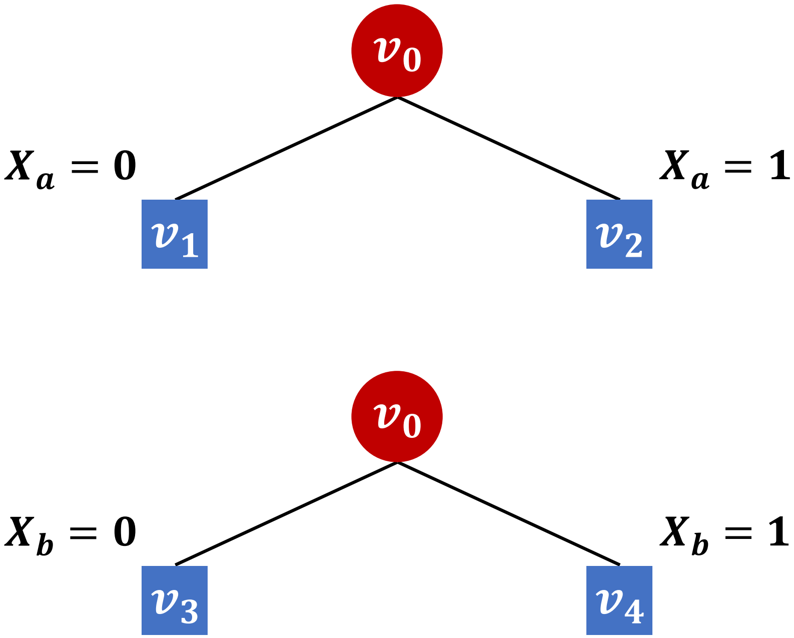
- Finding the optimal **decision tree** for a dataset is computationally expensive due to the **exponential size of the search space**.
- Efficient algorithms have been developed to induce accurate but suboptimal **decision trees** using a **greedy strategy**.
- **Greedy strategy for decision tree induction:**
  - Start with a single set containing all observations in the training set (**root node**).
  - Split set into subsets based on an attribute that optimizes certain **splitting criterion** (or **split evaluation function**).
  - Repeat on each subset until reaching certain **stopping criterion**.
- This greedy strategy is the basis for many current implementations of **decision tree** classifiers, including the **ID3**, **C4.5**, and **CART** algorithms.

Classification
Exercise 13.1

THINK PAIR SHARE

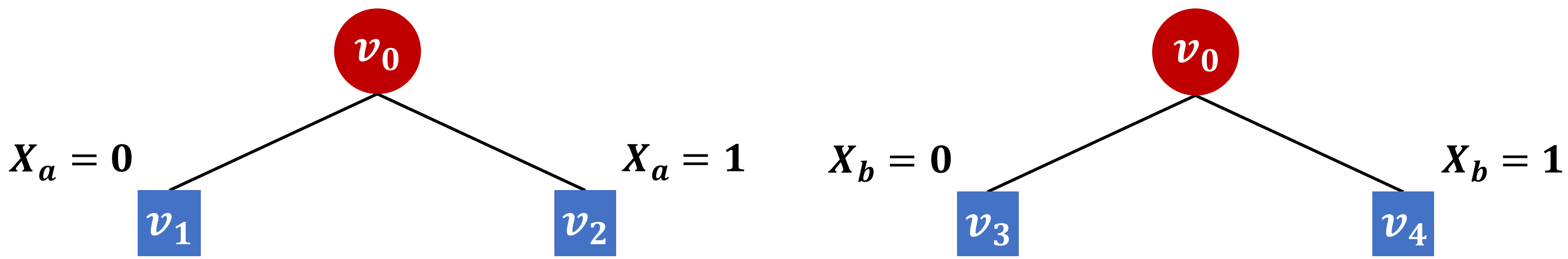
Suppose that you are building a decision tree from the following dataset. What is the best attribute to split on? Why?

X_a	X_b	Class
1	1	1
1	1	1
0	0	0
1	1	0
0	0	1
0	0	0
1	1	1
1	0	0
0	1	1
0	0	0



Decision Trees
Splitting Criterion

- **Which split is better?**



3 obs. from class 0 **2 obs. from class 1**	**2 obs. from class 0** **3 obs. from class 1**	**1 obs. from class 0** **4 obs. from class 1**	**1 obs. from class 0** **4 obs. from class 1**

Higher degree of impurity **Lower degree of impurity**

- Splits with **purer** nodes are preferred because a node where all observations have the same class label does not need to be expanded further.
- How do we measure the **impurity** of a node?
  - **Entropy**.
  - **Gini index**.

Decision Trees: Splitting Criterion
Entropy (I)

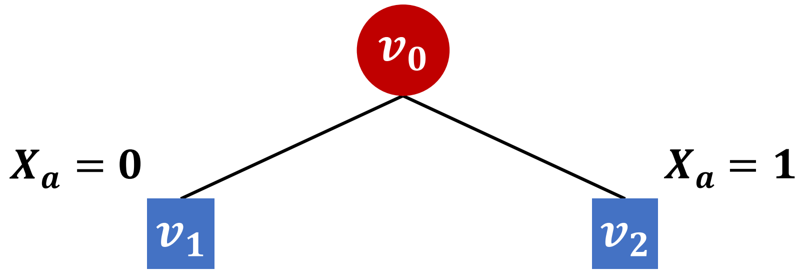
- **Entropy** measures the **impurity** of a node of the tree.
- The **entropy** of a node v_m is given by:

$$\boldsymbol{H(v_m) = -\sum_k p(y_k|v_m)\log_2 p(y_k|v_m)}$$

  where $\boldsymbol{p(y_k|v_m)}$ is the proportion of observations from class y_k at node v_m.
- **Minimum value:** 0.
  - All observations belong to the **same class**.
- **Maximum value:** $\log_2 k$ where k is the number of classes.
  - Observations are **equally distributed among all classes**.

Decision Trees: Splitting Criterion
Entropy (II)

- ***Example***:



3 obs. from class 0	**2 obs. from class 0**
2 obs. from class 1	**3 obs. from class 1**

$$H(v_0) = -\left(\frac{5}{10}\log_2\frac{5}{10}\right) - \left(\frac{5}{10}\log_2\frac{5}{10}\right) = \mathbf{1}$$

$$H(v_1) = -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right) = \mathbf{0.9710}$$

$$H(v_2) = -\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right) = \mathbf{0.9710}$$

Decision Trees: Splitting Criterion
Information Gain (I)

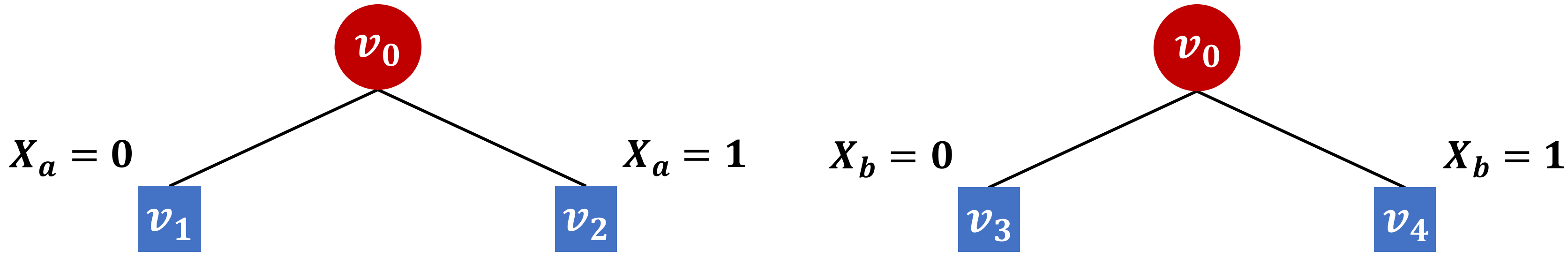
- **Information gain** measures the **difference in entropy** resulting from splitting a node of the tree on a given attribute.
- The **information gain** resulting from splitting a node v_m on attribute X_i into M nodes is given by:

$$\boldsymbol{IG(X_i, v_m) = H(v_m) - \sum_{h=1}^{M} \frac{N_h}{N} H(v_h)}$$

  where $\boldsymbol{N_h}$ is the number of observations at node v_m.
- The best attribute to split on is the one that **maximizes** the **information gain**.
- **Information gain** is the splitting criterion function used by the **ID3** and the **C4.5** algorithms.

Decision Trees: Splitting Criterion Information Gain (II)

- ***Example***:



3 obs. from class 0 **2 obs. from class 1**	**2 obs. from class 0** **3 obs. from class 1**	**1 obs. from class 0** **4 obs. from class 1**	**1 obs. from class 0** **4 obs. from class 1**

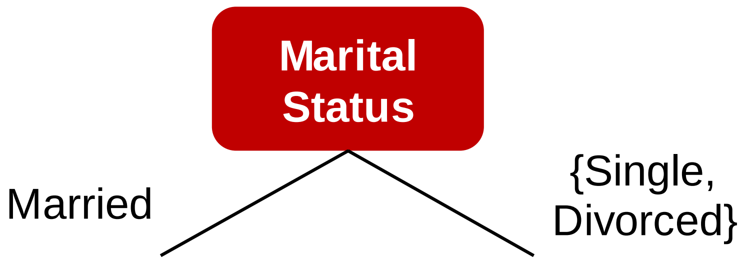
$$\boldsymbol{IG(X_a, v_0)} = H(v_0) - \left(\frac{5}{10} \cdot H(v_1)\right) - \left(\frac{5}{10} \cdot H(v_2)\right) = \mathbf{0.0290}$$

$$\boldsymbol{IG(X_b, v_o)} = H(v_0) - \left(\frac{5}{10} \cdot H(v_3)\right) - \left(\frac{5}{10} \cdot H(v_4)\right) = \mathbf{0.2781}$$

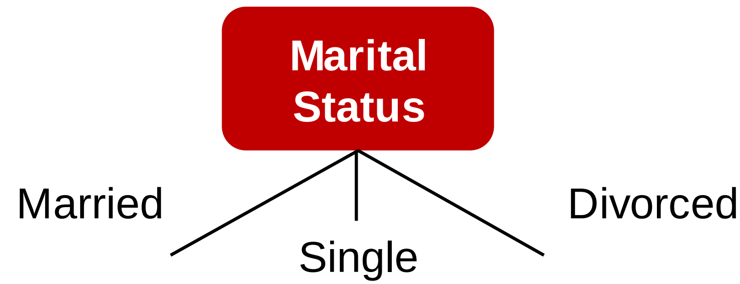
Splitting on X_b is better than splitting on X_a.

Decision Trees: Splitting Criterion
Non-Binary Attributes

- What if attributes have **more than two possible values**?

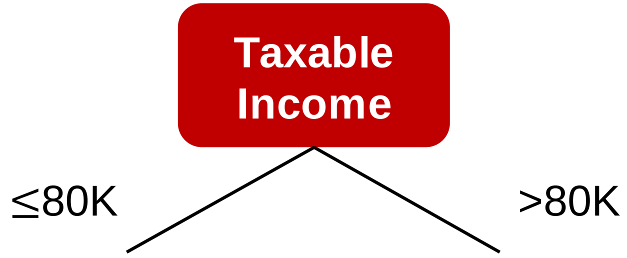


Binary split

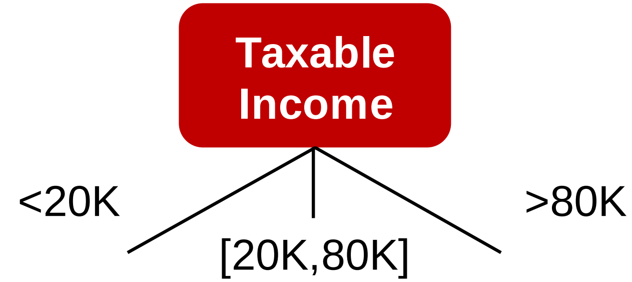


Multi-way split

- What if attributes are **continuous**?



Binary split



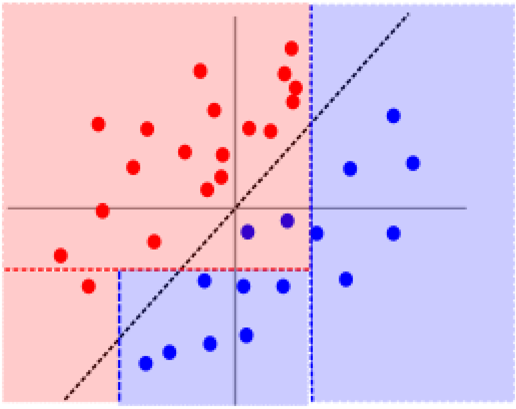
Multi-way split

Decision Trees
Stopping Criterion

- The basic algorithm for **decision tree** induction stops expanding a node only when all the observations associated with the node have the **same class label** or the **same attribute values**.
- In some cases, we may want to stop the **decision tree** induction algorithm early to avoid **overfitting**.
- There are two approaches to avoid **overfitting**.
  - **Pre-pruning** (or **early stopping**): in this approach, the **decision tree** induction algorithm is stopped before generating the full tree (for example, when the observed improvement in the **estimate of the generalization error is below a certain threshold**).
  - **Post-pruning**: in this approach, the **decision tree** is initially grown to its maximum size and is then trimmed. The tree pruning step terminates when **no further improvement in the estimate of the generalization error** is observed.

Decision Trees
Advantages and Disadvantages

- What are some of the **advantages** of **decision trees**?
  - **Interpretability**. **Decision trees** are very easy to understand and interpret.
  - **Applicability to a wide variety of datasets**. **Decision trees** are applicable to categorical and numeric attributes and make no assumptions about the probability distribution of the data.
  - **Computational efficiency**. Greedy algorithms for **decision tree** induction are very efficient.
  - **Robust to irrelevant and redundant attributes**.
- What are some of the **disadvantages** of **decision trees**?
  - **Easy to overfit**. Small changes in the data can lead to large changes in the structure of **decision trees**.
  - **Rectilinear decision boundaries**.



Classification
References

- Joel Grus. *Data Science from Scratch* (2015).
- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. *Introduction to Data Mining* (2019).