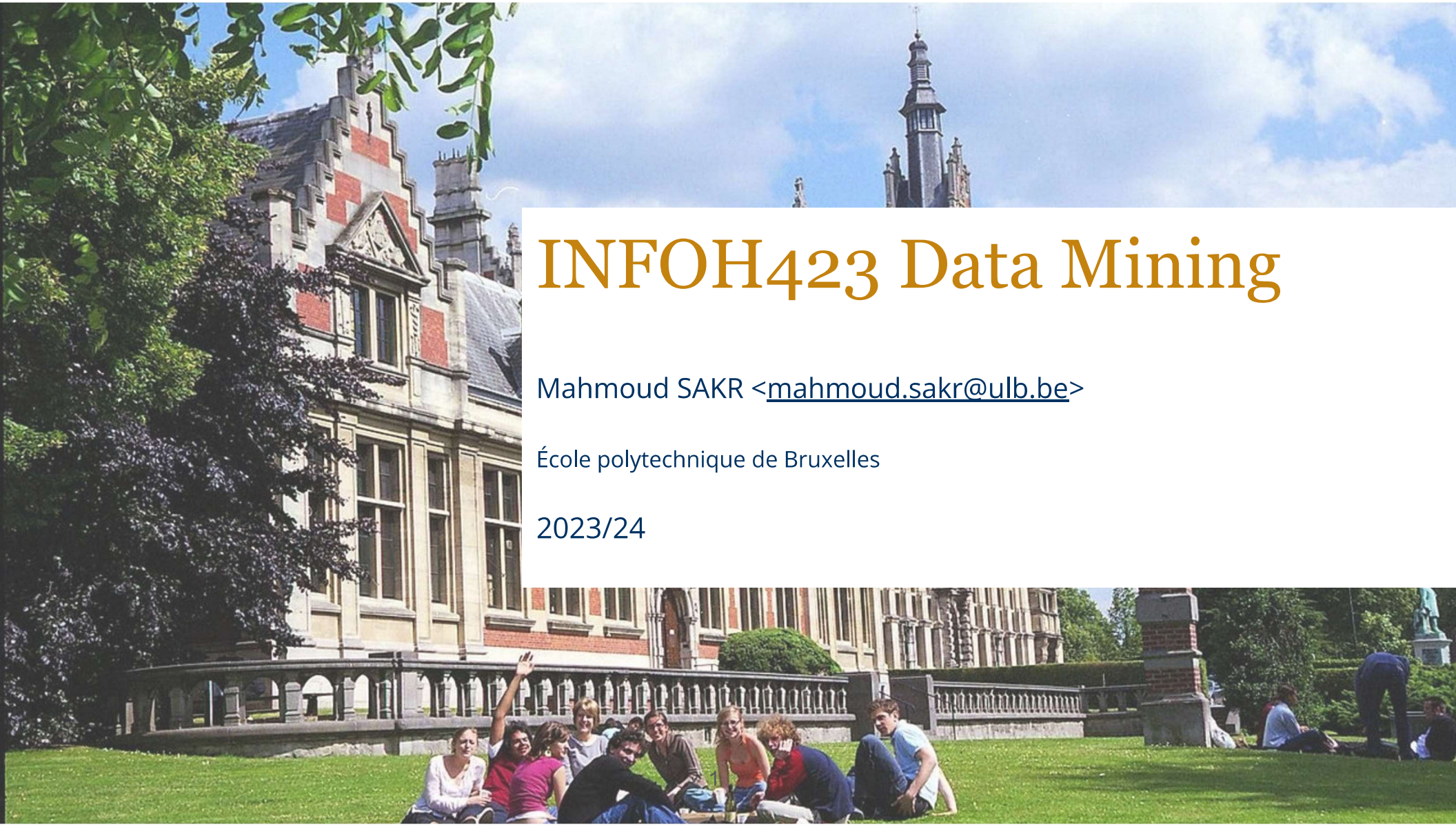


INFOH423 Data Mining

Mahmoud SAKR <mahmoud.sakr@ulb.be>

École polytechnique de Bruxelles

2023/24

What is data Mining ?

- Data mining is the study of collecting, cleaning, processing, analyzing, and gaining useful insights from data. [C.Aggarwal 2015].
- Data mining (knowledge discovery from data) is the process of nontrivial extraction of information from data, information that is implicitly present in that data, previously unknown and potentially useful for the user. [Frawley et al. Knowledge discovery in databases: an overview. 1992].

Course Goals

- To introduce the fundamental concepts and techniques of data mining
- To develop skills of using recent data mining software for solving practical problems
- To establish the main characteristics and limitations of algorithms for addressing data mining tasks
- To select the most appropriate combination of algorithms to solve a data mining problem
- To develop and execute a data mining workflow on real-life datasets, and to solve a data-driven analysis problem
- To identify promising business applications of data mining

Course Topics

- Classification.
- Model validation and data preparation
- Clustering
- Outlier mining
- Frequent pattern and association rule mining
- Stream data mining
- Applications

Prerequisites

- Good knowledge of programming
- General knowledge of Data structures, Algorithms and Complexity.
- General knowledge of Databases & SQL

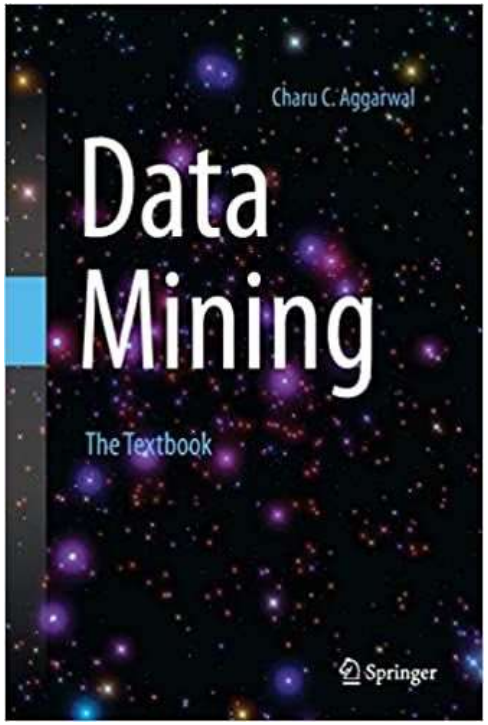
Course Organization

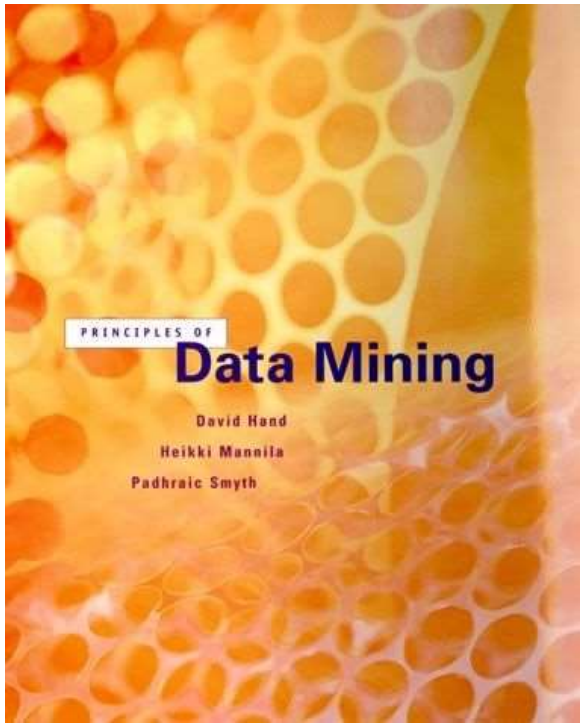
- Schedule and room on timedit: Check regularly for schedule updates
- Lectures by mahmoud.sakr@ulb.be
- Lab sessions by: Raphaël Gyori
    - Get your Rapidminer educational license
- Project
    - Practice a real world data science problem
    - Group project, 4 members
- Grading
    - Project 40%
    - Written exam 60%
- Course notes, please enroll in Université virtuelle

Skills

- Lectures cover: **Theory: Concepts, Algorithms, complexity**
- Lab sessions cover: **Rapidminer, quick prototyping, suitable for engineers**
- Project:
  - **Practice a real world data science problem**
  - **Apply the studied concepts, and further self learn**
  - **Advanced programming might be needed**

Recommended Readings





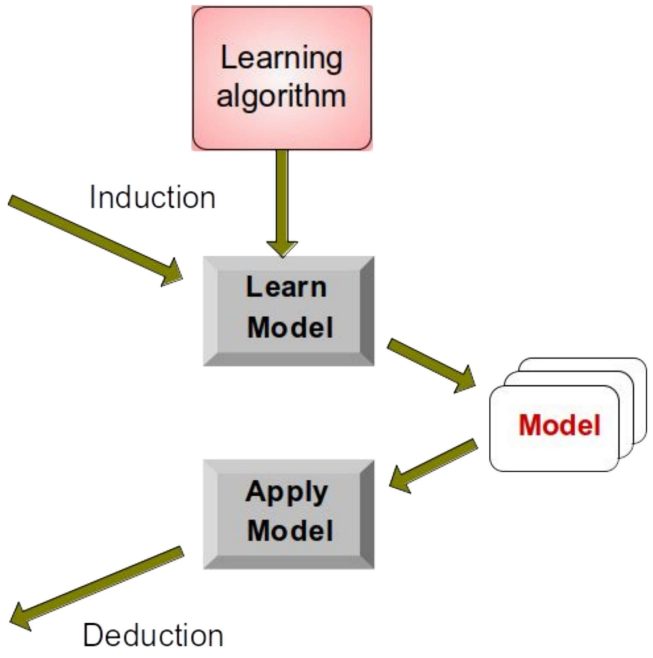
Classification

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification

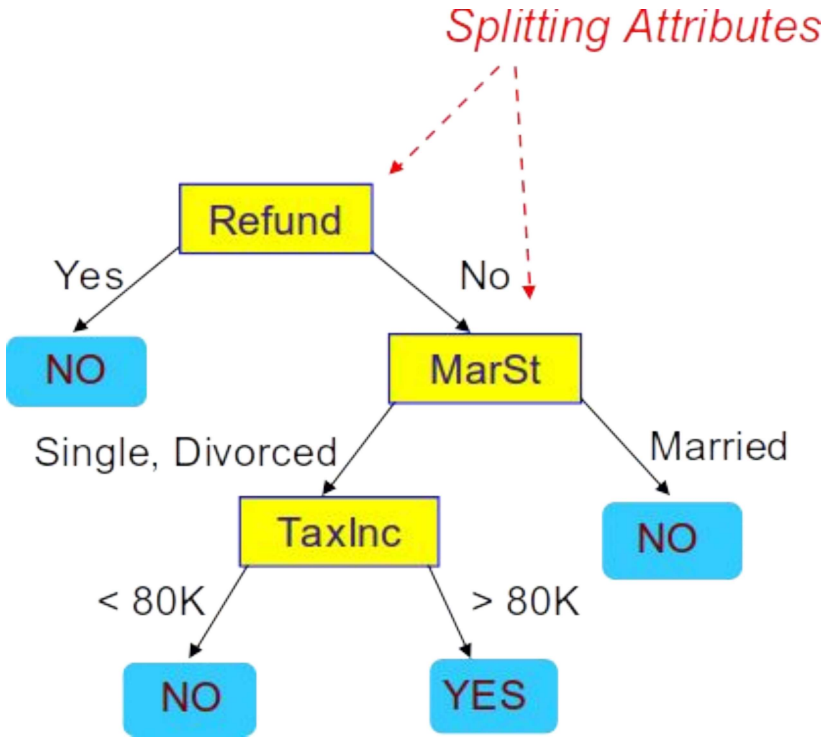
- Given a collection of records, training set, each record contains a set of attributes, one of the attributes is the class.
- Find a model for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.

Decision Tree Induction

Tid	Refund	Marital Status	Taxable Income	Cheat
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

categorical categorical continuous class

Training Data

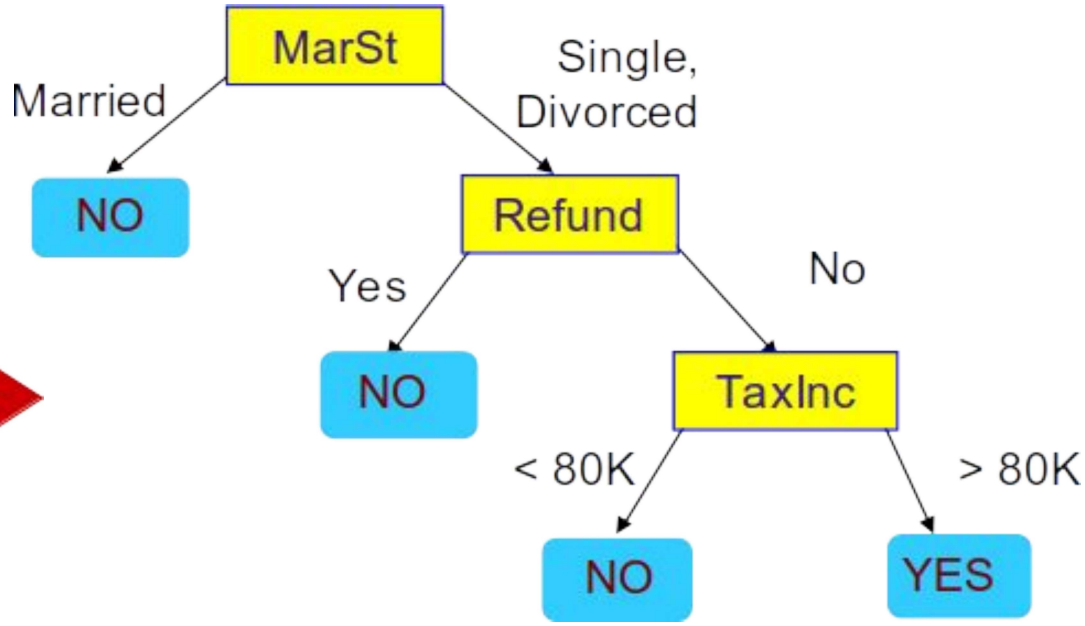


Model: Decision tree

Decision Tree Induction

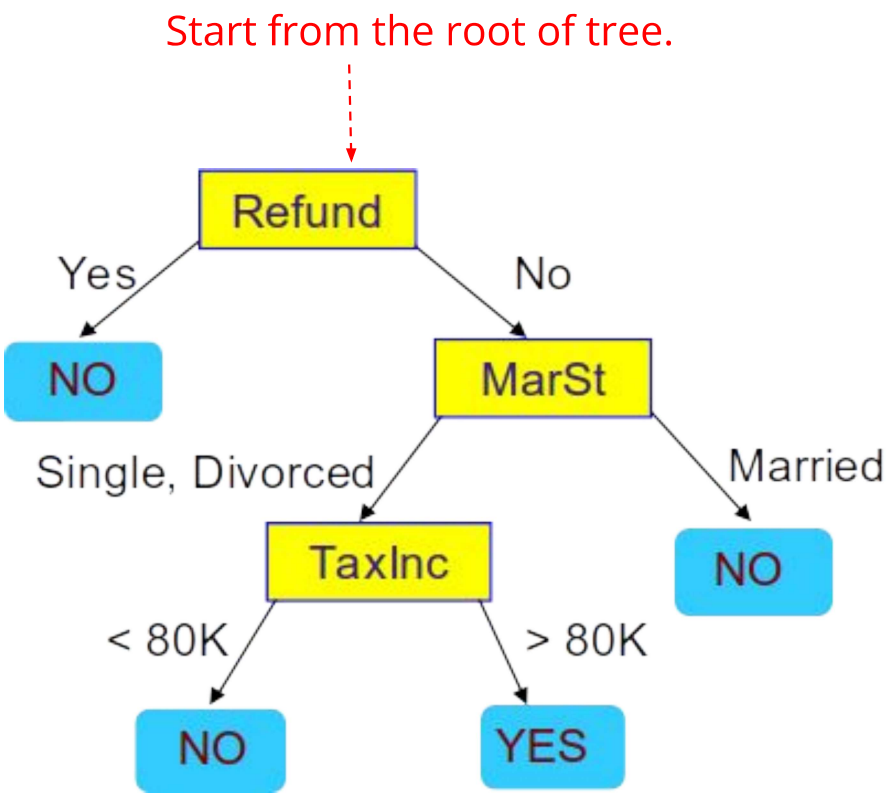
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

categorical categorical continuous class



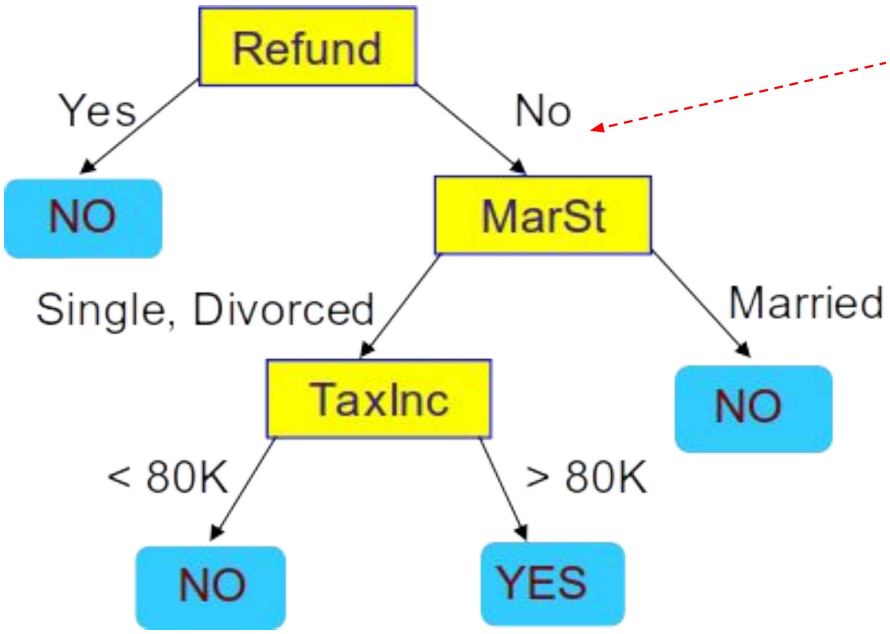
There could be more than one tree that fits the same data !

Decision Tree Prediction



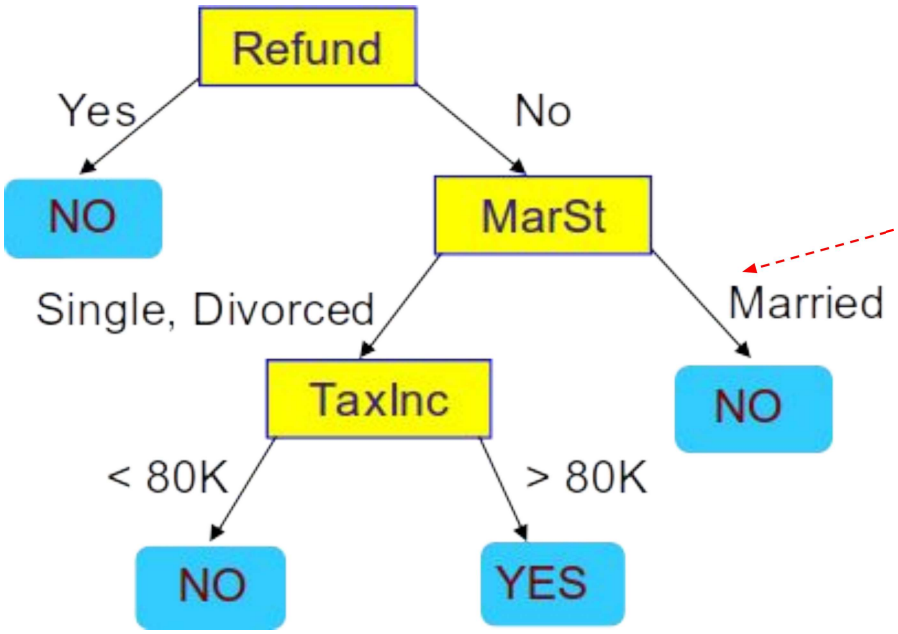
Refund	Marital Status	Taxable Income	Cheat
No	Married	80 k	?

Decision Tree Prediction



Refund	Marital Status	Taxable Income	Cheat
No	Married	80 k	?

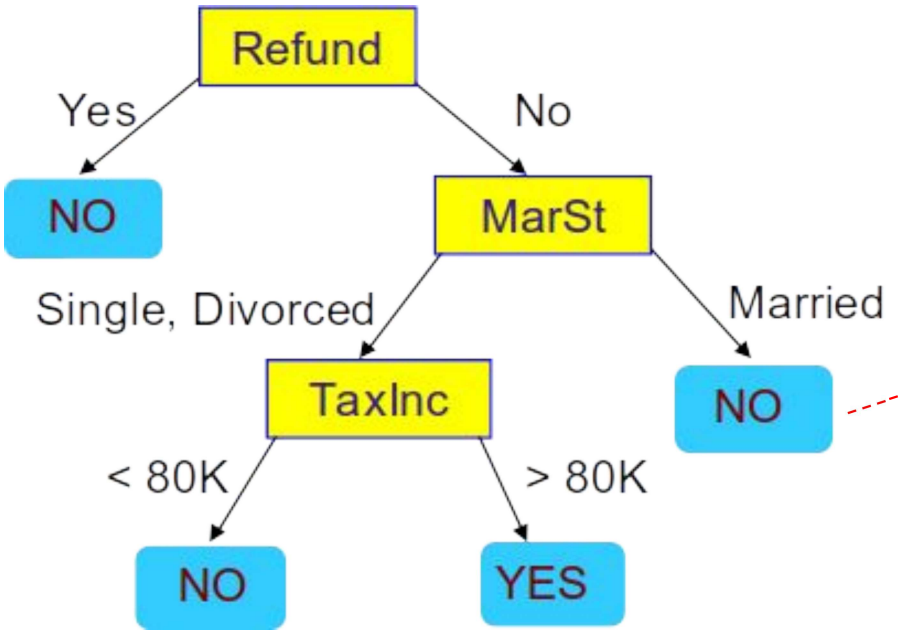
Decision Tree Prediction



Refund	Marital Status	Taxable Income	Cheat
No	Married	80 k	?

ULB

Decision Tree Prediction



Refund	Marital Status	Taxable Income	Cheat
No	Married	80 k	?

Predict Cheat= No

ULB

Exploratory Questions

1) Data Cleaning - Remove outliers, Handle missing values
2) Feature Selection: Select relevant features: Choose features that are less susceptible to noise. Feature selection techniques can help identify and retain the most informative features.
3) Feature Engineering:
Create new features: Derive new features that might capture useful information and reduce the impact of noise.
Discretize continuous features: Convert continuous variables into discrete ones to make the decision tree more robust to noise
4) Pruning: Prune the tree: Decision trees can be pruned to remove branches that contribute to overfitting. Pruning helps simplify the model and makes it more robust to noisy data.
5) Use ensemble methods
6) Weighted Samples: Assign weights to samples: Give higher weights to cleaner samples and lower weights to noisier samples. Some decision tree algorithms allow you to assign weights to instances, influencing their importance during the training process
7) Cross-Validation: Use cross-validation: Evaluate the model using cross-validation techniques to get a more robust estimate of its performance. This helps ensure that the model generalizes well to unseen data.

- How to encode a decision tree as a computer program ?
  by using sequence of if else block
- For the same dataset, multiple decision trees can be constructed. How to choose the best ? one that requires less number of search in decision tree
- How to deal with missing values ?
  ignore if there are less null values
  check if some default value can be added based on business usecase
  estimate the missing value
  use another decision tree to find missing value
- How to deal with noisy data ? up
- What is a best split for numerical values ?
  2 ways to do this
  1) r way split - Each split represents each distinct value
  2) binary split - Split based on binary condition x<=a for attribute value x and constant a
- Is there a way to incorporate uncertainty ?
- How to measure the correctness of a model ?
  training data can be divided into training and test data
  prepare model from training data
  test it via test data

Traditional decision trees typically output a deterministic prediction, but there are ways to account for uncertainty.
1. Probabilistic Decision Trees - Instead of providing a single prediction, some decision tree variants can estimate probabilities for each class. Techniques like Bayesian Decision Trees.
2. Bootstrap Aggregating (Bagging) - Bagging is an ensemble method that can be applied to decision trees. By training multiple decision trees on bootstrapped samples of the data and combining their predictions, you can obtain a more robust model that inherently captures some level of uncertainty.
3. Random Forests: Random Forests are an extension of bagging. They build multiple decision trees by using random subsets of features for each tree and then average the predictions.
This randomness adds a level of uncertainty to the model and makes it more robust.

Decision Tree Induction

Algorithm *GenericDecisionTree*(Data Set: $\mathcal{D}$)
begin
  Create root node containing $\mathcal{D}$;
  repeat
    Select an eligible node in the tree;
    Split the selected node into two or more nodes
      based on a pre-defined split criterion;
  until no more eligible nodes for split;
  Prune overfitting nodes from tree;
  Label each leaf node with its dominant class;
end

Many Algorithms: Hunt, CART, ID3, C4.5.

Entropy

- Entropy is a measure developed in Information Theory.
- Entropy is a measure of the uncertainty about a source of messages.
- The more uncertain a receiver is about a source of messages, the more information that receiver will need in order to know what message has been sent.
- If the source always sends exactly the same message, the entropy of such a source is zero.
- If the source send n messages, with equal probability, the uncertainty is maximized, as well as the entropy.
- In such a case, the receiver needs to ask $\log_2 n$ yes/no questions to know the message. In other words, the receiver needs to acquire $\log_2 n$ bits of information to know the message.
- Why ?

Entropy

- Consider a source S that can produce k messages (s_1, ..., s_k), independently, with equal probabilities, the entropy of such a source is:

$$E(S) = -\sum_{j=1}^{k} p_j \log_2(p_j)$$

Example

$$E(S) = -\sum_{j=1}^{k} p_j \log_2(p_j)$$

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

If the class attribute (the message) is 'play', what is the entropy of this source ?

Example

$$E(S) = -\sum_{j=1}^{k} p_j \log_2(p_j)$$

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

If the class attribute (the message) is 'play', what is the entropy of this source ?

Entropy (5/14, 9/14)= - ((5/14) $\log_2$ (5/14) + (9/14) $\log_2$(9/14))= 0.940

Entropy-Split

Now consider that we split the database into r groups (r-way split of a decision tree node). The information needed to identify a record within each group is calculated as in the previous slide. The weighted average over all groups is:

$$\text{Entropy-Split}(S \Rightarrow S_1 \dots S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} E(S_i)$$

Example

$$\text{Entropy-Split}(S \Rightarrow S_1 \ldots S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} E(S_i)$$

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

Entropy-split (humidity)=

[(7/14 E(humidity=high) + 7/14 E(humidity=normal)) =

[(-7/14 (4/7log(4/7) + 3/7log(3/7))) + (-7/14 (1/7log(1/7) + 6/7log(6/7)))] = 0.7884504573

Gain

- In choosing the split attribute in a decision tree, we are interested in how much information we get about the output attribute if we choose a certain input attribute as a split.
- This is just the difference between the information needed to classify the record before and after knowing the value of the input attribute.
- The information gain is equal to the reduction in the entropy:

$$E(S) - \text{Entropy-Split}(S \Rightarrow S_1 \dots S_r)$$

- Large values of the gain are more desirable.

Example

$$E(S) = -\sum_{j=1}^{k} p_j \log_2(p_j)$$

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

Gain (temperature)= E(S) – Entropy-Split(Temperature)= 0.94028595867 – 0.91106339301= 0.02922256565

ID3 Tree Induction Algorithm

- A mathematical algorithm for building the decision tree.
- Invented by J. Ross Quinlan in 1979.
- Uses Information Theory invented by Shannon in 1948.
- Builds the tree from the top down, with no backtracking.
- Information Gain is used to select the most useful attribute for classification.

```
function ID3 (I, O, T) {
/*  I is the set of input attributes
*   O is the output attribute
*   T is a set of training data
*
*   function ID3 returns a decision tree
*/
    if (T is empty) {
        return a single node with the value "Failure";
    }
    if (all records in T have the same value for O) {
        return a single node with that value;
    }
    if (I is empty) {
        return a single node with the value of the most frequent value of
        O in T;
        /* Note: some elements in this node will be incorrectly classified */
    }

    /* now handle the case where we can't return a single node */
    compute the information gain for each attribute in I relative to T;
    let X be the attribute with largest Gain(X, T) of the attributes in I;
    let {x_j| j=1,2, .., m} be the values of X;
    let {T_j| j=1,2, .., m} be the subsets of T when T is partitioned
        according the value of X;
        return a tree with the root node labelled X and
        arcs labelled x_1, x_2, .., x_m, where the arcs go to the
        trees ID3(I-{X}, O, T_1), ID3(I-{X}, O, T_2), .., ID3(I-{X}, O, T_m);
}
```

Example

Using ID3, induce the decision tree of:

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

The problem of UID

- What will happen if a unique attribute exists in the training set ?
- In general attributes that have very many values have very high gain, but can lead to useless decision trees.
- Quinlan (1986) suggests choosing the attribute with the highest:
  - GainRatio(X, S)= Gain(X) / Entropy(S) when X is the label attribute
  - GainRatio is proportional with the Gain, favoring attributes with higher gain as before.
  - GainRatio is inversely proportional with the Entropy of the attribute, discouraging attributes with many values.
- Repeat the example using the GainRatio.

Model Validation

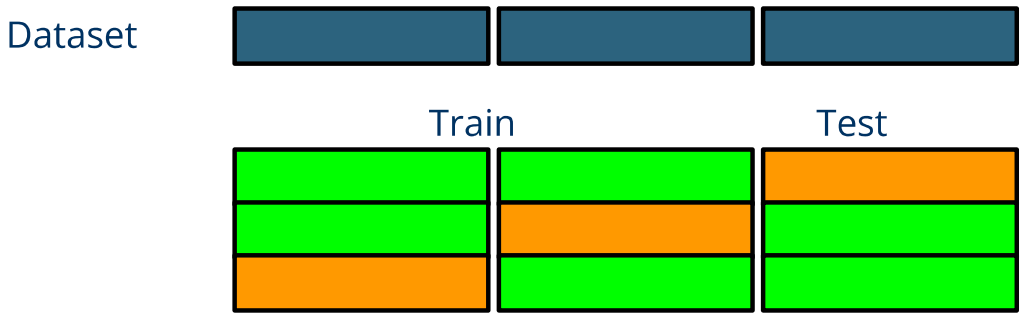
- Holdout

Labelled data is divided into two disjoint sets, training data and testing data, typically more data in training set
when large training data, there is high probability that model will predict test data correctly and vice versa
There is chance that data with certain class may completely be present in training data but absent in test data in which case model will not be able to predict correctly
since only subset of data is used for training, full power of training data is not reflected in error estimate
can also repeat process n times and mean error estimate can be determined

  - Divide labeled data into two disjoint subsets (training set 60%-75%, testing set), randomly selected.
  - Validate the model accuracy using the test set.
  - Classes that are over-represented in the training, will be under-represented in the testing.
- Cross Validation
  - The labeled data is divided into m disjoint subsets of equal size n/m.
  - One of the m segments is used for testing, and the other (m − 1) segments are used for training.

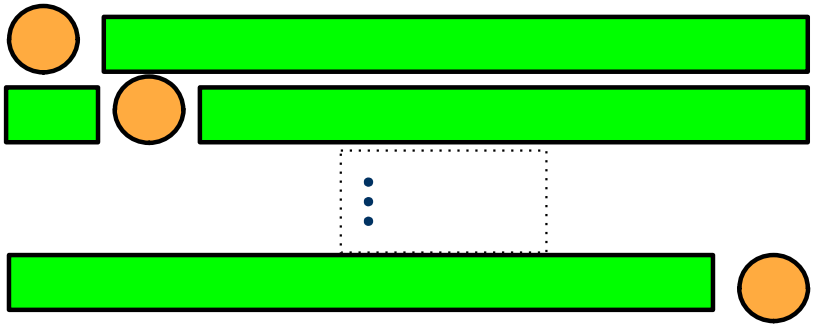
Common Splitting Strategies

- k-fold cross-validation



- Leave-one-out (n-fold cross validation)

data with n records is divided into m subset where (n=m)
each time 1 record is treated as test and m-1 records as training data



Cross Validation

- Average results reported.
- The variance helps determining the statistical confidence intervals of the error.
- Helps to qualify the model. If the model is not performing well on the average, another model should be used.
- Stratified cross validation splits the folds, so that every fold has almost the same distribution of class labels as in the complete dataset.
- Stratified cross validation leads to less pessimistic estimates.

Bootstrap

- Data is sampled uniformly with replacement to create the training set.
- A set of size n is sampled n times, each time a record is randomly selected, copies, then returned back.
- This results in n records that may contain duplicates, and that can miss some of the input records.
- The probability that a certain record is missed in one sampling is 1- 1/n.
- The probability that a certain record is missed in n samplings is $(1- 1/n)^n$. For large n, this approaches 1/e.
- The fraction of the labeled data points included at least once in the training data is therefore $1-1/e \approx 0.632$.

Bootstrap

- The training set hence consists of ≈63.2 % of original dataset, and ≈36.8 % duplicates.
- The original dataset is used as test set to estimate the model accuracy.
- This estimate is highly optimistic because of the high overlap (63.2 %) between training and testing datasets.
- To obtain the variance (and confidence intervals), the whole Bootstrap can be repeated k times.

Bootstrap

- The leave one out bootstrap is a pessimistic variant.
- The accuracy of a labeled record x in the test set is computed using only the b bootstraps (training datasets) that don't include x.
- The overall accuracy is the average of the accuracy estimates for all possible x.

Bootstrap

- The 0.632-bootstrap further improves this accuracy with a “compromise” approach.

$$A = (0.632) \cdot A_l + (0.368) \cdot A_t$$

- A is the overall accuracy.
- A_l is the overall accuracy of leave-on-out bootstrap (pessimistic).
- A_t is the overall accuracy of bootstrap (optimistic).

Credits

- These slides are mostly made by copying material from:
- Classification: Basic Concepts and Decision Trees, slides by Ruoming Jin,

  http://www.cs.kent.edu/~jin/DM07/ClassificationDecisionTree.ppt
- The ID3 Decision Tree Algorithm, by Daniela Zaharie,

  http://staff.fmi.uvt.ro/~daniela.zaharie/dm2016/projects/DecisionTrees/DecisionTrees_ID3Tutorial.pdf