Decision and Regression Trees: Exercise

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

Exercise 1
Theoretical Questions

Theoretical Questions

1. *What are the differences among ID3, CART, and C4.5?*
2. *How can we avoid overfitting in decision tree training?*
3. *Describe some commonly used pruning techniques for decision trees*
4. *Describe differences and similarities between regression trees and linear regression*

What are the differences among ID3, CART, and C4.5?

	Splitting	Attributes	Missing values	Pruning	Outliers
ID3	Information Gain	Only Categorical	Not handled	No pruning	Susceptible
CART	Gini index / Towing Criteria	Categorical and Numeric	Handled	Cost-complexity pruning	Handled
C4.5	Gain Ratio	Categorical and Numeric	Handled	Error-based pruning	Susceptible

Source: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf

How can we avoid overfitting in decision tree training?

Goal: A tree that generalizes to new data and doesn't overfit

Early stopping

- Idea: Define a minimum size for the tree leaves

Pruning

- Idea: Cut branches that seem to overfit
- Techniques
  - Reduced Error Pruning
  - Pessimistic Pruning
  - Confidence Level Pruning
  - Minimum description length

3. Pruning

Describe some commonly used pruning techniques for decision trees

- Reduced Error Pruning
- Pessimistic Pruning
- Confidence Level Pruning
- Minimum description length

3. Reduced Error Pruning

- Classify a set of **new example cases** with the decision tree. (These cases must not have been used for the learning!)
- Determine the **number of errors** for all leaves.
- The number of errors of a subtree is the sum of the errors of all of its leaves.
- Determine the number of errors for leaves that replace subtrees.
- If such a leaf leads to the same or fewer errors than the subtree, replace the subtree by the leaf.
- If a subtree has been replaced, recompute the number of errors of the subtrees it is part of.

3. Reduced Error Pruning

- **Advantage:**
  Very good pruning, effective avoidance of overfitting.

- **Disadvantage:**
  Additional example cases needed. Number of cases in a leaf has no influence.

3. Pessimistic Pruning

- Classify a set of example cases with the decision tree. (These cases **may or may not** have been used for the learning.)
- Determine the number of errors for all leaves and **increase this number by a fixed, user-specified amount** ***r*****.**
- The number of errors of a subtree is the sum of the errors of all of its leaves.
- Determine the number of errors for leaves that replace subtrees (also increased by *r*).
- If such a leaf leads to the same or fewer errors than the subtree, replace the subtree by the leaf and recompute subtree errors.

3. Pessimistic Pruning

- **Advantage:**
  No additional example cases needed.

- **Disadvantage:**
  Number of cases in a leaf has no influence.

Like pessimistic pruning, but the number of errors is computed as follows:

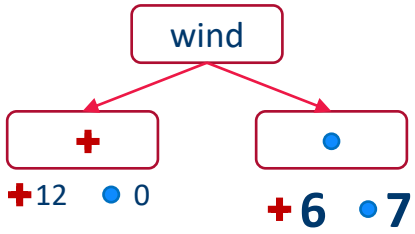
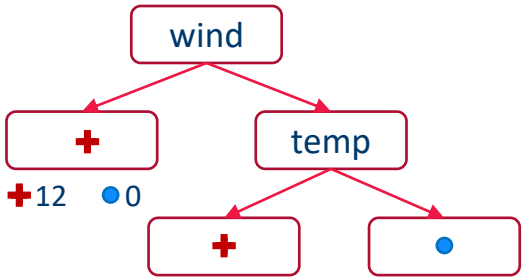
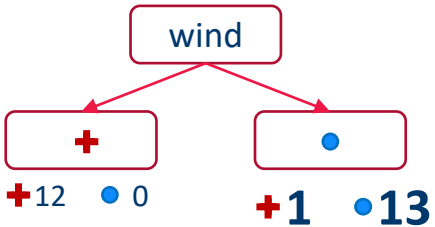
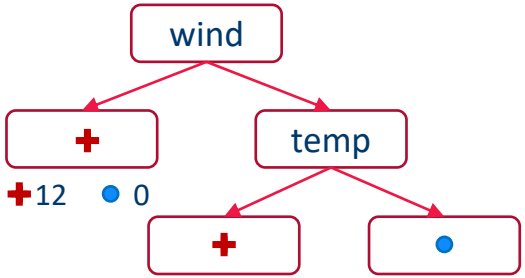
- See classification in a leaf as a Bernoulli experiment (error/no error): p, $p(1-p)$
  - Expected success rate: $f = \frac{no\ error}{error + no\ error}$
  - For a large enough number of classifications f follows a normal distribution
- Estimate an interval for the error probability $p(1-p)$ based on a user-specified confidence level α. (use approximation of the binomial distribution by a normal distribution)
- Increase error number to the upper level of the confidence interval times the number of cases assigned to the leaf.
- Formal problem: Classification is not a random experiment.

3. Confidence Level Pruning

- **Advantage:**
  No additional example cases needed. Good pruning.

- **Disadvantage:**
  Statistically dubious foundation.

Description length = #bits(tree) + #bits(misclassified samples)

	Tree 1	Tree 2	Note
Example 1			Many misclassified in samples in tree 1 => DL(Tree 1) > DL(Tree 2) => Select Tree 2
Example 2			Only 1 misclassified sample in tree 1 => DL(Tree 1) < DL(Tree 2) => Select Tree 1

Describe differences and similarities between regression trees and linear regression

- **Similarities**: both implement solutions to numerical prediction tasks
- **Differences**:
  - Closed form solution for linear regression; heuristic solution for regression tree
  - Linear regression uses a linear function to fit the data
  - Regression tree uses a step-wise function to fit the data

Exercise 2
Hands-on Decision Tree

Hands-on Decision Tree

- *Check the examples from dataset in the table below*
  - *Three boolean values A1; A2; A3*
  - *Class attribute C.*

A1	F	W	F	W	F	F	W	W
A2	F	W	F	W	W	W	F	F
A3	W	W	W	F	F	F	W	F
C	W	W	W	W	F	F	F	F

- *Use the ID3-algorithm (by hand, no tools) to construct a decision tree. The solution should contain all necessary calculations for building the decision tree.*
- *Why is the resulting decision tree too complex? How can this be prevented?*

Use the ID3-algorithm (by hand, no tools) to construct a decision tree

$C = \{4\ \mathrm{W}, 4\ \mathrm{F}\}$

$$H(C) = -\frac{4}{8} log_2 \frac{4}{8} - \frac{4}{8} log_2 \frac{4}{8} = \frac{4}{8} + \frac{4}{8} = 1$$

A1	F	W	F	W	F	F	W	W
A2	F	W	F	W	W	W	F	F
A3	W	W	W	F	F	F	W	F
C	W	W	W	W	F	F	F	F

Use the ID3-algorithm (by hand, no tools) to construct a decision tree

$C|A1$:

- for $A1 = F \Rightarrow C' = \{2W, 2F\}$
- for $A1 = W \Rightarrow C'' = \{2W, 2F\}$

Same values for $C|A2$

A1	F	W	F	W	F	F	W	W
A2	F	W	F	W	W	W	F	F
A3	W	W	W	F	F	F	W	F
C	W	W	W	W	F	F	F	F

$$H(C|A1) = H(C|A2) = \frac{4}{8}\left(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}\right) + \frac{4}{8}\left(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}\right) = 1$$

$$I_{gain}(C, A1) = I_{gain}(C, A2) = 1 - 1 = 0$$

Use the ID3-algorithm (by hand, no tools) to construct a decision tree

$C|A3$:

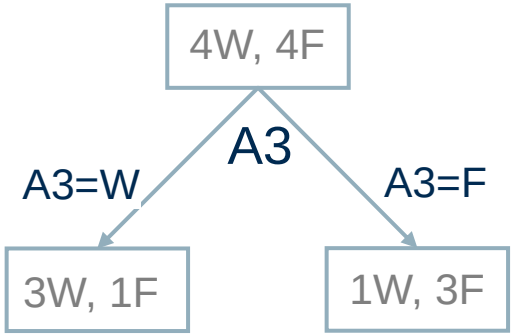
- for $A3 = F \Rightarrow C' = \{1W, 3F\}$
- for $A3 = W \Rightarrow C'' = \{3W, 1F\}$

A1	F	W	F	W	F	F	W	W
A2	F	W	F	W	W	W	F	F
A3	W	W	W	F	F	F	W	F
C	W	W	W	W	F	F	F	F

$$H(C|A3) = \frac{4}{8}\left(-\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4}\right) + \frac{4}{8}\left(-\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4}\right) = (0.311 + 0.5) = 0.811$$

$$I_{gain}(C, A3) = 1 - 0.811 = 0.189$$

Hands-on Decision Tree



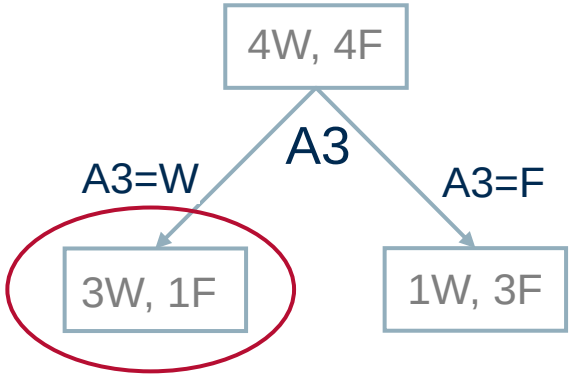
A3=W

A1	F	W	F	W
A2	F	W	F	F
A3	W	W	W	W
C	W	W	W	F

A3=F

A1	W	F	F	W
A2	W	W	W	F
A3	F	F	F	F
C	W	F	F	F

Hands-on Decision Tree



A3=W

A1	F	W	F	W
A2	F	W	F	F
A3	W	W	W	W
C	W	W	W	F

$$H(C) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.311 + 0.5 = 0.811$$

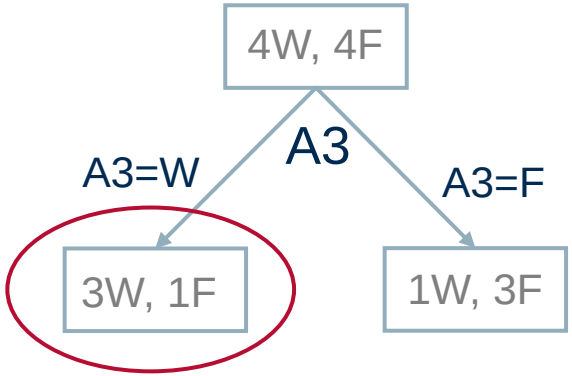
$C|A1$:

- for $A1 = F \Rightarrow \{2W\}$
- for $A1 = W \Rightarrow \{1W, 1F\}$

$$H(C|A1) = \frac{2}{4}\left(-\frac{2}{2}log_2\frac{2}{2}\right) + \frac{2}{4}\left(-\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2}\right) = 0.5$$

$$I_{gain}(C, A1) = 0.811 - 0.5 = 0.311$$

Hands-on Decision Tree



A3=W

A1	F	W	F	W
A2	F	W	F	F
A3	W	W	W	W
C	W	W	W	F

$$H(C) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.311 + 0.5 = 0.811$$

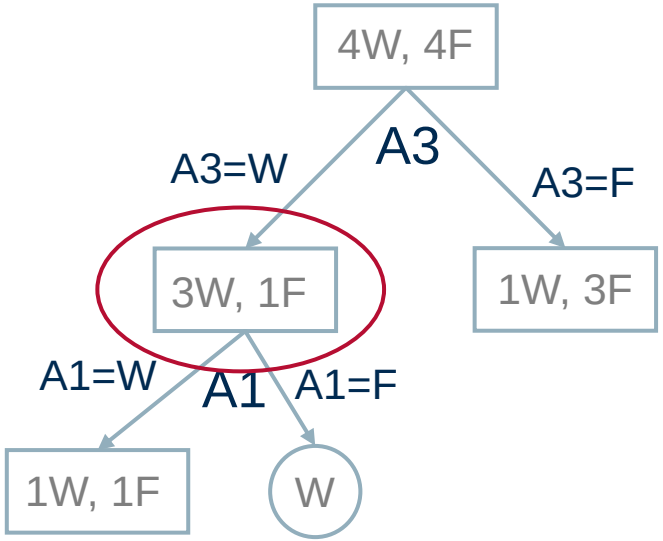
$C|A2$:

- for $A2 = F \Rightarrow \{2W, 1F\}$
- for $A2 = W \Rightarrow \{1W\}$

$$H(C|A2) = \frac{3}{4}\left(-\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}\right) + \frac{1}{4}(-1log_2 1) = 0.693$$

$$I_{gain}(C, A2) = 0.811 - 0.693 = 0.118$$

Hands-on Decision Tree

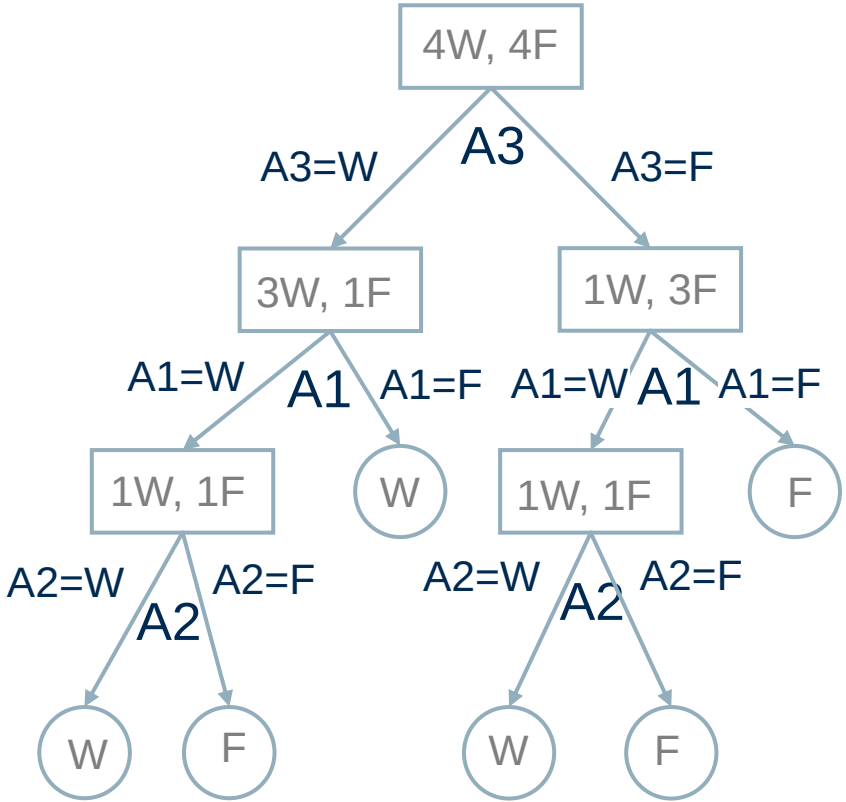


A1	F	W	F	W
A2	F	W	F	F
A3	W	W	W	W
C	W	W	W	F

A1	W	W
A2	W	F
A3	W	W
C	W	F

A1	F	F
A2	F	F
A3	W	W
C	W	W

Hands-on Decision Tree



A3=W

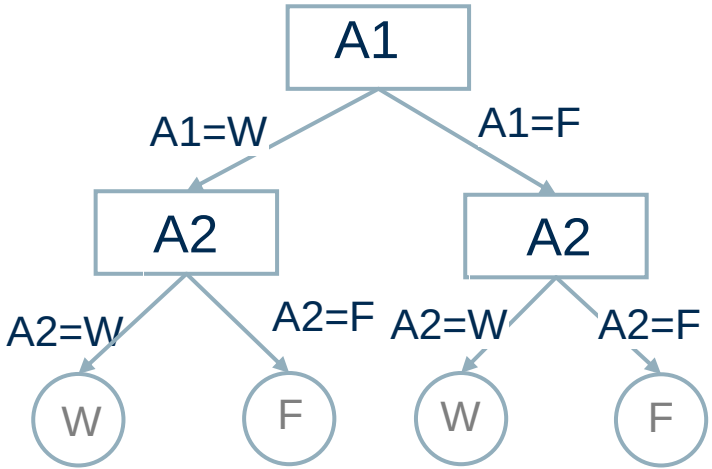
A1	F	W	F	W
A2	F	W	F	F
A3	W	W	W	W
C	W	W	W	F

A3=F

A1	W	F	F	W
A2	W	W	W	F
A3	F	F	F	F
C	W	F	F	F

Why is the resulting decision tree too complex? How can this be prevented?

- From the previous slide, because there are redundant examples in the training data, the attribute A3 is overrated (high information gain). If the duplicates are removed, the decision tree becomes simpler and more intuitive.
- The decision tree describes the function of NOT XOR. This example shows that the ID3 algorithm does not always construct the smallest decision tree. However, with respect to the distribution of training examples the algorithm is correct.



Exercise 3
Entropy and Information Gain

Given the following examples of a binary task with input attributes A1, A2.

a) *Calculate the entropy H(C) of the class distribution.*

b) *Calculate the expected entropy H(C | A1) and H(C | A2) of the class distribution when A1 or A2 are known.*

c) *Calculate the information gain I_{gain}(C; A1) and I_{gain}(C; A2)*

X	1	2	3	4	5	6
A1	T	T	T	F	F	F
A2	T	T	F	F	T	T
C	+	+	-	+	-	-

a) Calculate the entropy H(C) of the class distribution.

X	1	2	3	4	5	6
A1	T	T	T	F	F	F
A2	T	T	F	F	T	T
C	+	+	-	+	-	-

$$H(C) = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{3}{6} log_2 \frac{3}{6} - \frac{3}{6} log_2 \frac{3}{6} = 1$$

b) Calculate the expected entropy H(C | A1) and H(C | A2) of the class distribution when A1 or A2 are known.

$C|A1$:

- for $A1 = F \Rightarrow \{2-, 1+\}$
- for $A1 = T \Rightarrow \{2+, 1-\}$

$$H(C|A1) = \frac{3}{6}\left(-\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}\right) + \frac{3}{6}\left(-\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3}\right) = 0.918$$

$C|A2$:

- for $A2 = F \Rightarrow \{1-, 1+\}$
- for $A2 = T \Rightarrow \{2+, 2-\}$

$$H(C|A2) = \frac{4}{6}\left(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}\right) + \frac{2}{6}\left(-\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2}\right) = 1$$

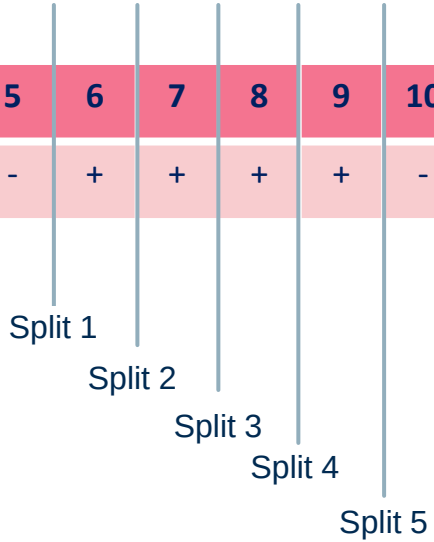
c) *Calculate the information gain $I_{gain}(C; A1)$ and $I_{gain}(C; A2)$*

X	1	2	3	4	5	6
A1	T	T	T	F	F	F
A2	T	T	F	F	T	T
C	+	+	-	+	-	-

- $I_{gain}(C, A) = H(C) - H(C|A)$
- $I_{gain}(C, A1) = 1 - 0.918 = 0.082$
- $I_{gain}(C, A2) = 1 - 1 = 0$

– *Given the following table, calculate the original information gain and the information gain for all 5 possible splits. What do you notice?*

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+



2. Information Gain: Split 1

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+

Split 1

$$H = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

- Split 1:

$$H^1_{left} = -\frac{1}{5} log_2 \frac{1}{5} - \frac{4}{5} log_2 \frac{4}{5} = 0.722$$

$$H^1_{right} = -\frac{8}{9} log_2 \frac{8}{9} - \frac{1}{9} log_2 \frac{1}{9} = 0.503$$

$$I_1 = H - \frac{5}{14} H^1_{left} - \frac{9}{14} H^1_{right} = 0.940 - 0.258 - 0.323 = \boxed{0.359}$$

2. Information Gain: Split 2

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+

Split 2

$$H = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

- Split 2:

$$H^2_{left} = -\frac{2}{6} log_2 \frac{2}{6} - \frac{4}{6} log_2 \frac{4}{6} = 0.918$$

$$H^2_{right} = -\frac{7}{8} log_2 \frac{7}{8} - \frac{1}{8} log_2 \frac{1}{8} = 0.544$$

$$I_2 = H - \frac{6}{14} H^2_{left} - \frac{8}{14} H^2_{right} = 0.940 - 0.393 - 0.311 = \boxed{0.236}$$

2. Information Gain: Split 3

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+

Split 3

$$H = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

- Split 3:

$$H^3_{left} = -\frac{3}{7} log_2 \frac{3}{7} - \frac{4}{7} log_2 \frac{4}{7} = 0.985$$

$$H^3_{right} = -\frac{6}{7} log_2 \frac{6}{7} - \frac{1}{7} log_2 \frac{1}{7} = 0.592$$

$$I_3 = H - \frac{7}{14} H^3_{left} - \frac{7}{14} H^3_{right} = 0.940 - 0.493 - 0.296 = 0.151$$

2. Information Gain: Split 4

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+

Split 4

$$H = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

– Split 4:

$$H^4_{left} = -\frac{4}{8} log_2 \frac{4}{8} - \frac{4}{8} log_2 \frac{4}{8} = 1.0$$

$$H^4_{right} = -\frac{5}{6} log_2 \frac{5}{6} - \frac{1}{6} log_2 \frac{1}{6} = 0.650$$

$$I_4 = H - \frac{8}{14} H^4_{left} - \frac{6}{14} H^4_{right} = 0.940 - 0.571 - 0.279 = \boxed{0.090}$$

2. Information Gain: Split 5

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	-	-	+	-	-	+	+	+	+	-	+	+	+	+

Split 5

$$H = -\sum_{i=1}^{n_C} p_i \, log_2 p_i = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

– Split 5:

$$H_{left}^5 = -\frac{5}{9} log_2 \frac{5}{9} - \frac{4}{9} log_2 \frac{4}{9} = 0.991$$

$$H_{right}^5 = -\frac{4}{5} log_2 \frac{4}{5} - \frac{1}{5} log_2 \frac{1}{5} = 0.722$$

$$I_5 = H - \frac{9}{14} H_{left}^5 - \frac{5}{14} H_{right}^5 = 0.940 - 0.637 - 0.258 = \boxed{0.045}$$

Exercise 4
Practice with KNIME

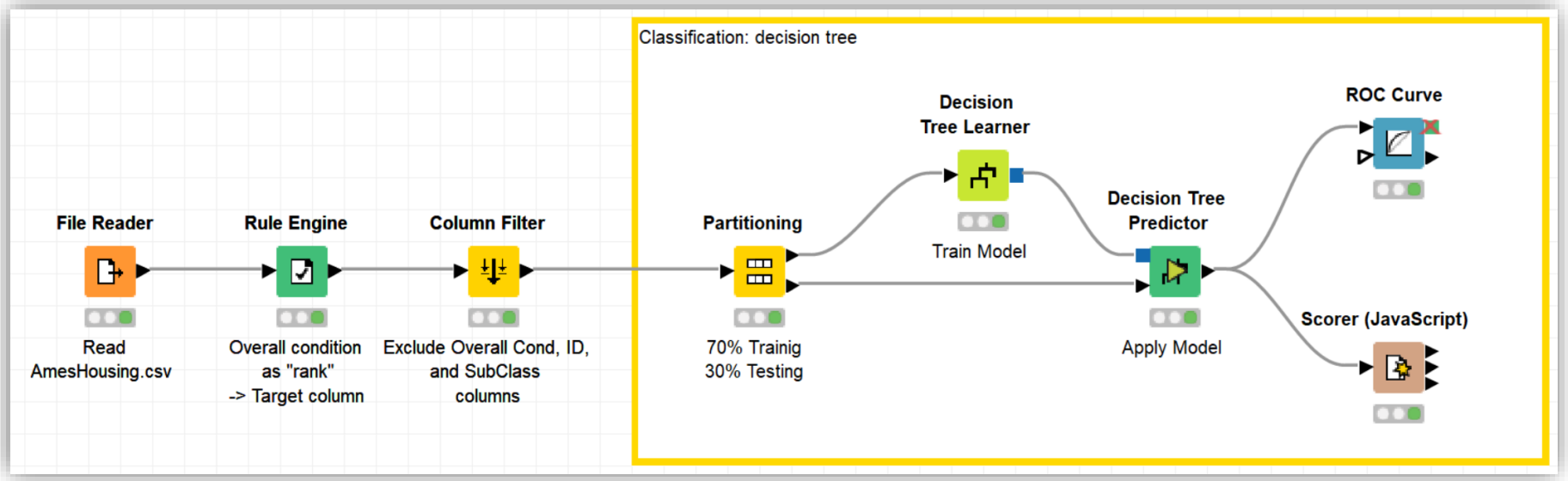
1. Decision Tree

Train a binary classification model that predicts the overall condition of a house (either “high” or “low”).

1. Read the *AmesHousing.csv* file. It describes the sale of individual residential properties in Ames, Iowa (USA). One of the columns is the overall condition ranking, with values between 1 and 10.
2. Create a new column "rank" using the **Rule Engine node**.
   - An house will have rank "Low" if the value of the attribute "Overall Cond" is <= 5. Otherwise it will be ranked as "High".
3. Remove the following column
   - PID
   - MS SubClass
   - Overall Cond
4. Use a **Partitioning node** to split data into training (70%) e test set (30%)
   - Use stratified sampling based on the column rank, to retain the distribution of the class values in both output tables.
5. Train a Decision Tree model to predict the overall condition of a house (high/low) (**Decision Tree Learner node)**
   - Select the "rank" column as the class column
6. Use the trained model to predict the rank of the houses in the test set (**Decision Tree Predictor node)**
7. Evaluate the accuracy of the decision tree model (**Scorer (Java Script) node)**
   - What is the accuracy of the model?
8. Visualize the ROC curve (**ROC Curve node)**
   - Make sure that checkbox "append columns with normalized class distribution" in the Decision Tree Predictor node is activated
   - Select "rank" as Class column and "High" as Positive class value. Include only the "P (rank=High)" column
9. Try different setting options for the decision tree algorithm. Can you improve the model performance?

1. Decision Tree

Train a binary classification model that predicts the overall condition of a house (either “high” or “low”).



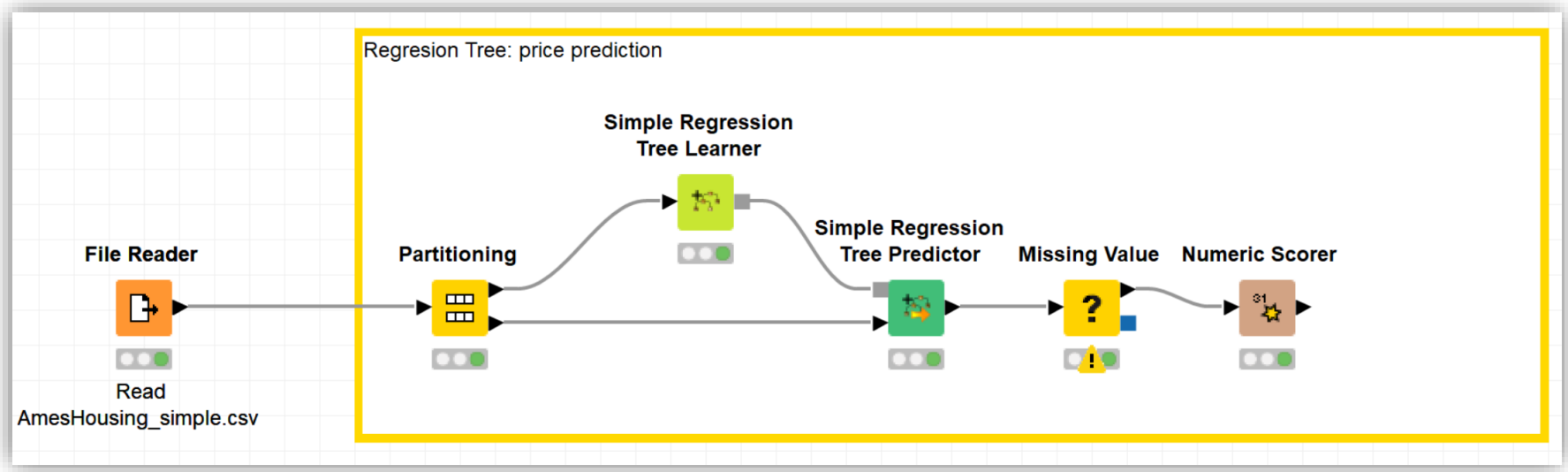
2. Regression Tree

Predict the price of an house in Ames (Iowa, USA) given a number of features (size, neighborhood, heating...) using a Regression Tree.

1. Read the dataset *AmesHousing_simple.csv.* It contains information about houses sold in Ames (only numerical values) as well as the *SalePrice.*
2. Add Partitioning node to File Reader output
   - Top port should have 70 % of the rows
   - Draw randomly such rows
3. Add Simple Regression Tree Learner to top output port of Partitioning node
   - Select price column to be learned
   - Execute the node and open its decision tree view. Which column is used in the beginning of the tree?
4. Add Simple Regression Tree Predictor
   - Predict test set (remaining 30% rows) by simply connecting the remaining unconnected output ports
5. Remove rows with missing values
6. Add Numeric Scorer to Regression Predictor Output
   - Reference Column: the column you learned
   - Predicted Column: the new column created by the predictor node

2. Regression Tree

Predict the price of an house in Ames (Iowa, USA) given a number of features (size, neighborhood, heating...) using a Regression Tree.



Thank you