Modelado com Árvores de Decisão.

Geiser Chalco Challco[1] Lucas Santos de Oliveira[1]

[1]Departamento de Ciência da Computação.

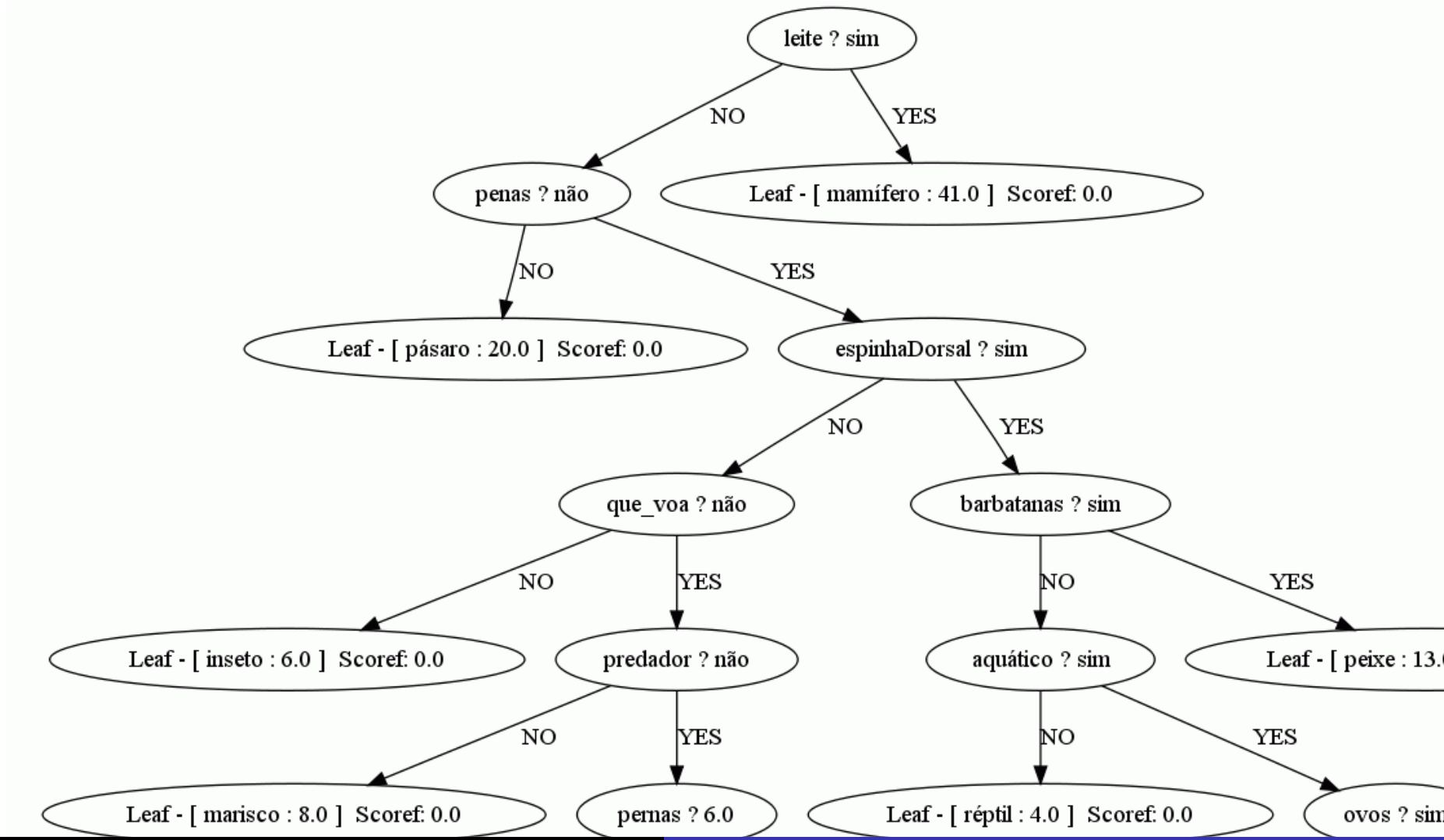
Intituto de Matemática e Estatística.
Universidade de São Paulo.

Tabela de Conteúdo.

Árvore de Decisão.

É um classificador automático que apresenta um dos métodos mais simples de máquina de aprendizado. A entrada é um situação descrita por um conjunto de propiedades e a saida é uma decisão. Após o treinamento, a árvore parece com uma série de comandos *if-else*.

Árvore de Decisão.



Treinando a Árvore.

Impureza de Gini

É a taxa de erro esperada, se um dos resultados num grupo é aplicado aleatoriamente a um dos itens no grupo.

$I_G(i) = 1 - \sum p(i,j)^2 = \sum p(i,j)\,p(i,k)\ \forall 1 \leq j \leq m \text{ e } j \neq k$

Entropia

É a quantidade de desordem num conjunto determinado pela frequência de cada item.

$H(x) = \sum f(x_i)\, log_2\left(\frac{1}{f(x_i)}\right) = -\sum f(x_i).log_2 f(x_i)$

Treinando a Árvore.

Variância

É uma medida de como uma lista de números varia com relação a média. Uma variância pequena significa que os números estão aproximados.

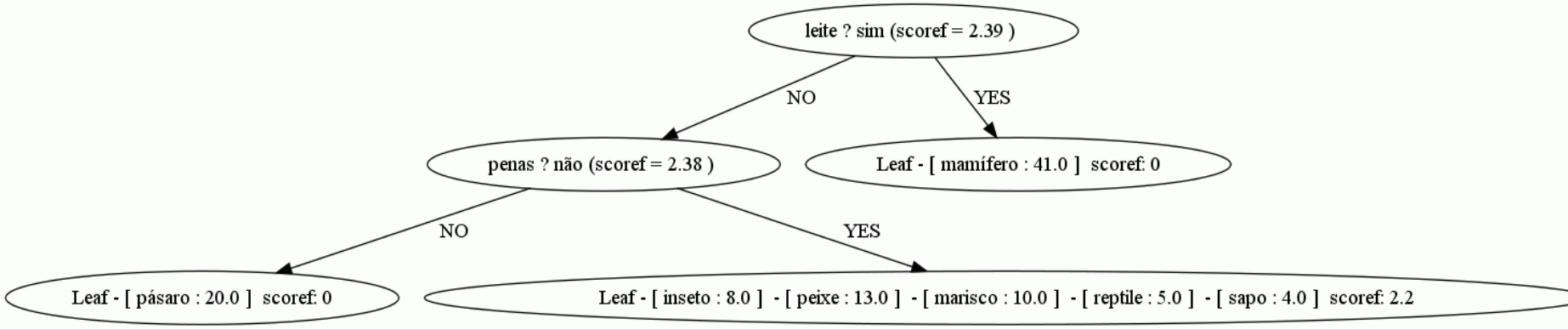
$$\sigma^2 = \frac{1}{N} \sum_{1 \leq i \leq N} (x_i - \bar{x})^2$$

Treinando a Árvore.

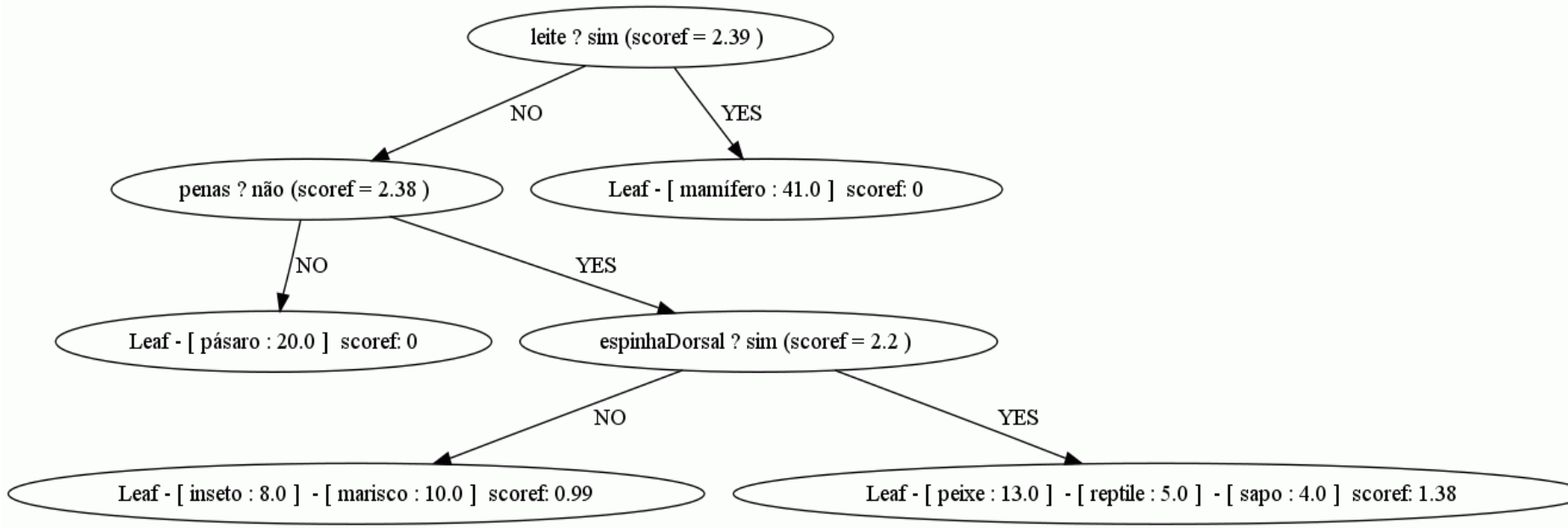
Tree build (*itemSet*)

if $size(itemSet) = 0$ **then**
  return $newTree()$
else
  $cScore \leftarrow scoref(itemSet)$
  for all *attribute* in $getAttributeSet(itemSet)$ **do**
    for all *value* in $getValues(itemSet, attribute)$ **do**
      $splitSet[V, F] \leftarrow split(itemSet, attribute, value)$
      $p \leftarrow size(splitSet[V])/size(itemSet)$
      $gain \leftarrow cScore - (p * scoref(splitSet[V])) - ((1-p) * scoref(splitSet[F]))$
      if $(gain > bestGain)(size(splitSet[V]) > 0)(splitSet[0].size() > 0)$ **then**
        $bestGain, bestAttribute, bestValue \leftarrow gain, attribute, value$
        $bestSets[V], bestSets[F] \leftarrow splitItem[V], splitSet[F]$
      end if
    end for
    if $bestGain > 0$ **then**
      $falseBranch = build(bestSets[F])$
      $trueBranch = build(bestSets[V])$
      return $newTree(bestAttribute, bestValue, falseBranch, trueBranch)$
    else
      return $newTree(itemSet)$

Treinando a Árvore.



Treinando a Árvore.



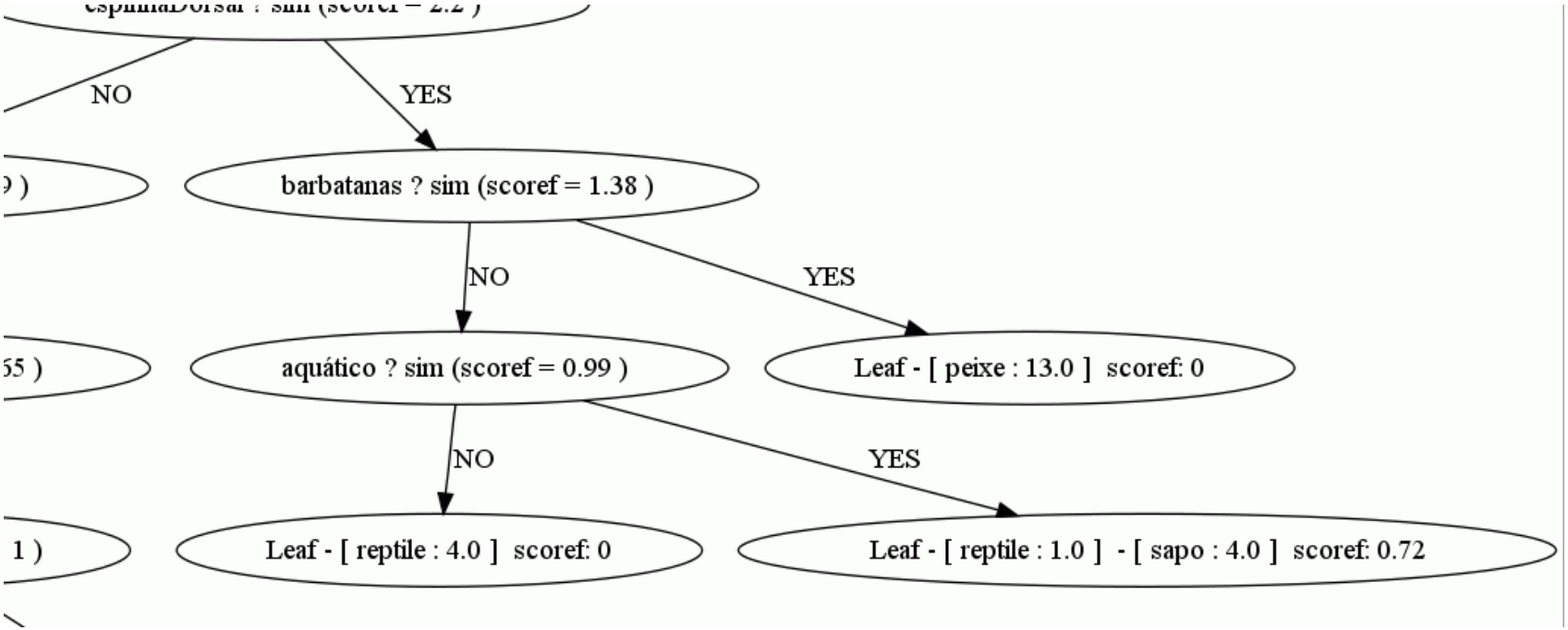
Poda da árvore

Evita que a árvore seja superaprimorada. A poda elimina nós supérfluos permitindo maior entropia menor do que um limite especifico.

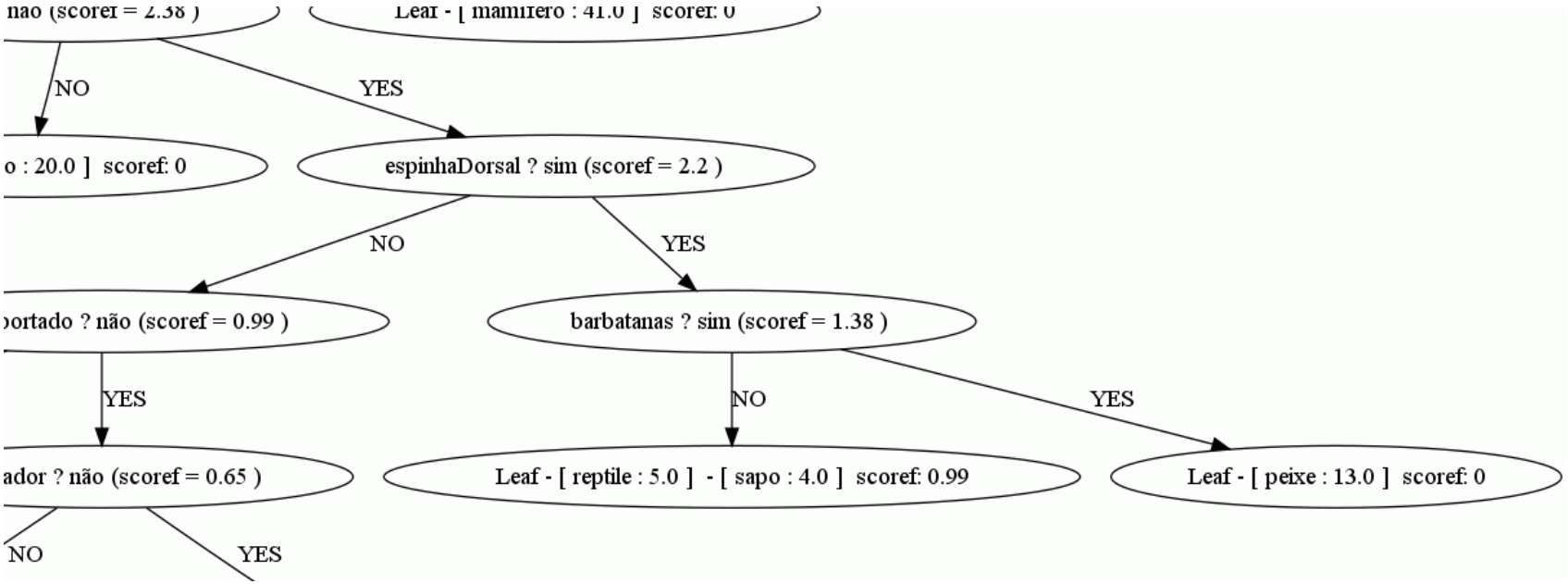
prune $(tree, minGain)$

if $tree.trueBranch.resultSet = null$ **then**
  $prune(tree.trueBranch, minGain)$
end if
if $tree.falseBranch.resultSet = null$ **then**
  $prune(tree.falseBranch, minGain)$
end if
$itemTSet, itemFSet \leftarrow tree.trueBranch.resultSet, tree.falseBranch.resultSet$
if $(itemTSet! = null)(itemFSet! = null)$ **then**
  $itemCSet \leftarrow itemTSetUitemFSet$
  $delta \leftarrow scoref(itemCSet) - (scoref(itemTSet) + scoref(itemFSet)/2)$
  if $delta < minGain$ **then**
    $tree.trueBranch, tree.falseBranch \leftarrow null, null$
    $tree.resultSet \leftarrow itemCSet$
  end if
end if

Poda da árvore



Poda da árvore



Classificando Animais

Modelos de preço de casas.

Conclusões

- A árvore de decisão é fácil de interpretar e utilizar.
- Serve como classificador e modelo de predições.
- Trabalha melhor com dados categóricos e números discretos.
- Muito aplicado no mundo real em problemas de tomada de decisão.
- Apresenta problemas com conjunto de dados objetivos de muitas possibilidades.
- Não é util para dados financieros e análise de imagens.

Refêrencias

 Toby Segaran. Programming Collective Intelligence. O'Really, 2007.

 http://www.zillow.com/howto/api/APIOverview.htm