

Inteligência Artificial  
**Segundo Trabalho – Aprendizado de Máquina**  
Profa. Heloisa 2016/2

Este trabalho consta da execução dos algoritmos de aprendizado supervisionado (árvore de decisão e árvore de regressão) estudados na disciplina, com o uso de funções disponíveis em pacotes de R e descritas neste documento.

Os conjuntos de dados podem ser selecionados entre os sugeridos no repositório UCI (<http://archive.ics.uci.edu/ml/>) ou no site da ferramenta Weka (<http://www.cs.waikato.ac.nz/ml/weka/>). Conjuntos sugeridos: Car evaluation, Iris, Forest Fires, Credit Approval, Mushroom, Soybean, Auto\_mpg.

Parte 1: Árvore de decisão para classificação

- Selecionar conjunto de dados adequado para problemas de classificação (classe nominal). Se o arquivo for muito grande é possível selecionar parte dos exemplos, mantendo no mínimo 80 exemplos e mantendo a mesma distribuição de exemplos de cada classe;
- Gerar arquivo .csv para leitura com `read.table()`;
- Dividir o conjunto de dados em treinamento (75% dos exemplos) e teste (25% dos exemplos) aleatoriamente, mantendo a mesma proporção de exemplos de cada classe;
- Fazer a leitura dos arquivos de treinamento e teste usando a função `read.table()`; (Alternativamente pode-se ler o arquivo todo e separar em treinamento e teste depois da leitura);
- Aplicar o algoritmo no conjunto de treinamento usando a função `rpart()` do pacote **rpart**;
- Construir a árvore gerada (gráfico) usando a função `rpart.plot()` do pacote **rpart.plot**;
- Executar o modelo nos dados de teste usando a função `predict()` do pacote **rpart**;
- Calcular erro (taxa de erros) e precisão do modelo gerado usando os resultados obtidos com os dados de teste.

Parte 2: Árvore de regressão

- Selecionar conjunto de dados adequado para problemas de regressão (classe contínua). Se o arquivo for muito grande é possível selecionar parte dos exemplos, mantendo no mínimo 80 exemplos e mantendo a mesma distribuição de exemplos de cada classe;
- Gerar arquivo .csv para leitura com `read.table()`;
- Dividir o conjunto de dados em treinamento (75% dos exemplos) e teste (25% dos exemplos) aleatoriamente, mantendo a mesma proporção de exemplos de cada classe;
- Fazer a leitura dos arquivos de treinamento e teste usando a função `read.table()`; (Alternativamente pode-se ler o arquivo todo e separar em treinamento e teste depois da leitura);
- Aplicar o algoritmo no conjunto de treinamento usando a função `rpart()` do pacote **rpart**;
- Construir a árvore gerada (gráfico) usando a função `rpart.plot()` do pacote **rpart.plot**;
- Executar o modelo nos dados de teste usando a função `predict()` do pacote **rpart**;
- Calcular erro (erro médio quadrático) e precisão do modelo gerado usando os resultados obtidos com os dados de teste.

Descrição das funções:

1) Construção de árvore de decisão ou de regressão:

```
modelo_ad <- rpart(fórmula, dados, método, controle, parâmetros)
```

- fórmula – define qual é o atributo de classe e quais são os atributos descritivos dos dados (usar o formato do exemplo abaixo);
- dados – conjunto de dados armazenado como um **data.frame**;
- método – parâmetro que define se a árvore será usada como classificação (“class”) ou regressão (“anova”);
- controle – define parâmetros que controlam o crescimento da árvore (ver exemplo abaixo);
- parâmetros – variável na qual é definido o critério de seleção de atributos (usar “information”, ver exemplo abaixo).

## 2) Predição (classificação ou regressão) usando a árvore gerada

```
y_estimado <- predict(modelo_ad, dados_teste, tipo)
```

- modelo\_ad – árvore de decisão ou de regressão gerada;
- dados\_teste – conjunto de dados que contém os exemplos de teste (sem a classe);
- tipo – especifica como será apresentado o resultado da predição: um vetor para valores numéricos (“vector”), classe para valores categóricos (“class”).

## 3) Exemplo do uso de rpart():

```
> install.packages("rpart")
> library("rpart")

> dados <- read.table("../caminho/...", header=TRUE, sep=";")
(leitura de dados de um arquivo com linha de cabeçalho e separador ;)

> modelo_ad <- rpart(Classe ~ At1 + At2 + At3, data = dados,
                     method = "class",
                     control = rpart.control(minsplit = 1),
                     parms = list(split = "Information"))
(modelo gerado pelo algoritmo de indução de árvore. O conjunto de dados tem
atributos At1, At2, At3 e Classe)
```

## 4) Exemplo de uso de rpart.plot():

```
> install.packages("rpart.plot")
> library("rpart.plot")

> plot_ad <- rpart.plot(modelo_ad, type=3)
```

## Relatório:

O trabalho deve ser acompanhado de um relatório que documente claramente o procedimento adotado e como esse procedimento foi definido na implementação. As principais etapas que devem ser *descritas* são: seleção e preparação do conjunto de dados, aplicação do algoritmo de indução da árvore, aplicação do processo de predição, geração da representação gráfica da árvore, cálculo da precisão. O relatório deve incluir as representações das árvores geradas na forma de texto e na forma gráfica, e os resultados dos cálculos de erros de cada exemplo do conjunto de teste e da precisão total.

## Observações:

- O trabalho pode ser feito em duplas.
- Entregar (no moodle – tarefa de arquivo único): Relatório, com o conteúdo detalhado anteriormente e código específico definido para o problema.
- DATA DE ENTREGA: 16/02/2017