**Relatório do 2º Projeto**

**Grupo 5**

**Descrição do problema:**

O problema que nos é dado no contexto do projeto envolve árvores de decisão. Árvores de decisão são modelos estatísticos que utilizam conjuntos de treino para classificarem e preverem futuros dados. Neste projeto são nos dados conjuntos de treino para os quais temos de inferir uma árvore.

**Solução:**

A solução base para criar uma árvore de decisão consiste em descobrir para todos os atributos qual destes tem maior ganho de informação e selecionar esse atributo para ser raiz da árvore. Ao encontrar-se o melhor atributo, isto é, o atributo que nos permite classificar mais exemplos de uma só vez, retiramos esse atributo, descobrimos o novo conjunto de dados e corremos o algoritmo novamente para este novo conjunto, recursivamente. Quando chegamos a algum caso terminal, a árvore é construída a partir daí.

Para calcular o atributo que tem maior ganho de informação, procedemos da seguinte forma:

* Identificamos o número de exemplos que são positivos com classificação verdadeira (*posTrue*) e com classificação falsa (*posFalse*). O número total de positivos no atributo é a soma entre *posTrue* e *posFalse*.
* Fazemos exatamente o mesmo para os casos negativos, em que o número total de negativos no atributo é a soma entre *negTrue* e *negFalse*.
* Calculamos o ganho de informação através do cálculo da entropia (mede a quantidade de incerteza numa distribuição de probabilidade)

**Fórmula para calcular o ganho de informação:**

Onde: total = positivos + negativos;

negDivisao = ou 0 se negativos = 0;

posDivisao = ou 0 se positivos = 0;

O atributo escolhido é o atributo com maior valor no cálculo do ganho de informação.

**Problemas encontrados no algoritmo base:**

1. O primeiro problema que identificámos enquanto testávamos o algoritmo base implementado foi que quando existiam dois ou mais exemplos com a mesma descrição, mas com diferente classificação, isto é, os valores em D são iguais, mas em Y não, o algoritmo não conseguia encontrar uma árvore de decisão consistente com os dados.
2. O segundo problema que identificámos enquanto testávamos é que por vezes, a árvore apreendida não era a menor possível, isto é, era possível a árvore ter tamanho menores e ser correta à mesma.

O algoritmo base implementado não tinha em conta os problemas mencionados acima.

**Soluções para resolver os problemas detetados:**

1. A solução que iríamos implementar para o primeiro problema baseava-se em fazer com que cada folha indicasse uma classificação por maioria para o seu conjunto de exemplos e se, esta alteração não afetasse o resultado final da árvore, então a alteração era mantida.
2. A solução que iríamos implementar para o segundo problema baseava-se em aplicar a técnica da poda da árvore de decisão. Esta técnica baseia-se em reduzir o tamanho da árvore removendo secções da árvore que não são críticas e que são redundantes para classificar as instâncias.

**Análise crítica dos resultados:**

Avaliando os resultados do nosso código aos testes fornecidos pelos professores, verificamos que estes resultados são os esperados por nós. Aos testes onde não há ruído o algoritmo base implementado funciona corretamente e como foi calculado por nós previamente, não funciona para os restantes testes. Isto acontece, porque, como identificado anteriormente, há problemas neste algoritmo (ruído, árvores extensas). Assim, para conseguirmos passar a todos os testes fornecidos, tentámos implementar as soluções identificadas por nós.

**Se conseguirmos:** Ao implementar a solução descoberta para o problema (…), passámos aos testes relativamente a este problema, como era esperado.