Trabalho 03 - Melhorando o akinator

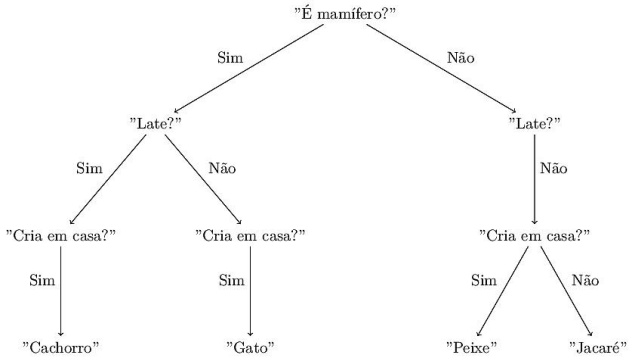
Data de entrega: 28/06/2024

Importante:

- **Não** olhe códigos de outros grupos ou da internet. Exceto os que são fornecidos ou vistos em aula.
- O trabalho pode ser feito em duplas.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do runcodes.hokama.com.br não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma runcodes.hokama.com.br (se estiver em dupla, apenas 1 deve submeter).
- Coloque o(s) número(s) de matricula na primeira linha do código.

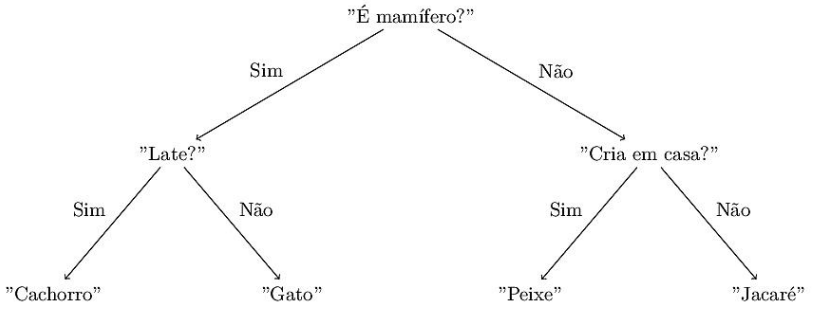
Em uma árvore binária de decisão, cada nó interno corresponde a uma pergunta, caso a resposta para essa pergunta seja SIM, vamos para o filho esquerdo, e caso seja NÃO vamos para o filho direito. Fazemos isso até chegar em uma folha, que corresponde a uma decisão. Podemos usar árvores de decisão para classificar ou identificar objetos (que chamaremos de elementos ou personagens), como é o caso do Akinator simplificado visto em sala.

No akinator simplificado, cada nível da árvore correspondia a uma mesma pergunta, e a ordem das perguntas seguia a ordem com que elas eram lidas da entrada. Isso pode gerar várias ramificações que são desnecessárias. Veja o Exemplo 1 abaixo.



Você pode notar que na subárvore esquerda da raiz, a pergunta “Cria em casa?” é inútil, uma vez que antes dela ser feita já seria possível identificar o animal. Já na subárvore direita a pergunta “Late?” é ínutil, pois não separa nenhum dos elementos. Neste trabalho você deverá implementar o seguinte heurística para encontrar árvores menores. (Uma heúristica é um algoritmo que não garante que irá encontrar a melhor solução possível)

Em um dado nó, seja X o conjunto dos elementos que chegaram naquele nó, seja p uma pergunta, seja $p_s(X)$ o número de elementos de X que respondem "sim" para p e seja $p_n(X)$ o número de elementos de X que respondem "não" para p. A pergunta que **deverá** ser feita naquele nó é a pergunta que minimiza $|p_s(X) - p_n(X)|$ (o valor absoluto da diferença entre os que respondem sim e os que respondem não) em caso de empate a pergunta que aparece antes é a escolhida. Para o mesmo conjunto de dados, a árvore obtida seria o Exemplo 2:



No nó raiz todos os 4 animais são considerados, das 3 perguntas possíveis:

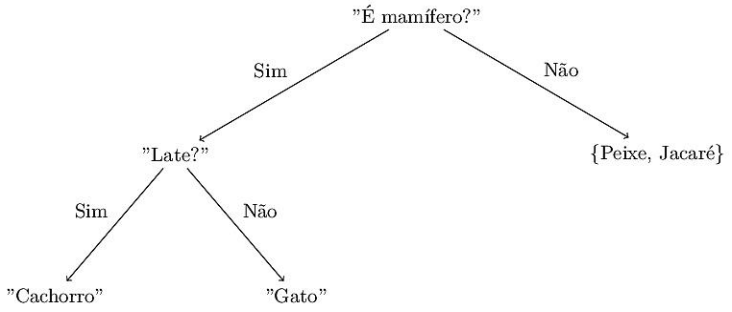
"É mamifero?": 2 respondem "sim" e 2 respondem "não", portanto a diferença é 0 (zero).
"Late?": 1 responde "sim" e 3 respondem "não", portanto a diferença é 2.
"Cria em casa?": 3 respondem "sim" e 1 responde "não", portanto a diferença é 2.

Portando a pergunta escolhida é "É mamifero?" pois essa minimiza a diferença. Dessa forma, Cachorro e Gato vão para a subárvore esquerda e Peixe e Jacaré vão para a subárvore direita.

Na subárvore esquerda então vamos considerar novamente as 3 questões, mas agora somente o conjunto {Cachorro, Gato} será considerado.

"É mamifero?": 2 respondem "sim" e 0 respondem "não", portanto a diferença é 2.
"Late?": 1 responde "sim" e 1 responde "não", portanto a diferença é 0 (zero).
"Cria em casa?": 2 respondem "sim" e 0 responde "não", portanto a diferença é 2.

Dessa forma a pergunta escolhida é "Late?". Quanto um nó tiver 1 só elemento, ele não precisa mais ser ramificado, como Cachorro e Gato agora estão sozinhos, não há novas ramificações.



O procedimento é análogo para a subárvore direita. Obtendo o resultado apresentado anteriormente.

Utilizaremos a altura média dos elementos como métrica para medir a qualidade de uma solução. A altura de um elemento é a distância até a raiz, que pode ser entendida como a quantidade de perguntas necessárias para chegar naquele elemento. No Exemplo 1, para qualquer elemento foi necessário responder 3 perguntas, portanto a média é $\frac{3+3+3+3}{4} = 3$ já no Exemplo 2, apenas duas perguntas foram necessárias para todos os elementos, e portanto a média é $\frac{2+2+2+2}{4} = 2$. Vale notar que essa média nem sempre é um número inteiro.

O seu programa deverá ler da entrada padrão o nome de um arquivo em formato csv (Comma-separated values), por exemplo:

```
animais.csv
```

Esse arquivo contêm a resposta de todas as perguntas para cada um dos elementos, e na primeira coluna da primeira linha esse arquivo tem 2 inteiros com o número de elementos e o número de perguntas (normalmente arquivos csv não tem essas informações, mas dependendo de como você leia seu arquivo essas informações podem ser úteis). No exemplo, o arquivo animais.csv é:

```
4 3,É mamífero?,Late?,Cria em casa?
Gato,1,0,1
Cachorro,1,1,1
Peixe,0,0,1
Jacaré,0,0,0
```

Você poderá usar o código akinator.py visto em aula como base para seu programa se assim desejar. Seu programa deverá imprimir apenas a altura média obtida pelo algoritmo, com 2 casas decimais. Nesse exemplo:

```
2.00
```

- Você deverá implementar em linguagem python.
- Seu programa deve executar no runcodes.hokama.com.br em menos de 1 segundo.
- Você não deve usar nenhuma função pronta muito complexa.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!
- Não deixe para os últimos dias!