Gabriel Baptista - nUSP: 8941300

Relatório EP2 - MAC 0425(Inteligência Artificial)

(a) Não consegui gerar dados o suficiente parar criar o pedido nesse item, explicarei no item (d)
(b) Os dados que obtive nos testes que consegui executar são que para dada profundidade máxima, uma métrica pode ser melhor que a outra, em especial, para a profundidade que estava como *default* no classify (200), a melhor classificação encontrada foi a de mais frequentes, a qual após os treinamentos e testes acabava acertando em cerca de 59% dos casos.
(c)

Questão 1: Métricas

A escolha de uma boa métrica é muito importante, pois ao lidarmos com aprendizado de uma árvore de decisão, podemos enfrentar alguns problemas, como, a não completude dos dados fornecidos(o livro chama de *missing data*), atributos com muitos valores associados, o input e output de atributos de valores inteiros e contínuos e o input de atributos de valores inteiros. Uma boa métrica deve ser capaz de manusear esses problemas de forma a satisfazer o proposto pelo sistema.

Questão 2 - Condição de parada

A condição de parada no caso é verificar se é folha ou não, então, além do nível máximo que uma árvore pode ter, considerei como condição de parada o que chamei de frequência(dict do parâmetro labels), onde se o tamanho dele for 1 ou se o mesmo estiver vazio deve-se parar, outra condição é que também deve-se verificar se todos os dados passados (parâmetro *data*) são iguais, se forem, deve-se parar. Todos esses casos resultam em serem folhas da árvore de decisão.

Questão 3 - Acurácia

Ao aumentarmos o nível máximo de profundidade, acabamos criando várias vezes um nó já presente na árvorey, além disso, aumentamos a quantidade de vezes que chamamos recursivamente a função de construção da árvore, o que podia dar uma aparente melhora na acurácia acaba tornando inviável para rodar o programa… Quando aumentei o número do maxDepth aos poucos ia melhorando a acurácia de fato, mas em um instante o programa me retornou a seguinte mensagem: “maximum recursion depth exceeded while calling a Python object”.

Questão 4 - Comparação

Esta questão pode ser respondida em parte pelo começo da anterior, se formos colocar uma instância muito repetida na árvore, acabariamos criando várias vezes nós que já aparecem na árvore, o que causa uma complexidade um tanto quanto desnecessária, então ao rotularmos as instâncias mais frequentes acabaria com esse problema, tornando-se mais eficiente que a árvore puramente em si.

(d) Por conta do código de classify(muito difícil de ser alterado), tive dificuldades para testar as coisas sem o autograder, por isso acabei não conseguindo coletar dados suficientes para criar uma tabela/gráficos como pedido, apenas consegui executar parte das classificações/métricas