Aluno: João Vitor de Camargo
Matrícula: 00274722

Relatório de Atividade Prática: Algoritmo DT (Decision Tree)

Código e demais arquivos relevantes para a implementação podem ser encontrados em: https://github.com/camargodev/ml-algorithms/tree/main/decision-tree

Durante esse trabalho prático, utilizei a biblioteca sugerida **skicit-learn** (sklearn), disponível em Python, para treinar as árvores de decisão. Para *plottar* os resultados, usei a biblioteca **graphviz**.

Comparação: Entropy x Gini

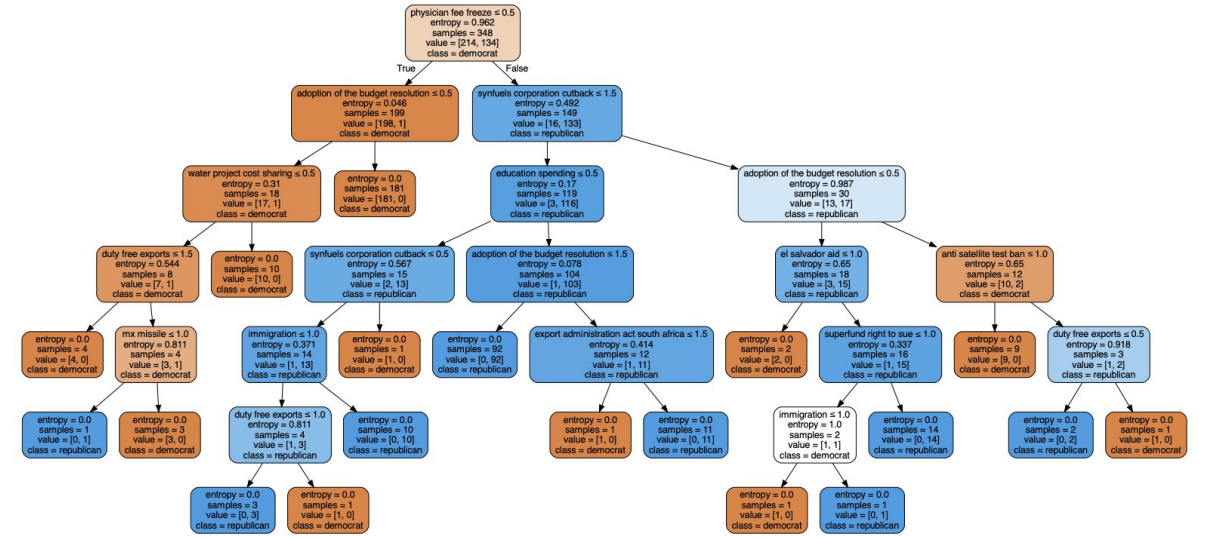
Nessa primeira etapa, a árvore de decisão foi treinada separadamente com os critérios de Entropia e Gini para que seus resultados fossem comparados. Como solicitado, usei a estratégia de *holdout* para separar as instâncias em 80% de treino e 20% de teste.

Na fase de separação de instâncias do *holdout*, usei a função *train_test_split* do sklearn. Devido a randomicidade imposta por essa função, eu repeti o treino 100 vezes para cada critério. Abaixo os resultados:

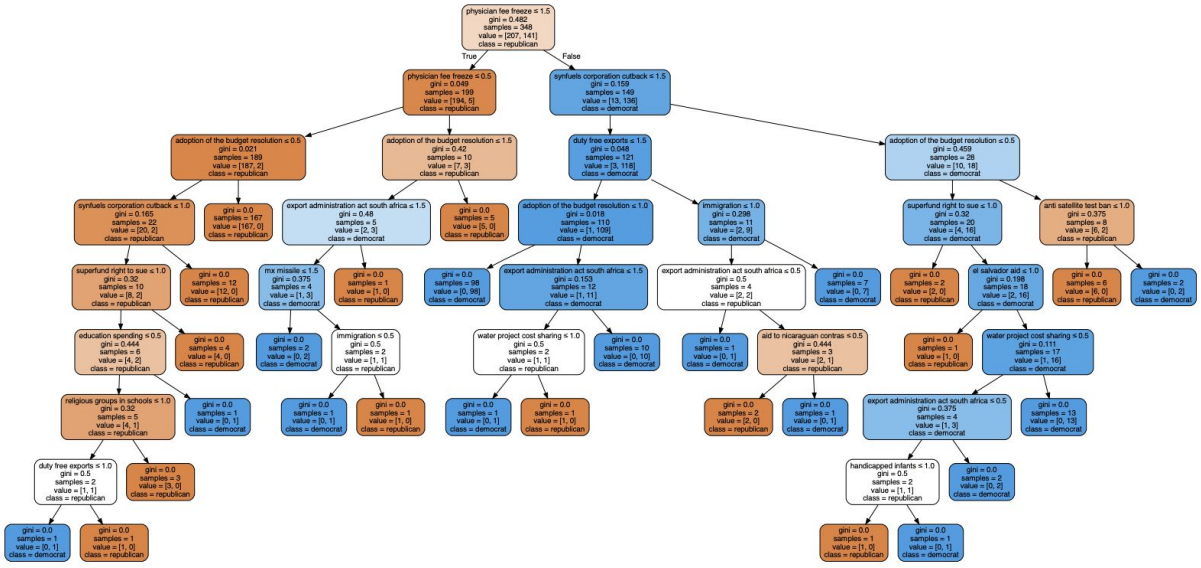
	Acurácia Mínima	Acurácia Média	Acurácia Máxima
Entropia	0.8735632183908046	0.9408045977011493	0.9770114942528736
Gini	0.8275862068965517	0.9324137931034484	0.9885057471264368

Como é possível observar, o valor de acurácia média dos dois critérios é muito próximo, tendo menos de 1% de diferença absoluta. A árvore construída com Gini possui ambas a mínima e a máxima acurácia obtida dentre todos os treinos, se mostrando menos constante. Enquanto a margem entre mínimo e máximo com Entropia foi de cerca de 10%, a com Gini foi 16%.

Essa foi a árvore obtida com o critério de Entropia:



E essa foi a árvore obtida com o critério de Gini:



É possível observar que a árvore com Gini é consideravelmente maior e possui decisões mais específicas. O menor caminho de raiz a folha com Entropia foi de 2 pulos, enquanto o maior foi de de pulos. Já com Gini, o menor caminho tem 3 pulos e o maior 8. A maior especificidade de Gini é o que gera resultados tão variáveis. Por um lado, ela não erra tanto por generalizar demais, mas por outro, erra por fazer possíveis *overfittings*.

Ambas as árvores começam com o atributo *physician fee freeze*, que se mostra ser o mais relevante para a classificação. Em seguida, ambas usam *synfuels corporation cutback*. Os atributos usados são *handicapped infants*, *religious groups in school*, *aid to nicaraguan contras* foram usados pelo critério Gini. O atributo *crime* não foi usado por nenhum. Todos os outros foram usados pelas duas árvores.

Para visualizar as árvores em melhor resolução, clique aqui: Gini e Entropia.

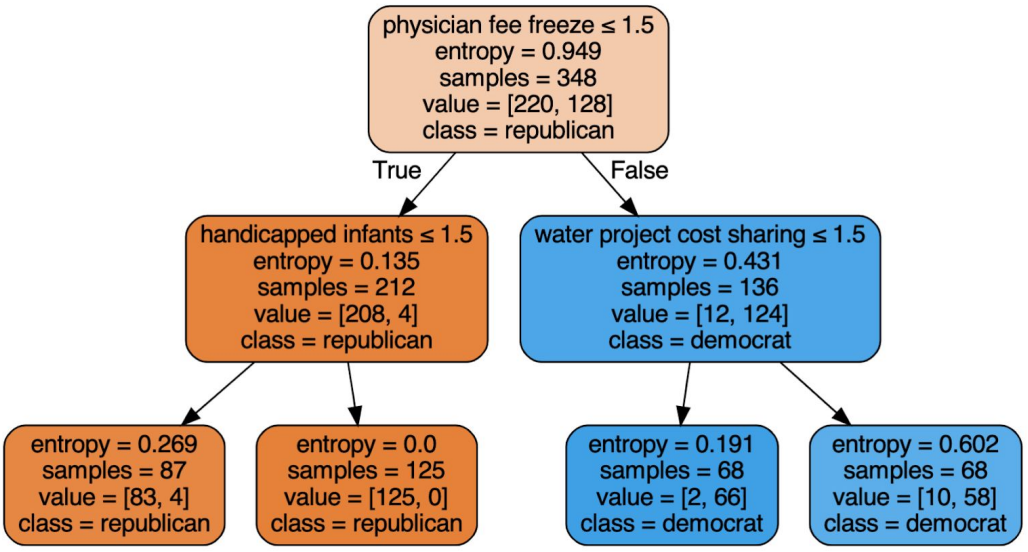
Critério de Entropia com diferentes parâmetros

Nessa atividade, comparamos o resultado obtido ao variar a profundidade máxima da árvore e o número mínimo de *samples* por folhas. O algoritmo escolhido foi o de Entropia, visto que possuiu valores mais estáveis. O mesmo processo de calcular a acurácia média com 100 execuções foi empregado.

A árvore com Entropia mostrada anteriormente possui profundidade máximo 7 (contando a raiz). Já o número de *samples* varia de 1 a 181, mas se mantendo geralmente menor que 10. Portanto, ao variar, foram escolhidos valores que fiquem em torno desses. Para a profundidade máxima, os valores foram [5, 7, 10] e para o número mínimo de *sample* por folhas [1, 5, 25, 50].

A acurácia média se manteve entre 93% e 95%, condizente com os 94% da Entropia no exercício anterior. As árvores geradas variaram como esperado: quanto maior o número de *samples* por folha, menor a árvore. Todavia, a acurácia se manteve, o que levanta a possibilidade de que a árvore observada no exercício 1 já sofria de *overfitting*.

Abaixo a árvore com profundidade máxima 5 (menor valor) e número de *samples* por folha igual a 50 (maior valor). Essa é a árvore que mais foi limitada de crescer. Todavia, mesmo utilizando apenas 3 atributos, ela atinge 95.2% de acurácia:



Já no caso contrário, a árvore de profundidade máximo 10 (maior valor) e número de *samples* por folha 1 (menor valor) foi a instância a qual mais foi permitido crescer. Seu desenho pode ser visto aqui. Essa árvore possui 21 nodos folha e acurácia de 93%. O fato da acurácia cair ao aumentarmos a complexidade da árvore reforça a suspeita de overfitting. Na pequena árvore ilustrada na imagem acima também é possível ver dois lados bem claros (republicanos para um lado, democratas para outro). Na árvore maior disponível no link, existe uma mistura maior, com *outliers* presentes em partes da árvore onde a maioria é do outro grupo. Isso levanta a suspeita de que o *overfitting* foi causado pela existência de ruído.

Por fim, é interessante comentar que em todas as 12 combinações, o atributo *physician fee freeze* se mostrou presente no nodo raiz. Isso reforça a hipótese de que esse atributo é o mais relevante para a classificação.

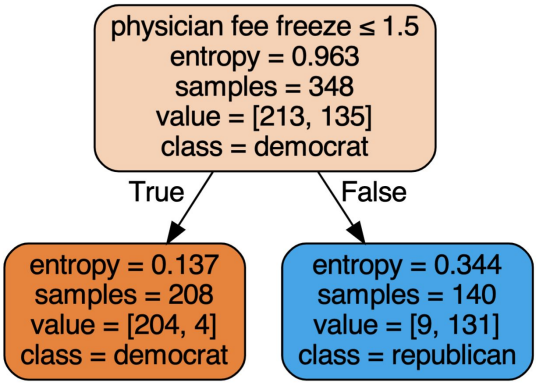
Critério de Entropia com e sem poda

Para esse exercício voltei para o status do exercício, ou seja, sem alterar a profundidade máxima e não número mínimo de *samples* por folha. No sklean, para realizar a poda, é necessário alterar o valor do parâmetro *ccp_alpha* que define o quão severa a poda será. Quanto maior o valor do parâmetro, mais severa a poda. Por padrão o valor é 0, ou seja, a biblioteca não realiza a poda. Com base em exemplos, foram definidos os seguintes valores para o parâmetro: [0 (sem poda), 0.001, 0.01, 0.1, 1.0].

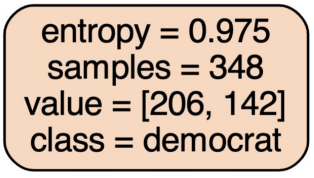
Abaixo os resultados de acurácia com cada um dos valores

Valor de ccp_alpha	**0**	**0.001**	**0.01**	**0.1**	**1.0**
Acurácia	93.64%	93.75%	94.27%	95.24%	61.64%

Como é possível observar, a aplicação de um certo nível de poda melhorou os resultados. Como esperado, quanto maior o coeficiente de poda, menor árvore gerada. O melhor resultado foi obtido com coeficiente 0.1 e pode ser observado abaixo:



O fato de essa árvore ter atingido 95% de acurácia confirma dois pontos previamente levantados: a complexidade desnecessária das árvores geradas inicialmente com Entropia ou Gini e a extrema relevância do atributo *physician fee freeze*. Por outro lado, o pior desempenho foi obtido com coeficiente de poda 1.0 observado abaixo:



Com só um nó, essa árvore classifica como democrata todas as instâncias que receber, independente de atributos. Em conclusão: para esse *dataset* de treino e teste, o uso da poda agrega pouco valor, visto que sem ela, uma acurácia média de cerca de 94% já era possível. Todavia, o excesso da poda claramente deteriora os resultados. Ao meu ver, para esse *dataset*, o custo de calibrar o valor ideal do coeficiente de poda talvez não valha a pena quando analisada a pouca melhora de acurácia que o uso da poda oferece.