Relatório de Atividade Prática

Árvores de Decisão

IA1s2024

Aluno
Raissa Barbosa - RA 148551

São José dos Campos
2024

1 Estudo comparativos de Parâmetros das Árvores de Decisão

Para iniciar as comparações, segue a árvore final para todos os parâmetros com seus valores default, com acurácia e acurácia balanceada iguais a 0.79 e 0.77, respectivamente.

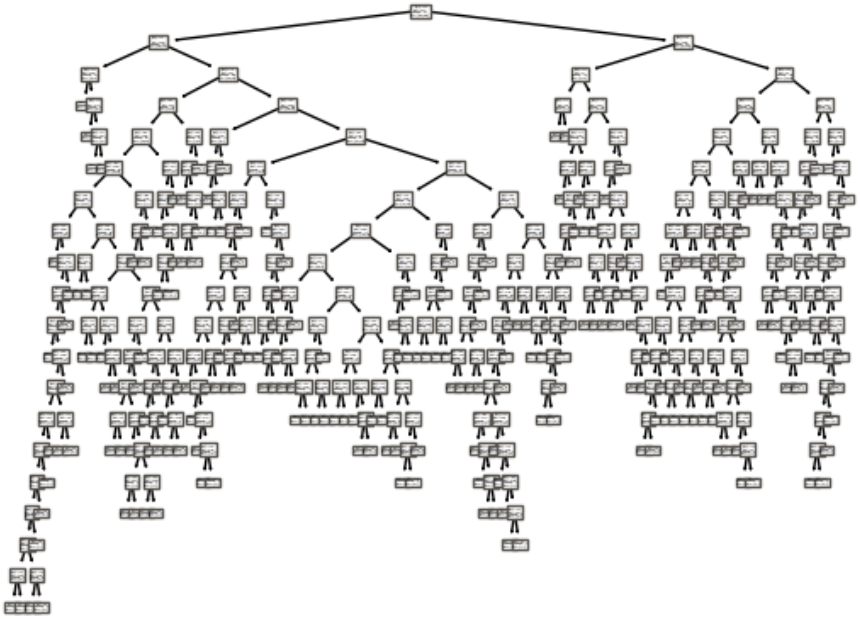

Figura 1: Árvore com parâmetros default

1.1 Criterion

Sabendo que o criterion mede a qualidade das divisões dos nós na construção da árvore, que o "Gini" leva em consideração as classificações erradas numa escolha aleatória, e que o "Entropy" olha para o quão desordenada a distribuição das classes em um nó está, segue as árvores obtidas.

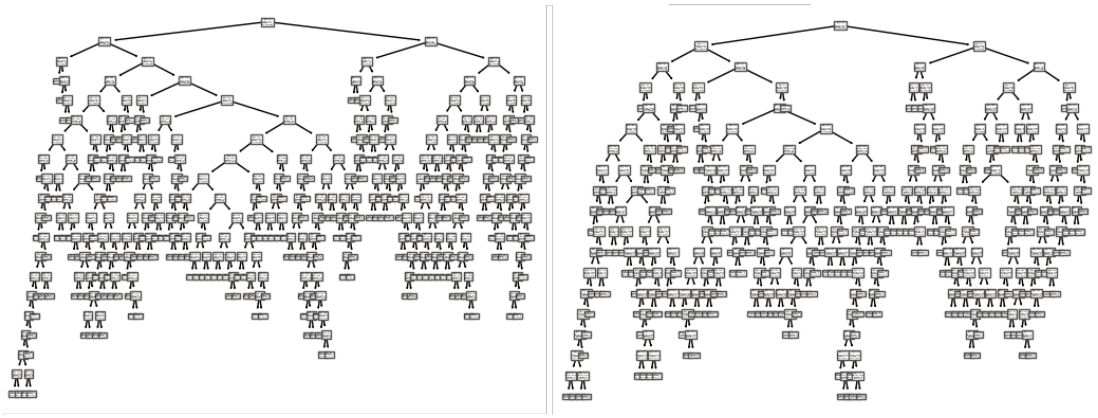

Figura 2: Gini vs Entropy

Observando as árvores acima, pode-se perceber que a árvore com o critério de Entropy ficou com seus nós folha mais fundos, fazendo com que os nós passassem por mais decisões.

Como acurácia, os valores foram: Gini = 0.77 e Entropy = 0.78. Teve uma pequena melhora, portanto, apesar da árvore ter sido mais funda, a qualidade das divisões dos nós baseado na aleatoriedade foi melhor nesse caso.

1.2 Splitter

Esse parâmetro determina a estratégia da divisão dos nós. No modo Best, a melhor escolha de divisão é levada adiante, já no modo Random, a escolha é feita de forma mais aleatória.

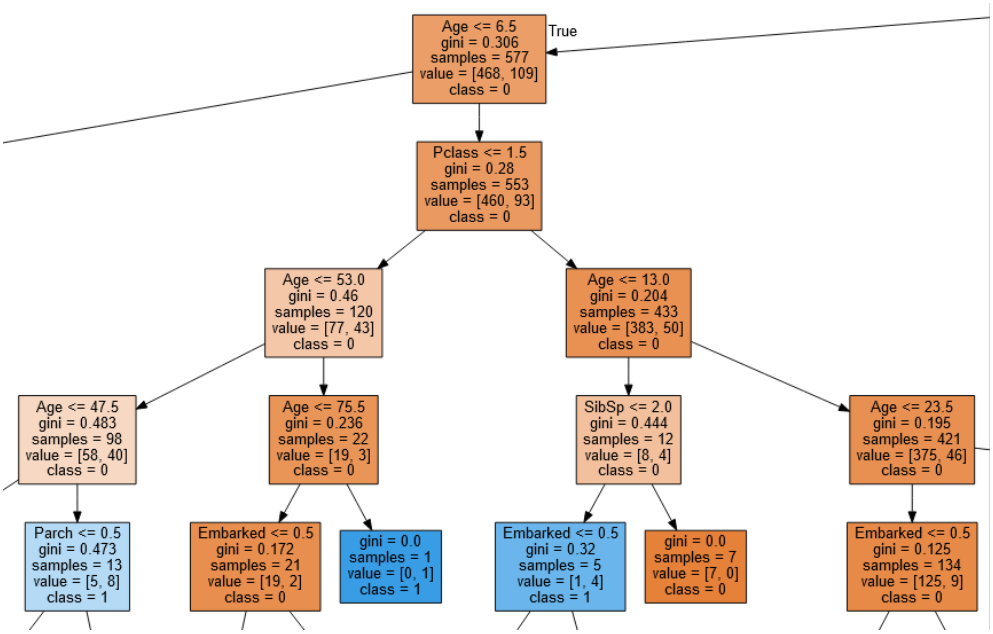

Figura 3: Parte de uma árvore com o Splitter Best

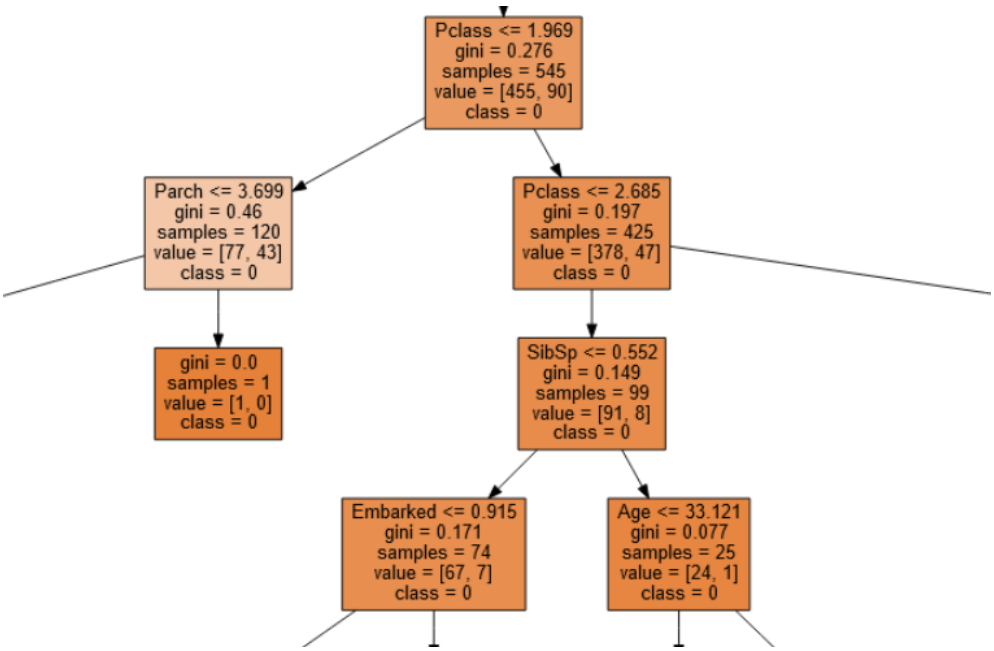

Figura 4: Parte de uma árvore com o Splitter Random

Olhando as árvores mais de perto, é possível observar que na árvore 3, que foi dividida por Best, um critério aparece mais de uma vez para fazer as decisões,

que nesse caso é 'Age', e essa repetição acontece com mais frequência entre nó pai-nó filho, então ela deixa claro algumas características bem importantes nas divisões para chegar na classificação final. Já na 4, que teve seus nós divididos de maneira mais aleatória, não aconteceu tanta repetição seguida, houve uma maior variedade nos parâmetros analisados.

Como resultado, a acurácia balanceada para Random ficou em 0.78, contra 0.77 para Best, logo, os nós foram divididos de maneira bem diferente mas isso não afetou muito no resultado final de classificação.

1.3 Max_depth

A profundidade máxima da árvore é um critério muito importante durante a criação de uma árvore de decisão, pois caso não haja um controle dele, algo chamado de overfitting pode acontecer, que é quando uma árvore fica muito comprida, e isso acaba a deixando muito eficiente, mas somente a uma situação específica, o que não é o esperado, pois o treinamento, por exemplo, pode ser bem diferente do teste propriamente dito.

A árvore resultante sem uma quantia máxima de profundidade já é conhecida, então árvores com máximos de 2 e 4 níveis serão analisadas.

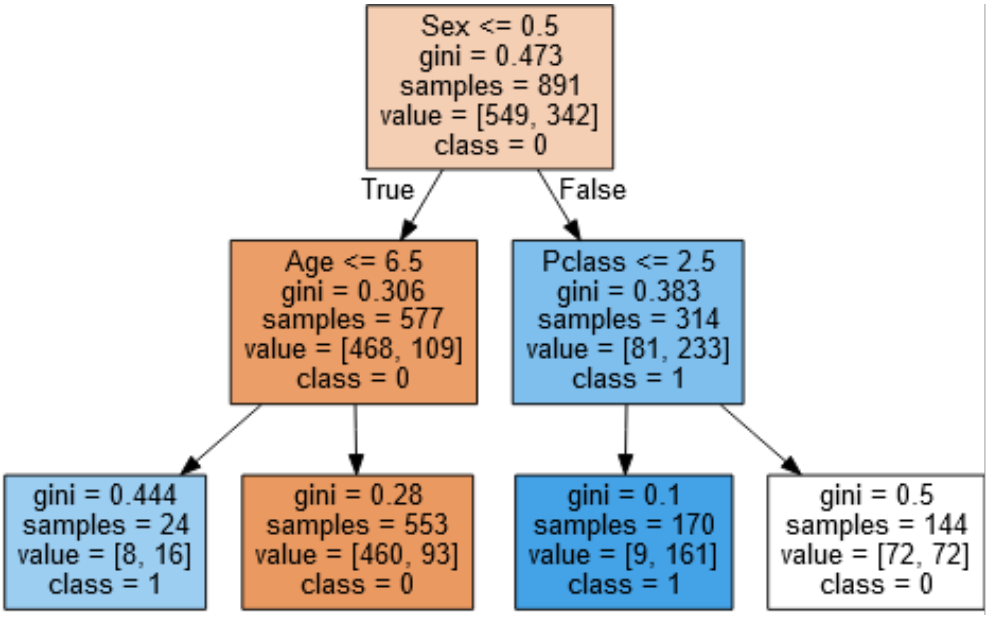

Figura 5: Profundidade 2

Para o máximo de 2, como pode ser visto na Figura 5, somente 3 classes de tabela com 7 foram utilizados, o que não resultou em um bom desempenho, uma vez que esses 3 critérios não foram suficientes para gerar uma boa classificaçao quanto aos sobreviventes do Titanic. Por outro lado, quando a profundidade aumentou para 4, todas as classes foram utilizadas na separação dos nós, o que trouxe uma acurácia de 0.84, em comparação com 0.74 da profundidade 2. Então, para esse caso, quando um número maior de classes conseguiu ser utilizado para dividir os nós da árvore, o resultado ficou mais preciso, uma vez que somente 7 nós é um número muito baixo para fazer a classificação desejada quando se tem uma quantidade grande de dados e cujos atributos existentes não gerem a separação ótima. Abaixo, é possível observar a diferença da acurácia balanceada nos três casos, profundidade None, 2 e 4.

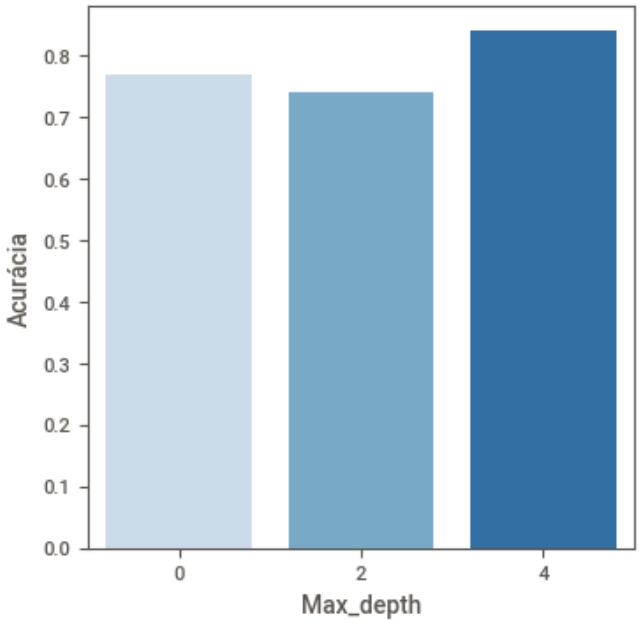

Figura 6: Profundidades

1.4 Min_sample_split

Por fim, tem-se o parâmetro que determina o mínimo de amostras necessárias para que a divisão de um nó seja feita. Através dos testes, ao trocar o default 2 pelo 10, a acurácia teve uma boa evolução, chegando a 81% de acurácia. Portanto, para o caso em estudo, para que um nó pudesse se dividir, pelo menos 10 amostras devem ser consideradas, caso contrário, o resultado fica pior. Ao olhar a árvore abaixo, nota-se que ela ficou mais compacta em relação a da Figura 1, ela evitou o overfitting e gerou um resultado melhor.

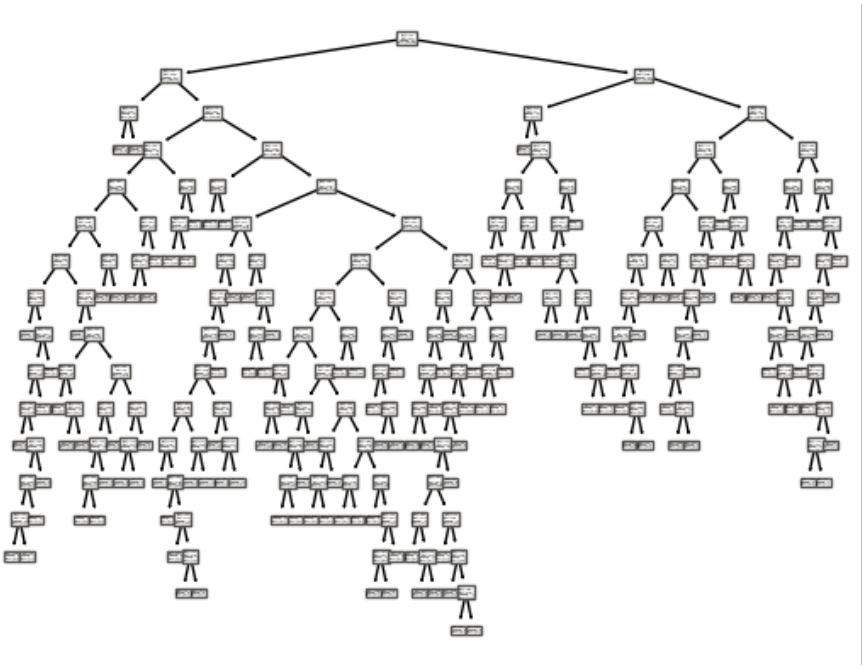

Figura 7: Árvore com min sample split 10

2 Etapas de Mineração de Dados

Para iniciar a criação das árvores da competição, o código dado pelo professor serviu como base, logo todo o desenvolvimento levou esse código inicial como referência.

2.1 Conhecimento do Domínio

Antes de começar a mexer nos hiperparâmetros, foram criados 4 gráficos, como mostra a Figura 8, para que a visualização de alguns dados ficasse mais clara, então, a partir disso, os dados puderam ser entendidos de forma mais rápida e generalizada.

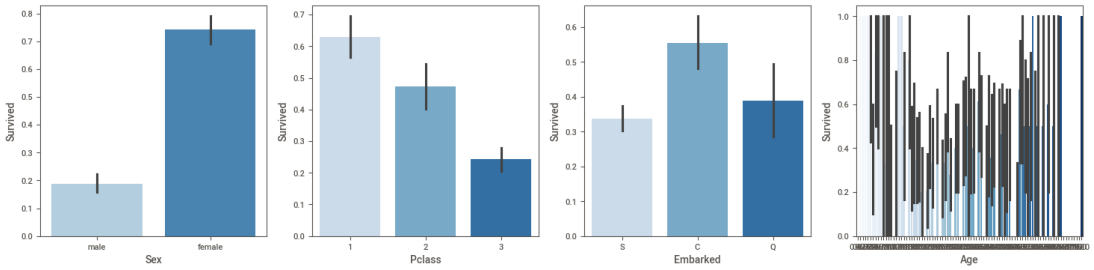

Figura 8: Gráfico inicial de algumas relações com a coluna 'Survived'

Após isso, os métodos .head() e .describe() foram chamados para que os dados fossem visualizados de maneira quantitativa e para que as necessidades de melhoria nos dados fossem descobertas. Com isso, foi visto que as colunas 'Age', 'Embarked' e 'Fare' necessitavam de tratamento para prosseguir com a análise.

E com uma rápida observação no quarto gráfico da imagem acima, foi perceptível que havia bastante idosos e crianças com idades parecidas, e que o resto das pessoas se distribuía entre e infância e velhice. Baseado nisso, na tentativa de fazer com que as previsões das idades fossem mais acertivas, foi decidido que as idades seriam restringidas a somente alguns valores.

2.2 Pré-processamento

Uma vez que ja era conhecido que as colunas 'Age', 'Embarked' e 'Fare' estavam com dados faltantes, o primeiro trabalho foi encima delas. Mas além de completar essas colunas, algumas outras coisa foram feitas.

Como mencionado anteriormente, parte do código foi providenciado pelo professor, logo grande parte desse tratamento para completar os dados faltantes já havia sido feito.

Para a coluna 'Age', foi utilizado um método que envolve a média e o desvio padrão para completar as idade. E como as idades precisavam ser generalizadas, uma lógica para separar todas as idades em somente 3 quantias foi feita. No código, há 3 possíveis combinações de idades, podendo ser [10, 25, 40], [10, 40, 70] ou [60, 85, 95], dependendo da distrubuição das idades no banco de dados em análise.

Para a coluna 'Embarked', foi feita uma consulta quanto aos dados faltantes e em quais linhas eles se encontravam, e como só estava faltando duas informações, foi visto qual resposta era a que mais aparecia no banco de dados, então os 2

dados faltantes foram completados com esse dado mais recorrente, que no caso era "S".

E para a coluna 'Fare', algo parecido com lógica anterior foi feita, mas nesse caso o dado mais recorrente foi "0".

Para todas as colunas que tinham resultado numérico em float, e para todas as colunas que tinham resultado em string, esses dados foram transformados em inteiro, porém para os dados float houve um arredondamento, e para os dados em string, os dados viraram 0, 1 ou 2.

Apesar dos comentários positivos acerca desse tratamento de dados, antes de se obter esse resultado, uma tentativa que relacionava a coluna 'Fare' com a 'Sex' foi explorada (que se encontra comentado no código), porém não obteve sucesso.

2.3 Extração de Padrões

Aqui, foi utilizado um algoritmo para classificação, que é a árvore de decisão. Com as mudanças feitas nos dados, foi perceptível uma melhora na acurácia, portanto o algoritmo do jeito que estava escrito permaneceu o mesmo.

Com a observação da árvore construída, foi visto que alguns parâmetros se repetiam bastante nos nós da árvore, e apesar de 'Age' não ser muito visto na árvore, como na moldagem dos dados ele resultou em uma melhora considerável, foi mantido. Portanto nenhuma outra coluna foi removida, além das iniciais removidas pelo professor.

E como já havia tido uma melhora no resultado de acurácia com esse algoritmo, os hiperparâmetros foram variados para encontrar a melhor combinação, chegando ao resultado de que os melhores seriam criterion="entropy", randomstate=0, splitter="best", maxdepth=8, minsamplessplit=10.

2.4 Pós-Processamento

Após todas as mudanças no código e se obter o melhor resultado para acurácia, com acurácia balanceada batendo mais de 0.90, alguns padrões foram observados. Como resultado final, foi obtida uma árvore balanceada, como mostra 9.

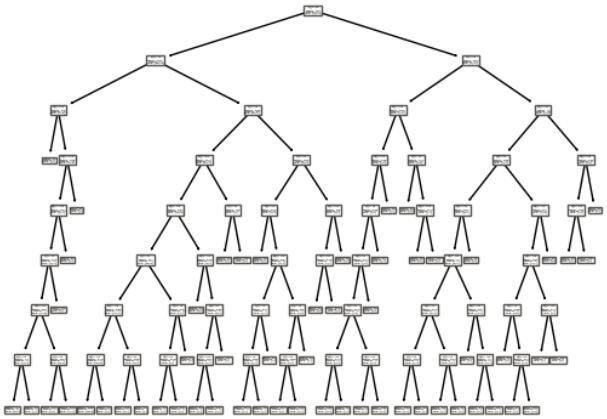

Figura 9: Árvore geral do melhor resultado

E olhando mais de perto a árvore citada anteriormente, com a Figura 10 é possível perceber que o primeiro fator que começou a criar a árvore, foi o sexo.

Do lado direito, tem-se as mulheres e do lado esquerdo, os homens. Depois, o fator que separa mais uma vez é a classe de embarque.

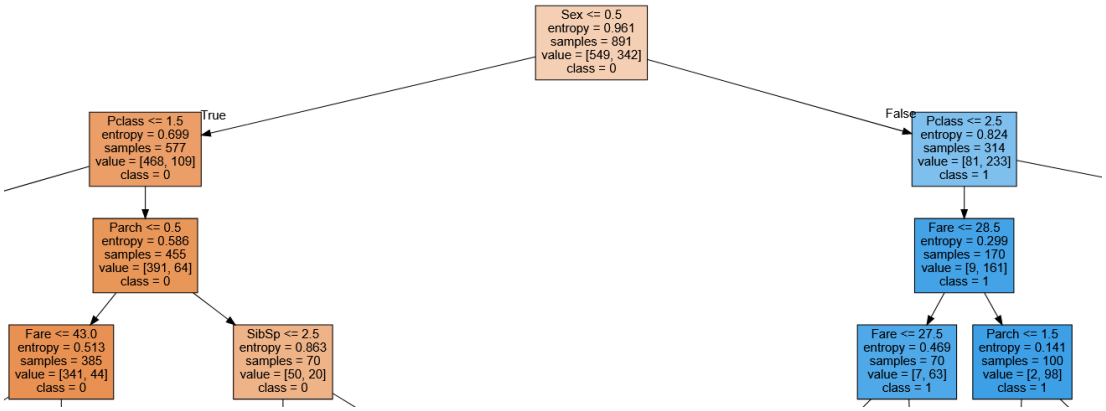

Figura 10: Primeiros nós

Com isso, pela árvore criada, pode-se notar que na Figura 11, o lado esquerdo foi mais mortal, logo, a conclusão que fica bastante nítida é a de que as mulheres foram as que mais se salvaram do acidente.

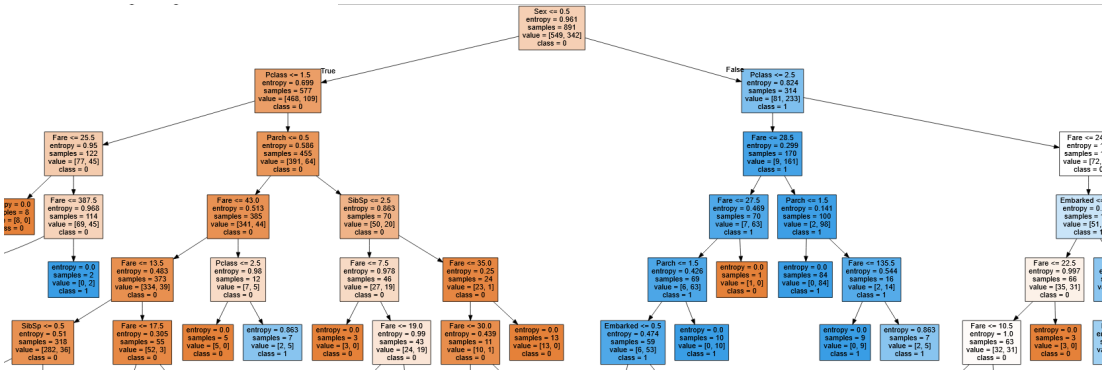

Figura 11: Separação dos nós iniciais da árvore final

Ao fazer uma pesquisa sobre o acontecido do Titanic, é informado que as primeiras pessoas que receberam prioridade no salvamento foram as mulheres, crianças e idosos, que em sua maioria eram os mais ricos. E após analisar melhor a árvore construída, as mulheres se enquandram no primeiro critério de separação, depois, a classe e o quanto as pessoas pagaram no ticket influenciaram bastante, uma vez que para os tickets mais baratos e classes mais baixas, no lado das mulheres que tem mais pessoas sobreviventes, foi onde se concentrou a maioria das mortes.

3 Conclusão

Após fazer a análise dos dados iniciais, entender um pouco mais sobre o conceito de árvore de decisão e como ela funciona, a importância da completude dos dados ficou bem mais clara. Além disso, também foi possível observar que uma singela mudança em um hiperparâmetro faz toda a diferença na construção de uma árvore, e que a criação de lógicas para encontrar boas relações entre os dados é fundamental para encontrar um resultado melhor pelo algoritmo trabalhado. Portanto, quanto mais analisados os dados forem, mais pesquisas feitas sobre o

contexto do problema tratado e criação de relações entre os fatores envolvidos, melhor a árvore vai ficar. Além disso, é importante ressaltar que nem sempre algo que parecia fazer sentido como uma boa relação vai funcionar, então é necessário estudar novamente as possibilidades e criar novas relações, sempre visando o melhor comportamento da árvore, baseando, também, nos hiperparâmetros envolvidos.

Isso tudo confirma o que a pesquisa sobre o titanic diz, e como a árvore chegou em 90% de acerto, pode-se afirmar que a árvore deu um resultado bom mas que pode ser melhorada com mais estudo e tentativas.