Tarefa de IA - ÁRVORE DE DECISÃO - 2020.1
Paula Macedo da Cruz
113049909

Relatório

1. O dataset

O dataset do Kaggle escolhido para este trabalho foi o dataset sobre quais fatores contribuem para a satisfação de clientes de uma companhias aéreas em relação aos serviços oferecidos. O link para o mesmo https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction.

São dois arquivos, train.csv para realização do treinamento e teste do modelo e test.csv que foi utilizado para validação do modelo.

O arquivo train.csv apresenta 103904 instâncias e test.csv 25976 instâncias originalmente. Em ambos arquivos há 24 atributos que serão melhor descritos adiante. Destes 24 atributos, 5 são categóricos e o restante numéricos.

O conceito alvo está representado pelo atributo “satisfaction”, que gera duas classes : sim - estar satisfeito e não - não estar satisfeito ou neutro.

1.1. Atributos e valores que os mesmos podem assumir

ID: Pode ser um número inteiro positivo qualquer

Gender: Female, Male

Customer Type: Loyal customer, disloyal customer

Age: Pode variar entre 7 a 85

Type of Travel: Personal Travel, Business Travel

Class: Business, Eco, Eco Plus

Flight distance: Pode variar entre 31 e 4983

Inflight wifi service: Pode variar de 0 - 5, onde 0 indica não aplicável

Departure/Arrival time convenient: Pode variar de 0 - 5, onde 0 indica não aplicável

Ease of Online booking: Pode variar de 0 - 5, onde 0 indica não aplicável

Gate location: Pode variar de 0 - 5, onde 0 indica não aplicável

Food and drink: Pode variar de 0 - 5, onde 0 indica não aplicável

Online boarding: Pode variar de 0 - 5, onde 0 indica não aplicável

Seat comfort: Pode variar de 0 - 5, onde 0 indica não aplicável

Inflight entertainment: Pode variar de 0 - 5, onde 0 indica não aplicável

On-board service: Pode variar de 0 - 5, onde 0 indica não aplicável

Leg room service: Pode variar de 0 - 5, onde 0 indica não aplicável

Baggage handling: Pode variar de 0 - 5, onde 0 indica não aplicável

Check-in service:Pode variar de 0 - 5, onde 0 indica não aplicável

Inflight service: Pode variar de 0 - 5, onde 0 indica não aplicável

Cleanliness: Pode variar de 0 - 5, onde 0 indica não aplicável

Departure Delay in Minutes: Pode assumir qualquer valor numérico positivo

Arrival Delay in Minutes: Pode assumir qualquer valor numérico positivo

Satisfaction: Satisfaction, neutral or dissatisfaction

2. Pré Processamento

Nesta etapa é realizado um tratamento dos dados.

A coluna ID é deletada de ambos arquivos, pois a mesma não contribui e nem interfere na construção do modelo.

Em seguida, é conferido se há existência de algum dado faltante. Em ambos os arquivos houve um atributo em específico que apresentava valor nulo. Esses exemplos, por não serem muitos, foram excluídos. Cerca de 309 exemplos de train.csv e 83 exemplos de test.csv . A decisão foi tomada, pois ambos arquivos apresentavam um grande número de exemplos e excluir alguns dados faltantes não impactaria na construção do modelo e nem na classificação. Talvez, deixar os dados faltantes pudesse até prejudicar a construção do modelo.

Os dados categóricos são convertidos para que o classificador de árvore de decisão do scikit funcione.

Alguns dados numéricos se encontravam em escalas muito diferentes, então foi feita uma reescala desses dados. Os atributos que tiveram seus dados com uma nova escala foram: 'Age', 'Flight Distance', 'Departure Delay in Minutes', 'Arrival Delay in Minutes'. A reescala foi feita utilizando o método *QuantileTransformer* e usando como parâmetro *output_distribution='normal'*. Esse método atua sobre as colunas selecionadas, transformando os valores de forma que os mesmos tendem a se aproximar de uma distribuição normal.

3. Experimento base

Após o tratamento dos dados, foi checado a quantidade de exemplos com classificação sim e a quantidade de exemplos com a quantidade não. Como para cada arquivo, a classificação “sim” ficou em torno de 57% e classificação “não” ficou

em torno de 43%, isso mostra que os arquivos encontram-se balanceados, algo muito bom para construção do modelo.

O trabalho leva em consideração apenas a taxa de acerto da classificação, por isso, só é levado em consideração a acurácia.

Durante a realização de todos os experimentos, o atributo “Gender” não é utilizado.

A decisão de manter todos os atributos, exceto “Gender” ocorreu, pois avaliando cada um dos atributos restantes, todos pareciam ser realmente importantes para construção da avaliação.

Poderia ser utilizada uma propriedade chamada feature_importances para contribuir para essa escolha de atributos. Mas como demandava tempo para entender como funcionava a mesma, num primeiro momento, optei por não utilizá-la. Para saber mais sobre essa propriedade é possível encontrar neste link https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html . E realizei a escolha dos atributos utilizados para experimento base pelo o que eu julgava ser os fatores mais importantes.

O experimento não utiliza k fold, apenas a divisão de train.csv para treino e teste e o uso de test.csv para fornecer a validação do modelo.

O experimento base tem a divisão de train.csv em 55% para treinamento e 45% para teste. A ideia era ter um experimento inicial que permitisse observar a piora ou melhora dos resultados quando se aumentasse o tamanho do conjunto de treinamento e diminuísse o tamanho do conjunto de testes. E a melhora ou piora dos resultados após a retirada de alguns atributos. Essa mudança de tamanho dos conjuntos seria alterada posteriormente para 20% teste e 80% treinamento, que é o considerada a divisão ideal e também para 40% teste e 60% treinamento.

O método DecisionTreeClassifier() é utilizado com todos os parâmetros default. O gini é usado, pois a ideia era minimizar os erros de classificação que poderiam existir e para esse caso, o gini parecia ter desempenho melhor que o cálculo da entropia. Para checar se eu estava certa, eu alterei esse parâmetro para entropia e não houve nenhuma mudança significativa da acurácia para teste. Logo, para esse caso, a escolha entre gini ou entropia não tinha tanta relevância.

O random_state= 0 é para manter sempre a mesma parte do dataset selecionada a cada execução. Essa decisão foi tomada, pois considerei ser algo que facilitasse a comparação de resultados.

O shuffle encontra-se falso, pois inicialmente considerei que os dados não estariam ordenados em função da classe.

4. Explorar alternativas

Fazendo a alteração das porcentagens houve sim uma melhoria da acurácia, mas algo quase que insignificante. Com o treinamento em 80% e o teste em 20% a acurácia de teste melhorou, mas a acurácia de validação diminuiu. Apesar de ser uma diminuição de cerca de 0.001, isso não gera a conclusão de que o modelo gerado seja ruim.

A próxima mudança foi alterar o shuffle de False para True, a ideia era ver o que acontecia ao embaralhar os dados. Se após esta ação os dados estivessem muito mais balanceados, até mais que a versão com shuffle falso. Não houve uma modificação significativa nas acurácias, o que mostra que o dataset estava bem balanceado ou talvez o treinamento teria uma classe com mais incidência que a outra em ambos os casos. Mas para descobrir isso eu precisaria checar outras medidas, sensibilidade ou precisão. Para esse meu conceito, para esse meu dataset checar outras medidas não parece ser necessário, pois a ideia é descobrir o quão eficiente é meu modelo e quanto ele consegue acertar.

No geral, nenhuma dessas modificações gerou uma alteração significativa nas acurácias de teste e validação.

Depois, além de “Gender” foi retirado mais 5 atributos. Estes atributos possuíam o valor 0, de não se aplicam. Após isso, a acurácia que antes era de em torno de 94% para treino e validação, teve uma queda expressiva para 82%. O que mostra que talvez seria necessário usar a propriedade mencionada anteriormente, a feature_importances , para definir quais atributos deveriam ser excluídos do modelo.

Por fim, realizei o mesmo teste, retirando apenas 3 atributos. Foi escolhido “Gender”, “Age” e “Customer type”, atributos mais relacionados ao perfil do passageiro e não ao serviço em si. Como esperado, a acurácia do teste voltou à proximidade dos 93%.

Isso me levou a conclusão de que para esse caso, a escolha do atributo certo influenciou mais na acurácia do que a troca dos parâmetros. E a escolha dos parâmetros mais focados no serviço e não no perfil do passageiro, acabam tendo acurácias melhores.

5. Comparação de resultados com o do Kaggle

Comparando os resultados obtidos com os do Kaggle, mostra que escolher os atributos iniciais de forma aleatória não conduz a bons resultados. Mostrando a necessidade de se usar uma nova abordagem durante a escolha. Mas no geral, meu resultado final de 93% não foi tão ruim em comparação ao resultado de 95,4% do Kaggle que utiliza regressão linear e outras abordagens.