UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS - C3
ENGENHARIA DE COMPUTAÇÃO

SISTEMAS INTELIGENTES

Comparação entre os algoritmos de classificação KNN e Árvore de Decisão

Gabriel Moraes, 88919
Rafael Bulsing, 85477
Vinicius Lucena, 85522

1 Introdução

A sub-área da Inteligência Artificial, conhecida como Aprendizagem de Máquina, vem ganhando espaço e já faz parte do dia-a-dia da maioria das pessoas. Prova disso são os sistemas criados por grandes empresas como a Google (Home), Amazon (Alexa), Microsoft (Cortana), Apple (Siri), entre outras. Todos esses sistemas possuem, por trás, um grande trabalho de Inteligência Artificial.

Atualmente a Aprendizagem de Máquina é tão difusa que existem diversos métodos diferentes que podem ser utilizados em cima de múltiplos *datasets***,** dependendo somente do objetivo no qual se quer chegar. Neste trabalho, utilizamos dois métodos que serão citados abaixo na sessão 2.

2 Objetivo

O presente relatório tem por objetivo comparar dois algoritmos que implementam a técnica de classificação. Os algoritmos escolhidos foram Árvore de Decisão e KNN. A comparação entre os algoritmos levará em consideração o tamanho do conjunto de treino e o conjunto de teste.

3 Metodologia

3.1 Dataset

O *dataset*[1] escolhido trata-se de informações referentes a estrelas, sendo que o atributo-alvo diz se a estrela é pulsar ou não. Foi escolhido um problema de classificação binária pela sua simplicidade, porém, as técnicas abordadas neste relatório se aplicam também à classificação envolvendo múltiplas classes.

3.2 Métricas de Avaliação

Com o objetivo de comparar os algoritmos, foram utilizadas a acurácia, o *recall*[2] e o *precision* como métricas de avaliação, devido sua alta aceitação no meio acadêmico e na indústria.

[1] Disponível em: https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star
[2] Também conhecido como TPR (*True Positve Rate*)

3.3 Parametrização dos algoritmos

Para a implementação dos classificadores, o *dataset* utilizado foi analisado e pré-processado. Primeiramente, para não haver uma tendência maior entre as *features*, nós realizamos uma normalização nos dados. A normalização dos dados é importante nesse cenário pois foi utilizado um algoritmo baseado em distâncias (KNN). Em seguida, para realizar um treinamento equilibrado, foi necessário balancear o *dataset*, pois o mesmo continha quase 10 vezes mais alvos em 0 (não pulsar) do que em 1 (pulsar). Assim, utilizando uma técnica de *undersampling*[3], balanceamos os dados.

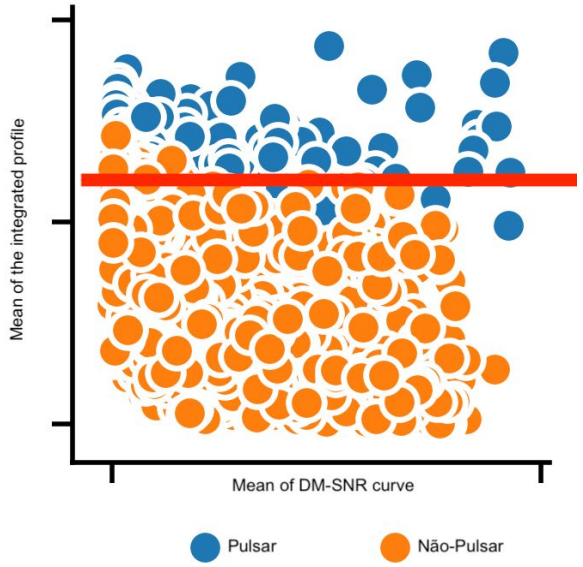
Para implementar a técnica de árvore de decisão, foi utilizado a classe *DecisionTreeClassifier*, provida pela biblioteca *Scikit-Learn,* o qual implementa uma versão otimizada do algoritmo CART.

Para o KNN, foi utilizado o classificador *KNeighborsClassifier,* também provido pela biblioteca *Scikit-Learn*, com parâmetro K=3, isto é, o algoritmo considera os 3 vizinhos mais próximos para rotular cada nova amostra durante a fase de treinamento.

3.4 Análise Exploratória

Com o objetivo de determinar quais as melhores *features* para dividir as estrelas entre pulsar e não-pulsar, foi utilizado a análise do gráfico *scatter*, como é ilustrado pela figura 1. Observa-se que somente ao olhar para a imagem, é possível separar (com um certo erro) as estrelas pulsar e não-pulsar, nesse sentido, vemos que o atributo *Mean of integrated profile* é discriminante na classificação das estrelas.

Figura 1. *Mean of integrated profile versus Mean of DM-SNR curve*

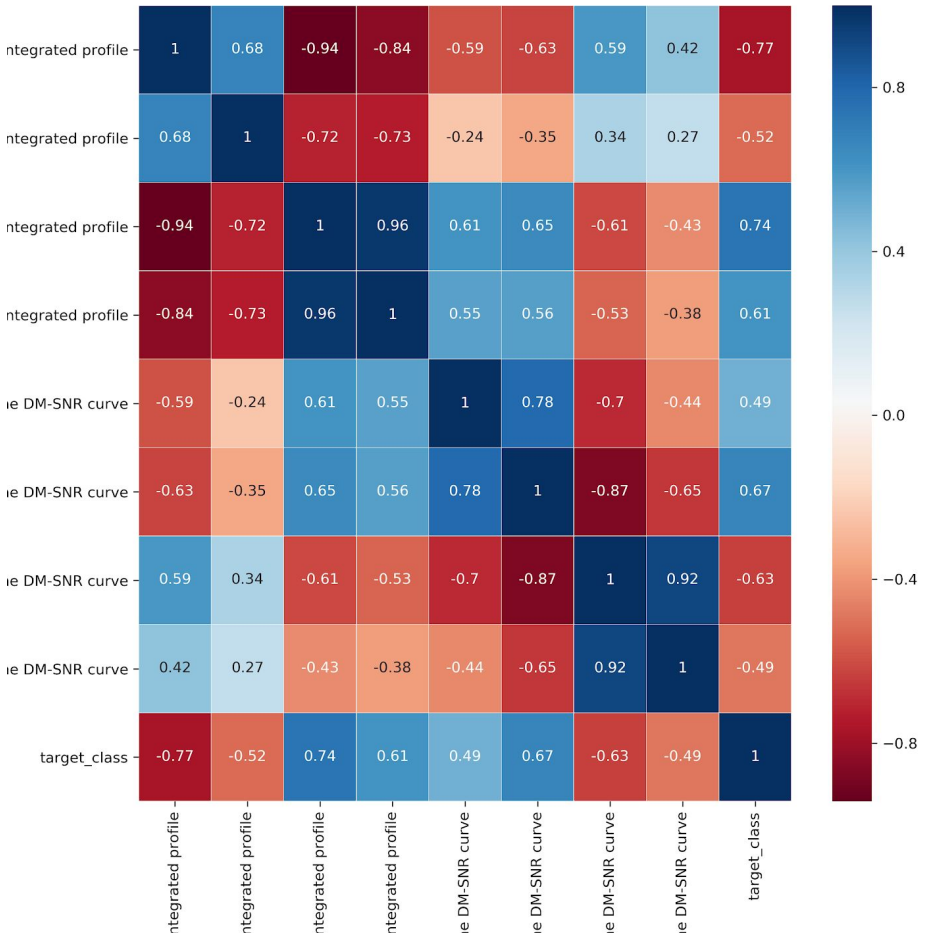


[3] A técnica de *undersampling* consiste em pegar amostras da classe majoritária, de forma a tornar igual ao número de amostras da classe minoritária.

Outra maneira de analisar a relação entre as variáveis, é observar a matriz de correlação, como é ilustrado na figura 2. Embora o problema não seja de regressão, analisar a matriz de correlação dos atributos é de suma importância. Para o atual problema, nota-se que o atributo *mean of integrated profile* possui uma correlação forte e negativa com o atributo-alvo (algo que já tinha-se observado ao analisar a figura 1).

De maneira geral, não é desejável que haja forte correlação dos atributos entre si, apenas entre o atributo-alvo e as demais *features*. Caso isso aconteça, o algoritmo de classificação estaria sendo alimentado com dados duplicados, o que deixa os algoritmos de aprendizagem de máquina mais lentos e menos eficientes.

Figura 2. Matriz de correlação.



4 Resultado

A figura 3.a demonstra como a acurácia dos algoritmos KNN e Árvore de Decisão variam com relação ao tamanho do conjunto de teste. Para o atual dataset, não houve grandes variações nos níveis de acurácia dos modelos, por se tratar de um conjunto muito bem pré-processado, ambos algoritmos tiveram bom desempenho.

Quando se analisa a curva de *Recall* (taxa de verdadeiro positivo) *versus* tamanho do conjunto de teste (figura 3.b) observa-se que ambos algoritmos se comportaram de

maneira muito próxima até o tamanho do conjunto de teste de 80%, após isso, o KNN teve melhor desempenho. Pode-se interpretar tal resultado dizendo que o KNN necessita de menos dados de treinamento em relação a Árvore de Decisão (para este conjunto de dados).

Em relação ao *Precision versus* o tamanho do conjunto de teste (figura 3.c), nota-se que o algoritmo KNN tem um desempenho superior para qualquer tamanho do conjunto de teste.

Figura 3.a: Acurácia *versus* Conjunto de Teste

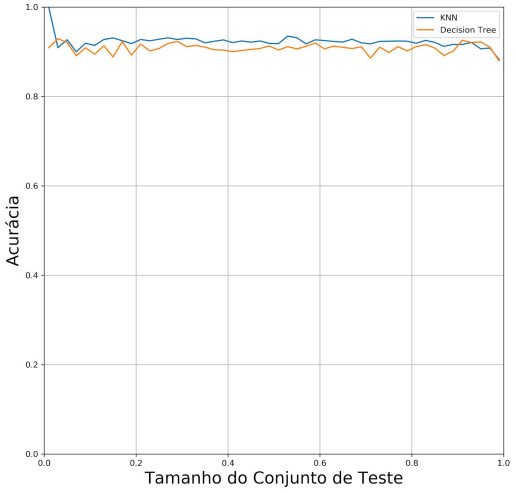


Figura 3.b: *Recall versus* Conjunto de Teste

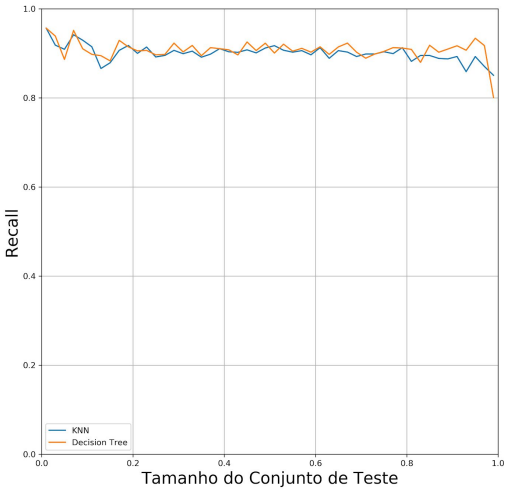
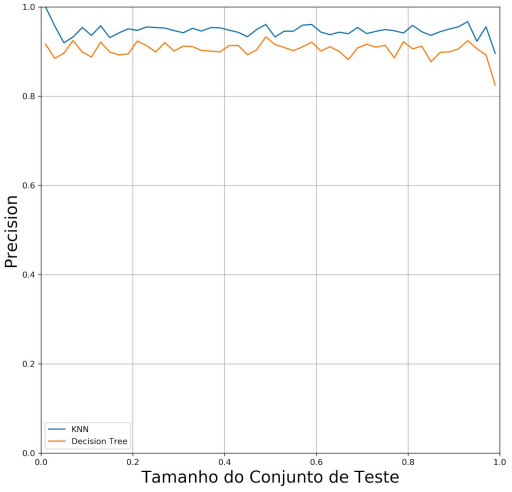


Figura 3.c: *Precision* versus Conjunto de Teste



5 Conclusão

Pode-se concluir, através da figura 4, que o algoritmo KNN oferece melhor desempenho em termos de acurácia, *recall e precision*, independente do tamanho do conjunto de teste e de treino. Isso não significa que o KNN sempre será melhor do que um algoritmo que implementa a árvore de decisão, dependerá do conjunto de dados que será utilizado. Para o presente relatório, foi utilizado um *dataset* numérico e normalizado, o que é ideal para algoritmos baseados em distância, como é o caso do KNN.