Comparação de Técnicas de Aprendizado de Máquina

Carlos Eduardo da Silva , Giovanni do Nascimento Castro Filho , Mario Arthur Sclafani Pujati , Vitor Silva Reis

Universidade Federal de São João del–Rei

carlose060@gmail.com, {giovannifilho0905, mario_arthur_spujatti, vitorejuvian}@hotmail.com

Abstract

This short example shows a contrived example on how to format the authors' information for *IJCAI–21 Proceedings* using LaTeX.

1 Introdução

Atualmente os algoritmos de inteligência artificial, em especial os de aprendizado de máquina, tem se mostrado cada vez mais presente no dia a dia, sendo utilizados nos mais diversos tipos de tarefas, diferentemente dos algoritmos comuns que utilizam cálculos lógicos e matemáticos para resolver problemas, estes tentam aplicar o conhecimento adquirido em situações diversas para resolvê-los.

Os algoritmos de aprendizado supervisionados, por exemplo, utilizam bases de dados e tentam relacionar entradas e saídas, estas bases de dados tendem a ser obtidas através de um usuário que fica encarregado de supervisionar o progresso do algoritmo, sendo responsável por fornecer as bases, corrigir falhas e impedir que o algoritmo entre em casos de overfitting, que ocorre quando o algoritmo se ajusta demais aos exemplos fornecidos pela base, e underfitting, que ocorre quando o algoritmo generaliza demais os exemplos fornecidos.

Com o auxílio destes pode-se resolver problemas que antes seriam necessária atenção de um humano, mas com o auxílio da máquina o problema pode ser resolvido de maneira muito mais rápida e algumas vezes até mais eficiente.

Para problemas de alto custo computacional, um modelo de aprendizado é muito mais adequado que uma resolução convencional dada sua natureza *onde tentam replicar conhecimento adquirido em diversas situações, conseguindo assim uma generalização, dado um determinado problema [Luger 2004].*

"A área de Machine Learning tenta resolver problemas em que seria necessária uma atenção de um humano, mas com o diferencial de ser uma máquina que, por sua vez, realiza cálculos de maneira mais veloz" – Mitchell, 1997.

Muitos acham que um algoritmo de machine learning trata-se apenas de um amontoado de *"if/else"*, o que não é verdade já que seria custoso tanto para projetar, quanto para executar os comandos. Funções matemáticas são mais simples de implementar, barato no custo computacional e com maior possibilidade de generalização [Braga *et al.,* 2000].

Como sempre, estes algoritmos nunca são perfeitos, ainda necessitando de ajustes e de diferentes algoritmos para diferentes tarefas, atualmente não existe um algoritmo perfeito para todas as situações mas existem algoritmos muito eficientes para situações específicas.

Neste trabalho teremos como objetivo demonstrar a eficiência de diferentes tipos de algoritmos em diferentes tipos de bases de dados (datasets), e comparar os resultados entre eles para mostrar seus pontos fortes e fracos em cada situação apresentada.

2 Desenvolvimento

Para a realização deste projeto desenvolvemos um algoritmo em Python visando facilitar a compreensão, os testes foram realizados utilizando hold-out, onde 80% dos dados foram destinados ao treinamento e 20% para os testes de acurácia.

2.1 Base de Dados

Utilizamos a biblioteca sklearn para importar as bases de dados necessárias, realizamos os testes em 4 delas, sendo estas Íris, Digits, Wine e Breast Cancer, todas do tipo classificação, todas possuindo poucas classes e valores, assim sendo consideradas de dificuldade fácil.

Wine

A base de dados Wine possui 3 classes e 13 atributos. As classes são *class 0*, *class 1* e *class 2* e os 13 atributos são relacionados às características de cada vinho, como mostrado na figura 1. Ao total estão dispostas 178 instâncias para aprendizado e testes de algoritmos.

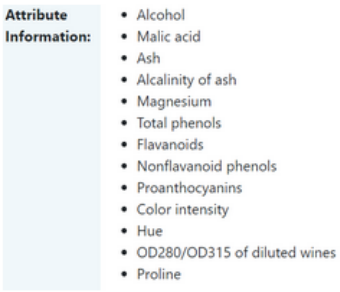

Figure 1: Atributos de Wine

Após os testes podemos observar na figura 2 que nosso modelo KNN obteve resultados tão satisfatórios quanto os outros modelos, mostrando que ele não sofre de overfitting e muito menos de underfitting.

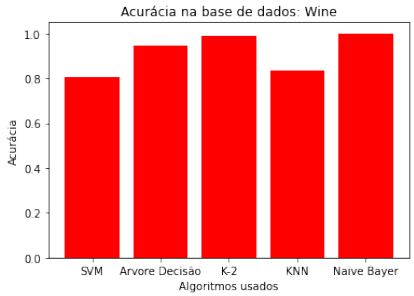

Figure 2: Acurácia dos algoritmos no dataset Wine

Iris

A base de dados íris possui 3 classes (figura 3) e 4 atributos. Os atributos são separadas em *sepal_length*, *sepal_width*, *petal_length*, *petal_width* e as classes são Setosa, Versicolour e Virnica. Ao total estão dispostas 150 instâncias para aprendizado e testes de algoritmos.

Figure 3: Classes de Iris

Após realizarmos alguns testes através do hold-out, podemos observar na figura 4 que a maioria dos modelos conseguem obter uma alta acurácia devido aos poucos parâmetros necessários para classificação.

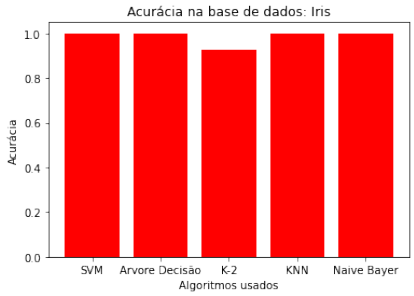

Figure 4: Acurácia dos algoritmos no dataset Iris

Digits

A base de dados Digits possui 10 classes e sua entrada é uma imagem 8x8 (Figura 5). Cada uma de suas classes representa um dígito decimal de 0 á 10 que representam a cor em cada pixel. Ao total estão dispostas 1797 instancias para aprendizado e testes de algoritmos.

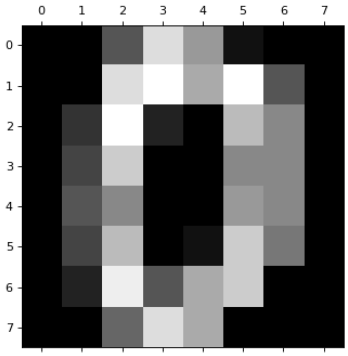

Figure 5: Exemplo de uma amostra de Digits

Após os testes foi observado que os algoritmos tiveram uma acurácia superior a 80% (gráfico na figura 6). Por ser uma base grande, com vários valores de entrada no entanto é mais difícil realizar um treinamento rápido.

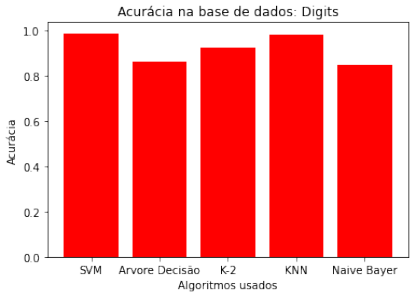

Figure 6: Acurácia dos algoritmos no dataset Digits

Breast Cancer

A base de dados Breast Cancer possui 2 classes e 30 artibutos. Suas classes representam somente os tipos de tumores malignos e benignos e suas entradas são características desses tumores.

Estão disponíveis 357 instancias de tumores benignos e 212 de tumores malignos, contabilizando 569 instancias dispostas para aprendizado e testes de algoritmos.

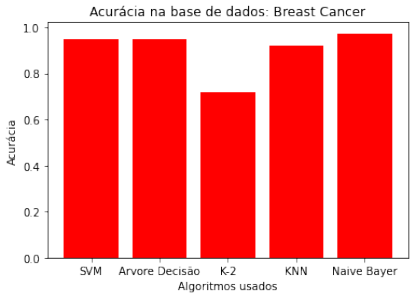

Figure 7: Acurácia dos algoritmos no dataset Breast Cancer

Após a realização dos testes tivemos a pior acurácia por

volta de 95%, ou seja, podemos concluir que não houve Underfitting nem Overfitting e obtivemos um resultado mais que satisfatório, com exceção do algoritmo K-2, que obteve um resultado bem abaixo dos demais. (Mostrado na Figura 7)

3 KNN

3.1 K Nearest Neighbor

Neste algoritmo possuímos uma variável chamada de K, a qual é parte do nome do modelo e também o principal parâmetro a ser selecionado. Este parâmetro direcionará a quantidade de vizinhos. Imagine que temos um valor P1 o qual queremos predizer, entre um grupo de duas classes onde o valor atribuído a K foi 1 (K=1), primeiro iremos identificar o ponto mais próximo a ele e depois qual a label que o identifica (classe A por exemplo) como mostra a Figura 8.

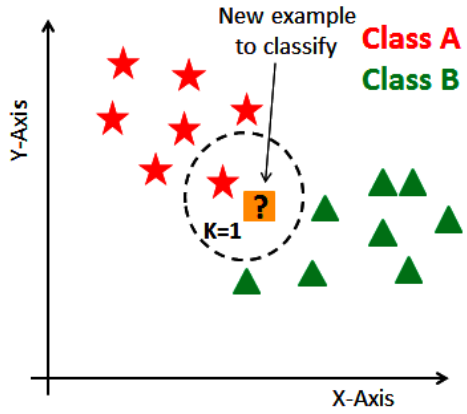

Figure 8: Escolhendo os k vizinho

Após identificar o ponto mais próximo e identificar a label deste ponto (Ex.: Classe A), iremos predizer a que classe o ponto P1 faz parte.

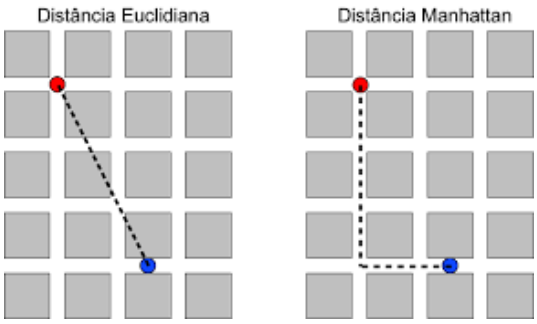

Figure 9: Distância Euclidiana e Distância Manhattan

Para identificar de fato a que grupo o ponto P1 faz parte, iremos realizar uma votação onde a maioria irá dizer a que classe este ponto P1 realmente faz parte. Iremos utilizar medida de distância para identificar a distância existente entre o ponto P1 e os demais pontos do meu dataset, como K=1 o algoritmo irá verificar ponto a ponto, caso coloque o valor K=3 ele ira olhar a distância de P1 em relação a 3 pontos. Desta forma termos e distância existente entre P1 e todos os pontos do meu dataset, assim conseguiremos saber a quais pontos P1 é mais próximo, desta forma teremos qual classe ele é mais similar. Assim a "votação" será concluída, e saberemos como classificar P1. E para encontrar essas distâncias, podemos utilizar medidas como, distância euclidiana, distância de Manhattan, distância de Hamming etc. (Distância euclidiana e distância de Manhattan mostrada na figura 9.)

Como o algoritmo iria classificar o 'bloco em amarelo' na Figura 10.

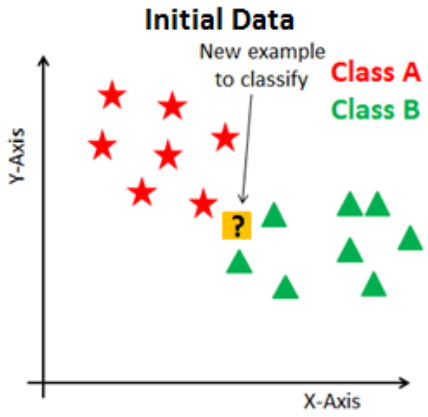

Figure 10: Classificar o bloco

O passo a passo pelo algoritmo é:

1. Calcular a distância - como é mostrado na Figura 11 - utilizando alguma das medidas ja citada acima.
2. Encontrar os vizinhos mais próximos (como mostra na Figura 12). Sendo k = 3 no exemplo, ele mapeia os 3 vizinho mais próximos.
3. Votar a label para o ponto a ser previsto.

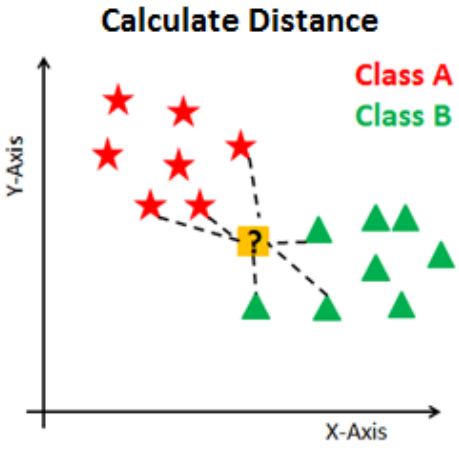

Figure 11: Calculando as distâncias.

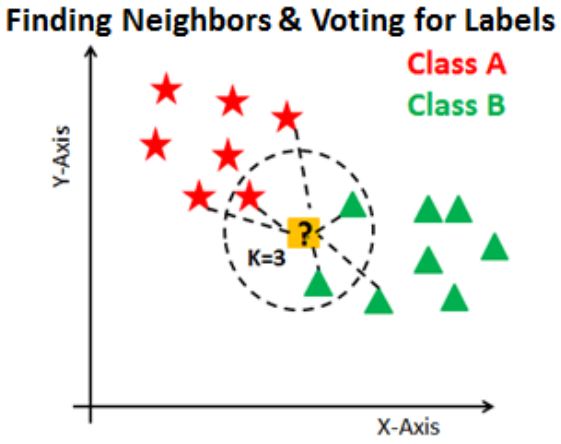

Figure 12: Iniciando a classificação

3.2 Implementação KNN

O algoritmo recebe os arquivos já formatado e divido, sendo uma parte de treino e outra de teste, além disso, dividido entre feature e rótulos. (Figura 13)

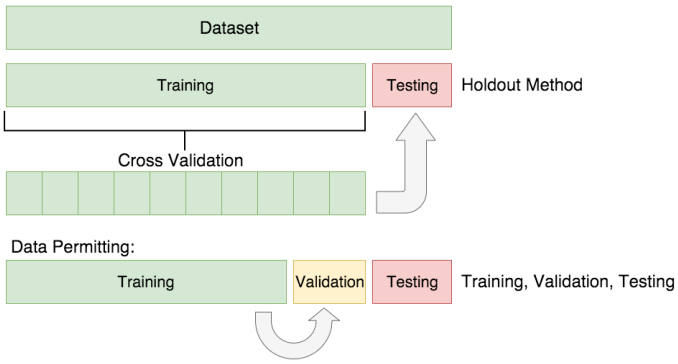

Figure 13: Divisão dos dados

Na seção de treinamento, ele armazena todos os itens de treino com o seu tipo. A parte de teste é feita a comparação da predição das distância com o rótulo da base de dados. Para realizar a predição, calcula-se a distância de todos os pontos (figura 11), ordena e faz a comparação de k's pontos mais proximo (no algoritmo k=3), como é mostrado na figura 12. Após isso o tipo mais presente no intervalo é retornado para comparação do rótulo da base de dados.

4 SVM

Métodos de aprendizado supervisionado que analisa os dados e reconhecem padrões, usado para classificação e análise de regressão. O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico. Dados um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra.

Um modelo SVM é uma representação de exemplos como pontos no espaço, mapeados de maneira que os exemplos de cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível. Os novos exemplos são então mapeados no mesmo espaço e preditos como pertencentes a uma categoria baseados em qual o lado do espaço eles são colocados.

Em outras palavras, o que uma SVM faz é encontrar uma linha de separação, mais comumente chamada de hiperplano entre dados de duas classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes, como mostra a figura 14

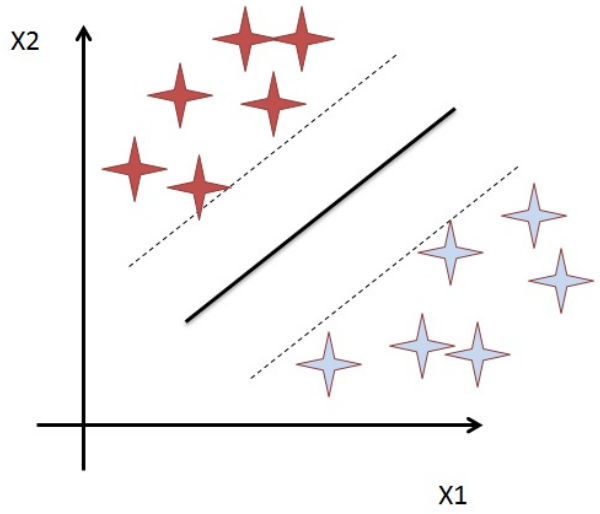

Figure 14: Algoritmo SMV

Essa distância entre o *hiperplano* e o primeiro ponto de cada classe costuma ser chamada de margem. A SVM coloca em primeiro lugar a classificação das classes, definindo assim cada ponto pertencente a cada uma das classes, e em seguida maximiza a margem. Ou seja, ela primeiro classifica as classes corretamente e depois em função dessa restrição define a distância entre as margens.

5 Árvore de Decisão

É uma das abordagens de modelagem preditiva usadas em estatística, mineração de dados e aprendizado de máquina. Ele usa uma árvore de decisão (como um modelo preditivo) para ir de observações sobre um item (representado nos ramos) para conclusões sobre o valor alvo do item (representado nas folhas). Os modelos de árvore em que a variável de destino pode assumir um conjunto discreto de valores são chamados de árvores de classificação; nessas estruturas de árvore, as folhas representam rótulos de classe e os ramos representam conjuntos de características que levam a esses rótulos de classe. As árvores de decisão em que a variável de destino pode assumir valores contínuos (normalmente números reais) são chamadas de árvores de regressão. As árvores de decisão

estão entre os algoritmos de aprendizado de máquina mais populares devido à sua inteligibilidade e simplicidade.

Um exemplo de uma Árvore de Decisão de Jogar Ténis são tidos em conta exemplos (dias) passados (exemplo figura 15):

Exemplos de Treino

Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Figure 15: Tabela de exemplo

Através destes exemplos é possível construir a seguinte árvore de decisão(figura 16):

Figure 16: Arvore de decisão

A relação entre os elementos da árvore (nós e folhas) e os atributos, valores e classificações pode ser entendida na seguinte imagem(figura 17):

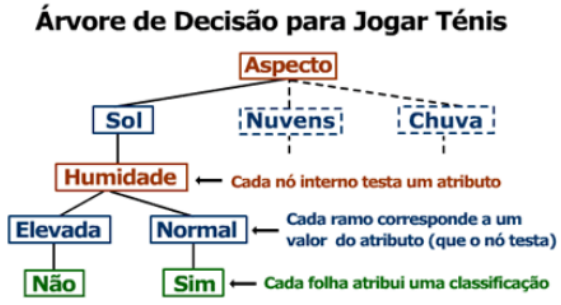

Figure 17: Relação entre os elementos da árvore

A classificação de um exemplo de acordo com a esta árvore é feita da seguinte forma(figura18):

Figure 18: Classificando a arvore de decisão

O atributo ”Aspecto” tem o valor ”Sol” e a ”Humidade” tem o valor ”Elevada”. O exemplo é classificado como ”Não”, ou seja, quando esteve ”sol” e ”humidade” elevada não se jogou ténis. Os atributos ”Temperatura” e ”Vento” não são considerados, pois são desnecessários para classificar este exemplo.

Figure 19: Arvore de decisão: Aspecto: sol (e) vento: fraco

Figure 20: Arvore de decisão: Aspecto: sol v(ou) vento: fraco

Com Árvores de Decisão é possível representar a conjunção e disjunção de atributos. A árvore de decisão que representa a classificação para os dias em que o ”Aspecto” é ”Sol” e que o ”Vento” está ”Franco” encontra-se na figura 19.

A árvore de decisão que representa os dias em que o Aspecto é Sol ou o Vento está Franco é dada pela figura 20.

6 Conclusão

Após a realização de diversos testes, utilizando os conjuntos de dados propostos, todos os modelos que testamos se saíram bem, já que todos os modelos tem um bom desempenho em problemas de classificação, que foi tipo de problema no qual realizamos os testes, no entanto, podemos concluir também que não existe um modelo que seja ótimo para toda situação. Ou seja, o desempenho de um modelo está relacionado com a situação na qual ele é aplicado, como foi demonstrado nos nossos testes.

Por se tratar de uma base simples, uma classe pode muito facilmente ser separada das outras. Por isso, ao desenvolver um modelo para classificações distantes entre as classes, qualquer técnica pode ser utilizada. Dadas essas características no conjunto de dados Íris todos os modelos testados conseguiram um bom desempenho.

Já no caso da base da dados "Wine" de acordo com nossos testes, pode-se perceber uma certa dificuldade na classificação, visto que a média de acurácia foi por volta de 80%, o que deixa claro que seria necessária uma mudança nos modelos. Uma das opções seria por exemplo SVM, que pode ser modificado conforme a função é modelada.

7 Referências

Russel, S., Norvig P. Inteligência Artificial, 2a edição. Elsevier. 2004

J. Hair Jr, R. Anderson, R. Tatham, W. Black, Análise Multivariada de Dados, Artmed, 2005.

Luger, George F. Inteligência Artificial Estruturas e Estratégias para a solução de problemas complexos. 4a edição. Bookman. 2004

Rich, Elaine. Inteligencia artificial. Sao Paulo: McGraw-Hill, 1988

Braga, A.P; Carvalho, A. C. P.; Ludermir, T. B. Redes Neurais Artificiais - Teoria e aplicações. Livros Técnicos e Científicos, Editora S.A, 2000.

T. Mitchell, Machine Learning, McGraw Hill, 1997.