

Comparação dos Resultados de Algoritmos de Classificação

Augusto Ribas¹, Bruno Nazário¹ e Douglas Sorgatto¹

¹Faculdade de Computação - Universidade Federal de Mato Grosso do Sul

Resumo

Avaliamos o desempenho de três algoritmos de classificação: o KNN (K-nearest neighbors), a árvore de decisão e o Naive Bayes para 10 conjunto de dados públicos. Foi feita a normalização dos dados, a separação em 10-folds para validação cruzada estratificada e, por fim, foi realizado o cálculo da acurácia e do desvio padrão das informações encontradas. Os resultados comprovam que o melhor algoritmo de classificação para os conjunto de dados que utilizamos foi o KNN e que classificar dados é uma tarefa importante para a compreensão da realidade e dos muitos dados produzidos.

Palavras-chave: Algoritmos de classificação, acurácia, desempenho, inteligência artificial.

1 Introdução

O problema da classificação é central nos trabalhos de inteligência artificial. São várias as situações onde é necessário separar, organizar e visualizar os dados de forma organizada [8], e é isso que pretende os algoritmos de classificação que apresentaremos neste relatório.

Os algoritmos solicitados pelo professor para que fizéssemos o estudo de seu desempenho em conjunto de dados variados foram escolhidos por sua simplicidade de compreensão, eficiência e eficácia. Foram solicitados a avaliação do K-Vizinhos mais próximos (KNN- K-nearest neighbors), a árvore de decisão e o Naive Bayes. Cada algoritmo tem seu ponto forte e sua fraqueza para situações específicas. Portanto, este trabalho pretende avaliar esse desempenho, comparando o uso dos três algoritmos para os mesmos conjunto de dados.

A descrição e avaliação dos algoritmos são partes do relatório que segue.

1.1 Problema

O problema que trata esse relatório é a comparação dos resultados para o uso dos algoritmos de classificação ensinados pelo professor da disciplina de Inteligência Artificial. Devemos aplicar os algoritmos a partir das bibliotecas disponíveis na linguagem Python, sendo a principal a SKLearn, e avaliar os desempenhos a partir da acurácia demonstradas nos mesmos conjunto de dados.

1.2 Objetivos

Aplicar os algoritmos KNN, Decision Tree e Naive Bayes para 10 (dez) conjunto de dados distintos.

Comparar os resultados das acurácias destes algoritmos no tratamento dos dados.

Comprovar os resultados através de gráficos, tabelas e informações complementares que enriqueçam e comprovem as informações.

2 Material e Métodos

O trabalho consiste em escolher 10 conjunto de dados públicos no site UCI - Machine Learning¹ que fossem próprios para o problema de classificação de dados. Neste site há 234 conjunto de dados disponíveis para classificação. Os critérios que utilizamos para a escolha dos que usamos neste trabalho foram:

- Atributos reais ou inteiros;
- Baixo número de atributos para classificação;
- Número de instâncias superior a 100;

Depois de escolhermos os conjunto de dados que estão descritos abaixo, seria preciso implementar os algoritmos de classificação propostos pelo professor. Para uso destes algoritmos foi necessário aprender a utilizar a biblioteca SKLearn [1], para a linguagem Python. Esta biblioteca gratuita, aberta e

¹Disponível no site <http://archive.ics.uci.edu/ml/index.html>

apropriada para linguagem Python² possui implícita os algoritmos solicitados pelo professor e vários outros que não utilizamos.

Foi necessário um bom tempo de estudo para o uso da biblioteca, pois esta exige o conhecimento e uso de outras bibliotecas que facilitam a manipulação de dados, o tratamento de erros e agilizam a execução do programa, bem como possibilitam a produção de gráficos e árvores de decisão. As bibliotecas que utilizamos nos algoritmos para obtermos os resultados analisados nesta trabalho foram: *Numpy* [10], *Panda* [7] e *Matplotlib* [6].

2.1 Algoritmos de Classificação

Após o estudo sobre o funcionamento das bibliotecas necessárias para o trabalho teve início a análise do funcionamento dos algoritmos propriamente ditos. As ideias gerais de seu funcionamento foram dados pelo professor durante as aulas teóricas ao longo do semestre. Um resumo do que fazem esses algoritmos pode ser encontrado a seguir.

2.1.1 Vizinhos mais Próximos

O algoritmo KNN, K-Nearest Neighbors, ou Vizinhos mais Próximos, é considerado um dos algoritmos mais simples para a classificação de dados do paradigma de aprendizado supervisionado. Este algoritmo precisa de uma base de dados que será consultada quando for preciso classificar uma nova instância de dados. Por essa característica é conhecido como um algoritmo “Lazy”.

O algoritmo trabalha com uma função que calcula a distância da nova instância para as instâncias da base de dados que possui, retornando a classificação pela “proximidade” com essas informações. Uma informação importante para o bom funcionamento do algoritmo é definir o “K” que será usado para comparar as novas instâncias.

Esse “K” é o número de vizinhos que serão considerados para rotular a nova instância é muito importante, pois um “K” pequeno pode sofrer forte influência de ruídos ou dados falsos (*outliers*), um “K” tem um custo computacional alto e não necessariamente retornar um resultado com melhor acurácia.

Neste trabalho avaliamos apenas 5 possíveis valores para “K” e escolhemos aquele que teve o melhor resultado de acurácia para os conjuntos de teste. Os valores que escolhemos foram: 3, 5, 7, 9 e 11. Não se adota escolher “K” como número par para evitar a possibilidade de empates.

²Disponível para download no site <http://scikit-learn.org/stable/>

2.1.2 Árvore de Decisão

É um algoritmo que usa diagramas para mapear as varias alternativas e resultados de decisões, baseados nas probabilidades de ocorrerem. Baseia-se em estimativas e probabilidades associadas aos resultados de cursos de ação que competem entre si. O resultado de cada curso de ação é ponderado pela probabilidade associada a ele. O resultado ponderado é somado e o valor esperado de cada curso de ação é, então determinado. A alternativa que proporciona o valor esperado mais alto é preferível.

Essencialmente, árvores de decisões são diagramas que permitem representar e avaliar problemas que envolvem decisões sequenciais, colocando em destaque os resultados identificados nos diversos cursos de ação. É um algoritmo cuja interpretabilidade é fácil para os humanos, ou seja, é possível entender o que foi feito pelo programa para se chegar aquela classificação.

As árvores de decisão podem se tornar enormes e possuem custo computacional alto para se atingir todas as possibilidades. É comum realizar “podas” na árvore para diminuir seu tamanho e melhorar o desempenho, evitando o “*overfitting*”³.

O custo computacional é compensado pela grande velocidade de processamento das novas instâncias, pois estas serão classificadas respeitando as regras construídas com base no treino. Essas regras são uma série de “IF-ELSE” que conduzem a uma classificação da instância apresentada.

Neste trabalho deixamos o algoritmo criar a árvore completa e não realizamos nenhuma “poda” o que gerou algumas árvores de decisão tão grandes que não puderam ser colocadas na análise dos dados.

2.1.3 Naive Bayes

É um algoritmo de classificação supervisionado, que utiliza a teoria da probabilidade de Bayes como base para selecionar a categoria que a nova instância pertencerá. Assim como o KNN, é um algoritmo “Lazy”, ou seja, mantém o banco de dados na memória e compara a nova instância com as instâncias de treino para avaliar a probabilidade da nova instância pertencer a uma classe.

Possui um custo computacional relativamente baixo, uma grade eficiência para dados reais e inteiros, desde que os valores sejam diferentes de zero. Para conjunto de dados que possuam muitos zeros em seus atributos o algoritmo precisa tratar esses dados, pois a probabilidade de qualquer al-

³Situação na qual o algoritmo superajusta sua função de classificação ao conjunto de treino e acaba tendo sua acurácia diminuída nas aplicações dos conjuntos de testes ou na vida real

goritmo com zero será zero, o que compromete a eficácia. Para solucionar o problema é comum acrescentar uma unidade em todos os atributos que serão computados quando um deles é igual a zero.

Muito usado para atributos textuais ou grandes volumes de dados numéricos (inteiros e reais) que não contenham muitos zeros.

2.2 Procedimentos gerais

Procuramos criar um algoritmo único para trabalhar com todos os conjunto de dados, isso não foi possível devido a necessidade de alguns tratamentos de dados especiais que serão descritos na análise de cada conjunto de dados. Mas conseguimos uma rotina de tratamento padrão para os conjunto de dados que nos permitiu atingir os resultados esperados.

No código utilizamos a biblioteca *Panda* [7] para importar os dados do conjunto de dados. Depois separamos os dados em dois grupos: *Valores*, que contém as informações dos atributos que serão comparados e *classes* que são os resultados de classificação do conjunto de dados. O conjunto de *valores* é então normalizado⁴ antes de ser apresentado a cada algoritmo de classificação.

Após a normalização os valores são divididos em “*folds*”, ou partes, que serão usadas para o treino e o teste das funções de classificação. Esta separação é feita pela função apropriada da biblioteca SKLearn [1], que utiliza o sistema de *cross-validation*, ou validação cruzada estratificada, que procura manter em cada *fold* de treino a mesma proporção que há na base de dados original.

Terminada a formação dos *folds* vem a aplicação dos algoritmos. Primeiro aplicamos o *KNN* que é executado para os cinco valores de “K” e tem guardada apenas a média das acurácias do melhor resultado conseguido. Depois aplicamos o algoritmo *Naive Bayes*, guardamos as acurácias obtidas com sua aplicação e lançamos as informações para a formação do gráfico de acurácias. O último algoritmo que aplicamos é o *Decision Tree*, que segue o padrão acima, guardamos a acurácia e informamos para a formação do gráfico; a diferença é que exportamos os diagramas da árvore de decisão para que possam ser visualizados na discussão dos resultados. Essa parte da exportação da árvore de decisão só pode ser realizada no Sistema Operacional Linux, não sendo possível rodar o script no ambiente windows.

Terminada a execução dos algoritmos solicitados pelo professor e de posse das acurácias, construímos o gráfico de comparação entre os três al-

⁴Técnica que permite escrever todos os dados com média zero e variando em unidades de desvio padrão, de modo que valores muito grandes não exerçam grande influência sobre a classificação dos dados, homogeneizando-a.

goritmos para o dataset em questão. Este gráfico mostra o desempenho do algoritmo numa escala percentual. O cálculo da acurácia segue a fórmula 2.2, vista em sala e dada pelo professor.

$$\frac{TP + TN}{P + N}$$

Figura 1: Fórmula da acurácia

Onde, TP e TN são os valores de classificação que foram corretamente atribuídos e P e N formam o total de classes do dataset.

Por fim, exportamos uma tabela que contém todos os valores de acurácia obtidos nos *folds* para a realizarmos o cálculo da média e do desvio padrão e apresentarmos na discussão dos dados realizadas adiante no relatório.

2.3 Conjuntos de dados

Foram utilizados 10 conjuntos de dados, obtidos nos locais indicados pelo professor na apresentação do trabalho e com as descrições que são feitas em cada tópico seguinte:

2.3.1 Iris

Este é talvez o conjunto de dados mais comum em estudos de reconhecimento de padrões na literatura, usado pela primeira vez pelo famoso biólogo evolucionista Ronald Aylmer Fisher [3]. O conjunto de dados contém 3 classes com 50 exemplos de cada classe e com quatro atributos por exemplo, o tamanho e largura de pétalas e sépalas, onde cada classe é uma espécie vegetal do gênero *Iris*.

O dataset Iris pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Iris> e suas informações básicas são:

Tabela 1: Iris Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	150
Atributos	4

2.3.2 Ecoli

O dataset Ecoli pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Ecoli> e suas informações básicas são:

Tabela 2: Ecoli Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	336
Atributos	8

2.3.3 Fertility

O dataset Fertility pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Fertility> e suas informações básicas são:

Tabela 3: Fertility Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	100
Atributos	10

2.3.4 Yeast

O dataset Yeast pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Yeast> e suas informações básicas são:

Tabela 4: Yeast Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	1484
Atributos	8

2.3.5 Planning Relax

O dataset Planning Relax pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Planning+Relax> e suas informações básicas são:

Tabela 5: Planning Relax Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	182
Atributos	13

2.3.6 Habermans Survival

O dataset Haberman's Survival pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Haberman's+Survival> e suas informações básicas são:

Tabela 6: Haberman's Survival Data Set

Característica	Multivariável
Tipo dados	inteiro
Instâncias	306
Atributos	3

2.3.7 Banknote Authentication

O dataset Banknote Authentication pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/banknote+authentication> e suas informações básicas são:

Tabela 7: Banknote Authentication Data Set

Característica	Multivariável
Tipo dados	real
Instâncias	1372
Atributos	5

2.3.8 Breast Cancer Wisconsin

O dataset Breast Cancer Wisconsin pode ser obtido no endereço [https://archive.ics.uci.edu/ml/conjuntodedados/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/conjuntodedados/Breast+Cancer+Wisconsin+(Original)) e suas informações básicas são:

Tabela 8: Breast Cancer Wisconsin Data Set

Característica	Multivariável
Tipo dados	inteiro
Instâncias	699
Atributos	10

2.3.9 Mammographic Mass

O dataset Mammographic Mass pode ser obtido no endereço <https://archive.ics.uci.edu/ml/conjuntodedados/Mammographic+Mass> e suas informações básicas são:

Tabela 9: Mammographic Mass Data Set

Característica	Multivariável
Tipo dados	inteiro
Instâncias	961
Atributos	6

2.3.10 Pima Indians Diabetes

O dataset Pima Indians Diabetes pode ser obtido no endereço <https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes> e suas informações básicas são:

Tabela 10: Pima Indians Diabetes Data Set

Característica	Multivariável
Tipo dados	inteiro e Real
Instâncias	768
Atributos	8

3 Resultados e Discussão

De modo geral, o algoritmo KNN obteve a maior média geral das acurácias e com maior precisão (menor variabilidade entre os resultados obtidos) como pode ser visto na tabela 11.

No entanto, dos 10 conjuntos de dados avaliados, o algoritmo Naive Bayes obteve a maior acurácia média em 7 deles, seguido do KNN que obteve

Tabela 11: Média e desvios gerais de todas as acurácias obtidas

Algoritmo	Média	Desvio Padrão
Árvore de decisão	0.7549212	0.1846360
Knn	0.7872640	0.1733436
Naive Bayes	0.7463543	0.2417319

a melhor acurácia em 3 e o algoritmo de Árvore de Decisão que não teve a melhor acurácia para nenhum conjunto de dados como podemos observar na tabela 12 e na figura 2

Tabela 12: Médias obtidas por conjunto de dados para cada algoritmo

Conjunto de dados	Média		
	decision tree	knn	naive bayes
Banknote Authentication	0.9832223	0.9985401	0.8206231
Breast Cancer	0.9370844	0.9590793	0.9634271
Ecoli	0.7303922	0.7594474	0.8104278
Fertility	0.8100000	0.8600000	0.8300000
Habermans Survival	0.6431183	0.7023656	0.7453763
Iris	0.9266667	0.9200000	0.9466667
Mammographic Mass	0.7614458	0.7819277	0.8096386
Pima Indians Diabetes	0.6914046	0.7460526	0.7551777
Planning Relax	0.5935673	0.6210526	0.6426901
Yeast	0.4723109	0.5241747	0.1395157

Conjunto de dados	Desvio Padrão		
	decision tree	knn	naive bayes
Banknote Authentication	0.01036215	0.003077642	0.06519682
Breast Cancer	0.04473660	0.038628010	0.02506400
Ecoli	0.16417374	0.239384217	0.20628346
Fertility	0.12866839	0.126491106	0.17029386
Habermans Survival	0.14484407	0.098540629	0.07948557
Iris	0.08577893	0.075686162	0.06126244
Mammographic Mass	0.04879146	0.050305113	0.04536553
Pima Indians Diabetes	0.07348156	0.069537673	0.04507926
Planning Relax	0.15234015	0.063453212	0.09191607
Yeast	0.04864499	0.051461378	0.05181031

Para o conjunto de dados Yeast, a baixa acurácia em relação aos outros conjuntos de dados foi observada em outras tentativas também, por exemplo, Horton e Nakai [5] reportam uma acurácia de 55% usando um

modelo de Redes Bayesianas para classificar os exemplos enquanto Nakai e Kaneshisa [9] obtiveram 83%, no entanto, para esse artigo é um precursor desse conjunto de dados, ou seja, os dados usados aqui não são exatamente os usados por estes autores, que ainda contaram com regras pre-determinadas derivadas a partir da consulta a especialistas.

Observamos também na figura 2 que apesar do Naive Bayes se sair bem na maioria dos casos, nos três casos onde ele obteve menor acurácia, banknote authentication, fertility e yeast, dois deles tratam de atributos em sua maioria contínuos, enquanto fertility tem somente classes hierárquicas que já são fornecidas como valores reais. No caso do "Banknote Authentication", trata-se de atributos de imagens, parâmetros das imagens registradas de cheques legítimos e falsificados, assim imperfeições como quando se apaga palavras de um cheque original podem ser detectados. Ghazvini e colaboradores [4] também observaram uma performance inferior do Naive Bayes em relação ao algoritmo de Máquina de vetores de suporte e Redes neurais artificiais, estes autores reportam que seus melhores resultados foram obtidos com rede neural do tipo Multilayer Perceptron.

Apesar da Árvore de decisão ter o pior resultado para 7 dos 10 conjuntos de dados avaliados aqui, seu resultado nunca é tão discrepante dos outros algoritmos, como quando o Naive bayes teve sua acurácia 5 vezes menor que o melhor resultado para o dados Yeast. Um ponto positivo da Árvore de decisão é a facilidade com que o seu resultado pode ser interpretado, mesmo por pessoas leigas quanto ao funcionamento do algoritmo. Ao produzir as Árvores de decisão, como apresentado na figura 4, esse resultado pode ser facilmente interpretado e utilizado por outras pessoas [2]

Nossa avaliação sugere que não é simples determinar o melhor algoritmo independente do conjunto de dados. Mesmo o Naive Bayes, que teve a melhor performance geral, teve uma performance insatisfatória para alguns problemas, como por exemplo no caso de Yeast em que ele foi o pior, muito abaixo da média dos outros dois algoritmos e no caso dos Cheques falsificados, onde uma performance superior, como a obtida pelo Knn pode significar uma grande economia para os bancos. Além disso, para problemas como do Iris, onde estamos tentando classificar espécies vegetais a partir de medidas de flores, usuários finais podem ficar mais satisfeitos com uma Árvore de decisão impressa em um guia de identificação de plantas do que ter a necessidade de usar um computador para conseguir se utilizar de um modelo de classificação criado.

Figura 2: Comparação das acurácias para os três métodos de classificação

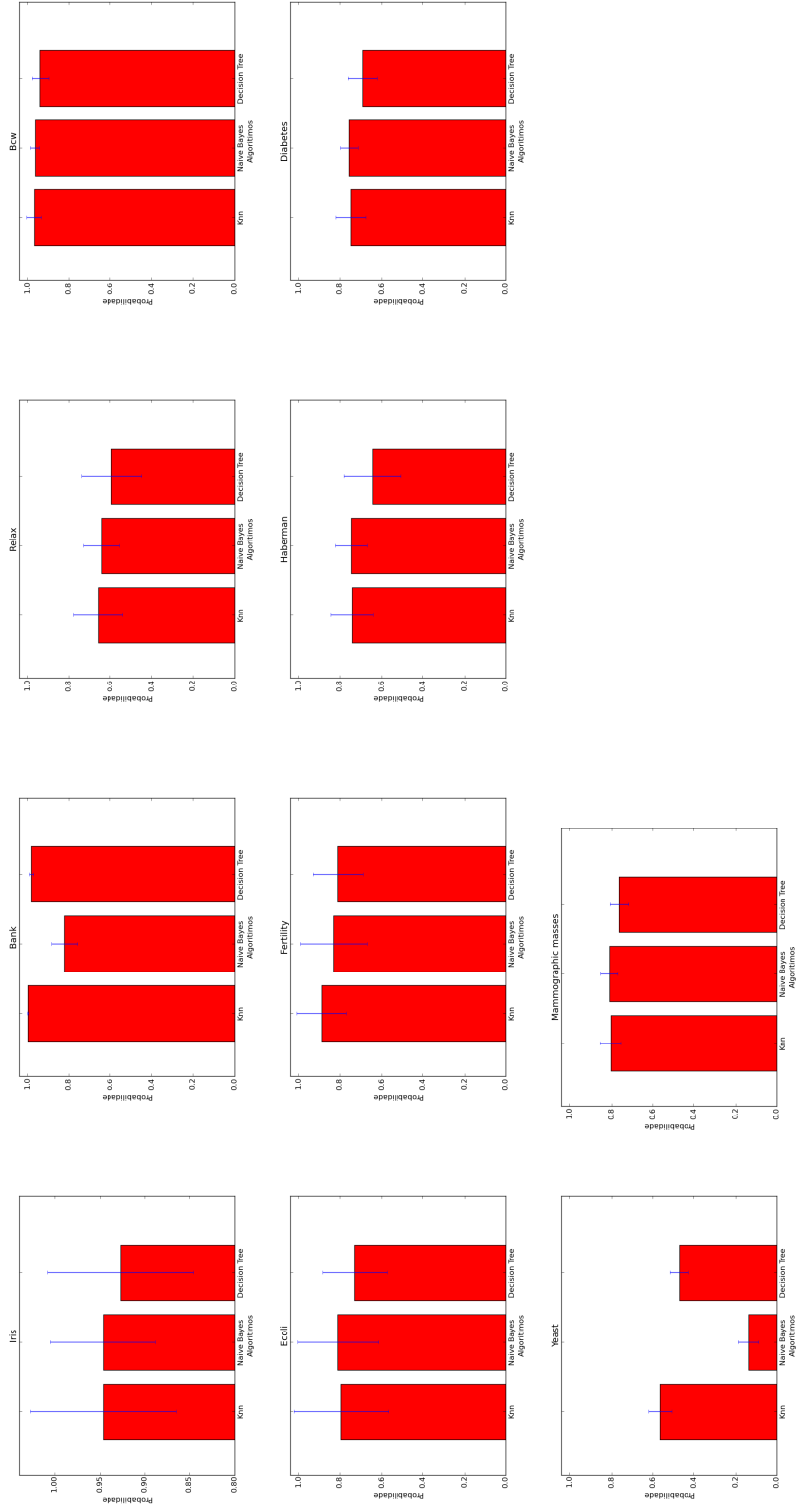


Figura 3: Escolha do número de vizinhos mais próximos

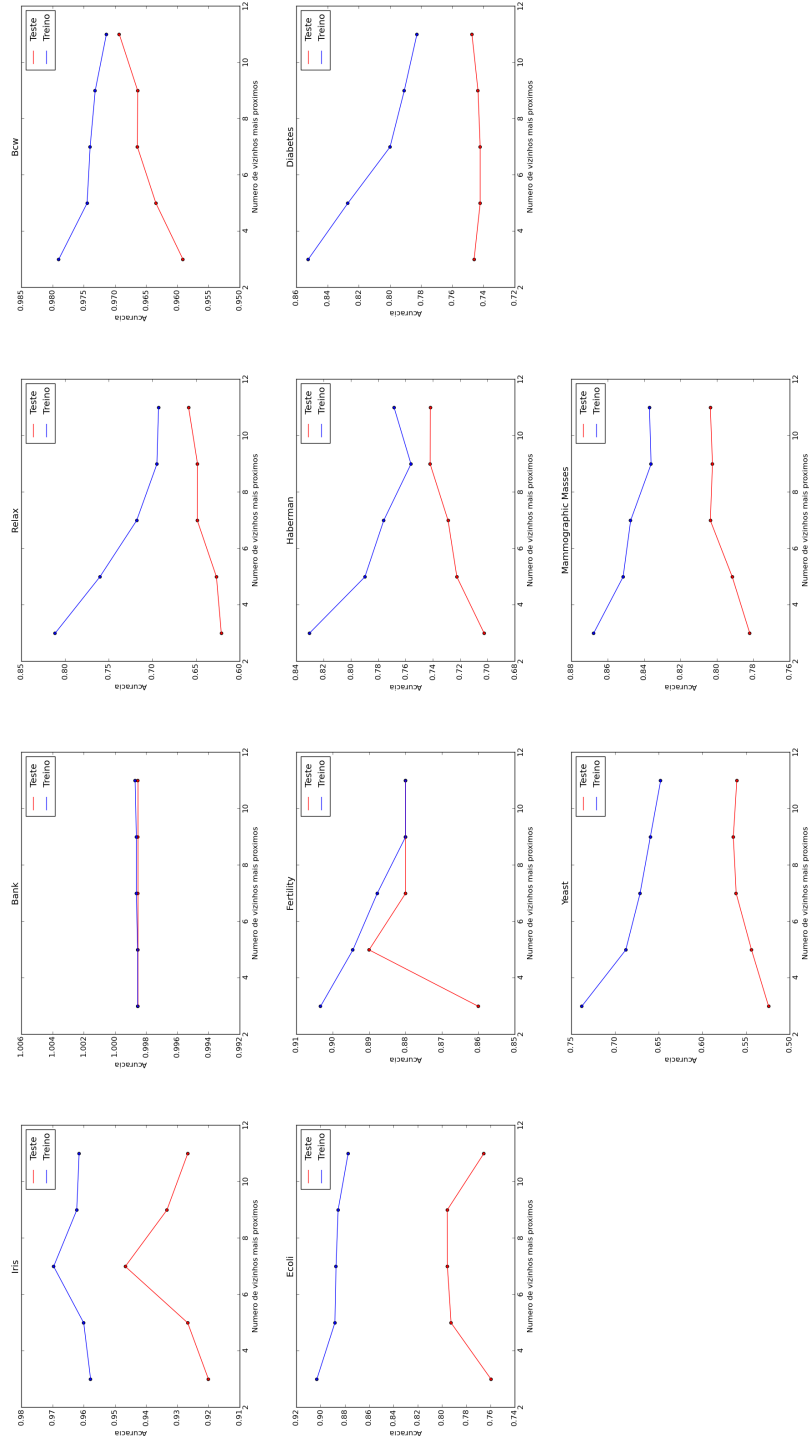
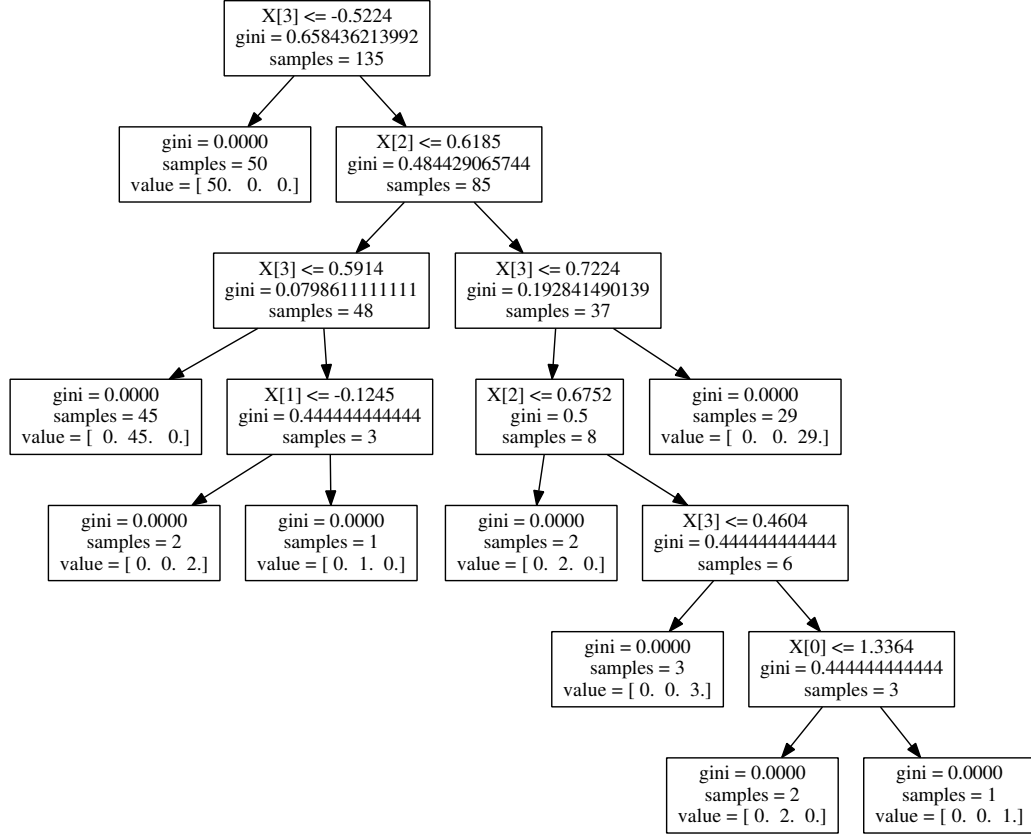


Figura 4: Exemplo de árvore de decisão gerada para o conjunto de dados Iris



Referências

- [1] Scikit-learn: Machine Learning in Python.
- [2] Fernando Berzal, Juan-Carlos Cubero, Fernando Cuenca, and Maria J. Martín-Bautista. On the quest for easy-to-understand splitting rules. *Data & Knowledge Engineering*, 44:31–48, 2003.
- [3] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7:179–188, 1936.
- [4] Anahita Ghazvini, Jamilu Awwalu, and Azuraliza Abu Bakar. Comparative Analysis of Algorithms in Supervised Classification: A Case study of Bank Notes Dataset. *International Journal of Computer Trends and Technology*, 17(1):39–43, 2014.
- [5] Paul Horton and Kenta Nakai. A Probabilistic Classification Localization System for Predicting Sites of Proteins. *Intelligent systems for molecular biology*, 96:109–115, 1996.
- [6] John D. Hunter. Matplotlib: A 2D Graphics Environment. *omputing in Science & Engineering*, 9:90–95, 2007.
- [7] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51–56, 2010.
- [8] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education, 1997.
- [9] Kenta Nakai and Minoru Kanehisa. Expert System for Predicting Protein Localization Sites in Gram-Negative Bacteria. *PROTEINS: Structure, Function, and Genetics*, 11:95–110, 1991.
- [10] Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13:22–30, 2011.