ANÁLISE COMPARATIVA ENTRE OS ALGORITMOS DE CLASSIFICAÇÃO SVM E NAIVE BAYES

COMPARATIVE ANALYSIS BETWEEN SVM AND NAIVE BAYES CLASSIFICATION ALGORITHMS

Bruno Dezorzi[1]
Roberta Vanessa Rojo Parcianello[2]
Matheus Raffael Simon[3]

Junho de 2025

RESUMO

Este artigo apresenta uma análise comparativa entre dois algoritmos supervisionados amplamente utilizados em tarefas de classificação: *Naive Bayes* e *Support Vector Machine (SVM)*. Os experimentos foram conduzidos na linguagem de programação Python com o conjunto de dados *Iris*. O processo incluiu pré-processamento (remoção de *outliers*, engenharia de atributos e normalização), aplicação dos modelos e avaliação por validação cruzada estratificada em cinco dobras, com acurácia como principal métrica. Os resultados mostraram que o *Naive Bayes* obteve acurácia perfeita, enquanto o SVM apresentou desempenho consistente, com média de 96%. A comparação evidencia diferenças de desempenho, bem como vantagens e limitações de cada abordagem frente às características do conjunto analisado.

Palavras-chave: Aprendizado de Máquina; Aprendizado Supervisionado; Naive Bayes; SVM; Classificação de Dados.

ABSTRACT

This article presents a comparative analysis of two widely used supervised learning algorithms for classification tasks: *Naive Bayes* and *Support Vector Machine (SVM)*. Experiments were conducted in Python programming language using the *Iris* dataset. The methodology included preprocessing steps (outlier removal, feature engineering, and normalization), model application, and performance evaluation through stratified 5-fold cross-validation, using accuracy as the main metric. Results showed that *GaussianNB* achieved perfect accuracy, while the SVM delivered consistent performance with an average accuracy of 96%. The comparison highlights not only performance differences but also the strengths and limitations of each approach in relation to the dataset's characteristics.

Keywords: Machine Learning, Supervised Learning, Naive Bayes, SVM, Data Classification.

[1] Faculdade Senac, Cascavel, PR, Brasil. E-mail: bruno.1983@aluno.pr.senac.br
[2] Faculdade Senac, Cascavel, PR, Brasil. E-mail: roberta.parcianello@docente.pr.senac.br
[3] Faculdade Senac, Cascavel, PR, Brasil. E-mail: matheus.simon@docente.pr.senac.br

1 INTRODUÇÃO

O aprendizado de máquina é um campo da inteligência artificial que permite que sistemas computacionais identifiquem padrões e tomem decisões com base em dados, sem a necessidade de programação explícita para cada tarefa. Essa abordagem se baseia na construção de modelos matemáticos que ajustam seus parâmetros a partir de exemplos, melhorando sua performance à medida que recebem mais informações, quanto mais informação e ajustes for disponibilizado para o algoritmo, melhor sua eficácia. O aprendizado de máquina tem sido utilizado em áreas como reconhecimento de imagens, processamento de linguagem natural e sistemas de recomendação, o que mostra seu uso em diferentes contextos.

2 FUNDAMENTAÇÃO TEÓRICA

Aprendizado de máquina é o campo de estudo que possibilita aos computadores a habilidade de aprender sem explicitamente programá-los (SAMUEL, 1959). Os sistemas de aprendizado de máquina podem ser classificados de acordo com a quantidade e tipo de supervisão que recebem durante seu treinamento. Existem quatro categorias principais de aprendizado: supervisionado, não supervisionado, semi- supervisionado e aprendizado por reforço (GÉRON, 2021).

2.1 Tipos de Aprendizagem

Na aprendizagem supervisionada tenta-se prever uma variável dependente a partir de variáveis independentes. Diz-se “supervisionado” porque, no conjunto de dados de treino, a resposta (ou respostas) desejada está contida, ou seja, o resultado desejado do processo de aprendizagem é conhecido (GARZILLO, 2022). Os dados então são rotulados com as previsões ou classes (HONDA H.; YAOHAO, 2017). Os problemas são divididos em Regressão ou Classificação. Exemplo: Utilizar classificação para determinar se um e-mail é um spam.

Com base nos itens de Osisanwo et al. (2017), alguns exemplos de algoritmos de aprendizado supervisionado incluem o K-ésimo vizinho mais próximo (*K-Nearest Neighbors - KNN*), *Naive Bayes*, Regressão Logística (*Logistic Regression*), Máquinas de Vetores de Suporte (*Support Vector Machine - SVM*), Árvore de Decisão (*Decision Tree*) e Redes Neurais (*Neural Networks*).

Diferentemente do aprendizado supervisionado, no aprendizado não supervisionado os dados de treinamento não são rotulados (GÉRON, 2021). A tarefa desse formato de aprendizagem é aprender sozinho, rotulando os dados com base em sua padronização e na identificação de padrões intrínsecos, sem a necessidade de um "professor"ou rótulos previamente definidos.

Como citado em Mahesh (2020), alguns exemplos de algoritmos de aprendizado não supervisionado incluem Análise de Componentes Principais (*Principal Component Analysis - PCA*) e K-ésimo vizinho mais próximo (*K-Nearest Neighbors - KNN*).

Segundo Zhu (2005), o aprendizado semi-supervisionado é uma abordagem específica para classificação. Enquanto classificadores tradicionais exigem dados rotulados (pares de características e rótulos) para serem treinados, o processo de rotular essas instâncias pode ser trabalhoso, caro e demorado,

pois requer a atuação de especialistas. Conforme mencionado por Zhu (2005), os dados não rotulados são mais fáceis de serem coletados, mas suas aplicações práticas ainda são limitadas.

O aprendizado semi-supervisionado, tal qual cita Zhu (2005), resolve essa limitação ao combinar grandes volumes de dados não rotulados com dados rotulados, resultando em classificadores mais eficazes. De acordo com o autor, por demandar menos esforço humano e alcançar maior precisão, essa técnica tem se tornado uma área de grande interesse tanto na pesquisa quanto na aplicação prática.

Diante de um cenário onde a obtenção de dados e variáveis de interesse representam um processo custoso e demorado, precisa-se buscar alternativas para aproveitar as informações disponíveis. O aprendizado semi-supervisionado surge como uma alternativa promissora às abordagens de aprendizado supervisionado e não supervisionado. O objetivo é trabalhar em um cenário com problemas de rotulação parcial dos dados, onde as informações sobre a variável alvo são poucas (ALMEIDA, 2023).

No artigo de Almeida (2023) é descorrido sobre as vantagens da utilização deste algortimo em casos de deteccção de fraudes em transações, as vantagens são:

- Precisão aprimorada;
- Redução de falsos positivos;
- Aproveitamento de dados não rotulados;
- Adaptação a mudanças;
- Controle personalizado.

Com base nos itens de Almeida (2023), alguns exemplos de algoritmos de aprendizado semi-supervisionado incluem SVM Transdutivo (*Transductive SVM*), Modelos Generativos (*Generative Models*) e Autotreinamento (*Self-Training*).

O Aprendizado com Reforço tem o objetivo de melhorar o desempenho final, aumentar a acurácia, no qual nenhuma entrada/saída é fornecida, mas sim *feedbacks* sobre as decisões como um meio de maximizar um sinal de recompensa, levando ao aprendizado de ações desejadas em determinados ambientes (SCHLEDER; FAZZIO, 2021).

2.2 Classificação

Classificação é uma abordagem de mineração de dados (aprendizado de máquina) usada para prever a associação de grupos para instâncias de dados (TERA, 2021). Embora haja uma variedade de técnicas disponíveis para aprendizado de máquina, a classificação é a técnica mais amplamente usada (HONDA H.; YAOHAO, 2017).

De acordo com Osisanwo et al. (2017), o modelo de aprendizado deve aprender (aproximar o comportamento de) uma função que mapeia um vetor (ou uma matriz) para uma das várias classes, analisando diversos exemplos de entrada e saída dessa função. O aprendizado de máquina indutivo é o processo de aprender um conjunto de regras a partir de instâncias (exemplos em um conjunto de treinamento) ou, de forma mais geral, criar um classificador que possa ser utilizado para generalizar a

partir de novas instâncias. Isso significa que um classificador é um método de aprendizado supervisionado que utiliza dados rotulados para aprender e fazer previsões. Dados rotulados são registros que já possuem uma ou mais colunas com características distintas de interesse para quem realiza a análise. Cada registro nesses dados tem um rótulo associado que representa a categoria ou classe à qual pertence. O classificador usa essas informações para identificar padrões e prever a qual classe novos dados não rotulados pertencem.

Um exemplo clássico de dados rotulados é o conjunto de dados Íris[4], que contém amostras de flores com medições de suas características. Cada registro inclui informações sobre o comprimento e a largura das sépalas e pétalas, além da espécie correspondente, indicada na última coluna. A tabela 1 apresenta 10 exemplos desses registros, destacando as características utilizadas para classificação das espécies.

O conjunto de dados Íris[5] é amplamente utilizado nos primeiros estudos em machine learning, pois está disponível na biblioteca scikit-learn e apresenta uma estrutura simples. Suas características bem definidas entre as classes tornam-no ideal para testar e comparar diferentes classificadores de forma intuitiva. Esse *dataset* serve como uma base para compreender o funcionamento dos algoritmos de aprendizado supervisionado na prática.

Tabela 1 – Amostra com dez registros do conjunto de dados Íris

sepal length (cm)	**sepal width (cm)**	**petal length (cm)**	**petal width (cm)**	**species**
6.1	2.8	4.7	1.2	versicolor
5.7	3.8	1.7	0.3	setosa
7.7	2.6	6.9	2.3	virginica
6.0	2.9	4.5	1.5	versicolor
6.8	2.8	4.8	1.4	versicolor
5.4	3.4	1.5	0.4	setosa
5.6	2.9	3.6	1.3	versicolor
6.9	3.1	5.1	2.3	virginica
6.2	2.2	4.5	1.5	versicolor
5.8	2.7	3.9	1.2	versicolor

Fonte: (SCIKIT-LEARN, 2024)

2.3 Os Classificadores

Neste artigo foi implementou-se dois algortimos para realizar a comparação da eficácia: o algoritmo de *Naive Bayes* e *Support Vector Machine (SVM)*.

2.3.1 *Naive Bayes*

O *Naive Bayes* é um classificador linear baseado no teorema de Bayes, que utiliza probabilidades condicionais para determinar a classe de um dado a partir da combinação entre a probabilidade a priori e a verossimilhança da classe. Esse algoritmo é conhecido como um "aprendedor ansioso", pois possui uma rápida capacidade de aprendizado, sendo amplamente utilizado para classificação de texto (NAULAK, 2023).

[4] O site do conjunto de dados se encontra no Apêndice A

[5] Veja mais em (Scikit-learn developers, 2024)

Entre os classificadores probabilísticos ele pressupõe uma independência entre os atributos dos dados. É popular para aplicações de classificação de textos, por exemplo, para identificação de *spams*. Uma vantagem desse modelo é que, por supor uma independência entre os atributos, ele é capaz de fazer o treinamento de um modelo com um número pequeno de amostras, uma dessas vantagens é que o modelo não conseguirá capturar as interações entre os atributos. Esse modelo simples também pode trabalhar com dados que tenham vários atributos. Desse modo, serve como um bom modelo de base (HARRISON, 2019).

De acordo com Harrison (2019) há três principais classes no sklearn, *GaussianNB*, *MultinominalNB* e *BernoulliNB*. A primeira pressupõe uma distribuição gaussiana (atributos contínuos com uma distribuição normal), a segunda é para contadores e geocorrentes discretos, e a terceira, para atributos booleanos discretos.

Segundo Pardo e Nunes (2002), o algoritmo Naive Bayes apresenta um desempenho eficiente em tarefas como classificação de e-mails, Processamento de Linguagem Natural (NPL) e detecção de anomalias, devido à sua simplicidade e baixo custo computacional.

2.3.2 SVM (*Support Vector Machine*)

Uma máquina de vetores de suporte (SVM) é um modelo muito robusto e versátil de aprendizado de máquina, capaz de fazer classificações lineares ou não lineares, de regresão e até mesmo deteccção de *outliers* e são particularmente adaptadas para a classificação de conjunto de dados complexos pequenos ou de médio porte (GÉRON, 2021).

O SVM é um dos algoritmos mais efetivos para Classificação e que também pode ser utilizado para problemas de Regressão, apesar de menos comum. O SVM pode ser aplicado em dados lineares ou não lineares. Apesar de o treinamento dos modelos de SVM costumar ser lento, estes modelos exigem poucos ajustes, tendem a apresentar boa acurácia e conseguem modelar fronteiras de decisão complexas e não lineares. Além disso, são menos propensos a *overfitting* se comparados com outros métodos (ESCOVEDO, 2024).

Essencialmente, o SVM realiza um mapeamento não linear para transformar os dados de treino originais em uma dimensão maior. Nesta nova dimensão, o algoritmo busca pelo hiperplano que separa os dados linearmente de forma ótima. Com um mapeamento apropriado para uma dimensão suficientemente alta, dados de duas classes podem ser sempre separados por um hiperplano. O SVM encontra este hiperplano usando vetores de suporte (exemplos essenciais para o treinamento) e margens, definidas pelos vetores de suporte (ESCOVEDO, 2024).

3 TRABALHOS RELACIONADOS

Diversos estudos aplicam técnicas de classificação ao conjunto de dados Iris, buscando avaliar o desempenho de diferentes algoritmos. Hussain et al. (2020) realizou um treinamento utilizando o algoritmo SVM onde separou 70% de dados para treinamento e 30% para teste. Utilizou *Principal Component Analysis* (PCA) para reduzir a dimensionalidade do dataset de quatro atributos para três. Seus

resultados geraram dois valores, utilizando os parâmetros `c` e `gamma` ele conseguiu uma acurácia de 98.7%, sem utilizar parâmetros o resultado foi de 95.3%.

No estudo de Dani e Ginting (2024), o autor os algoritmos *Naive Bayes* e *Decision Tree* aplicados ao conjunto de dados Iris. A base foi dividida em 67% para treinamento e 33% para teste. Ambos os modelos apresentaram métricas elevadas de desempenho, com destaque para o *Decision Tree*, que obteve acurácia de 100% no treinamento e 98% no teste, enquanto o *Naive Bayes* alcançou 95% e 96%, respectivamente. As métricas de precisão, revocação e *F1-score* permaneceram acima de 90% em todas as classes, indicando resultados consistentes.

No trabalho de Iqbal e Yadav (2020), foi aplicada a metodologia de classificação baseada no algoritmo *Gaussian Naive Bayes*, utilizando o conjunto de dados Iris disponibilizado pelo repositório UCI *Machine Learning Repository*. Os autores estruturaram o experimento realizando a preparação dos dados, visualizações exploratórias e a divisão entre conjuntos de treino e teste e o treinamento utilizando o modelo *Naive Bayes Gaussiano*. Os resultados evidenciaram uma acurácia de aproximadamente 95%, o artigo adota uma abordagem direta e didática, sem explorar o ajuste de hiperparâmetros.

Diferentemente desses trabalhos, o presente estudo realiza uma comparação direta entre SVM com diferentes kernels e Naive Bayes, incorporando etapas de limpeza de dados, seleção de atributos e ajuste de hiperparâmetros, com o objetivo de identificar o modelo mais eficiente para esse cenário de classificação.

4 METODOLOGIA

Os experimentos foram conduzidos utilizando a linguagem de programação Python, com as bibliotecas *pandas*, *scikit-learn*, *matplotlib* e *seaborn*. O dataset utilizado foi o *Iris Dataset*, que contém amostras de três espécies de flores: *setosa*, *versicolor* e *virginica*.

O pré-processamento dos dados incluiu:

- **Mapeamento das classes**: os rótulos numéricos foram convertidos em nomes de espécies para fins de análise exploratória e depois retornado para números com o objetivo de ajustar para o treinamento.
- **Tratamento de *outliers***: realizado por meio do método do Intervalo Interquartil (IQR), substituindo os valores considerados extremos pela média dos valores não *outliers* de cada atributo.
- **Engenharia de atributos**: criação de novas *features*, como a área da pétala (`petal_area`), a área da sépala (`sepal_area`), a proporção entre o comprimento e a largura da pétala (`petal_prop`) e da sépala (`sepal_prop`).
- **Normalização**: realizada com *StandardScaler*, ajustando todas as variáveis para média zero e desvio padrão um.

A base de dados foi dividida em três partes: treino (72.25%), validação (12.75%) e teste (15%). O *DummyClassifier*, configurado com a estratégia `most_frequent`, foi utilizado como modelo *baseline*.

Os modelos principais foram:

- **Support Vector Machine (SVM)**: foram testados os *kernels* `linear`, `poly`, `rbf` e `sigmoid`. O melhor desempenho foi obtido com o kernel `linear`, o qual foi posteriormente refinado com os parâmetros `C = 0.5` e `class_weight = 'balanced'`.
- **Naive Bayes**: avaliou-se o *GaussianNB* e o *BernoulliNB*. O *GaussianNB* teve sua performance otimizada utilizando *GridSearchCV*, ajustando o parâmetro `var_smoothing` com valores logarítmicos entre 10^{-11} e 10^{-1}.

A validação dos modelos foi realizada com *cross-validation* de 5 *folds*, utilizando a acurácia como métrica principal.

5 RESULTADOS

Os resultados obtidos nos experimentos foram os seguintes:

- **DummyClassifier**: apresentou acurácia de 0.35 na validação e 0.26 no teste, refletindo a limitação de modelos baseados no parâmetro de estratégia `most_frequent`.
- **SVM com diferentes kernels**:
  - `linear`: acurácia média de validação cruzada de 0.96 ($\pm$ 0.04), com 0.95 na validação e 0.96 no teste.
  - `poly`: acurácia média de 0.88 ($\pm$ 0.09).
  - `rbf`: acurácia média de 0.94 ($\pm$ 0.03).
  - `sigmoid`: acurácia média de 0.94 ($\pm$ 0.05).
- **Naive Bayes**:
  - *GaussianNB*: acurácia média de 0.93 ($\pm$ 0.02) na validação cruzada. Após ajuste do parâmetro `var_smoothing` com `GridSearchCV`, a acurácia alcançou 1.00 tanto na validação quanto no teste.
  - *BernoulliNB*: obteve acurácia média inferior, de 0.82 ($\pm$ 0.05).
- **SVM ajustado**: o modelo final com kernel `linear`, `C = 0.5` e `class_weight = 'balanced'` alcançou acurácia de 0.95 na validação e 0.96 no teste.

Os valores apresentados com o símbolo $\pm$ representam o desvio padrão das acurácias obtidas nas diferentes divisões de validação cruzada (5-fold cross-validation). Esse desvio indica o grau de variação dos desempenhos do modelo entre os folds, sendo uma medida de sua estabilidade. Quanto menor o desvio padrão, mais consistente é o modelo em diferentes subconjuntos dos dados.

Esses resultados mostram que, embora o *GaussianNB* ajustado tenha alcançado acurácia perfeita, há um possível indício de *overfitting*, já que o desempenho foi idêntico na validação e no teste. Essa suspeita é reforçada pelo fato de o *Iris Dataset* ser um conjunto de dados pequeno e bem estruturado, com classes relativamente balanceadas e baixa complexidade, o que pode favorecer modelos mais simples a memorizar os padrões. Por outro lado, o modelo *SVM* com kernel `linear` demonstrou excelente desempenho, com resultados robustos e consistentes, o que o torna uma alternativa altamente confiável para aplicação em dados reais.

6 CONCLUSÃO

Os experimentos realizados demonstraram que tanto o classificador *SVM* quanto o *Naive Bayes* são capazes de alcançar altos níveis de acurácia no *Iris Dataset*, especialmente após ajustes criteriosos de seus hiperparâmetros. Apesar de o *GaussianNB* ter atingido desempenho perfeito após otimização, esse resultado levanta suspeitas de *overfitting*, principalmente devido à simplicidade e ao tamanho reduzido do conjunto de dados.

O modelo *SVM* com kernel `linear`, por sua vez, apresentou resultados robustos e estáveis mesmo antes de ajustes finos, confirmando sua adequação para problemas lineares bem definidos, como é o caso do conjunto de dados utilizado. Sua consistência nos diferentes subconjuntos de validação cruzada indica maior generalização, tornando-o uma escolha confiável em contextos reais com dados similares.

Portanto, conclui-se que, embora o *GaussianNB* otimizado tenha alcançado os melhores números em termos absolutos, o *SVM linear* oferece um equilíbrio superior entre desempenho e estabilidade. Além disso, este trabalho reforça a importância do pré-processamento cuidadoso e da engenharia de atributos como etapas fundamentais para o sucesso de qualquer modelo de aprendizado de máquina, mesmo em bases de dados consideradas simples.

Referências

ALMEIDA, M. *Machine Learning: o que é aprendizado semi-supervisionado*. 2023. <https://www.alura.com.br/artigos/machine-learning-aprendizado-semi-supervisionado?srsltid=AfmBOorhmQOk9xfK_LzB6sE88sD2WMnRfJgY0loEOawtqq91jcTuReLe>. Acesso em: 20 nov. 2024.

DANI, Y.; GINTING, M. A. Comparison of iris dataset classification with gaussian naïve bayes and decision tree algorithms. *International Journal of Electrical & Computer Engineering (2088-8708)*, v. 14, n. 2, 2024.

ESCOVEDO, T. *Introdução à Estatística para Ciência de Dados: Da exploração dos dados à experimentação contínua com exemplos de código em Python e R*. [S.l.]: Aovs Sistemas de Informática Ltda., 2024.

GARZILLO, M. J. W. *Classificação de tumores cerebrais com algoritmos de machine learning*. Tese (Doutorado) — Instituto Politécnico de Lisboa, Escola Superior de Tecnologia da Saúde de Lisboa, 2022.

GÉRON, A. Mãos à obra: aprendizado de máquina com scikit-learn. *Keras & TensorFlow: Conceitos, ferramentas e técnicas para a construção de sistemas inteligentes.*, 2021.

HARRISON, M. *Machine Learning Pocket Reference*. [S.l.]: O'Reilly Media, Inc., 2019.

HONDA H., F. M.; YAOHAO, P. *Os Três Tipos de Aprendizado de Máquina*. 2017. LAMFO - Laboratório de Aprendizado de Máquina em Finanças e Organizações. Obtido em 10 de setembro de 2021.

HUSSAIN, Z. F. et al. A new model for iris data set classification based on linear support vector machine parameter's optimization. *International Journal of Electrical and Computer Engineering*, IAES Institute of Advanced Engineering and Science, v. 10, n. 1, p. 1079, 2020.

IQBAL, Z.; YADAV, M. Multiclass classification with iris dataset using gaussian naive bayes. *International Journal of Computer Science and Mobile Computing*, p. 27–35, 2020.

MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, v. 9, n. 1, p. 381–386, 2020.

NAULAK, C. A comparative study of naive bayes classifiers with improved technique on text classification. *Authorea Preprints*, Authorea, 2023.

OSISANWO, F. et al. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, v. 48, n. 3, p. 128–138, 2017.

PARDO, T. A. S.; NUNES, M. d. G. V. Aprendizado bayesiano aplicado ao processamento de línguas naturais. 2002.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, IBM, v. 3, n. 3, p. 210–229, 1959.

SCHLEDER, G. R.; FAZZIO, A. Machine learning na física, química, e ciência de materiais: Descoberta e design de materiais. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 43, n. Suppl 1, p. e20200407, 2021.

SCIKIT-LEARN. *scikit-learn: Machine Learning in Python*. 2024. <https://scikit-learn.org/stable/>. Acesso em: 23 ago. 2024.

Scikit-learn developers. *The Iris Dataset Example — Scikit-learn Documentation*. 2024. Acesso em: 23 ago. 2024. Disponível em: <https://scikit-learn.org/1.4/auto_examples/datasets/plot_iris_dataset.html>.

TERA, R. *O que é Machine Learning: o guia para entender aprendizado de máquina*. 2021. <https://blog.somostera.com/data-science/machine-learning>. Acesso em: 05 dez. 2024.

ZHU, X. J. Semi-supervised learning literature survey. *University of Wisconsin-Madison*, University of Wisconsin-Madison Department of Computer Sciences, 2005.

APÊNDICE A – Conjunto de Dados e Repositório

O conjunto de dados bruto pode ser encontrado no site (Scikit Learn - Íris Dataset) ou tratado em meu repositório no Github (Repositório no Github).