Comparação de Nove Algoritmos Clássicos de Aprendizado de Máquina em Conjuntos de Dados de Referência

Saulo Pereira da Silva

FACOM – Faculdade de Computação

Universidade Federal de Mato Grosso do Sul (UFMS)

CEP 79070-900 – Campo Grande – MS - Brasil

{pereira,saulo}@ufms.br

Resumo

- Este estudo avaliou comparativamente o desempenho de diversos algoritmos de aprendizado de máquina em múltiplos conjuntos de dados
- Investigou a influência das características intrínsecas dos dados no desempenho do modelo
- Resultados demonstram que conjuntos de dados com menor complexidade apresentam desempenho superior e mais consistente
- Conjuntos de dados mais complexos exibem maior variabilidade no desempenho
- A escolha do algoritmo ideal depende da complexidade dos dados, recursos computacionais e requisitos de precisão

Palavras-chave: acurácia, árvores de decisão, Bayes, KNN, perceptron, random forest, regressão, redes neurais artificiais, SVM

https://github.com/saulopereira2018/comparacao_algoritmos

Introdução

Problema

A seleção de algoritmos de aprendizado de máquina adequados é uma tarefa crítica em problemas reais

Desafio

O desempenho pode variar conforme o domínio e as características dos dados

Objetivo

Comparar o desempenho de nove algoritmos clássicos em nove conjuntos de dados públicos amplamente utilizados na literatura

Metodologia

Uso da métrica de acurácia e testes estatísticos para reforçar a robustez das conclusões

Trabalhos Relacionados

Caruana e Niculescu-Mizil (2006)

Exploraram a performance de algoritmos em problemas de classificação supervisionada

Demšar (2006)

Argumenta que para comparações estatisticamente válidas entre classificadores, é fundamental o uso de:

- Teste de Friedman
- Análises post-hoc

Fernandes et al. (2021)

Reforçam esse posicionamento ao aplicar os testes em diferentes domínios práticos

Algoritmos Avaliados

1. Árvores de Decisão
DecisionTreeClassifier

2. Regressão Linear
LinearRegression

3. Regressão Logística
LogisticRegression

4. Redes Neurais Artificiais
MLPClassifier

5. Perceptron
Perceptron

6. SVM
SVC

7. Classificadores Bayesianos
GaussianNB

8. KNN
KNeighborsClassifier

9. Random Forest
RandomForestClassifier

Biblioteca utilizada: *scikit-learn 1.4.2*

Conjuntos de Dados

1. Iris
Classificação de flores

2. Wine
Tipos de vinho

3. Breast Cancer Wisconsin
Diagnóstico de câncer

4. Digits
Reconhecimento de dígitos

5. Diabetes
Predição de diabetes

6. Heart Disease
Doença cardíaca

7. Parkinson
Doença de Parkinson

8. Titanic
Sobrevivência no Titanic

9. Bank Marketing
Marketing bancário

Fontes: *UCI Machine Learning Repository e scikit-learn*

Metodologia Experimental

Validação Cruzada

Estratificada com 10 folds, mantendo proporção das classes

Normalização

Atributos normalizados quando aplicável

Métrica Principal

Acurácia média para classificação

Análise Estatística

- Teste de Friedman
- Teste post-hoc de Nemenyi

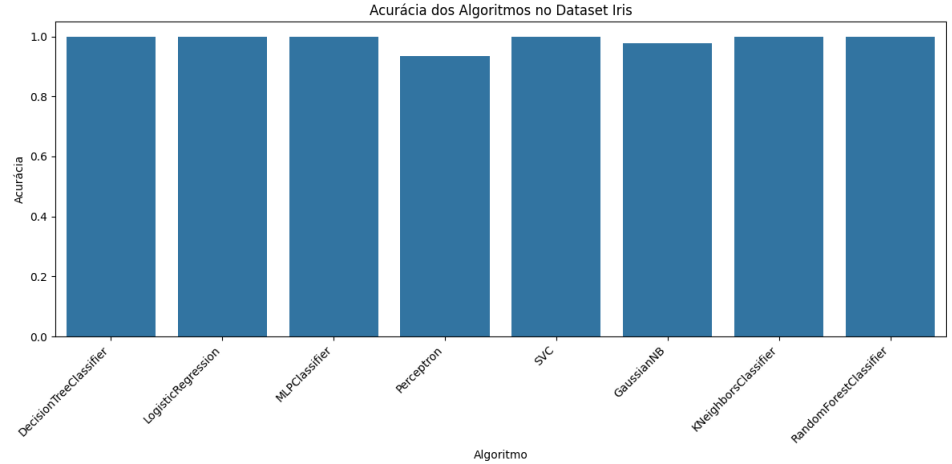
Ferramentas

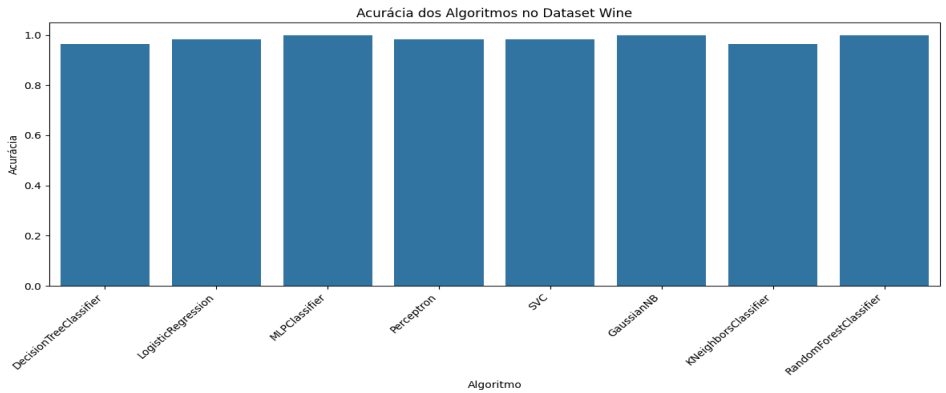
scikit-learn, pandas, seaborn, matplotlib

Principais Resultados - Datasets Simples

Iris

- Maioria dos algoritmos: **acurácia 1.00**
- GaussianNB: 0.978
- Perceptron: 0.933
- *Dataset relativamente simples e linearmente separável*



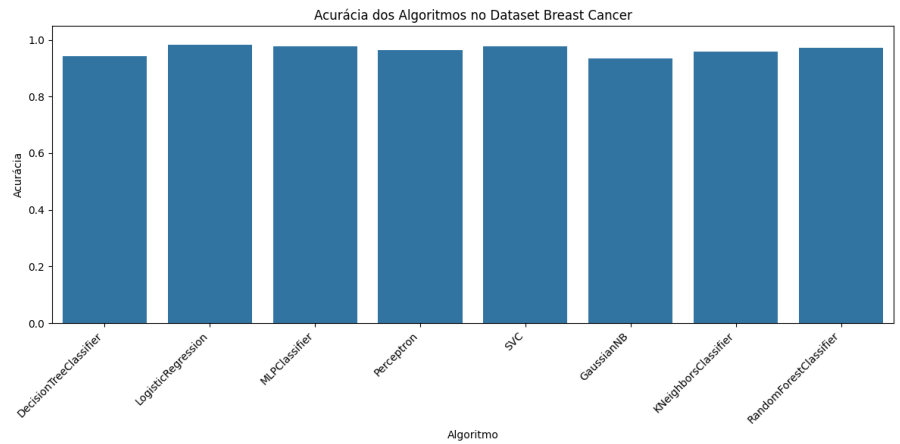


Wine

- GaussianNB e RandomForest: **acurácia 1.00**
- Logistic Regression, MLP, Perceptron, SVC: 0.981
- Decision Tree e KNN: 0.963
- *Classes bem definidas, boa separação*

Breast Cancer

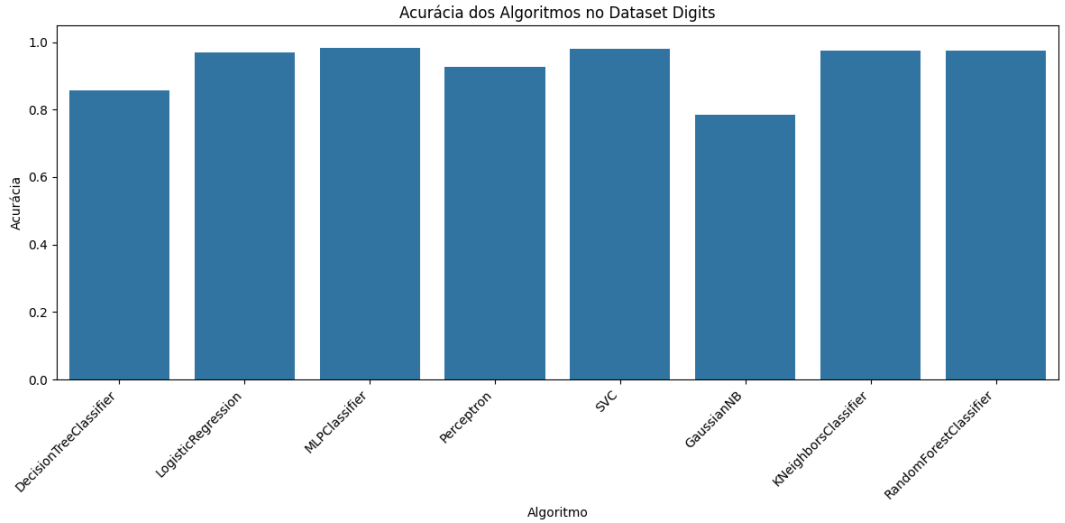
- Logistic Regression: **0.982**
- MLP e SVC: 0.977
- Maioria acima de 0.93
- *Boa separabilidade, complexidade moderada*

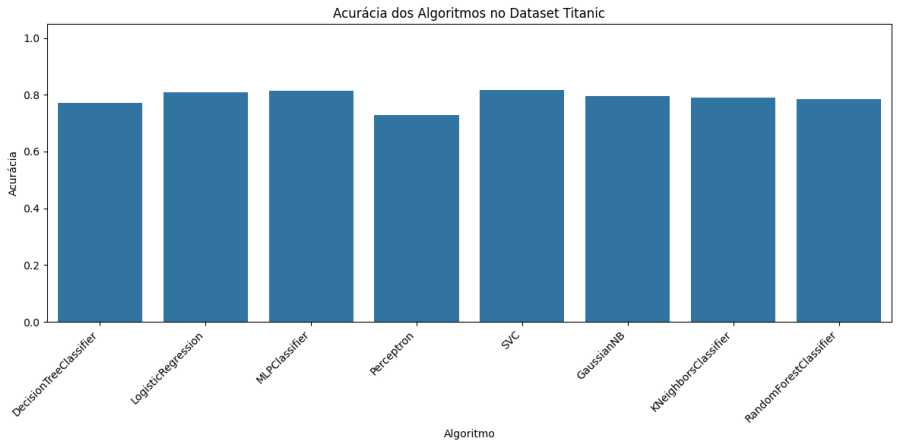


Principais Resultados - Datasets Complexos

Digits

- SVC: **0.980**
- MLP e KNN: 0.976
- Decision Tree: 0.865
- GaussianNB: 0.783
- *Maior variação, modelos sofisticados se destacam*



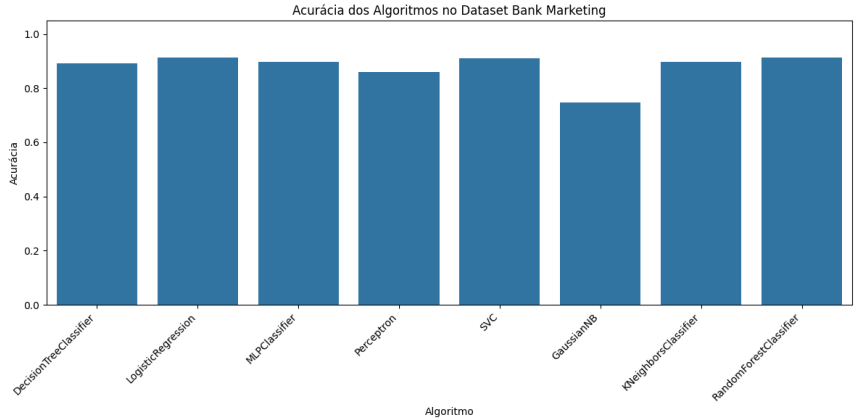


Titanic

- MLP: **0.825**
- SVC: 0.817
- Logistic Regression: 0.810
- Perceptron: 0.728
- *Complexidade alta, variáveis categóricas*

Bank Marketing

- RandomForest: **0.914**
- Logistic Regression: 0.912
- SVC: 0.909
- GaussianNB: 0.748
- *Performance geral boa, GaussianNB com dificuldade*



Análise por Algoritmo

🏆 Melhores Desempenhos

- **SVM (SVC):** Excelente em datasets complexos
- **Random Forest:** Consistente e robusto
- **MLP:** Boa performance em dados não-lineares
- **Logistic Regression:** Confiável em vários cenários

📊 Observações Importantes

- Algoritmos simples (Perceptron, Naive Bayes) competitivos em contextos específicos
- Datasets simples: pouca diferença entre algoritmos
- Datasets complexos: algoritmos sofisticados se destacam
- Trade-off entre precisão e custo computacional

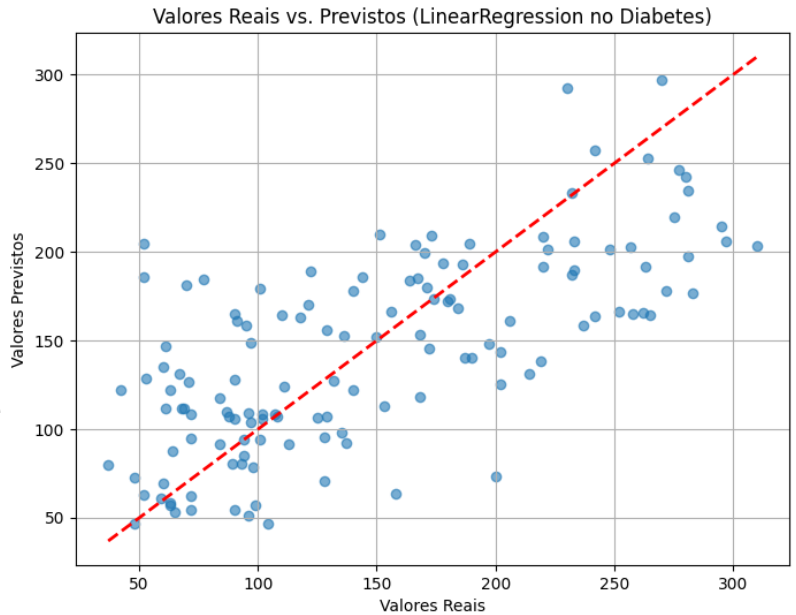
Visualizações dos Resultados

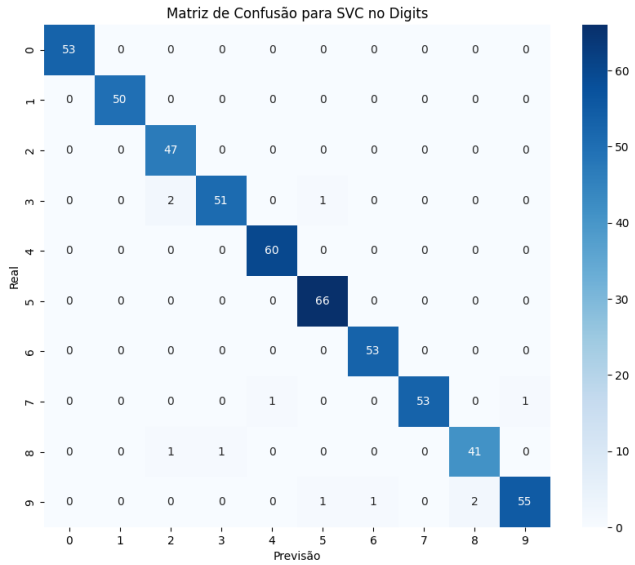
Diagrama de Dispersão - Dataset Diabetes

- Regressão Linear aplicada ao dataset Diabetes
- MSE (Mean Squared Error): 2821.751
- Tendência positiva capturada pelo modelo
- Dispersão indica variação nas previsões individuais
- Ajuste razoável mas com erros consideráveis

Matriz de Confusão - SVC no Dataset Digits

- Diagonal principal com valores altos = classificações corretas
- Pouquíssimos erros de classificação
- Desempenho excelente do SVC
- Demonstra eficácia em reconhecimento de dígitos manuscritos





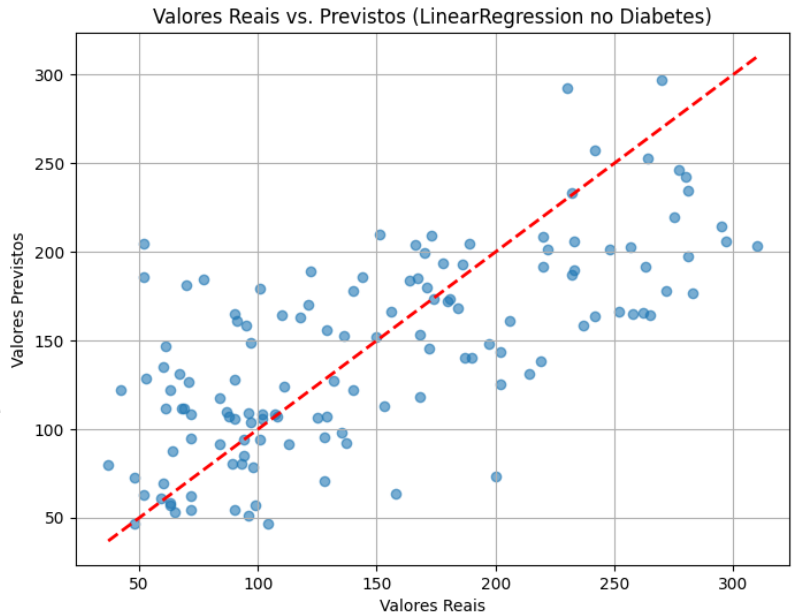
Visualizações dos Resultados

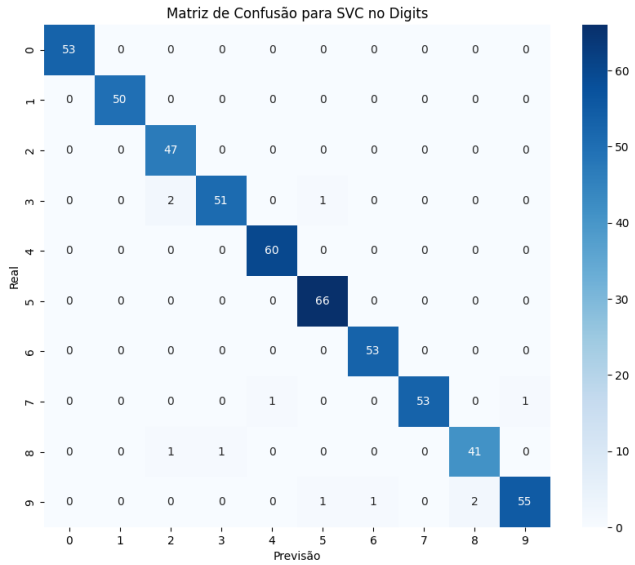
Diagrama de Dispersão - Dataset Diabetes

- Regressão Linear aplicada ao dataset Diabetes
- MSE (Mean Squared Error): 2821.751
- Tendência positiva capturada pelo modelo
- Dispersão indica variação nas previsões individuais
- Ajuste razoável mas com erros consideráveis

Matriz de Confusão - SVC no Dataset Digits

- Diagonal principal com valores altos = classificações corretas
- Pouquíssimos erros de classificação
- Desempenho excelente do SVC
- Demonstra eficácia em reconhecimento de dígitos manuscritos





Conclusões

🎯 Descobertas Principais

- Datasets simples: desempenho excelente e consistente
- Datasets complexos: maior variabilidade nos resultados
- Algoritmos sofisticados se destacam em cenários complexos
- Algoritmos simples competitivos em contextos específicos

⚖️ Fatores Decisivos

- Natureza e complexidade dos dados
- Recursos computacionais disponíveis
- Requisitos de precisão da aplicação
- Necessidade de interpretabilidade

🔮 Trabalhos Futuros

- Análise de tempo de treinamento
- Interpretabilidade dos modelos
- Performance com dados ruidosos ou ausentes
- Aproximação das exigências do mundo real

Recomendações Práticas

🔍 Para Datasets Simples e Pequenos

- Logistic Regression ou SVM
- Rápidos e eficientes
- Boa interpretabilidade

🌐 Para Datasets Complexos e Grandes

- Random Forest ou MLP
- Melhor capacidade de modelagem
- Maior robustez a ruídos

⚡ Para Aplicações em Tempo Real

- KNN ou Naive Bayes
- Treinamento rápido
- Baixo custo computacional

🎯 Para Máxima Precisão

- Ensemble methods (Random Forest)
- SVM com kernel apropriado
- Fine-tuning de hiperparâmetros

Agradecimentos

FACOM - UFMS

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

🔗 Código Fonte Disponível

github.com/saulopereira2018/comparacao_algoritmos.git

📧 Contato

{pereira,saulo}@ufms.br

Obrigado!