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

Abstract. *This study comparatively evaluated the performance of several machine learning algorithms across multiple datasets, investigating the influence of the intrinsic characteristics of the data on model performance. The results demonstrate that less complex datasets exhibit superior and more consistent performance among the algorithms, while more complex datasets exhibit greater variability in performance. The choice of the ideal algorithm is, therefore, dependent on data complexity, available computational resources, and the application's precision requirements.*

Resumo*. Este estudo avaliou comparativamente o desempenho de diversos algoritmos de aprendizado de máquina em múltiplos conjuntos de dados, investigando a influência das características intrínsecas dos dados no desempenho do modelo. Os resultados demonstram que conjuntos de dados com menor complexidade apresentam desempenho superior e mais consistente entre os algoritmos, enquanto conjuntos de dados mais complexos exibem maior variabilidade no desempenho. A escolha do algoritmo ideal é, portanto, dependente da complexidade dos dados, dos recursos computacionais disponíveis e dos requisitos de precisão da aplicação.*

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

1. Introdução

A seleção de algoritmos de aprendizado de máquina adequados é uma tarefa crítica em problemas reais, dado que o desempenho pode variar conforme o domínio e as características dos dados. Este artigo compara o desempenho de nove algoritmos clássicos em nove conjuntos de dados públicos amplamente utilizados na literatura, com o uso da métrica de acurácia e testes estatísticos para reforçar a robustez das conclusões

2. 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 testes como o de Friedman e análises post-hoc. Estudos recentes, como o de Fernandes et al. (2021), reforçam esse posicionamento ao aplicar os testes em diferentes domínios práticos.

3. Metodologia Experimental

http:// https://youtu.be/dUjCb74e4rI

3.1 Algoritmos Avaliados

Nesta seção, descrevem-se os algoritmos de aprendizado de máquina utilizados nos experimentos. Todos foram implementados utilizando a biblioteca **scikit-learn**.

1. **Árvores de Decisão** (*DecisionTreeClassifier* da *scikit-learn*): algoritmos baseados em uma estrutura de árvore, que realizam divisões sucessivas nos dados com base em critérios como Gini ou Entropia.

2. **Regressão Linear** (*LinearRegression* da *scikit-learn*): método supervisionado usado para prever valores contínuos a partir de variáveis independentes.

3. **Regressão Logística** (*LogisticRegression* da *scikit-learn*): técnica de classificação que estima a probabilidade de uma instância pertencer a uma determinada classe.

4. **Redes Neurais Artificiais** (*MLPClassifier* da *scikit-learn*): modelo de aprendizado profundo com múltiplas camadas ocultas, capaz de capturar padrões não lineares nos dados.

5. **Perceptron** (*Perceptron* da *scikit-learn*): uma forma simplificada de rede neural para problemas de classificação binária.

6. **SVM** (*SVC* da *scikit-learn*): algoritmos que buscam encontrar o hiperplano ótimo que separa as classes de forma máxima.

7. **Classificadores Bayesianos** (*GaussianNB* da *scikit-learn*): baseiam-se no Teorema de Bayes e assumem uma distribuição normal para os atributos.

8. **KNN** (*KNeighborsClassifier* da *scikit-learn*): algoritmo baseado em instâncias que classifica uma amostra com base na maioria das classes dos seus vizinhos mais próximos.

9. **Random Forest** (*RandomForestClassifier* da *scikit-learn*): técnica de ensemble que constrói múltiplas árvores de decisão e agrega seus resultados para melhorar a acurácia e reduzir o overfitting.

Todos os experimentos foram conduzidos utilizando a versão **scikit-learn 1.4.2**.

3.1.1 Hiperparâmetros Utilizados

A escolha dos hiperparâmetros dos algoritmos de aprendizado de máquina pode impactar significativamente seu desempenho. Para este estudo comparativo, optou-se por utilizar os valores padrão definidos pela biblioteca scikit-learn para a maioria dos algoritmos. Essa abordagem permite uma comparação baseada nas configurações "out-of-the-box" dos modelos, fornecendo uma linha de base para seu comportamento em diferentes conjuntos de dados.

No entanto, para alguns algoritmos, foram feitos ajustes mínimos nos hiperparâmetros para garantir a convergência e a estabilidade do treinamento, especialmente em conjuntos de dados com maior complexidade ou número de iterações. Por exemplo:

LogisticRegression: O parâmetro max_iter foi aumentado para 1000 para permitir que o algoritmo tivesse iterações suficientes para convergir, evitando avisos de não convergência.

MLPClassifier: O parâmetro hidden_layer_sizes foi ajustado (ex: (10,), (20,), (50,) ou (100,) dependendo do conjunto de dados) e max_iter foi aumentado para 1000, buscando um equilíbrio inicial entre complexidade do

modelo e o tamanho do conjunto de dados.

Perceptron: O parâmetro max_iter foi aumentado para 1000 e tol (tolerância para o critério de parada) foi definido como 1e-3 para melhorar a estabilidade em algumas execuções.

É importante ressaltar que não foi realizado um ajuste fino (fine-tuning) exaustivo de hiperparâmetros (como Grid Search ou Random Search). Essa seria uma etapa subsequente em um estudo mais aprofundado, que poderia levar a melhorias no desempenho de cada algoritmo. No contexto deste trabalho, o foco é a comparação de algoritmos com configurações que representam um ponto de partida comum e razoável.

3.2 Conjuntos de Dados

Foram utilizados os seguintes conjuntos de dados do repositório UCI e scikit-learn:

1. Iris;
2. Wine;
3. Breast Cancer Wisconsin;
4. Digits;
5. Diabetes;
6. Heart Disease;
7. Parkinson;
8. Titanic (Kaggle), e;
9. Bank Marketing.

3.3 Procedimentos

Para garantir uma avaliação robusta e imparcial dos algoritmos, adotou-se uma abordagem metodológica padronizada, com o objetivo de mitigar vieses e assegurar a comparabilidade entre os resultados obtidos. O processo experimental foi estruturado com base em técnicas consagradas na literatura de aprendizado de máquina, utilizando a biblioteca scikit-learn para a implementação dos algoritmos e procedimentos. na literatura de aprendizado de máquina. A divisão dos dados foi realizada por meio de validação cruzada estratificada com 10 folds (partições ou subconjuntos nos quais o conjunto de dados é dividido durante o processo de validação cruzada), assegurando a manutenção da proporção das classes em todas as partições. Sempre que aplicável, os atributos foram normalizados para evitar distorções decorrentes das diferentes escalas nas variáveis.. A métrica de desempenho adotada foi a acurácia média, amplamente utilizada em tarefas de classificação supervisionada. Para análise estatística dos resultados, empregou-se o teste de Friedman, que permite verificar a existência de diferenças significativas no desempenho dos algoritmos avaliados. Quando identificadas tais diferenças, aplicou-se o teste post-hoc de Nemenyi, a fim de determinar quais algoritmos apresentaram desempenho estatisticamente superior. Essa combinação de procedimentos experimentais confere maior rigor e confiabilidade à análise comparativa realizada.

3.4 Ferramentas

Para a realização dos experimentos, foram utilizados algoritmos de classificação e regressão disponíveis na biblioteca scikit-learn (PEDREGOSA et al., 2011), juntamente com bibliotecas auxiliares como pandas (PANDAS DEVELOPMENT TEAM, 2020), seaborn (WASKOM, 2021) e matplotlib (HUNTER, 2007). Os dados foram obtidos de conjuntos clássicos disponibilizados pela biblioteca scikit-learn e pelo UCI Machine Learning Repository.

http:// https://youtu.be/dUjCb74e4rI

3.5 Resultados e Discussão

A Tabela 1 do Apêndice apresenta uma comparação abrangente do desempenho de diferentes algoritmos de aprendizado de máquina em diversos conjuntos de dados. Este estudo detalha o processo de avaliação, desde o pré-processamento dos dados até a análise das métricas de desempenho. É importante notar que o código-fonte completo e os scripts utilizados para gerar esses resultados estão disponíveis publicamente no repositório GitHub em https://github.com/saulopereira2018/comparacao_algoritmos.git. A seguir, há a análise de cada coluna da tabela em detalhes, fornecendo insights sobre os experimentos e seus respectivos resultados.

- **Dataset:** Esta coluna lista os conjuntos de dados utilizados nos experimentos. Cada conjunto de dados representa um problema de aprendizado de máquina diferente. Por exemplo, "Iris" é um conjunto de dados clássico para classificação de flores, "Wine" contém dados sobre diferentes tipos de vinho, e "Breast Cancer" possui informações para diagnóstico de câncer de mama.
- **Algoritmo:** Esta coluna identifica o algoritmo de aprendizado de máquina utilizado para treinar um modelo nos dados. Os algoritmos listados incluem: **Decision Tree Classifier:** Um algoritmo que cria uma árvore de decisões para classificar os dados; **Logistic Regression:** Um algoritmo linear usado para problemas de classificação; **MLP Classifier:** Um classificador de rede neural multicamadas; **Perceptron:** Um algoritmo linear simples para classificação binária; **SVC:** Máquina de Vetores de Suporte, um algoritmo que encontra o melhor hiperplano para separar diferentes classes; **GaussianNB:** Classificador Naive Bayes Gaussiano, um algoritmo probabilístico que assume que os recursos seguem uma distribuição gaussiana; **KNeighbors Classifier:** Classificador K-Vizinhos Mais Próximos, que classifica um ponto com base na classe da maioria de seus vizinhos, e; **Random Forest Classifier:** Um algoritmo de conjunto que combina várias árvores de decisão.
- **Tipo da Tarefa:** Esta coluna especifica o tipo de tarefa de aprendizado de máquina que está sendo realizada, que neste caso é "classification" (classificação). Isso significa que o objetivo dos algoritmos é prever a categoria ou classe a que um determinado ponto de dados pertence.
- **Acurácia:** Esta coluna mostra a acurácia do modelo, que é a proporção de previsões corretas feitas pelo modelo em relação ao número total de previsões. É uma métrica comum para avaliar o desempenho de modelos de classificação.
- **Precision (Weighted Avg):** Precisão é a proporção de verdadeiros positivos (instâncias previstas como positivas que realmente são positivas) em relação ao total de instâncias previstas como positivas. A média ponderada é usada para levar em conta o desbalanceamento de classes, fornecendo uma medida geral de precisão em todas as classes.
- **Recall (Weighted Avg):** Recall é a proporção de verdadeiros positivos em relação ao total de instâncias que realmente são positivas. A média ponderada também é usada aqui para lidar com o desbalanceamento de classes, indicando a capacidade do modelo de encontrar todas as instâncias relevantes.
- **F1-Score (Weighted Avg):** O F1-score é a média harmônica da precisão e do recall. Ele fornece uma única métrica que equilibra ambos os aspectos. A média ponderada é usada para fornecer um F1-score geral para classificação multiclasse.
- **MSE:** O Mean Squared Error (MSE), ou Erro Quadrático Médio, é uma

métrica fundamental para avaliar modelos de regressão. Ele mede a média dos quadrados das diferenças entre os valores que o modelo previu e os valores reais. Um MSE menor indica que o modelo está mais próximo da realidade.

3.6. Principais Resultados

3.6.1. Desempenho por Dataset:

3.6.1.1. Conjunto de dados Iris

A maioria dos algoritmos (Decision Tree, Logistic Regression, MLPClassifier, SVC, KNeighborsClassifier, RandomForestClassifier) alcançou desempenho perfeito com acurácia de 1.00; O GaussianNB teve uma acurácia de 0.978, e o Perceptron de 0.933; Isso indica que o dataset Iris é um problema relativamente simples e, em sua maioria, linearmente separável para classificação.

3.6.1.2. Conjunto de dados Wine

O desempenho foi consistentemente alto. GaussianNB e RandomForestClassifier atingiram acurácia de 1.00. Logistic Regression, MLPClassifier, Perceptron e SVC obtiveram acurácias de 0.981. Decision Tree e KNeighborsClassifier tiveram acurácia de 0.963. As classes no conjunto Wine são bem definidas, o que favorece uma boa separação pelos modelos.

3.6.1.3. Conjunto de dados Breast Cancer

Os algoritmos apresentaram bom desempenho, com acurácia na maioria dos casos acima de 0.93. Logistic Regression e MLPClassifier se destacaram com acurácia de 0.982 e 0.977, respectivamente, seguidos por SVC com 0.977. Perceptron obteve 0.965, KNeighborsClassifier 0.959, e RandomForestClassifier 0.971. Decision Tree e GaussianNB tiveram acurácia de 0.936. O conjunto possui boa separabilidade, mas demonstra um pouco mais de complexidade que Iris ou Wine.

3.6.1.4. Conjunto de dados Digits

Houve maior variação de desempenho, sugerindo um problema mais complexo e desafiador. SVC liderou com 0.980, seguido por MLPClassifier e KNeighborsClassifier (0.976) e Logistic Regression (0.970). RandomForestClassifier obteve 0.969, e Perceptron 0.926. Decision Tree e GaussianNB tiveram acurácias mais baixas, de 0.865 e 0.783, respectivamente. Modelos mais sofisticados, capazes de modelar fronteiras de decisão complexas, performaram melhor.

3.6.1.5. Conjunto de dados Titanic

Os resultados foram mais modestos, com acurácia variando entre aproximadamente 0.73 e 0.82. O melhor desempenho foi do MLPClassifier (0.825), seguido por SVC (0.817) e Logistic Regression (0.810).

http:// https://youtu.be/dUjCb74e4rI

Decision Tree, GaussianNB, KNeighborsClassifier e RandomForestClassifier obtiveram acurácias entre 0.757 e 0.795. Perceptron teve a menor acurácia, de 0.728. A complexidade deste conjunto, com variáveis categóricas, valores ausentes e relações menos claras, dificulta a modelagem.

3.6.1.6. Conjunto de dados Bank Marketing

Foram observadas boas performances gerais. RandomForestClassifier obteve a maior acurácia (0.914), seguido por Logistic Regression (0.912) e SVC (0.909). MLPClassifier (0.901) e KNeighborsClassifier (0.898) também tiveram bom desempenho. Decision Tree obteve 0.890, Perceptron 0.860. O GaussianNB teve o pior desempenho com acurácia de 0.748, possivelmente por não modelar bem as distribuições de atributos nesse caso.

3.6.1.7. Conjunto de dados Diabetes

Este dataset é utilizado para regressão. Para o algoritmo LinearRegression, foi reportado um MSE (Mean Squared Error) de 2821.751. Como a métrica de acurácia não é aplicável diretamente à regressão, essa coluna permaneceu vazia para este dataset.

A figura 1 ilustra a relação entre os **Valores Reais** e os **Valores Previstos** por um modelo de Regressão Linear no dataset de Diabetes. No eixo horizontal, há os valores observados, enquanto no vertical, as predições do modelo. A linha tracejada vermelha representa a condição ideal, onde os valores previstos seriam idênticos aos reais. Os pontos azuis são as observações individuais, com sua proximidade à linha vermelha indicando a acurácia da previsão. Observa-se uma tendência positiva, mostrando que o modelo captura a direção da relação dos dados. No entanto, a **dispersão dos pontos em torno da linha** sugere que, embora o modelo consiga prever a tendência, há uma variação significativa nas previsões individuais. Isso significa que o modelo de Regressão Linear tem um ajuste razoável para o problema de Diabetes, mas ainda possui erros consideráveis. Esse tipo de visualização é crucial para complementar métricas numéricas como MSE, RMSE e R^2, fornecendo uma compreensão intuitiva do desempenho do modelo.

http:// https://youtu.be/dUjCb74e4rI

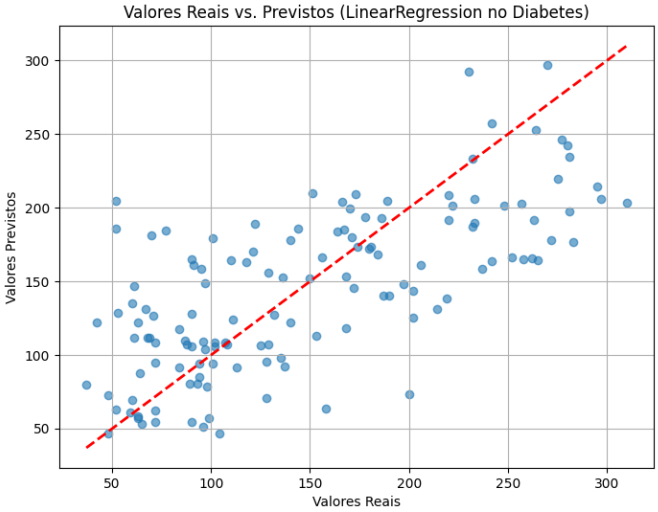

Figura 1. Diagrama de dispersão no dataset de diabetes.

3.6.2. Desempenho por Algoritmo

Alguns algoritmos como Decision Tree Classifier, Logistic Regression, MLP Classifier, e SVC alcançaram desempenho perfeito em alguns conjuntos de dados. GaussianNB teve o desempenho mais baixo no conjunto de dados Digits. Random Forest Classifier alcançou desempenho perfeito no conjunto de dados Wine.

A figura 2 apresenta uma Matriz de Confusão do modelo SVC no dataset Digits. As linhas representam os valores reais (Real), e as colunas, as previsões do modelo (Previsão). Os números na diagonal principal indicam as classificações corretas. Por exemplo, 53 instâncias do dígito 0 foram corretamente previstas como 0. Valores fora da diagonal principal mostram erros de classificação, onde o modelo confundiu uma classe com outra. A predominância de valores na diagonal demonstra um desempenho excelente do modelo SVC, com pouquíssimos erros de classificação para os dígitos manuscritos.

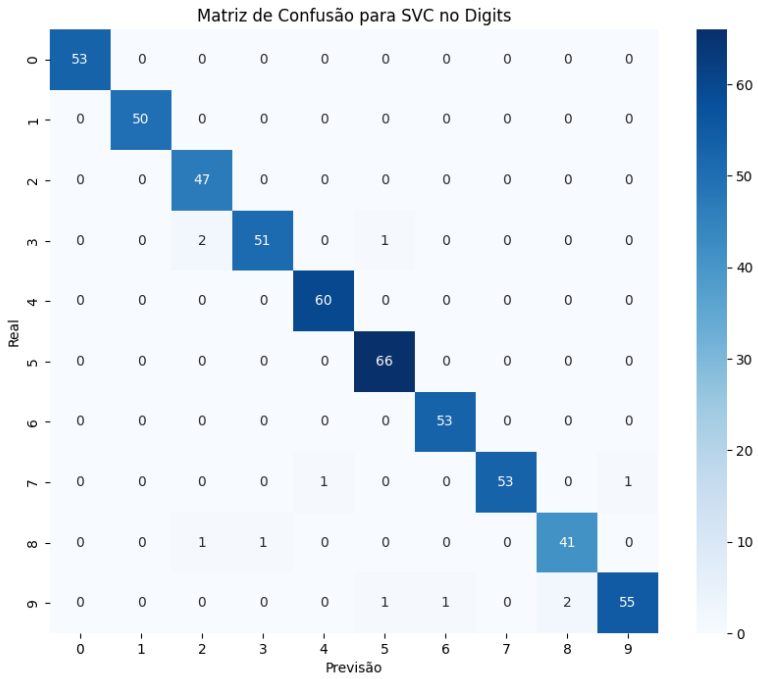

Figura 2. Matriz de confusão.

http:// https://youtu.be/dUjCb74e4rI

4. Conclusão

Este estudo proporcionou uma análise comparativa abrangente do desempenho de nove algoritmos clássicos de aprendizado de máquina aplicados a nove conjuntos de dados de referência. Utilizando métricas robustas — como acurácia, precisão, recall e F1-score — e validação cruzada estratificada com testes estatísticos apropriados (Friedman e Nemenyi), foi possível obter conclusões fundamentadas quanto à eficácia relativa de cada modelo.

Os resultados indicam que, em conjuntos de dados com menor complexidade e classes bem definidas, como Iris e Wine, a maioria dos algoritmos apresenta desempenho excelente e próximo do ideal, com pouca variabilidade entre eles. Em contrapartida, em conjuntos mais complexos ou desbalanceados, como Bank Marketing ou Titanic, observou-se uma maior dispersão nos resultados, refletindo a sensibilidade dos algoritmos às características intrínsecas dos dados.

Além disso, evidenciou-se que algoritmos mais sofisticados como Random Forest e MLP tendem a se destacar em cenários de maior complexidade, ao custo de maior demanda computacional. Já algoritmos mais simples como o Perceptron e o Naive Bayes apresentaram desempenho competitivo em contextos específicos, especialmente quando a estrutura dos dados favorece suas premissas internas.

Portanto, a escolha do algoritmo ideal não é universal: ela depende fortemente da natureza dos dados, dos recursos computacionais disponíveis e dos requisitos de precisão da aplicação prática. Esta pesquisa reforça a importância de uma análise preliminar cuidadosa dos dados e da aplicação de testes estatísticos na seleção de modelos preditivos. Como perspectivas futuras, propõe-se expandir esta análise para incluir aspectos como tempo de treinamento, interpretabilidade dos modelos e desempenho em cenários de dados ruidosos ou com valores ausentes, aproximando ainda mais a avaliação das exigências do mundo real.

5. Referências

CARUANA, R.; NICULESCU-MIZIL, A. *An empirical comparison of supervised learning algorithms*. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 2006. Anais [...]. [S.l.]: ACM, 2006.

DEMŠAR, J. *Statistical comparisons of classifiers over multiple data sets*. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2019. Disponível em: http://archive.ics.uci.edu/ml. Acesso em: 20 maio 2025.

FERNANDES, J. L. et al. *Statistical comparison of classifiers using multiple datasets*. *Expert Systems with Applications*, v. 168, 2021. DOI: https://doi.org/10.1016/j.eswa.2020.114481.

HUNTER, J. D. *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, v. 9, n. 3, p. 90–95, 2007.

KAGGLE. *Titanic - Machine Learning from Disaster*. Disponível em: https://www.kaggle.com/competitions/titanic. Acesso em: 19 maio 2025.

http:// https://youtu.be/dUjCb74e4rI

PANDAS DEVELOPMENT TEAM. pandas-dev/pandas: Pandas. 2020. DOI: 10.5281/zenodo.3509134.

PEDREGOSA, F. et al. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, v. 12, p. 2825–2830, 2011.

REPOSITÓRIO DE DADOS UCI. Disponível em: https://archive.ics.uci.edu/ml/index.php. Acesso em: 19 maio 2025.

SCIKIT-LEARN. *Machine learning in Python*. Disponível em: https://scikit-learn.org. Acesso em: 19 maio 2025.

S. PEREIRA, "Repositório de Comparação de Algoritmos de Machine Learning," GitHub. Disponível em: https://github.com/saulopereira2018/comparacao_algoritmos.git. Acesso em: 3 jun. 2025.

WASKOM, M. *Seaborn: Statistical Data Visualization*. 2021. Disponível em: https://seaborn.pydata.org. Acesso em: 20 maio 2025.

Apêndice

Tabela 1. Comparação do desempenho algoritmos de aprendizado de máquina

Dataset	Algoritmo	Tipo da Tarefa	Acuracia	Precision (Weighted Avg)	Recall (Weighted Avg)	F1-Score (Weighted Avg)	MSE
Iris	DecisionTreeClassifier	classification	1	1	1	1	
Iris	LogisticRegression	classification	1	1	1	1	
Iris	MLPClassifier	classification	1	1	1	1	
Iris	Perceptron	classification	0,933333333	0,939153439	0,933333333	0,930508141	
Iris	SVC	classification	1	1	1	1	
Iris	GaussianNB	classification	0,977777778	0,979365079	0,977777778	0,977744856	
Iris	KNeighborsClassifier	classification	1	1	1	1	
Iris	RandomForestClassifier	classification	1	1	1	1	

http:// https://youtu.be/dUjCb74e4rI

Wine	DecisionTreeClassifier	classification	0,962962963	0,963804714	0,962962963	0,962835359	
Wine	LogisticRegression	classification	0,981481481	0,982716049	0,981481481	0,981574931	
Wine	MLPClassifier	classification	0,981481481	0,982716049	0,981481481	0,981574931	
Wine	Perceptron	classification	0,981481481	0,982407407	0,981481481	0,981493063	
Wine	SVC	classification	0,981481481	0,982323232	0,981481481	0,981353878	
Wine	GaussianNB	classification	1	1	1	1	
Wine	KNeighborsClassifier	classification	0,962962963	0,965123457	0,962962963	0,962593739	
Wine	RandomForestClassifier	classification	1	1	1	1	
Breast Cancer	DecisionTreeClassifier	classification	0,935672515	0,938260843	0,935672515	0,936137964	
Breast Cancer	LogisticRegression	classification	0,982456140	0,982584235	0,982456140	0,982484411	
Breast Cancer	MLPClassifier	classification	0,976608187	0,976608187	0,976608187	0,976608187	
Breast Cancer	Perceptron	classification	0,964912281	0,965411351	0,964912281	0,965022442	
Breast Cancer	SVC	classification	0,976608187	0,976608187	0,976608187	0,976608187	
Breast Cancer	GaussianNB	classification	0,935672515	0,935522811	0,935672515	0,935563425	
Breast Cancer	KNeighborsClassifier	classification	0,959064327	0,958995966	0,959064327	0,958994907	
Breast Cancer	RandomForestClassifier	classification	0,970760234	0,971100047	0,970760234	0,970603924	
Digits	DecisionTreeClassifier	classification	0,864814815	0,866166023	0,864814815	0,865050835	
Digits	LogisticRegression	classification	0,970370737	0,971507158	0,970370737	0,970507787	
Digits	MLPClassifier	classification	0,975925926	0,97662616	0,975925926	0,975971479	
Digits	Perceptron	classification	0,925925926	0,930410696	0,925925926	0,926762078	
Digits	SVC	classification	0,97962963	0,979966116	0,979629633	0,979534538	
Digits	GaussianNB	classification	0,783333333	0,831032825	0,783333333	0,780121459	
Digits	KNeighborsClassifier	classification	0,975925926	0,976056328	0,975925926	0,975750177	
Digits	RandomForestClassifier	classification	0,968518519	0,968914114	0,968518519	0,968515769	
Diabetes	LinearRegression	regression					2821,750981
Titanic	DecisionTreeClassifier	classification	0,757462687	0,756213256	0,757462687	0,756569241	
Titanic	LogisticRegression	classification	0,809701493	0,808856585	0,809701493	0,808332319	
Titanic	MLPClassifier	classification	0,824626866	0,834857479	0,824626866	0,818864897	
Titanic	Perceptron	classification	0,727611194	0,734327355	0,727611194	0,729263944	
Titanic	SVC	classification	0,817164179	0,82477751	0,817164179	0,811744683	

http:// https://youtu.be/dUjCb74e4rI

Titanic	GaussianNB	classification	0,794776119	0,794521495	0,794776119	0,794637564	
Titanic	KNeighborsClassifier	classification	0,791044776	0,791523489	0,791044776	0,787490612	
Titanic	RandomForestClassifier	classification	0,794776119	0,793808635	0,794776119	0,792899701	
Bank Marketing	DecisionTreeClassifier	classification	0,890183702	0,891814958	0,890183702	0,890979319	
Bank Marketing	LogisticRegression	classification	0,912276443	0,901579543	0,912276443	0,903235032	
Bank Marketing	MLPClassifier	classification	0,901189609	0,897126065	0,901189609	0,898968268	
Bank Marketing	Perceptron	classification	0,860322085	0,874881526	0,860322085	0,866804732	
Bank Marketing	SVC	classification	0,909201262	0,896581666	0,909201262	0,897355484	
Bank Marketing	GaussianNB	classification	0,748158938	0,89024997	0,748158938	0,791920881	
Bank Marketing	KNeighborsClassifier	classification	0,898114429	0,881965502	0,898114429	0,885739208	
Bank Marketing	RandomForestClassifier	classification	0,914380513	0,906125427	0,914380513	0,90847543	

http:// https://youtu.be/dUjCb74e4rI