



UNIVERSIDADE DE
COIMBRA

Assignment 3: Classifiers

Miguel António - 2022237621

Introdução

Neste trabalho prático foi nos pedido para construirmos diferentes tipos de classificadores em diferentes datasets, treiná-los e comparar as suas performances. Os três classificadores que utilizei foram os que o docente recomendou e que tínhamos abordado nas aulas teóricas: KNN, SVM e MLP. Treinei estes classificadores com três *datasets* diferentes: *iris*, *digits* e *MNIST*.

Metodologia

Os três algoritmos usados foram os seguintes:

- O algoritmo *KNN* é baseado em distâncias e tem como objetivo classificar um ponto de acordo com as classes dos seus vizinhos mais próximos. O número de vizinhos é definido pelo parâmetro *K* e usei a distância Euclidiana.
- O algoritmo *SVM*, tenta encontrar uma hiperplano ótimo que separa as classes, tendo como principal objetivo maximizar a margem entre essas classes (distância entre os pontos de fronteira e a linha de separação). O kernel que usei foi o linear, que é usado em dados que são linearmente separáveis e calcula apenas o produto interno entre vetores.
- O algoritmo *MLP* é uma rede neuronal artificial com várias camadas de neurónios, constituída por três camadas (camada de entrada, camada(s) oculta(s) e camada de saída). Cada neurónio tem uma função de ativação que permite aprender relações entre dados e o seu treino é feito com *backpropagation*.

Os três datasets utilizados foram os seguintes:

- *Iris Dataset*: contém 150 amostras de flores, divididas em três classes (Setosa, Versicolor e Virginiana) e cada amostra tem quatro features (Comprimento da pétala, comprimento da sépala, largura da pétala, largura da sépala), além disso o *dataset* é balanceado como mostra a figura 1.
- *Digits Dataset*: contém 1797 imagens de dígitos manuscritos (0-9), em escala de cinzentos, com resolução 8x8 pixels em que cada amostra tem 64 features e também é um dataset balanceado como mostra a figura 2.
- *MNIST Dataset*: contém 70000 imagens de dígitos manuscritos (0-9), as imagens são 28x28 tendo 784 features e também é balanceado como mostra a seguinte figura.

Em apêndice estão os gráficos que verificam o balanceamento dos três *datasets*.

Experiências e Resultados

Nesta etapa foram treinados e testados os três classificadores descritos anteriormente para os três *datasets* distintos. Para todos os *datasets* usei a mesma base de cada classificador: *KNN* (com três vizinhos e cinco vizinhos), *SVM* (com kernel linear) e *MLP* (com dez *hidden layers* e trinta *hidden layers*, ambos com 1000 iterações máximas). Além disso, dividi os dados em conjunto de treino e teste num rácio de 80/20.

Os resultados obtidos encontram-se nas seguintes tabelas, que dizem respeito aos *classifications reports* e matriz de confusão.

Model	Precision	Recall	F1-Score	Accuracy
KNN	1.00	1.00	1.00	1.00
SVM	1.00	1.00	1.00	1.00
MLP	0.97	0.97	0.97	0.97

Tabela 1 - *Weighted Sum* das métricas de avaliação para *Iris Dataset*

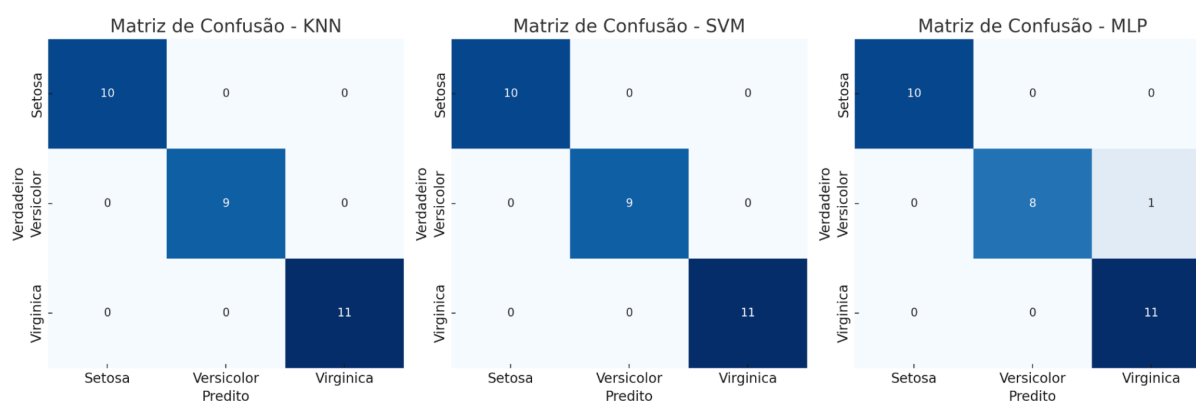


Figura 1 - Matrizes Confusão dos modelos no *Iris Dataset*

No dataset *Iris*, todos os classificadores apresentaram desempenhos bastante elevados. O *KNN* e a *SVM* atingiram uma *accuracy* de 100%, classificando corretamente todas as amostras das três espécies de flores, o que confirma que este conjunto de dados é linearmente separável e adequado para métodos clássicos. Já o *MLP* obteve uma *accuracy* ligeiramente inferior, de 97%, devido a um único erro de classificação em que uma amostra da classe *Versicolor* foi identificada como *Virginica*. Apesar deste pequeno desvio, os valores de *precision*, *recall* e *f1-score* mantiveram-se muito altos, evidenciando que a rede neuronal também conseguiu generalizar bem os padrões.

Model	Precision	Recall	F1-Score	Accuracy
KNN	0.99	0.99	0.99	0.99
SVM	0.98	0.98	0.98	0.98
MLP	0.95	0.94	0.94	0.94

Tabela 2 - *Weighted Sum* das métricas de avaliação para *Digits Dataset*

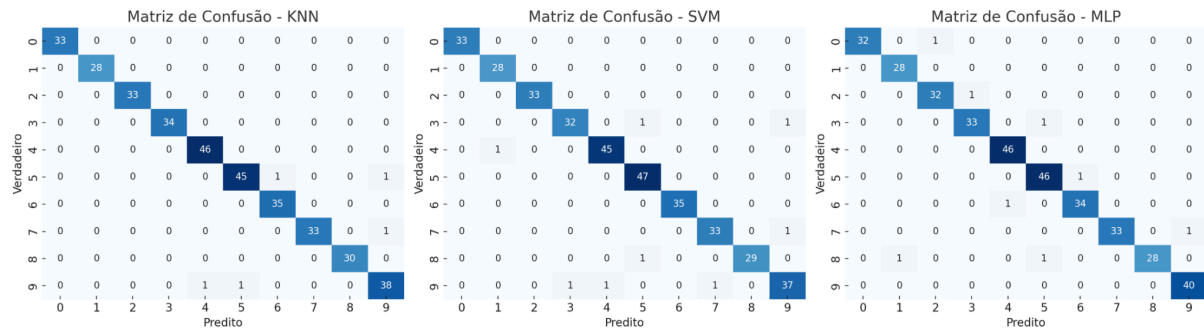


Figura 2 - Matrizes Confusão dos modelos no *Digits Dataset*

No dataset *Digits*, todos os classificadores apresentaram desempenhos elevados, embora com pequenas diferenças entre si. O *KNN* foi o que obteve melhor resultado, com uma *accuracy* de 99%, classificando corretamente praticamente todos os dígitos. O *SVM* alcançou também uma performance muito elevada, com *accuracy* de 98%, apresentando apenas alguns erros pontuais, sobretudo nas classes 3 e 9, que foram ligeiramente confundidas com dígitos semelhantes. O *MLP*, apesar de manter valores bastante competitivos, apresentou a menor taxa de acerto, com *accuracy* de 94%. Neste caso, as confusões foram mais frequentes em classes como 1, 2, 7 e 9, o que se refletiu numa ligeira quebra dos valores de *precision* e *recall*, demonstrando que todos os modelos são eficazes no reconhecimento de dígitos manuscritos de baixa resolução.

Model	Precision	Recall	F1-Score	Accuracy
KNN	0.97	0.97	0.97	0.97
SVM	0.94	0.94	0.94	0.94
MLP	0.93	0.93	0.93	0.93

Tabela 3 - *Weighted Sum* das métricas de avaliação para *MNIST Dataset*

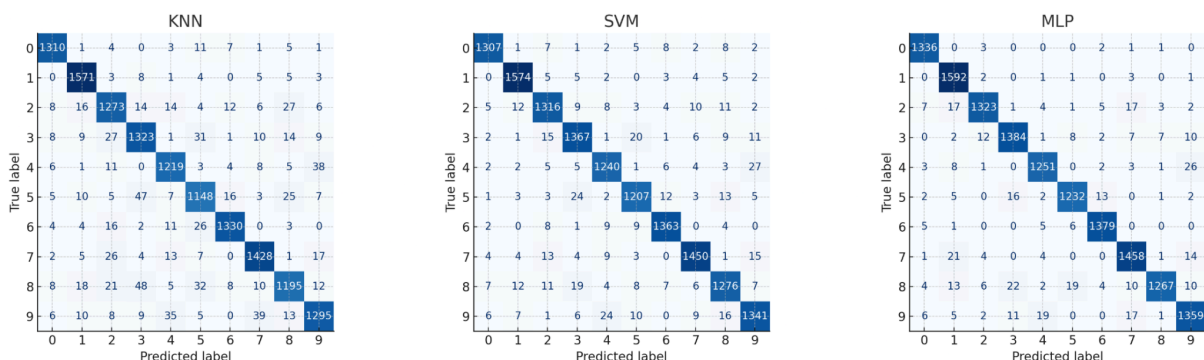


Figura 3 - Matrizes Confusão dos modelos no *MNIST Dataset*

No dataset *MNIST*, os três classificadores apresentaram desempenhos sólidos, mas com diferenças mais evidentes do que nos datasets anteriores. O *KNN* obteve a melhor performance, alcançando uma *accuracy* de 97%, com valores de *precision* e *recall* bastante elevados em praticamente todas as classes, embora com pequenas confusões pontuais em dígitos semelhantes, como o 8 e o 9. A *SVM*

apresentou uma *accuracy* inferior, de 94%, mostrando maior dificuldade em distinguir algumas classes, especialmente os dígitos 2, 3, 4 e 9, que registaram quebras visíveis nas métricas de avaliação. Já o *MLP* atingiu também uma *accuracy* de 93%, um pouco abaixo da *SVM*, com confusões recorrentes em dígitos com formas próximas, como 5, 8 e 9, mas apesar disso, manteve resultados consistentes e equilibrados em todas as classes. De forma geral, nota-se que, embora o *KNN* continue a apresentar um desempenho competitivo, este método não é escalável para datasets de maior dimensão devido ao elevado custo computacional na fase de teste. A *SVM* e o *MLP*, apesar da ligeira queda de desempenho, mostraram-se mais adequados em termos de escalabilidade, reforçando a diferença de comportamento dos classificadores quando aplicados em problemas mais complexos e de maior dimensão como o *MNIST*.

Conclusão

Em suma, os três classificadores apresentaram desempenhos elevados, mas com comportamentos distintos consoante a complexidade dos dados: no *Iris*, todos alcançaram quase a perfeição, confirmando a simplicidade do conjunto; no *Digits*, o *KNN* destacou-se com resultados quase perfeitos, seguido da *SVM*, enquanto o *MLP* mostrou maior propensão a erros; já no *MNIST*, o *KNN* manteve elevada precisão, mas com limitações de escalabilidade, ao passo que a *SVM* e o *MLP*, embora ligeiramente inferiores em *accuracy*, revelaram-se mais adequados para problemas de maior dimensão. Assim, conclui-se que o *KNN* é ideal para datasets pequenos, a *SVM* oferece um bom equilíbrio entre desempenho e generalização, e o *MLP* apresenta maior potencial de adaptação e escalabilidade. Nas figuras seguintes conseguimos observar uma comparação mais intuitiva entre os modelos e o seu comportamento para cada *dataset*.

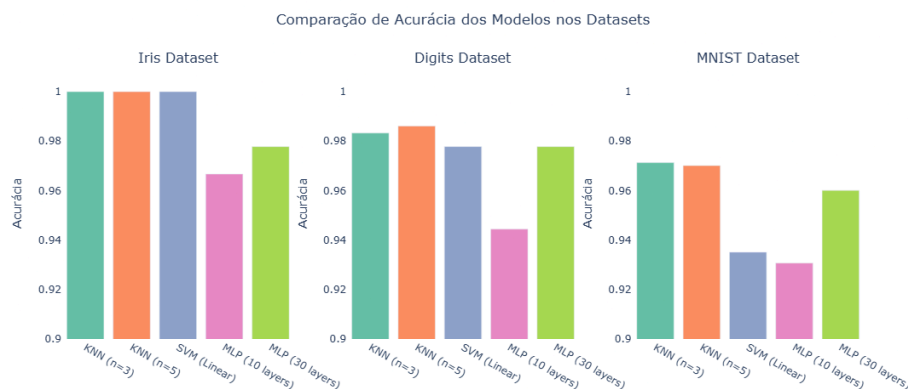


Figura 4 - Comparação da *accuracy* entre modelos



Figura 5 - Comparação das métricas entre modelos

Referências

- [1] Scikit-learn developers, “Support Vector Machines,” *Scikit-learn*. [Online]. Disponível: <https://scikit-learn.org/stable/modules/svm.html>. [Accessed: Oct. 2, 2025].
- [2] Scikit-learn developers, “MLPClassifier,” *Scikit-learn*. [Online]. Disponível: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. [Accessed: Oct. 2, 2025].
- [3] Scikit-learn developers, “KNeighborsClassifier,” *Scikit-learn*. [Online]. Disponível: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>. [Accessed: Oct. 2, 2025].

Apêndice

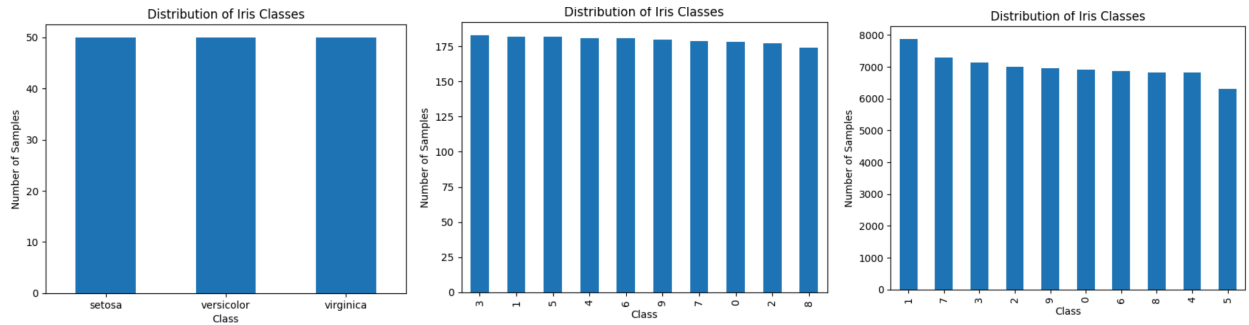


Figura 6 - Distribuição das classes por *dataset* (*Iris*, *Digits* e *MNIST*)

Class	Precision	Recall	F1-Score	Support
Setosa	1.00	1.00	1.00	10
Versicolor	1.00	1.00	1.00	9
Virginica	1.00	1.00	1.00	11
Accuracy	—	—	1.00	30
Macro avg	1.00	1.00	1.00	30
Weighted avg	1.00	1.00	1.00	30

Class	Precision	Recall	F1-Score	Support
Setosa	1.00	1.00	1.00	10
Versicolor	1.00	1.00	1.00	9
Virginica	1.00	1.00	1.00	11
Accuracy	—	—	1.00	30
Macro avg	1.00	1.00	1.00	30
Weighted avg	1.00	1.00	1.00	30

Class	Precision	Recall	F1-Score	Support
Setosa	1.00	1.00	1.00	10
Versicolor	1.00	0.89	0.94	9
Virginica	0.92	1.00	0.96	11
Accuracy	—	—	0.97	30
Macro avg	0.97	0.96	0.97	30
Weighted avg	0.97	0.97	0.97	30

Figura 7 - *Classifications report* para *KNN*, *SVM* e *MLP* para *IrisDataset*

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	33
1	1.00	1.00	1.00	28
2	1.00	1.00	1.00	33
3	1.00	1.00	1.00	34
4	0.98	1.00	0.99	46
5	0.98	0.96	0.97	47
6	0.97	1.00	0.99	35
7	1.00	0.97	0.99	34
8	1.00	1.00	1.00	30
9	0.95	0.95	0.95	40
Accuracy	—	—	0.99	360
Macro avg	0.99	0.99	0.99	360
Weighted avg	0.99	0.99	0.99	360

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	33
1	0.97	1.00	0.98	28
2	1.00	1.00	1.00	33
3	0.97	0.94	0.96	34
4	0.98	0.98	0.98	46
5	0.96	1.00	0.98	47
6	1.00	1.00	1.00	35
7	0.97	0.97	0.97	34
8	1.00	0.97	0.98	30
9	0.95	0.93	0.94	40
Accuracy	—	—	0.98	360
Macro avg	0.98	0.98	0.98	360
Weighted avg	0.98	0.98	0.98	360

Class	Precision	Recall	F1-Score	Support
0	0.94	1.00	0.97	33
1	0.89	0.86	0.87	28
2	0.86	0.97	0.91	33
3	0.97	0.94	0.96	34
4	1.00	0.96	0.98	46
5	0.94	0.96	0.95	47
6	0.97	0.94	0.96	35
7	0.97	0.97	0.97	34
8	0.93	0.83	0.88	30
9	0.95	0.97	0.96	40
Accuracy	—	—	0.94	360
Macro avg	0.94	0.94	0.94	360
Weighted avg	0.95	0.94	0.94	360

Figura 8 - *Classifications report* para *KNN*, *SVM* e *MLP* para *Digits Dataset*

Class	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.98	1343
1	0.96	0.99	0.98	1600
2	0.97	0.97	0.97	1380
3	0.97	0.96	0.97	1433
4	0.97	0.96	0.97	1295
5	0.98	0.97	0.97	1273
6	0.98	0.99	0.99	1396
7	0.97	0.98	0.97	1503
8	0.99	0.94	0.96	1357
9	0.96	0.95	0.96	1420
Accuracy	—	—	0.97	14000
Macro avg	0.97	0.97	0.97	14000
Weighted avg	0.97	0.97	0.97	14000

Class	Precision	Recall	F1-Score	Support
0	0.96	0.96	0.96	1343
1	0.95	0.97	0.96	1600
2	0.92	0.92	0.92	1380
3	0.93	0.91	0.92	1433
4	0.92	0.93	0.92	1295
5	0.91	0.90	0.91	1273
6	0.95	0.95	0.95	1396
7	0.93	0.95	0.94	1503
8	0.91	0.90	0.90	1357
9	0.92	0.92	0.92	1420
Accuracy	—	—	0.93	14000
Macro avg	0.93	0.93	0.93	14000
Weighted avg	0.93	0.93	0.93	14000

Class	Precision	Recall	F1-Score	Support
0	0.97	0.98	0.97	1343
1	0.96	0.98	0.97	1600
2	0.91	0.92	0.92	1380
3	0.91	0.92	0.92	1433
4	0.93	0.94	0.94	1295
5	0.90	0.90	0.90	1273
6	0.97	0.95	0.96	1396
7	0.95	0.95	0.95	1503
8	0.92	0.88	0.90	1357
9	0.93	0.91	0.92	1420
Accuracy	—	—	0.94	14000
Macro avg	0.93	0.93	0.93	14000
Weighted avg	0.94	0.94	0.94	14000

Figura 9 - *Classifications report* para *KNN*, *SVM* e *MLP* para *MNIST Dataset*