

EFC 2

Cláudio Ferreira Carneiro - RA 263796

9 de Novembro de 2019

1 Parte II – Classificação binária com redes MLP e SVMs

O código referente às atividades se encontra no repositório:

<https://github.com/carneirofc/IA006.git>

a)

Antes da inicialização do treinamento, os dados de entrada foram redimensionados utilizando o *standard scaler*, tendo a média removida e escalado à variância unitária, equação 1. A classe representada por -1 passará a ser representada por 0 a fim de utilizar a função de custo *Binary Cross-Entropy* no treinamento do modelo. A figura 2 apresenta o histograma das saídas (classes) do *dataset* de treinamento.

$$z = (x - \mu) / \sigma \quad (1)$$

A figura 1 apresenta os dados de treinamento de entrada originais e escalados pelo *standard scaler*. Nota-se que não ocorreram mudanças significativas no *dataset*.

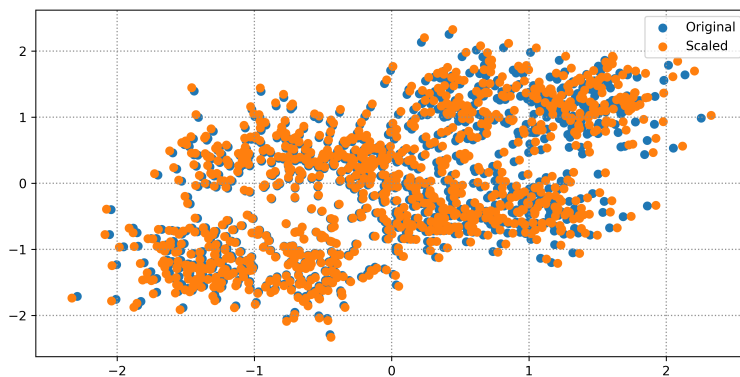


Figura 1: *Dataset* de treinamento, atributos de entrada originais e escalados.

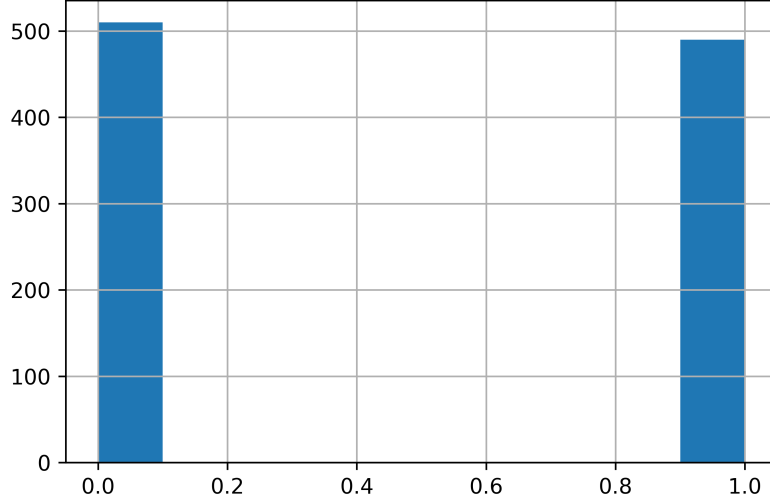


Figura 2: *Dataset* de treinamento, classes 0 e 1.

A MPL foi definida como uma rede de 35 neurônios na camada intermediária, utilizando como função de ativação a *ReLU*. Na camada de saída apenas um neurônio é utilizado, cuja função de ativação é a *sigmoid*, equação 2. O processo de treinamento da MPL utilizou o *RMSprop* como algoritmo de treinamento.

$$y(x) = \frac{1}{1 + \exp^{-x}} \quad (2)$$

O processo de treinamento foi definido com duração de 1000 épocas com critérios de parada prematura. Os dados de entrada foram embaralhados e apresentados em *batch* com 10 amostras. Como critérios de parada prematura, foram utilizados *snapshots* da configuração de pesos de menor custo nos dados de validação e a variação do erro de validação. Quando o erro de validação não superar o limiar especificado (0.001) por *patience* = 200 épocas, o treinamento é finalizado.

A MLP foi treinada por 200 épocas, quando o processo foi interrompido pela parada prematura, com custo de treinamento *loss* : 0.3225 e custo de validação *val_loss* : 0.3459, figura 3. Os pesos utilizados são da época 146, onde o custo de validação é *val_loss* : 0.3391.

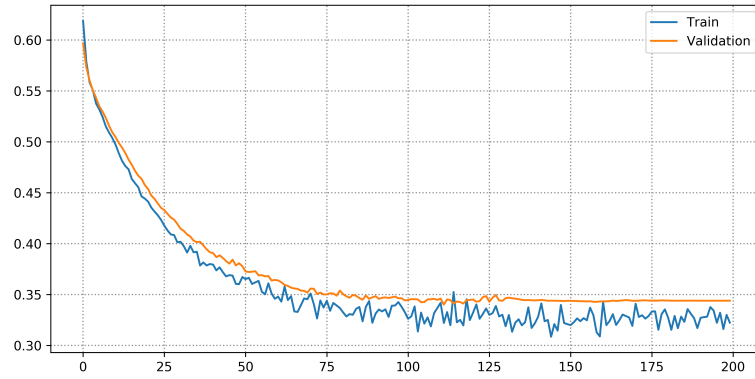


Figura 3: Histórico de erro no processo de treinamento.

A curva ROC da MLP é vista na figura 4, com pontuação de ≈ 0.9261 para os dados de validação.

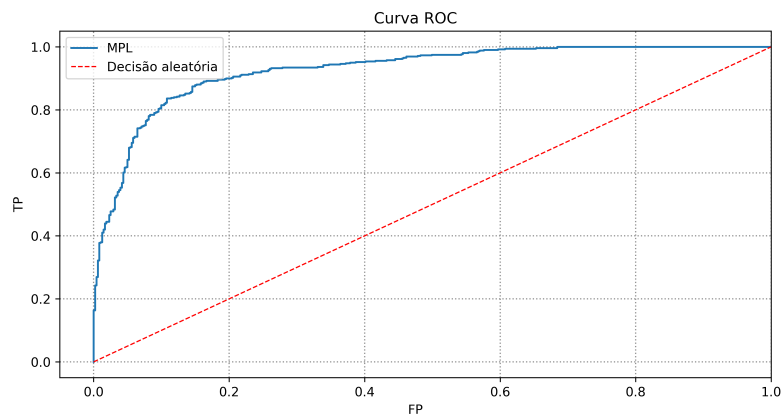


Figura 4: Curva ROC da MLP para os dados de validação.

b)

A região de decisão se assemelha à região ótima porém é visível que ainda existem padrões a serem corretamente capturados pela MLP.

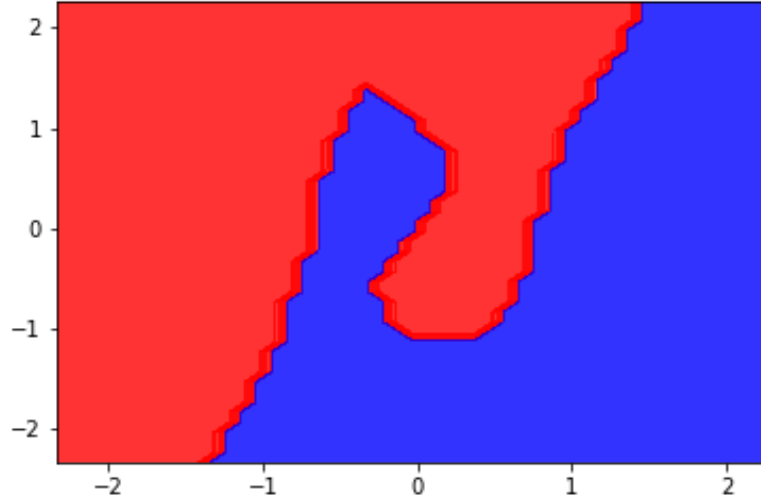


Figura 5: Regiões de decisão.

c)

A MPL obteve uma acurácia de 87% como é visto na tabela 1.

	Precision	Recall	f1-score	support
Classe -1	0.86	0.90	0.88	499
Classe 1	0.89	0.85	0.87	501
accuracy			0.87	1000
macro avg	0.87	0.87	0.87	1000
weighted avg	0.87	0.87	0.87	1000

Tabela 1: Classificação do *dataset* de teste, MLP.

d)

e)

Vetores suport e regiões de decisão da SVM de $C = 1$ e kernel RBF.

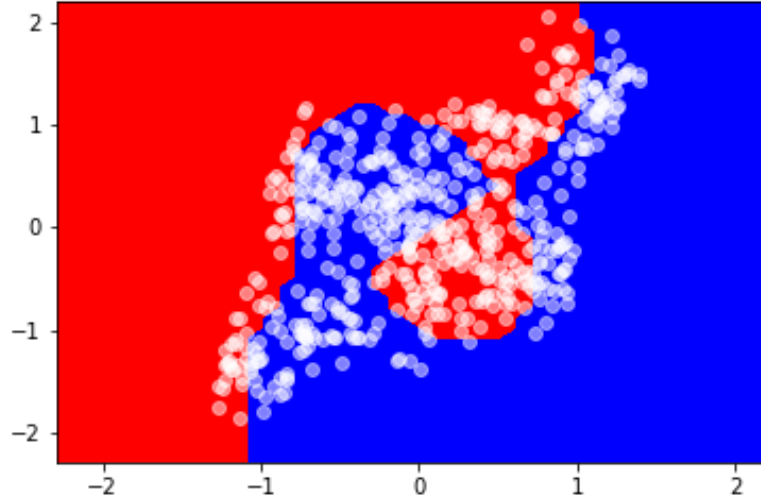


Figura 6: Regiões de decisão SVM.

f)

Com o uso da SVM é obtida uma acurácia de 87% conforme a tabela 2.

	Precision	Recall	f1-score	support
Classe -1	0.86	0.87	0.87	499
Classe 1	0.87	0.86	0.87	501
accuracy			0.87	1000
macro avg	0.87	0.87	0.87	1000
weighted avg	0.87	0.87	0.87	1000

Tabela 2: Classificação do *dataset* de teste, SVM.

g)

As figuras 7 e 8 apresentam as regiões de decisão de SVMs com kernel linear e polinomial respectivamente. Nota-se que a escolha do kernel possui um impacto significativo no desempenho da SVM. Conforme as regiões de decisão obtidas, os kernels não possuem a flexibilidade exigida pelo problema.

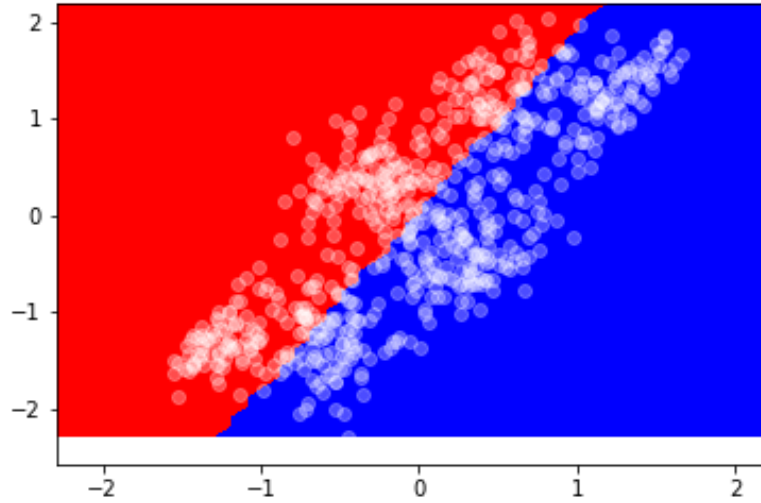


Figura 7: Regiões de decisão, SVM com kernel linear.

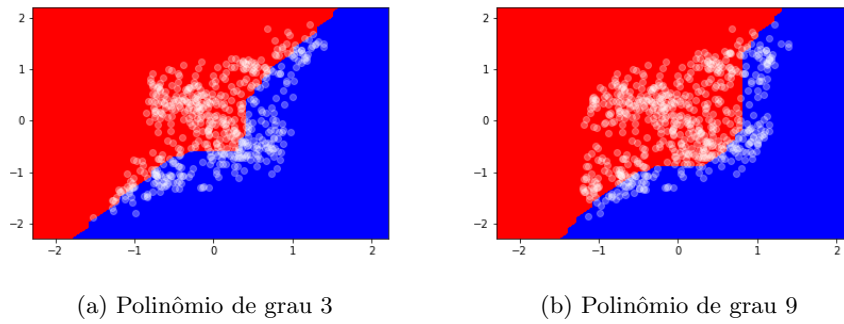


Figura 8: Regiões de decisão SVM, com kernel polinomial de grau 3 e 9.

A figura 9 apresenta as regiões de decisão conforme o parâmetro C é modificado. Para valores baixos de C , a SVM tende a classificar incorretamente maiores quantidades de amostras.

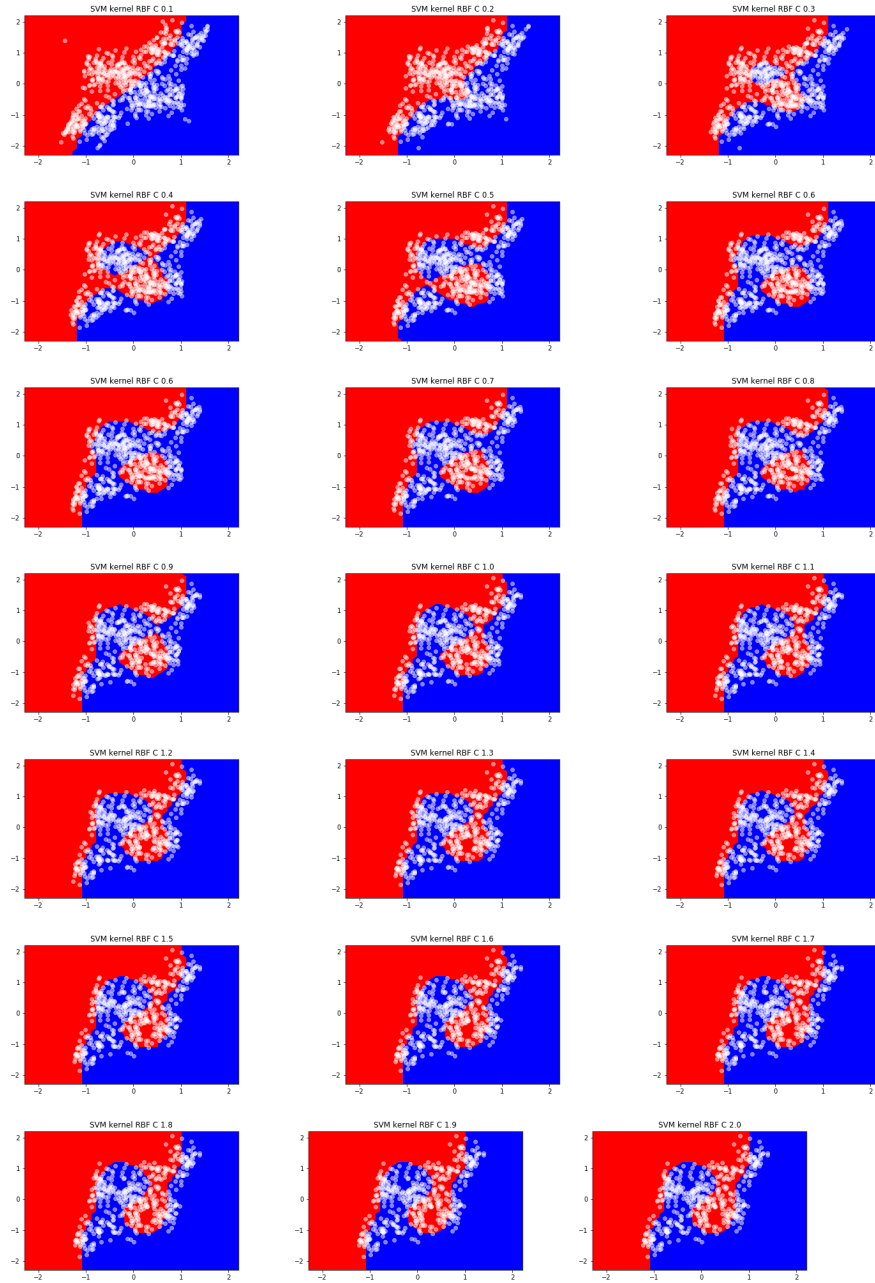


Figura 9: Regiões de decisão SVM, com kernel RBF para diversos C .