

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*.

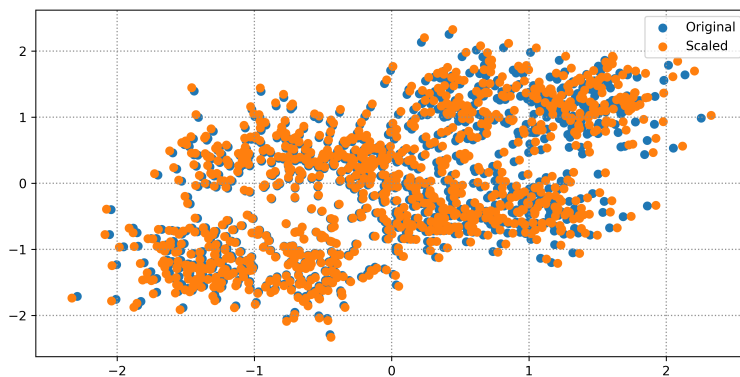


Figura 1: *Dataset* de treinamento, atributos de entrada originais e escalados.

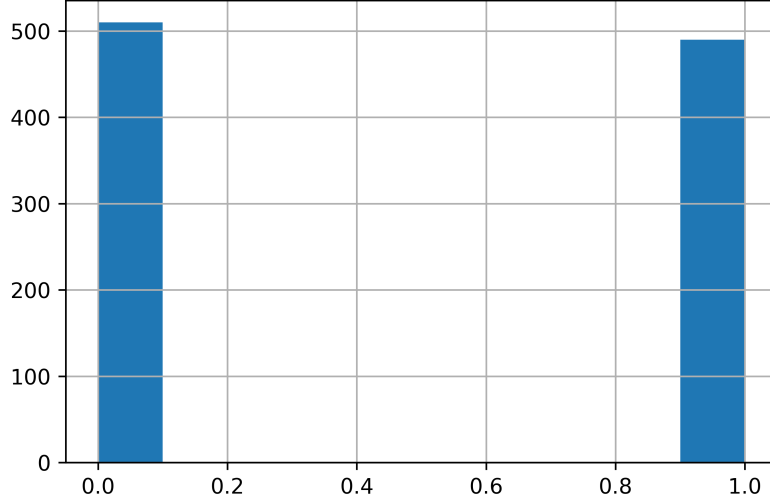


Figura 2: *Dataset* de treinamento, classes 0 e 1.

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 critério de parada prematura é utilizada a variação do erro de validação. A melhor configuração da MLP durante o treinamento é restaurada por *snapshots* da configuração de pesos de menor custo nos dados de validação.

Quando o erro de validação da época não superar o limiar especificado (0.001) por *patience* = 200 épocas, o treinamento é finalizado.

A MLP foi treinada por 200 épocas, onde o treinamento foi interrompido pela parada prematura, com custo de treinamento 0.3225 e custo de validação 0.3459. A figura 3 apresenta o histórico do erro de treinamento e validação. Os pesos atribuídos à MLP são provenientes da época 146, onde o custo de validação é 0.3391.

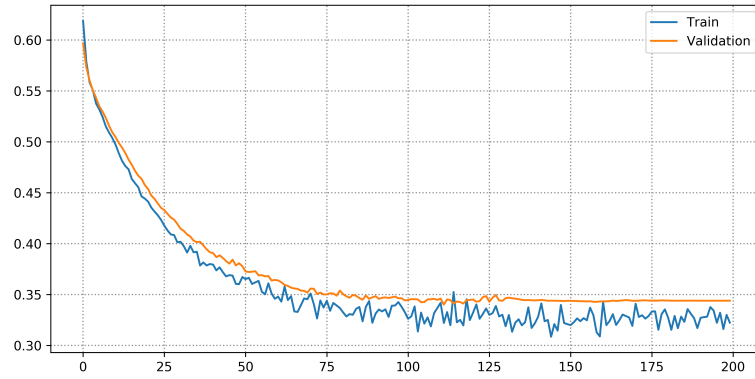


Figura 3: Histórico de erro no processo de treinamento.

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

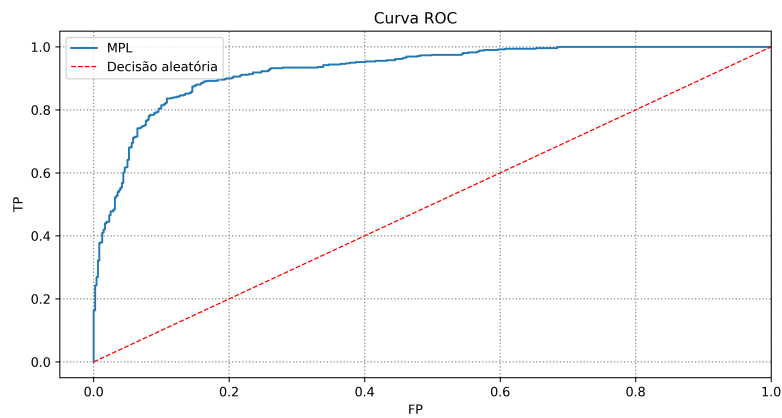


Figura 4: Curva ROC da MLP para os dados de validação.

b)

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

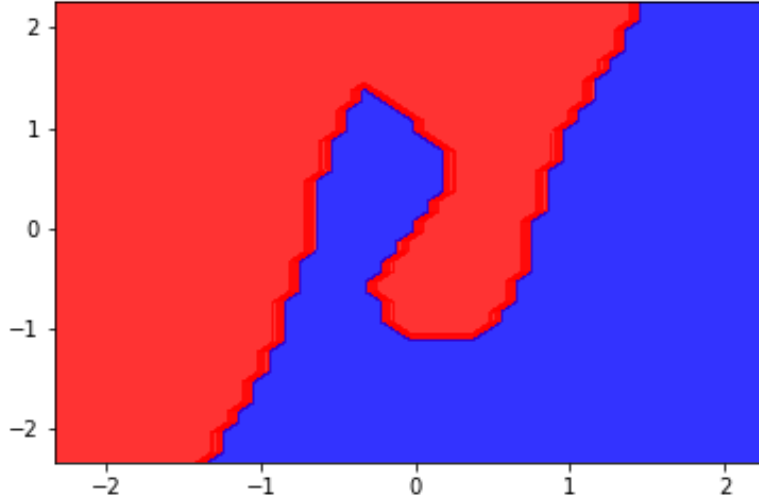


Figura 5: Regiões de decisão.

c)

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

	Precision	Recall	f1-score	support
Classe -1	0.86	0.88	0.87	499
Classe 1	0.88	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 1: Classificação do *dataset* de teste, MLP.

d)

Utilizando as configurações de treinamento citadas anteriormente, 7 MLPs foram treinadas com camadas escondidas de 5, 15, 25, 35, 45, 55 e 65 neurônios. As curvas ROC dos classificadores são apresentadas na figura 6.

As redes com 5 e 15 neurônios apresentaram desempenho inferior às demais configurações, um forte indicador de que o modelo não é suficientemente flexível para a classificação dos dados.

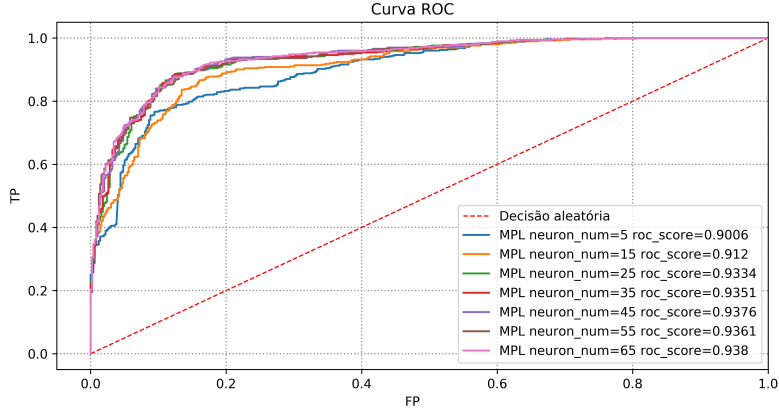


Figura 6: Curvas ROC dos classificadores MLP.

Conforme os resultados da classificação do *dataset* apresentado na tabela 2, o ganho no desempenho de classificação conforme o número de neurônio não é linear. As MLPs de 25, 35, 45, 55 e 65 apresentaram performance similar, tanto em sua acurácia quanto na $f1 - score$.

	macro avg precision	macro avg recall	macro avg f1-score	accuracy
5	0.7975081828740365	0.7817711270845084	0.7790265816831282	0.782
15	0.8324343163756176	0.8281153124612499	0.8274589111453519	0.828
25	0.8750660095053688	0.8750135000540002	0.8749968749218731	0.875
35	0.8717637345901665	0.8710454841819367	0.8709430859008823	0.871
45	0.8723651286157541	0.8720314881259525	0.8719749070817879	0.872
55	0.8682107087827426	0.8670574682298728	0.8669029722667825	0.867
65	0.8705731075673273	0.8700394801579207	0.869957866348697	0.87

Tabela 2: Resultado da classificação das MLPs.

e)

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

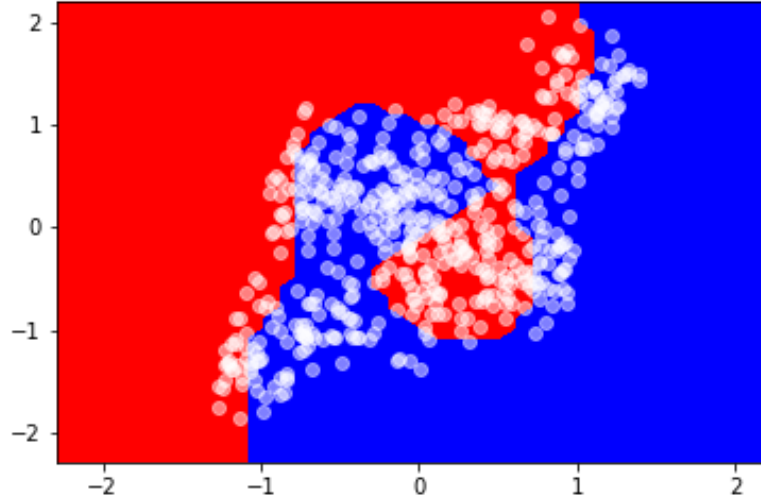


Figura 7: Regiões de decisão SVM.

f)

Com o uso da SVM é obtida uma acurácia de 87% conforme a tabela 3.

	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 3: Classificação do *dataset* de teste, SVM.

g)

As figuras 8 e 9 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.

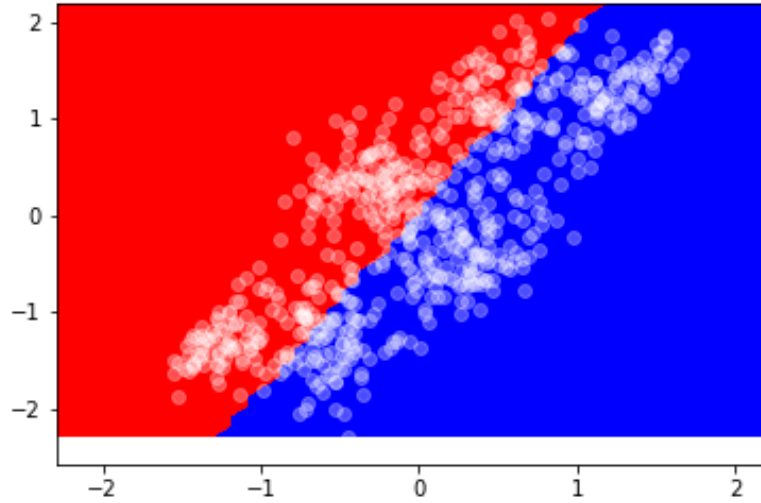


Figura 8: Regiões de decisão, SVM com kernel linear.

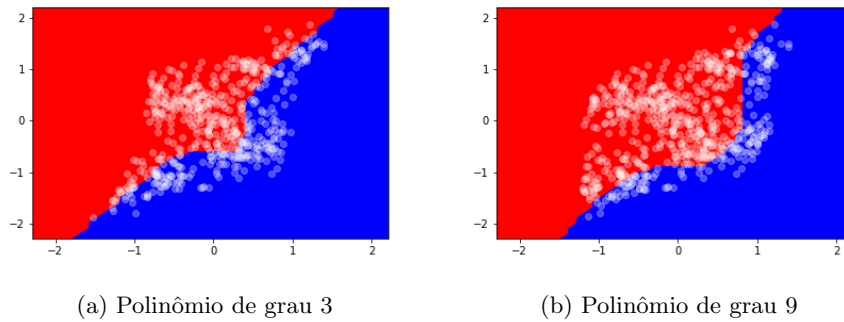


Figura 9: Regiões de decisão SVM, com kernel polinomial de grau 3 e 9.

A figura 10 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.

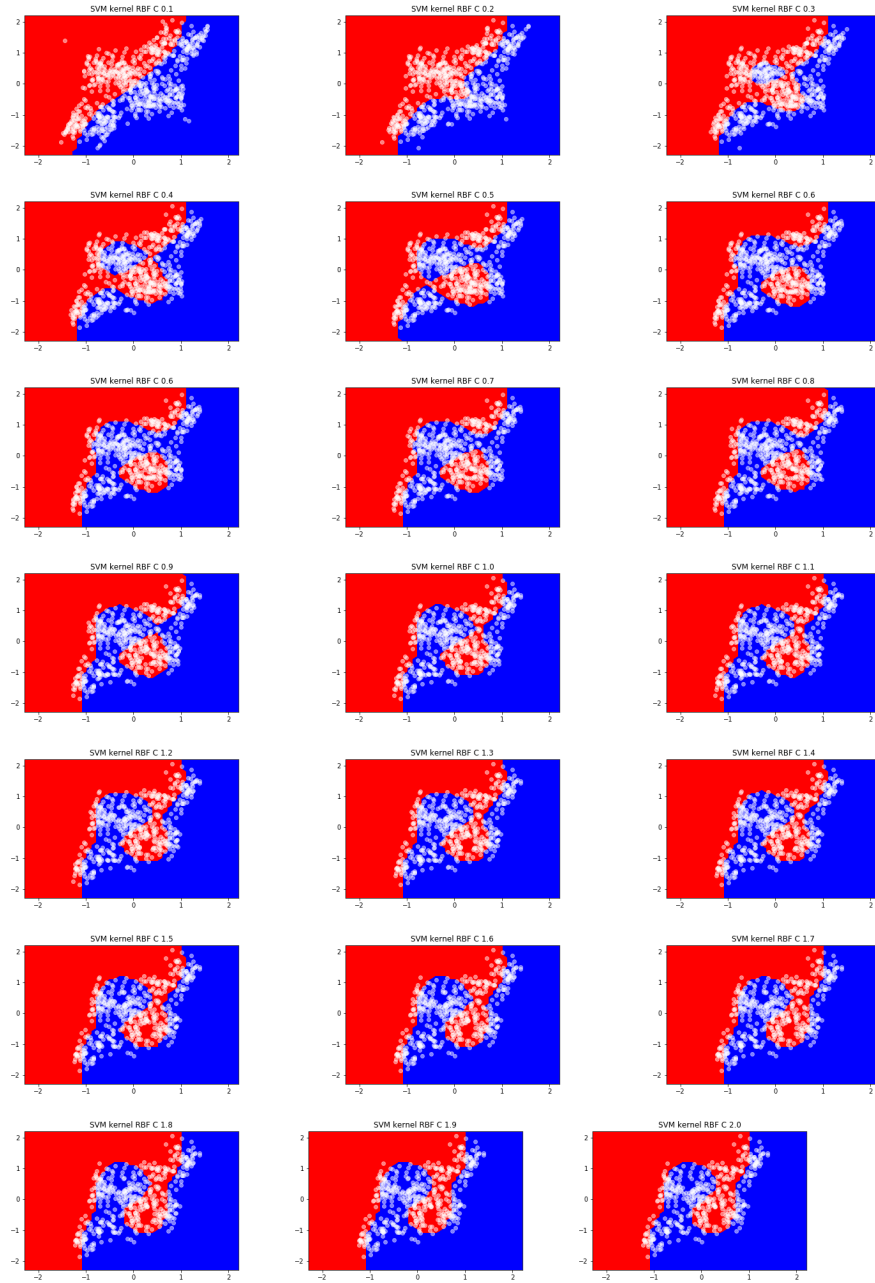


Figura 10: Regiões de decisão SVM, com kernel RBF para diversos C .