UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Reconhecimento de Padrões

Borges, C.D.B

REGRESSÃO 2D E 3D POR MÍNIMOS QUADRADOS E CLASSIFICAÇÃO VIA ELM, PERCEPTRON E MLP

SOBRAL

2019

QUESTÃO 1

Usando o conjunto de dados do aerogerador (variável de entrada: velocidade do vento – m/s, variável de saída: potência gerada – kW), determine o modelo de regressão polinomial (graus 2, 3, 4, 5 e 6) com parâmetros estimados pelo método dos mínimos quadrados. Avaliar a qualidade de cada modelo por meio do coeficiente de determinação (R^2).

A solução dessa questão encontra-se no script rp_mqreg2d.py, que aceita como parâmetro o valor k referente ao grau do polinômio de regressão. Os resultados obtidos com a execução dos scripts para diferentes valores de k são exibidos nos gráficos das Figuras 1, 2, 3, 4 e 5.

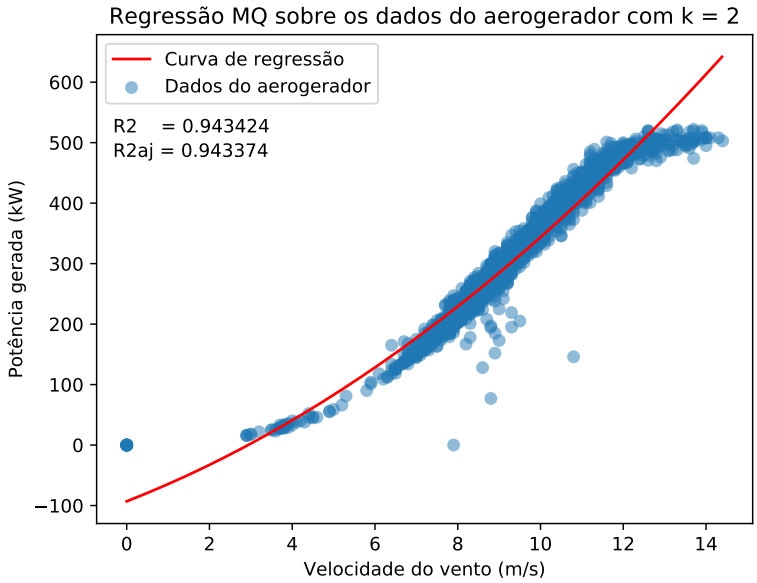

Figura 1 – Regressão polinomial de grau 2

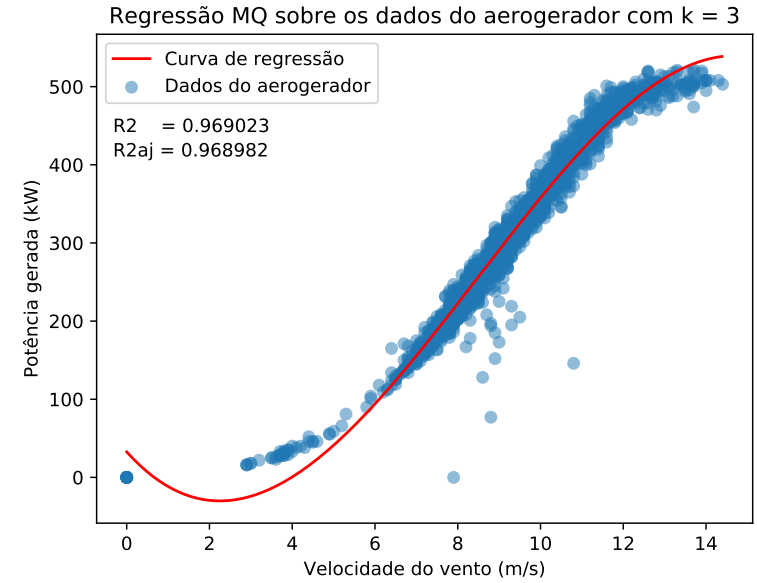

Figura 2 – Regressão polinomial de grau 3

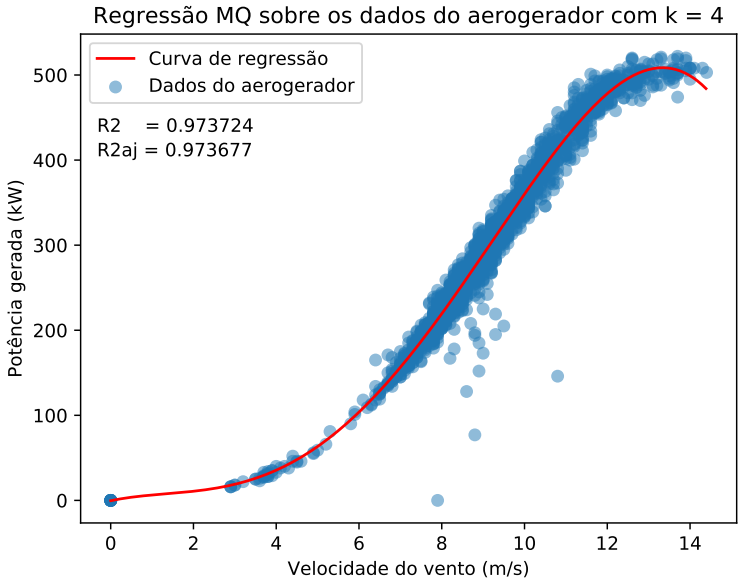

Figura 3 – Regressão polinomial de grau 4

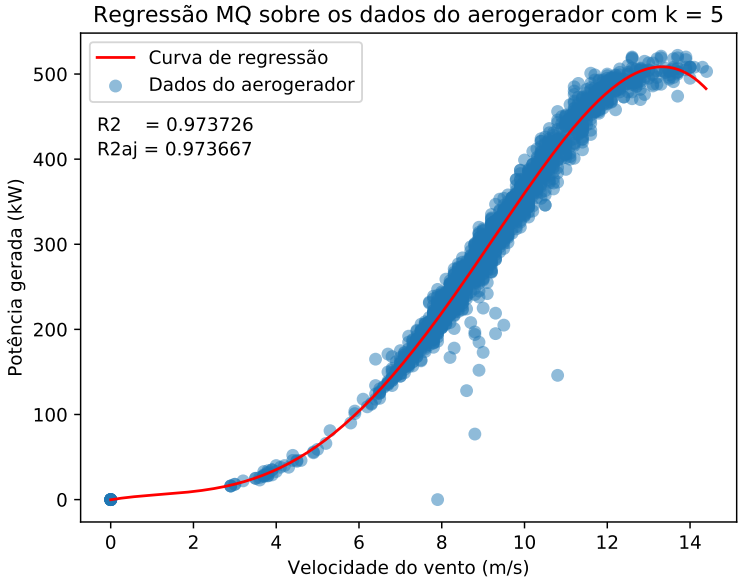

Figura 4 – Regressão polinomial de grau 5

Nota-se que valor de R^2 cresce conforme o aumento de k. R^2_{aj}, no entanto, apresenta uma tendência de crescimento para os valores $k = 2, 3, 4$, decresce com $k = 5$ e aumenta sutilmente com $k = 6$. A informação obtida com R^2 permite dizer que o encaixe da curva de regressão melhora conforme k aumenta. Dessa maneira, de acordo com R^2, o melhor modelo seria a regressão polinomial de grau 6. Apesar disso, o aumento de k gera modelos mais complexos e maior custo computacional de processamento e memória. O uso de R^2_{aj} permite balancear a qualidade do encaixe da curva de regressão e a complexidade do modelo. Segundo R^2_{aj}, o modelo mais adequado seria a regressão polinomial de grau 4.

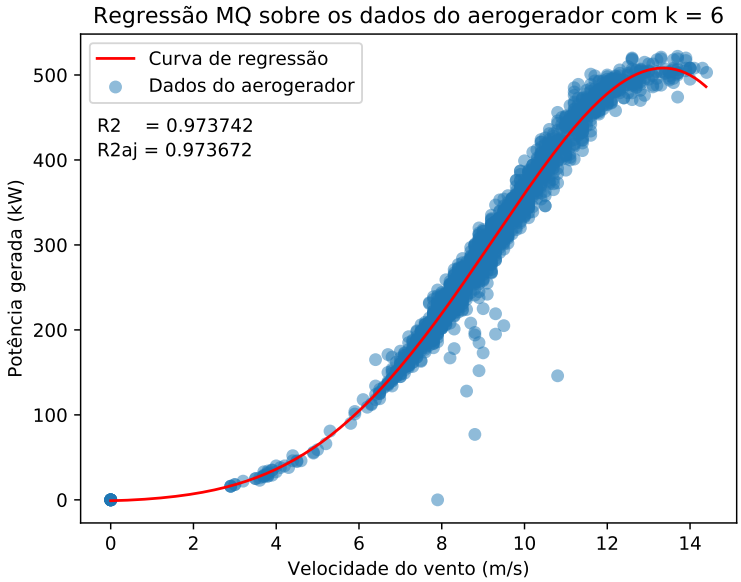

Figura 5 – Regressão polinomial de grau 6

QUESTÃO 2

Dada a base de dados abaixo, na qual a primeira e segunda colunas são as variáveis regressoras (x_1 e x_2) e a terceira coluna é a variável dependente (y), determine o modelo de regressão múltipla (plano) com parâmetros estimados pelo método dos mínimos quadrados. Avalie a qualidade do modelo por meio do coeficiente de determinação (R^2)

$$M = \begin{pmatrix} 122 & 139 & 0.115 \\ 114 & 126 & 0.120 \\ 86 & 90 & 0.105 \\ 134 & 144 & 0.090 \\ 146 & 163 & 0.100 \\ 107 & 136 & 0.120 \\ 68 & 61 & 0.105 \\ 117 & 62 & 0.080 \\ 71 & 41 & 0.100 \\ 98 & 120 & 0.115 \end{pmatrix}$$

A solução dessa questão encontra-se no arquivo rp_mqreg3d.py. Apesar de não ter sido solicitado no enunciado do problema, o script também aceita o valor de k como parâmetro de entrada e produz modelos polinomiais de diferentes graus.

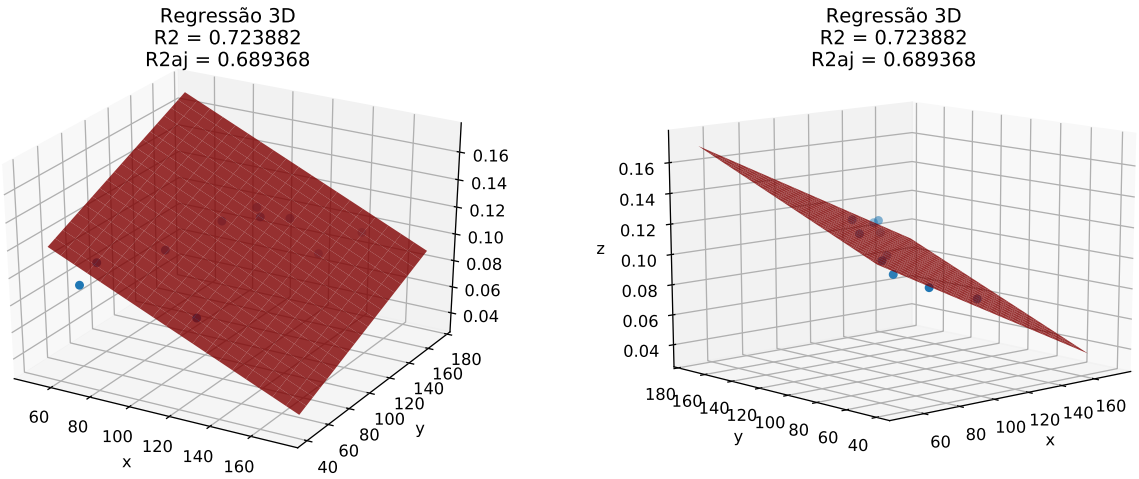

Figura 6 – Regressão usando polinômios de grau 1

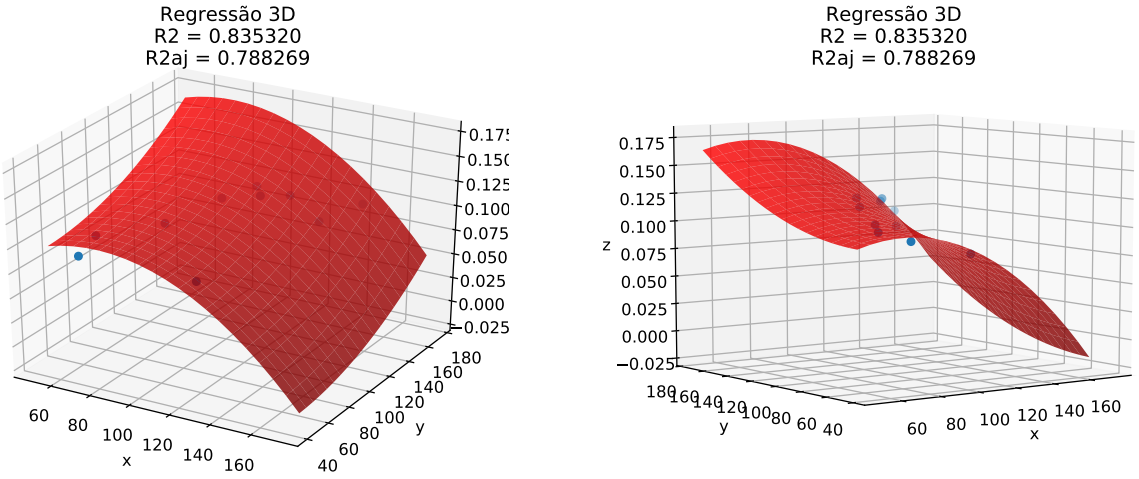

Figura 7 – Regressão usando polinômios de grau 2

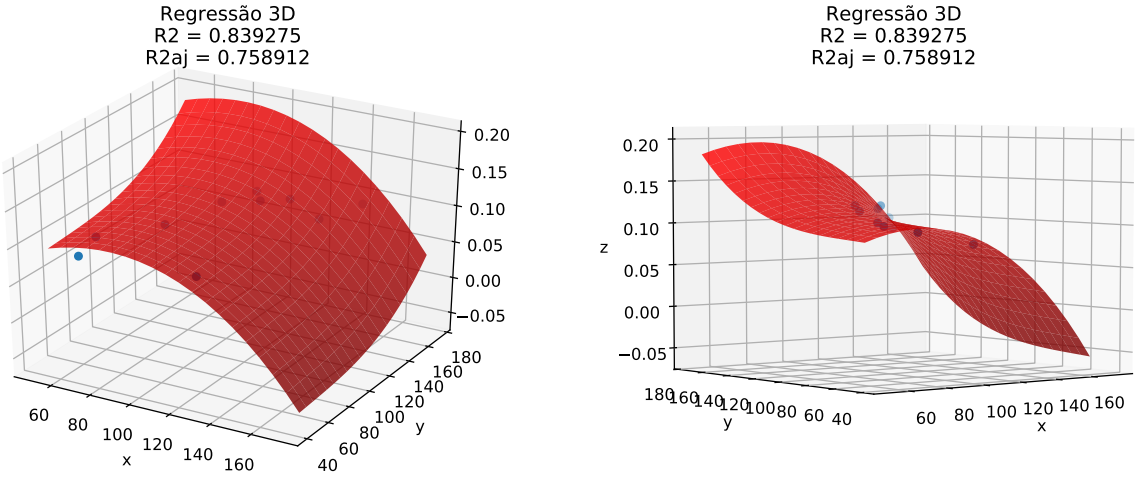

Figura 8 – Regressão usando polinômios de grau 3

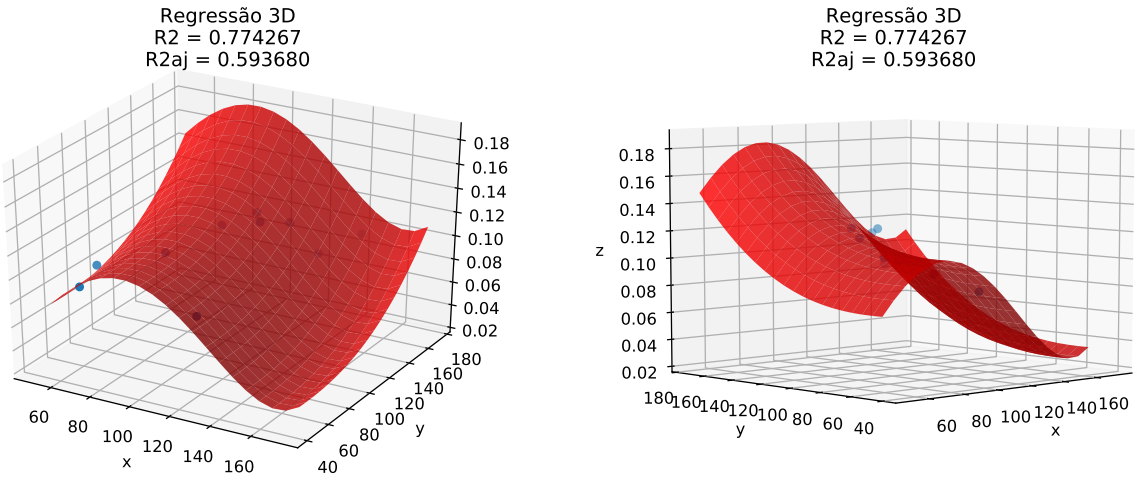

Figura 9 – Regressão usando polinômios de grau 4

Uma linha de raciocínio similar à da questão anterior pode ser aplicada aqui. R^2 apresenta crescimento para $k = 1, 2, 3$, mas decresce quando k chega a 4. O valor de R^2_{aj} cresce apenas entre $k = 1, 2$ e decresce a partir de $k = 3$. Visualmente, o plano produzido quando se utiliza $k = 1$ já apresenta um bom ajuste. Pode-se notar, também visualmente, a alta similaridade das superfícies produzidas com $k = 2$ e $k = 3$. Quando se usa $k = 4$, a superfície resultante já não se ajusta bem aos dados. Se utilizássemos R^2 como parâmetro de julgamento, decidiríamos pelo modelo de regressão de grau 3. Usando R^2_{aj}, o modelo mais adequando seria o de grau 2.

QUESTÃO 3

Classificar o conjunto de dados Dermatology, disponível em <http://archive.ics.uci.edu/ml/datasets/Dermatology>, conforme as instruções a seguir:

- *Classificadores: ELM, Perceptron, MLP (implementar ELM e Perceptron);*
- *Estratégia de validação cruzada: leave-one-out e 5-fold;*
- *Testar com e sem o zscore;*
- *Montar a matriz de confusão para cada classificação usando os dados de testes.*

As soluções para essa questão encontram-se distribuídas em diversos arquivos. A implementação da ELM está em rp_elm_loo_derm.py e rp_elm_kfold_derm.py. Ambos os scripts aceitam o parâmetro -q (número de neurônios). O perceptron está implementado nos scripts rp_perceptron_loo_derm.py e rp_perceptron_kfold_derm.py e aceita os parâmetros -a (taxa de aprendizagem) e -e (número de épocas). Enfim, a MLP foi implementada usando os pacotes keras e tensorflow nos arquivos rp_mlp_loo_derm.py e rp_mlp_kfold_derm.py. Os parâmetros -q (número de neurônios) e -e (número de épocas) estão disponíveis. Para todos os scripts, existe a opção de normalizar os dados com zscore usando --normalize.

	10 Neurônios		30 Neurônios		50 Neurônios	
	SN	N	SN	N	SN	N
Leave-One-Out	46.93	82.12	65.64	95.25	72.07	95.53
5-Fold	42.18	82.40	55.87	94.41	66.20	94.41

Tabela 1 – Desempenho da ELM na classificação da base Dermatology

	1 Época		10 Épocas		20 Épocas	
	SN	N	SN	N	SN	N
Leave-One-Out	58.10	86.31	84.08	91.90	90.78	91.90
5-Fold	48.04	83.80	72.91	90.50	71.79	89.94

Tabela 2 – Desempenho do Perceptron na classificação da base Dermatology

	10 Neurônios		30 Neurônios		50 Neurônios	
	SN	N	SN	N	SN	N
Leave-One-Out	67.88	81.28	91.62	95.53	94.69	96.65
5-Fold	62.85	77.37	88.55	93.02	96.37	96.37

Tabela 3 – Desempenho da MLP na classificação da base Dermatology

As Tabelas 1, 2 e 3 mostram os resultados de diversas execuções da ELM, Perceptron e MLP, respectivamente, sobre a base Dermatology. A ELM e a MLP (1 camada oculta) foram validadas com 10, 30 e 50 neurônios em 20 épocas de treinamento. Já que o número de neurônios do Perceptron é fixo, esse foi validado com variação do número de épocas. Para todos os casos, foram feitas validações Leave-One-Out e 5-Fold, sem normalização (SN) e com normalização (N).

Observa-se na Tabela 1 que a ELM apresentou baixo desempenho nos experimentos sem normalização. Houve melhora significativa quando os dados foram apresentados à rede normalizados com zscore. O aumento do número de neurônios teve um impacto positivo no desempenho, no entanto, pode-se ver que ocorreu saturação quando as taxas de acerto atingiram a faixa de 95%. Incrementar o número de neurônios de 30 para 50 resultou em um ganho mínimo na acurácia, portanto, nesse caso, o modelo ELM com 30 neurônios pode ser dito o mais adequado. As matrizes de confusão para essa ELM são exibidas nas Tabelas 4, 5, 6 e 7.

Na Tabela 2 mostra-se que o Perceptron apresentou um comportamento similar à ELM em relação à normalização dos dados: o zscore teve impacto positivo significativo. O acréscimo de épocas de treinamento também teve impacto positivo, com ocorrência de saturação na faixa de 90% de acurácia. Treinar o Perceptron por 20 épocas não refletiu um aumento nas taxas de acerto em comparação com os resultados obtidos com apenas 10 épocas. Assim, o Perceptron treinado por 10 épocas foi selecionado. Nas Tabelas 8, 9, 10 e 11 são mostradas as matrizes de confusão para esse Perceptron.

Na Tabela 3 são exibidos os resultados obtidos com uma MLP com camada de entrada de tamanho 35 (34 atributos + 1 bias); uma camada oculta contendo os números especificados de neurônios; e a camada de saída com 6 neurônios (um para cada classe). Novamente pode ser observada a influência da normalização: o zscore melhorou a acurácia de forma expressiva para os casos de 10 e 30 neurônios. O acréscimo de neurônios também teve impacto positivo sobre o desempenho da MLP, que atingiu cerca de 96% quando foram utilizados 50 neurônios em sua camada oculta. Nesse ponto, a normalização dos dados já não teve um impacto significativo: a MLP foi capaz de se adaptar bem aos dados normalizados e não normalizados, ao contrário da ELM, que mesmo com o uso de 50 neurônios, não assimilou bem os dados não normalizados. Neste caso, devido às melhorias consideráveis observadas com o incremento da camada oculta, foi selecionada a MLP com 50 neurônios. As matrizes de confusão para esse modelo podem ser visualizadas nas Tabelas 12, 13, 14 e 15.

	Classes reais					
65.64%	1	2	3	4	5	6
Classes preditas 1	96	14	3	13	13	3
2	2	28	1	13	4	2
3	6	10	65	5	2	2
4	2	5	0	13	3	5
5	3	3	2	4	25	0
6	2	0	0	0	1	8

Tabela 4 – Matriz de confusão da ELM de 30 neurônios e dados não normalizados (Leave-One-Out)

	Classes reais					
55.87%	1	2	3	4	5	6
Classes preditas 1	82	27	12	24	16	6
2	8	25	3	8	9	0
3	9	4	55	11	1	0
4	2	2	0	3	0	0
5	3	1	1	2	22	1
6	7	1	0	0	0	13

Tabela 5 – Matriz de confusão da ELM de 30 neurônios e dados não normalizados (5-Fold)

	Classes reais					
95.25%	1	2	3	4	5	6
Classes preditas 1	110	0	0	0	1	0
2	0	50	0	3	1	0
3	0	0	71	0	0	0
4	1	8	0	44	0	0
5	0	2	0	1	46	0
6	0	0	0	0	0	20

Tabela 6 – Matriz de confusão da ELM de 30 neurônios e dados normalizados (Leave-One-Out)

	Classes reais					
94.41%	1	2	3	4	5	6
Classes preditas 1	110	0	0	0	1	0
2	0	51	0	5	3	0
3	0	1	71	0	0	0
4	1	7	0	43	1	0
5	0	1	0	0	43	0
6	0	0	0	0	0	20

Tabela 7 – Matriz de confusão da ELM de 30 neurônios e dados normalizados (5-Fold)

Classes preditas \ Classes reais	84.08%	1	2	3	4	5	6	-1
	1	110	0	0	0	0	0	0
	2	0	40	0	2	0	0	0
	3	0	0	71	0	0	0	0
	4	0	3	0	22	0	0	0
	5	0	0	0	0	45	0	0
	6	0	0	0	0	0	13	0
	-1	1	17	0	24	3	7	0

Tabela 8 – Matriz de confusão do Perceptron usando dados não normalizados e 10 épocas (Leave-One-Out)

Classes preditas \ Classes reais	72.91%	1	2	3	4	5	6	-1
	1	105	0	0	0	0	0	0
	2	0	17	0	2	0	0	0
	3	0	0	71	0	0	0	0
	4	0	2	0	8	0	0	0
	5	0	0	0	0	46	0	0
	6	0	0	0	0	0	14	0
	-1	6	41	0	38	2	6	0

Tabela 9 – Matriz de confusão do Perceptron usando dados não normalizados e 10 épocas (5-Fold)

Classes preditas \ Classes reais	91.90%	1	2	3	4	5	6	-1
	1	106	0	0	0	0	0	0
	2	0	51	0	6	0	0	0
	3	0	0	68	0	0	0	0
	4	0	2	0	37	0	0	0
	5	0	0	0	0	47	0	0
	6	0	0	0	0	0	20	0
	-1	5	7	3	5	1	0	0

Tabela 10 – Matriz de confusão do Perceptron usando dados normalizados e 10 épocas (Leave-One-Out)

Classes preditas \ Classes reais	90.50%	1	2	3	4	5	6	-1
	1	108	1	0	0	0	0	0
	2	0	46	0	5	0	0	0
	3	0	0	65	0	0	0	0
	4	0	4	0	40	0	0	0
	5	0	0	0	0	46	0	0
	6	0	0	0	0	0	19	0
	-1	3	9	6	3	2	1	0

Tabela 11 – Matriz de confusão do Perceptron usando dados normalizados e 10 épocas (5-Fold)

As células das matrizes de confusão são exibidas em cores vermelhas de intensidade proporcional ao número de amostras nela contido. Se a classificação fosse perfeita, apenas a diagonal principal estaria em vermelho. Se a confusão do modelo for alta, as células aparecem fortemente coloradas em posições fora da diagonal principal, como pode ser visto nas Tabelas 4 e 5. Nas Tabelas 8, 9, 10 e 11, a classe -1 foi utilizada para representar as instâncias de teste na qual o Perceptron não pôde realizar uma decisão classificatória coerente. Esses são os casos em que nenhum ou mais de um neurônio são ativados na saída.

Um fato interessante, extraído das matrizes de confusão, é a aparente dificuldade de diferenciar as classes 2 e 4. Isso pode ser visto claramente nas Tabelas 6 e 7 (ELM), 8, 9, 10 e 11 (Perceptron) e 12, 13, 14 e 15 (MLP). Nota-se nas Tabelas 8 e 9 que as duas classes mais frequentemente não classificadas pelo Perceptron são justamente as classes 2 e 4.

Classes preditas \ Classes reais	94.69%	1	2	3	4	5	6
1		111	0	0	0	0	0
2		0	57	0	16	0	0
3		0	0	71	0	0	0
4		0	3	0	32	0	0
5		0	0	0	0	48	0
6		0	0	0	0	0	20

Tabela 12 – Matriz de confusão da MLP com 50 neurônios e dados não normalizados (Leave-One-Out)

Classes preditas \ Classes reais	96.37%	1	2	3	4	5	6
1		111	0	0	0	0	0
2		0	59	0	12	0	0
3		0	0	71	0	0	0
4		0	1	0	36	0	0
5		0	0	0	0	48	0
6		0	0	0	0	0	20

Tabela 13 – Matriz de confusão da MLP com 50 neurônios e dados não normalizados (5-Fold)

Classes preditas \ Classes reais	96.65%	1	2	3	4	5	6
1		111	1	0	0	0	0
2		0	54	0	6	0	0
3		0	0	71	0	0	0
4		0	5	0	42	0	0
5		0	0	0	0	48	0
6		0	0	0	0	0	20

Tabela 14 – Matriz de confusão da MLP com 50 neurônios e dados normalizados (Leave-One-Out)

Classes preditas \ Classes reais	96.37%	1	2	3	4	5	6
1		111	1	0	0	1	0
2		0	55	0	7	0	0
3		0	0	71	0	0	0
4		0	4	0	41	0	0
5		0	0	0	0	47	0
6		0	0	0	0	0	20

Tabela 15 – Matriz de confusão da MLP com 50 neurônios e dados normalizados (5-Fold)