PALESTRA

REGRESSÃO POLINOMIAL E REDES NEURAIS ARTIFICIAIS

Norberto Hochheim

- Engenheiro Civil (UFSC)
- Mestre em Engenharia de Produção (UFSC)
- Doutor pela *Université de Nancy I* (França)
- Professor Titular da UFSC
- Coordenador do Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial (PPGTG)
- Coordenador do Grupo de Pesquisa em Engenharia de Avaliações e Perícias (GEAP)

Promoção: IBAPE NACIONAL Instituto Brasileiro de Avaliações e Perícias de Engenharia

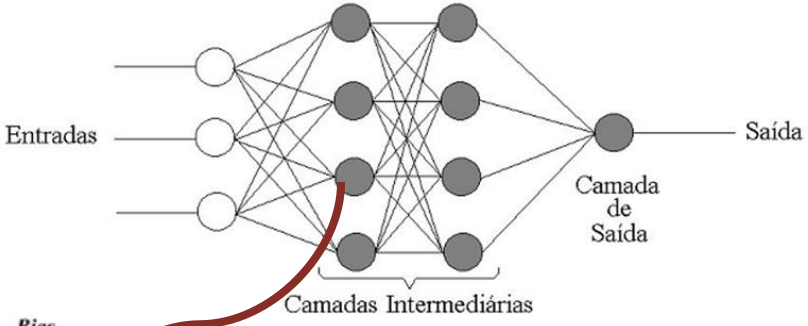
Realização: IBAPE BAHIA Instituto Brasileiro de Avaliações e Perícias de Engenharia da Bahia

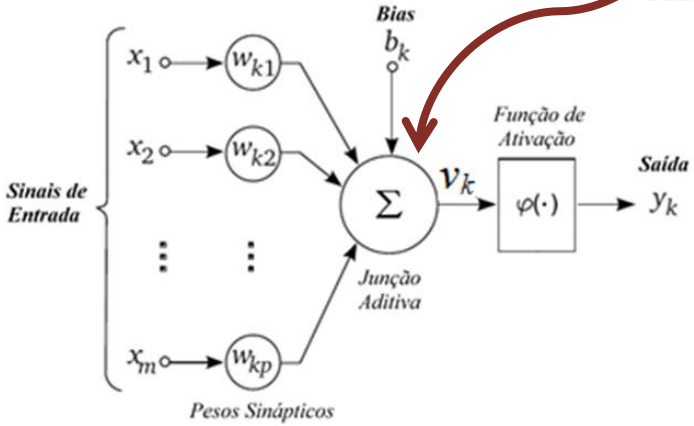
REGRESSÃO POLINOMIAL E REDES NEURAIS ARTIFICIAIS

Objetivos:

- **Discorrer sobre regressão polinomial (RP) e redes neurais (RNs)**
  - **Diferenças**
  - **Similaridades**
- **Apresentar estudo de caso**

REDES NEURAIS ARTIFICIAIS



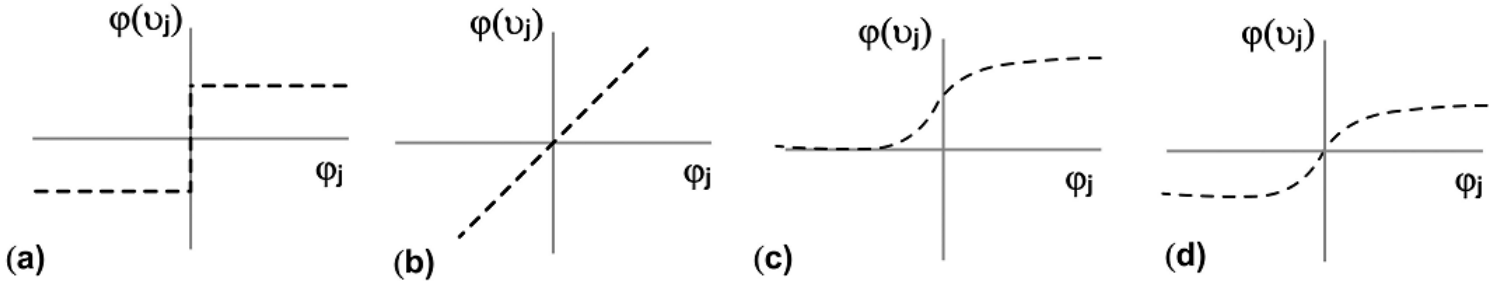


$$v_k = \sum_{j=1}^{n} w_{kj} \cdot x_j + b_k$$

$$y_k = \varphi(v_k)$$

Fonte: Guarnieri (2006) e Haykin (2001)

Função de ativação (φ): limita o intervalo permissível de amplitude do sinal de saída (y_k)



(a) degrau (b) linear (c) logística (d) tanh

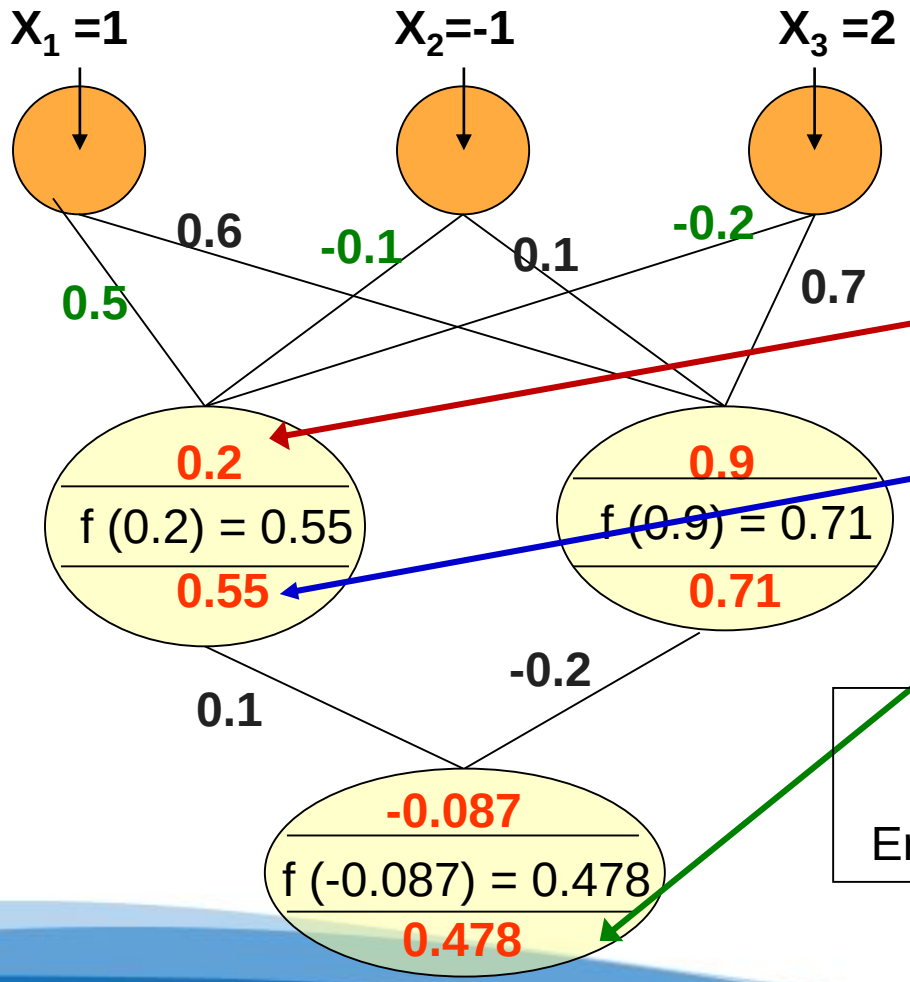
Fonte: Fiorin *et al* (2011)

APRENDIZAGEM DE UMA REDE NEURAL

RNs aprendem por intermédio de exemplos e analisam os seus resultados, melhorando gradativamente o seu desempenho

- **Para a aprendizagem, as RNs usam um algoritmo cuja tarefa é ajustar os pesos de suas conexões**

- ***Overfitting*: a rede 'decora' os dados, ao invés de aprender os seus padrões (perde capacidade de generalização)**



Entrada: $X_1\ X_2\ X_3$

Saída: Y

Modelo: $Y = f(X_1\ X_2\ X_3)$

$0.2 = 0.5 * 1 - 0.1*(-1) - 0.2 * 2$

$f(x) = e^x / (1 + e^x)$

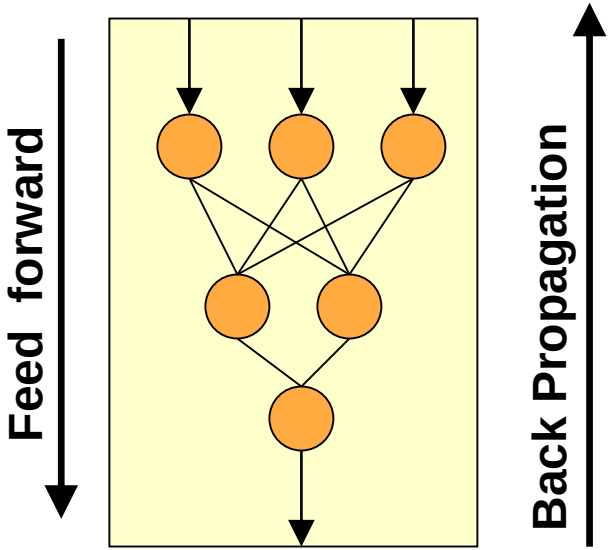
$f(0.2) = e^{0.2} / (1 + e^{0.2}) = 0.55$

Y previsto = 0.478

Se Y = 2
então
Erro = (2-0.478) =1.522

Fonte: Angshuman Saha (2003)

RNA: Treinamento do modelo



$E = \Sigma (Yi - Vi)^2$

- **Inicie** com um conjunto de pesos aleatórios
- Passe a **primeira observação** (Y_1) pela rede
  Valor estimado:V_1 Erro = ($Y_1 - V_1$)
- **Ajuste de pesos** de maneira a reduzir este erro
  (modelo faz um bom ajuste para a primeira observação)
- Passe a **segunda observação** (Y_2) pela rede
  (**Ajuste de pesos** para um bom ajuste da segunda observação)
- **Repita** a observação até a última observação (Y_n)
- **Fim** do primeiro ciclo de treinamento
- **Repita os ciclos de teinamento até que o erro do modelo (E) seja pequeno para todo o conjunto de dados**

Fonte: Angshuman Saha (2003)

RNA: Treinamento do modelo – Critério de Convergência

Dividir os dados em 2 conjuntos:

- **de treinamento: para construir o modelo**
- **de validação: para testar a performance do modelo**

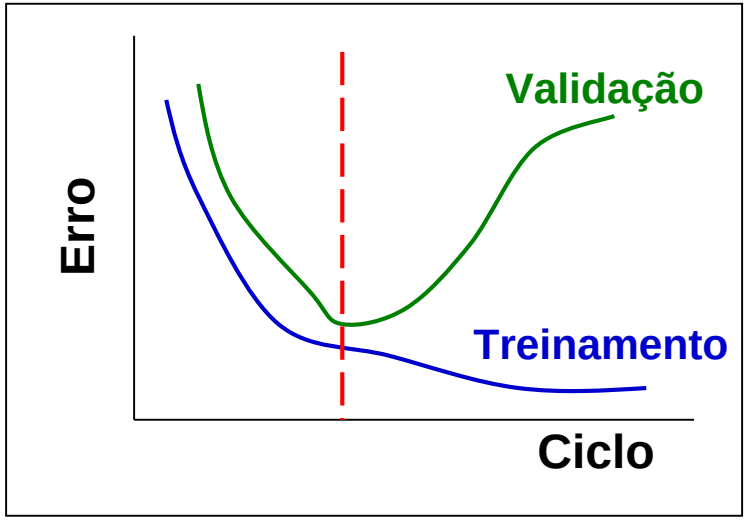
Ao longo dos ciclos de treinamento:

✓ **Erro no conjunto de treinamento diminui**

✓ **Erro no conjunto de validação primeiro cai e depois aumenta (overfitting)**

Fonte: Angshuman Saha (2003)

RNA: Treinamento do modelo – Critério de Convergência



- **Termine** o treinamento quando o **erro** no conjunto de **validação aumenta**

Fonte: Angshuman Saha (2003)

Regressão Polinomial (RP)

RLS

$$y = f(x_1) = \beta_0 + \beta_1 x_1 + e$$

RLM

$$y = f(x_1, x_2, \ldots, x_m) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m + e$$

RP

$$y = f(x_1) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^{\,2} + \ldots + \beta_m x_1^{\,m} + e$$

RP

$$y = f(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^{\,2} + \beta_4 x_2^{\,2} + \beta_5 x_1 x_2 + e$$

REGRESSÃO POLINOMIAL E REDES NEURAIS ARTIFICIAIS

Matloff et al. (2018), em seu artigo

"Polynomial regression as an alternative to neural nets"

afirmam que redes neurais são uma forma de regressões polinomiais

Redes Neurais (RN) são Regressões Polinomiais (RP)

Matloff et al. (2018)

1º Caso: Redes Neurais (RN) com

Funções de Ativação polinomiais

O argumento básico: Considere o caso com $p = 2$. Sejam as variáveis u and v. As entradas para a primeira camada oculta, incluindo o nó "1", serão da forma:

$$a00 + a01u + a02v \quad \text{e} \quad a03 + a04u + a05v$$

Exemplo: função de ativação $a(t) = t^2$

- **Saídas da primeira camada serão funções quadráticas de u e v**
- **Segunda camada produzirá polinômios de grau 4**
- **Depois, de grau 8 e assim por diante**
- **Funciona da mesma maneira para qualquer função de ativação polinomial**
- **Assim, para uma função de ativação polinomial, treinar uma rede neural é a mesma coisa que uma regressão polinomial**

Redes Neurais (RN) são Regressões Polinomiais (RP)

Matloff et al. (2018)

2º Caso: Redes Neurais (NN) com Funções de Ativação Transcendentais

Considerando agora funções de ativação mais realistas, como funções transcendentais do tipo *tanh*()

Implementação computacional frequentemente usa séries de Taylor

Nestes casos, a situação é a mesma da anterior, e RNs são novamente RP

Redes Neurais (RN) são Regressões Polinomiais (RP)

Matloff et al. (2018)

3º Caso: Redes Neurais (NN) com Funções de Ativação Genéricas

Funções de ativação genéricas e implementações são próximas de um polinômio, de acordo com o Teorema de Stone-Weierstrass, que estabelece que qualquer função contínua de um conjunto delimitado pode ser aproximado por polinômios

Em qualquer situação, para objetivos práticos, a maioria das funções de ativação podem ser aproximadas por um polinômio

REGRESSÃO POLINOMIAL E REDES NEURAIS ARTIFICIAIS

Resumindo (Matloff et al., 2018):

- **redes neurais são uma forma de regressões polinomiais**
- **para cada camada oculta de uma rede neural, há um modelo de regressão polinomial equivalente**
- **O grau da aproximação polinomial aumenta de camada para camada da rede neural**

Por que existem problemas de convergência nas Redes Neurais?

- Modelos de RP tendem a produzir multicolinearidade quando em graus maiores
- Ver RNs como um tipo de regressão polinomial sugere que elas também sofrem de problemas de colinearidade
- Esta colinearidade aumenta de uma camada para outra, pois ela tende a aumentar com o grau do polinômio
- Assim, RNs podem sofrer de um problema de colinearidade oculto
- Isto pode resultar em problemas de convergências nas RNs

Fonte: Matloff et al. (2018)

O que fazer para evitar problemas de convergência em RNs?

- Seria interessante incluir nos softwares de RNs checagem para multicolinearidade, camada por camada
- Caso a camada produza um alto grau de colinearidade, pode-se considerar diminuir o número de neurônios dela, ou mesmo eliminar esta camada por completo
- Assim, camadas posteriores podem ter menos neurônios do que camadas anteriores

Fonte: Matloff et al. (2018)

Limitações da RP:
Memória, tempo de processamento e multicolinearidade

- **Redução da dimensão via Análise de Componentes Principais (ACP) pode remediar problemas de memória, tempo de processamento e multicolinearidade**
- **Outra possibilidade: aplicar ACP para a matriz dos termos polinomiais, ao invés dos termos originais**
- **Multicolinearidade também pode ser tratada com Regressão Ridge ou LASSO (Least Absolute Shrinkage and Selection Operator)**
- **Multicolinearidade também pode ocorrer em RNs**
- **Multicolinearidade pode provocar *overfitting* em RNs ou RP**

Fonte: Matloff et al. (2018)

RP como alternativa à RN

- **Matloff *et. al.*(2018) aplicaram RP e RNs através de várias implementações (Keras, DeepNet, NeuralNet) a vários conjuntos de dados**
- **Em todos os casos, RPs tiveram performance similar ou superior às RNs**
- **Exemplo: Resistência à compressão do concreto**
  1030 dados, 8 variáveis

Método	Correlação (Previsto vs. Observados)
NeuralNet	0,608
Keras Formula	0,546
PR, grau 2	0,869

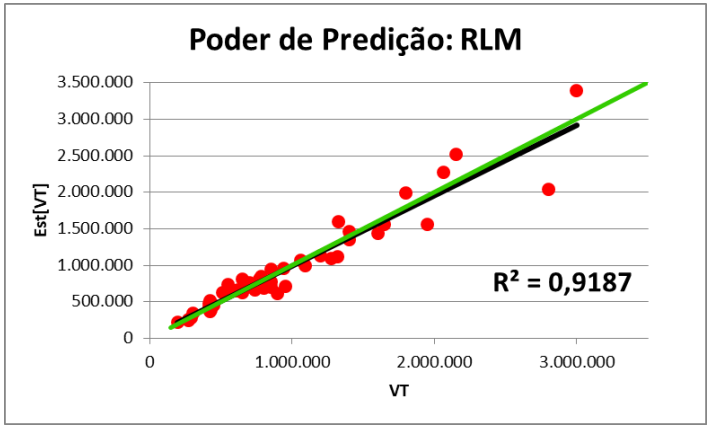
ESTUDO DE CASO: Aptos Florianópolis

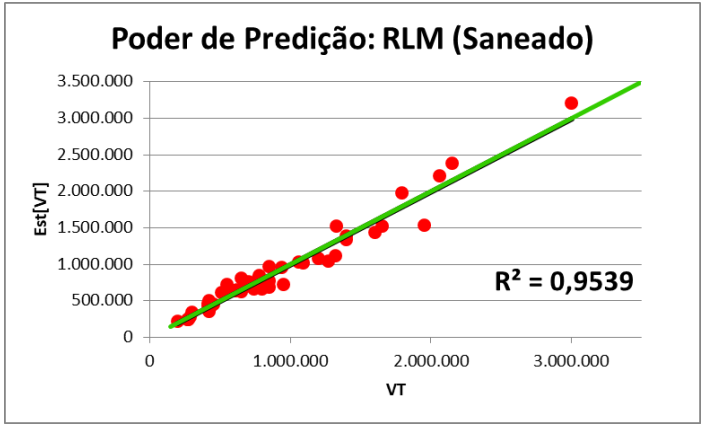
Dados coletados em Abril/2015

ESTUDO DE CASO: Aptos Florianópolis

Ident	VT	Área Total	nº Quartos	nº Suítes	nº Garagens	Dist B Mar	Padrão	Coord E	Coord N
AP_01	1.060.000,00	350,00	3	1	2	720	2	741.617	6.945.437
AP_02	510.000,00	136,56	3	1	1	665	2	741.470	6.945.464
AP_03	780.000,00	164,77	3	1	2	415	2	741.562	6.945.757
AP_04	550.000,00	174,58	3	1	1	320	2	741.049	6.945.701
AP_05	850.000,00	123,01	3	1	3	895	3	741.859	6.945.340
AP_06	300.000,00	89,83	2	0	1	645	1	741.180	6.945.388
AP_07	750.000,00	174,00	2	1	2	860	3	741.989	6.945.407
AP_08	650.000,00	123,00	3	1	1	745	3	741.053	6.945.154
AP_09	620.000,00	121,00	3	1	1	745	3	741.009	6.945.298
AP_10	740.000,00	109,00	3	1	1	300	2	741.019	6.945.704
...	...	...	...	...	...	...	...	...	...
AP_49	1.650.000,00	246,00	3	3	3	307	3	742.188	6.946.149
AP_50	650.000,00	159,72	3	1	1	120	2	740.940	6.945.889
Aval	???	205,00	3	1	2	250	2	741.506	6.945.904

Regressão Linear Múltipla

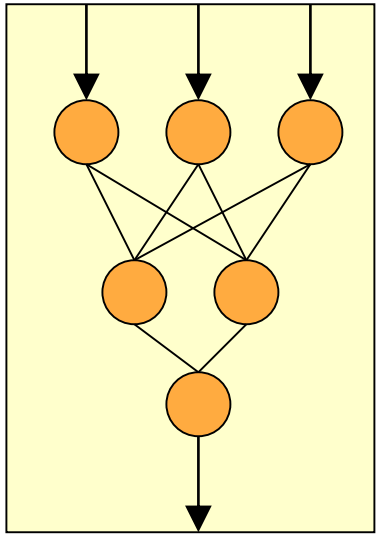




Estatísticas (Resíduos) RLM	
Desvio médio Abs	118.431,60
Desvio médio Rel	11,4%
RMSE	179.634,41
Valor Estimado:	962.663,91

Estatísticas (Resíduos) RLM (Saneado)	
Desvio médio Abs	93.855,89
Desvio médio Rel	10,3%
RMSE	126.250,22
Valor Estimado:	961.660,64

Rede Neural - *Software* NN: Angshuman Saha



Amostra:

50 dados, 6 variáveis explicativas

Modelo:

Camada oculta: 1

Parâmetro de aprendizado: 0,4

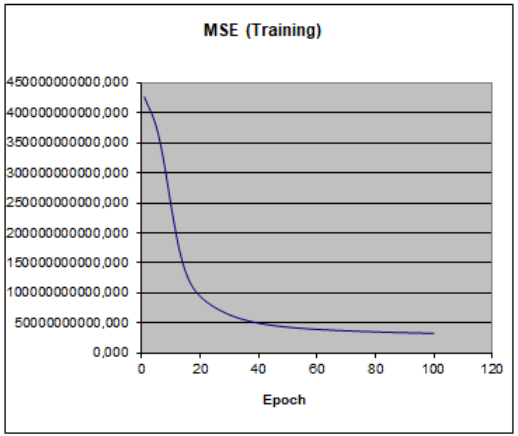
Momentum: 0,4

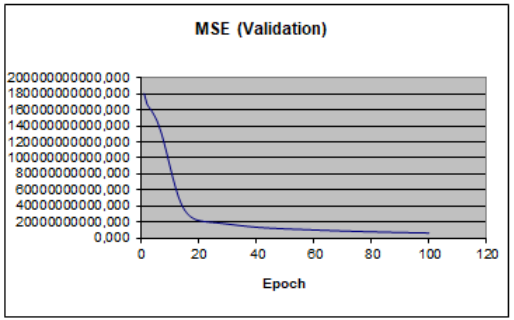
Partição: 10% dos dados para validação (seleção aleatória)

Rede Neural - *Software* NN: Angshuman Saha

Fonte: www.geocities.com/adotsaha/NNinExcel.html

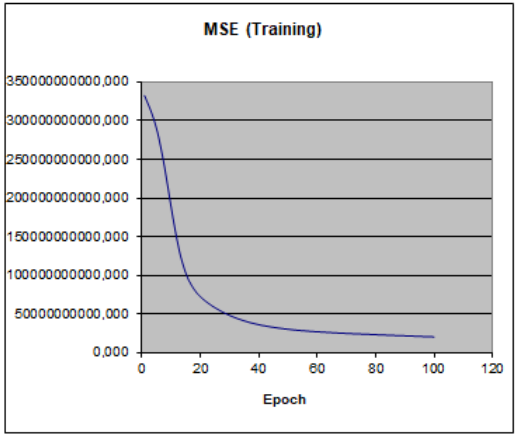
Amostra original

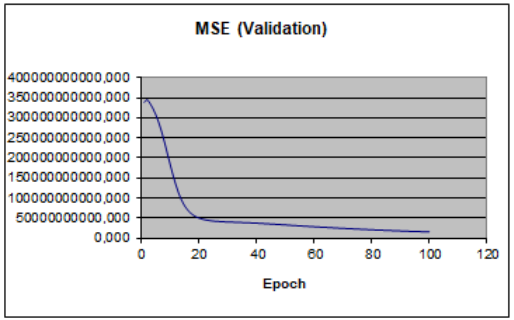




Validação: 14,95%
Treinamento: 9,49%

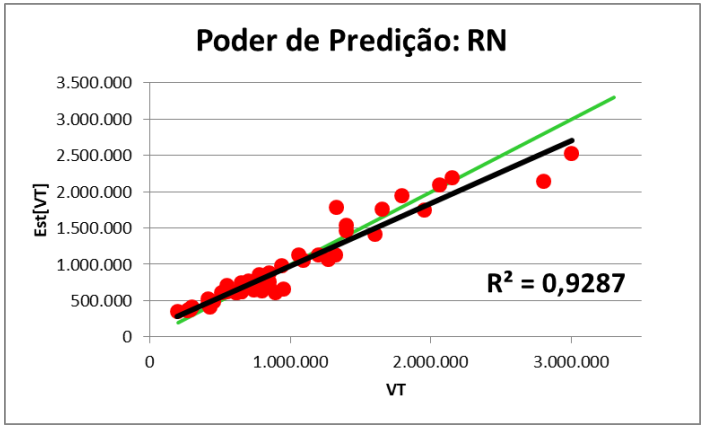
Amostra saneada

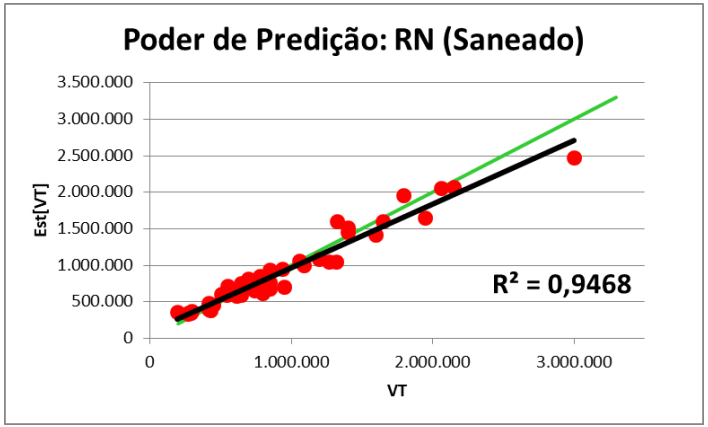




Validação: 13,21%
Treinamento: 7,48%

Rede Neural





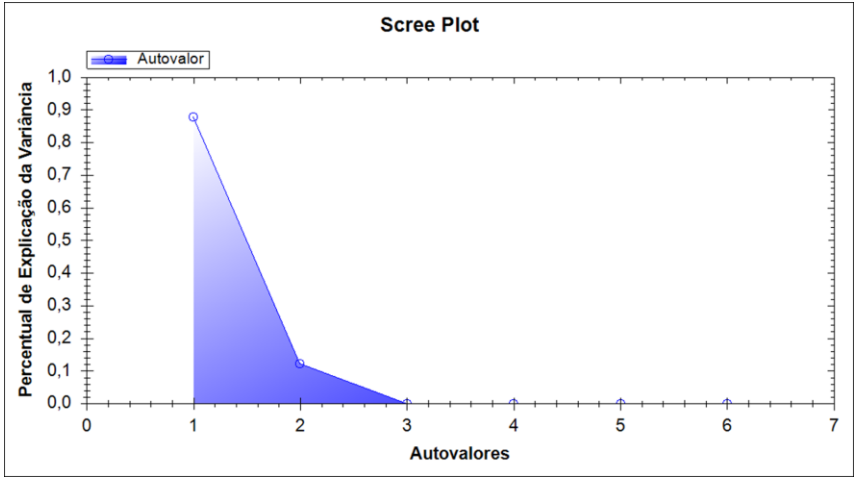
Estatísticas (Resíduos)	
RN	
Desvio médio Abs	119.446,67
Desvio médio Rel	14,1%
RMSE	172.314,90
Valor Estimado:	1.013.207,17

Estatísticas (Resíduos)	
RN (Saneado)	
Desvio médio Abs	102.400,66
Desvio médio Rel	12,8%
RMSE	141.013,49
Valor Estimado:	965.886,27

Regressão Polinomial

Número grande de variáveis:

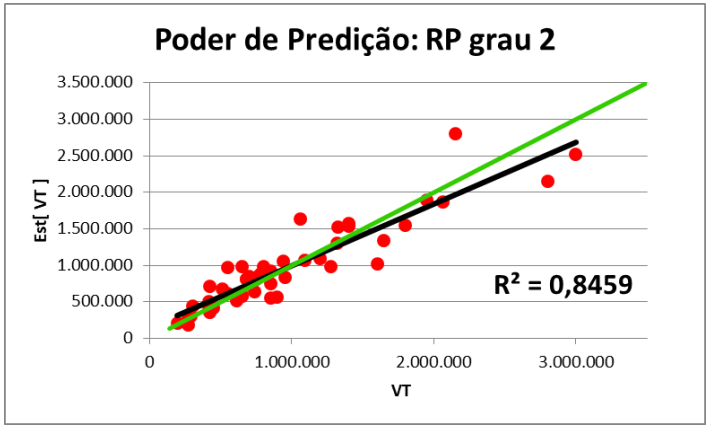
redução dimensional via Análise de Componentes Principais (ACP)

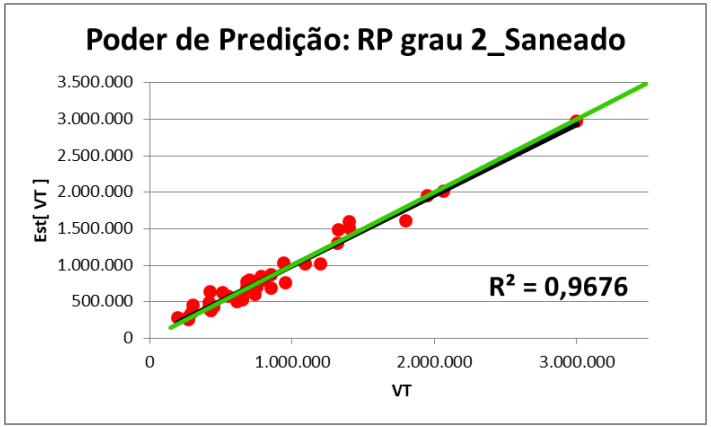


Fonte:
Software
IpeaGEO

IDENT	VT	Area_Total	N_Quartos	N_Suites	N_Garagens	Dist_BMar	Padrao	Comp_1	Comp_2
AP_01	**1.060.000,00**	350	3	1	2	720	2	**156,314**	**180,391**
AP_02	**510.000,00**	136,56	3	1	1	665	2	**124,148**	**-37,659**
AP_03	**780.000,00**	164,77	3	1	2	415	2	**-127,433**	**-35,989**
AP_04	**550.000,00**	174,58	3	1	1	320	2	**-222,937**	**-36,268**
AP_05	**850.000,00**	123,01	3	1	3	895	3	**354,291**	**-26,841**

Regressão Polinomial – Grau 2





Estatísticas (Resíduos) RP grau 2	
Desvio médio Abs	175.638,70
Desvio médio Rel	19,3%
RMSE	243.755,04
Valor Estimado:	1.153.131,16

Estatísticas (Resíduos) RP grau 2 (Saneado)	
Desvio médio Abs	81.700,91
Desvio médio Rel	13,0%
RMSE	101.644,68
Valor Estimado:	1.129.864,05

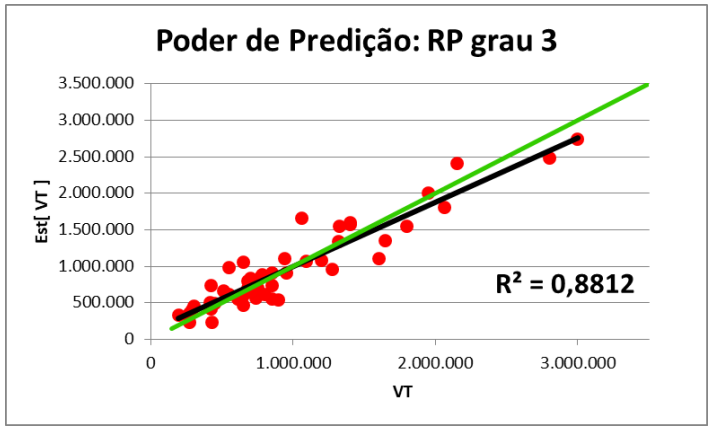
Regressão Polinomial – Grau 2

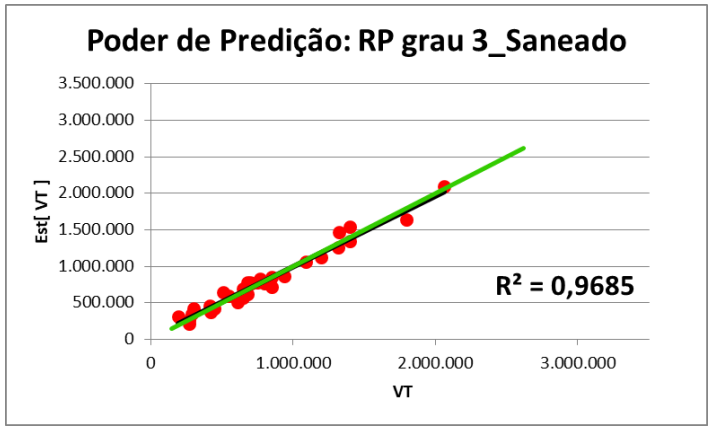
Fator de Inflação da Variância (FIV)

Variance Inflation Factor (VIF)
Valor limite: 10

Variável:	CP1	CP2	CP1^2	CP2^2	CP1*CP2
RP grau 2	1,43	2,51	1,17	2,65	1,55
RP grau 2 (Saneado)	1,63	1,78	1,24	1,65	1,50

Regressão Polinomial – Grau 3





Estatísticas (Resíduos)	
RP grau 3	
Desvio médio Abs	166.747,98
Desvio médio Rel	22,0%
RMSE	214.007,94
Valor Estimado:	1.186.470,30

Estatísticas (Resíduos)	
RP grau 3 (Saneado)	
Desvio médio Abs	64.202,57
Desvio médio Rel	11,0%
RMSE	76.362,30
Valor Estimado:	1.052.612,58

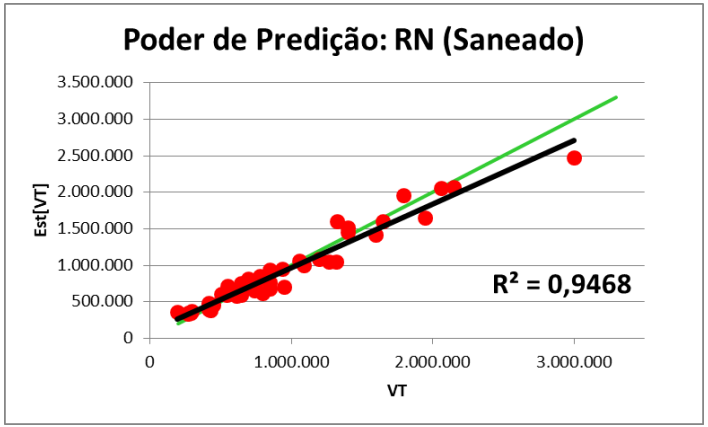
Regressão Polinomial – Grau 3

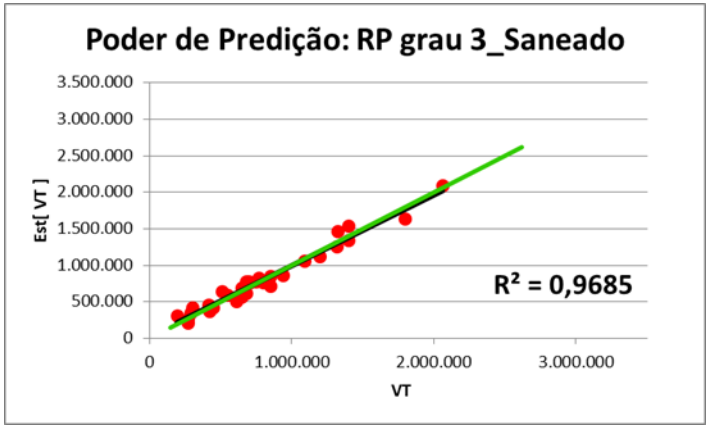
Fator de Inflação da Variância (FIV)
Variance Inflation Factor (VIF)
Valor limite: 10

Variável:	CP1	CP2	CP1^2	CP2^2	CP1*CP2	CP1^3	CP2^3	CP1*CP2^2	CP2*CP1^2
RP grau 3	5,07	7,01	5,66	17,92	3,69	11,80	28,76	5,87	5,68

Variável:	CP1	CP2	CP1*CP2	CP2^3	CP1*CP2^2
RP grau 3 (Saneado)	1,59	5,70	1,74	6,71	2,01

Melhor Rede Neural X Melhor Regressão Polinomial





Estatísticas (Resíduos) Rede Neural (Saneado)	
Desvio médio Abs	102.400,66
Desvio médio Rel	12,8%
RMSE	141.013,49
Valor Estimado:	965.886,27

Estatísticas (Resíduos) RP grau 3 (Saneado)	
Desvio médio Abs	64.202,57
Desvio médio Rel	11,0%
RMSE	76.362,30
Valor Estimado:	1.052.612,58

Rede Neural X Regressão Polinomial

Zilli, Droubi e Hochheim (2018), usando os mesmos dados realizaram os seguintes procedimentos:

1. **Para uma determinada porcentagem porc, inicialmente de 20% do número de dados disponíveis n, foram criadas duas partições, uma partição de treinamento, com porc × n = 10 dados e uma partição de testes,**
   com (1 – porc) × n = 40 dados, em que os dados das partições foram escolhidos randomicamente

2. **Com as partições de treinamento assim obtidas, foram ajustados 100 modelos de regressão polinomial de grau 2, ou de redes neurais artificiais com 1 camada oculta**

3. **Com os modelos obtidos no passo anterior, foram realizadas estimativas sobre os dados da partição de testes com respectivo cálculo da RMSE**

Rede Neural X Regressão Polinomial

Zilli, Droubi e Hochheim (2018) - usando os mesmos dados realizaram os seguintes procedimentos:

4. **Iterativamente aumentou-se, de 2 em 2 pontos percentuais o tamanho da partição de treinamento, calculando-se o valor da RMSE para a partição de teste**

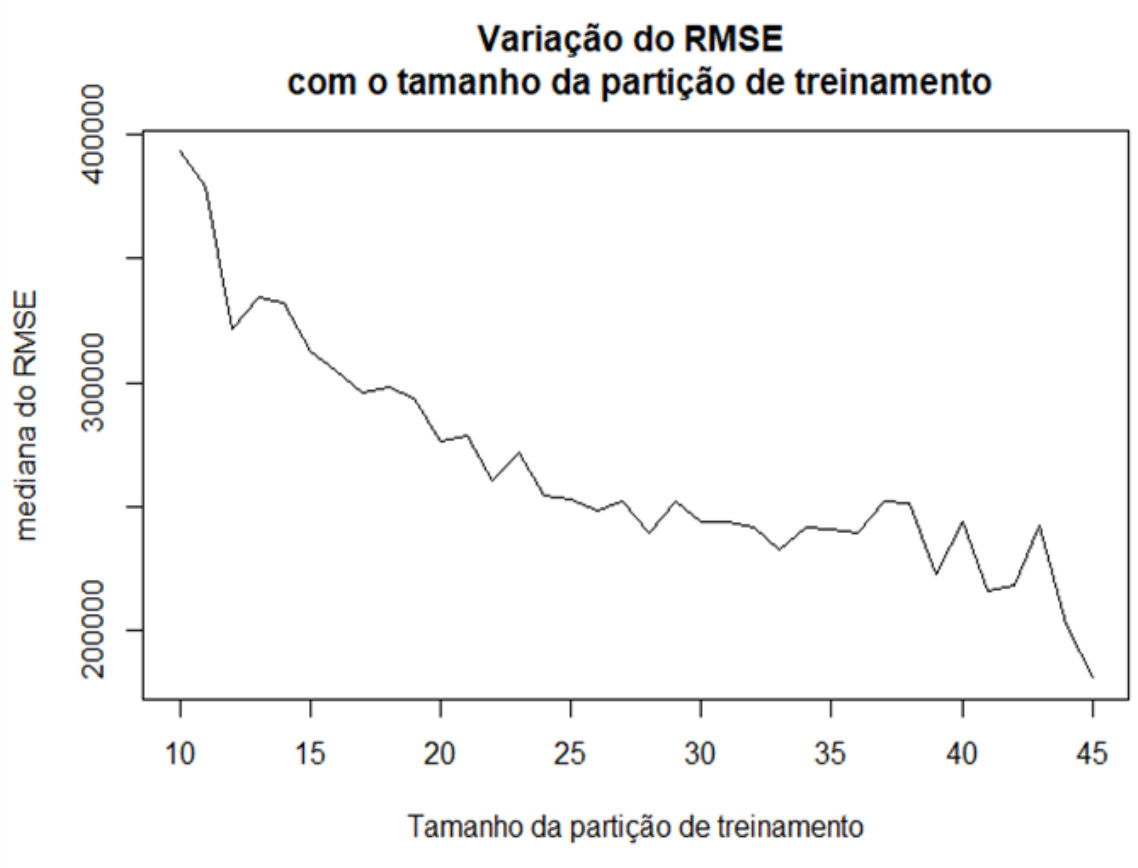
5. **Finalmente, comparou-se o comportamento dos valores medianos do RMSE com o aumento do número de dados da partição de testes para os dois métodos, assim como a distribuição final do RMSE com a utilização de 90% dos dados na partição de treinamento**

***Software* R:**

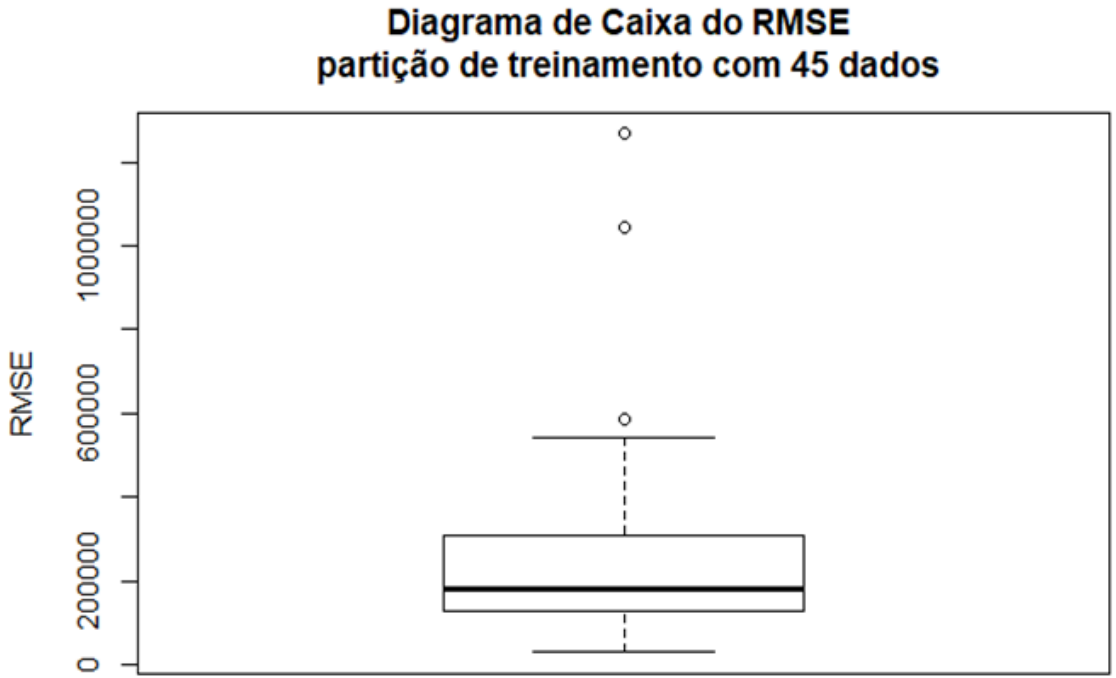
RN → Neuralnet **RP** → Polyreg

Redução Dimensional → MACP

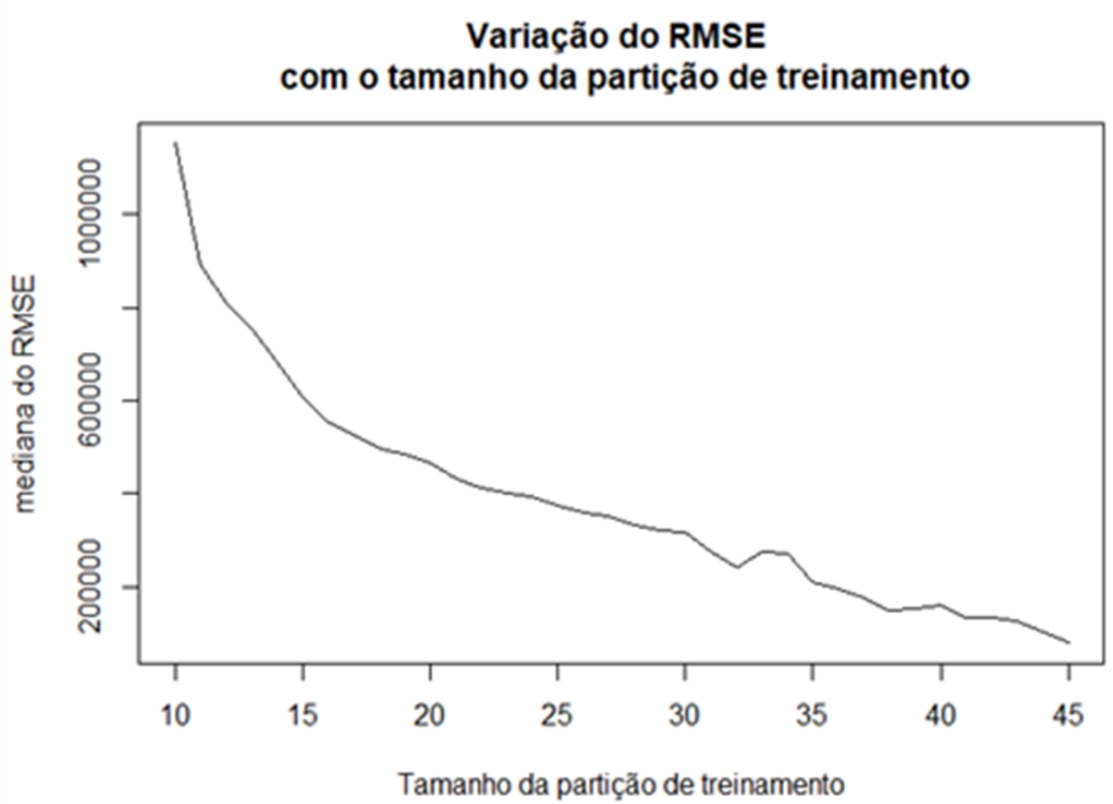
Rede Neural: Resultados



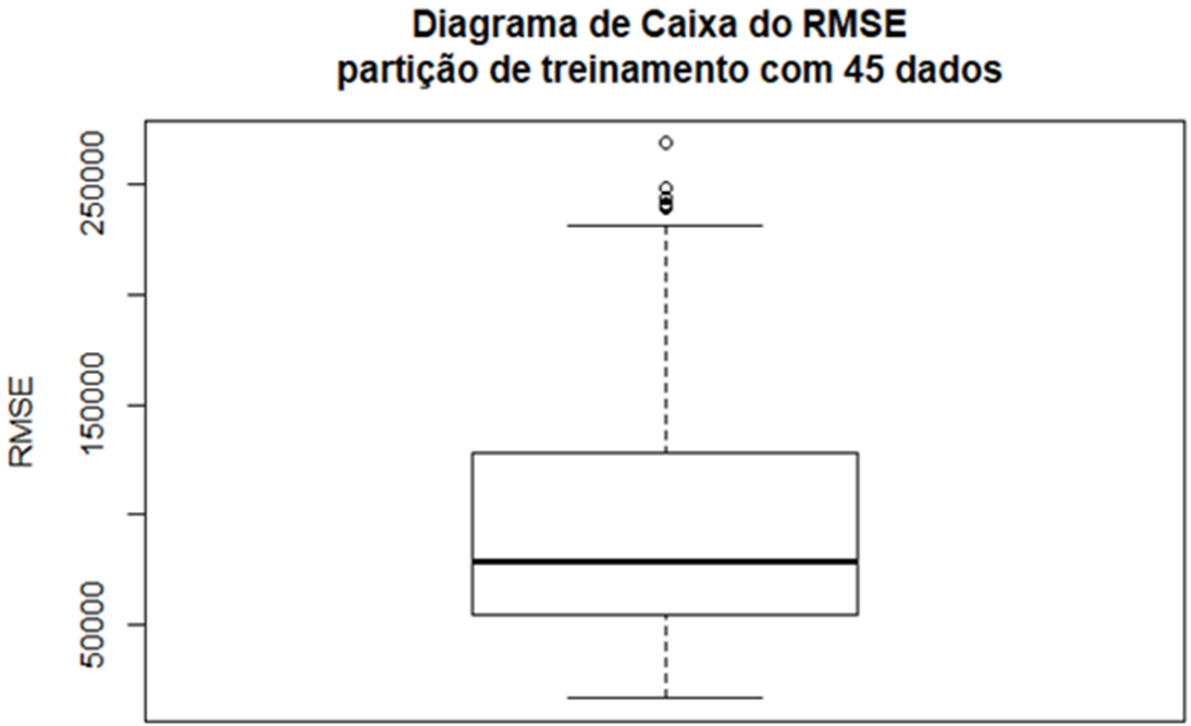
Rede Neural: Resultados



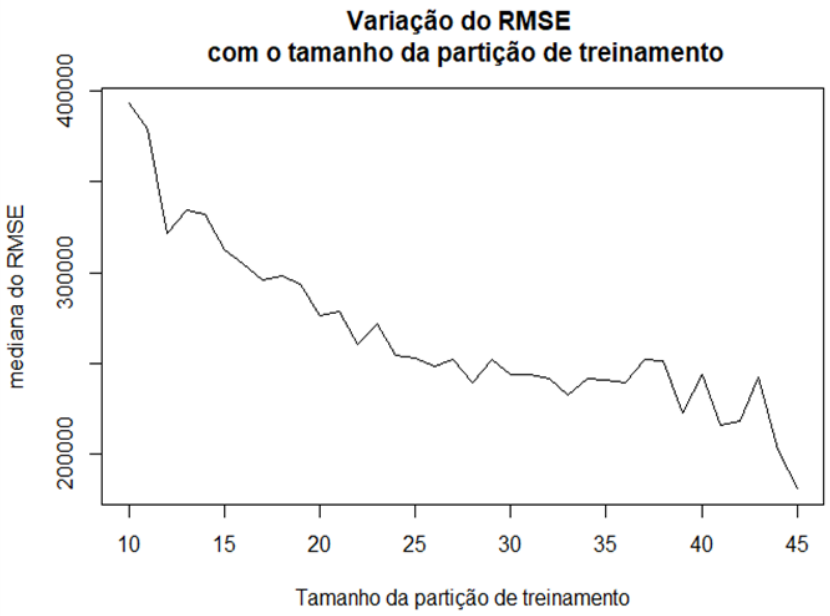
Regressão Polinomial: Resultados

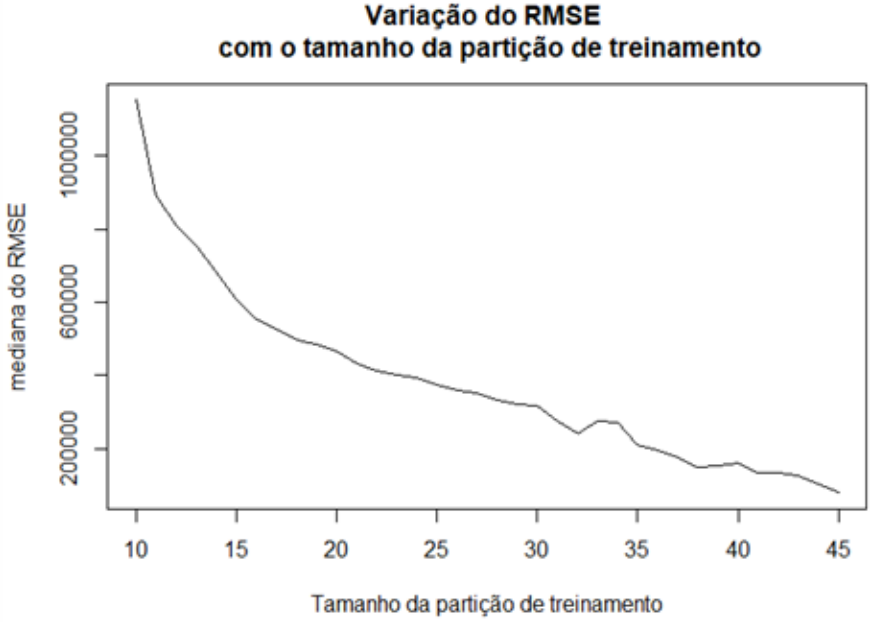


Regressão Polinomial: Resultados

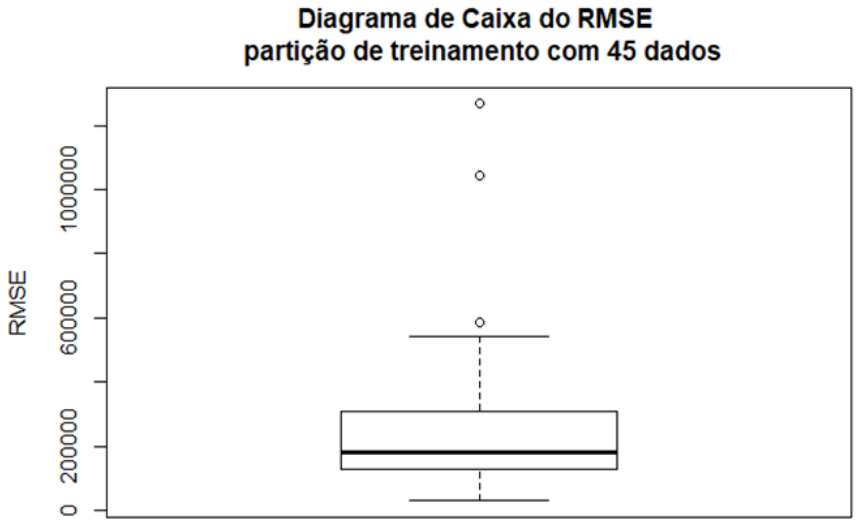


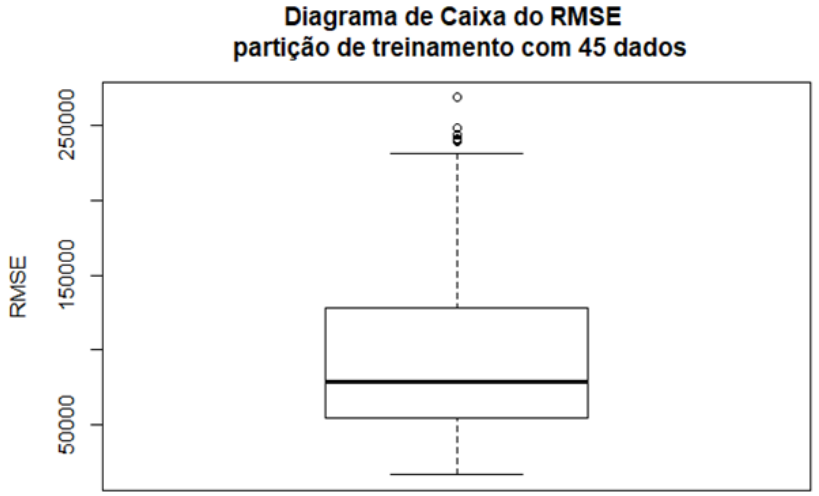
Rede Neural X Regressão Polinomial





Rede Neural X Regressão Polinomial



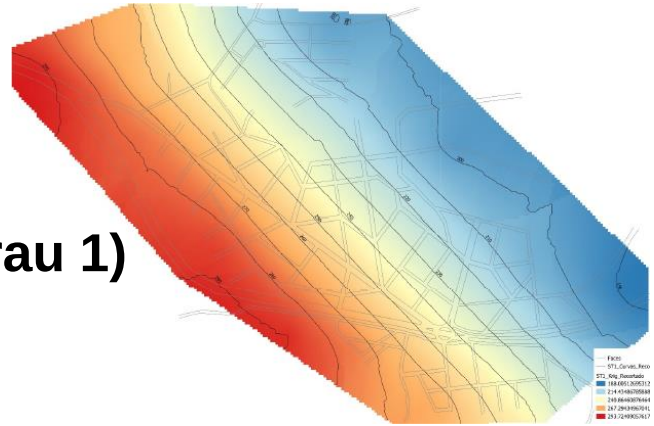


Considerações Finais: Uma ‘parceria’ de sucesso:

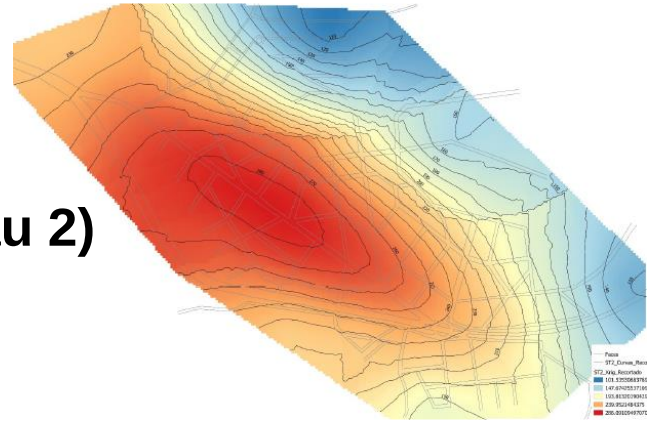
RLM + RP(coord X,coord Y) = ST (Superfície de Tendência)

Fonte: GLANERT (2019)

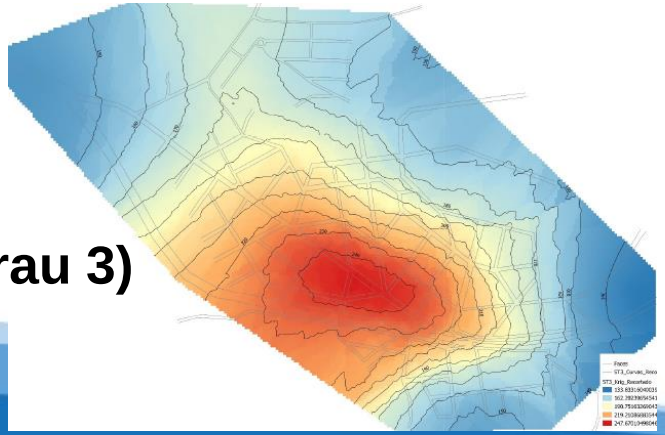
ST (grau 1)



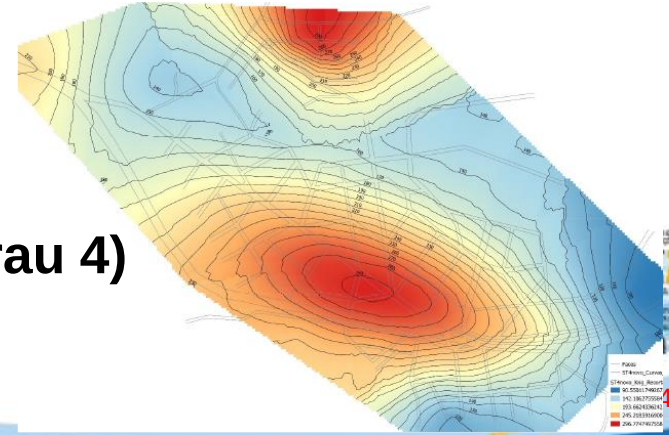
ST (grau 2)



ST (grau 3)

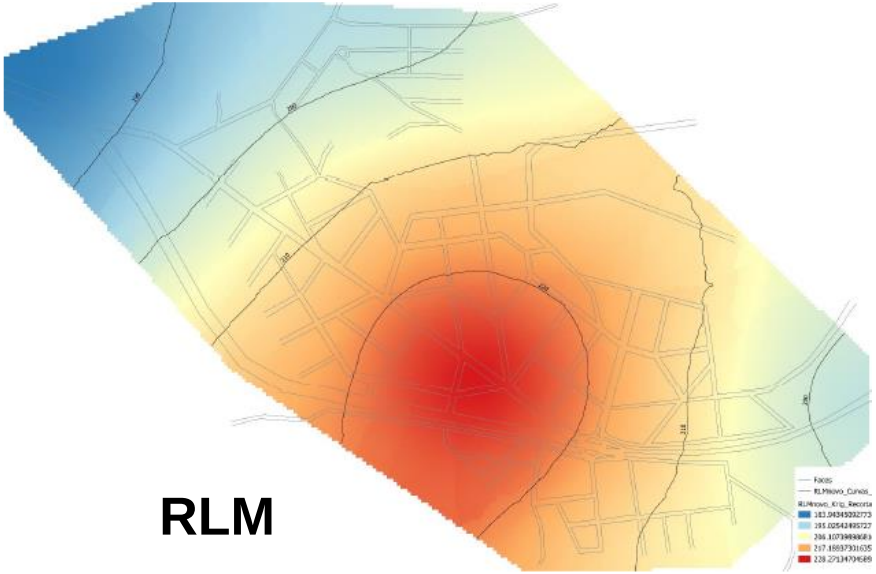


ST (grau 4)

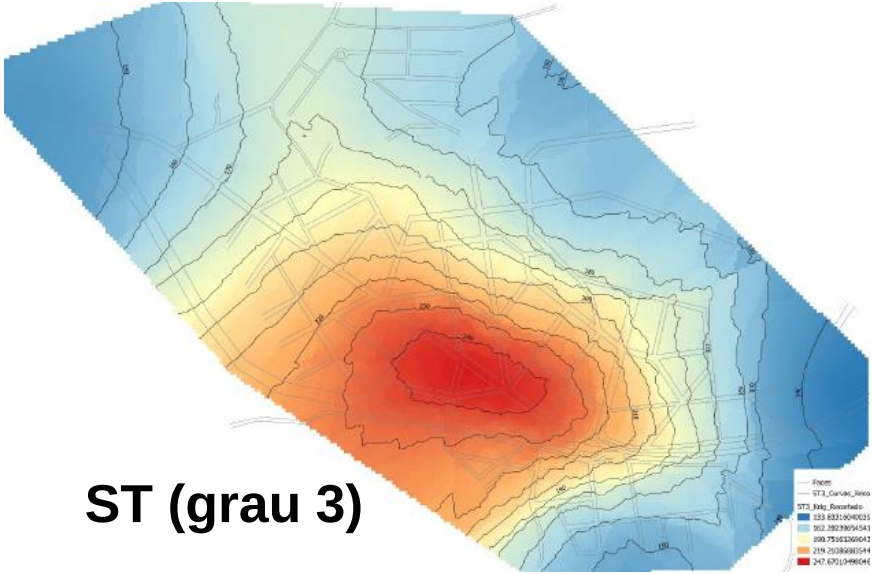


Considerações Finais: Uma ‘parceria’ de sucesso:

RLM + RP(coord X,coord Y) = ST (Superfície de Tendência)



RLM



ST (grau 3)

Fonte: GLANERT (2019)

Considerações Finais

- ✓ **Resultados da raiz do erro médio quadrático (RMSE) para modelos de regressão polinomial foram menores e com menor dispersão quando comparados com os obtidos por redes neurais artificiais**

- ✓ **Realizar mais estudos relacionados ao uso da regressão polinomial na Engenharia de Avaliações**

  - → **Regressão polinomial é um caso particular de regressão linear**
  - → **Método e aplicação são conhecidos e estão normatizados pela NBR 14.653-02**

Referências

FIORIN, D. V.; MARTINS, F. R.; SCHUCH, N. J.; PEREIRA, E. B. **Aplicações de redes neurais e previsões de disponibilidade de recursos energéticos solares.** Revista Brasileira Ensino de Física. São Paulo, v. 33, n 11, mar. 2011.

GLANERT, A. **Diagnóstico e incorporação de efeitos espaciais aplicados à planta de valores genéricos no município de Nova Erechim-SC**. Monografia de especialização. Unochapecó, 2019.

GUARNIERI, Ricardo André. **Emprego de redes neurais artificiais e regressão linear múltipla no refinamento das previsões de radiação solar do modelo ETA**. Dissertação (Mestrado em Meteorologia) - Programa de Pós-Graduação em Meteorologia, IMPE, São José dos Campos, 2006.

HAYKIN, Simon. **Redes neurais:** princípios e prática. 2. ed. Porto Alegre: Bookman, 2001.

MATLOFF, Norman; CHENG, Xi; KHOMTCHOUK, Bohdan; MOHANTY, Pete. **Polynomial regression as an alternative to neural nets.** 2018. Disponível em: < https://arxiv.org/abs/1806.06850>. Acesso em outubro 2018.

SAHA, Angshuman**. Introduction to Artificial Neural Network Models.** Disponível em: **<**www.geocities.com/adotsaha/NNinExcel.html>. Acesso em agosto 2003.

ZILLI, C. A.; DROUBI, L. F. P.; HOCHHEIM, N. **Regressão polinomial e redes neurais artificiais: um estudo de caso.** Anais VIII Simpósio da Sociedade Brasileira de Engenharia de Avaliações (SOBREA), João Pessoa (PB), 2018.

Agradecimentos

UNIVERSIDADE FEDERAL DE SANTA CATARINA (UFSC)

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TRANSPORTES E GESTÃO TERRITORIAL (PPGTG)

IBAPE-SC
Instituto Catarinense de Engenharia de Avaliações e Perícias

OBRIGADO !

hochheim@gmail.com norberto.nochheim@ufsc.br

UFSC - Universidade Federal de Santa Catarina

IBAPE-SC