Álgebra Linear - Aula Prática 4

Iara Cristina Mescua Castro

1 de outubro de 2022

1. MÉTODO DOS MÍNIMOS QUADRADOS

1) (Texto do livro Cálculo - Volume 2 – James Stewart) Em 1928, Charles Cobb e Paul Douglas publicaram um estudo no qual modelaram o crescimento da economia norte-americana durante o período de 1899- 1922. Eles consideraram uma visão simplificada da economia em que a saída da produção é determinada pela quantidade de trabalho envolvido e pela quantidade de capital investido. Apesar de existirem muitos outros fatores afetando o desempenho da economia, o modelo mostrou-se bastante preciso. A função utilizada para modelar a produção era da forma:

$$P = bL^{\alpha}K^{1-\alpha}$$

onde P é a produção total (valor monetário dos bens produzidos no ano); L é a quantidade de trabalho (número total de pessoas-hora trabalhadas no ano); e K é a quantidade de capital investido (valor monetário das máquinas, equipamentos e prédios); b e α são parâmetros (constantes) a serem determinados. Cobb e Douglas usaram os dados da tabela a seguir e o Método dos Mínimos Quadrados para obter os valores de b e de α.

a) Faça como Cobb e Douglas: use o Método dos Mínimos Quadrados para estimar os valores dos parâmetros b e α. Mostre a sua modelagem para o problema ser resolvido pelo Método dos Mínimos Quadrados.

DESENVOLVIMENTO

Queremos encontrar b e α a partir da fórmula, então para isso vamos tentar isolá-los:

$$P = bL^{\alpha}K^{1-\alpha}$$

Calculando logaritmo da equação:

$$ln(P) = ln(b) + \alpha ln(L) + (1-\alpha)ln(K)$$

$$ln(P) = ln(b) + \alpha ln(L) + ln(K) - \alpha ln(K)$$

$$ln(P) - ln(K) = ln(b) + \alpha ln(L) - \alpha ln(K)$$

$$ln(b) + \alpha(ln(L) - ln(K)) = ln(P) - ln(K)$$

Para utilizar o método dos mínimos quadrados, precisamos de expressão na forma $Ax = b$. Podemos observar que $ln(b)$ não está multiplicando com ninguém, então podemos colocar uma coluna de 1 na frente de $A = (ln(L) - ln(K))$ para representá-la da seguinte forma:

$$\begin{bmatrix} 1 & (ln(L) - ln(K))^1 \\ & \vdots \\ 1 & (ln(L) - ln(K))^{24} \end{bmatrix} \begin{bmatrix} ln(b) \\ \alpha \end{bmatrix} = \begin{bmatrix} (ln(P) - ln(K))^1 \\ \vdots \\ (ln(P) - ln(K))^{24} \end{bmatrix}$$

Agora podemos aplicar o método dos mínimos quadrados, onde:

$$A = \begin{bmatrix} 1 & (ln(L) - ln(K))^1 \\ 1 & \vdots \\ 1 & (ln(L) - ln(K))^{24} \end{bmatrix}$$

$$x = \begin{bmatrix} ln(b) \\ \alpha \end{bmatrix}$$

$$b = \begin{bmatrix} (ln(P) - ln(K))^1 \\ \vdots \\ (ln(P) - ln(K))^{24} \end{bmatrix}$$

Logo:

$$ln(b) + \alpha(ln(L) - ln(K)) = ln(P) - ln(K)$$

$$Ax = b$$

Criando uma função que receba a planilha de dados e separe as colunas K, L e P para transformá-las nas matrizes A e b do sistema, e por fim, calcular a solução x que melhor aproxima esses dados através do método dos mínimos quadrados:

$$A^T Ax = A^T b$$

Lembrando que ao encontrar x, teremos os valores de $ln(b)$ e α,

$$x = \begin{bmatrix} ln(b) \\ \alpha \end{bmatrix}$$

Então para retornar b, basta elevar o primeiro valor do vetor x, $ln(b)$ a exponencial:

Código:

```
function [alfa, b]=calcular_alfa_b(A)

P = A(:,2);
L = A(:,3);
K = A(:,4);

P_ln = log(P);
L_ln = log(L);
K_ln = log(K);

n = size(L,1);
A = [ones(n,1) (L_ln - K_ln)];
b = (P_ln - K_ln);
//nao esquecer de executar a funcao Gaussian_Elimination_4
x = Gaussian_Elimination_4(A' * A, A' * b);
alfa = x(2);
b = exp(x(1));
endfunction

```

Ao testar a função na tabela:

Ano	*P*	*L*	*K*
1899	100	100	100
1900	101	105	107
1901	112	110	114
1902	122	117	122
1903	124	122	131
1904	122	121	138
1905	143	125	149
1906	152	134	163
1907	151	140	176
1908	126	123	185
1909	155	143	198
1910	159	147	208
1911	153	148	216
1912	177	155	226
1913	184	156	236
1914	169	152	244
1915	189	156	266
1916	225	183	298
1917	227	198	335
1918	223	201	366
1919	218	196	387
1920	231	194	407
1921	179	146	417
1922	240	161	431

```
--> A = csvRead('dados.csv');

--> [alfa, b]=calcular_alfa_b(A)
 alfa  =

   0.7446062
 b  =

   1.0070689
```

Encontramos que:
· $\alpha = 0.7446062$
· $b = 1.0070689$

> b) Agora, use a função de Cobb-Douglas encontrada no item a) e teste a sua adequação calculando os valores da produção nos anos de 1910 e 1920. Comente!

Agora que temos os valores de α e b, podemos substituí-los na fórmula de produção para encontrar a previsão de P. Para isso criei uma função que recebe a tabela de dados, e verifica o ano de entrada para escolher os valores de K e L em suas respectivas colunas, e por último, substituir na fórmula.

```
function [P]=calcular_p(A, ano)

L = A(:,3); //L na coluna 3
K = A(:,4); //K na coluna 4

n = size(K,1)
for i = 1:n
  if (A(i,1) == ano) //verifica a linha do ano selecionado
    x = A(i,:);
```

```
    L = x(3);
    K = x(4);
    //funcao utilizada para modelar a producao
    P = b * L^alfa * K^(1-alfa);
  end
end
endfunction
```

Os valores de *alfa* e b não precisam estar na entrada, pois as variáveis no scilab são globais, basta haver executado $[alfa, b] = calcular\_alfa\_b(A)$ anteriormente. Da seguinte forma, podemos encontrar os valores de P nos anos de 1910 e 1920, e em qualquer outro ano desejado.

```
--> [alfa, b]=calcular_alfa_b(A)
 alfa  =

   0.7446062
 b  =

   1.0070689

--> [P]=calcular_p(A, 1910)
 P  =

   161.76185

--> [P]=calcular_p(A, 1920)
 P  =

   236.07215
```

Os valores previstos para P foram:
· $P_{1910} = 161.76185$
· $P_{1920} = 236.07215$

Os valores reais de P nesses anos foram:
· $P_{1910} = 159$
· $P_{1920} = 231$

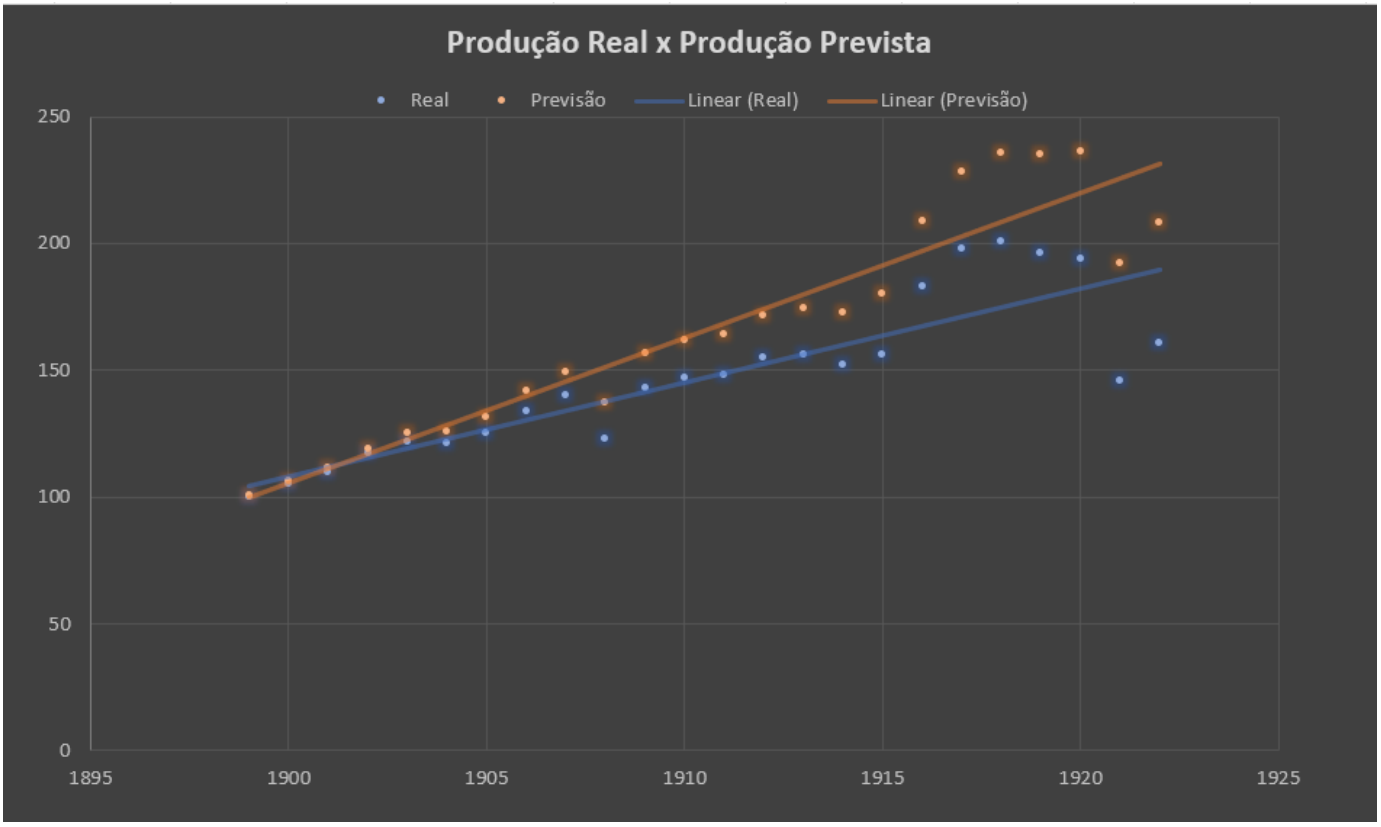
Podemos concluir que os valores de α e b calculam a produção de forma relativamente precisa. Vamos calcular mais alguns anos:

```
--> [P]=calcular_p(A, 1900)
 P  =

   106.25302

--> [P]=calcular_p(A, 1905)
 P  =

   131.65873

--> [P]=calcular_p(A, 1915)
 P  =

   180.04169

--> [P]=calcular_p(A, 1918)
 P  =

   235.90134
```

Os valores previstos para P foram:
· $P_{1900} = 106.25302$
· $P_{1905} = 131.65873$
· $P_{1915} = 180.04169$
· $P_{1918} = 235.90134$

Os valores reais de P nesses anos foram:
· $P_{1900} = 101$
· $P_{1905} = 143$
· $P_{1915} = 189$
· $P_{1918} = 223$

De forma geral, a previsão está proxima, há um erro de mais ou menos 10. Segue o gráfico feito em excel dos valores reais de produção juntamente com os valores obtidos da função de previsão ao longo dos anos:



Ao comparar a linha de tendência dos valores previstos de produção com a dos valores reais, nota-se que com o passar dos anos a precisão vai diminuindo.

2. "MACHINE LEARNING"

2) Agora vamos usar o método dos mínimos quadrados para implementar um método rudimentar de "machine learning" para diagnosticar câncer de mama a partir de um conjunto de características fornecidas para cada paciente. São dados dois arquivos: um arquivo para "treinamento" (cancer_train.csv) do modelo e um arquivo para "teste" (cancer_test.csv). O primeiro arquivo contém 300 registros e o segundo 260 registros, partes do "Wiscosin Diagnostic Breast Cancer dataset". Cada registro de cada arquivo contém 11 valores: os 10 primeiros correspondem a valores reais de 10 características dos núcleos celulares observados em imagens digitalizadas de uma fina camada de massa mamária coletada de cada paciente. O décimo primeiro valor é +1 se a paciente tem câncer de mama e -1, caso contrário. Sendo x o vetor das 10 características de cada paciente (variáveis independentes) e y o valor (+1 ou -1) que indica o diagnóstico (variável dependente), a ideia é, usando o arquivo de treinamento, obter o hiperplano y = h(x) que "melhor se ajuste aos dados fornecidos" usando o método dos mínimos quadrados. Uma vez obtido o hiperplano, o mesmo será usado para classificar cada paciente da seguinte forma: se $h(x) \geq 0$, então o diagnóstico é +1 (tem câncer), caso contrário, o diagnóstico é -1 (não tem câncer). Use o seu classificador (hiperplano) e calcule a porcentagem de acertos sobre o arquivo de treinamento (de certa forma é uma medida do ajuste do seu modelo aos dados de treinamento) e sobre o arquivo de teste (de certa forma é uma medida da capacidade de generalização do seu modelo). Construa uma Matriz de Confusão (Confusion Matrix) (pesquise a respeito) com o conjunto de teste e calcule as diversas medidas daí decorrentes, tais como: acurácia, precisão, recall, probabilidade de falso alarme, probabilidade de falsa omissão de alarme. Interprete essas medidas e comente os resultados obtidos.

Para essa questão vamos precisar classificar os pacientes da tabela de testes através de previsões seguindo o classificador h(x). Ele provém do $alfa\_tr$ que será calculado pela tabela de treinamento.

Para calcular $alfa\_tr$, criei uma função $[alfa\_barra] = Encontra\_alfa\_barra(x\_tr, y\_tr)$ que encontra uma solução aproximada resolvendo o sistema linear Ax = b pelo método dos mínimos quadrados, onde:

$$A = x_t r' * x_t r$$

$$b = x_t r' * y_t r$$

Então para isso, precisamos encontrar x_tr e y_tr primeiro.
Calculando todos os dados necessários, juntamente com o alfa: $y\_tr, y\_tt, x\_tr, x\_tt, alfa\_tr, h\_tr$:

Código:

```
function [y_tr, y_tt, x_tr, x_tt, alfa_tr, h_tr]=dados(A_tr, A_tt)
n = size(A_tr,2);

y_tr = A_tr(:,n);
y_tt = A_tt(:,n);

x_tr = [ones(y_tr) A_tr(:,1:n - 1)];
x_tt = [ones(y_tt) A_tt(:,1:n - 1)];

alfa_tr =Encontra_alfa_barra(x_tr, y_tr);
h_tr = x_tr * alfa_tr;
endfunction

function [alfa_barra]=Encontra_alfa_barra(x_tr, y_tr)
alfa_barra = Gaussian_Elimination_4(x_tr' * x_tr, x_tr' * y_tr);
endfunction
```

Agora já obtemos $alfa\_tr$ e podemos encontrar as previsões para a tabela de testes (e treinamento, para comparação), que já teve seus x_tt e y_tt obtidos na função anterior $[y\_tr, y\_tt, x\_tr, x\_tt, alfa\_tr, h\_tr] = dados(A\_tr, A\_tt)$:

Agora com outra função de encontrar o número de acertos, multiplicamos x_tr e x_tt pelo $alfa\_tr$ obtido. Para conferir essa previsão, basta multiplicar ela pelos resultados reais. Lembrando que se a previsão está certa, teremos pacientes doentes (+1) sendo classificados com um valor **positivo**, e pacientes não-doentes (-1) sendo classificados com um valor **negativo**.

Note que em ambos casos em que a previsão funciona os sinais são iguais, logo, o seu produto sempre será positivo. Ao multiplicar esses dois valores: $prev\_tr$ e $prev\_tt$ por y_tr e y_tt, respectivamente, podemos conferir se a previsão está correta, checando quantos deles são positivos.

A função seguinte calcula a previsão e o número de acertos da tabela de treinamento e da tabela de testes:

```
[prev_tr,prev_tt,acertos_tr, acertos_tt]=acertos(x_tr, x_tt, alfa_tr)
prev_tr = x_tr*alfa_tr;
prev_tt = x_tt*alfa_tr;

conf_prev_tr = prev_tr .*y_tr;
conf_prev_tt = prev_tt .*y_tt;

acertos_tr = sum(conf_prev_tr > 0 | conf_prev_tr == 0);
acertos_tt = sum(conf_prev_tt > 0 | conf_prev_tt == 0);
endfunction
```

Testando ambas funções, primeiro encontramos os dados necessários e depois calculamos os acertos com a função acima:

```
--> A_tr = csvRead('cancer_train.csv');

--> A_tt = csvRead('cancer_test.csv');

--> [y_tr, y_tt, x_tr, x_tt, alfa_tr, h_tr]=dados(A_tr, A_tt);

--> [prev_tr,prev_tt,acertos_tr, acertos_tt]=acertos(x_tr, x_tt, alfa_tr);

--> acertos_tr
 acertos_tr  =

   279.

--> acertos_tt
 acertos_tt  =

   185.
```

Concluímos que com $alfa\_tr$ teve 289 acertos de 300 casos na tabela de treinamento, enquanto na tabela de testes teve apenas 185 acertos de 260 casos.

Note que y_tt, y_tr, $prev\_tt$ e $prev\_tt$ não foram utilizados ainda, mas serão importantes para encontrar a matriz de confusão.

3. MATRIZ DE CONFUSÃO

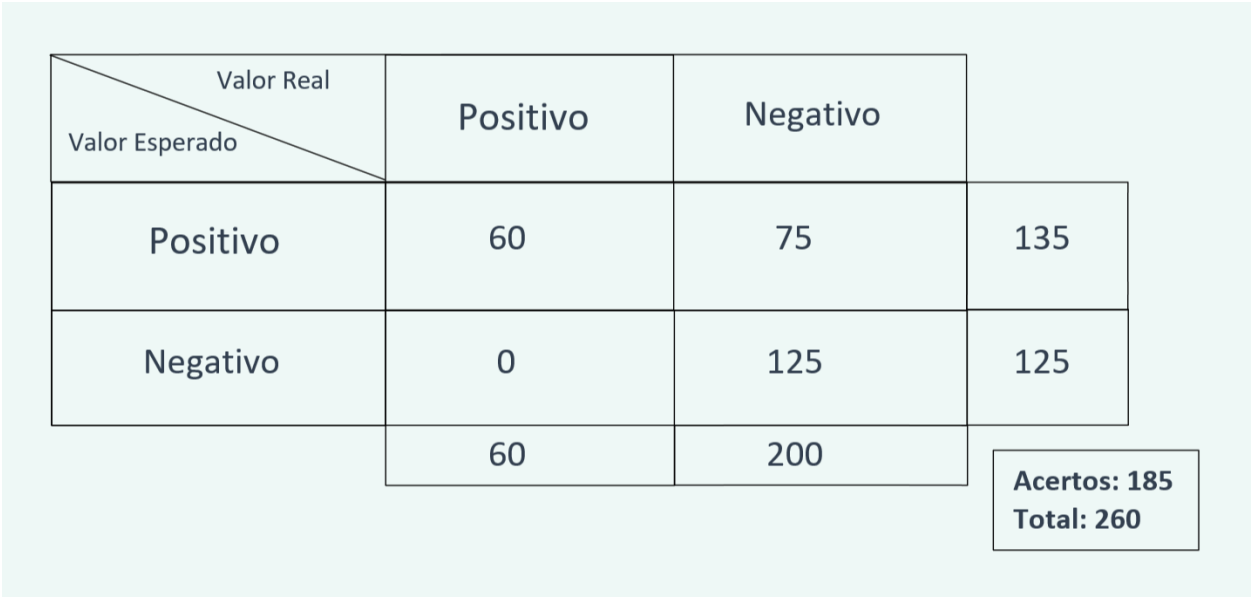
Para encontrar a matriz de confusão, optei também por criar uma função que gerasse cada valor a partir das previsões do teste *'prev_tt'* e os resultados reais em *'y_tt'*

Sabemos que cada caso na previsão mantém a ordem dos valores obtidos da planilha de dados, então basta usar cada índice para comparar se o seu valor em *'prev_tt'* é maior ou menor que 0, e checar se naquele índice de *'y_tt'* é igual a +1 ou -1.

· VP: $h(x) > 0$ e y(i) = 1
· VN: $h(x) < 0$ e y(i) = -1
· FN: $h(x) > 0$ e y(i) = -1
· FP: $h(x) < 0$ e y(i) = 1

```
function [VP, VN, FN, FP]=check(prev_tt, y_tt)

[VP]=check_pos(prev_tt, y_tt); //verdadeiro positivo: 1
[VN]=check_neg(prev_tt, y_tt); //verdadeiro negativo: 2
[FN]=check_neg_pos(prev_tt, y_tt); //falso negativo: 3
[FP]=check_pos_neg(prev_tt, y_tt); //falso positivo: 4

endfunction

function [VP]=check_pos(prev_tt, y_tt) //1: diz que tem, e tem mesmo
n = size(prev_tt, 1);
VP = 0;
for i = 1:n
  if (prev_tt(i) > 0 & y_tt(i) == 1)
    VP = VP + 1;
  end
end
endfunction

function [VN]=check_neg(prev_tt, y_tt) //2: diz que nao tem, e nao tem
n = size(prev_tt, 1);
VN = 0;
for i = 1:n
  if (prev_tt(i) < 0 & y_tt(i) == -1)
    VN = VN + 1;
  end
end
endfunction

function [FN]=check_neg_pos(prev_tt, y_tt) //3: diz que tem, mas nao tem
n = size(prev_tt, 1);
FN = 0;
for i = 1:n
  if (prev_tt(i) > 0 & y_tt(i) == -1)
    FN = FN + 1;
  end
end
endfunction

function [FP]=check_pos_neg(prev_tt, y_tt) //4: diz que nao tem, mas tem
n = size(prev_tt, 1);
FP = 0;
for i = 1:n
  if (prev_tt(i) < 0 & y_tt(i) == 1)
    FP = FP + 1;
  end
end
endfunction
```

```
--> [prev_tr,prev_tt,acertos_tr, acertos_tt]=acertos(x_tr, x_tt, alfa_tr);

--> [VP, VN, FN, FP]=check(prev_tt, y_tt)
 VP  =

   60.
 VN  =

   125.
 FN  =

   0.
 FP  =

   75.
```



Com essa matriz, podemos calcular outras medidas e fazer algumas análises:

· **Acurácia:** O cálculo da acurácia se dá pela fórmula: $\frac{VP+VN}{total}$, ou seja, o total de acertos (soma da diagonal da matriz) dividido pelo total:

$$\frac{185}{260} = 0.7115385 = 71.1\%$$

· **Precisão:** O cálculo da precisão se dá pela fórmula: $\frac{VP}{VP+FP}$, ou seja, o total de acertos/alarmes de pessoas doentes pelo total de pessoas doentes esperadas pela previsão:

$$\frac{60}{60+75} = 0.4444444 = 44.4\%$$

Houve 44%, ou seja, de todas as pessoas esperadas doentes, um pouco mais da metade delas, na verdade não tinha câncer. A precisão na matriz de testes foi baixa.

· **Recall ou Revocação:** O cálculo da recall se dá pela fórmula: $\frac{VP}{VP+FN}$, ou seja, o total de acertos/alarmes de pessoas doentes pelo total de pessoas que realmente estão doentes:

$$\frac{60}{60+0} = 1 = 100\%$$

Houve 100% de precisão, isso significa que a previsão não deu negativo $NENHUMA$ vez para alguém que realmente tinha câncer.

· **Falso Alarme:** O cálculo de falso alarme se dá pela fórmula: $\frac{FP}{VP+FP}$, ou seja, o total de pessoas que foram esperadas com câncer, mas na verdade não tinham, dividido pelo total de alarmes positivos.

$$\frac{75}{75+60} = 0.5555556 = 55.5\%$$

Houve 55% de probabilidade de falsos alarmes, ou seja, cerca de metade das pessoas esperadas com câncer pela previsão, na verdade não tinham câncer. Note que a probabilidade de falso alarme é complementar à precisão.

· **Falsa Omissão de Alarme:** O cálculo da falsa omissão de alarme se dá pela fórmula: $\frac{FN}{FN+VN}$, ou seja, o total de pessoas que tinham câncer mas foram esperadas que não tinham, dividido pelo total de alarmes negativos.

$$\frac{0}{0+120} = 0 = 0\%$$

Houve uma probabilidade de 0% de falsas omissões de alarme, visto que houveram 0 casos de falsos-negativos.

Bônus: Podemos criar uma função pra calcular essas medidas automaticamente:

```
function [acuracia, precisao, recall, falsoalarme, falsaomisao]=medidas(VP, VN, FN, FP)

total = (VP + VN + FN + FP);
acuracia = (VP + VN)/ total;
precisao = VP / (VP + FP);
recall = VP / (VP + FN);
falsoalarme = FP / (VP + FP);
falsaomisao = FN / (FN + VN);

endfunction

```

Testando a função para os dados de teste: (já calculado)

```
--> [acuracia, precisao, recall, falsoalarme, falsaomisao]=medidas(VP, VN, FN, FP)
 acuracia  =

   0.7115385
 precisao  =

   0.4444444
 recall  =

   1.
 falsoalarme  =

   0.5555556
 falsaomisao  =

   0.
```

Analogamente, podemos calcular os valores da matriz de confusão da tabela de treinamento:

```
--> [prev_tr,prev_tt,acertos_tr, acertos_tt]=acertos(x_tr, x_tt, alfa_tr);

--> [VP, VN, FN, FP]=check(prev_tr, y_tr)
 VP  =

   135.
 VN  =

   144.
 FN  =

   11.
 FP  =

   10.
```

Valor Esperado \ Valor Real	Positivo	Negativo	
Positivo	135	10	145
Negativo	11	144	155
	246	154	

Acertos: 279
Total: 300

Encontrando as medidas para os dados de treinamento:

```
--> [VP, VN, FN, FP]=check(prev_tr, y_tr);

--> [acuracia, precisao, recall, falsoalarme, falsaomisao]=medidas(VP, VN, FN, FP)
 acuracia  =

   0.93
 precisao  =

   0.9310345
 recall  =

   0.9246575
 falsoalarme  =

   0.0689655
 falsaomisao  =

   0.0709677
```

· **Acurácia:** $= 0.93 = 93\%$
· **Precisão:** $= 0.9310345 = 93.1\%$
· **Recall ou Revocação:** $= 0.9246575 = 92.4\%$
· **Falso Alarme:** $= 0.0689655 = 6.8\%$
· **Falsa Omissão de Alarme:** $= 0.0709677 = 7.09\%$

Logo à primeira vista, nota-se uma diferença em relação às medidas da tabela de testes, apesar da probabilidade de falsa omissão de alarme ser maior, os resultados foram mais consistentes. Isso se dá especialmente pela boa precisão do $alfa\_tr$ nestes pacientes, onde conseguiu prever muito melhor quem realmente tinha câncer.