Fundação Getulio Vargas

Escola de Matemática Aplicada (EMAp)

Nome: Lucas Machado Moschen

Data: 14 de maio de 2019

Professor Antônio Branco

Aula Prática 4

Exercício 1: Nesse exercício, represento os anos com início em 0, isto é, 1920 representa 0, 1930, 10 e assim por diante. $A = [1,0; 1,10; 1,20; 1,30; 1,40; 1,50; 1,60; 1,70]$ e $b = [54.1; 59.7; 62.9; 68.2; 69.7; 70.8; 73.7; 75.4]$. Encontro as matrizes $A^T A$ e $A^T b$, para, então, resolver o sistema linear com o método Gaussian_Elimination $A^T Ax = A^T b$

```
Scilab 6.0.2 Console
--> A = [1,0;1,10;1,20;1,30;1,40; 1,50; 1,60; 1,70]
 A  =

   1.   0.
   1.   10.
   1.   20.
   1.   30.
   1.   40.
   1.   50.
   1.   60.
   1.   70.


--> b = [54.1;59.7;62.9;68.2;69.7;70.8;73.7;75.4]
 b  =

   54.1
   59.7
   62.9
   68.2
   69.7
   70.8
   73.7
   75.4


--> S = A'*A
 S  =

   8.     280.
   280.   14000.


--> c = A'*b
 c  =

   534.5
   19929.
```

a) A reta é $y = 56.633 \ldots + 0.290833x$ e com essa reta, a previsão da expectativa de vida é 79.9anos.

```
--> x = Gaussian_Elimination_4(S,c)
 x  =

   56.633333
   0.2908333


--> expectativa_2000 = [1,80]*x
 expectativa_2000  =

   79.9
```

b) A real expectativa de vida em 2000 é 76,64 anos segundo o Banco Mundial, então, a reta não caracteriza bem esses dados. Porém, posso analisar o vetor erro. $||e|| = ||Ax - b|| \approx 4{,}68$. Apesar do erro ser um número pequeno, ele não é preciso como forma de previsão, visto que o crescimento da expectativa não é linear.

Exercício 2: Inicialmente, tenho as matrizes $A = [1,0.5,0.25; 1,1,1; 1,1.5,2.25; 1,2,4; 1,3,9]$ e $b = [11; 17; 21; 23; 18]$. Encontro as matrizes $A^T A$ e $A^T b$, para, então, resolver o sistema linear com o método Gaussian_Elimination $A^T Ax = A^T b$.

```
Scilab 6.0.2 Console

--> A = [1,0.5,0.25;1,1,1;1,1.5,2.25;1,2,4;1,3,9]
 A  =

   1.   0.5   0.25
   1.   1.    1.
   1.   1.5   2.25
   1.   2.    4.
   1.   3.    9.


--> b = [11;17;21;23;18]
 b  =

   11.
   17.
   21.
   23.
   18.


--> S = A'*A
 S  =

   5.     8.     16.5
   8.     16.5   39.5
   16.5   39.5   103.125


--> c = A'*b
 c  =

   90.
   154.
   321.
```

a) A aproximação quadrática é $s(t) = 1.92 + 20.31t - 4.97t^2$

```
--> x = Gaussian_Elimination_4(S,c)
 x  =

   1.9175258
   20.306333
  -4.9720177
```

b) Chegamos em $s_0 \approx 1{,}92m$, $v_0 \approx 20{,}31\ m/s$ e $g \approx -9{,}94\ m/s^2$

c) O objeto atinge o chão tem o mesmo significado de $s(t) = 0$. Isto é, queremos encontrar a raiz positiva do polinômio acima mostrado. Aplicando a fórmula de Bháskara no próprio Scilab encontro um valor aproximadamente de $t = 4{,}17s$ para o objeto tocar no chão. É interessante que o vetor $(1, t, t^2)$é perpendicular ao vetor x.

Exercício 3: Aplico, inicialmente, a função logaritmo natural em ambos os lados da equação, de forma que torna-se uma resolução de sistema linear e posso utilizar o método dos mínimos quadrados com o nosso modelo de matrizes, assim $ln(p(t)) = ln(c) + kt$. Também utilizo 1950 como 0.

a) $A = [1,0; 1,10; 1,20; 1,30; 1,40; 1,50]$ e b = ln([150;179;203;227;250;281]). Basta encontrar as matrizes $A^T A$ e $A^T b$, para, então, resolver o sistema linear com o método Gaussian_Elimination $A^T Ax = A^T b$.

```
Scilab 6.0.2 Console

--> A = [1,0;1,10;1,20;1,30;1,40;1,50]
 A  =

   1.   0.
   1.   10.
   1.   20.
   1.   30.
   1.   40.
   1.   50.


--> b = log([150;179;203;227;250;281])
 b  =

   5.0106353
   5.1873858
   5.313206
   5.42495
   5.5214609
   5.6383547


--> S = A'*A
 S  =

   6.     150.
   150.   5500.


--> c = A'*b
 c  =

   32.095993
   823.66265
```

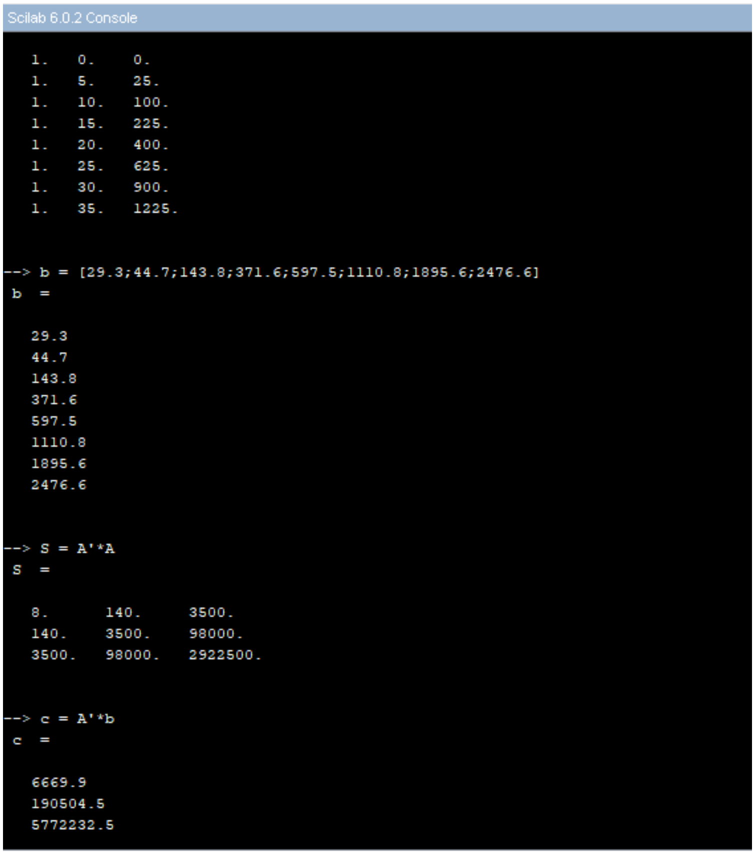
```
--> x = Gaussian_Elimination_4(S,c)
 x  =

   5.0455774
   0.0121502
```

Assim encontro que $c = e^{x(1)} \approx 155.3334$e $k \approx 0.0122$. A taxa de crescimento da população é p’(t) e $p'(t) = kce^{kt} = 1.89e^{0.0122t}$

b) Para estimar a população em 2010, basta calcularmos $p(60) = e^{x(1)}e^{x(2)\cdot 60} \approx 322.012$ milhões.

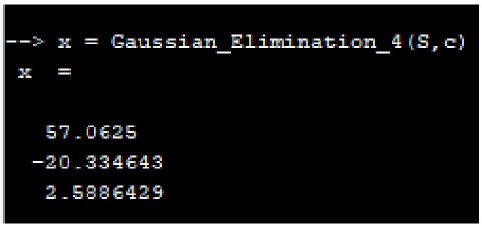
Exercício 4: Utilizo 1970 como 0.

a) Com os dados, obtenho as matrizes $A = [1,0,0; 1,5,25; 1,10,100; 1,15,225; 1,20,400; 1,25,625; 1,30,900; 1,35,1225]$ e $b = [29.3; 44.7; 143.8; 371.6; 597.5; 1110.8; 1895.6; 2476.6]$. Encontramos as matrizes $A^T A$ e $A^T b$ e resolvemos o sistema linear $A^T Ax = A^T b$.



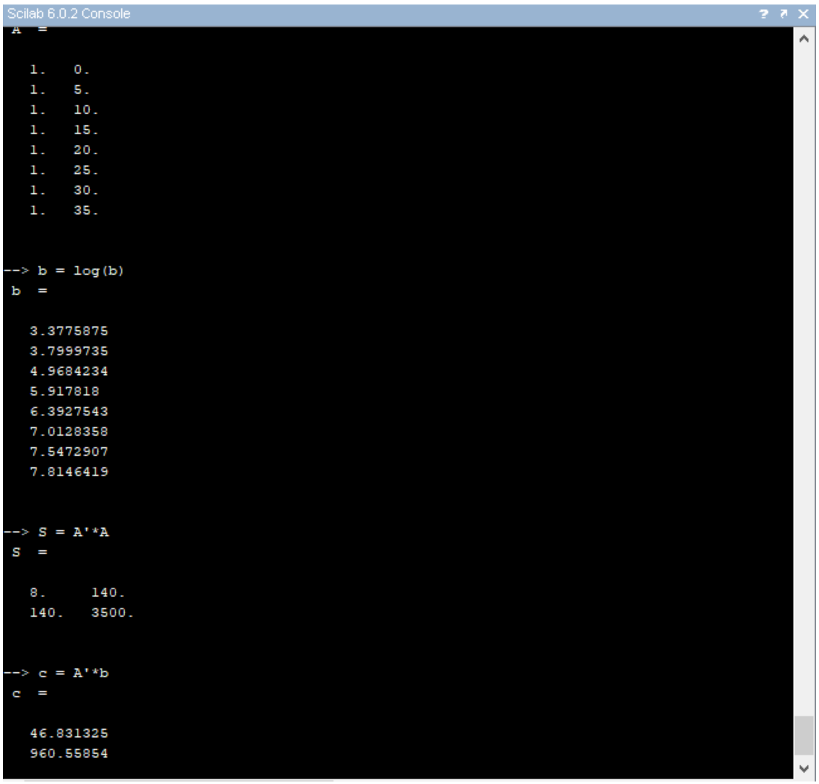
```
Scilab 6.0.2 Console

  1.   0.    0.
  1.   5.    25.
  1.   10.   100.
  1.   15.   225.
  1.   20.   400.
  1.   25.   625.
  1.   30.   900.
  1.   35.   1225.


--> b = [29.3;44.7;143.8;371.6;597.5;1110.8;1895.6;2476.6]
 b  =

  29.3
  44.7
  143.8
  371.6
  597.5
  1110.8
  1895.6
  2476.6


--> S = A'*A
 S  =

  8.      140.     3500.
  140.    3500.    98000.
  3500.   98000.   2922500.


--> c = A'*b
 c  =

  6669.9
  190504.5
  5772232.5
```

Utilizamos, mais uma vez, a função Gaussian_Elimination. Daqui, temos que a aproximação quadrática é $q(x) = -19020.833 - 10.733651x + 2.3350317x^2$



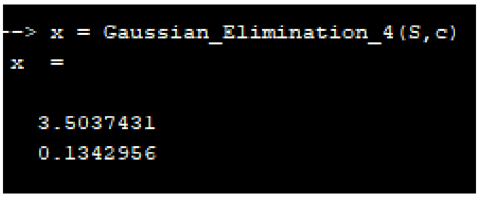
```
--> x = Gaussian_Elimination_4(S,c)
 x  =

   57.0625
  -20.334643
   2.5886429
```

b) Aqui $A = [1,0; 1,5; 1,10; 1,15; 1,20; 1,25; 1,30; 1,35]$ e $b = ln([29.3; 44.7; 143.8; 371.6; 597.5; 1110.8; 1895.6; 2476.6])$. Repito o mesmo processo de resolução do exercício anterior.



```
Scilab 6.0.2 Console
A  =

  1.   0.
  1.   5.
  1.   10.
  1.   15.
  1.   20.
  1.   25.
  1.   30.
  1.   35.


--> b = log(b)
b  =

  3.3775875
  3.7999735
  4.9684234
  5.917818
  6.3927543
  7.0128358
  7.5472907
  7.8146419


--> S = A'*A
S  =

  8.     140.
  140.   3500.


--> c = A'*b
c  =

  46.831325
  960.55854
```

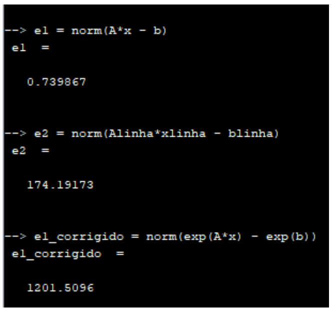
A partir disso, calcula-se o vetor x com o método Gaussian_Elimination.



```
--> x = Gaussian_Elimination_4(S,c)
x  =

  3.5037431
  0.1342956
```

Logo, a aproximação exponencial é $p(x) = e^{x(1)}e^{x(2)t} \approx 33.2396e^{0.1343t}$

c) Calculam-se os erros referentes às aproximações quadrática e exponencial. Para calcular o erro da aproximação quadrática, faço $||e|| = ||Ax - b||$, porém a norma da diferença na aproximação exponencial nos diz pouco, visto que $b = ln(p(t))$ e $Ax = ln(c) + kt$. Assim, $||e|| = ||exp(Ax) - exp(b)||$, onde estaremos usando b em sua devida escala e $exp(Ax)$ a sua real previsão com o modelo. Como a aproximação quadrática tem erro menor, considero ela uma melhor aproximação. Uso linha para aproximação quadrática.



```
--> e1 = norm(A*x - b)
e1  =

  0.739867


--> e2 = norm(Alinha*xlinha - blinha)
e2  =

  174.19173


--> e1_corrigido = norm(exp(A*x) - exp(b))
e1_corrigido  =

  1201.5096
```

d) As previsões para 2010 e 2015 são, respectivamente, utilizando a aproximação quadrática porque ela é uma melhor aproximação, $q(40) = ([1{,}40{,}1600] \cdot x)_{11} = 3385.505$e $q(45) = ([1{,}45{,}2025] \cdot x)_{11} = 4384.005$ milhares de dólares, em média.

Exercício 5: Inicialmente leio esse arquivo csv no Scilab e aplico o método conhecido para encontrar os parâmetros do hiperplano.

```
Scilab 6.0.2 Console
--> Train = csvRead("cancer_train.csv")
 Train  =


         column 1 to 9

   0.64     0.2643   0.6515   0.4006   0.8182   0.8037   0.7031   0.7311   0.7957
   0.7318   0.4524   0.705    0.5306   0.5856   0.2277   0.2036   0.3488   0.5961
   0.7005   0.541    0.6897   0.4814   0.7574   0.4629   0.4625   0.6357   0.6806
   0.4063   0.5188   0.4116   0.1545   0.9848   0.8219   0.5656   0.5229   0.8543
   0.7218   0.3651   0.7167   0.519    0.6932   0.3845   0.4639   0.5184   0.5951
   0.4429   0.3997   0.438    0.1909   0.8832   0.4922   0.3697   0.402    0.6865
   0.6492   0.5087   0.6345   0.4162   0.654    0.3156   0.2641   0.3678   0.5901
   0.4877   0.5303   0.4785   0.2313   0.8217   0.4763   0.2194   0.2975   0.7224
   0.4625   0.5555   0.4642   0.208    0.8798   0.5594   0.4356   0.4649   0.773
   0.4433   0.612    0.4455   0.1904   0.8196   0.6937   0.5326   0.4246   0.6678
   0.5699   0.5916   0.5448   0.3192   0.5671   0.1931   0.0773   0.1652   0.5026
   0.5614   0.4554   0.5496   0.3125   0.671    0.3741   0.2332   0.3283   0.6059
   0.682    0.6314   0.7024   0.4494   0.6731   0.7116   0.4838   0.5557   0.7885
   0.5639   0.6097   0.5501   0.3132   0.5806   0.2901   0.2328   0.2666   0.6076
   0.4884   0.5756   0.4966   0.2314   0.7816   0.6639   0.4986   0.3989   0.6806
   0.5173   0.7011   0.5132   0.2636   0.7871   0.4618   0.384    0.366    0.7576
   0.5222   0.5125   0.5026   0.2739   0.6819   0.2085   0.1733   0.2614   0.5217
   0.5738   0.5265   0.5735   0.3196   0.8086   0.5854   0.4035   0.5109   0.7118
   0.7047   0.5639   0.6897   0.5042   0.6794   0.2973   0.3465   0.4721   0.5204
   0.4817   0.3656   0.464    0.2266   0.6758   0.2354   0.1561   0.2376   0.6201
   0.4653   0.3999   0.4543   0.2081   0.7429   0.3677   0.107    0.1546   0.647
   0.3381   0.3167   0.3201   0.1096   0.7077   0.188    0.0693   0.1032   0.597
   0.5457   0.363    0.5438   0.2819   0.7415   0.6181   0.4866   0.4849   0.8293
   0.7528   0.5866   0.7279   0.5618   0.6516   0.2959   0.257    0.429    0.5819
   0.5923   0.5443   0.5836   0.362    0.7747   0.4218   0.3573   0.4558   0.6563
   0.6097   0.4175   0.6154   0.3652   0.8196   0.6589   0.5223   0.6963   1.
   0.5187   0.5481   0.5168   0.258    0.7284   0.5408   0.3339   0.4365   0.7408
   0.662    0.5155   0.6477   0.4378   0.6524   0.3086   0.3491   0.3842   0.5582
   0.5443   0.6433   0.5432   0.2931   0.7478   0.4913   0.3943   0.4349   0.6336
   0.625    0.3831   0.6101   0.3822   0.6805   0.335    0.2314   0.3953   0.572
   0.6628   0.6393   0.6621   0.4354   0.7353   0.5463   0.5433   0.6183   0.7181
```

```
Scilab 6.0.2 Console

--> S = A'*A
 S  =


         column 1 to 7

   300.       154.141     147.5548    150.0015    82.0813     203.1371    96.269
   154.141    83.966305   77.525382   81.963232   47.467449   104.77098   52.478042
   147.5548   77.525382   76.132807   75.562809   42.228591   99.965872   48.967562
   150.0015   81.963232   75.562809   80.046643   46.477097   102.13569   51.596091
   82.0813    47.467449   42.228591   46.477097   28.479059   56.057643   29.632482
   203.1371   104.77098   99.965872   102.13569   56.057643   140.2849    68.176907
   96.269     52.478042   48.967562   51.596091   29.632482   68.176907   38.899519
   70.        40.579645   36.521385   40.079624   24.386751   50.351289   30.851984
   81.1051    47.50576    42.127761   46.806794   28.705552   57.999954   34.02661
   182.5533   94.309236   90.049635   91.9514     50.489161   125.14329   61.475012
   193.7372   98.655056   95.15013    96.122647   52.089901   132.48735   64.342993

         column 8 to 11

   70.         81.1051     182.5533    193.7372
   40.579645   47.50576    94.309236   98.655056
   36.521385   42.127761   90.049635   95.15013
   40.079624   46.806794   91.9514     96.122647
   24.386751   28.705552   50.489161   52.089901
   50.351289   57.999954   125.14329   132.48735
   30.851984   34.02661    61.475012   64.342993
   27.571073   29.145927   45.502713   46.997318
   29.145927   33.203387   52.095687   53.300484
   45.502713   52.095687   113.83674   118.95237
   46.997318   53.300484   118.95237   126.84905
```

```
--> c = A'*b
 c  =

  -8.
   22.3732
   11.9498
   23.7317
   26.1507
   4.3607
   25.6336
   34.4296
   40.4757
   3.7953
  -5.1184
```

Encontro os 11 parâmetros, então, representados como coordenadas do vetor x. $c_0 = x(1)$ e assim por diante.

```
--> x = Gaussian_Elimination_4(S,c)
 x  =

  -6.7579731
   29.311052
   2.0765803
  -18.730222
  -7.3665161
   1.2222756
   0.2283419
   0.0503253
   2.2385058
   0.0249405
   0.7704282
```

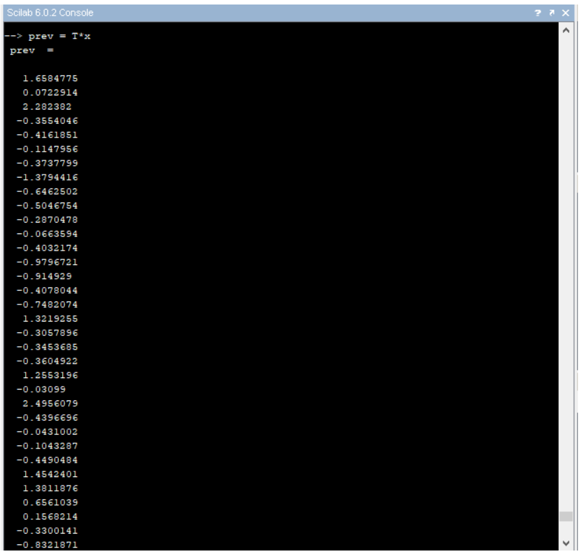
a) Vou calcular a porcentagem sobre o arquivo de treinamento, a fim de obter uma medida de capacidade de generalização do modelo. Para isso, realizo o produto $Ax.\cdot b$ que faz o produto índice a índice. Se o resultado der positivo, é porque o modelo acertou, se der negativo, porque o modelo errou.

$$comparing = (Ax).\cdot b$$

$percent = size(find(comparing >= 0),2)/size(comparing,2) =$ $0.93 = 93\%$, taxa de acerto.

b) Agora, calculemos a porcentagem de acerto sobre os dados na planilha de teste. Capturam-se os dados da planilha cancer_test e dispomos numa matriz T. Obtemos as previsões com o vetor Tx e para calcular a porcentagem, fazemos como no item anterior.

```
Scilab 6.0.2 Console

--> T = zeros(260,11);T(:,1) = ones(260,1);T(:,2:11) = Test(:,1:10)
 T  =


         column 1 to 10

   1.   0.7123   0.6152   0.6929   0.4866   0.7038   0.6374   0.6044   0.5551   0.6975
   1.   0.4544   0.6475   0.4303   0.1884   0.5172   0.3936   0.1879   0.162    0.6933
   1.   0.7327   0.7767   0.7207   0.4986   0.661    0.7135   0.6281   0.6691   0.8754
   1.   0.3826   0.6058   0.3577   0.1337   0.6536   0.2592   0.0632   0.093    0.5769
   1.   0.4179   0.5911   0.3937   0.1612   0.5418   0.2987   0.092    0.0785   0.5492
   1.   0.423    0.7972   0.3972   0.1668   0.4574   0.2208   0.0543   0.0686   0.7532
   1.   0.4814   0.515    0.4498   0.2148   0.5209   0.2038   0.004    0.017    0.6353
   1.   0.3282   0.4688   0.3016   0.0985   0.4287   0.121    0.0101   0.0181   0.696
   1.   0.4923   0.4137   0.4585   0.2264   0.4514   0.1403   0.0076   0.0231   0.5313
   1.   0.4759   0.4505   0.4425   0.2122   0.5111   0.145    0.0125   0.0462   0.5656
   1.   0.4267   0.6221   0.3977   0.1674   0.5394   0.2039   0.0435   0.06     0.7536
   1.   0.5328   0.5107   0.4959   0.2659   0.4662   0.1365   0.0398   0.0981   0.6353
   1.   0.4654   0.4352   0.4403   0.2016   0.5382   0.3085   0.1115   0.1332   0.6232
   1.   0.4209   0.349    0.3945   0.1636   0.5261   0.2317   0.0376   0.0466   0.7135
   1.   0.3135   0.6055   0.2894   0.0884   0.6573   0.227    0.       0.       0.842
   1.   0.4555   0.5485   0.4237   0.1926   0.5209   0.1488   0.0123   0.0336   0.4729
   1.   0.4442   0.4583   0.4133   0.1845   0.4733   0.1247   0.0309   0.0264   0.6512
   1.   0.6645   0.6143   0.6351   0.4106   0.5965   0.4336   0.3109   0.4156   0.7034
   1.   0.3298   0.6152   0.3214   0.0978   0.61     0.7655   0.5433   0.2566   0.907
   1.   0.4533   0.5534   0.4205   0.1908   0.4625   0.1341   0.0369   0.0888   0.573
   1.   0.3738   0.5267   0.3559   0.1296   0.6493   0.4313   0.185    0.2073   0.6785
   1.   0.7352   0.64     0.7014   0.5094   0.4908   0.3325   0.3177   0.4039   0.7505
   1.   0.469    0.4336   0.4431   0.2018   0.694    0.3429   0.1045   0.1777   0.6006
   1.   0.7418   0.7002   0.7271   0.5054   0.716    0.7279   0.7056   0.7862   1.
   1.   0.4449   0.4951   0.4174   0.1831   0.5308   0.2541   0.0549   0.0884   0.6376
   1.   0.4621   0.5632   0.4347   0.1959   0.6291   0.2975   0.0878   0.1101   0.6645
   1.   0.5146   0.4193   0.4817   0.2465   0.5697   0.206    0.0486   0.1429   0.5344
   1.   0.4387   0.5837   0.4071   0.1783   0.4702   0.1511   0.0043   0.0292   0.538
   1.   0.5934   0.6742   0.572    0.3253   0.7154   0.512    0.4066   0.4437   0.7583
   1.   0.593    0.7122   0.5752   0.3306   0.713    0.4981   0.4949   0.4172   0.7275
```

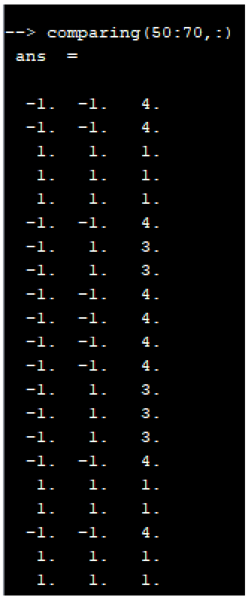


```
Scilab 6.0.2 Console
--> prev = T*x
 prev  =

   1.6584775
   0.0722914
   2.282382
  -0.3554046
  -0.4161851
  -0.1147956
  -0.3737799
  -1.3794416
  -0.6462502
  -0.5046754
  -0.2870478
  -0.0663594
  -0.4032174
  -0.9796721
  -0.914929
  -0.4078044
  -0.7482074
   1.3219255
  -0.3057896
  -0.3453685
  -0.3604922
   1.2553196
  -0.03099
   2.4956079
  -0.4396696
  -0.0431002
  -0.1043287
  -0.4490484
   1.4542401
   1.3811876
   0.6561039
   0.1568214
  -0.3300141
  -0.8321871
```

$$comparing = (Tx).\cdot b$$

$percent = size(find(comparing >= 0),2)/size(comparing,2) = 0.7115 = 71.15\%$, taxa de acerto. De fato a taxa foi menor do que eu esperava e a nível geral não é um bom preditor, porém é melhor do que um randômico.

Por fim, vejamos a porcentagem de falsos positivo e negativo e verdadeiros positivo e negativo na tabela "cancer_train", baseado no nosso hiperplano. Criei uma matriz de comparing com os dados da real presença ou ausência da doença e uma terceira coluna com sua classificação.



```
--> comparing(50:70,:)
 ans  =

  -1.  -1.   4.
  -1.  -1.   4.
   1.   1.   1.
   1.   1.   1.
   1.   1.   1.
  -1.  -1.   4.
  -1.   1.   3.
  -1.   1.   3.
  -1.  -1.   4.
  -1.  -1.   4.
  -1.  -1.   4.
  -1.  -1.   4.
  -1.   1.   3.
  -1.   1.   3.
  -1.   1.   3.
  -1.  -1.   4.
   1.   1.   1.
   1.   1.   1.
  -1.  -1.   4.
   1.   1.   1.
   1.   1.   1.
```

Defini: 1 = Verdadeiro Positivo, 2 = Falso Negativo (teste negativo e presença da doença), 3 = Falso Positivo (teste positivo e ausência da doença) e 4 = Verdadeiro Negativo.

Calculam-se, as porcentagens de cada tipo:

```
--> porcentagens
 porcentagens  =

   0.2307692   0.   0.2884615   0.4807692
```

Assim:

Taxa de verdadeiro positivo: 23.08%

Taxa de verdadeiro negativo: 48.08%

Taxa de falso positivo: 28.84%

Taxa de falso negativo: 0%