I have the hope that you might guide me with respect to a way to solve binary classification problems. Suppose that, for instance, a tumor may or may not be malignant. The question is whether data science can help us predict with fair confidence what category the objective column which consists of just zeros and ones; zero for benign and one for malignant.
We want to avoid a biopsy.

I have a way to solve the following problem: we are given 617 patients (our “training set”) and 56 measurements of those patients. We can form a matrix g such that the first column corresponds to the malignancy of a tumor and the 56 other columns all represent what that measurement is for that patient. For example, one column could be the resting blood pressure of the patient. Another column could be tumor density as determined by ultrasound.

The data I used here came from Kaggle and can be found here: https : // www . kaggle . com/competitions/icr - identify - age - related - conditions/data

Let the first column of g consist of just zero or one. The other columns are measurements related to the tumor. Here is the approach I found. First, introduce a bias between -1 and 1 (inclusive) among the columns of g except the first column.

Now compute a synthetic score which is the sum of the products of the biases with the numbers in that column.

Next, find the biases that maximize the absolute value of the Pearson correlation coefficient computed using the first column of g and the column of synthetic scores.

Once you find the biases that maximize this absolute value of correlation, on a patient not in the training set, you can compute the synthetic score.

Find the least squares linear fit between the column of synthetic scores and the first column of g. Let's call this f.

If f(synthetic score)>0.5 then patient with that synthetic score probably has a malignant tumor. If f(synthetic score)<0.5 then patient with that synthetic score probably has a benign tumor.

Finally, one can adjust the 0.5 to be a bit different and see if this leads to a better fit of the training set.

```
m = Dimensions[g][[1]](*number of rows of g*)
n = Dimensions[g][[2]] (*number of columns of g*)
```

In[62]:=
```
y[i_] := g[[i, 1]](* First column of g *)
s[i_] := Sum[(b[j] × g[[i, j]]), {j, 2, n}];(* Synthetic score *)
r = Correlation[Table[s[i], {i, 1, m}], Table[y[i], {i, 1, m}]];
(* r is a function of the b[j] which are n-1 in number*)
```

In[40]:=
```
chop2[z_] := Chop[z, 1/(n - 1)^2] (*if z is less than (1/(n-1))^2 then it becomes zero*)
DateString[](*prints current date data*)
time1 = AbsoluteTime[];(*starts a stopwatch*)
t2 =
 Quiet[NMaximize[Join[{Abs[r]}, Table[-1 ≤ b[j] ≤ 1, {j, 2, n}]], Table[b[j], {j, 2, n}]]]
 (*numerically finds the absolute value of the maximum
   correlation r subject to the constraints -1≤b[j]≤1*)
t3 = t2 // chop2
DateString[]
time2 = AbsoluteTime[] - time1(*prints length of time elapsed since stopwatch started*)
NotebookSave[EvaluationNotebook[]];
```

Out[41]=
```
Fri 1 Mar 2024 19:59:21
```

Out[43]=
```
{0.679868, {b[2] → 0.0305554, b[3] → -0.00199494, b[4] → 0.0243235,
  b[5] → 0.00348519, b[6] → -0.563483, b[7] → -0.271741, b[8] → -0.242691,
  b[9] → -0.135231, b[10] → -0.0971756, b[11] → -0.000302599, b[12] → -0.696535,
  b[13] → -0.00339498, b[14] → -0.0673379, b[15] → -0.000240439, b[16] → 0.00240864,
  b[17] → 0.00491102, b[18] → 0.0260086, b[19] → -0.100762, b[20] → -0.0861163,
  b[21] → 0.081959, b[22] → -0.632098, b[23] → 0.00828078, b[24] → 0.177738,
  b[25] → 0.43315, b[26] → 0.0521646, b[27] → 0.11211, b[28] → 0.00623399,
  b[29] → -0.257635, b[30] → -0.0741465, b[31] → -0.0331952, b[32] → 0.141026,
  b[33] → 0.850756, b[34] → -0.875012, b[35] → -0.382207, b[36] → -0.13077,
  b[37] → -0.833291, b[38] → 0.763521, b[39] → 0.000820257, b[40] → -0.236788,
  b[41] → -0.0037755, b[42] → -0.0243798, b[43] → 0.0978243, b[44] → -0.00109136,
  b[45] → -0.00106953, b[46] → 0.134811, b[47] → -0.00034654, b[48] → 1., b[49] → -1.,
  b[50] → -0.0464082, b[51] → 0.0505254, b[52] → -0.00752409, b[53] → 0.0276737,
  b[54] → 0.0000783868, b[55] → -0.0574719, b[56] → 0.0302571, b[57] → -0.211548}}
```

Out[45]=
```
Fri 1 Mar 2024 21:55:01
```

Out[46]=
```
6940.2086947
```

In[65]:=
```
Table[B[j] = (Table[b[j], {j, 2, n}] //. t3[[2]])[[j - 1]], {j, 2, n}];
(*Sets B[j] to t3[[2]][[j-1]]*)
```

```
syntheticscore[i_] := Sum_{j=2}^{n} (B[j] g[[i, j]]) (*The B[j] are the ideal biases*)

temp2 = Table[syntheticscore[i], {i, 1, m}];
temp3 = Table[g[[i, 1]], {i, 1, m}];
Correlation[temp2, temp3];
```

In[66]:=
```
f[s_] := Evaluate[Fit[Table[{syntheticscore[i], g[[i, 1]]}, {i, 1, m}], {1, s}, s]]
f[s]
```

Out[67]=
$$0.255931 - 0.0115552\, s$$

In[68]:=
```
δ = 0.500;
prediction[i_] := If[f[ syntheticscore[i]] > δ, 1., 0.]
temp1 = Tally[Table[prediction[i] == g[[i, 1]], {i, 1, m}]] // Sort
```

Out[70]=
```
{{False, 58}, {True, 490}}
```

In[71]:=
```
δ = 0.368;
prediction[i_] := If[f[ syntheticscore[i]] > δ, 1., 0.]
temp1 = Tally[Table[prediction[i] == g[[i, 1]], {i, 1, m}]] // Sort
```

Out[73]=
```
{{False, 51}, {True, 497}}
```

In[74]:=
```
Correlation[Table[prediction[i], {i, 1, m}], Table[g[[i, 1]], {i, 1, m}]]
```

Out[74]=
```
0.68723
```