How to submit the homework

This homework has been produced to you as an .qmd. I expect you to complete the notebook and submit the following on canvas:

- the completed notebook with your answers (feel free to add anything you source code used to answer the questions)
- a .pdf file of the notebook that shows all your answers. This will make grading easier and I will appreciate that.

If you submit only one of these, you will lose points. Do the homework with quarto. Its extensions are much or other notebooks will not work.

Question 1 (one point)

$$(0,1,-1) = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) = \mathbf{u}$$

Consider the linear classifier with the weights and decision hyperplane $\mathbf{w}^T \mathbf{x} = 0$ in the figure above. In the figure, each dash = 1 in data space. Draw the decision boundary in data space. Hint: $(0, 0, 0)$ is always part of the plane.

Can this classification accuracy of this classifier be improved? If it can be improved, provide example that would improve the accuracy. If it cannot be improved, say why it cannot be improved.

ANSWER

$\mathbf{w}$ = [0.0, 0.0, 0.0]

$\mathbf{x}$ = [0.0, 0.0, 0.0]

$\mathbf{w}^T\mathbf{x}$ = 0.0

Question 2 (one point)

In question 2 we showed that a linear classifier with a decision boundary $\mathbf{w}^T \mathbf{x} = 0$ in the figure below, in the figure, $\mathbf{w} = (0, 1, -1)$ can be at most expensive power as all training number of binary classes combined is a dataset. In your answer, you will use or improve the classifier. Consider the dataset in the figure below. Each dot = 1 in data space. Each color represents a class.

1. Express the function (activation) to make it by applying one the linear combination of scalars multiplied by the matrix mistake:

$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ for $\mathbf{x}$ and $\mathbf{A}$ such that the mistake matrix $\mathbf{A}$ is the same for $\mathbf{x}$.

2. Express the function (activation) to make it by applying the linear combination of scalars multiplied by the matrix $\mathbf{A}$.
3. Can you remember your time the time the function (activation) $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ (quadratic) $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ (linear) or $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ (cubic) are identical? Hint: $\mathbf{A}$ is symmetric. Write the proof.
4. Now, can we use the sign about the expensive power of the network compared to a single node.

ANSWER

1.

$f(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$, $\mathbf{A}^T \mathbf{x}$ $\mathbf{A}\mathbf{x}$ $\mathbf{A}^T \mathbf{A} \mathbf{x}$ $\mathbf{A}^T \mathbf{A}^T \mathbf{A} \mathbf{x}$

2.

$f(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$, $\mathbf{A}^T \mathbf{x}$ $\mathbf{A}\mathbf{x}$ $\mathbf{A}^T \mathbf{A} \mathbf{x}$ $\mathbf{A}^T \mathbf{A}^T \mathbf{A} \mathbf{x}$

3.

$f(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$, $\mathbf{A}^T \mathbf{x}$ $\mathbf{A}\mathbf{x}$ $\mathbf{A}^T \mathbf{A} \mathbf{x}$ $\mathbf{A}^T \mathbf{A}^T \mathbf{A} \mathbf{x}$

4.

Yes, $\mathbf{A}^T \mathbf{A} \mathbf{x}$ $\mathbf{A}^T \

```
def make_DNF_decision(tree, get_class):
    ans = []
    for i in range(0, len(tree)):
        if tree[i] == -1:
            ans.append(get_class(tree, i, tree))
        elif tree[i] == tree[tree[i]]:
            ans.append(get_class(tree, i, tree))
    return ans

# make data
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X = data['data']
y = data['target']
feature_names = data['feature_names']

model = DNF_DTC(max_depth=2)
model.fit(X, y, feature_names=feature_names, criterion='entropy', lambda_reg=0.1, max_features=None)
print('Accuracy', model.score(X, y))
print('DNF decision')
print(model.get_DNF())
print('Tree')
print(model.tree_)

# check DNF decision
tree = model.tree_
DNF = make_DNF_decision(tree, lambda x, t, tree: [x[t[0]], x[t[1]], x[t[2]], x[t[3]], x[t[4]], x[t[5]], x[t[6]], x[t[7]], x[t[8]], x[t[9]], x[t[10]], x[t[11]], x[t[12]], x[t[13]], x[t[14]], x[t[15]], x[t[16]], x[t[17]], x[t[18]], x[t[19]], x[t[20]], x[t[21]], x[t[22]], x[t[23]], x[t[24]], x[t[25]], x[t[26]], x[t[27]], x[t[28]], x[t[29]], x[t[30]], x[t[31]], x[t[32]], x[t[33]], x[t[34]], x[t[35]], x[t[36]], x[t[37]], x[t[38]], x[t[39]], x[t[40]], x[t[41]], x[t[42]], x[t[43]], x[t[44]], x[t[45]], x[t[46]], x[t[47]], x[t[48]], x[t[49]], x[t[50]], x[t[51]], x[t[52]], x[t[53]], x[t[54]], x[t[55]], x[t[56]], x[t[57]], x[t[58]], x[t[59]], x[t[60]], x[t[61]], x[t[62]], x[t[63]], x[t[64]], x[t[65]], x[t[66]], x[t[67]], x[t[68]], x[t[69]], x[t[70]], x[t[71]], x[t[72]], x[t[73]], x[t[74]], x[t[75]], x[t[76]], x[t[77]], x[t[78]], x[t[79]], x[t[80]], x[t[81]], x[t[82]], x[t[83]], x[t[84]], x[t[85]], x[t[86]], x[t[87]], x[t[88]], x[t[89]], x[t[90]], x[t[91]], x[t[92]], x[t[93]], x[t[94]], x[t[95]], x[t[96]], x[t[97]], x[t[98]], x[t[99]], x[t[100]], x[t[101]], x[t[102]], x[t[103]], x[t[104]], x[t[105]], x[t[106]], x[t[107]], x[t[108]], x[t[109]], x[t[110]], x[t[111]], x[t[112]], x[t[113]], x[t[114]], x[t[115]], x[t[116]], x[t[117]], x[t[118]], x[t[119]], x[t[120]], x[t[121]], x[t[122]], x[t[123]], x[t[124]], x[t[125]], x[t[126]], x[t[127]], x[t[128]], x[t[129]], x[t[130]], x[t[131]], x[t[132]], x[t[133]], x[t[134]], x[t[135]], x[t[136]], x[t[137]], x[t[138]], x[t[139]], x[t[140]], x[t[141]], x[t[142]], x[t[143]], x[t[144]], x[t[145]], x[t[146]], x[t[147]], x[t[148]], x[t[149]], x[t[150]], x[t[151]], x[t[152]], x[t[153]], x[t[154]], x[t[155]], x[t[156]], x[t[157]], x[t[158]], x[t[159]], x[t[160]], x[t[161]], x[t[162]], x[t[163]], x[t[164]], x[t[165]], x[t[166]], x[t[167]], x[t[168]], x[t[169]], x[t[170]], x[t[171]], x[t[172]], x[t[173]], x[t[174]], x[t[175]], x[t[176]], x[t[177]], x[t[178]], x[t[179]], x[t[180]], x[t[181]], x[t[182]], x[t[183]], x[t[184]], x[t[185]], x[t[186]], x[t[187]], x[t[188]], x[t[189]], x[t[190]], x[t[191]], x[t[192]], x[t[193]], x[t[194]], x[t[195]], x[t[196]], x[t[197]], x[t[198]], x[t[199]], x[t[200]], x[t[201]], x[t[202]], x[t[203]], x[t[204]], x[t[205]], x[t[206]], x[t[207]], x[t[208]], x[t[209]], x[t[210]], x[t[211]], x[t[212]], x[t[213]], x[t[214]], x[t[215]], x[t[216]], x[t[217]], x[t[218]], x[t[219]], x[t[220]], x[t[221]], x[t[222]], x[t[223]], x[t[224]], x[t[225]], x[t[226]], x[t[227]], x[t[228]], x[t[229]], x[t[230]], x[t[231]], x[t[232]], x[t[233]], x[t[234]], x[t[235]], x[t[236]], x[t[237]], x[t[238]], x[t[239]], x[t[240]], x[t[241]], x[t[242]], x[t[243]], x[t[244]], x[t[245]], x[t[246]], x[t[247]], x[t[248]], x[t[249]], x[t[250]], x[t[251]], x[t[252]], x[t[253]], x[t[254]], x[t[255]], x[t[256]], x[t[257]], x[t[258]], x[t[259]], x[t[260]], x[t[261]], x[t[262]], x[t[263]], x[t[264]], x[t[265]], x[t[266]], x[t[267]], x[t[268]], x[t[269]], x[t[270]], x[t[271]], x[t[272]], x[t[273]], x[t[274]], x[t[275]], x[t[276]], x[t[277]], x[t[278]], x[t[279]], x[t[280]], x[t[281]], x[t[282]], x[t[283]], x[t[284]], x[t[285]], x[t[286]], x[t[287]], x[t[288]], x[t[289]], x[t[290]], x[t[291]], x[t[292]], x[t[293]], x[t[294]], x[t[295]], x[t[296]], x[t[297]], x[t[298]], x[t[299]], x[t[300]], x[t[301]], x[t[302]], x[t[303]], x[t[304]], x[t[305]], x[t[306]], x[t[307]], x[t[308]], x[t[309]], x[t[310]], x[t[311]], x[t[312]], x[t[313]], x[t[314]], x[t[315]], x[t[316]], x[t[317]], x[t[318]], x[t[319]], x[t[320]], x[t[321]], x[t[322]], x[t[323]], x[t[324]], x[t[325]], x[t[326]], x[t[327]], x[t[328]], x[t[329]], x[t[330]], x[t[331]], x[t[332]], x[t[333]], x[t[334]], x[t[335]], x[t[336]], x[t[337]], x[t[338]], x[t[339]], x[t[340]], x[t[341]], x[t[342]], x[t[343]], x[t[344]], x[t[345]], x[t[346]], x[t[347]], x[t[348]], x[t[349]], x[t[350]], x[t[351]], x[t[352]], x[t[353]], x[t[354]], x[t[355]], x[t[356]], x[t[357]], x[t[358]], x[t[359]], x[t[360]], x[t[361]], x[t[362]], x[t[363]], x[t[364]], x[t[365]], x[t[366]], x[t[367]], x[t[368]], x[t[369]], x[t[370]], x[t[371]], x[t[372]], x[t[373]], x[t[374]], x[t[375]], x[t[376]], x[t[377]], x[t[378]], x[t[379]], x[t[380]], x[t[381]], x[t[382]], x[t[383]], x[t[384]], x[t[385]], x[t[386]], x[t[387]], x[t[388]], x[t[389]], x[t[390]], x[t[391]], x[t[392]], x[t[393]], x[t[394]], x[t[395]], x[t[396]], x[t[397]], x[t[398]], x[t[399]], x[t[400]], x[t[401]], x[t[402]], x[t[403]], x[t[404]], x[t[405]], x[t[406]], x[t[407]], x[t[408]], x[t[409]], x[t[410]], x[t[411]], x[t[412]], x[t[413]], x[t[414]], x[t[415]], x[t[416]], x[t[417]], x[t[418]], x[t[419]], x[t[420]], x[t[421]], x[t[422]], x[t[423]], x[t[424]], x[t[425]], x[t[426]], x[t[427]], x[t[428]], x[t[429]], x[t[430]], x[t[431]], x[t[432]], x[t[433]], x[t[434]], x[t[435]], x[t[436]], x[t[437]], x[t[438]], x[t[439]], x[t[440]], x[t[441]], x[t[442]], x[t[443]], x[t[444]], x[t[445]], x[t[446]], x[t[447]], x[t[448]], x[t[449]], x[t[450]], x[t[451]], x[t[452]], x[t[453]], x[t[454]], x[t[455]], x[t[456]], x[t[457]], x[t[458]], x[t[459]], x[t[460]], x[t[461]], x[t[462]], x[t[463]], x[t[464]], x[t[465]], x[t[466]], x[t[467]], x[t[468]], x[t[469]], x[t[470]], x[t[471]], x[t[472]], x[t[473]], x[t[474]], x[t[475]], x[t[476]], x[t[477]], x[t[478]], x[t[479]], x[t[480]], x[t[481]], x[t[482]], x[t[483]], x[t[484]], x[t[485]], x[t[486]], x[t[487]], x[t[488]], x[t[489]], x[t[490]], x[t[491]], x[t[492]], x[t[493]], x[t[494]], x[t[495]], x[t[496]], x[t[497]], x[t[498]], x[t[499]], x[t[500]], x[t[501]], x[t[502]], x[t[503]], x[t[504]], x[t[505]], x[t[506]], x[t[507]], x[t[508]], x[t[509]], x[t[510]], x[t[511]], x[t[512]], x[t[513]], x[t[514]], x[t[515]], x[t[516]], x[t[517]], x[t[518]], x[t[519]], x[t[520]], x[t[521]], x[t[522]], x[t[523]], x[t[524]], x[t[525]], x[t[526]], x[t[527]], x[t[528]], x[t[529]], x[t[530]], x[t[531]], x[t[532]], x[t[533]], x[t[534]], x[t[535]], x[t[536]], x[t[537]], x[t[538]], x[t[539]], x[t[540]], x[t[541]], x[t[542]], x[t[543]], x[t[544]], x[t[545]], x[t[546]], x[t[547]], x[t[548]], x[t[549]], x[t[550]], x[t[551]], x[t[552]], x[t[553]], x[t[554]], x[t[555]], x[t[556]], x[t[557]], x[t[558]], x[t[559]], x[t[560]], x[t[561]], x[t[562]], x[t[563]], x[t[564]], x[t[565]], x[t[566]], x[t[567]], x[t[568]], x[t[569]]
    return ans

# check DNF
DNF = make_DNF
```