# Part A:

**A1:** $w_1=1$, $w_2=1$, $w_0=-1.5$.

① (0,0): $w_1 \cdot 0 + w_2 \cdot 0 - 1.5 = -1.5 < 0$
output = 0

② (1,0): $w_1 \cdot 1 + w_2 \cdot 0 - 1.5 = -0.5 < 0$
output = 0

③ (0,1): $w_1 \cdot 0 + w_2 \cdot 1 - 1.5 = -0.5 < 0$
output = 0

④ (1,1): $w_1 \cdot 1 + w_2 \cdot 1 - 1.5 = 0.5 > 0$
output = 1

**A2:**

① AND: weight: $w_1=1$, $w_2=1$
bias: $w_0=-1.5$

② NOT: weight: $w_1=-1$
bias: $w_0=0.5$

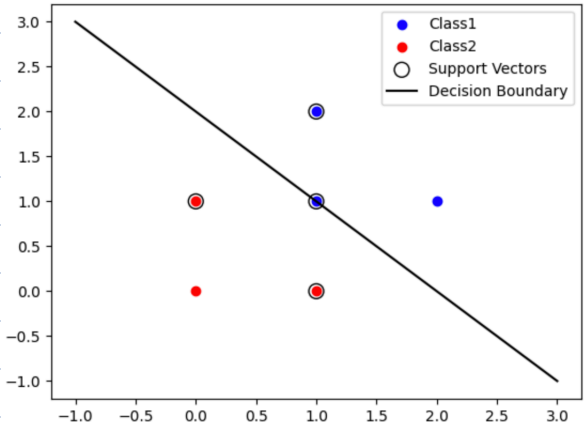
③ NAND: weight: $w_1=-1$, $w_2=-1$
bias: $w_0=1.5$

④ NOR: weight: $w_1=-1$, $w_2=-1$
bias: $w_0=0.5$

**A3:**

No, cause the single-layer perceptron is not linearly separable.

**A4:**




The support vectors are the points lie closest to the decision boundary, and define the position of the boundary.

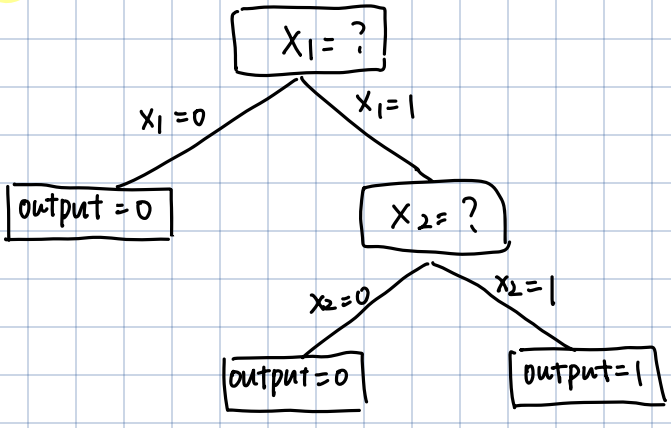
**A5:** ① Entropy (H) is calculated as:

$\because H = -\sum_i P(x_i) \log_2 P(x_i)$

$\therefore H = -[0.5 \times \log_2(0.5) + 4 \times 0.125 \times \log_2(0.125)]$
$= -[0.5 \times (-1) + 4 \times 0.125 \times (-3)]$
$= 0.5 + 1.5$
$= 2$

② Entropy measures the uncertainty in the system. A higher entropy value indicates more unpredictability.

**A6:** ① A decision tree:




② They both solve the AND function, but:
the decision tree can check each condition;
the perceptron uses a weighted sum and threshold.

**A7:**

$I_G(\text{parent}) = 1-(\frac{3}{8})^2-(\frac{5}{8})^2 = 0.47$.

① Height: small or tall

$I_G(\text{small}) = 1-(\frac{2}{3})^2-(\frac{1}{3})^2 = 0.44$

$I_G(\text{Tall}) = 1-(\frac{3}{5})^2-(\frac{2}{5})^2 = 0.48$

$IG_G = 0.47-[(\frac{3}{8})\times 0.44 + (\frac{5}{8})\times 0.48] = 0.005$

② Hair: Blonde, Dark, Red

$I_G(\text{Blonde}) = 1-(\frac{2}{4})^2-(\frac{2}{4})^2 = 0.5$

$I_G(\text{Dark}) = 1-0-1 = 0$

$I_G(\text{Red}) = 1-0-1 = 0$

$IG_G = 0.47-[(\frac{4}{8})\times 0.5 + (\frac{3}{8})\times 0 + (\frac{1}{8})\times 0] = 0.22$

③ Eyes: Brown, Blue

$I_G(\text{Brown}) = 1-(\frac{1}{3})^2-(\frac{2}{3})^2 = 0.44$

$I_G(\text{Blue}) = 1-(\frac{2}{5})^2-(\frac{3}{5})^2 = 0.48$

A8: ① $P(y=1) = \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2)}}$

$= \frac{1}{1+e^{-(-6+0.05\times40+1\times3.5)}}$

$= \frac{1}{1+e^{0.5}} \approx 0.62 \Rightarrow 62\%.$

② $0.5 = \frac{1}{1+e^{-(-6+0.05x_1+3.5\times1)}}$

$\Rightarrow 0.5(1+e^{-(-6+0.05x_1+3.5)}) = 1$

$e^{-(-6+0.05x_1+3.5)} = 1$

$\Rightarrow -(-6+0.05x_1+3.5) = 0$

$\Rightarrow x_1 = 50.$

So, this student need to learn 50 hours.

$IG_G = 0.47 - [(\frac{3}{8})\times 0.44 + (\frac{5}{8})\times 0.48] = 0.005$

∴ Hair's $IG_G$ is max, so use Hair to split.

Split the data based on Hair:

Blonde:

use Eyes to split:

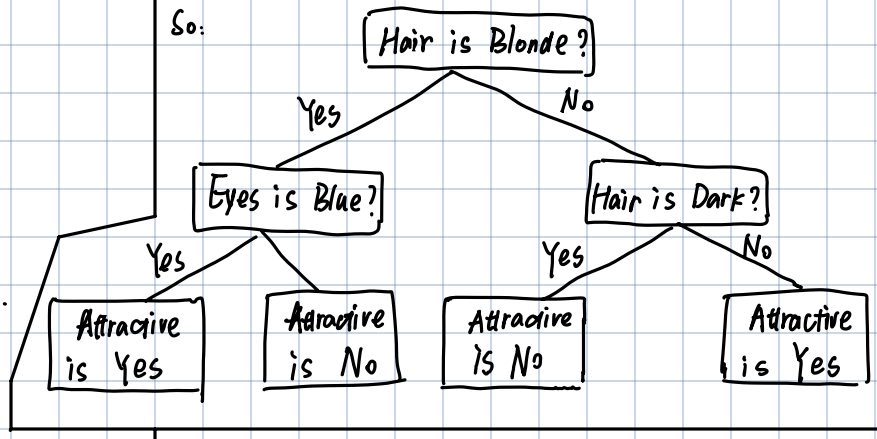
$I_G$(Brown) = 1 - 0 - 1 = 0

$I_G$(Blue) = 1 - 1 - 0 = 0

Dark: all "No"

Red: all "Yes"

So:




A9: We need to choose 1-nearest neighbors. Because it has a lower average error rate than Logistic regression.

A10: Algorithms like Logistic regression, SVM, KNN might suffer from it.

① Logistic Regression:

Impact: If some features with large scales will dominate the medol, and weaken the influence of other important features with not large scales.

Solution: Feature scaling, such as standardize or normalize like Z-score normalization.

② SVM:

Impact: Very sensitive to the scale of features, cause it needs to use kernel functions to calculate distances in feature space. If there are large differences in feature scales, then the calculation of distances will be affected.

Solution:

Feature scaling. such as standardize or normalize. Like using feature selection methods: Gini Impurity.

③ KNN:

Impact:

It uses distance measures to find the nearest

A11:

① Increase the number of weak learners, such as the number of decision trees.

② Adjust the learning rate. Can decrease it if it's too high to learning well from the data.

③ Sample weight. Need to check if the upnate of it is vaild.

A12:

Out-of-bag evaluation provides an unbiased estimate of the model's performance, don't need the separate validation set.

A13:

Hard voting classifers predict the class with the majority vote, while soft voting classifiers predict the class with the highest average probability.

neigbors. Features with large scales can dominate the calculation of distance.

Solution: Feature scaling. Like: use L1 norm instead of L2 norm.