1)

PATIENT ID	CHEST PAIN?	MALE?	SMOKES?	EXERCISES?	HEART ATTACK?
1.	yes	yes	no	yes	yes
2.	no	no	yes	no	yes
3.	yes	yes	yes	no	yes
4.	no	yes	no	yes	no
5.	yes	no	yes	yes	yes
6.	no	yes	yes	yes	no
7.	no	no	no	yes	no
8.	yes	no	yes	no	yes

(a) **(3 points)** What is the overall entropy of whether or not a patient is likely to have a heart attack, without considering any features?

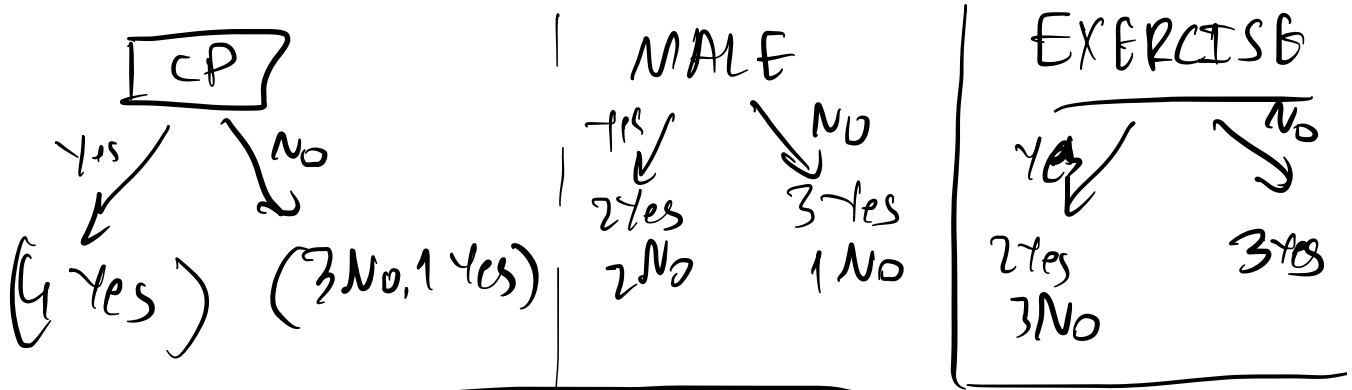
Yes → 5 $\quad P(\text{Yes}) = \frac{5}{8}$

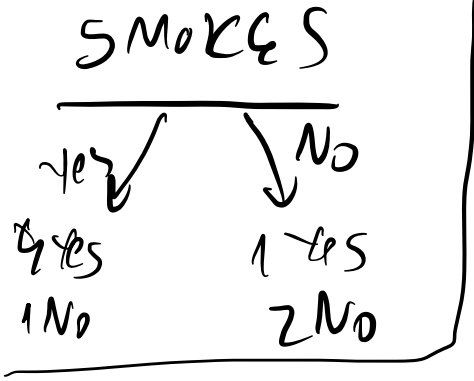
No → 3 $\quad P(\text{No}) = \frac{3}{8}$

$$H = -\left(\frac{5}{8}\log_2\frac{5}{8} + \frac{3}{8}\log_2\frac{3}{8}\right)$$

$$\simeq 0.9544$$, nearly uniform

(b) **(8 points)** Suppose we want to build a decision tree by selecting **one** feature to split. What are the information gains for the four given features, and which feature do you want to choose at the first step?





CP Entropy:

$$H(R_1) = -\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right)$$ (0)

$$H(R_2) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right)$$

$$= 0.81127$$

Next page:

$$\text{Gain (CP)} = 0.9544 - \frac{1}{2}(0) - \frac{1}{2}(0.81127)$$

$$= 0.5468$$

MALE Entropy:

$$H(R_1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

$$H(R_2) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 0.81127$$

$$\text{Gain (MALE)} = 0.9544 - \frac{1}{2}(1) - \frac{1}{2}(0.81127) = 0.0468$$

EXERCISE ENTROPY:

$$H(R_1) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.97095$$

$$H(R_2) = 0 \rightarrow \text{pure}$$

$$\text{Gain (Exercise)} = 0.9544 - \frac{1}{2}(0) - \frac{0.97095}{2}$$

$$= 0.4689$$

SMOKE ENTROPY

$$H(R_1) = -\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right) = -0.7219$$

$$H(R_2) = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) = -0.9162$$

$$\text{Gain}() = 0.1547$$

I will choose CHEST PAIN feature since it has the most information gain.

(c) **(3 points)** To construct a full decision tree, we will need to re-calculate the Information Gain for the remaining features and continue the splits, until a stop criterion is met. What are possible stop criterions for this process?

1) We can limit the depth of the tree.
2)We can set a threshold for information gain, If the gain is less than the threshold, we will stop branching for that branch.
3) We can stop when the region is pure(region consists only labels that belong to same class).
4) We can stop splitting if the number of instances in a region is below some threshold
5) We can stop splitting if the class distribution of the training points inside the region are independent of the predictors.

(d) **(3 points)** Should we standardize or normalize our features?

For this specific example, no standardization or normalization can be done since all of the features are categorical variables.

When there are numerical features, we might do standardization or normalization to our features. Note that, tree-based models are not based on the distance where the features have an effect on one another. Gino index and entropy are both used to calculate information gain, therefore normalization is not required,

(e) **(3 points)** Are decision trees robust to outliers?

Decision trees are robust to outliers, because decision trees divide items by lines, so it does not make a difference whether the point is far from the line or not. The main reason is also [redacted] splitting criteria does not care about the distance from the splitting line, however if a criteria that cares about the distance for splitting is used, decision tree might not be robust to outliers.

2 Perceptron [15 points]

Consider training a Perceptron model $y = \text{sgn}(\mathbf{w}^\top \mathbf{x})$, $\mathbf{w} \in \mathbb{R}^d$ on a dataset $D = \{(\mathbf{x}_i, y_i)\}, i = 1 \ldots 5$. Both $\mathbf{w}$ and $\mathbf{x}_i$ are vectors of dimension d, and $y \in \{+1, -1\}$ is binary. Assume that the bias term is already augmented in $\mathbf{x}_i$: $\mathbf{x}_i = [1, x_1, \ldots, x_{d-1}]$. The activation function is a sign function where $\text{sgn}(x) = 1$ for all $x > 0$ and $\text{sgn}(x) = -1$ otherwise. The Perceptron algorithm is given below,

Algorithm 1 Perceptron

```
Initialize w = 0
for i = 1 ... N do
    if y_i ≠ sgn(w^T x_i) then
        w ← w + y_i x_i
    end if
end for
return w
```

(a) **(5 points)** The Perceptron model is trained for 1 epoch, i.e. iterated over the entire dataset once, and made mistakes on the following three data points: $\{(\mathbf{x}_1, y_1), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4)\}$. What will be the weight vector $\mathbf{w}$ after this training epoch? Express $\mathbf{w}$ in using variables $\mathbf{x}_i$ and y_i where you will figure out what values of $i \in \{1, 2, 3, 4, 5\}$ will be used from the algorithm.

$$\hat{y} = \text{sign}\left(\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \\ x^5 \end{bmatrix} \right)$$

use x_1: $\text{sign}(0^\top x_1) = -1 \neq y_1 \Rightarrow y_1 = 1$

update:

$w \leftarrow 0 + x_1 \rightarrow w = x_1$

use x_2: no mistake, skip update

use x_3: $\text{sign}(w^\top x_3) = \text{sign}(x_1^\top x_3) = -y_3$

$w \leftarrow x_1 - \text{sign}(x_1^\top x_3) x_3$

use x_4: $\text{sign}(w^\top x_4) = \text{sign}\left((x_1 - \text{sign}(x_1^\top x_3) x_3)^\top x_4 \right) = -y_4$

$w \leftarrow x_1 - \text{sign}(x_1^\top x_3) x_3 - \text{sign}\left((x_1 - \text{sign}(x_1^\top x_3) x_3)^\top x_4 \right) x_4$

use x_5: no mistake, skip update

$w_f = x_1 - \text{sign}(x_1^\top x_3) x_3 - \text{sign}\left((x_1 - \text{sign}(x_1^\top x_3) x_3)^\top x_4 \right) x_4$

(b) **(5 points)** Let $d = 3$ and the data points be given as follows

i	$\mathbf{x}_{i,1}$	$\mathbf{x}_{i,2}$	$\mathbf{x}_{i,3}$	y_i
1	1	1	0	+1
2	1	2	-1	+1
3	1	1	-3	-1
4	1	3	-1	+1
5	1	1	-1	+1

Following the formulation of your answer in (a), what is $\mathbf{w}$ given the values of the data points? Express $\mathbf{w}$ as vector of numbers this time. Furthermore, if we iterate through the dataset again, will the model make a mistake on $\mathbf{x}_1$ again? Write it's prediction on $\mathbf{x}_1$.

I will use algorithm to find w, then check with formula

Use x_1: $\text{sign}(0) = -1 \rightarrow y_1 = 1 \rightarrow$ update

$$w = 0 + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

use x_3: $\text{sign}(w^T x_3) = \text{sign}\left(\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ -3 \end{bmatrix}\right) = \text{sign}(2) = 1$ ✓ wrong

$$w = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}$$

use x_4: $\text{sign}\left(\begin{bmatrix} 0 & 0 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \\ -1 \end{bmatrix}\right) = \text{sign}(-3) = -1$ wrong

$$w = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$$

$$w_f = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ -3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$$ ✓ Formula also holds

---

Iterate on x_1: $\text{sign}\left(\begin{bmatrix} 1 & 3 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 4\right) = 1$, $y_1 = 1$

Therefore, model with w_f not make a mistake on x_1 again.

$y_1' = 1 = y_1$, prediction on x_1 is 1.

(c) **(5 points)** State one difference between the given Perceptron model and the logistic regression model taught in class.

Logistic regression calculates probabilities for belonging to a class. Perceptron model can only output -1,1. Logistic regression produces smooth output between(0,1) due to sigmoid behavior and it is interpretable.There is no probability interpretation for perceptron.

3 Neural Networks [20 points]

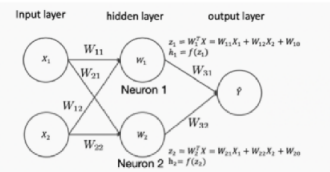


Figure 1: Neural Network

(a) **(4 points)** Refer to Lecture 13 for the activation functions of neural networks. Considering a binary classification problem where $y \in \{0, 1\}$, what are possible activations choices for the hidden and output layers repectively? Briefly explain why.

Since we are extracting features in the hidden layer, we can use any activation function for hidden layer which are sigmoid, tanh, ReLU, leaky ReLU, Generalized ReLU, softplus, swish. Note that, swish is used for bigger networks.

For output activation since we require to use binary-cross entropy for binary classification, it is appropriate to use sigmoid function to map layer outputs to probabilities.

(b) **(3 points)** Consider a binary classification problem where $y \in \{0, 1\}$. And we consider the neural network in Figure 2 with 2 inputs, 2 hidden neurons, and 1 output. We let neuron 1 use **ReLU** activiation and neuron 2 use **sigmoid** activiation, respectively. And the other layers use the linear activiation function. Suppose we have a input $X_1 = 2$ and $X_2 = -3$, with label $y = 1$. And the weights are initialized as $W_{11} = 0.9, W_{12} = 0.4, W_{21} = -1.5, W_{22} = -0.7, W_{31} = -0.2, W_{32} = 1.6$, and the bias term W_{10}, W_{20}, W_{30} are all initialized to be 0. Compute the output of the network. (Round to the 2nd decimal in your final answer.)

$z_1 = W_{11}X_1 + W_{12}X_2 + W_{10}^{0} = 0.9 * 2 + 0.4 * -3 = 0.6 \rightarrow 0.6$

$z_2 = W_{21}X_1 + W_{22}X_2 + W_{20}^{0} = -1.5 * 2 + (0.7)(-3) = -0.9 \rightarrow 0$ ✓

$z_3 = W_{31}h_1 + W_{32}h_2^{0} + W_{30}^{0} = -0.2 * 0.6 + 0 = \boxed{-0.12}$ $h_2 = 0$, $h_1 = 0.6$

$h_3 = \text{output} = \sigma(-0.12) = \frac{1}{1+e^{0.12}} = 0.4700$

(c) **(4 points)** We consider the binary cross entropy loss function. What is the loss of the network on the given data point in (b)? What is $\frac{\partial \mathcal{L}}{\partial \hat{y}}$ where $\hat{y}$ is the output of the neural network? (**Hint.** Refer to the lecture slides for the definition of a binary cross entropy loss function)

$$\mathcal{L}(y,\hat{y}) = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{1-\hat{y}}\right) = -\frac{y}{\hat{y}} + \frac{(1-y)}{1-\hat{y}}$$

$$\mathcal{L}(1, 0.47) = -(1 \log(0.47)) = \boxed{0.75502}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = -\frac{1}{0.47} = \boxed{-2.1276}$$

(d) **(6 points)** We now consider the backward pass. Given the same initialized weights and input as in (b), write the formula and calculate the derivative of the loss w.r.t W_{12}, i.e. $\frac{\partial \mathcal{L}}{\partial W_{12}}$.

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{1-\hat{y}}\right) = -2.1276$$

$$\frac{\partial \mathcal{L}}{\partial z_3} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} = 0.52999 \frac{\partial \hat{y}}{\partial z_3} = \sigma(z_3)(1-\sigma(z_3)) = 0.2491$$

$0.47 \quad (1-0.47)$

$$\frac{\partial \mathcal{L}}{h_1} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial h_1} = 0.106 \frac{\partial z_3}{\partial h_1} = w_{31} = -0.2$$

$$\frac{\partial \mathcal{L}}{z_1} = \frac{\partial \mathcal{L}}{\partial h_1} \frac{\partial h_1}{\partial z_1} = 0.106 \frac{\partial h_1}{\partial z_1} = \begin{cases} 0 & z_1 < 0 \\ 1 & z_1 > 0 \end{cases} \rightarrow 0.6 > 0 \rightarrow \boxed{1}$$

$$\frac{\partial \mathcal{L}}{w_{12}} = \frac{\partial \mathcal{L}}{z_1} \frac{\partial z_1}{\partial w_{12}} = \boxed{0.318} \frac{\partial z_1}{\partial w_{12}} = x_2 = -3$$

Formula next page:

$$\frac{\partial L}{\partial W_{12}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{1-\hat{y}}\right) * \underbrace{\sigma(z_3)(1-\sigma(z_3))}_{0.2491} * \underbrace{W_{31}}_{-0.2} * \underbrace{\mathbb{I}(z_1>0)}_{1} * \underbrace{x_2}_{-3}$$

$$= -2.1276$$

$$= \boxed{-0.318}$$

(c) **(3 points)** Given the neural network as in (b), how many parameters does the network have? (**Hint**. Each weight unit counts as a parameter, and we also consider the bias terms $(W_{10}, \ldots)$ as parameters.)

Parameters: $W_{11}, W_{12}, W_{10}, W_{21}, W_{22}, W_{20}$,
W_{31}, W_{32}, W_{30} $\Rightarrow$ 9 parameters (6 weights, 3 bias)

4 Multi-class Classification [10 points]

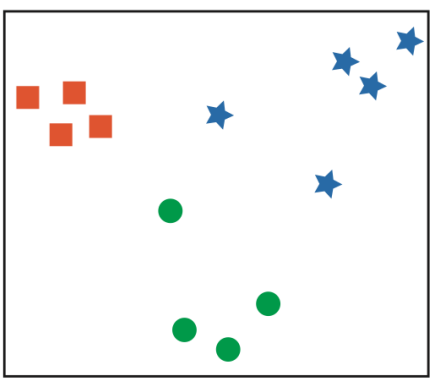


Figure 2: Multiclass Logistic Regression

(a) **(5 points)** Consider a multi-class classification problem with 5 classes and 20 features. We will use the logistic regression model to first build our binary classifier. Then, what will be the total number of parameters for using **One vs. Rest (ovr)** strategies for the multi-class classification task using logistic regression? (**Hint**. Refer to lecture 11 for multi-class logistic regression model.)

For each class, we train a single logistic regression model. Therefore, there will be 5 models. Each model will consist of a bias and multiple coefficients. In our case, there will be a single bias and 20 coefficients. Therefore, a model will consist 21 parameters including the bias. Since we have 5 models, total number of parameters for OvR is 5*21 = 105.

(b) **(5 points)** Consider a new multi-class classification problem with 3 classes. The distribution of the points is shown in the figure (Square - Class 1, Circle - Class 2, Star - Class 3). Draw the linear classifiers used for classifying the three classes, using (i) One vs. Rest (OvR), and (ii) Multinomial approaches. Your drawing does not have to be precise, roughly indicate the classifiers and jut make sure you can show the differences.

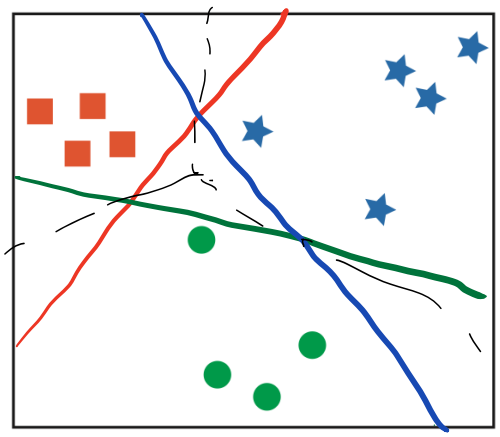
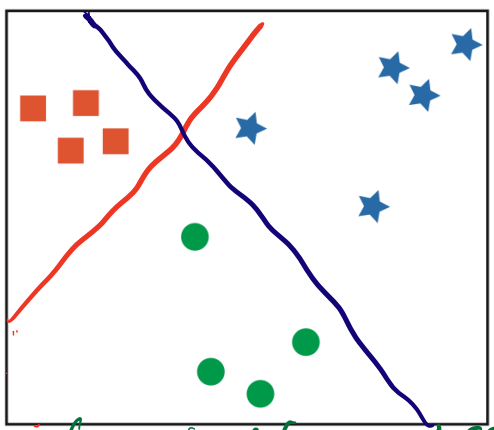


Figure 2: Multiclass Logistic Regression



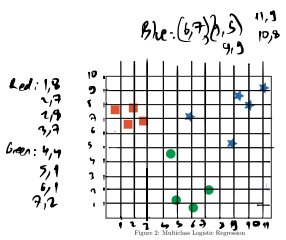
green is reference class

Figure 2: Multiclass Logistic Regression

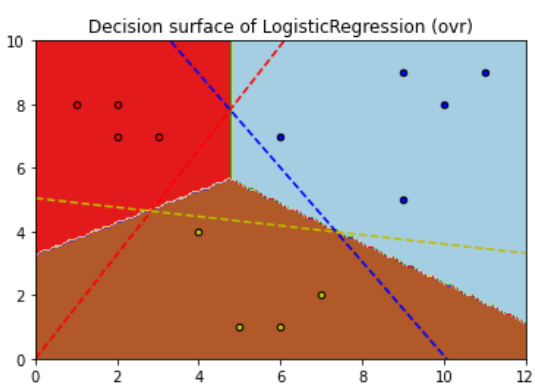
OVR

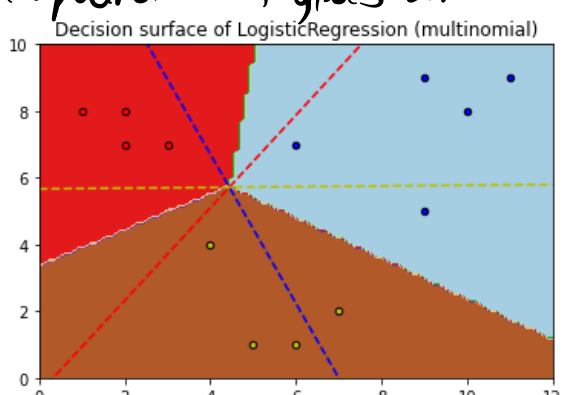
Multinomial

Checked answer with python ? Data created with below matrix



Note that multinomial implementation gives different coef.





SKLEARN

5 Decision Boundary [10 points]

Consider the classification problems with two classes, which are illustrated by circles and crosses in the plots below. In each of the plots, one of the following classification methods has been used, and the resulting decision boundary is shown:

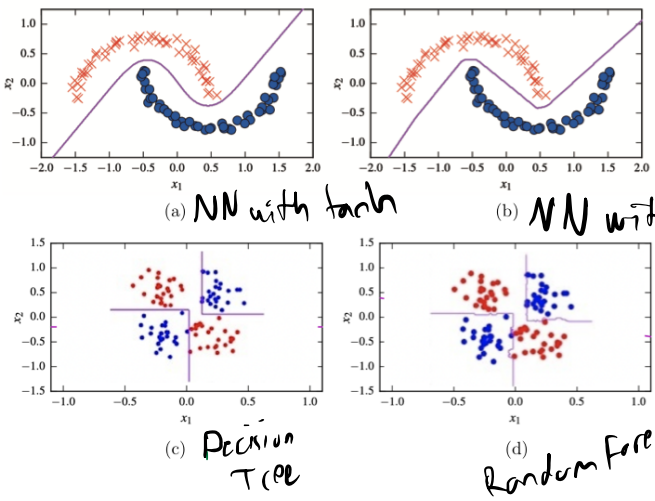
(1) **(2.5 points)** Decision Tree

(2) **(2.5 points)** Random Forest

(3) **(2.5 points)** Neural Network (1 hidden layer with 10 ReLU)

(4) **(2.5 points)** Neural Network (1 hidden layer with 10 tanh units)

Assign each of the previous methods to exactly one of the following plots (in a one to one correspondence) by annotating the plots with the respective letters, and **explain briefly** why did you make each assignment.



Firstly, a and b definitely belongs to NN and c,d definitely belongs to an algorithm with a tree involved. a and b belongs to NN because, decision boundaries are non-linear but they are not horizontal or vertical lines like in the case of c and d. Decision boundaries are sum of several linear functions in the case of a and b.

a is Neural Network (1 hidden layer with 10 tanh units) because boundary is smoother than the boundary in b. Tanh is a smooth and continuous function, therefore boundary is smoother than ReLU.

b is Neural Network(1 hidden layer with 10 ReLU) because boundary is harder than a. ReLU is a not continuous function, therefore decision boundary is not smooth.

c is Decision Tree, by looking at the graph we can easily deduct the structure of the tree and splits. Decision boundary is linear vertical and horizontal lines without any perturbations, therefore c should be Decision Tree

d is Random Forest, decision boundary is similar to decision tree, however it is easy to see the perturbations occurred on the lines due to multiple number of classifiers. When you draw a smooth line, you can easily obtain the decision tree in c.