1 True/False Questions (Add a 1 sentence justification.) [18 pts]

(a) **(3 pts)** For a continuous random variable x and its probability density function $p(x)$, it holds that $0 \leq p(x) \leq 1$ for all x.

F This only applies to discrete random variables.

(b) **(3 pts)** K-NN is a linear classification model.

F KNN can divide space into oddly shaped regions that cannot be described by a (hyper) plane.

(c) **(3 pts)** Logistic regression is a probabilistic model and we use the maximum likelihood principle to learn the model parameters.

T For logistic regression, we use the maximum likelihood principle to learn model parameters that minimize the cost function.

(d) **(3 pts)** Suppose you are given a dataset with 990 cancer-free images and 10 images from cancer patients. If you train a classifier which achieves 98% accuracy on this dataset, it is a reasonably good classifier.

F a classifier that always guesses "cancer-free" would achieve 99% accuracy.

(e) **(3 pts)** A classifier that attains 100% accuracy on the training set is always better than a classifier that attains 70% accuracy on the training set.

F The classifier that attains 100% accuracy on the training set might be overfitted.

(f) **(3 pts)** A decision tree is learned by minimizing information gain.

F a decision tree is learned by maximizing information gain.

2 Short Questions [23 pts]

(a) **(4 pts)** What is the main difference between gradient descent and stochastic gradient descent (in one sentence)? Which one require more iterations to converge, why?

Gradient descent aims to minimize error with respect to the entire dataset, while stochastic gradient descent does so with a subset of the data. Assuming we only do one subset for stochastic gradient descent, gradient descent would take longer to converge since it requires a stricter fit.

(b) **(3 pts)** What is the motivation to have a development set?

A development set can be used to cross-validate and avoid overfitting.

(c) **(3 pts)** Describe the differences between linear regression and logistic regression (in less than two sentences).

Linear regression is used to predict continuous variables while logistic regression is used to predict discrete values. Linear regression gives values while logistic regression gives probabilities.

(d) **(3 pts)** Consider the models that we have discussed in lecture: decision trees, k-NN, logistic regression, Perceptrons. If you are required to train a model that predicts the probability that the patient has cancer, which of these would you prefer, and why?

Logistic regression, because it is the one that predicts with a probability. The others don't use a probability.

(e) **(10 pts)** Given n linearly independent feature vectors in n dimensions, show that for any assignment to the binary labels you can always construct a linear classifier with weight vector w which separates the points. Assume that the classifier has the form $sign(w \cdot x)$. Hint: a set of vectors are linearly independent if no vector in the set can be defined as a linear combination of the others.

Let $X = \begin{bmatrix} - \vec{x}_1 - \\ \vdots \\ - \vec{x}_n - \end{bmatrix}$. If all $\vec{x}_i$ are linearly independent, then $X\vec{w}$ can be anything in $\mathbb{R}^n$ baud on our choice of $\vec{w}$. Therefore there is a $\vec{w}$ s.t. $\text{sign}(\vec{w}^T\vec{x}_i) = y_i \ \forall i$.

3 Decision Trees [15 pts]

For this problem, you can write your answers using $\log_2$, but it may be helpful to note that $\log_2 3 \approx 1.6$ and entropy $H(S) = -\sum_{v=1}^{K} P(S = v) \log_2 P(S = v)$. The information gain of an attribute A is $G(S, A) = H(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} H(S_v)$, where S_v is the subset of S for which A has value v.

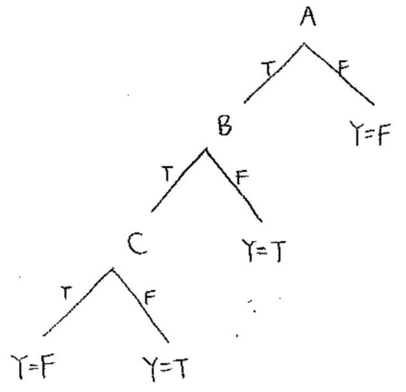
(a) We will use the dataset below to learn a decision tree which predicts the output Y, given by the binary values of A, B, C.

A	B	C	Y
F	F	F	F
T	F	T	T
T	T	~~T~~F	T
T	T	T	F

i. **(2 pts)** Calculate the entropy of the label y.

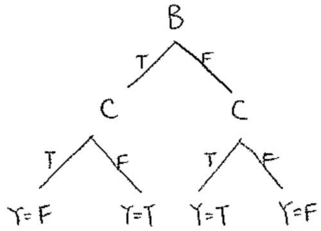
$$H(Y) = -\left[\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right]$$

ii. **(5 pts)** Draw the decision tree that will be learned using the ID3 algorithm that achieves zero training error.



iii. **(3 pts)** Is this tree optimal (i.e. does it get minimal training error with minimal depth?) explain in two sentences, and if it isn't optimal draw the optimal tree.

The above tree is not optimal. It gets minimal training error (0) but not minimal depth. The tree to the right is optimal, but it's not something the ID3 algorithm would learn.



(b) **(5 pts)** You have a dataset of 400 positive examples and 400 negative examples. Now suppose you have two possible splits. One split results in (300+, 100-) and (100+, 300-). The other choice results in (200+, 400-), and (200+, 0). Which split is most preferable and why?

The latter split is preferable because it has lower entropy.

4 Perceptron Algorithm [23 pts]

(a) **(4 pts)** Assume that you are given training data $(x, y) \in R^2 \times \{\pm 1\}$ in the following order:

Instance	1	2	3	4	5	6	7	8
Label y	$+1$	-1	$+1$	-1	$+1$	-1	$+1$	$+1$
Data (x_1, x_2)	(10, 10)	(0, 0)	(8, 4)	(3, 3)	(4, 8)	(0.5, 0.5)	(4, 3)	(2, 5)

We run the Perceptron algorithm on all the samples once, starting with an initial set of weights $w = (1, 1)$ and bias $b = 0$. On which examples, the model makes an update?

instance	$\vec{w}$	$\vec{w}^T\vec{x}+b$		
1	(1,1)	20	✓	
2	(1,1)	0	✗	} update (but no effect)
3	(1,1)	12	✓	
4	(1,1)	6	✗	update
5	(-2,-2)	-24	✗	
6	(2,6)	4	✗	
7	(0.5,5.5)	18.5	✓	
8	(0.5,5.5)	28.5	✓	

(b) **(8 pts)** Suggest a variation of the Perceptron update rule which has the following property: If the algorithm sees two consecutive occurrences of the same example, it will never make a mistake on the second occurrence. (Hint: determine an appropriate learning rate that guarantees this property). Prove your answer is correct.

The update rule is :

$w \leftarrow w +$ $2^{-i} y\vec{x}$ __________

(c) **(3 pts)** Linear separability is a pre-requisite for the Perceptron algorithm. In practice, data is almost always inseparable, such as XOR.

x_1	x_2	y
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	-1

Provide a solution to convert the inseparable data to be linearly separable. The XOR can be used for the illustration.

The data can be made linearly separable by adding an x_3 which is a non-linear transformation of x_1 and x_2.

(d) **(3 pts)** Design (specify w_0, w_1, w_2 for) a two-input Perceptron (with an additional bias or offset term) that computes "OR" Boolean functions. Is your answer the only solution?

add this bias term →

x_0	x_1	x_2	y
1	-1	-1	-1
1	1	-1	1
1	1	1	1
1	-1	1	1

$$\vec{W} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

This is not the only solution. Any multiple of $\vec{w}$ is also a solution.

(e) **(5 pts)** What is the maximal margin γ in the above OR dataset.

$\gamma = 1$

5 Logistic Regression[19 pts]

Considering the following model of logistic regression for a binary classification, with a sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$:

$$P(Y=1|X, w_0, w_1, w_2) = \sigma(w_0 + w_1X_1 + w_2X_2)$$

(a) **(3 pts)** Suppose we have learned that for the logistic regression model, $(w_0, w_1, w_2) = (-\ln(4), \ln(2), -\ln(3))$. What will be the prediction ($y = 1$ or $y = -1$) for the given $x = (1, 2)$?

$$\sigma(w_0 + w_1X_1 + w_2X_2)$$

$$= \sigma(-\ln 4 + \ln 2 - 2\ln 3)$$

$$= \sigma(\ln \tfrac{1}{4} + \ln 2 + \ln \tfrac{1}{9})$$

$$= \sigma(\ln \tfrac{1}{18})$$

$$= \frac{1}{1+e^{-\ln\frac{1}{18}}}$$

$$= \frac{1}{1+18}$$

$$= \frac{1}{19} < \frac{1}{2} \Rightarrow \tilde{y} = -1$$

(b) **(6 pts)** Is logistic regression a linear or non-linear classifier? Prove your answer.

It is a linear classifier. Logistic regression is equivalent to a linear classifier $\vec{W}^T\vec{x} + b \geq 0$ using the same weights.

(c) **(10 pts)** In the hoemwork, we mention an alternative formulation of learning a logistic regression model when $y \in \{1, 0\}$

$$\arg\min_{w} \sum_{i=1}^{m} y_i \log \sigma(w^T x_i) + (1 - y_i)\log(1 - \sigma(w^T x_i))$$

. Derive its gradient.

$$\underset{\vec{w}}{\arg\min} \sum_{i=1}^{m} \left[y_i \log \frac{1}{1+e^{-\vec{w}^T\vec{x}_i}} + (1-y_i) \log \left(1 - \frac{1}{1+e^{-\vec{w}^T\vec{x}_i}} \right) \right]$$

$$\underset{\vec{w}}{\arg\min} \sum_{i=1}^{m} \left[y_i \log \frac{1}{1+e^{-\vec{w}^T\vec{x}_i}} + (1-y_i) \log \frac{e^{-\vec{w}^T\vec{x}_i}}{1+e^{-\vec{w}^T\vec{x}_i}} \right]$$

$$\underset{\vec{w}}{\arg\min} \sum_{i=1}^{m} \left[y_i \log \frac{1}{1+e^{\vec{w}^T\vec{x}_i}} + \log \frac{e^{-\vec{w}^T\vec{x}_i}}{1+e^{\vec{w}^T\vec{x}_i}} - y_i \log \frac{e^{-\vec{w}^T\vec{x}_i}}{1+e^{-\vec{w}^T\vec{x}_i}} \right]$$

$$\underset{\vec{w}}{\arg\min} \sum_{i=1}^{m} \left[\cancel{y_i \log \frac{1}{1+e^{-\vec{w}^T\vec{x}_i}}} + \log \frac{1}{e^{\vec{w}^T\vec{x}_i}+1} - y_i \log e^{-\vec{w}^T\vec{x}_i} - \cancel{y_i \log \frac{1}{1+e^{-\vec{w}^T\vec{x}_i}}} \right]$$

$$\underset{\vec{w}}{\arg\min} \sum_{i=1}^{m} \left[-\log \left(e^{\vec{w}^T\vec{x}_i} + 1 \right) + y_i \vec{w}^T\vec{x}_i \right]$$

$0 = \frac{\partial}{\partial w_j}$

$$0 = \sum_{i=1}^{m} \left[- \frac{x_{ij} e^{\vec{w}^T\vec{x}_i}}{e^{\vec{w}^T\vec{x}_i} + 1} + y_i x_{ij} \right]$$

$$0 = \sum_{i=1}^{m} \left[- \frac{x_{ij}}{1+e^{-\vec{w}^T\vec{x}_i}} + y_i x_{ij} \right]$$

$$0 = \sum_{i=1}^{m} \left(y_i - \sigma(\vec{w}^T\vec{x}_i) \right) x_{ij}$$