Exam #1

Instructions. This is a 120-minute test. You may use your notes. You may assume anything that we proved in class or in the homework is true.

Question	Score	Points
1		10
2		10
3		10
4		10
5		10
6		10
Out Of		60

Name: ______________________________

edX Username: ______________________________

1.

(a) Recall that the Gini index for a decision tree is defined as $C(a) = 2a(1-a)$. Graph the function for $a \in [0,1]$. Is $C(a)$ a concave function? (You do not need to prove your answer.) Recall that a concave function $f : \mathbb{R} \to \mathbb{R}$ is one that satisfies

$$\forall x_1, x_2 \in \mathbb{R}, \forall \lambda \in [0,1] : f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda) f(x_2)$$

(b) Fix some training set S with Boolean labels. Prove that for any i, $\Pr_S[y = 0] = \Pr_S[x_i = 1](\Pr_S[y = 0|x_i = 1]) + \Pr_S[x_i = 0](\Pr_S[y = 0|x_i = 0])$.

(c) Recall the algorithm described from lecture (or in the book) for picking a variable to be at the root of a decision tree via the Gini index. Explain why parts (a) and (b) directly imply that the Gain for any variable x_i must be nonnegative (for partial credit simply define the Gain of variable x_i).

2. Start with vector $w = (1, 0)$ and run the Perceptron algorithm using data points 1 through 5 below (just run one pass, not several passes until convergence). What is the output after running the algorithm on points 1 through 5? Compute an estimate for its generalization error rate using held-out points 6 through 9 below. Show your work.

	Point	Label
1	$(1, 1)$	-1
2	$(-0.5, 1.5)$	$+1$
3	$(-1, 1)$	$+1$
4	$(-1, 0)$	$+1$
5	$(-1, 2)$	-1
6	$(0.5, 0.5)$	-1
7	$(3, -1)$	$+1$
8	$(0.4, 0.6)$	$+1$
9	$(0, 1)$	-1

3. In this problem we look at PAC-learning using a consistent learner. Recall that a *consistent learner* is one that, given a training data set, outputs a classifier that classifies the entire data set correctly. We work in the Boolean setting, where the domain is $\{0,1\}^n$ (Boolean vectors of length n), and the labels lie in $\{0,1\}$.

(a) Let $\mathcal{H}$ be the concept class of Boolean literals, i.e. functions of the form $h(x) = x_i$ or $h(x) = \neg x_i$. This means we first choose a coordinate i for x (x has dimension n), then we choose x_i or $\neg x_i$. How large is this model class $\mathcal{H}$?

(b) Describe a simple and efficient consistent learner for $\mathcal{H}$. That is, given any finite training set of labeled points $\{(x^1, h(x^1)), \ldots, (x^m, h(x^m))\}$ for some $h \in \mathcal{H}$, describe a procedure to come up with a function $h \in \mathcal{H}$ such that h is consistent with all the points in the training set. (We think polynomial time complexity in the dimension as efficient.)

(c) Write pseudocode describing a simple and efficient PAC-learner for $\mathcal{H}$ that makes use of the consistent learner from part (b). State its sample complexity (i.e. how many training examples it needs) as a function of n, ϵ and δ. (No need for a proof; just use the right theorem from lecture.)

(d) Let $\mathcal{H}'$ be the concept class of majorities over literals, which are functions of the form $\text{MAJ}(\ell_1, \ell_2, \ldots, \ell_n)$ where each literal ℓ_i is either x_i or $\neg x_i$. (Assume n is odd, so there are no ties.) How large is this class? Would the brute force search approach yield an efficient consistent learner for $\mathcal{H}'$?

(e) Suppose that we had a consistent learner for $\mathcal{H}'$. Describe a PAC-learner for $\mathcal{H}'$, and state its sample complexity as a function of n, ϵ and δ (use the same theorem as for (c)).

4.

(a) Suppose we have a data set consisting of three points in $\mathbb{R}^2$: $(1,2),(2,4),(3,6)$. How many principal components does this data set have? Write down the first principal component.

(b) Given the SVD of matrix

$$A = U\Sigma V^T = \begin{bmatrix} 3 & 7 & 11 \\ 6 & -1 & -5 \\ 3 & 10 & 18 \end{bmatrix}$$

$$= \begin{bmatrix} -0.531 & 0.215 & -0.819 \\ 0.162 & 0.975 & 0.150 \\ -0.832 & 0.053 & 0.553 \end{bmatrix} \cdot \begin{bmatrix} 25.0197 & 0 & 0 \\ 0 & 6.916 & 0 \\ 0 & 0 & 0.416 \end{bmatrix} \cdot \begin{bmatrix} -0.124 & -0.487 & -0.864 \\ 0.962 & 0.153 & -0.225 \\ 0.242 & -0.859 & 0.449 \end{bmatrix}.$$

Write down the matrix that is the best rank-2 approximation to A. You don't need to calculate the exact numbers, a formula or expression is enough.

5.

(a) In each of the following plots, a training set of data points X in $\mathbb{R}^2$ labeled either $+$ or $-$ is given, where the original features are the coordinates (x, y). You can assume that the data is origin-centered. For each of the two training sets below, answer the following questions:

(i) Draw all the principal components (eyeball it) or specify them using coordinates.

(ii) Can we correctly classify this dataset by using a halfspace after projecting onto one of the principal components? If so, which principal component should we project onto? If not, explain in 1–2 sentences why it is not possible.

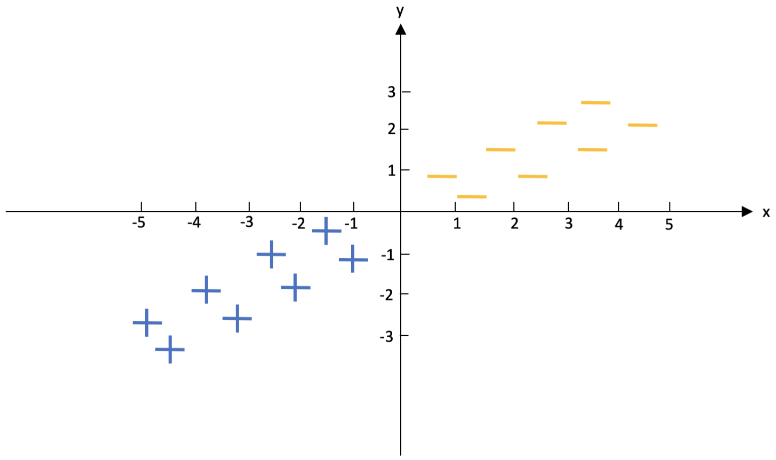

Figure 1: Dataset 1

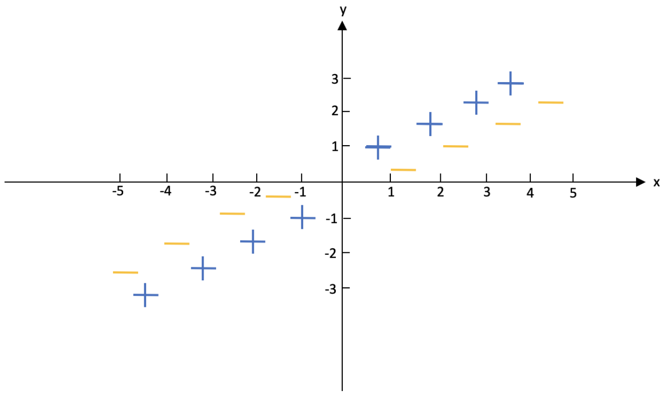

Figure 2: Dataset 2

(b) Give an example of four points in the plane (with two distinct principal components) that are linearly separable by a halfspace, but if you project onto either of the two principal components your data set is no longer linearly separable. You may simply give coordinates for these four points if you wish.

6. Regression problems.

(a) You are given a data set $S = (x_1, y_1), \ldots, (x_t, y_t)$ where each x_i and y_i are real numbers. You peform simple linear regression to obtain the line $\beta_0 + \beta_1 x$. Now re-scale the $y_i' s$ so that $y_i' = \alpha y_i$ for some real number α. Perform simple linear regression again. How do the coefficients β_0, β_1 change for the new line, quantitatively? You may reason by drawing a picture or using formulas for these coefficients from class.

(b) For each of the following scenarios, state whether or not we can use linear regression, and give a short reason.

(i) We have training data (x, y) (where $x \in \mathbb{R}^2, y \in \mathbb{R}$) satisfying $y = \alpha x_1 + \beta x_2$, and we want to learn the model parameters α, β. (That is, we have training data of the above form for various different x.)

(ii) We have training data (x, y) (where $x \in \mathbb{R}^2, y \in \mathbb{R}$) satisfying $y = \alpha x_1 x_2^3 + \beta x_2^2$, and we want to learn the model parameter α.

(iii) We have training data (x, y) (where $x \in \mathbb{R}^2, y \in \mathbb{R}$) satisfying $y = 2^\alpha x_1^\beta$, and we want to learn the model parameters α, β.