Exam #1

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

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

Name: ______________________

edX username: ______________________

UT eid: ______________________

1.

(a) Recall that the misclassification error rate function for a decision tree is $C(a) = 1 - \max(a, 1-a)$. Graph this function and its negation $-C(a)$ for $a \in [0,1]$. Is $-C(a)$ a convex function? (You do not need to prove your answer.) Recall that a convex function $f : \mathbb{R} \to \mathbb{R}$ is one that satisfies

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

(b) Fix some training set with Boolean labels S. Recall the algorithm described in lecture for building a decision tree in class. Let T_0 be the empty tree. What is the misclassification error rate of T_0 in terms of the fraction of negative examples in S? Let T_1 be the tree obtained after executing one iteration of the algorithm described in lecture using the misclassification error rate function $C(a)$ above. So, now T_1 has some variable at the root and two leaves. Prove that the misclassification error rate of T_1 is less than or equal to the misclassification error rate of T_0. (*Hint.* Use part (a); the proof is short.)

2. Start with vector $w = (1, 0)$ and run the Perceptron algorithm using data points 1 through 5 below. 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 $\{0,1\}^n$, and the labels lie in $\{0,1\}$.

(a) Let $\mathcal{H}$ be the concept class of Boolean literals, i.e. functions of the form $h_i(x) = x_i$ or $h_{\neg i}(x) = \neg x_i$. How large is this class?

(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. (Is brute force enough?)

(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 $\mathrm{MAJ}(\ell_1, \ell_2, \ldots, \ell_n)$ 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 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)). Compare your answer with the answer for $\mathcal{H}$ in part (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 (despite what the axes may suggest). For each of the two training sets below, answer the following questions:

(i) Draw all the principal components (eyeball it).

(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.

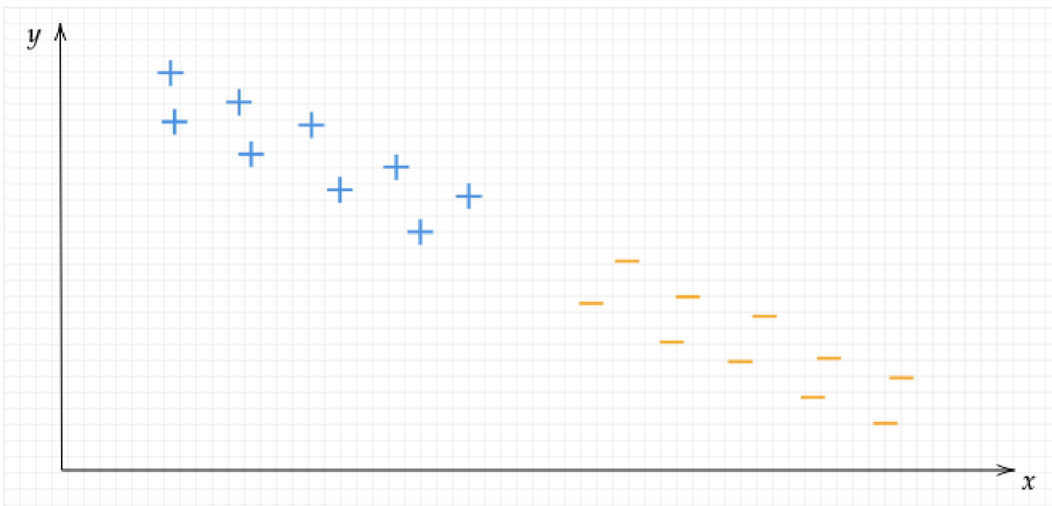

Figure 1: Dataset 1

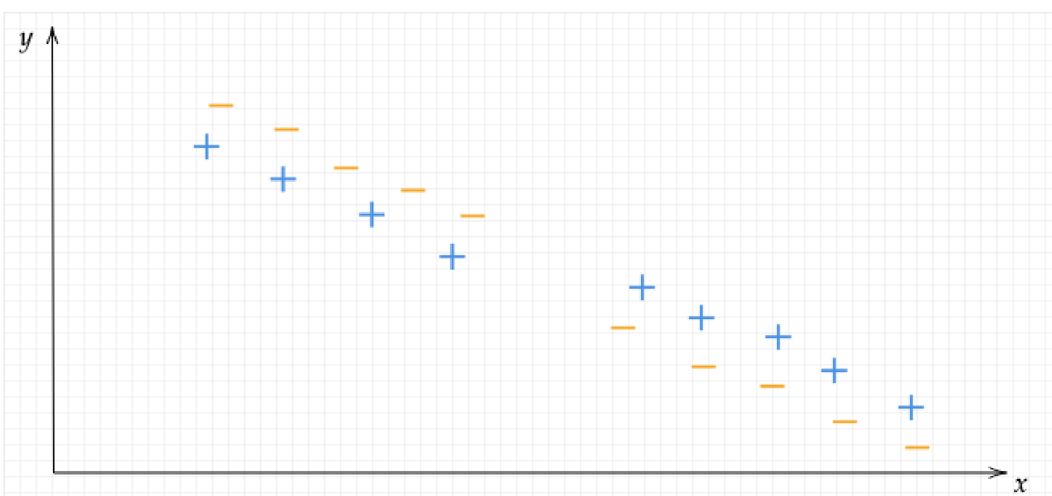

Figure 2: Dataset 2

(b) Is it possible to have a data set in $\mathbb{R}^2$ that is linearly separable by a halfspace in $\mathbb{R}^2$ but is not linearly separable after projecting onto *either* of the two principal components? If so, give a simple example along the lines of the above data sets. If not, explain in 1–2 sentences why it is not possible.

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 $x_i' s$ so that $x_i' = \alpha x_i$ 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 (or weights) 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 $y = \alpha x_1 x_2^3$, and we want to learn the model parameter (or weight) 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 (or weights) α, β.