Please answer clearly and succinctly. Show your work clearly for full credit. If an explanation is requested, think carefully before writing. Points will be removed for rambling answers with irrelevant information (and may be removed in cases of messy and hard to read answers).
If a question is unclear or ambiguous, feel free to make the additional assumptions necessary to produce the answer. State these assumptions clearly; you will be graded on the basis of the assumption as well as subsequent reasoning.
Negative points will be awarded for incorrect answers on true/false questions.
Exam is closed book, closed notes, no calculators or mechanical aids.

There are 13 questions worth **62** points, but a negative score is possible.

Problem 0 (1 point) Write your name on the top of each page.

Problem 1 (1 point) Consider a true/false question worth one point on the 446 final exam; your prior indicates a 75% chance that the answer is true. Since one point will be taken off for an incorrect answer, what is the MLE value of the question?

The answer is 1 point.

Problem 2: True or False

A) (1 point) Consider a non-uniform prior which assigns positive probability mass to each possible hypothesis. As the number of data points grows to infinity, the MLE estimate of a parameter approaches the MAP estimate to arbitrary precision.

TRUE. When the number of data points grows to infinity, we can estimate the prior perfectly.

B) (1 point) Consider a noise-free training set which can be fit perfectly by some decision tree of appropriate depth. The depth of the learned decision tree may never be larger than the number of training examples used to create the tree.

TRUE. Training examples in the tree will not appear more than one time, because of noise-free. So the number of nodes (and the depth of the tree as well) will be <= number of training examples.

C) (1 point) There is no training data set for which a decision tree learner and logistic regression will output the same decision boundary.

FALSE. Consider two data point on 2-dimension space: (-1,0) and (1,0). Then decision tree learner and logistic regression will have the same boundary x=0.

D) (1 point) Given enough training data, k-nearest neighbor learning can fit any hypothesis as long as k is odd.

TRUE. The hypothesis of k-nearest neighbor could be as complicated as you need by adding more training data. (Note you can add an identical example several times)

E) (1 point) Consider two different classifiers learning over the same training set, which contains N examples, each with m Boolean features. Suppose that the decision tree reaches 100% accuracy on the training data without any pruning. Will Naive Bayes also yield 100% accuracy on the training data (using the same features)?

False (or No). There is no guarantee at all.

Problem 3 (4 points) Consider two different classifiers learning over the same training set, which contains N examples, each with m Boolean features. Suppose that logistic regression reaches 100% accuracy on the training data. Is there a bound on the depth of a decision tree which is guaranteed to fit the training data perfectly? If so, state the depth. If there is no bound or if a decision tree is not guaranteed to yield 100% accuracy on the training data, then write infinity (∞).Please write down necessary steps.

Min (N,m).
Logistic regression reaches 100% accuracy, so the training set is noisy-free. Then every example will appear at most once in the tree, the depth is bounded by N. Also, every feature will tested at most once in the tree, so the depth is bounded by m as well.

Problem 4 Consider the following set of training examples:

Instance	**Classification**	**X_1**	**X_2**
1	+	T	T
2	+	T	T
3	-	T	F
4	-	F	F
5	-	F	T
6	-	F	F

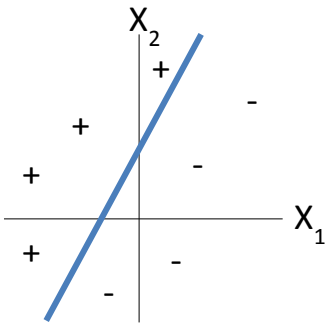
A. (3 points) What is the entropy of this collection of training examples with respect to the target function classification? Please write down necessary steps.

$$\frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2}$$

B. (3 points) What is the information gain of X_2 relative to these training examples? Please write down necessary steps.

$$\frac{1}{6}\log 3 + \frac{1}{3}\log\frac{3}{2}$$

Problem 5: (3 points) What are the weights w_0, w_1, and w_2 for the perceptron whose decision surface is illustrated below? You should assume that the decision surface crosses the X_1 axis at -5 and crosses the X_2 axis at 8.



$$w_0 + 8w_2 = 0$$
$$w_0 - 5w_1 = 0$$
$$w_0 < 0$$

Then we will get

$$w_0 = -40, w_1 = -8, w_2 = 5$$

Or anything proportional.

Problem 6: (3 points) Your friend, Joe, pulled you aside after class. "In order to reduce overfitting, I added a depth bound to my decision-tree code from PS1. It seems to work much better when I test it on the dataset from PS2" he said. "But I'm not sure why it's helping; what do you think?" Pick one of the following:

_____Having a depth bound reduces bias but increases variance.

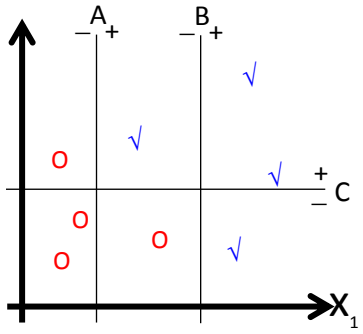
_____Having a depth bound reduces bias and also variance.

___X__Having a depth bound increases bias but reduces variance.

_____Having a depth bound increases both bias and variance.

_____Having a depth bound changes neither bias nor variance. Something else is going on.

Problem 7: (4 points; **minus 2 points for an incorrect answer**) The diagram here shows training data for a binary concept where positive examples are shown with a √ and negative examples are shown with a circle. Also shown are three decision stumps (A, B and C) each of which consists of a linear decision boundary and labels (+, -) indicating how that stump would classify the examples. Suppose that a boosting algorithm chooses A as the first stump in an ensemble and it has to decide between B and C as the next stump. Which will it choose? (Select one)



O It will clearly choose B

O It will clearly choose C

O No real preference; depends on how the tie is broken.

Problem 8: (2 points; **minus 1 point for an incorrect answer**) When learning a logistic regression classifier, you run gradient ascent for 50 iterations with the learning rate, η=0.3, and compute J(θ) after each iteration. Compute the conditional log-likelihood J(θ) after each iteration (where θ denotes the weight vectors). You find that the value of J(θ) increases quickly then levels off. Based on this, which of the following conclusions seems most plausible?

Rather than use the current value of η, it'd be more promising to try a larger value for the learning rate (say η=1.0).

η=0.3 is an effective choice of learning rate.

Rather than use the current value of η, it'd be more promising to try a smaller value (say η=0.1).

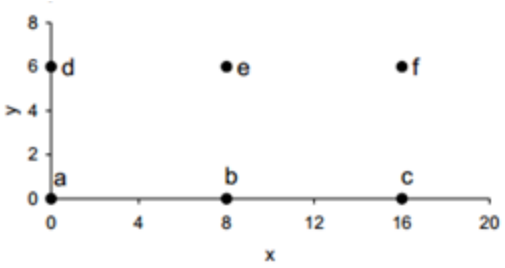
Problem 9. Suppose a 7-nearest neighbor regression search returns {7, 6, 8, 4, 7, 1, 100} as the 7 nearest *y* values for a given *x* value.

A) (2 points) What is the value of y^* that minimizes the L1 loss function on this data?

$\min \sum |y^* - y_i| \Rightarrow y^* = 7$

B) (2 points) What is the value of y^* that minimizes the L2 loss function on this data?
Average of {7, 6, 8, 4, 7, 1, 100}, which is 19.

Problem 10. (9 points) Consider the set, S, of 6 points shown in the plane below: a=(0,0), b-(8,0), c=(16,0), d=(0,6), e=(8, 6), and f=(16, 6). Suppose you run the k-means algorithm over these points with k=3 using the Euclidean distance metric (straight line distance between two points). When assigning points to centroids, ties should be broken in favor of the centroid to the left/down. Here are three definitions:



- A **k-starting configuration** is a set of k points that form the initial centroids, eg {a, b, c}.
- A **k-partition** is a partition of S into k nonempty and disjoint subsets whose union equals S. For example, {a, b, c}, {d, e}, {f} is a 3-partition.
- Supposing a k-partition induces a set of k centroids in the standard manner, we call a k-partition **stable** if a repetition of the k-means iteration with the induced centroids does not change the k-partition.

Fill in the following table

3-partition	Stable?	An example 3-starting configuration that can arrive at the 3 partition after 0 or more iterations of the k-means algorithm. (or "none" if no such 3-starting configuration exists)
{a, b}, {d, e}, {c, f}	Y	{b,c,e}
{a, b, e}, {c, d}, {f}	N	none
{a, d}, {b, e}, {c, f}	Y	{a,b,c}
{a}, {d}, {b, c, e, f}	Y	{a,b,d}
{a, b}, {d}, {c, e, f}	Y	none
{a, b, d}, {c}, {e, f}	Y	{a,c,f}

Problem 11 Consider the following 3 points in 2-d space: (1, 1), (2, 2), (3, 3)

A. (2 points) What is the first principle component? (Hint: the answer should be a normalized vector).

$(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$

B. (2 points) Suppose we now project the original data points into 1-d space defined by the principle component computed in Part A, what is the variance of the projected data (show your work)

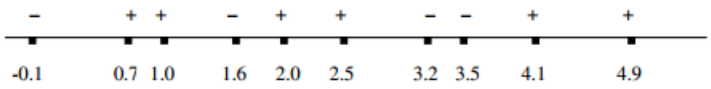
The variance is $\frac{1}{3}(2 + 0 + 2) = \frac{4}{3}$

C. (2 points) For the projected data from Part B, suppose we represent them back in the original 2-d space. What is the reconstruction error?

0

Problem 12 Consider the following dataset with one real-valued input and one binary output (+ or -). The following questions assume that we are using k-nearest-neighbor learning with unweighted Euclidean distance to predict y for an input x.

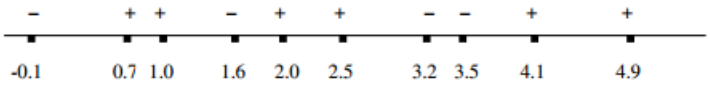
X	Y
-0.1	-
0.7	+
1.0	+
1.6	-
2.0	+
2.5	+
3.2	-
3.5	-
4.1	+
4.9	+



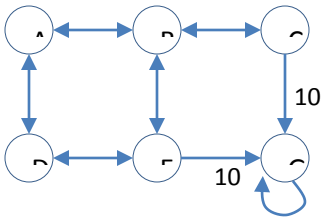
A. (2 points) What is the leave-one-out cross-validation error of 1-NN on this dataset. Give your answer as the number of misclassifications and circle them in the diagram above.

4. For each X, if the nearest neighbor has a different label, X would be misclassified.

B. (2 points) What is the leave-one-out cross-validation error of 3-NN on this dataset. Give your answer as the number of misclassifications and circle them in the diagram below.



8. For each X, consider the majority vote of 3 nearest neighbors.



Problem 13. Consider the 6-state deterministic grid world shown here. G is an absorbing goal state; once the agent enters this state, it can never leave. Possible actions are L, R, U, D signifying the different directions (Left, etc), but not all actions can be executed from every state (arrows indicate executability). Two of the transitions provide an immediate reward of (e.g. r(E,R)=10) as labeled; all other transitions have a reward of zero. (Note, in class we defined the reward function as taking three arguments, but this is unnecessary in our deterministic domain).

A) (8 points) Assuming that the discount rate, γ, is 0.8, compute the V* value of every state, the Q(s, a) value for every transition, and an optimal policy. Complete the following tables.

State	V*(s)	Policy π*(s)
A	6.4	R,D
B	8	R,D
C	10	D
D	8	R
E	10	R
G	0	self

State / Action	Q(a, s)		State/Action	Q(a,s)
A/D	6.4		D/U	5.12
A/R	6.4		D/R	8
B/L	5.12		E/L	6.4
B/D	8		E/U	6.4
B/R	8		E/R	10
C/L	6.4		G/D	0
C/D	10			

B) (1 point) True or false: It is possible to change the reward function r(s, a) in a way that alters the Q(s, a) values, but does not alter the optimal policy.
TRUE. Suppose two 10 are replaced by 20.

C) (1 point) true or false: It is possible to change r(s, a) in a way that alters Q(s, a) but does not change any value of V*.
TRUE. Change r(A,D)=-1; Then Q(A,D) would be different, but V* are the same.