Solutions

10-601 Machine Learning	**Name:**
Summer 2022	**Andrew Email:**
Exam 1 Practice Problems	**Room:**
June 8, 2022	**Seat:**
Time Limit: N/A	**Exam Number:**

Instructions:

- Fill in your name and Andrew ID above. Be sure to write neatly, or you may not receive credit for your exam.
- Clearly mark your answers in the allocated space **on the front of each page.** If needed, use the back of a page for scratch space, but you will not get credit for anything written on the back of a page. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- No electronic devices may be used during the exam.
- Please write all answers in pen.
- You have N/A to complete the exam. Good luck!

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Henry Chai

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Henry Chai

○ Marie Curie
⊗ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton
⊠ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601 10-~~7~~601

1 Decision Trees

1. To exploit the desirable properties of decision tree classifiers and perceptrons, Adam came up with a new algorithm called the "perceptron tree" that combines features from both. Perceptron trees are similar to decision trees, but each leaf node contains a perceptron rather than a majority vote.

   To create a perceptron tree, the first step is to follow a regular decision tree learning algorithm (such as ID3) and perform splitting on attributes until the specified maximum depth is reached. Once maximum depth has been reached, at each leaf node, a perceptron is trained on the remaining attributes which have not yet been used in that branch. Classification of a new example is done via a similar procedure. The example is first passed through the decision tree based on its attribute values. When it reaches a leaf node, the final prediction is made by running the corresponding perceptron at that node.

   Assume that you have a dataset with 6 binary attributes {A, B, C, D, E, F} and two output labels {-1, 1}. A perceptron tree of depth 2 on this dataset is given below. Weights of the perceptron are given in the leaf nodes. Assume bias $b = 1$ for each perceptron.

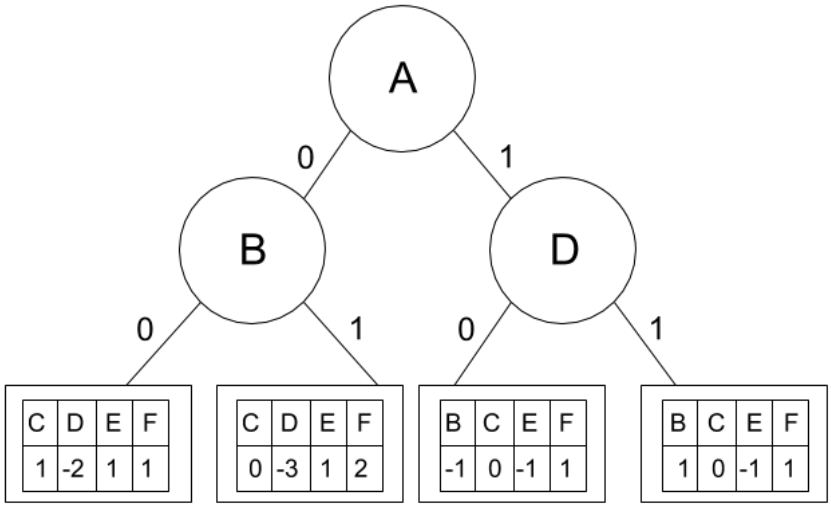


Figure 1: Perceptron Tree of depth 2

   (a) **Numerical answer:** What would the given perceptron tree predict as the output label for the sample $\mathbf{x} = [1, 1, 0, 1, 0, 1]$?

   [ ]

   1, Explanation: A=1 and D=1 so the point is sent to the right-most leaf node, where the perceptron output is (1*1)+(0*0)+((-1)*0)+(1*1)+1 = 3. Prediction = sign(3) = 1.

   (b) **True or False:** The decision boundary of a perceptron tree will *always* be linear.

◯ True

◯ False

False, since decision tree boundaries need not be linear.

(c) **True or False:** For small values of max depth, decision trees are *more* likely to underfit the data than perceptron trees.

◯ True

◯ False

True. For smaller values of max depth, decision trees essentially degenerate into majority-vote classifiers at the leaves. On the other hand, perceptron trees have the capacity to make use of "unused" attributes at the leaves to predict the correct class. Decision trees: Non-linear decision boundaries
Perceptron: Ability to gracefully handle unseen attribute values in training data/ Better generalization at leaf nodes

2. (2 points) **Select all that apply:** Given an input feature vector $\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$, you are tasked with predicting a label for y, where $y = 1$ or $y = -1$. You have no knowledge about the distributions of $\mathbf{x}$ and of y. Which of the following methods are appropriate?

☐ Perceptron

☐ k-Nearest Neighbors

☐ Linear Regression

☐ Decision Tree with unlimited depth

☐ None of the Above

k-Nearest Neighbours and Decision Tree with unlimited depth, since these two methods do not making the assumption of linear separation.

3. **True or False:** The ID3 algorithm is guaranteed to find an optimal decision tree.

◯ True

◯ False

False.

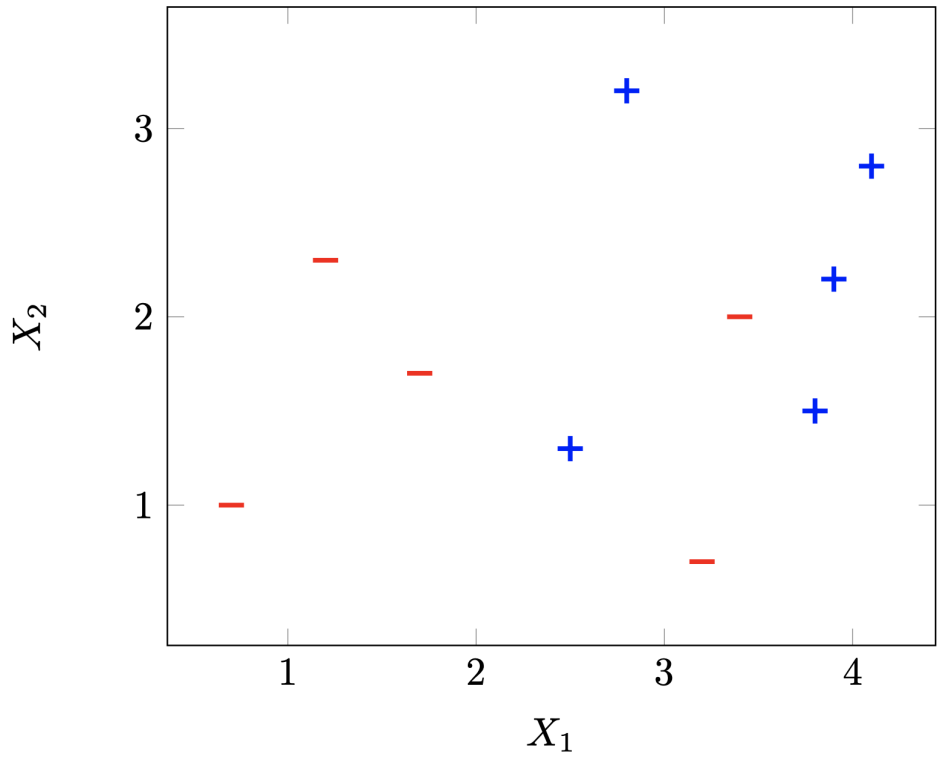
4. **True or False:** One advantage of decision trees is that they are not easy to overfit.

◯ True

◯ False

False.

5. Consider the following training data. The red '−' marks represent $Y = 0$ and the blue '+' marks represent $Y = 1$.



(a) **Numerical answer:** What is the entropy of Y in bits? (We expect you to be able to compute this value by hand.)

1 ± 0.001

(b) **Numerical answer:** What is the mutual information if we are splitting on $X_2 < 2.5$? You may write your answer as an arithmetic expression that includes $\log_2$ operations, but doesn't include symbolic information such as H.

$1 + 0.8 \cdot (5/8 \cdot \log_2(5/8) + 3/8 \cdot \log_2(3/8)) = 0.23645$

(c) **Select one:** Using error rate as the splitting criteria, which of the following is the best choice for the first binary split?

- ◯ $X_1 < 2$ and $X_1 \geq 2$
- ◯ $X_1 < 3$ and $X_1 \geq 3$

◯ $X_2 < 2.5$ and $X_2 \geq 2.5$

$X_1 < 2$ and $X_1 \geq 2$

(d) **Numerical answer:** If the tree only has one binary split (i.e. a decision stump), what would be the lowest training error rate we can get?

$0.2 + -0.001$

(e) **True or False:** It is possible to have a decision tree with zero training error for this dataset. Assume only binary splits and attributes selected with replacement.

◯ True

◯ False

True

6. The ID3 algorithm for learning decision trees greedily picks the split with the best mutual information at every node. In this question, you'll see an example of when this strategy may not result in a globally optimal tree.

(a) **Drawing:** On the training dataset below, draw the decision boundaries learned by a decision tree with a max depth of 2. The decision tree uses mutual information as its splitting criterion and breaks ties by choosing +. You may **not** use a feature twice in a single path from the root node to a leaf node in the tree.

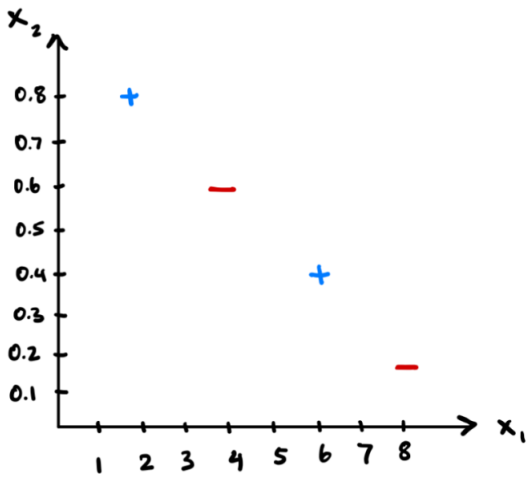
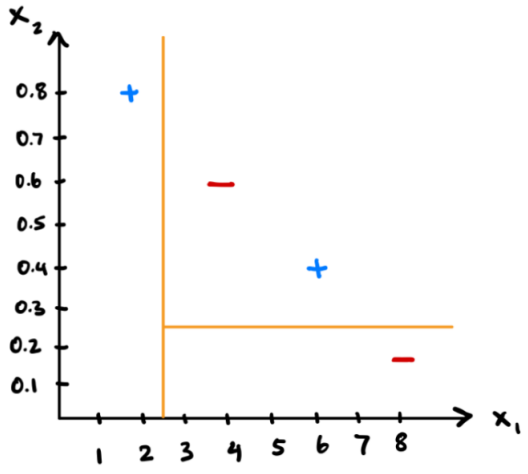


Figure 2: Dataset for Part (a)



Solution:

(b) **Numerical answer:** What is the training error of the tree drawn in part (a)?

1/4

(c) **Drawing:** On the training dataset below, draw the decision boundaries of the globally optimal tree–that is, the tree with the lowest training error. The same criteria apply from above: max depth of 2, ties are broken by in favor of +, and a feature may only be used once in a path from root to leaf.

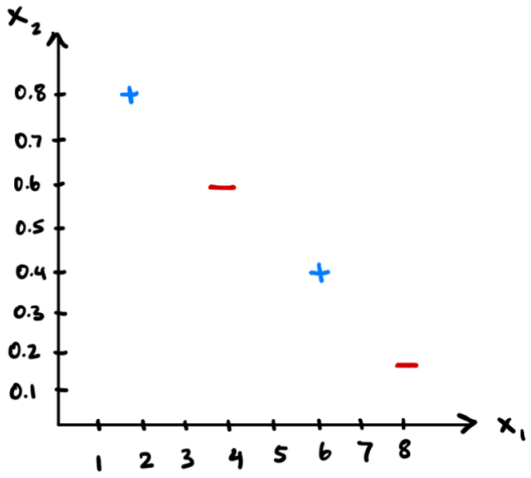
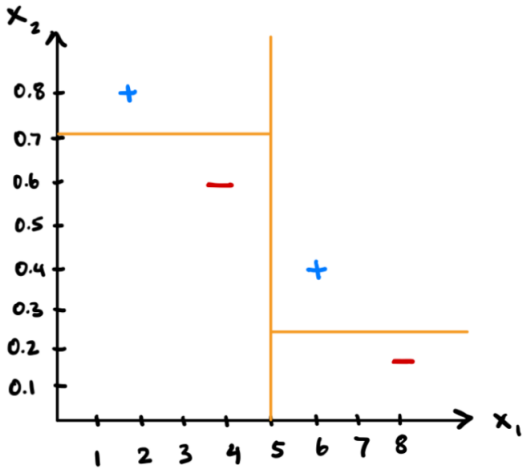


Figure 3: Dataset for Part (c)

Solution:



(d) **Numerical answer:** What is the training error of the tree drawn in part (c)?

0

2 K Nearest Neighbors

1. **Select all that apply:** Please select all that apply about k-NN in the following options. Assume a point can be its own neighbor.

   - ☐ k-NN works great with a small amount of data, but struggles when the amount of data becomes large.
   - ☐ k-NN is sensitive to outliers; therefore, in general we decrease k to avoid overfitting.
   - ☐ k-NN can be applied to classification problems but not regression problems.
   - ☐ We can always achieve zero training error (perfect classification) with k-NN, but it may not generalize well in testing.

   True: A, Curse of dimensionality; D, by setting k = 1
   False: B, we increase k to avoid overfitting; C, KNN regression

2. **Select one:** A k-Nearest Neighbor model with a large value of k is analogous to...

   - ◯ A *short* Decision Tree with a *low* branching factor
   - ◯ A *short* Decision Tree with a *high* branching factor
   - ◯ A *long* Decision Tree with a *low* branching factor
   - ◯ A *long* Decision Tree with a *high* branching factor

   A short Decision Tree with a low branching factor

3. **Select one:** Imagine you are using a k-Nearest Neighbor classifier on a dataset with lots of noise. You want your classifier to be *less* sensitive to the noise. Which of the following is likely to help and with what side effect?

   - ◯ Increase the value of $k \rightarrow$ Increase in prediction time
   - ◯ Decrease the value of $k \rightarrow$ Increase in prediction time
   - ◯ Increase the value of $k \rightarrow$ Decrease in prediction time
   - ◯ Decrease the value of $k \rightarrow$ Decrease in prediction time

   Increase the value of $k \rightarrow$ Increase in prediction time

4. **Select all that apply:** Identify the correct relationship(s) between bias, variance, and the hyperparameter k in the k-Nearest Neighbors algorithm:

   - ☐ Increasing k leads to increase in bias
   - ☐ Decreasing k leads to increase in bias
   - ☐ Increasing k leads to increase in variance
   - ☐ Decreasing k leads to increase in variance

- ☐ None of the above

A and D

5. Consider the following training dataset for a regression task:

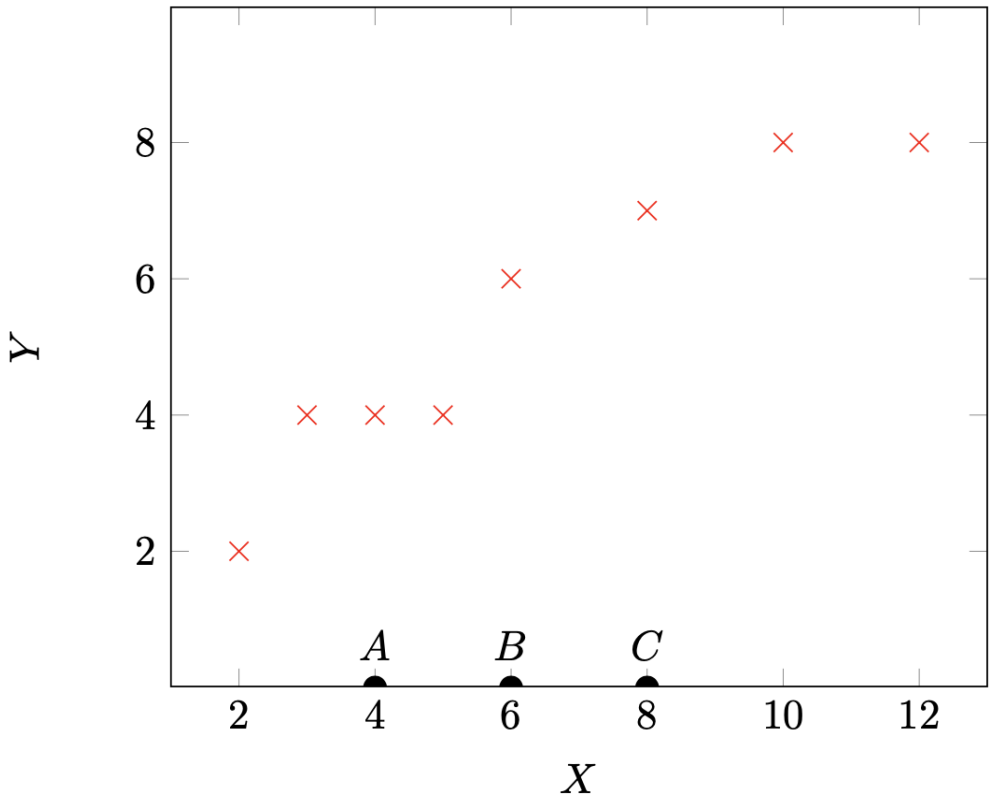
$$\mathcal{D} = \left\{ \left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(N)}, y^{(N)}\right) \right\}$$

with $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \mathbb{R}$.

For regression with k-nearest neighbors, we make predictions on unseen data points similar to the classification algorithm, but instead of a majority vote, we take the mean of the output values of the k nearest points to some new data point x. That is,

$$h(x) = \frac{1}{k} \sum_{i \in \mathcal{N}(x, \mathcal{D})} y^{(i)}$$

where $\mathcal{N}(x, \mathcal{D})$ is the set of indices of the k closest training points to x.



In the above dataset, the red ×'s denote training points and the black semi-circles A, B, C denote test points of unknown output values. For convenience, all training data points have integer input and output values.

Any ties are broken by selecting the point with the lower x value.

(a) **Numerical answer:** When $k = 1$, what is the mean squared error on the training set?

0 ± 0.00001

(b) **Numerical answer:** When $k = 2$, what is the predicted value at A?

4 ± 0.00001

(c) **Numerical answer:** When $k = 2$, what is the predicted value at B?

5 ± 0.00001

(d) **Numerical answer:** When $k = 3$, what is the predicted value at C?

7 ± 0.00001

(e) **Numerical answer:** When $k = 8$, what is the predicted value at C?

5.375 ± 0.1

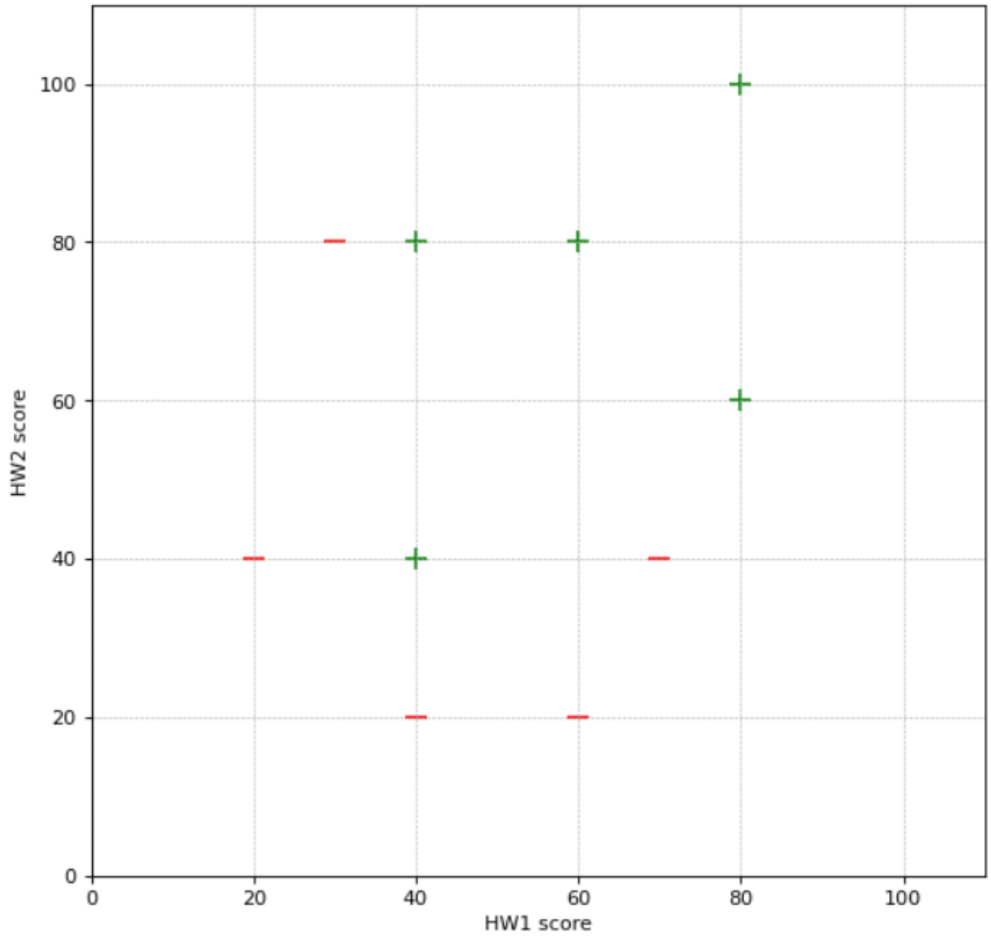
(f) **Math:** With $k = N$, for any dataset $\mathcal{D}$ with the form specified in the beginning of this question, write down a mathematical expression for the predicted value $\hat{y} = h(x)$. Your response shouldn't include a reference to the neighborhood function $\mathcal{N}()$.

$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y^{(i)}$

6. You have just enrolled into your favourite course at CMU - Introduction to Machine Learning 10-301/601 - but you have not yet decided if you want to take it for a grade or as pass/fail. You want to use your performance in HW1 and HW2 to make this decision. You follow a general rule that if you can get at least a B in the course, you will take it for a grade, and if not, you will take it as pass/fail.

You have just learnt the new classification technique, k-NN, and wish to employ it to make your decision. You start by collecting data on prior student performance in HW1 and HW2, along with their final letter grades. You then create a binary label (1/0) based on the final grades such that you assign a label of 1 if the final grade is at least a B and 0 otherwise. Next, you train the model on this data set and calculate the training error. The distance measure you use is **Euclidean distance**.

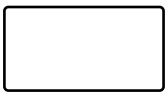
Note: For ease of computation, we will use only 10 randomly selected training data points as plotted below. Label 1 is represented by '+' and green color; label 0 is represented by '-' and red color.



(a) **Numerical answer:** What will be the training error if you choose $k = 1$?

0

(b) **Numerical answer:** What will be the training error if you choose $k = 3$?



2/10; (30,80) and (40, 40) will be miss-classified

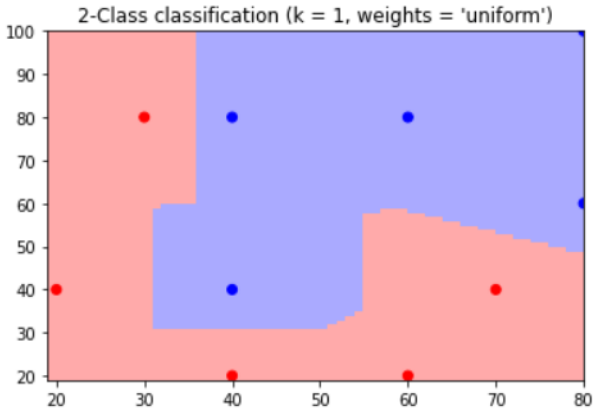
(c) **True or False:** Using Euclidean distance as the distance measure, the decision boundary of k-NN for $k = 1$ is a piece-wise straight line, that is, it contains only straight line segments. **Justify your answer.**

◯ True

◯ False

True. Because 1-NN decision boundary cuts through the center of the only two closest points or similar answers.

(d) **Drawing:** In the image above, draw a rough decision boundary for k = 1. Clearly label the + and - sides of the decision boundary.



(e) You have scored 60 in HW1 and 40 in HW2, and you now want to predict if your final grade would be at least a B.

i. **Numerical answer:** What would be the predicted class (1/0) for $k = 1$?

0

ii. **Numerical answer:** What would be the predicted class (1/0) for $k = 3$?

0

(f) **Short answer:** Looking at the training errors, you choose the model with k = 1 as it has the lowest training error. Do you think this is the right approach to select

a model? Why or why not?

No, we would use validation dataset (validation error) to choose k as k is a hyper parameter.

3 Model Selection and Errors

1. **Train and test errors:** In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained until convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

   (a) **Short Answer:** What is this scenario called?

   overfitting

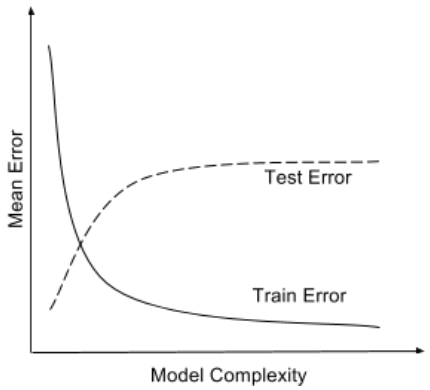
   (b) **Select all that apply:** Which of the following are expected to help?

   - ☐ Increasing the training data size.
   - ☐ Decreasing the training data size.
   - ☐ Increasing model complexity (For example, if your classifier is an SVM, use a more complex kernel. Or if it is a decision tree, increase the depth).
   - ☐ Decreasing model complexity.
   - ☐ Training on a combination of $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ and test on $\mathcal{D}^{\text{test}}$
   - ☐ None of the above

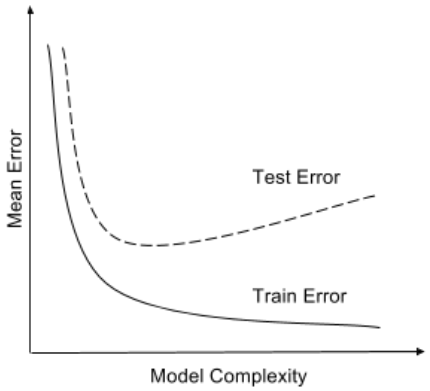
   a and d

   The model is overfitting. In order to address the problem, we can either increase training data size or decrease model complexity. We should never do (e), the model shouldn't see any testing data in the training process.

   (c) **Select one:** Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?



(a)



(b)

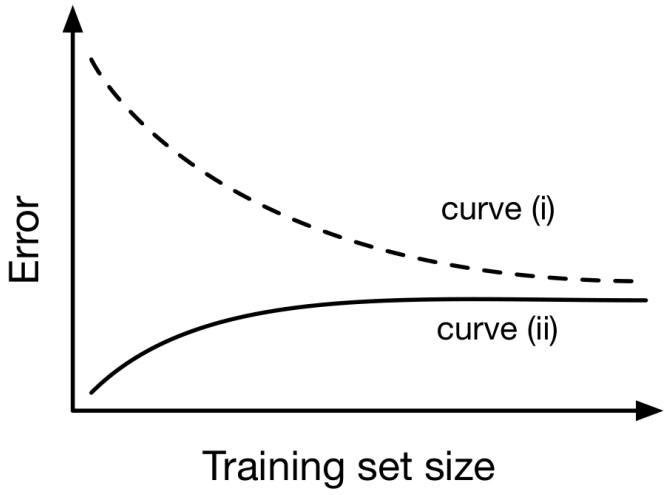
◯ Plot A

◯ Plot B

B. When model complexity increases, model can overfit better, so training error will decrease. But when it overfits too much, testing error will increase.

2. **Training Sample Size:** In this problem, we will consider the effect of training dataset size n on a logistic regression classifier with d features. The classifier is trained by optimizing the conditional log-likelihood. The optimization procedure stops if the estimated parameters perfectly classify the training data or they converge.

   The following plot shows the general trends in training and testing error as we increase the sample size $n = |S|$.



   (a) **Short Answer:** Which curve represents the training error? Provide 1-2 sentences of justification.

   Curve (ii) is the training set. Training error increases as the training set increases in size (more points to account for). However, the increase tapers out when the model generalizes well. Evidently, curve (i) is testing, since larger training sets better form generalized models, which reduces testing error.

   (b) **Short Answer:** In one word, what does the gap between the two curves represent?

   overfitting

3. What are the effects of the following on overfitting? Choose the best answer.

   (a) Increasing decision tree max depth.

   ◯ Less likely to overfit

◯ More likely to overfit

More likely to overfit

(b) Increasing decision tree mutual information split threshold.

◯ Less likely to overfit

◯ More likely to overfit

Less likely to overfit

(c) Increasing decision tree max number of nodes.

◯ Less likely to overfit

◯ More likely to overfit

More likely to overfit

(d) Increasing k in k-nearest neighbor.

◯ Less likely to overfit

◯ More likely to overfit

Less likely to overfit

(e) Increasing the training data size for decision trees. Assume that training data points are drawn randomly from the true data distribution.

◯ Less likely to overfit

◯ More likely to overfit

Less likely to overfit

(f) Increasing the training data size for 1-nearest neighbor. Assume that training data points are drawn randomly from the true data distribution.

◯ Less likely to overfit

◯ More likely to overfit

Less likely to overfit

4. Consider a learning algorithm that uses two hyperparameters, γ and ω, and it takes 1 hour to train *regardless* of the size of the training set.

   We choose to do random subsampling cross-validation, where we do K runs of cross-validation and for each run, we randomly subsample a fixed fraction αN of the dataset for validation and use the remaining for training, where $\alpha \in (0, 1)$ and N is the number of data points.

   (a) **Numerical answer:** In combination with the cross-validation method above, we choose to do grid search on discrete values for the two hyperparameters.

Given $N = 1000$ data points, $K = 4$ runs, and $\alpha = 0.25$, if we have 100 hours to complete the entire cross-validation process, what is the maximum number of discrete values of γ that we can include in our search if we also want to include 8 values of ω? Assume that any computations other than training are negligible.

$3 + -0.001$. Round $100/4/8 = 3.33$ down to 3.

(b) **Short answer:** In one sentence, give one advantage of increasing the value of α.

More data used for validation, giving a better estimate of performance on held-out data.

4 Perceptron

1. **Select all that apply:** Let $S = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(n)}, y^{(n)})\}$ be n linearly separable points by a separator through the origin in $\mathbb{R}^d$. Let S' be generated from S as: $S' = \{(c\mathbf{x}^{(1)}, y^{(1)}), \cdots, (c\mathbf{x}^{(n)}, y^{(n)})\}$, where $c > 1$ is a constant. Suppose that we would like to run the perceptron algorithm on both data sets separately, and that the perceptron algorithm converges on S. Which of the following statements are true?

   - ☐ The mistake bound of perceptron on S' is larger than the mistake bound on S
   - ☐ The perceptron algorithm when run on S and S' returns the same classifier, modulo constant factors (i.e., if $\mathbf{w}_S$ and $\mathbf{w}_{S'}$ are outputs of the perceptron for S and S', then $\mathbf{w}_S = c_1\mathbf{w}'_S$ for some constant c_1).
   - ☐ The perceptron algorithm converges on S'.
   - ☐ None of the above.

   B and C are true. Simply follow the perceptron update rule and we see that the update on $\mathbf{w}_S$ and $\mathbf{w}_{S'}$ is identical up to the constant c. A is false as the maximum margin between any point to the decision hyperplane is also scaled up by c, and the mistake bound is unchanged.

2. **True or False:** Given a linearly separable dataset, the running time of the perceptron algorithm depends on the sample size n.

   - ◯ True
   - ◯ False

   False. For a linearly separable dataset, the runtime of the perceptron algorithm does not depend on the size of the training data. Refer mistake bound concept.

3. **Select all that apply:** Which of the following are inductive biases of the perceptron algorithm?

   - ☐ Most of the cases in a small neighborhood in feature space belong to the same class.
   - ☐ The true decision boundary is linear.
   - ☐ We prefer to correct the most recent mistakes.
   - ☐ We prefer the simplest hypothesis that explains the data.
   - ☐ None of the above.

   BC

4. **True or False:** If the training data is linearly separable and representative of the true distribution, the perceptron algorithm always finds the optimal decision boundary for the true distribution.

   - ◯ True

◯ False

False.

5. (1 point) **True or False:** Consider two datasets $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, where $\mathcal{D}^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_n^{(1)}, y_n^{(1)})\}$ and $\mathcal{D}^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), ..., (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$, $d_1 > d_2$ and $n > m$. The maximum number of mistakes the perceptron algorithm will make is always higher for dataset $D^{(1)}$ than it is for dataset $D^{(2)}$.

   ◯ True

   ◯ False

   False.

6. Suppose you are given the following dataset:

Example Number	X_1	X_2	Y
1	-1	2	-1
2	-2	-2	+1
3	1	-1	+1
4	-3	1	-1

   You wish to perform the Batch Perceptron algorithm on this data. Assume you start with initial weights $\theta^T = [0, 0]$ and bias $b = 0$, and that you pass through all of the examples in order of their example number.

   i. **Numerical answer:** What would be the updated weight vector θ after we pass example 1 through the Perceptron algorithm?

   [1, −2]

   ii. **Numerical answer:** What would be the updated bias b after we pass example 1 through the Perceptron algorithm?

   −1

   iii. **Numerical answer:** What would be the updated weight vector θ after we pass example 2 through the Perceptron algorithm?

   [1, −2]

   iv. **Numerical answer:** What would be the updated bias b after we pass example 2 through the Perceptron algorithm?

-1

v. **Numerical answer:** What would be the updated weight vector θ after we pass example 3 through the Perceptron algorithm?

$[1, -2]$

vi. **Numerical answer:** What would be the updated bias b be after we pass example 3 through the Perceptron algorithm?

-1

vii. **True or False:** Your friend stops you here and tells you that you do not need to update the Perceptron weights or bias anymore; is this true or false?

◯ True

◯ False

True, all points are classified correctly.

7. **True or False:** Data (X, Y) has a non-linear decision boundary. Fortunately, there is a function $\mathcal{F}$ that maps (X, Y) to $(\mathcal{F}(X), Y)$ such that $(\mathcal{F}(X), Y)$ is linearly separable. We have tried to build a modified perceptron to classify (X, Y). Is the given (modified) perceptron update rule correct?

if sign$(w\mathcal{F}(x^{(i)}) + b)$!= $y^{(i)}$:
$w' = w + y^{(i)}\mathcal{F}(x^{(i)})$
$b' = b + y^{(i)}$

◯ True

◯ False

True

8. (a) **True or False:** All *examples* $(\mathbf{x}, y)$ that the perceptron algorithm has seen are weighted equally.

◯ True

◯ False

False. Only mistakes affect perceptron weights

(b) **True or False:** All *mistakes* the perceptron algorithm has made are weighted equally.

- ◯ True
- ◯ False

False. The most recent mistakes are more heavily weighted

5 Linear Regression

1. **Select one:** The closed form solution for linear regression is $\theta = (X^T X)^{-1} X^T y$. Suppose you have $n = 35$ training examples and $m = 5$ features (excluding the bias term). Once the bias term is folded in, what are the dimensions of X, y, θ?

   ◯ X is 35×6, y is 35×1, θ is 6×1

   ◯ X is 35×6, y is 35×6, θ is 6×6

   ◯ X is 35×5, y is 35×1, θ is 5×1

   ◯ X is 35×5, y is 35×5, θ is 5×5

   A.

2. Answer the following True/False questions, providing brief explanations to support your answers.

   (a) **True or False:** Consider a linear regression model with only one parameter, the bias, i.e., $y = \beta_0$. Then, given n data points (x_i, y_i) (where x_i is the feature and y_i is the output), minimizing the sum of squared errors results in β_0 being the median of the y_i values.

   ◯ True

   ◯ False

   False. $\sum_{i=1}^{n}(y_i - \beta_0)^2$ is the training cost, which when differentiated and set to zero gives $\beta_0 = \frac{\sum_{i=1}^{n} y_i}{n}$, the mean of the y_i values.

   (b) **True or False:** Given data $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, we obtain $\hat{w}$, the parameters that minimize the training error cost for the linear regression model $y = w^T \mathbf{x}$ we learn from $\mathcal{D}$.

   Consider a new dataset $\mathcal{D}_{\text{new}}$ generated by duplicating the points in D and adding 10 points that lie along $y = \hat{w}^T \mathbf{x}$. Then the $\hat{w}_{\text{new}}$ that we learn for $y = w^T \mathbf{x}$ from $\mathcal{D}_{\text{new}}$ is equal to $\hat{w}$.

   ◯ True

   ◯ False

True. The new squared error can be written as $2k + m$, where k is the old squared error. $m = 0$ for the 10 points that lie along the line, the lowest possible value for m. And $2k$ is least when k is least, which is when the parameters don't change.

3. **Select all that apply:** Which of the following are valid expressions for the mean squared error objective function for linear regression with dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, with each $\mathbf{x}^{(i)} \in \mathbb{R}^M$ and the design matrix $\mathbf{X} \in \mathbb{R}^{N \times (M+1)}$? $\mathbf{y}$ and $\boldsymbol{\theta}$ are column vectors.

   - ☐ $J(\boldsymbol{\theta}) = \frac{1}{N}\|\mathbf{y} - \boldsymbol{\theta}\mathbf{X}\|_2^2$
   - ☐ $J(\boldsymbol{\theta}) = \frac{1}{N}\|\mathbf{y}^T - \boldsymbol{\theta}\mathbf{X}\|_2^2$
   - ☐ $J(\boldsymbol{\theta}) = \frac{1}{N}\|\mathbf{y}^T - \mathbf{X}\boldsymbol{\theta}\|_2^2$
   - ☐ $J(\boldsymbol{\theta}) = \frac{1}{N}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$
   - ☐ None of the Above

   $J(\boldsymbol{\theta}) = \frac{1}{N}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$

4. **Numerical answer:** We have 2 data points:

$$\mathbf{x}^{(1)} = [2, 1]^T \quad y^{(1)} = 7$$
$$\mathbf{x}^{(2)} = [1, 2]^T \quad y^{(2)} = 5$$

   We know that for linear regression with a bias/intercept term and mean squared error objective function, there are infinite solutions with these two points.

   Give a specific third point $\mathbf{x}^{(3)}$, $y^{(3)}$ such that, when included with the first two, will cause linear regression to still have infinite solutions. Your $\mathbf{x}^{(3)}$ should not equal $\mathbf{x}^{(1)}$ or $\mathbf{x}^{(2)}$ and your $y^{(3)}$ should not equal $y^{(1)}$ or $y^{(2)}$.

   $x_1^{(3)}$

   [ ]

   $x_2^{(3)}$

   [ ]

   $y^{(3)}$

   [ ]

   Any $\mathbf{x}^{(3)}$ that is colinear with the first two $\mathbf{x}$'s; y doesn't matter.

   Select one: After adding your third point, if we then double the output of just the first point such that now $y^{(1)} = 14$, will this change the number of solutions for linear regression?

   - ◯ Yes
   - ◯ No

   No

5. Given that we have an input x and we want to estimate an output y, in linear regression we assume the relationship between them is of the form $y = wx + b + \epsilon$, where w and b are real-valued parameters we estimate and ϵ represents the noise in the data. When the noise is Gaussian, maximizing the likelihood of a dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to estimate the parameters w and b is equivalent to minimizing the squared error:

$$\arg\min_w \sum_{i=1}^{n} (y_i - (wx_i + b))^2.$$

Consider the dataset S plotted in Fig. 5 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 7, indicate which regression line (relative to the original one) in Fig. 6 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line	(b)	(c)	(b)	(a)	(a)

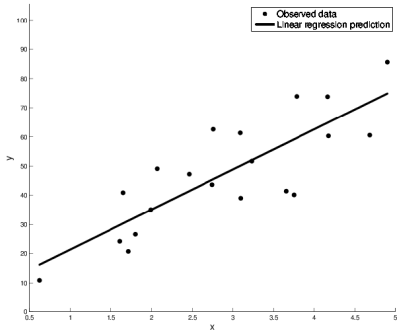
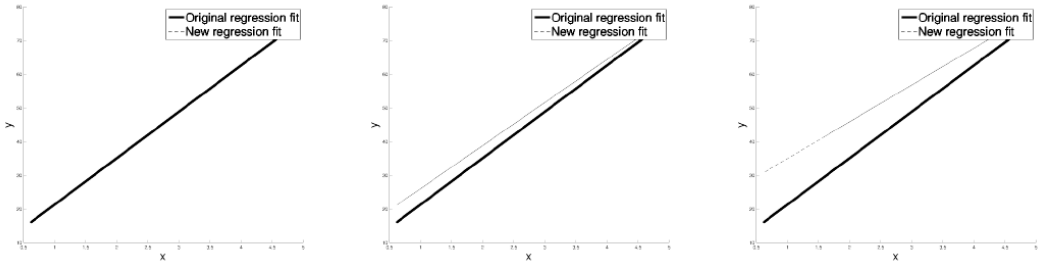
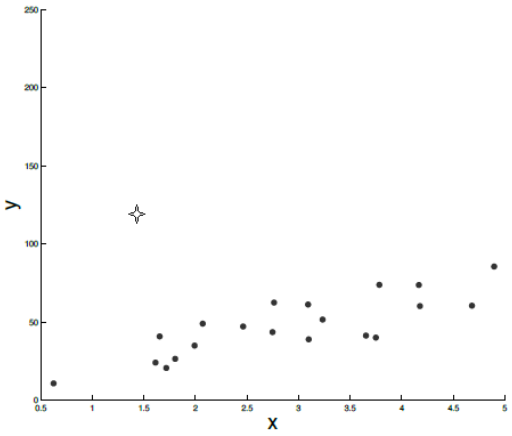


Figure 5: An observed data set and its associated regression line.

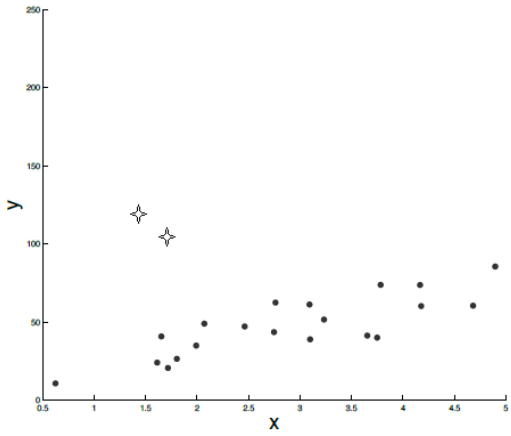


(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

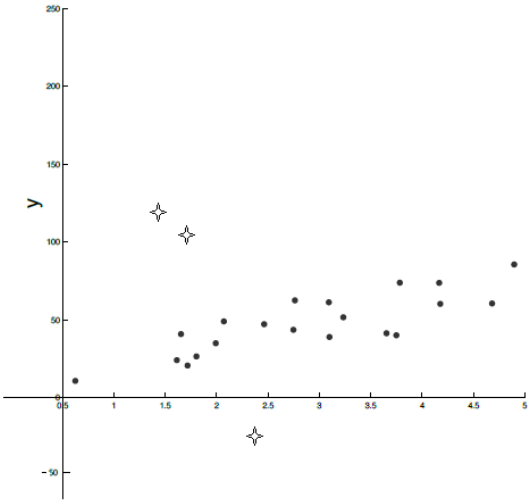
Figure 6: New regression lines for altered data sets S^{new}.



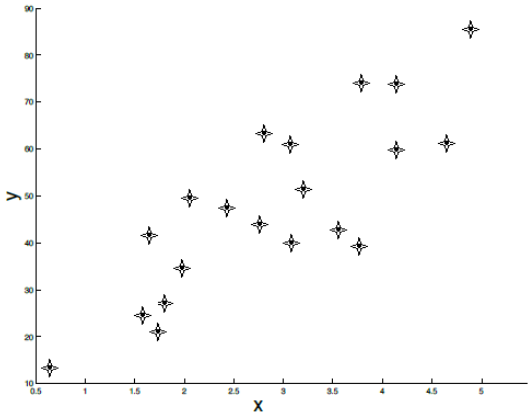
(a) Adding one outlier to the original data set.



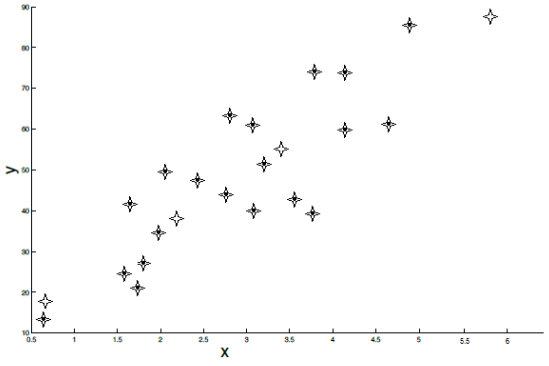
(b) Adding two outliers to the original data set.



(c) Adding three outliers to the original data set. Two on one side and one on the other side.



(d) Duplicating the original data set.



(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

Figure 7: New data set S^{new}.

6 Optimization

1. **Select all that apply:** Which of the following are correct regarding Gradient Descent (GD) and stochastic gradient descent (SGD)?

   - ☐ Each update step in SGD pushes the parameter vector closer to the parameter vector that minimizes the objective function.
   - ☐ The gradient computed in SGD is, in expectation, equal to the gradient computed in GD.
   - ☐ The gradient computed in GD has a higher variance than that computed in SGD, which is why in practice SGD converges faster in time than GD.
   - ☐ None of the above.

   B.

   A is incorrect, SGD updates are high in variance and may not go in the direction of the true gradient. C is incorrect, for the same reason.

2. (a) **Select all that apply:** Determine if the following 1-D functions are convex. Assume that the domain of each function is $\mathbb{R}$. The definition of a convex function is as follows:

   $f(x)$ is convex $\iff f(\alpha x + (1-\alpha)z) \leq \alpha f(x) + (1-\alpha)f(z), \forall \alpha \in [0,1]$ and $\forall x, z$.

   - ☐ $f(x) = x + b$ for any $b \in \mathbb{R}$
   - ☐ $f(x) = c^2 x$ for any $c \in \mathbb{R}$
   - ☐ $f(x) = ax^2 + b$ for any $a \in \mathbb{R}$ and any $b \in \mathbb{R}$
   - ☐ $f(x) = 0$
   - ☐ None of the above

   A, B, D

   $f(x) = x + b$ for any $b \in \mathbb{R}$, $f(x) = c^2 x$ for any $c \in \mathbb{R}$, $f(x) = 0$.

   (b) **Select all that apply:** Consider the convex function $f(z) = z^2$. Let α be our learning rate in gradient descent.

   For which values of α will $\lim_{t\to\infty} f(z^{(t)}) = 0$, assuming the initial value of z is $z^{(0)} = 1$ and $z^{(t)}$ is the value of z after the t-th iteration of gradient descent?

   - ☐ $\alpha = 0$
   - ☐ $\alpha = \frac{1}{2}$
   - ☐ $\alpha = 1$
   - ☐ $\alpha = 2$

☐ None of the above

$\alpha = \frac{1}{2}$

(c) **Numerical answer:** Give the range of all values for $\alpha \geq 0$ such that $\lim_{t \to \infty} f(z^{(t)}) = 0$, assuming the initial value of z is $z^{(0)} = 1$.

$(0, 1)$.