

Welcome to the first lab assignment for DSC 140A. Labs are meant to be short recaps of the essential content from the week's lectures. They are designed to take less than 30 minutes to complete.

Consider the following data set:

x	y
-4	4
-3	3
-1	2
0	2
2	2
3	4
5	8
7	10

Suppose k -nearest neighbor regression is used to predict the value of y at $x = 4$ using $k = 3$. What will be the prediction? State your answer as a decimal number.

—

The three nearest neighbors to $x = 4$ are at $x = 2$, $x = 3$, and $x = 5$. Their labels are $y = 2$, $y = 4$, and $y = 8$, respectively. Since we're doing *regression*, we average these labels to make a prediction, and the average of 2, 4, and 8 is $14/3$, or 4.66.

Let $\mathcal{X} = \{(\vec{x}^{(1)}, y_1), \dots, (\vec{x}^{(n)}, y_n)\}$ be a labeled dataset, where $\vec{x}^{(i)} \in \mathbb{R}^d$ is a feature vector and $y_i \in \{-1, 1\}$ is a binary label. Let \vec{x} be a new point that is not in the data set. Suppose a nearest neighbor classifier is used to predict the label of \vec{x} , and the resulting prediction is -1 . (You may assume that there is a unique nearest neighbor of \vec{x} .)

Now let \mathcal{Z} be a new dataset obtained from \mathcal{X} by subtracting the same vector $\vec{\delta}$ from each training point. That is, $\mathcal{Z} = \{(\vec{z}^{(1)}, y_1), \dots, (\vec{z}^{(n)}, y_n)\}$, where $\vec{z}^{(i)} = \vec{x}^{(i)} - \vec{\delta}$ for each i . Let $\vec{z} = \vec{x} - \vec{\delta}$.

Suppose a nearest neighbor classifier trained on \mathcal{Z} is used to predict the label of \vec{z} .

True or False: the predicted label of \vec{z} must also be -1 .

True

False

Subtracting the same vector $\vec{\delta}$ from every training vector has the effect of "shifting" the entire training set. But this shift doesn't affect distances: if a training point was the nearest neighbor of \vec{x} in the original data set, it will still be the nearest neighbor to $\vec{x} - \vec{\delta} = \vec{z}$ in the "shifted" data, so the predicted label stays the same. See the math review in the first discussion for why subtracting the same vector shifts the data.

Also, it's worth mentioning that this question was also meant to be practice in "unpacking" a formal, mathematical description into something we can more intuitively understand. This is something we do a lot in ML, because to precisely describe an algorithm, we have to use the language of math.

As above, let $\mathcal{X} = \{(\vec{x}^{(1)}, y_1), \dots, (\vec{x}^{(n)}, y_n)\}$ be a labeled dataset, where $\vec{x}^{(i)} \in \mathbb{R}^d$ is a feature vector and $y_i \in \{-1, 1\}$ is a binary label. Let \vec{x} be a new point that is not in the data set. Suppose a nearest neighbor classifier is used to predict the label of \vec{x} , and the resulting prediction is -1 . (You may assume that there is a unique nearest neighbor of \vec{x} .)

Now suppose we standardize the training data by subtracting the mean and dividing by the standard deviation of each feature. Let \mathcal{Z} be the standardized version of \mathcal{X} . Suppose a nearest neighbor classifier trained on \mathcal{Z} is used to predict the label of \vec{x} .

True or False: the predicted label of \vec{x} must also be -1 .

True

False

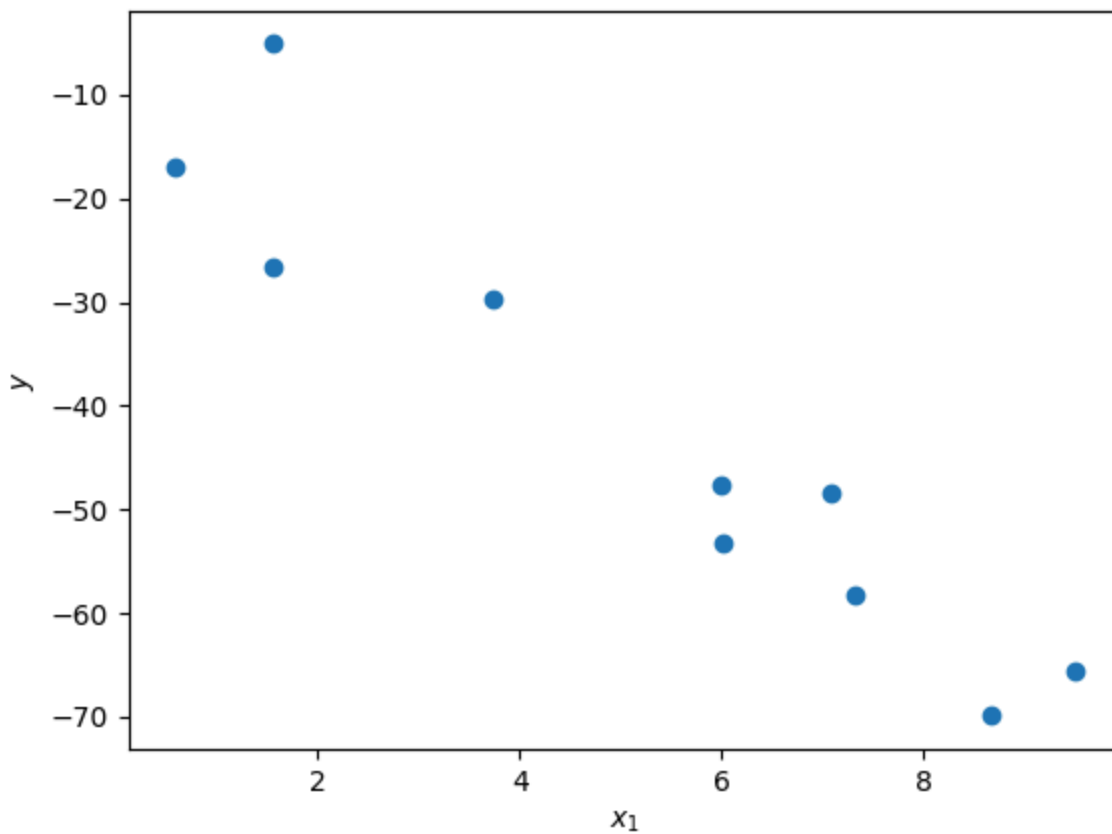
Scaling the features independently of one another *does* affect distances, as we saw in lecture.

Therefore, the prediction may change if we standardize. In fact, standardization may be necessary to improve the predictive performance of a nearest neighbor predictor.

Let $\vec{w} = (2, 4, 1, -3)^T$ be the parameter vector of a linear predictor $H(\vec{x}) = \vec{w} \cdot \text{Aug}(\vec{x})$. What is $H(\vec{x})$ if $\vec{x} = (3, 0, 1)$?

To make a prediction on a new point \vec{x} , we compute the dot product of the parameter vector \vec{w} with the augmented feature vector $\text{Aug}(\vec{x})$. The augmented feature vector is just \vec{x} with a 1 appended to the front, so $\text{Aug}(\vec{x}) = (1, 3, 0, 1)$. The dot product of \vec{w} with this augmented feature vector is $2 \cdot 1 + 4 \cdot 3 + 1 \cdot 0 + -3 \cdot 1 = 11$.

Consider the data shown below:



Which of the following plots could show the mean squared error $R(\vec{w})$ of a linear predictor $H(\vec{x}) = \text{Aug}(\vec{x}) \cdot \vec{w} = w_0 + w_1 x_1$ with respect to this data set?

()

(x)

()

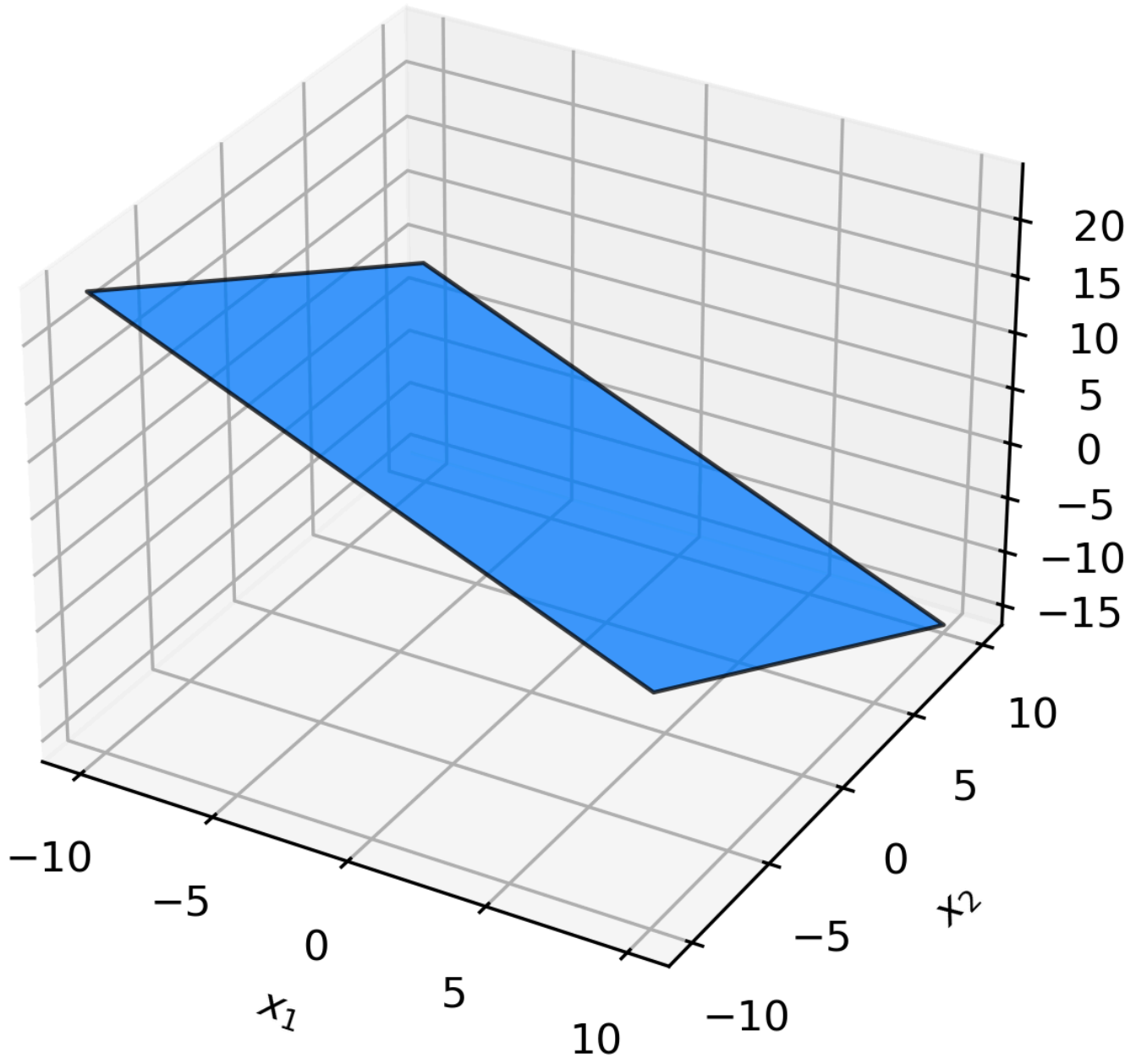
()

These plots show the "risk surface". It plots the risk for all possible choices of w_0 and w_1 (within some range). The key thing to understand for answering this question is that the minimum point on the risk surface corresponds to the choice of parameters w_0 and w_1 that "best fits" the training data (in the sense of minimizing the mean squared error). So, we are looking for the risk surface whose minimum is at a choice of w_0 and w_1 that seems to fit the data well.

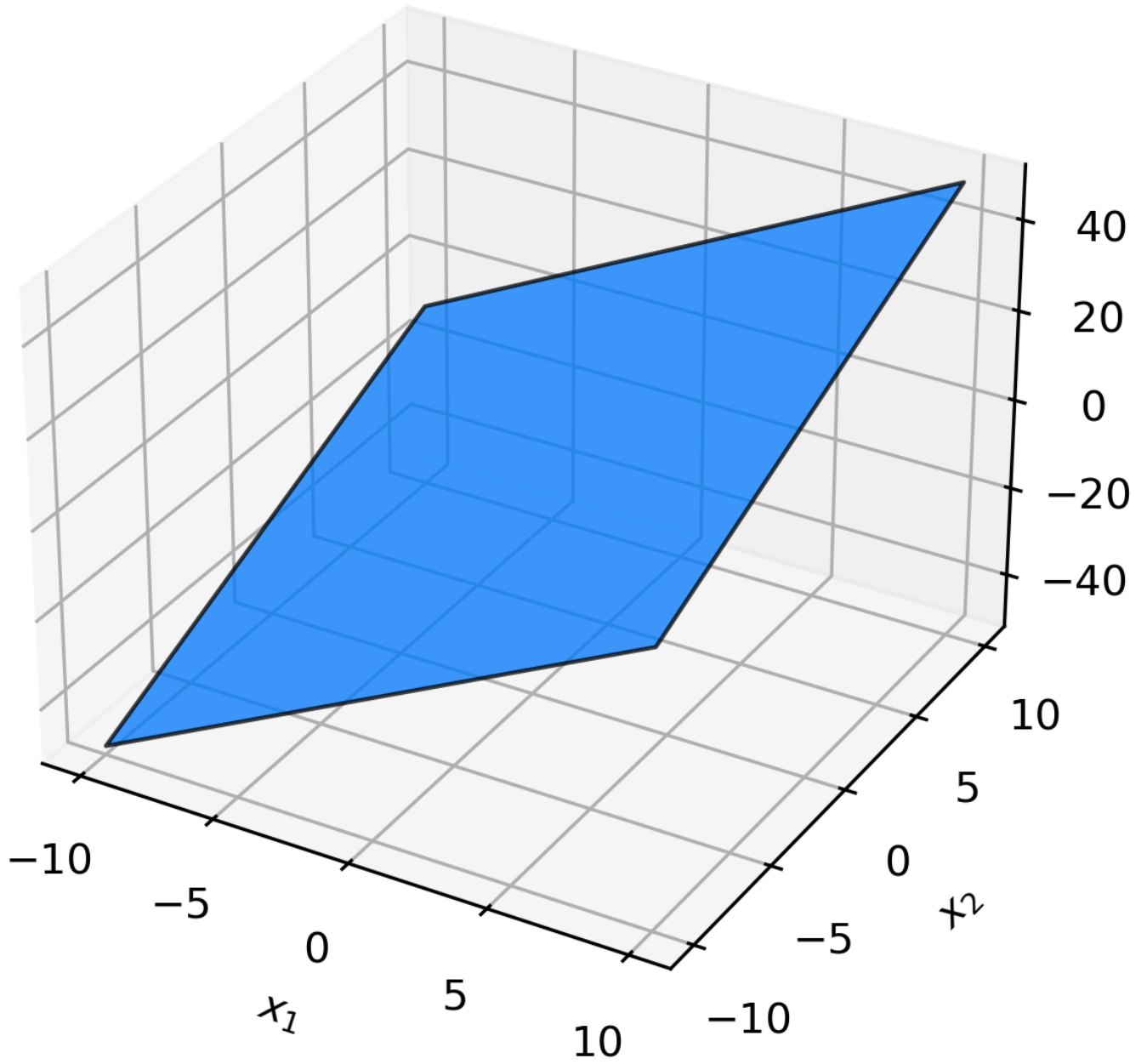
Looking at each plot, the first one seems to reach a minimum near the point $(w_0, w_1) = (0, 5)$, the second one near $(w_0, w_1) = (0, -5)$, the third one near $(w_0, w_1) = (0, 10)$, and the fourth one near $(w_0, w_1) = (-40, -20)$, maybe (it's hard to tell exactly where it is, but we don't need to know exactly). Looking then at the data, it's clear that the "line of best fit" will have a negative slope, which means only the second and fourth options are possibly correct (they are the only two whose w_1 is negative). But the slope is not as large negative as -20, so only the second choice is reasonably the risk surface for this data.

Which of the plots below shows the surface of the linear prediction function $H(\vec{x}) = \text{Aug}(\vec{x}) \cdot \vec{w}$ with $\vec{w} = (3, -2, 1)^T$?

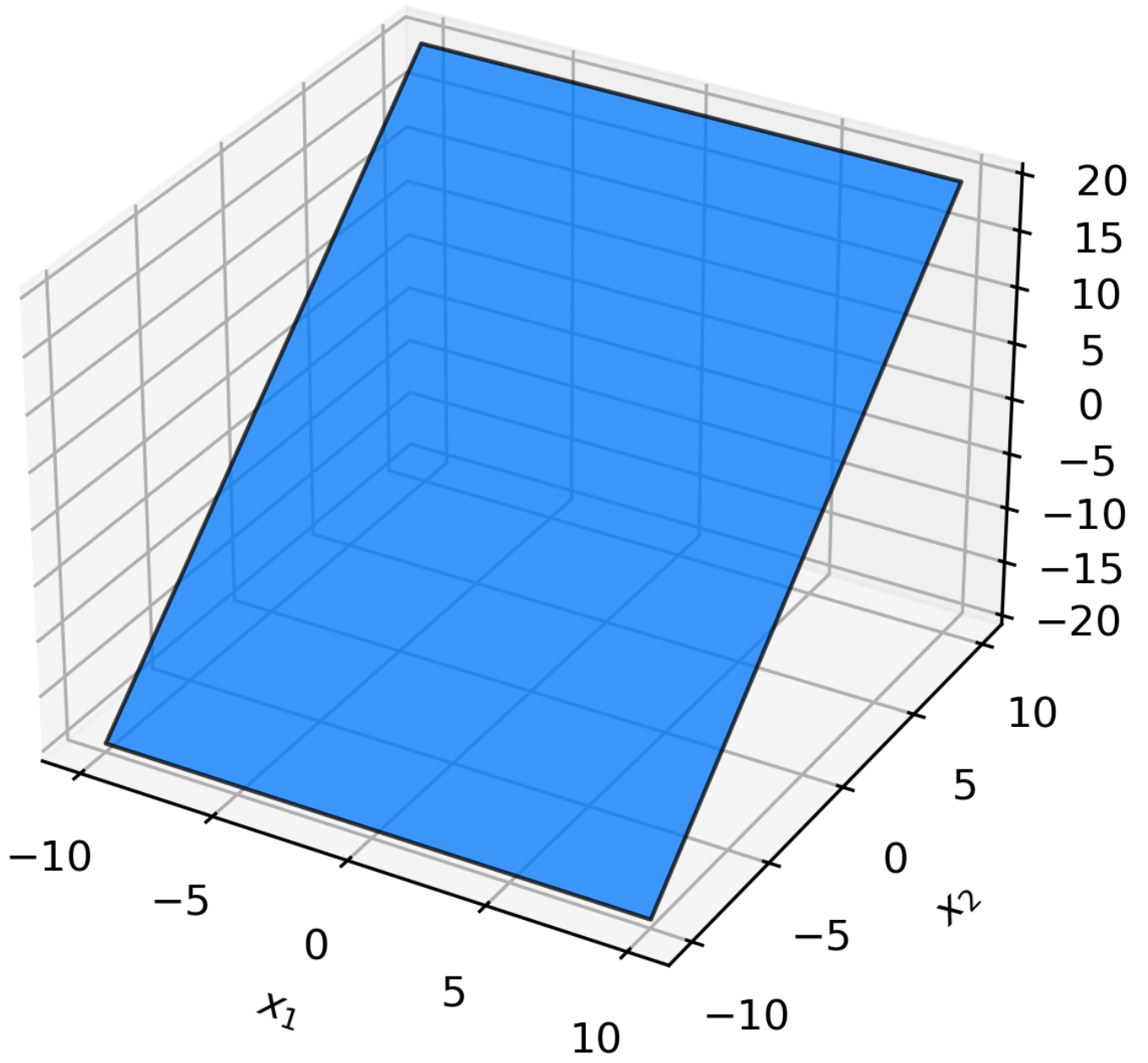
()



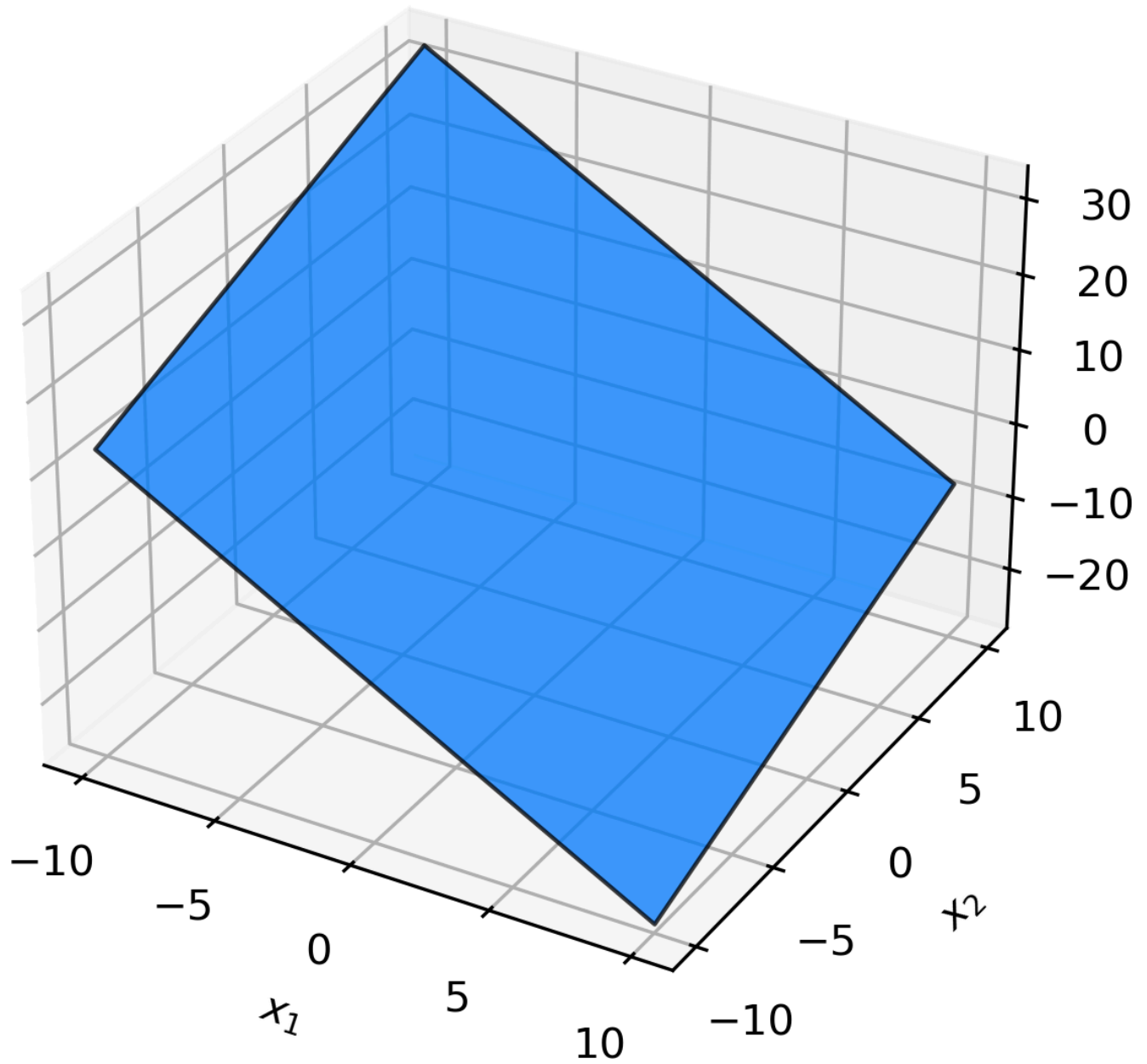
()



()

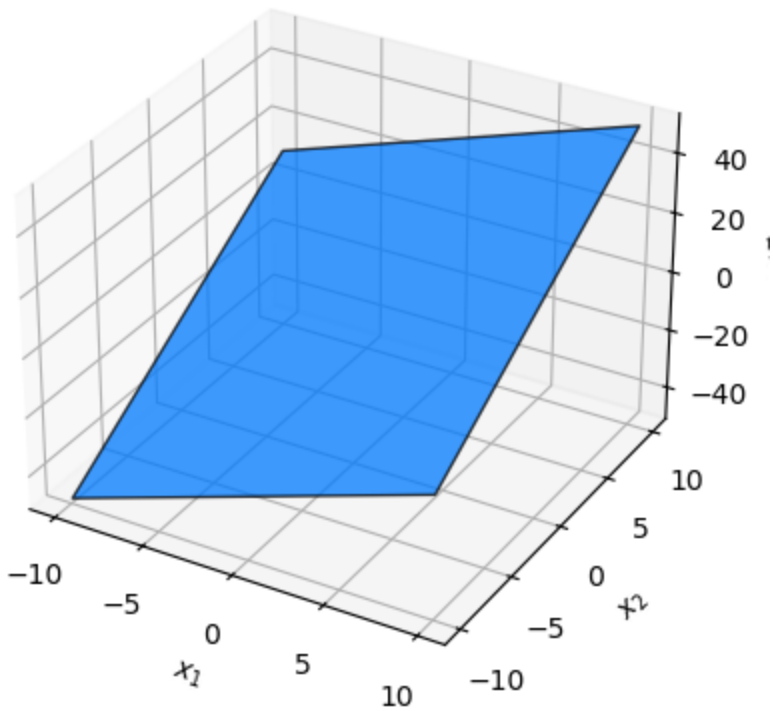


(x)



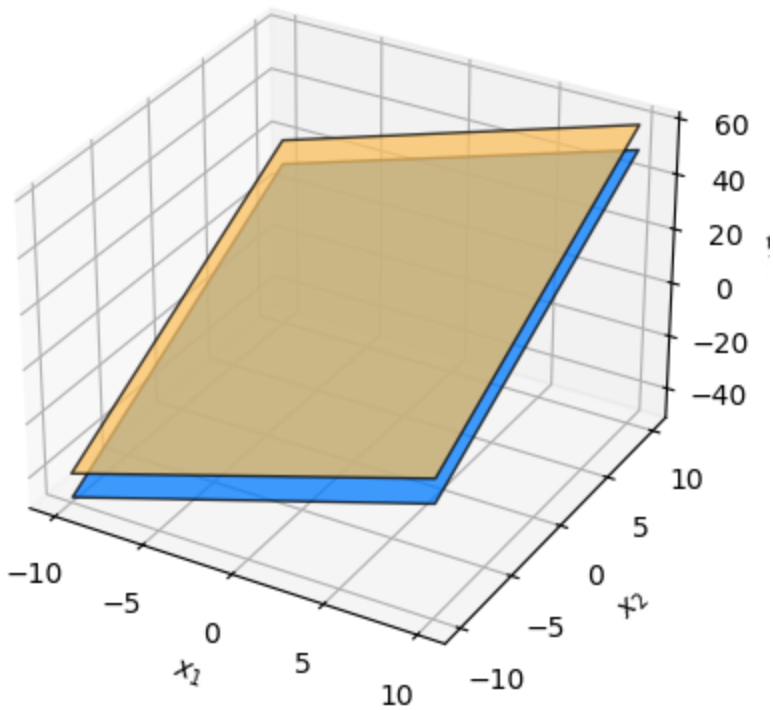
Remember from lecture that the weight w_i is the slope of the plane in the i th direction (and w_0 is the "bias"). Since $w_1 = -2$ is negative, we're looking for a plane that slopes downward as x_1 increases. And since $w_2 = 1$, the correct plane will slope upward as we increase x_2 . The only option that satisfies this is the last one.

The figure below shows the surface of the linear prediction function $H_1(\vec{x}) = \text{Aug}(\vec{x}) \cdot \vec{w}$ with $\vec{w} = (w_0, w_1, w_2)^T$.

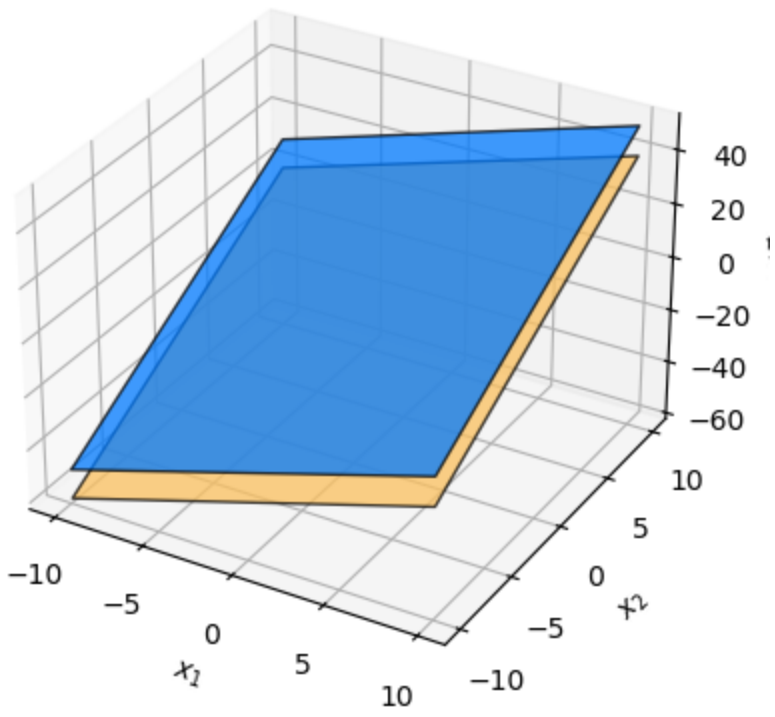


Now let $\omega = (w_0 + 10, w_1, w_2)^T$. Which of the below plots could show the surface of the prediction function $H_2(\vec{x}) = \text{Aug}(\vec{x}) \cdot \omega$ as a yellow plane? For convenience, the original surface is shown in blue.

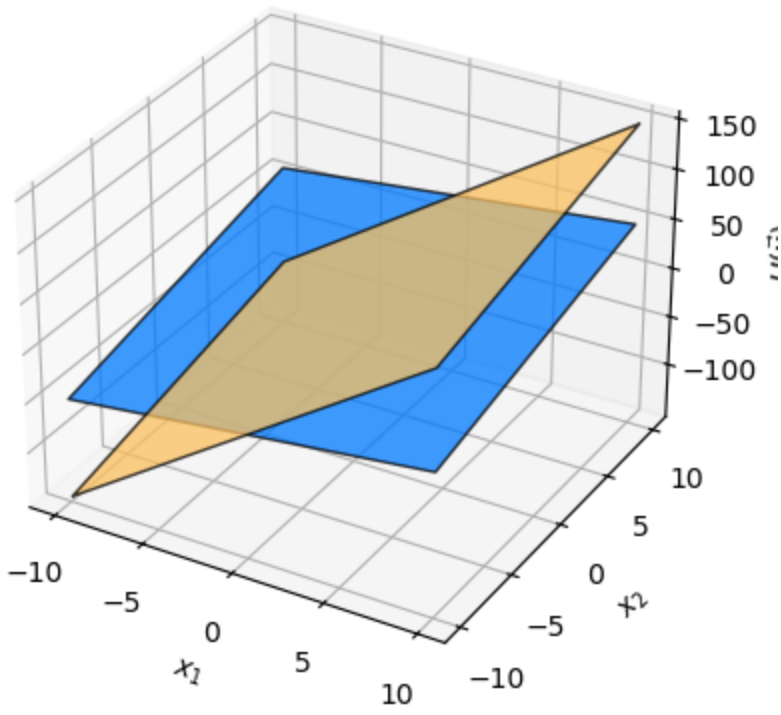
(x)



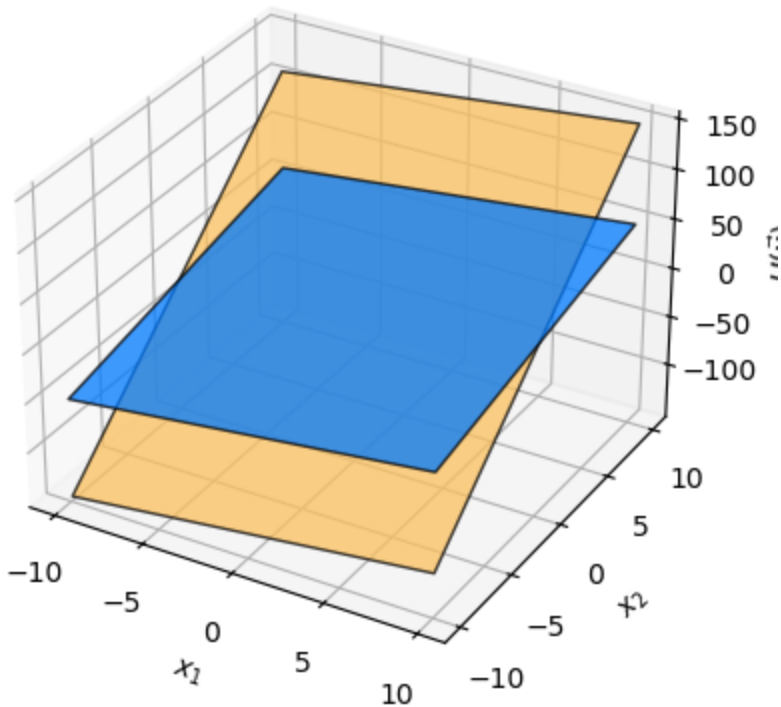
()



()



()



The first entry of the weight vector is the "bias" term, while the other terms are the slopes in the corresponding directions. So, adding 10 to the bias term will shift the plane up by 10 units, but not change the slopes. Therefore, the new plane is simply the old plane, "lifted" up by 10 units.

Consider a data set consisting of (x, y) pairs, where x is the size of a house in square feet, and y is the price of the house in USD.

Suppose a linear prediction function $H_1(\vec{x})$ is fit to the data by minimizing squared error and the resulting function is used to predict the price of a 900 square foot house. Call this predicted price y_1 .

Next, suppose a new data set is created by converting each of the house sizes to square yards (the prices are left unchanged). Another linear prediction function $H_2(x)$ is fit to this new data set by minimizing squared error, and it is used to predict the price of a 100 square yard house (100 square yards is equal to 900 square feet). Call this predicted price y_2 .

True or False: $y_1 = y_2$.

- True
- False

This situation is like rescaling a photo in an image editor, but only in one direction. Imagine a screenshot of a scatter plot of the original data, and then scaling it down, but only in the horizontal

direction (since converting to square yards is done by dividing the original x values by 9. The original line of best fit becomes steeper, but the residuals (the differences between the prediction and the correct answer) remain the exact same. Moreover, the same prediction is made for every house as in the original data set.

This was meant to be solved intuitively, but you might want a little more convincing. For that, we'll do the math on the homework and prove that the predictions are the same.

Consider the same data set consisting of (x, y) pairs, where x is the size of house in square feet, and y is the price of the house.

Suppose a straight line is fit to the data by minimizing squared error; suppose the slope of the resulting line is 600.

Suppose a pandemic causes the price of every house in the data set to exactly double. At the same time, each of the homeowners built an addition to their house, increasing its size by a factor of 1.5.

If a straight line is fit to this new data set, what will be its slope? State your answer as a decimal number.

We can think of this in two steps. First, the price of every house doubles, so the y values are all multiplied by 2. This means the line of best fit will have to be steeper to account for the increased prices. Since slope is "rise over run", doubling the "rise" means doubling the numerator of the slope. So the new slope is 1200.

Second, the size of every house increases by a factor of 1.5. This has the effect of increasing the x values, or the "run" of the slope. This increases the denominator of the slope by a factor of 1.5, meaning that our slope of 1200 becomes $1200/1.5 = 800$

Suppose a linear predictor is fit to a training data set by minimizing the expected square loss; let R_1 be the risk of this linear predictor (on the training set). Next, suppose a new feature is added to the data (but that everything else about the data remains the same). Suppose a linear predictor is fit to this new data set; let R_2 be the risk of this linear predictor (also on the training set).

True or False: it must be the case that $R_2 \leq R_1$.

- (x) True
() False

The empirical risk on the training data has to decrease (or stay the same) when we add a new feature. The informal explanation is that adding a new feature gives the predictor more "flexibility" to fit the data, and so the prediction errors can only go down.

More concretely, let's say we start with 1 feature. When we minimize the MSE, we're searching over all possible prediction functions $H(x_1) = w_0 + w_1x_1$ for the one that fits the data the closest and therefore minimizes the MSE. Let's call the best solution \vec{w}^* , and suppose (for concreteness) that $\vec{w}^* = (3, 5)$ and it achieves an empirical risk of 35.

When we add a new feature and minimize MSE, we're now searching through all functions of the form $H(x_1, x_2) = w_0 + w_1x_1 + w_2x_2$. But all of the functions of just one feature are *also* in this set and will be considered. For example, our best solution from before was $H(x_1) = 3 + 5x_1$, but this is the same as $H(x_1, x_2) = 3 + 5x_1 + 0x_2$, which will be considered during our search. And the empirical risk of $\vec{w} = (3, 5, 0)$ is exactly the same as before (35) since the predictions made using this parameter vector are exactly the same as before. So if with one feature our best result had an empirical risk of 35, then the best solution with two features can't be larger than this (and will most likely be smaller).

This problem is very careful to say that we're dealing with the empirical risk on the *training* data. It is a mathematical fact that the training error will go down when you add a new feature. However, the *test* error may very well go *up* when you add a new feature (this is indicative of overfitting.)