OMG, the next exercises are so hard! 😮

1) a) Using the same data from the example above, use a k nearest neighbors classifier to classify vector $\mathbf{x} = \begin{bmatrix} 100 & 210 \end{bmatrix}^T$ with $k = 1$, $k = 3$ and $k = 5$.

Solution:

The first step is to measure the $l1$ distance from the input to all labeled data vectors.

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
Class	0	1	1	0	0	1	0	0	0	0
$\|\mathbf{x} - \mathbf{x}^{(i)}\|_2$	86.0	22.4	14.1	64.0	78.1	36.1	41.2	30.0	33.5	20.0

For $k = 1$, the classifier outputs the label of the nearest neighbor. In this case, the closest example is $\mathbf{x}^{(3)}$ at a distance of 14.14, so the output class will be 1.

For $k = 3$, we need the three nearest neighbors. For this input the neighbors will be $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$ and $\mathbf{x}^{(2)}$ from nearest to farthest. Out of this group, a majority of two examples have class 1, so this would be the output.

Finally, for $k = 5$, the set of neighbors is $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(8)}$ and $\mathbf{x}^{(9)}$. In this case, we have a majority of three examples with label 0.

Literally just have to look at the distance values, pick the k-nearest and see which of the classes is the most present within those distances.

Let's see k=5, for example.

- 3 x 0
- 2 x 1

} 0 is the output class

b) Redo the exercise computing the distance with the $l1$ norm $\|\mathbf{x}\|_1 = \sum_i |x_i|$.

Same scheiße, weird distance formula.

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
Class	0	1	1	0	0	1	0	0	0	0
$\|\mathbf{x} - \mathbf{x}^{(i)}\|_1$	120	30	20	90	110	50	50	30	42	20

$\|x - x^{(1)}\| = |100 - 170| + |210 - 160| = 120$

...

For $k = 1$, the classifier outputs the label of the nearest neighbor. In this case, we have a tie for the closest example between $\mathbf{x}^{(3)}$ and $\mathbf{x}^{(10)}$ at a distance of 20. In practice one can choose randomly and take the class from that example.

For $k = 3$, we need the three nearest neighbors. For this input the neighbors will be $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$ and $\mathbf{x}^{(2)}$ from nearest to farthest. Out of this group, a majority of two examples have class 1, so this would be the output. It was also possible to choose $\mathbf{x}^{(8)}$ instead of $\mathbf{x}^{(2)}$ and in that case the label would be 0.

Finally, for $k = 5$, the set of neighbors is $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(8)}$ and $\mathbf{x}^{(9)}$. In this case, we have a majority of three examples with label 0.

Let's see k=3, for example.

Oh, no! A tie! What now? Perform a coin flip and choose either of them. It literally doesn't matter. Even if the answer would change, any of the choices is valid.

c) Again, repeate the exercise computing the distance with the infinity norm $\|\mathbf{x}\|_\infty = max_i |x_i|$.

Again, same scheiße.

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
Class	0	1	1	0	0	1	0	0	0	0
$\|\mathbf{x} - \mathbf{x}^{(i)}\|_\infty$	70	20	10	50	60	30	40	30	32	20

$\|x - x^{(1)}\| = \max(|100 - 170|, |210 - 160|) = 70$

...

For $k = 1$, the classifier outputs the label of the nearest neighbor. In this case, the closest example is $\mathbf{x}^{(3)}$ at a distance of 10, so the output class will be 1.

For $k = 3$, we need the three nearest neighbors. For this input the neighbors will be $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$ and $\mathbf{x}^{(2)}$ from nearest to farthest. Out of this group, a majority of two examples have class 1, so this would be the output.

Finally, for $k = 5$, the set of neighbors is $\mathbf{x}^{(3)}$, $\mathbf{x}^{(10)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(6)}$ and $\mathbf{x}^{(8)}$. In this case, we have a majority of three examples with label 1.

2) Assuming that 1 means *True* and 0 means *False*, consider the following features and class:

- X_1: "Fast processing"
- X_2: "Decent Battery"
- X_3: "Good Camera"
- X_4: "Good Look and Feel"
- X_5: "Easiness of Use"
- *Class*: "iPhone"

You are given the following training set:

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$
X_1	1	1	0	0	1	0	0
X_2	1	1	1	0	0	0	0
X_3	0	1	1	0	1	1	0
X_4	1	0	1	1	1	0	0
X_5	0	0	0	1	1	0	1
Class	1	0	0	0	1	1	1

Use a k nearest neighbors classifier based on the Hamming distance to classify vector $\begin{bmatrix}1&1&1&1&1\end{bmatrix}^T$ with $k = 1$ and $k = 3$.

Same scheiße again, with Hamming distance now.

The Hamming distance between two vectors is the number of positions at which the vectors are different.

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$
X_1	1	1	0	0	1	0	0
X_2	1	1	1	0	0	0	0
X_3	0	1	1	0	1	1	0
X_4	1	0	1	1	1	0	0
X_5	0	0	0	1	1	0	1
Class	1	0	0	0	1	1	1
$Hamming\left(\mathbf{x}, \mathbf{x}^{(i)}\right)$	2	3	3	4	1	4	4

For $k = 1$, the classifier outputs the label of the nearest neighbor. In this case, the closest example is $\mathbf{x}^{(5)}$ at a distance of 1, so the output class will be 1.

For $k = 3$, we need the three nearest neighbors. For this input these neighbors will be $\mathbf{x}^{(5)}$, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ (it could also be $\mathbf{x}^{(3)}$) from nearest to farthest. Out of this group, a majority of two examples have class 1, so this would be the output.

3) Consider the following data where a few preprocessed restaurant reviews (without stopwords) are classified as positive (1) or negative (0).

Sentence	*Class*
{"*Great*","*place*","*go*","*with*","*friends*"}	1
{"*Food*","*amazing*"}	1
{"*What*","*terrible*","*experience*","*no*","*words*"}	0
{"*Waiting*","*time*","*too*","*long*"}	0
{"*Terrible*","*place*","*for*","*family*","*dinner*"}	0

Consider as well the Jaccard similarity between two sets:

$$Jaccard_{sim}\left(A, B\right) = \frac{|A \cap B|}{|A \cup B|}$$

Using the similarity measure, compute the k nearest neighbor output for input {"*Terrible*","*food*","*overall*","*lousy*","*dinner*"} using $k = 1$ and $k = 3$.

Now, with sets. Not rocket science either.

	Sentence	*Class*	*Jaccard*
$x^{(1)}$	{"*Great*","*place*","*go*","*with*","*friends*"}	1	$\frac{0}{10} = 0$
$x^{(2)}$	{"*Food*","*amazing*"}	1	$\frac{1}{6}$
$x^{(3)}$	{"*What*","*terrible*","*experience*","*no*","*words*"}	0	$\frac{1}{9}$
$x^{(4)}$	{"*Waiting*","*time*","*too*","*long*"}	0	$\frac{0}{9} = 0$
$x^{(5)}$	{"*Terrible*","*place*","*for*","*family*","*dinner*"}	0	$\frac{2}{8} = \frac{1}{4}$

For $k = 1$, we need to find the closest neighbor. Working with a similarity measure instead of a distance measure, the closest example will be the one with highest similarity. In this case $x^{(5)}$ is most similar, so the classifier outputs its label of 0.

For $k = 3$, we need the three nearest neighbors. For this query set these neighbors will be $x^{(5)}$, $x^{(2)}$ and $x^{(3)}$. Out of this group, a majority of two examples have class 0, so this would be the output.

Careful, as we are given a similarity function, we need to consider the most similar cases first.

1) Consider the following training data:

$$\left\{\mathbf{x}^{(1)} = \begin{pmatrix}1\\1\end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix}2\\1\end{pmatrix}, \mathbf{x}^{(3)} = \begin{pmatrix}1\\3\end{pmatrix}, \mathbf{x}^{(4)} = \begin{pmatrix}3\\3\end{pmatrix}\right\}$$

$$\left\{t^{(1)} = 1.4, t^{(2)} = 0.5, t^{(3)} = 2, t^{(4)} = 2.5\right\}$$

a) Find the closed form solution for a linear regression that minimizes the sum of squared errors on the training data..

Now, stuff starts to get interesting.

First, add the bias to the input matrix and build the target vector:

$$X = \begin{pmatrix}1&1&1\\1&2&1\\1&1&3\\1&3&3\end{pmatrix} \qquad Y = \begin{pmatrix}1.4\\0.5\\2.0\\2.5\end{pmatrix}$$

Then, considering the formula, calculate each step:

$$\mathbf{w} = (X^T X)^{-1} X^T Y$$

$$X^T = \begin{pmatrix}1&1&1&1\\1&2&1&3\\1&1&3&3\end{pmatrix} \qquad X^T X = \begin{pmatrix}4&7&8\\7&15&15\\8&15&20\end{pmatrix} \qquad (X^T X)^{-1} = \begin{pmatrix}1.875&-0.5&-0.375\\-0.5&0.4&-0.1\\-0.375&-0.1&0.275\end{pmatrix}$$

$$(X^T X)^{-1} X^T = \begin{pmatrix}1.0&0.5&0.25&-0.75\\-0.2&0.2&-0.4&0.4\\-0.2&-0.3&0.35&0.15\end{pmatrix} \qquad (X^T X)^{-1} X^T y = \begin{pmatrix}0.275\\0.02\\0.645\end{pmatrix}$$

b) Predict the target value for $x_{query} = (2\ 3)^T$.

Solution:
From the previous question, we have our weights:

$$\mathbf{w} = \begin{pmatrix}0.275\\0.02\\0.645\end{pmatrix}$$

So, to compute the predicted value, we just need to augment the query vector with a bias dimension and apply the linear regression:

$$output\,(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \begin{pmatrix}0.275\\0.02\\0.645\end{pmatrix} \cdot \begin{pmatrix}1\\2\\3\end{pmatrix} = 2.25$$

Just have to add the bias and compute the result.

c) Sketch the predicted hyperplane along which the linear regression predicts points will fall.

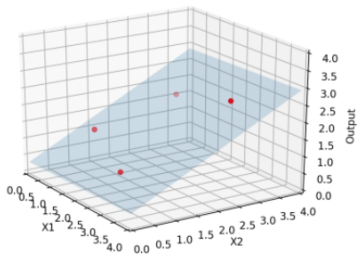
Useful for visualization.

Solution:

We can get the hyperplane's equation by taking the linear regression output for a general input $\begin{pmatrix} 1 & x_1 & x_2 \end{pmatrix}^T$ and equating it to zero:

$$\begin{aligned} output\,(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} &= 0 \\ = \begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} &= 0 \\ = 0.02x_1 + 0.645x_1 + 0.275 &= 0 \end{aligned}$$

From the equation, we get the following plot:



---

d) Compute the mean squared error produced by the the linear regression.

Nothing in particular, just use the formula.

Solution:

For each point in the training data, we must compute the linear regression prediction and then compute its squared error:

$$\left(t^{(1)} - output\left(\mathbf{x}^{(1)}\right)\right)^2 = \left(t^{(1)} - \mathbf{w} \cdot \mathbf{x}^{(1)}\right)^2 = \left(1.4 - \begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\right)^2 = (1.4 - 0.94)^2 = 0.2116$$

$$\left(t^{(2)} - output\left(\mathbf{x}^{(2)}\right)\right)^2 = \left(t^{(2)} - \mathbf{w} \cdot \mathbf{x}^{(2)}\right)^2 = \left(0.5 - \begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}\right)^2 = (0.5 - 0.96)^2 = 0.2116$$

$$\left(t^{(3)} - output\left(\mathbf{x}^{(3)}\right)\right)^2 = \left(t^{(3)} - \mathbf{w} \cdot \mathbf{x}^{(3)}\right)^2 = \left(2.0 - \begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}\right)^2 = (2.0 - 2.23)^2 = 0.0529$$

$$\left(t^{(4)} - output\left(\mathbf{x}^{(4)}\right)\right)^2 = \left(t^{(4)} - \mathbf{w} \cdot \mathbf{x}^{(4)}\right)^2 = \left(2.5 - \begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 3 \\ 3 \end{pmatrix}\right)^2 = (2.5 - 2.27)^2 = 0.0529$$

So, the mean error is:

$$\frac{0.2116 + 0.2116 + 0.0529 + 0.0529}{4} = 0.13225$$

5) Consider the following training data:

$$\mathbf{x}^{(1)} = \begin{pmatrix} -0.95 \\ 0.62 \end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix} 0.63 \\ 0.31 \end{pmatrix}, \mathbf{x}^{(3)} = \begin{pmatrix} -0.12 \\ -0.21 \end{pmatrix}, \mathbf{x}^{(4)} = \begin{pmatrix} -0.24 \\ -0.5 \end{pmatrix},$$
$$\mathbf{x}^{(5)} = \begin{pmatrix} 0.07 \\ -0.42 \end{pmatrix}, \mathbf{x}^{(6)} = \begin{pmatrix} 0.03 \\ 0.91 \end{pmatrix}, \mathbf{x}^{(7)} = \begin{pmatrix} 0.05 \\ 0.09 \end{pmatrix}, \mathbf{x}^{(8)} = \begin{pmatrix} -0.83 \\ 0.22 \end{pmatrix}$$

$$\left\{ t^{(1)} = 0, t^{(2)} = 0, t^{(3)} = 1, t^{(4)} = 0, t^{(5)} = 1, t^{(6)} = 0, t^{(7)} = 1, t^{(8)} = 0 \right\}$$

a) Plot the data points and try to choose a non-linear transformation to apply.

Solution:

Plotting the data points we see that the labels seem to change with the distance from the origin. A way to capture is this is to perform a quadratic feature transform:

$$\phi(x_1, x_2) = (x_1^2, x_2^2)$$

Sometimes, they will ask us to perform a transformation to the apply to the labels.

$$\Phi = \begin{pmatrix} 1 & (-0.95)^2 & (0.62)^2 \\ 1 & (0.63)^2 & (0.31)^2 \\ 1 & (-0.12)^2 & (-0.21)^2 \\ 1 & (-0.24)^2 & (-0.5)^2 \\ 1 & (0.07)^2 & (-0.42)^2 \\ 1 & (0.03)^2 & (0.91)^2 \\ 1 & (0.05)^2 & (0.09)^2 \\ 1 & (-0.83)^2 & (0.22)^2 \end{pmatrix} = \begin{pmatrix} 1 & 0.9025 & 0.3844 \\ 1 & 0.3969 & 0.0961 \\ 1 & 0.0144 & 0.0441 \\ 1 & 0.0576 & 0.2500 \\ 1 & 0.0049 & 0.1764 \\ 1 & 0.0009 & 0.8281 \\ 1 & 0.0025 & 0.0081 \\ 1 & 0.6889 & 0.0484 \end{pmatrix}$$

In that case, we apply this transformation to all points (exclude the bias) and solve the exercise normally.