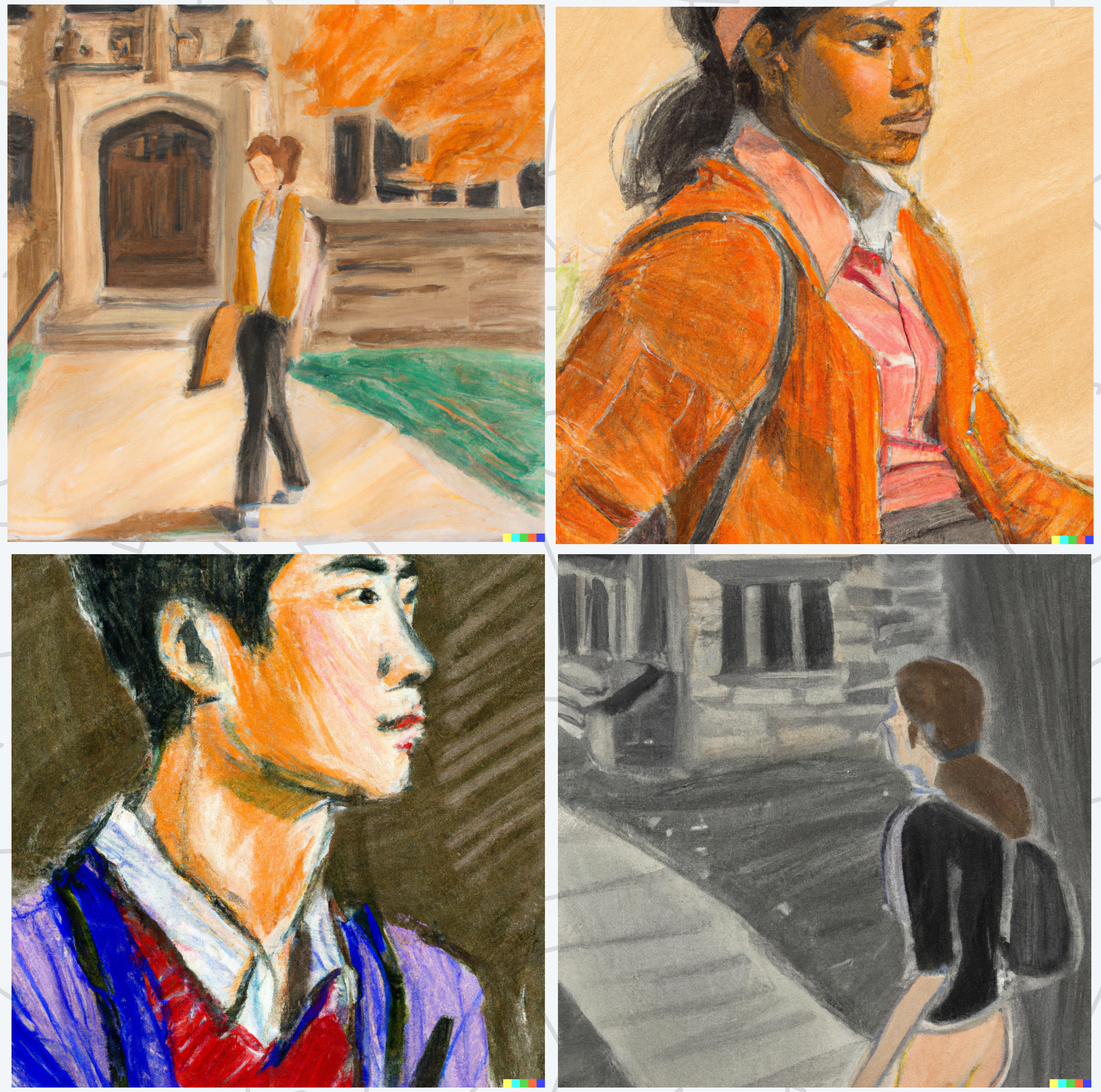
Please complete the mid-semester feedback survey ❤️

(details on Ed)



pastel drawing of a student at Princeton, DALL·E 2

Introduction to Machine Learning

- *what is machine learning?*
- *binary classifier*
- *the perceptron algorithm*
- *multi-class classifier*

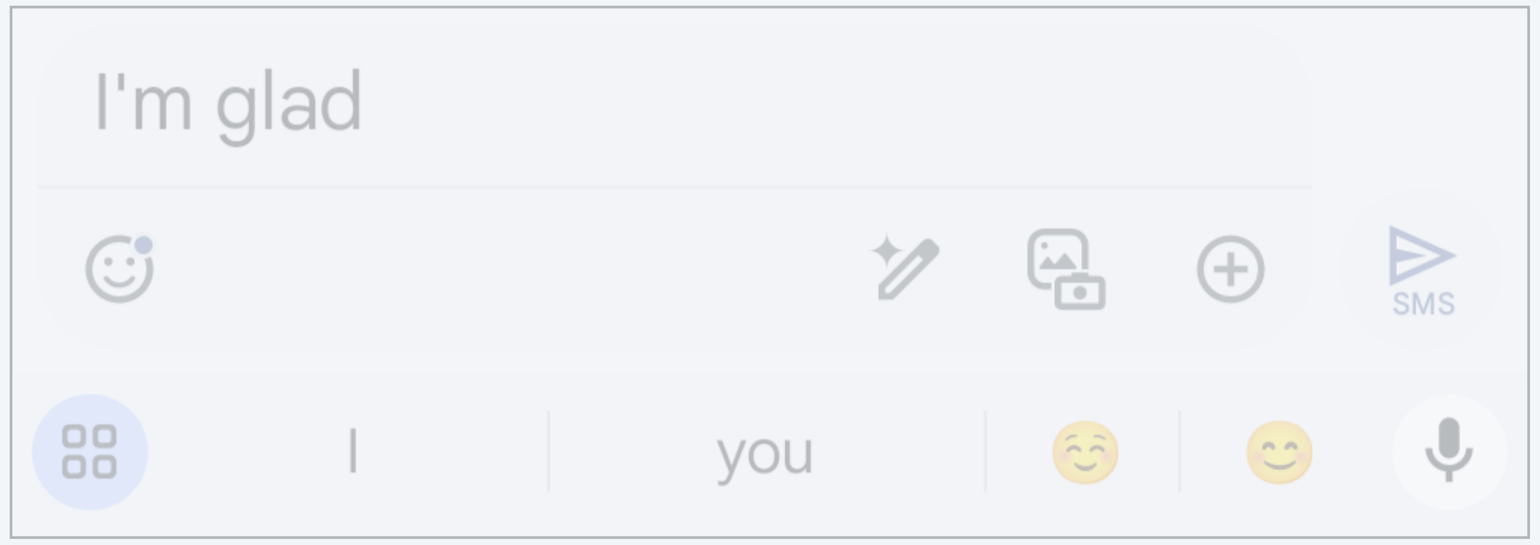

pastel drawing of a student at Princeton, DALL·E 2

Introduction to Machine Learning

- *what is machine learning?*
- *binary classifier*
- *the perceptron algorithm*
- *multi-class classifier*

Machine Learning Examples

Next word predictor



Crop detector

Length of stay predictor

Age	19
HR	80
RR	15
Temp	37

< 15 days in hospital

Image generator

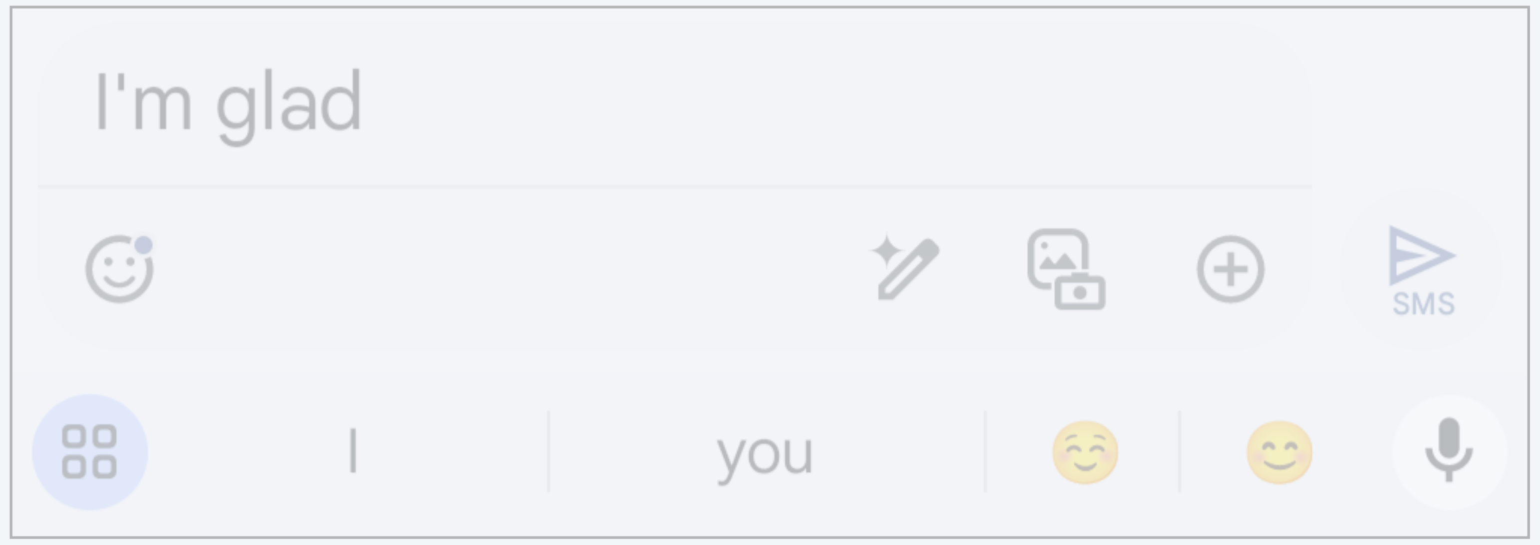
You

Tiger wearing a Princeton hat

Image Generator

Machine Learning Examples

Next word predictor



Crop detector

Length of stay classifier

Age	19
HR	80
RR	15
Temp	37

< 15 days in hospital

ChatGPT

You

Can you suggest me some fun activities in Princeton, NJ?

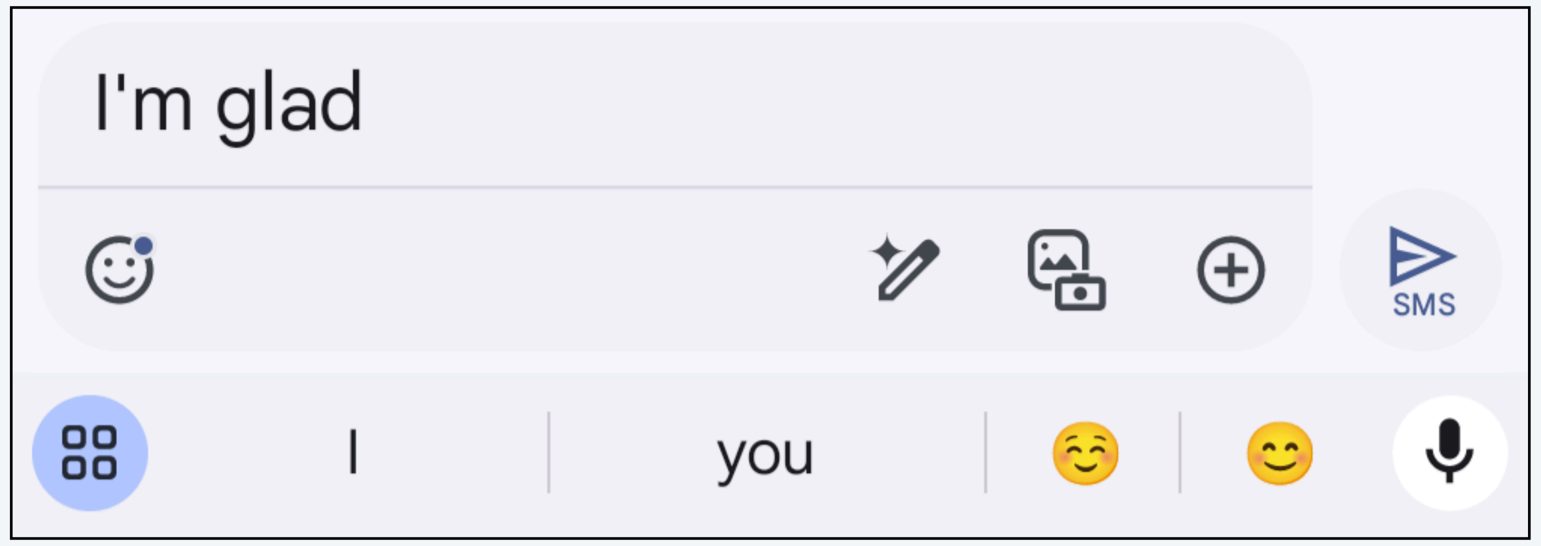
ChatGPT

Princeton, NJ, offers a blend of historic charm, cultural richness, and educational excellence, making it a great place to explore fun activities. Here are some suggestions: …

Machine Learning

A computer program **learns** if, for a defined task, it uses previous experience to improve a performance metric.

Next word predictor



Task	Experience	Performance Metric
Predict the next word in a **new** message	Old text messages	Fraction of words predicted correctly

Machine Learning

A computer program **learns** if, for a defined task, it uses previous experience to improve a performance metric.

Length of stay classifier

Age	19
HR	80
RR	15
Temp	37

$\Rightarrow$ < 15 days in hospital

Task	Experience	Performance Metric
Predict whether a **new** patient will have a long/short hospital stay	Data from previous patients (vital signs + length of stay)	Fraction of correct predictions

Machine Learning

A computer program **learns** if, for a defined task, it uses previous experience to improve a performance metric.

Crop detector

Task	Experience	Performance Metric
Detect crops in a **new** image	Previous images with bounding boxes	Fraction of crops identified

Machine Learning: quiz 1

We wish to build a computer program that learns to grasp objects using a robotic gripper.

What is an appropriate performance metric for this program?

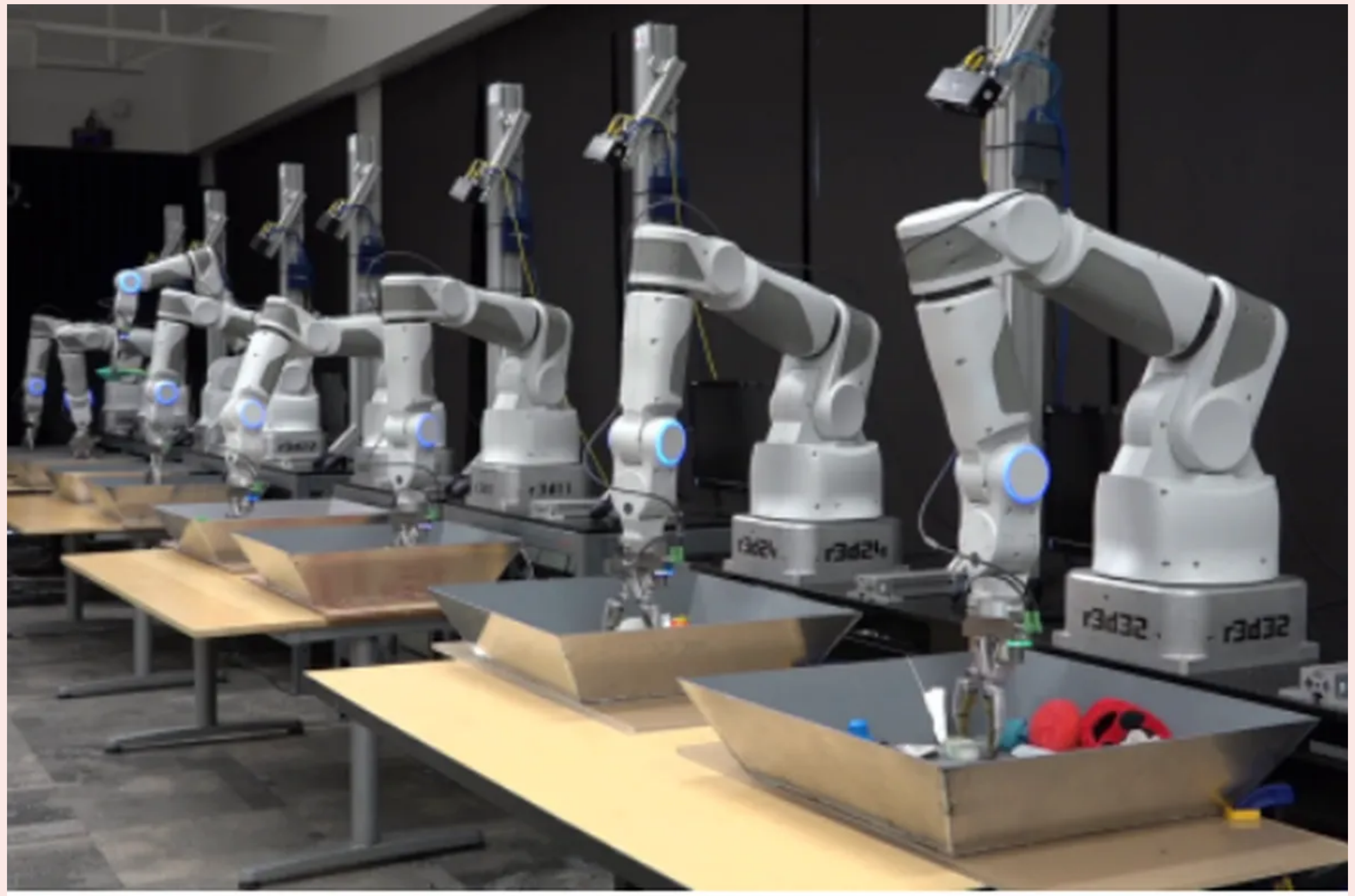
Task	Experience	Performance Metric
Grasp a **new** object from a box	800,000 random grasp attempts	?

A. Number of grasps until first success

B. Speed of successful grasps

C. Fraction of unsuccessful grasps

D. None of the above




pastel drawing of a student at Princeton, DALL·E 2

Introduction to Machine Learning

- *what is machine learning?*
- *binary classifier*
- *the perceptron algorithm*
- *multi-class classifier*

Binary Classifier

A binary classifier separates elements in a data set into one of two groups.

Task	Experience	Performance Metric
Classify **new** data into one of two groups	Previous data with group labels	Fraction of errors on **new** data

Length of stay classifier

Age	19
HR	80
RR	15
Temp	37

Long/short hospital stay

Code-switching detector

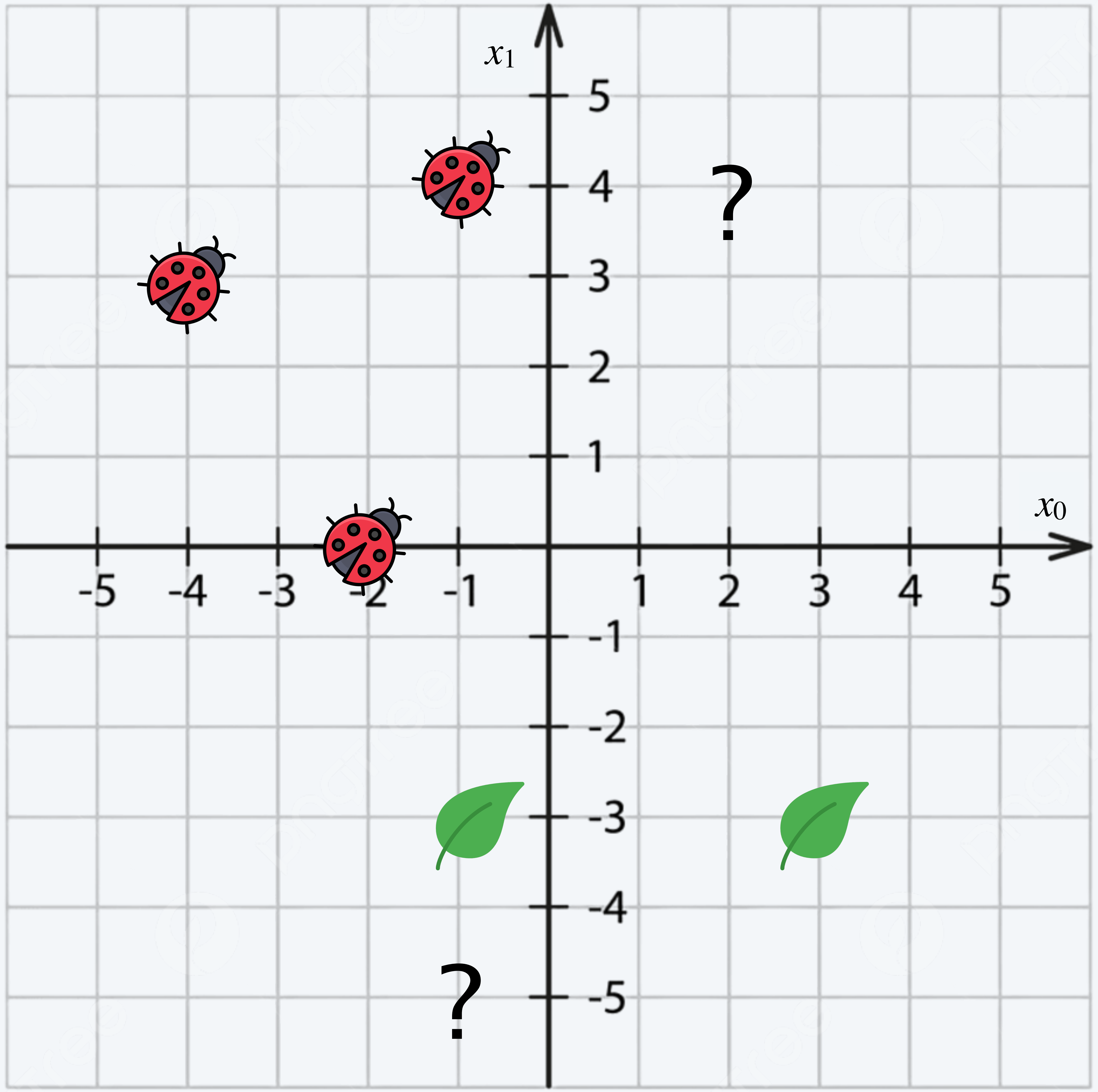
La party last night was increíble, everyone danced hasta el amanecer.

Monolingual/ bilingual

Deforestation classifier

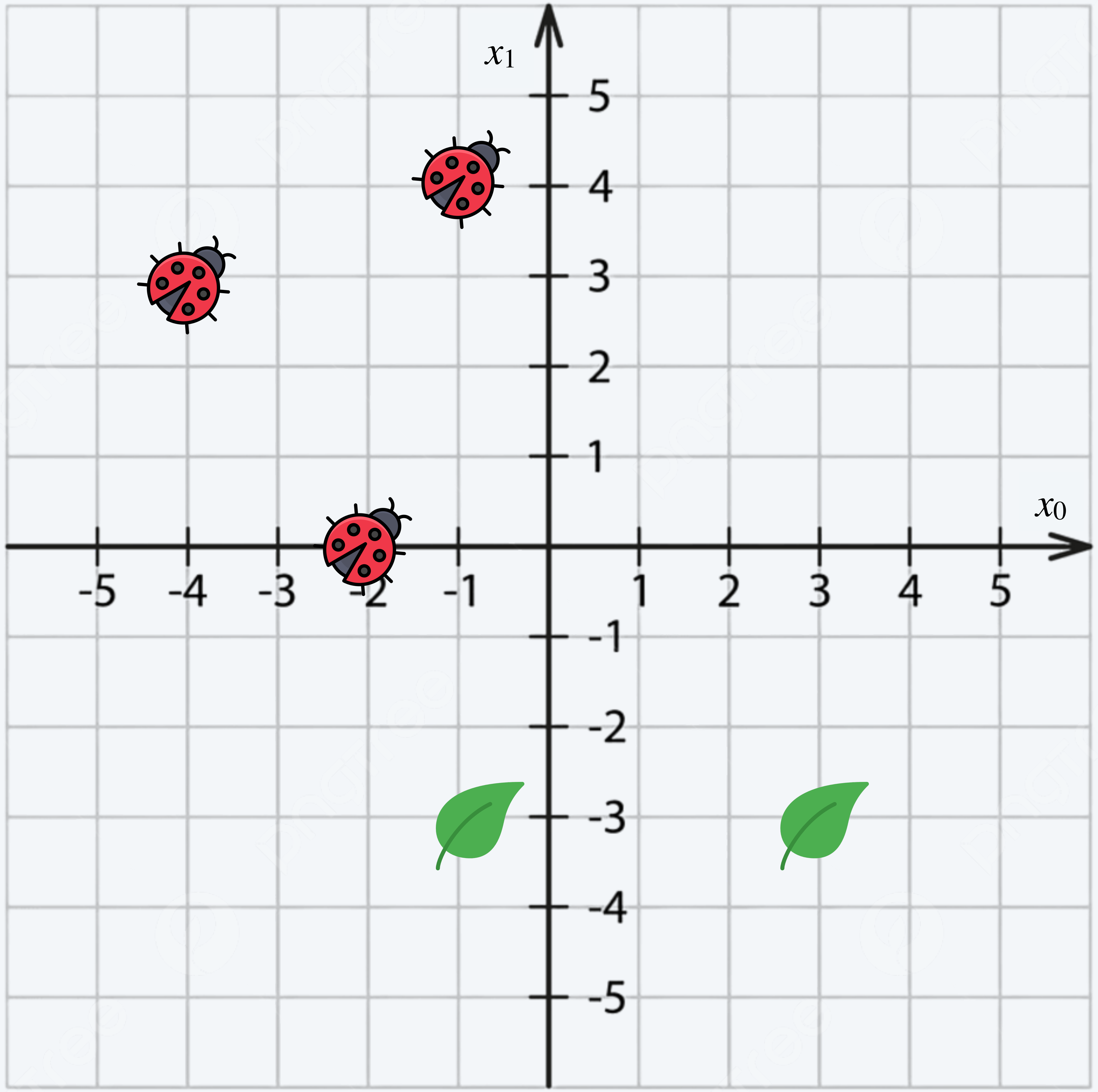
Deforestation/ no deforestation

A simple binary classification problem



We want to know whether we have a ladybug or a leaf based on our position in the cartesian plane.

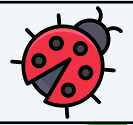
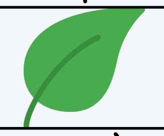
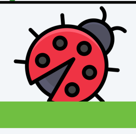
How to build a binary classifier



What do we need?

- Training data with binary labels to use as experience.
- Different test data with binary labels to evaluate our classifier.
- A learning algorithm that takes training data as input and returns a trained classifier.

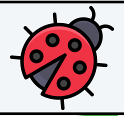
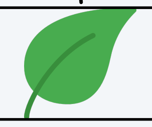
training data

x_0	x_1	binary label
−2	0	
−1	−3	
−1	4	

input/ features

output/ labels

test data

x_0	x_1	binary label
−4	3	
3	−3	

test points should not appear in the training data

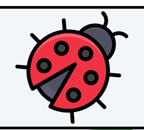
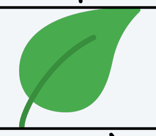
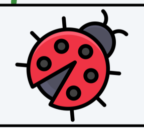
training data → learning algorithm → trained classifier

How to build a binary classifier

Steps.

1. **Training:** Use the learning algorithm and the training set (inputs and outputs) to obtain the binary classifier.

training data

x_0	x_1	**binary label**
−2	0	
−1	−3	
−1	4	

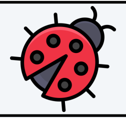
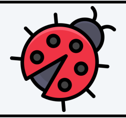
→ learning algorithm → binary classifier

2. **Testing:** Use the binary classifier and the test inputs to obtain predictions for each test data point.

test data (inputs)

x_0	x_1
−4	3
3	−3

→ binary classifier →

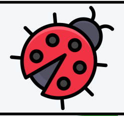
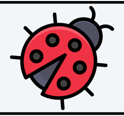
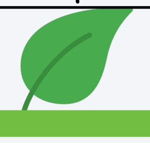
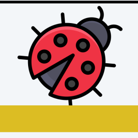
prediction



How to build a binary classifier

Steps.

3. **Evaluation:** Compute the performance metric with the test labels and the classifier predictions.

test data

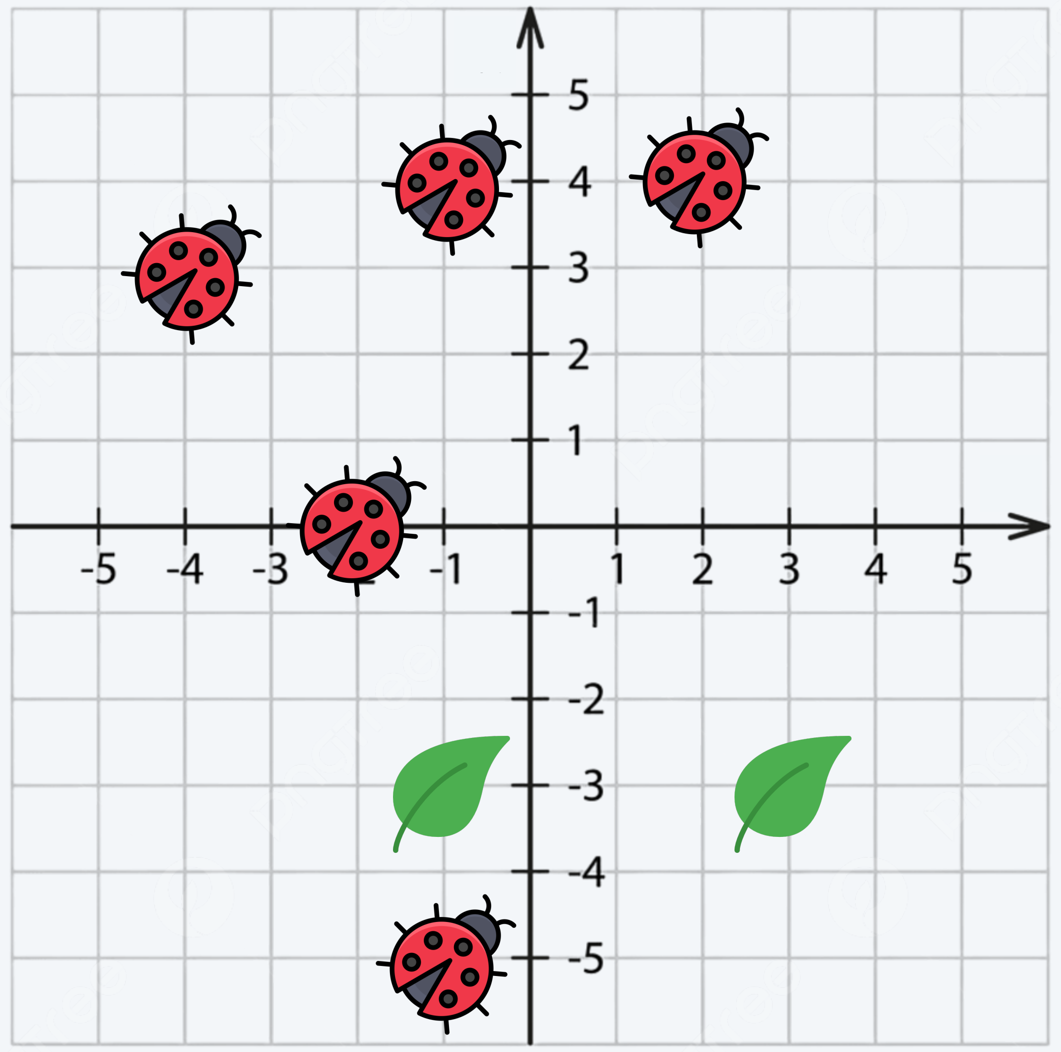
x_0	x_1	binary label	prediction
−4	3		
3	−3		

labels predictions

$$\text{error rate} = \frac{\text{number of errors}}{\text{total number of predictions}}$$

error: test output != prediction

4. **Deployment:** If we are satisfied with our error rate, we can use our classifier on unknown points.



Machine Learning: quiz 2

We have a binary classifier that always predicts the most frequent label in the training data.
What is the error rate of this classifier on the given test data?

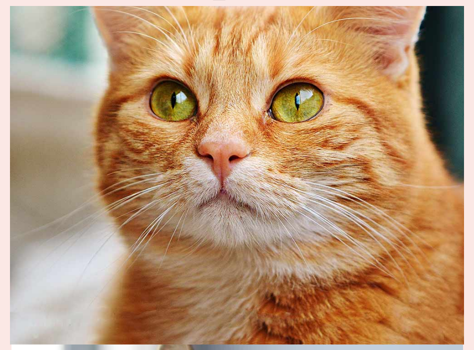
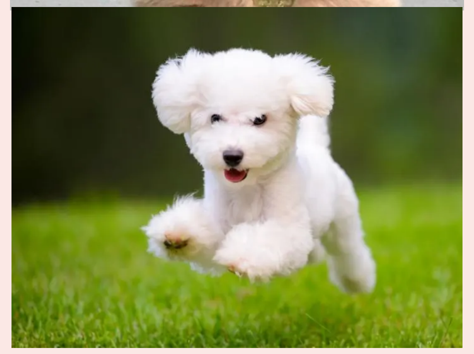
A. 30%

B. 40%

C. 60%

D. I don’t know

training data

input	**binary label**
	cat
	cat
	dog
	dog
	dog

test data

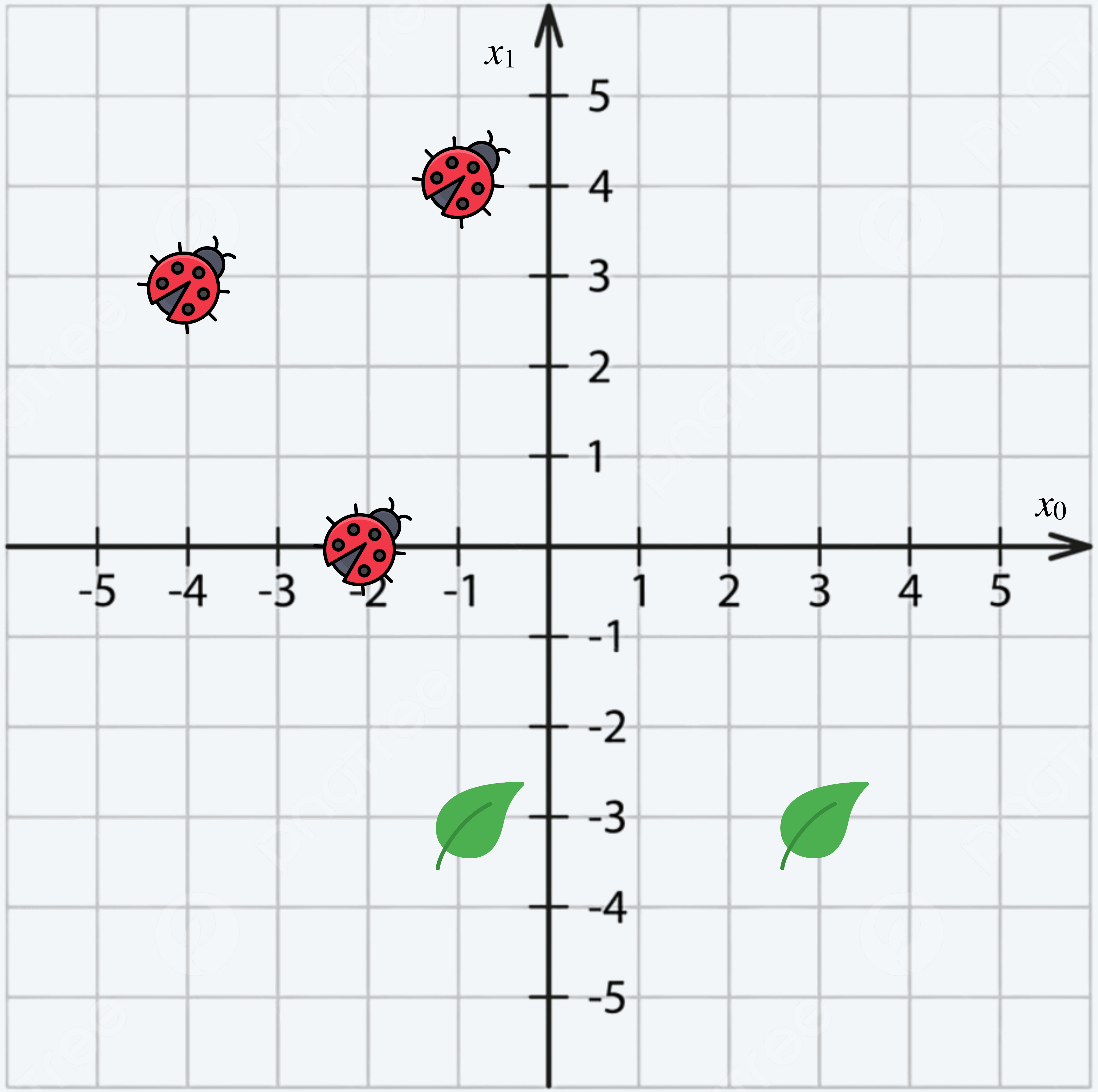
input	**binary label**
	cat
	cat
	cat
	dog
	dog

pastel drawing of a student at Princeton, DALL·E 2

MACHINE LEARNING

- *what is machine learning?*
- *binary classifier*
- *the perceptron algorithm*
- *multi-class classifier*

A simple binary classification problem



We want to know whether we have a ladybug or a leaf based on our position in the cartesian plane.

training data

x_0	x_1	**binary label**
−2	0	+1
−1	−3	−1
−1	4	+1

we will use +1 for the ladybugs and -1 for the leaves

test data

x_0	x_1	**binary label**
−4	3	+1
3	−3	−1

The perceptron algorithm

The perceptron is a learning algorithm that goes back to 1958.

It returns a vector of weights $\mathbf{w}$ which we use to make predictions.

To make a prediction, we take a weighted sum S of the features.

Then, we predict +1 if $S > 0$ and −1 otherwise.

training data

↓

perceptron algorithm

↓

trained classifier (learned weights $\mathbf{w}$)

training data (inputs)

x_0	x_1
−2	0
−1	−3
−1	4

$$\mathbf{w} = (w_0, w_1)$$

$$S = w_0 \cdot x_0 + w_1 \cdot x_1$$

Suppose $\mathbf{w} = (10, 20)$

$$S = 10 \cdot (-2) + 20 \cdot 0 = -20$$

Thus we predict −1, which corresponds to a 

The perceptron algorithm

The perceptron updates its weights whenever it makes a mistake on the training data.

Training data points will arrive sequentially.

We first make a prediction, and then use the label to update our weights.

t	$\mathbf{w}_t$
1	$(0, 0)$

Steps.

1. At time step $t = 1$, all weights in $\mathbf{w_t}$ start at 0.

The perceptron algorithm

The perceptron updates its weights whenever it makes a mistake on the training data.

Training data points will arrive sequentially.

We first make a prediction, and then use the label to update our weights.

t	$\mathbf{w}_t$	$\mathbf{x}$	binary label	weighted sum (S)	prediction
1	$(0, 0)$	$(-2, 0)$	$+1$	0	-1

Steps.

1. At time step $t = 1$, all weights in $\mathbf{w_t}$ start at 0.

2. For input $\mathbf{x}$ of a new data point, we predict +1 iff $S > 0$.

weighted sum of features

training data

x_0	x_1	binary label
−2	0	+1

The perceptron algorithm

The perceptron updates its weights whenever it makes a mistake on the training data.

Training data points will arrive sequentially.

We first make a prediction, and then use the label to update our weights.

t	w_t	x	binary label	weighted sum (S)	prediction	w_{t+1}
1	(0, 0)	(-2, 0)	+1	0	-1	(-2, 0)

we made an incorrect prediction on a positive point

Steps.

1. At time step $t = 1$, all weights in $\mathbf{w_t}$ start at 0.

2. For input $\mathbf{x}$ of a new data point, we predict +1 iff $S > 0$.

3. We update the weights using the following rules:
   A. If the prediction is correct, the weights don't change.
   B. If we made a mistake on a positive point, update $\mathbf{w_{t+1}} = \mathbf{w_t} + \mathbf{x}$
   C. If we made a mistake on a negative point, update $\mathbf{w_{t+1}} = \mathbf{w_t} - \mathbf{x}$

$$\mathbf{w_{t+1}} = (0,\ 0) \ + \ (-2,\ 0) \\ = (-2,\ 0)$$

The perceptron algorithm

The perceptron updates its weights whenever it makes a mistake on the training data.

Training data points will arrive sequentially.

We first make a prediction, and then use the label to update our weights.

t	w_t	x	binary label	weighted sum (S)	prediction	w_{t+1}
1	(0, 0)	(-2, 0)	+1	0	-1	(-2, 0)
2	(-2, 0)	(-1, -3)	-1	2	+1	(-1, 3)

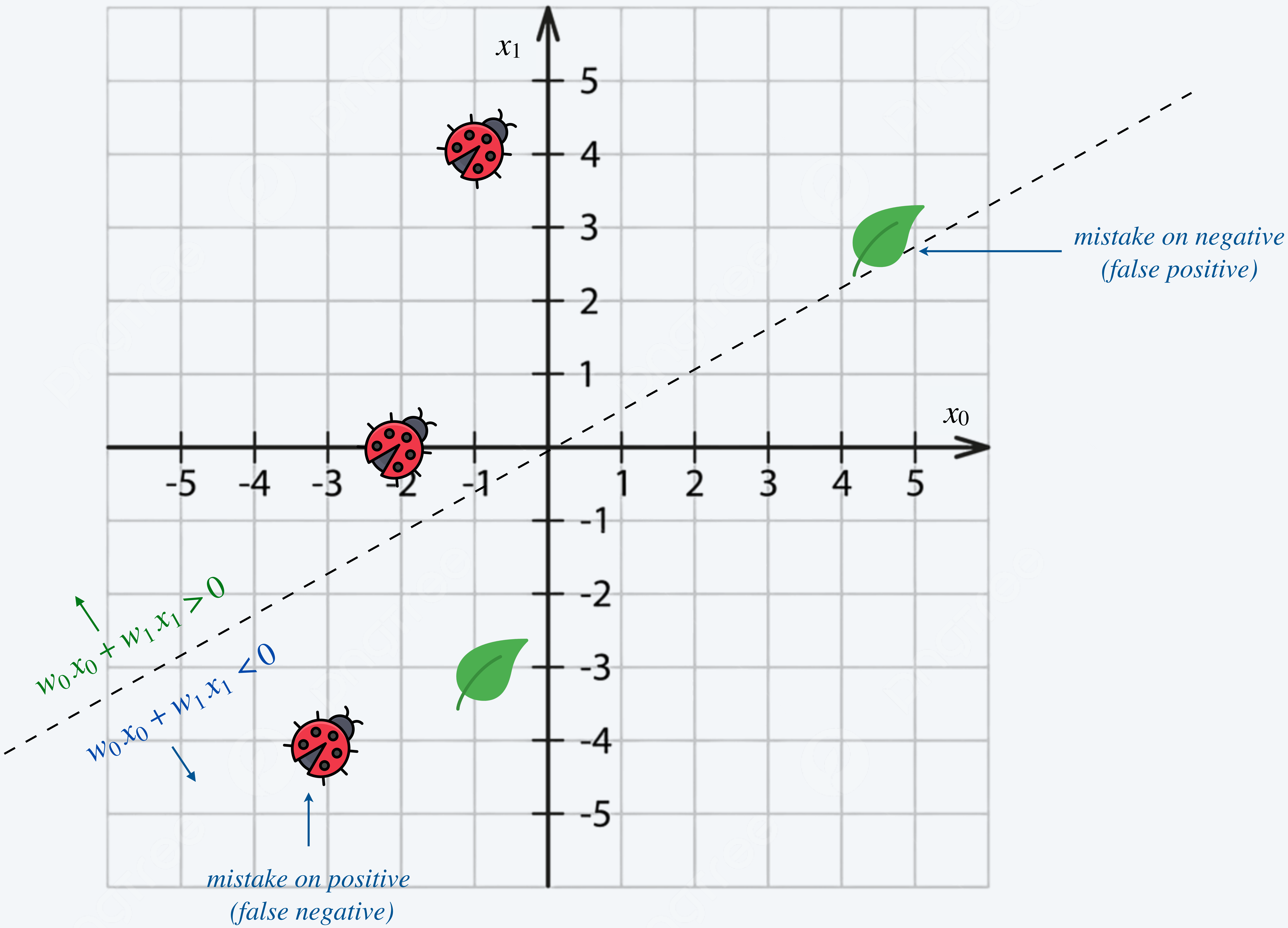
Steps.

1. At time step $t = 1$, all weights in $\mathbf{w_t}$ start at 0.
2. For input $\mathbf{x}$ of a new data point, we predict +1 iff $S > 0$.
3. We update the weights using the following rules:
   A. If the prediction is correct, the weights don't change.
   B. If we made a mistake on a positive point, update $\mathbf{w_{t+1}} = \mathbf{w_t} + \mathbf{x}$
   C. If we made a mistake on a negative point, update $\mathbf{w_{t+1}} = \mathbf{w_t} - \mathbf{x}$

$$S = (-2) \cdot (-1) - 0 \cdot (-3) \\ = 2$$

$$\mathbf{w_{t+1}} = (-2, 0) - (-1, -3) \\ = (-1, 3)$$

Geometric intuition



What are the weights w after training the perceptron algorithm on the following inputs?

A. (-2, 7)

B. (-1, 3)

C. (0, -1)

D. (-13, 39)

t	weights $\boldsymbol{w_t}$	input $\boldsymbol{x}$	binary label	weighted sum (S)	prediction	weights $\boldsymbol{w_{t+1}}$
1	(0, 0)	(-2, 0)	+1	0	-1	(-2, 0)
2	(-2, 0)	(-1, -3)	−1	2	+1	(-1, 3)
3	(-1, 3)	(−1, 4)	+1			

pastel drawing of a student at Princeton, DALL·E 2

Introduction to Machine Learning

- what is machine learning?
- binary classifier
- the perceptron algorithm
- multi-class classifier

Multi-class classifier

A multi-class classifier separates elements in a data set into one of multiple groups (more than two).

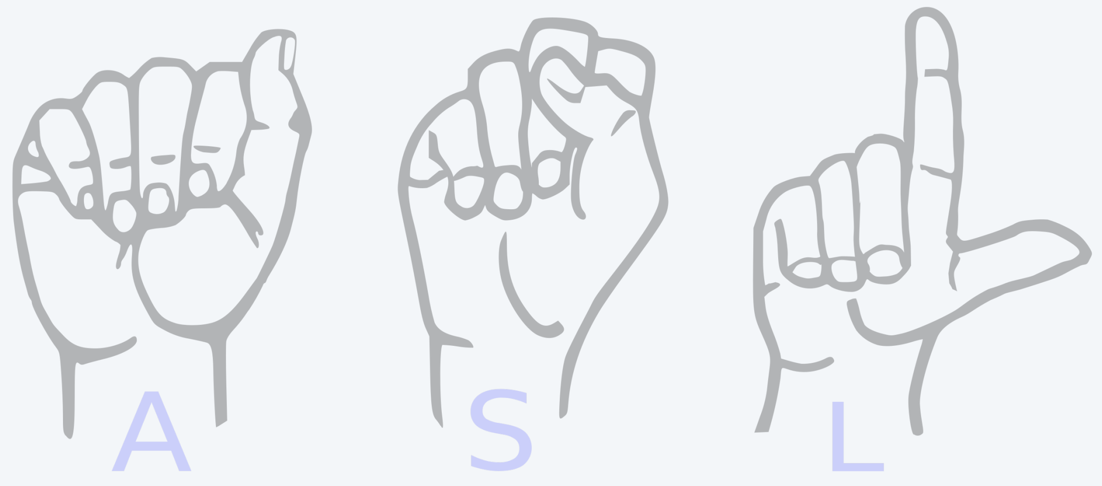
Task	Experience	Performance Metric
Classify **new** data into more than two groups	Previous data with group labels	Fraction of errors on **new** data

Length of stay classifier

Age	19
HR	80
RR	15
Temp	37

Long/short/no hospital stay

ASL Alphabet Recognizer

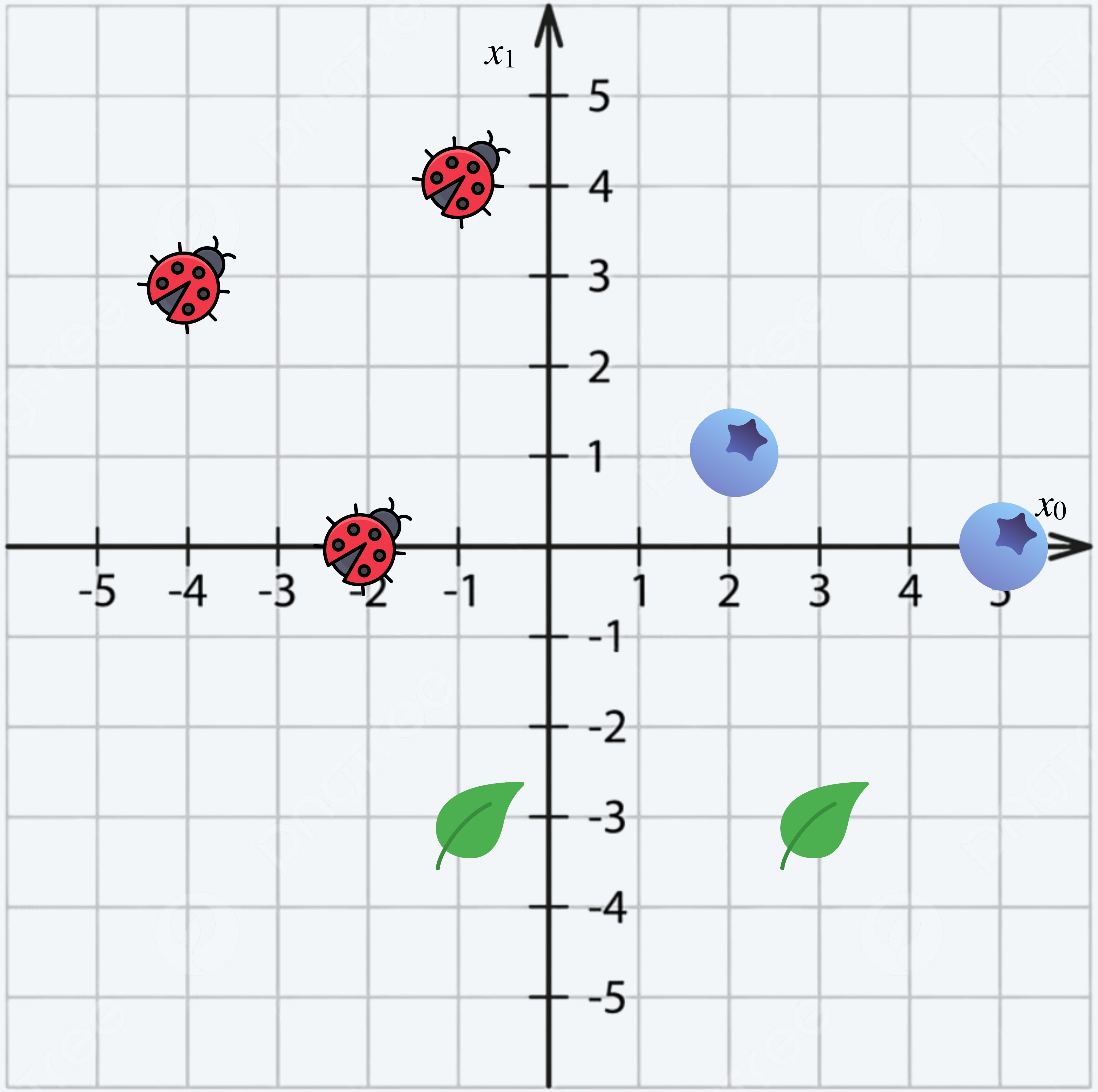


Sentiment Analysis

COS 126 is a very demanding yet rewarding course!

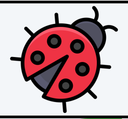
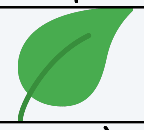
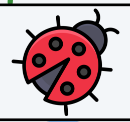
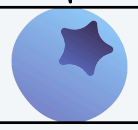
positive / negative /neutral

A simple multi-class classification problem

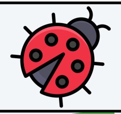
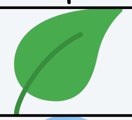


We want to know whether we have a ladybug, a leaf or a blueberry based on our position in the cartesian plane.

training data

x_0	x_1	class label
−2	0	
−1	−3	
−1	4	
5	0	

test data

x_0	x_1	class label
−4	3	
3	−3	
2	1	

How to build a multi-class classifier

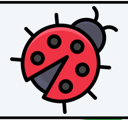
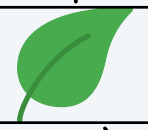
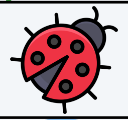
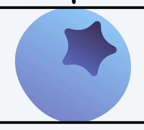
We can use most of the same machinery we used with a binary classifier.

However, we will use a multi-perceptron.

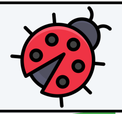
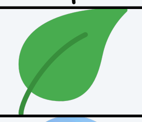
What we will use

- Training data with class labels.
- Test data with class labels.
- The perceptron algorithm.

training data

x_0	x_1	class label
−2	0	
−1	−3	
−1	4	
5	0	

test data

x_0	x_1	class label
−4	3	
3	−3	
2	1	

training data

↓

perceptron algorithm

↓

trained classifier
(learned weights $\mathbf{w}$)

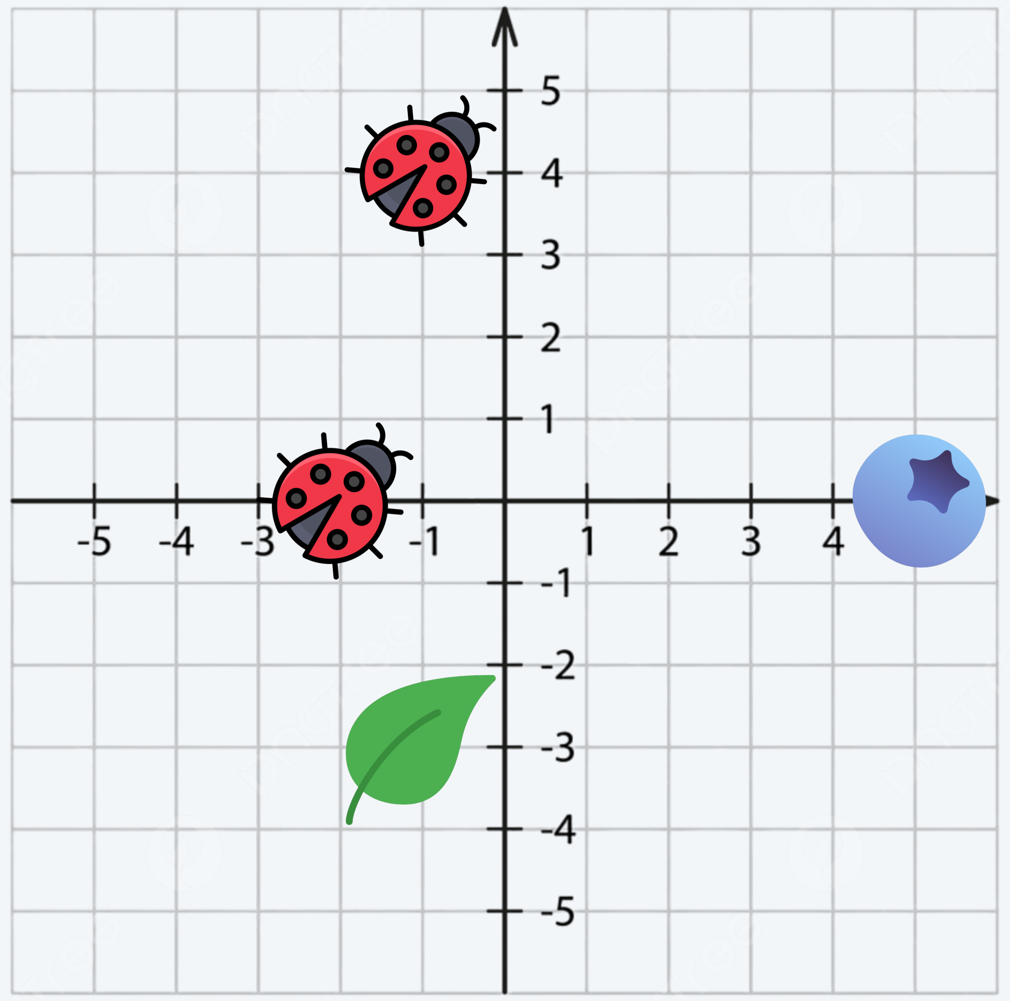
How to build a multi-class classifier

We can use most of the same machinery we used with a binary classifier.

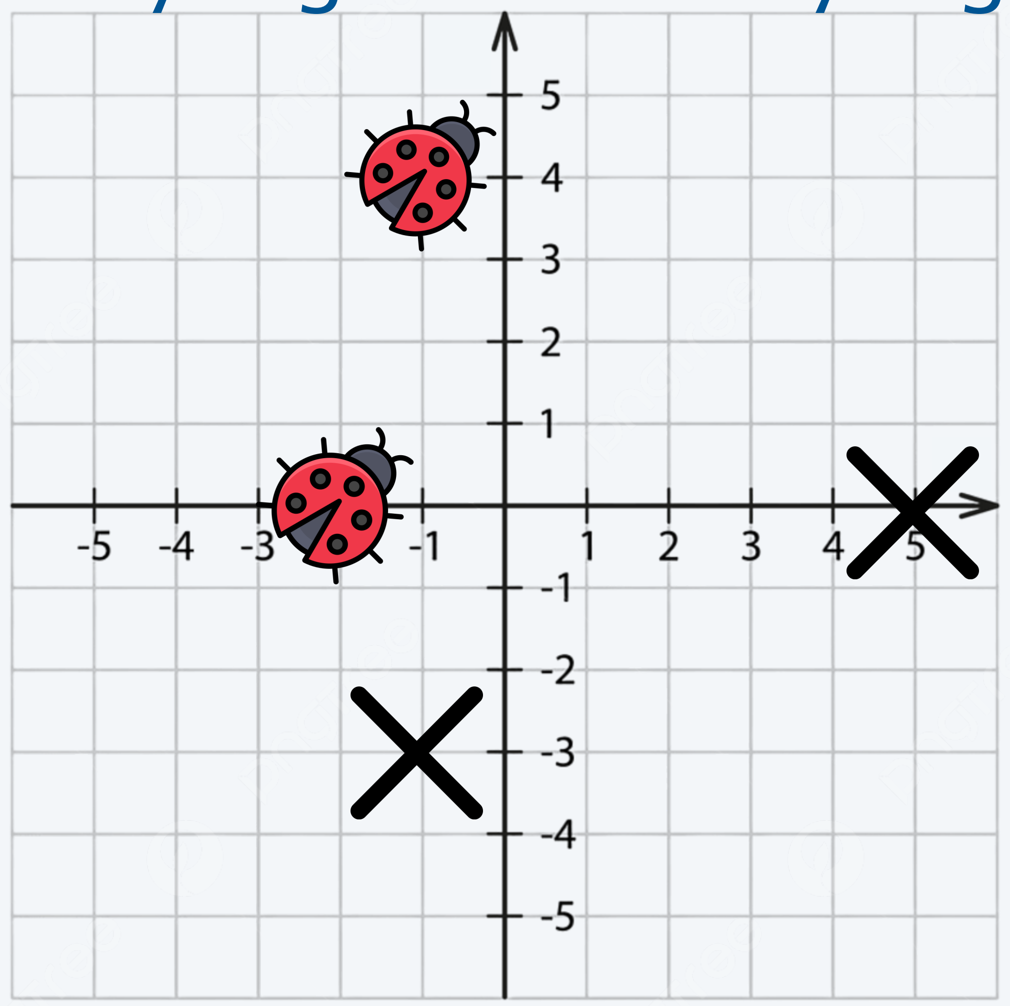
However, we will use a multi–perceptron.

Steps

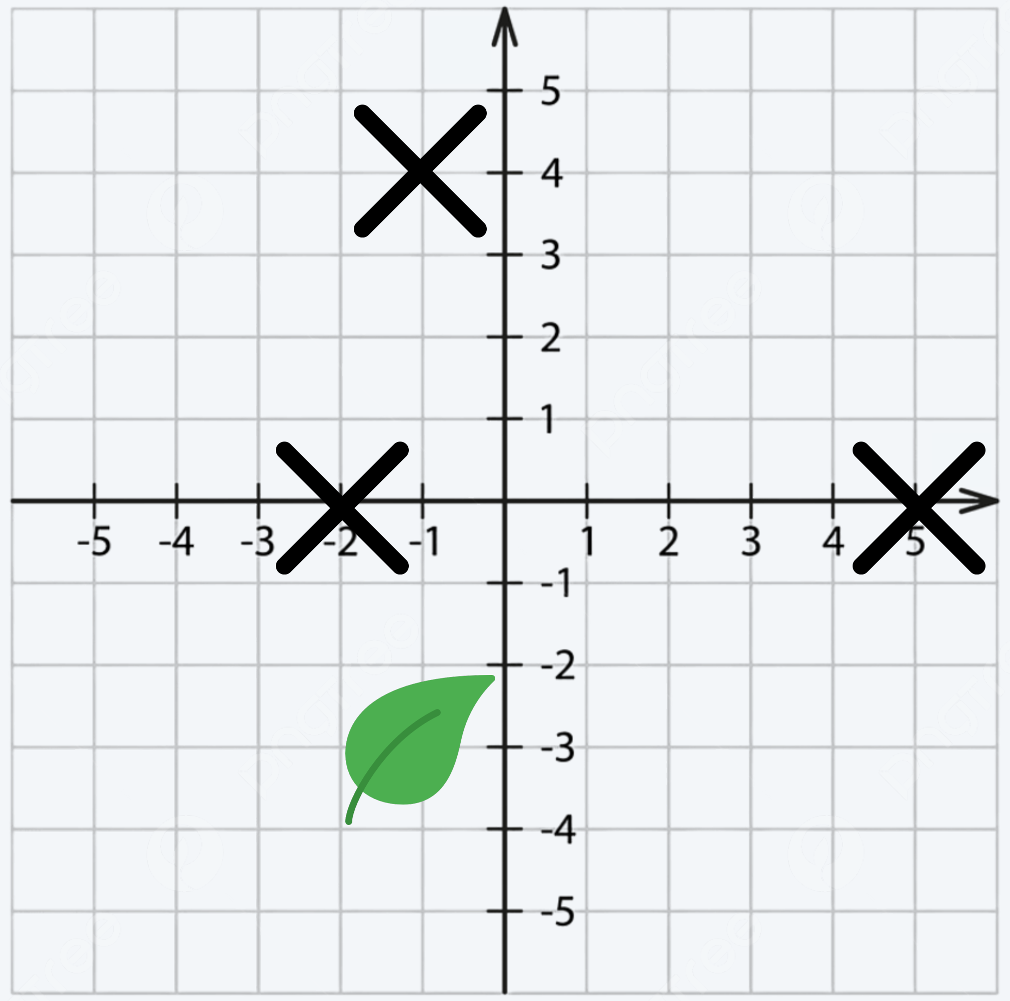
1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.



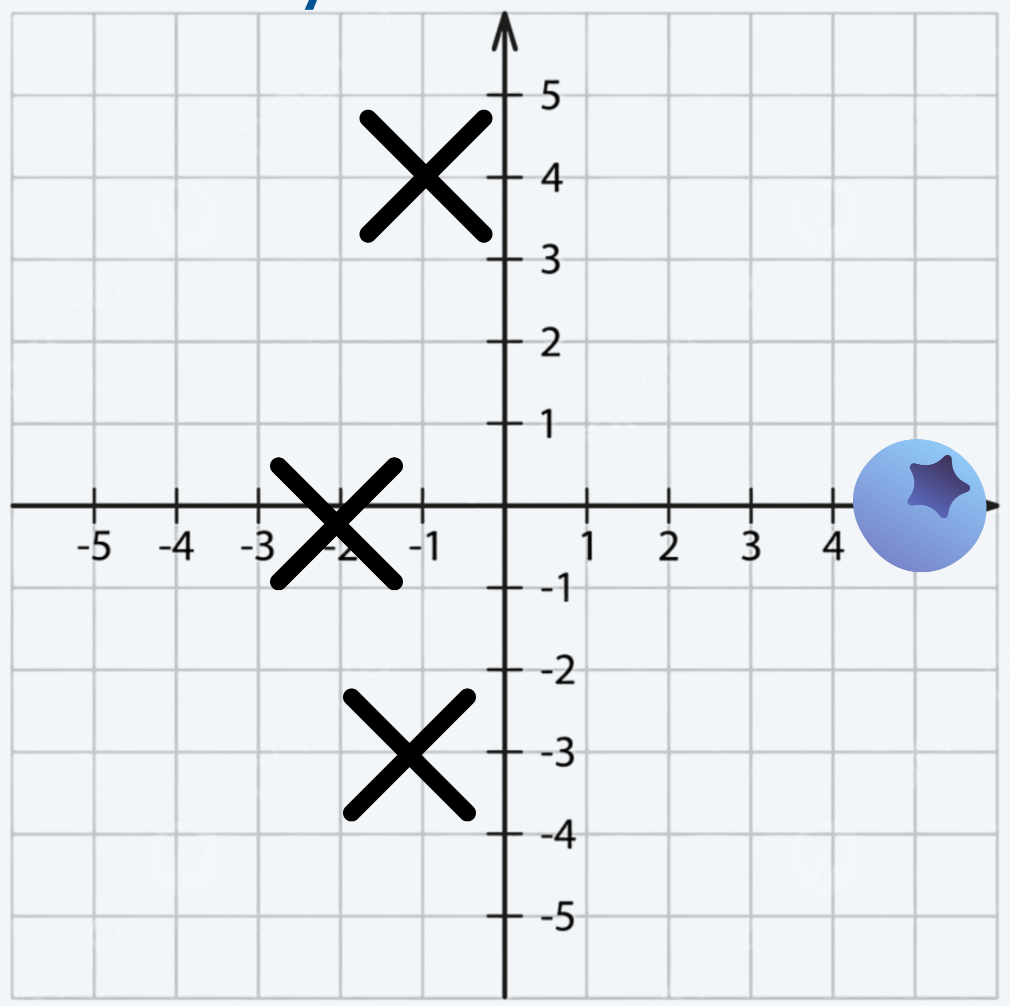
ladybug vs. not ladybug



leaf vs. not leaf



blueberry vs. not blueberry



How to build a multi-class classifier

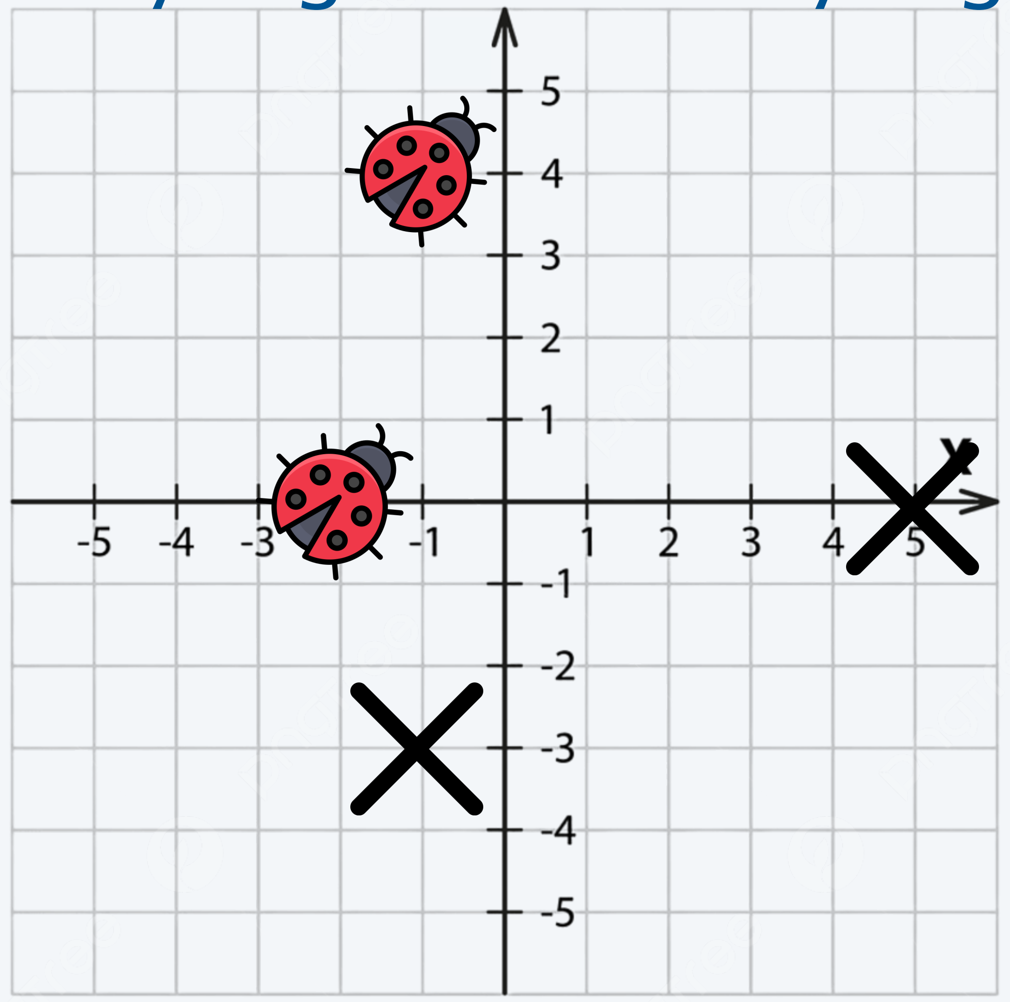
We can use most of the same machinery we used with a binary classifier.

However, we will use a multi–perceptron.

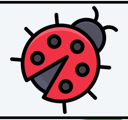
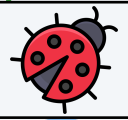
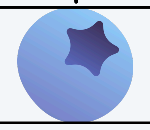
Steps

1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.
2. Train a perceptron for each binary subproblem.
    - A. Assign a positive binary label to members of the corresponding class.
    - B. Assign a negative binary label otherwise.

ladybug vs. not ladybug



training data

x_0	x_1	class label	binary label
−2	0		+1
−1	−3		−1
−1	4		+1
5	0		−1

→ perceptron algorithm → trained classifier (learned weights $\mathbf{w}$)

How to build a multi-class classifier

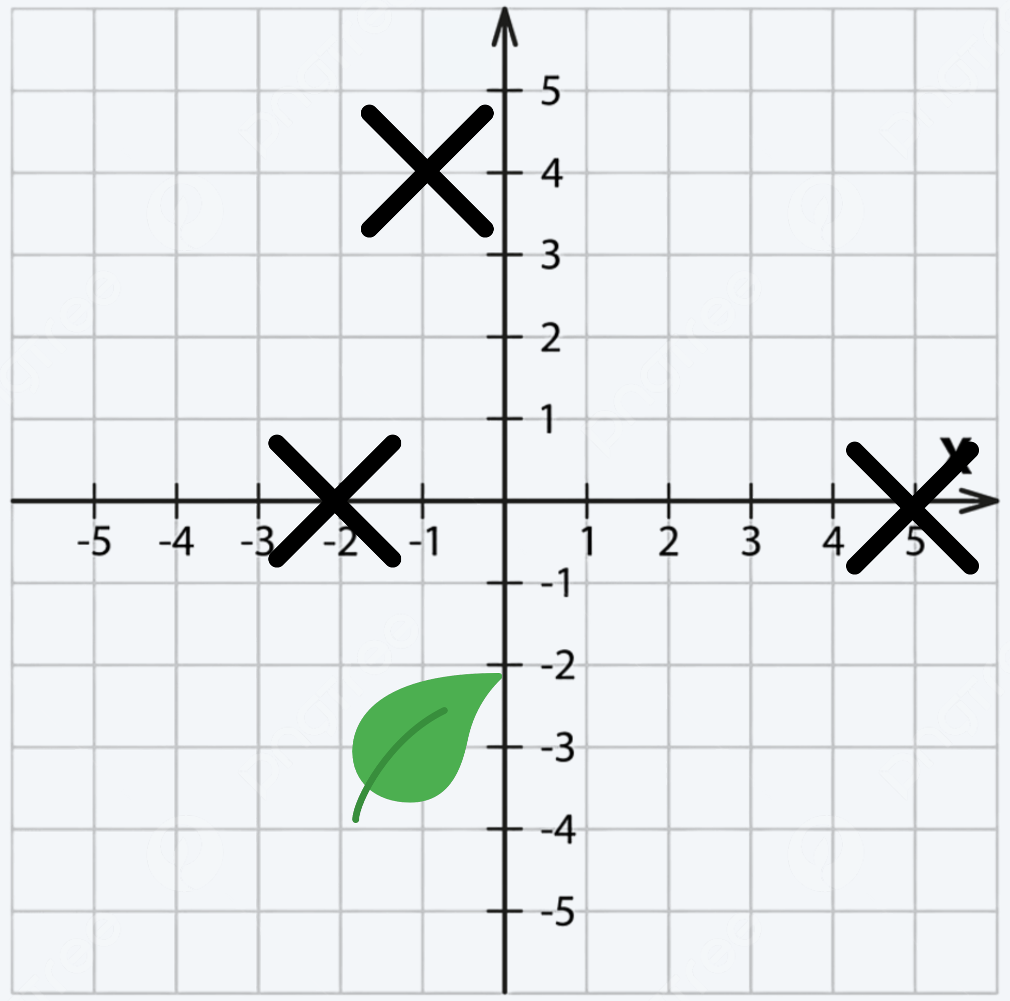
We can use most of the same machinery we used with a binary classifier.

However, we will use a multi–perceptron.

Steps

1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.
2. Train a perceptron for each binary subproblem.
   A. Assign a positive binary label to members of the corresponding class.
   B. Assign a negative binary label otherwise.

leaf vs. not leaf



training data

x_0	x_1	class label	binary label
−2	0		−1
−1	−3		+1
−1	4		−1
5	0		−1

→ perceptron algorithm → trained classifier (learned weights $\mathbf{w}$)

How to build a multi-class classifier

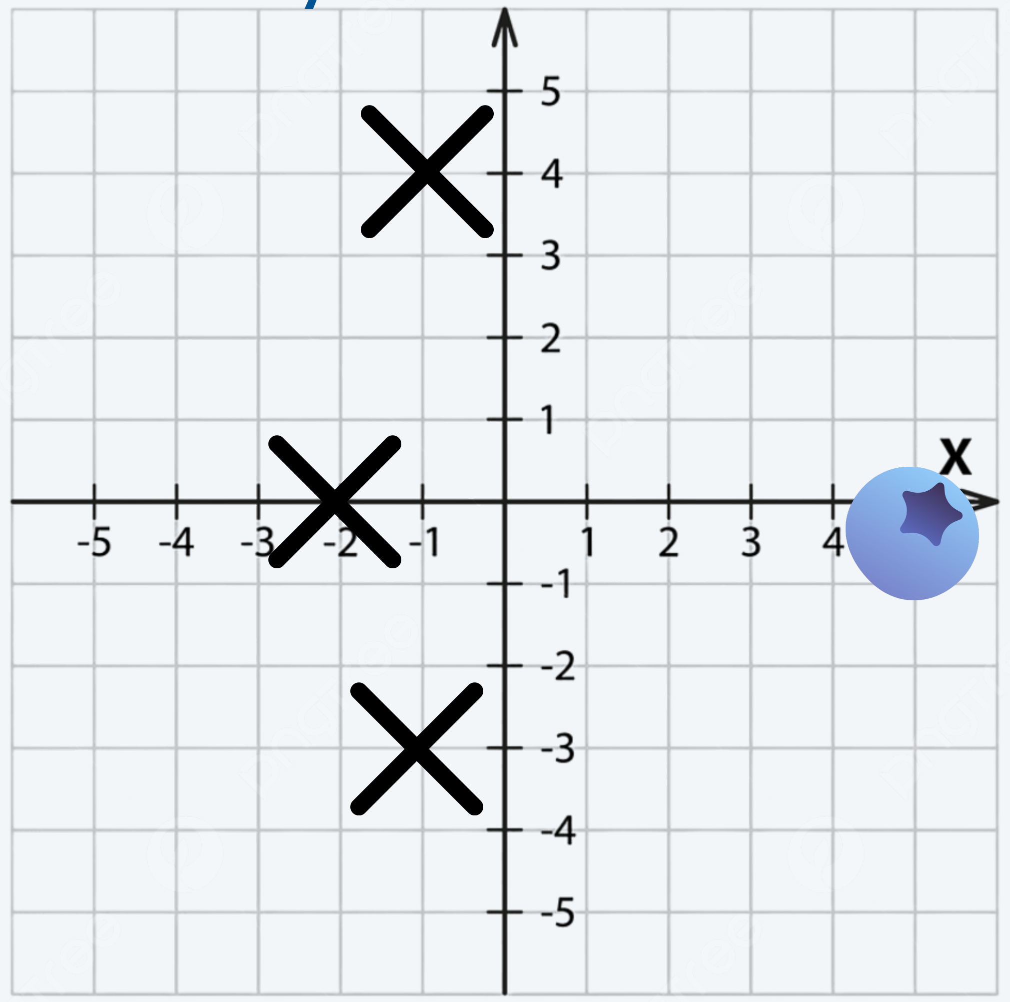
We can use most of the same machinery we used with a binary classifier.

However, we will use a multi–perceptron.

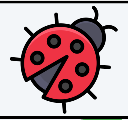
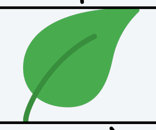
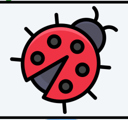
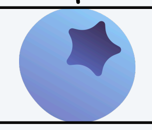
Steps

1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.
2. Train a perceptron for each binary subproblem.
    A. Assign a positive binary label to members of the corresponding class.
    B. Assign a negative binary label otherwise.

blueberry vs. not blueberry



training data

x_0	x_1	class label	binary label
−2	0		−1
−1	−3		−1
−1	4		−1
5	0		+1

→ perceptron algorithm → trained classifier (learned weights $\mathbf{w}$)

How to build a multi-class classifier

We can use most of the same machinery we used with a binary classifier.

However, we will use a multi–perceptron.

Steps

1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.
2. Train a perceptron for each binary subproblem.
    A. Assign a positive binary label to members of the corresponding class.
    B. Assign a negative binary label otherwise.

In the end we will have one vector of weights per binary subproblem.

Subproblem	learned weights w
ladybug vs. not ladybug	(–1, 3)
leaf vs. not leaf	(–1, –3)
blueberry vs. not blueberry	(5, 0)

How to build a multi-class classifier

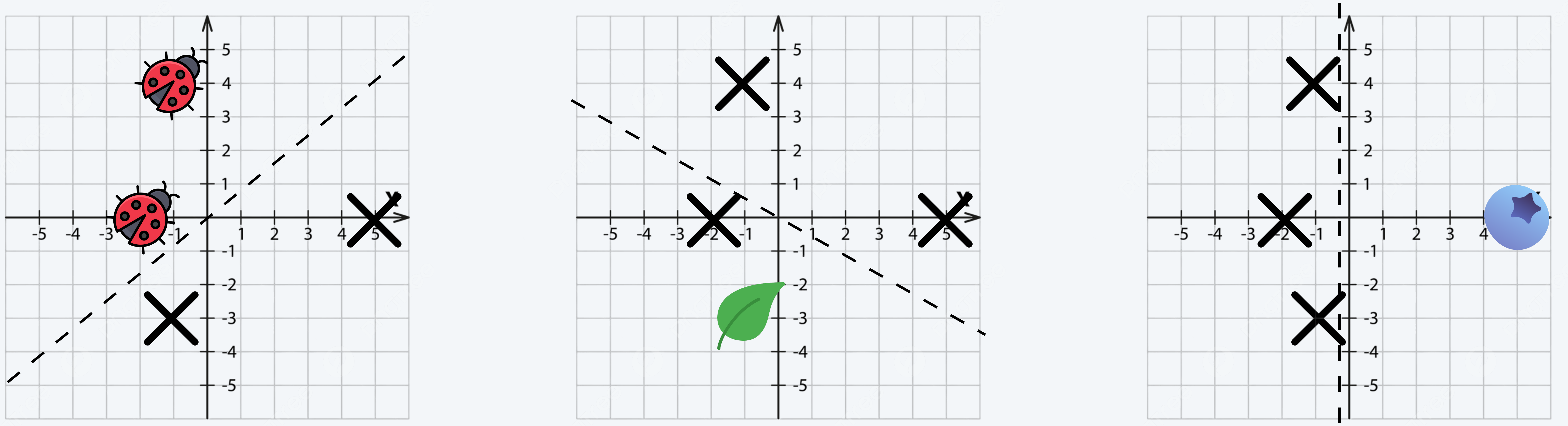
We can use most of the same machinery we used with a binary classifier.

However, we will use a multi–perceptron.

Steps

1. For each unique class label, construct a binary subproblem to recognize between data points of that class and all other data points.
2. Train a perceptron for each binary subproblem.
    A. Assign a positive binary label to members of the corresponding class.
    B. Assign a negative binary label otherwise.

In the end we will have one vector of weights per binary subproblem.



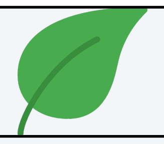
Predicting with a multi-perceptron

We have one vector of weights per binary subproblem.

For input $\mathbf{x}$ of a new data point, we compute the weighted sum S with each of the weight vectors.

Finally, we predict the class with the largest weighted sum.

test data

x_0	x_1	class label	prediction
3	−3		

Subproblem	learned weights w	Weighted Sum (S)
ladybug vs. not ladybug	(−1, 3)	−12
leaf vs. not leaf	(−1, −3)	6
blueberry vs. not blueberry	(5, 0)	15

$$-12 = -1 \cdot (3) + 3 \cdot (-3)$$

$$6 = -1 \cdot 3 + (-3) \cdot (-3)$$

$$-20 = 5 \cdot 3 + 0 \cdot (-3)$$

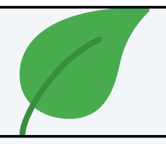
Predicting with a multi-perceptron

We have one vector of weights per binary subproblem.

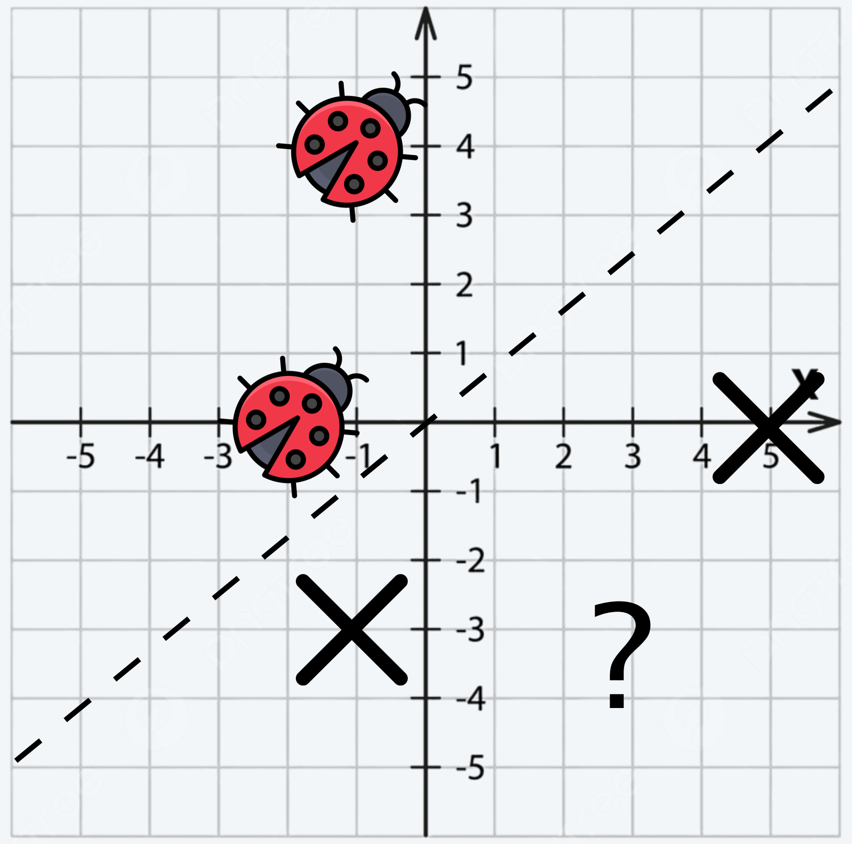
For input $\mathbf{x}$ of a new data point, we compute the weighted sum S with each of the weight vectors.

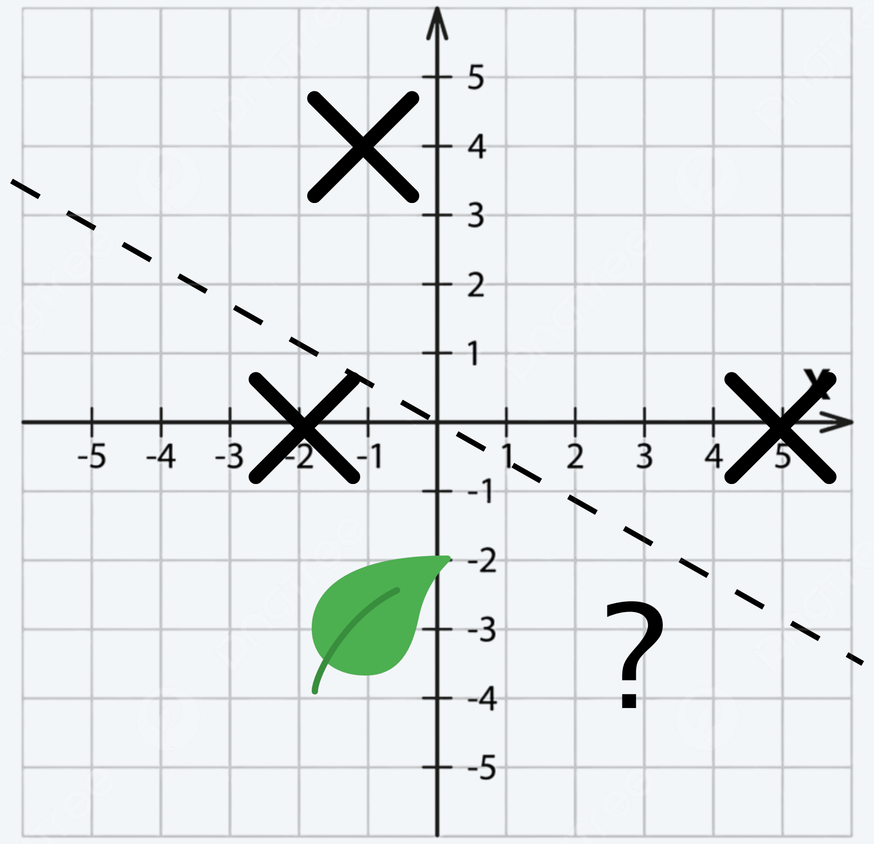
Finally, we predict the class with the largest weighted sum.

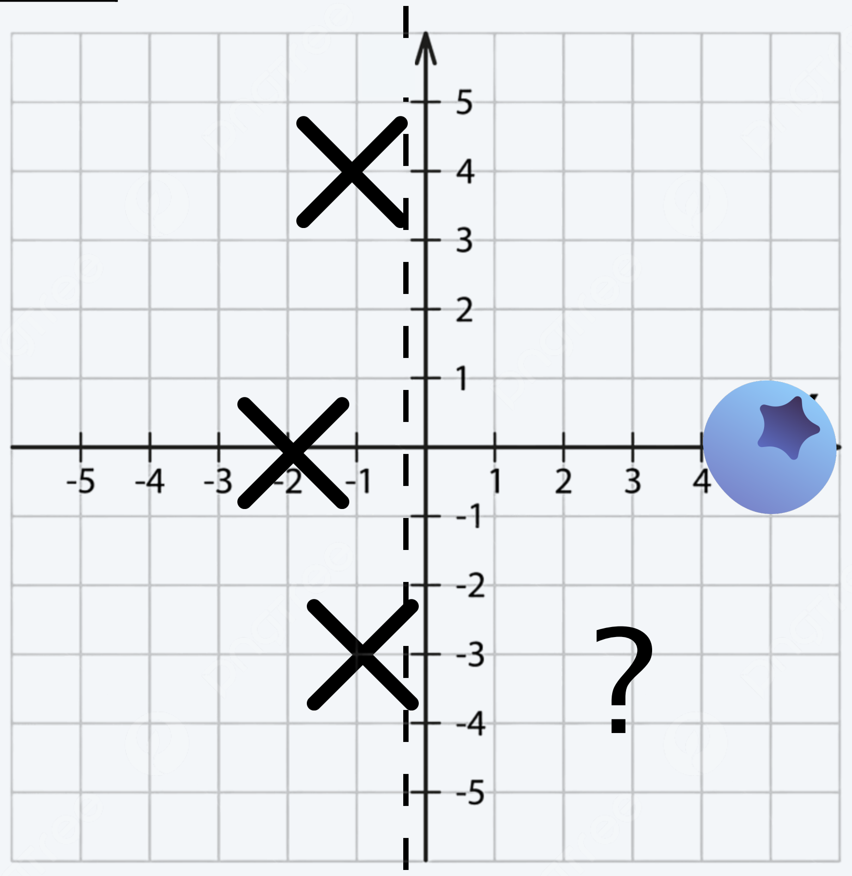
test data

x_0	x_1	class label	prediction
3	−3		

Subproblem	learned weights w	Weighted Sum (S)
ladybug vs. not ladybug	(−1, 3)	−12
leaf vs. not leaf	(−1, −3)	6
blueberry vs. not blueberry	(5, 0)	15







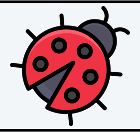
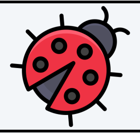
Predicting with a multi-perceptron

We have one vector of weights per binary subproblem.

For input $\mathbf{x}$ of a new data point, we compute the weighted sum S with each of the weight vectors.

Finally, we predict the class with the largest weighted sum.

test data

x_0	x_1	class label	prediction
−4	3		

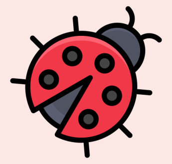
Subproblem	learned weights w	Weighted Sum (S)
ladybug vs. not ladybug	(−1, 3)	13
leaf vs. not leaf	(−1, −3)	−5
blueberry vs. not blueberry	(5, 0)	−20

$$13 = -1 \cdot (-4) + 3 \cdot 3$$

$$-5 = -1 \cdot (-4) + (-3) \cdot 3$$

$$-20 = 5 \cdot (-4) + 0 \cdot 3$$

What is the prediction of our multi-perceptron on the given test point?

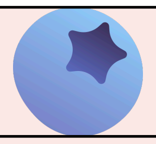
A. 

B. 

C. 

Subproblem	learned weights w
ladybug vs. not ladybug	(−1, 3)
leaf vs. not leaf	(−1, −3)
blueberry vs. not blueberry	(5, 0)

test data

x_0	x_1	class label
2	1	

Concluding Thoughts

We can write programs that improve with experience

- Machine learning uses old data as experience.
- We want to improve on new data.
- This is challenging and requires care.

Machine learning is inter-disciplinary.

- Intersection of computer science and statistics.
- The utility of our models is a domain specific matter.

Model utility goes beyond a performance metric.

- We want machine learning models to satisfy other criteria beyond good performance.
- For example, we want our models to be fair, interpretable, etc.

Please complete the mid-semester feedback survey ❤

(details on Ed)

More questions

attend office hours

ask on Ed