Attendance Code:
98446577

Artificial Intelligence and Machine Learning (AIML)

2023–24

Attendance Code:
98446577

- **Last lectures**:
  - Introduction to ML
    - Sequential Gradient Descent (SGD) algorithm
    - Supervised learning: regression
    - Unsupervised learning: clustering and K-means

- **This lecture**: classification in ML

AIML

Attendance Code:
98446577

Example: health insurance company

- Data on the annual premium paid by customers who bought insurance

Client	Age (yrs)	Income (k £)	Premium (£)
1	25	30	800
2	45	60	1500
3	30	50	1200
4	22	25	700
5	35	45	1400
6	55	70	1800
7	40	55	1300
8	60	80	2000
9	50	40	1600
10	28	35	900

- Task: predict the annual premium of a new customer, given their age and income. How can we approach this problem?

AIML

Attendance Code:
98446577

Example: health insurance company

- Data on the annual premium paid by customers who bought insurance

Client	Age (yrs)	Income (k £)	Premium (£)
1	25	30	800
2	45	60	1500
3	30	50	1200
4	22	25	700
5	35	45	1400
6	55	70	1800
7	40	55	1300
8	60	80	2000
9	50	40	1600
10	28	35	900

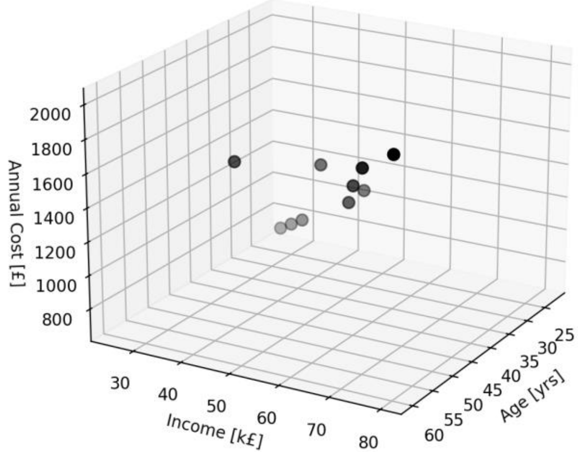
- Task: predict the annual premium of a new customer, given their age and income. How can we approach this problem?
- Regression model using gradient descent:

$$f(w,x) = w_1 + w_2 x^1 + w_3 x^2$$

$$= \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 1 \\ x^1 \\ x^2 \end{bmatrix}$$

$$= w^T x$$

$$F(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2$$



AIML

Attendance Code:
98446577

Example: health insurance company

- Data on the annual premium paid by customers who bought insurance

Client	Age (yrs)	Income (k £)	Premium (£)
1	25	30	800
2	45	60	1500
3	30	50	1200
4	22	25	700
5	35	45	1400
6	55	70	1800
7	40	55	1300
8	60	80	2000
9	50	40	1600
10	28	35	900

- Task: predict the annual premium of a new customer, given their age and income. How can we approach this problem?
- Regression model using gradient descent:

$$f(w,x) = w_1 + w_2 x^1 + w_3 x^2$$

$$= \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 1 \\ x^1 \\ x^2 \end{bmatrix}$$

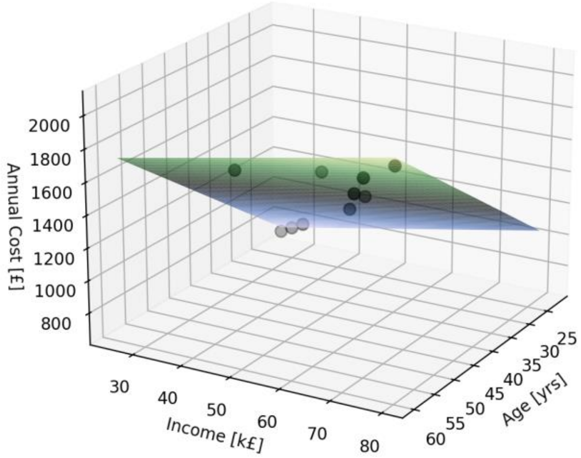
$$= w^T x$$

$$F(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2$$

$$w_0 = [0 \quad 0 \quad 0]^T$$

$$w_n = w_{n-1} - \alpha F_w(w,x)$$

$$w^* = [1.6 \quad 26.4 \quad 5.8]$$



AIML

Attendance Code:
98446577

Example: health insurance company

- Data on whether customers bought the plan

Client	Age (yrs)	Income (k £)	Bought?
1	25	30	No
2	45	60	Yes
3	30	50	Yes
4	22	25	No
5	35	45	Yes
6	55	70	Yes
7	40	55	No
8	60	80	Yes
9	50	40	No
10	28	35	No

- Task: predict whether a new customer is likely to buy or not the plan, given their age and income.
- How does this problem compare with the previous one?

AIML

Attendance Code:
98446577

Example: health insurance company

- Data on whether customers bought the plan

Client	Age (yrs)	Income (k £)	Bought?
1	25	30	No
2	45	60	Yes
3	30	50	Yes
4	22	25	No
5	35	45	Yes
6	55	70	Yes
7	40	55	No
8	60	80	Yes
9	50	40	No
10	28	35	No

- Task: predict whether a new customer is likely to buy or not the plan, given their age and income.
- How does this problem compare with the previous one?
    - **Supervised** Problem (labelled data)

AIML

Attendance Code:
98446577

Example: health insurance company

- Data on whether customers bought the plan

Client	Age (yrs)	Income (k £)	Bought?
1	25	30	No
2	45	60	Yes
3	30	50	Yes
4	22	25	No
5	35	45	Yes
6	55	70	Yes
7	40	55	No
8	60	80	Yes
9	50	40	No
10	28	35	No

- Task: predict whether a new customer is likely to buy or not the plan, given their age and income.
- How does this problem compare with the previous one?
  - **Supervised** Problem (labelled data)
  - Prediction of a categorical label: **classification model**

AIML

Attendance Code:
98446577

Example: health insurance company

- Data on whether customers bought the plan

Client	Age (yrs)	Income (k £)	Bought?
1	25	30	No
2	45	60	Yes
3	30	50	Yes
4	22	25	No
5	35	45	Yes
6	55	70	Yes
7	40	55	No
8	60	80	Yes
9	50	40	No
10	28	35	No

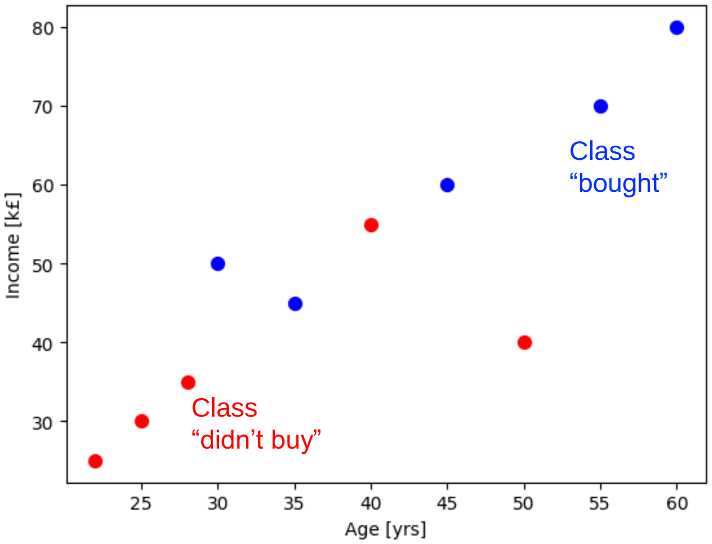
- Task: predict whether a new customer is likely to buy or not the plan, given their age and income.
- How does this problem compare with the previous one?
  - **Supervised** Problem (labelled data)
  - Prediction of a categorical label: **classification model**
  - Goal: split the data into 2 **classes** (bought/didn't buy) that best match **class-labeled training data**.

AIML

Attendance Code: **98446577**

Example: health insurance company

- Data on whether customers bought the plan

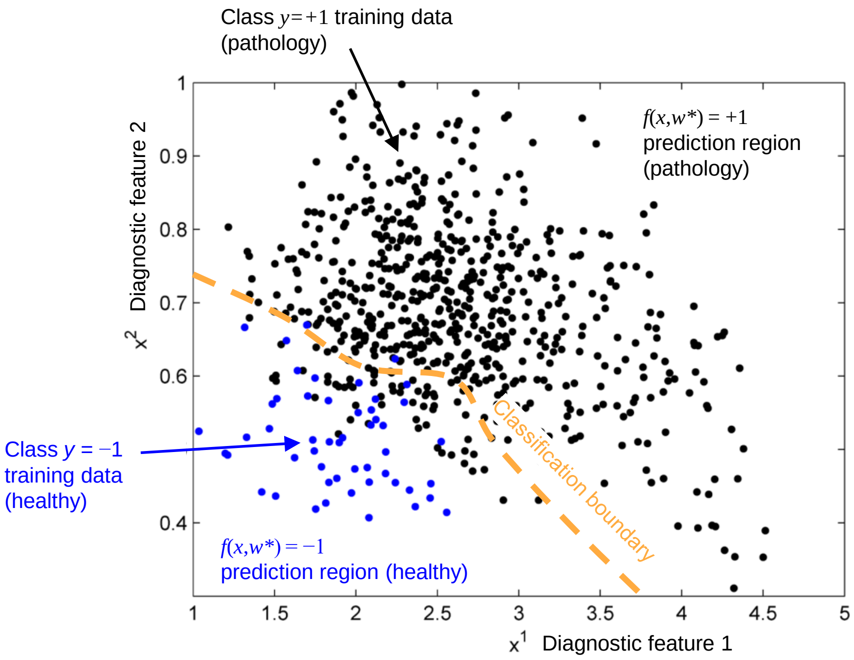


- Task: predict whether a new customer is likely to buy or not the plan, given their age and income.
- How does this problem compare with the previous one?
    - **Supervised** Problem (labelled data)
    - Prediction of a categorical label: **classification model**
    - Goal: split the data into 2 **classes** (bought/didn't buy) that best match **class-labeled training data**.
    - Hypothesis: there is some **decision boundary** in the data which makes this classification possible

AIML

Attendance Code: **98446577**

Machine learning (ML): classification

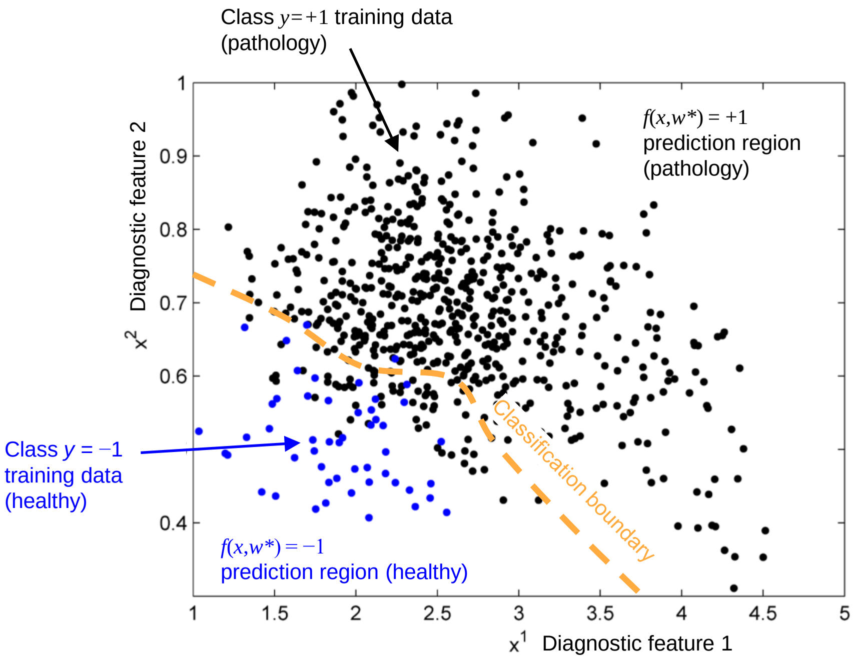


- Medical decision-making: automate process of triage (eliminating non-suspect cases), train a classification algorithm on healthy/pathological features, minimizing false positives/false negatives
- Email Classification: find spams to automatically send to Junk
- Service Business: churn prediction
- etc.

AIML

Attendance Code:
98446577

Machine learning (ML): classification

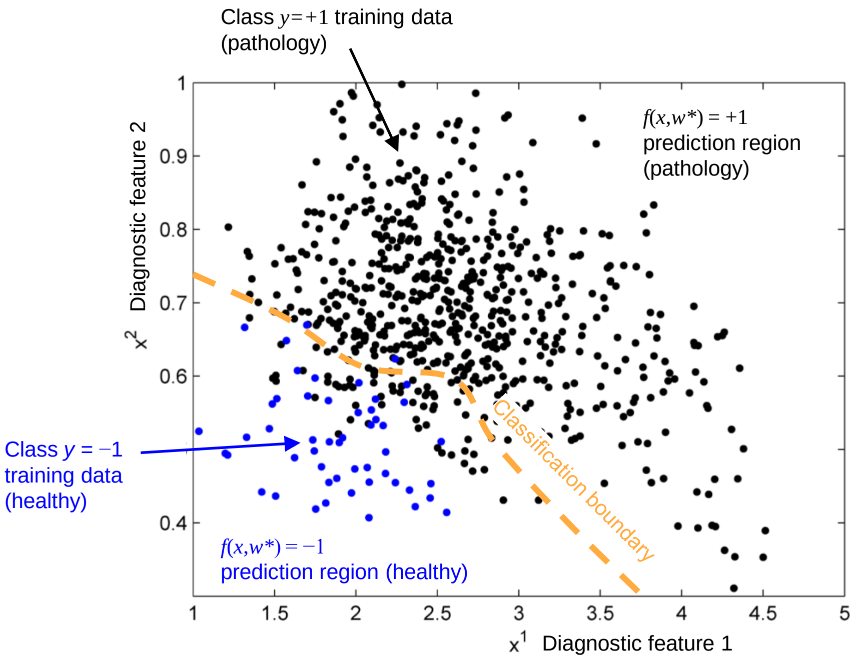


How to solve this problem?

AIML

Attendance Code:
98446577

The perfect classifier



- In principle, any supervised machine learning problem can be completely solved by storing a table of all possible input-output pairs, then prediction is just **table look-up**

Attendance Code:
98446577

The perfect classifier?

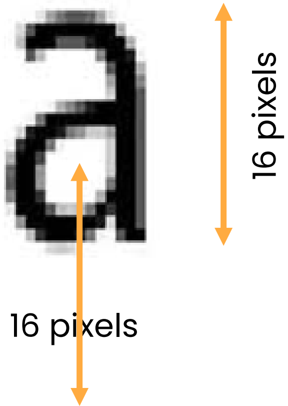
← $2^{16\times16\times8} \approx 10^{616}$ table entries →

Table w	**Image** x					…						…
	Label y	'a'	'a'	'a'	'a'	…	'b'	'b'	'b'	'b'	'b'	…

$f(w, x)$ = in table w, label y of column containing x

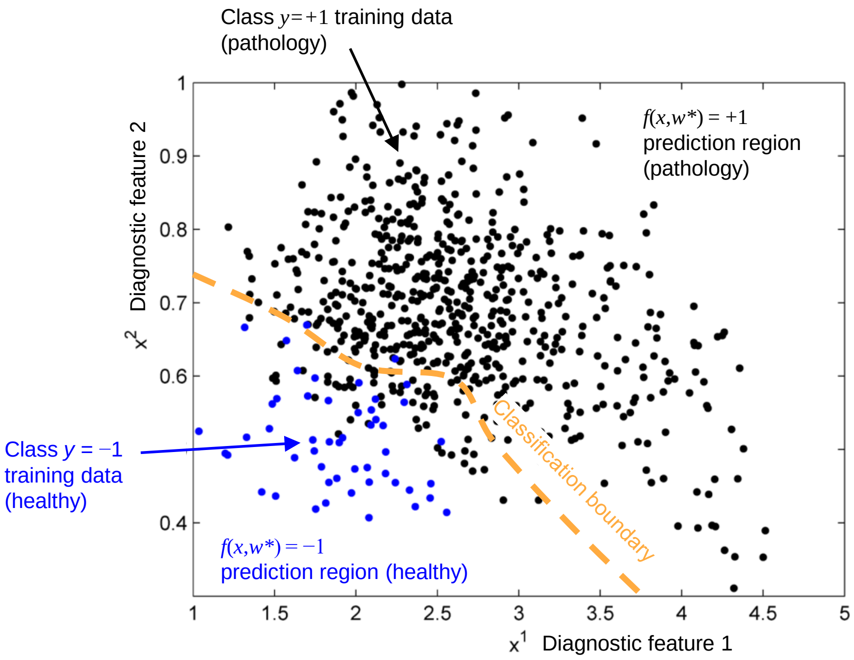
- **Problem:** automated handwriting transcription from digital images
- **Proposed classifier** $f(w, x)$: for every possible handwritten letter, store image and associated label in table w; **look up** letter for any new input image

Simple image: 16×16×8 bits



Attendance Code:
98446577

The perfect classifier – impractical

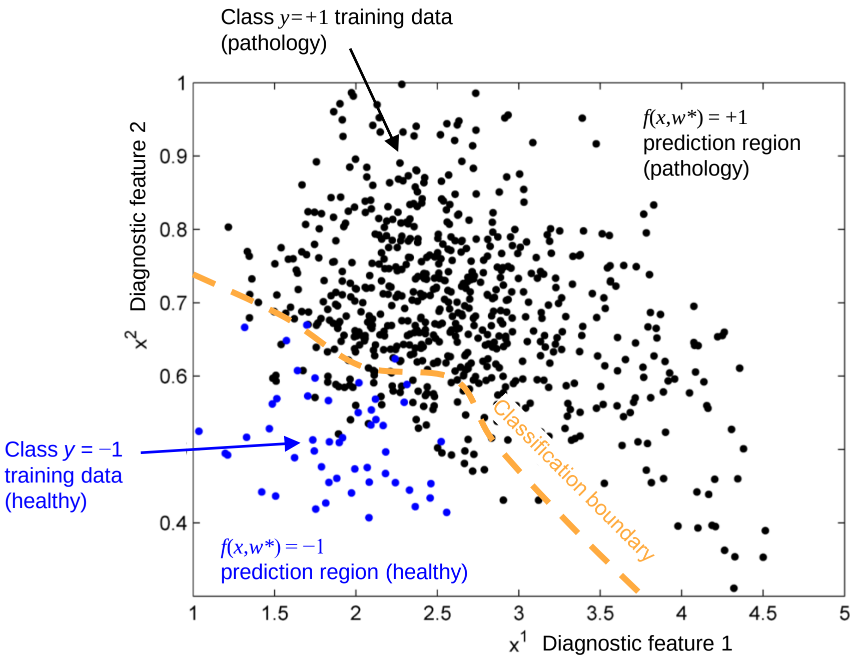


- In principle, any supervised machine learning problem can be completely solved by storing a table of all possible input-output pairs, then prediction is just **table look-up**
- **Impractical** due to combinatorial explosion, so in practice all useful ML classifiers are **imperfect models**

AIML

Attendance Code:
98446577

The perfect classifier – impractical



- In principle, any supervised machine learning problem can be completely solved by storing a table of all possible input-output pairs, then prediction is just **table look-up**
- **Impractical** due to combinatorial explosion, so in practice all useful ML classifiers are **imperfect models**
- **Takeaway**: machine learning is more than just **memorization**

#Code

Classification - outline

We will go through the same conceptual journey as before:

1) Model formulation
2) Cost function
3) Learning algorithm by gradient descent

#Code

1) Model

- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point

#Code

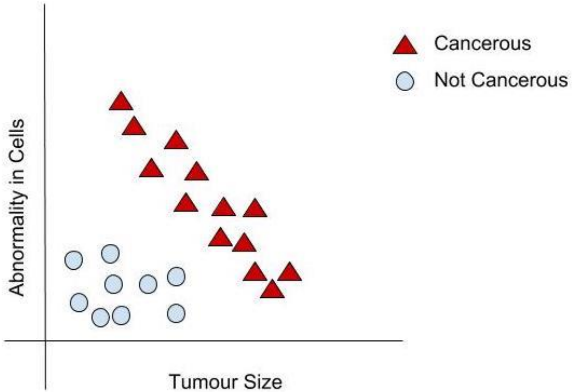
1) Model

- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point

1) Model

- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point

- If *x* has 2 attributes, we can do it with a line

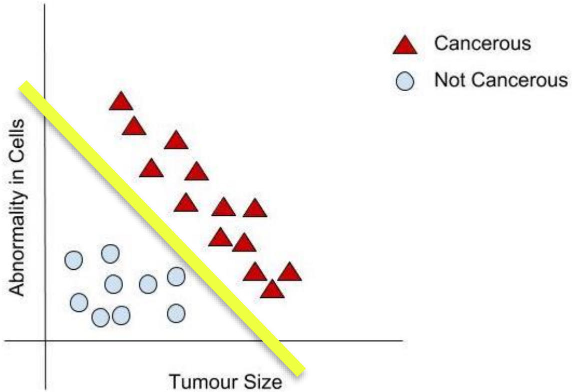


#Code

1) Model

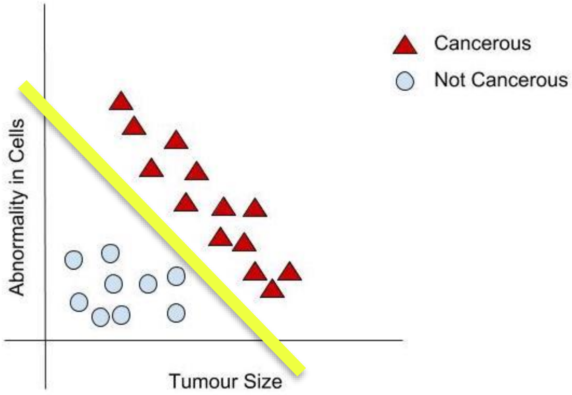
- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point

- If *x* has 2 attributes, we can do it with a line



1) Model

- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point
- If *x* has 2 attributes, we can do it with a line
- If *x* has 3 attributes, we can do it with a plane

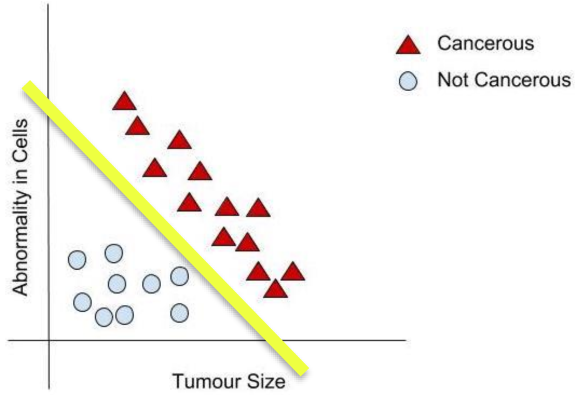


#Code

1) Model

- We want to put a boundary between 2 classes
- If *x* has a single attribute, we can do it with a point

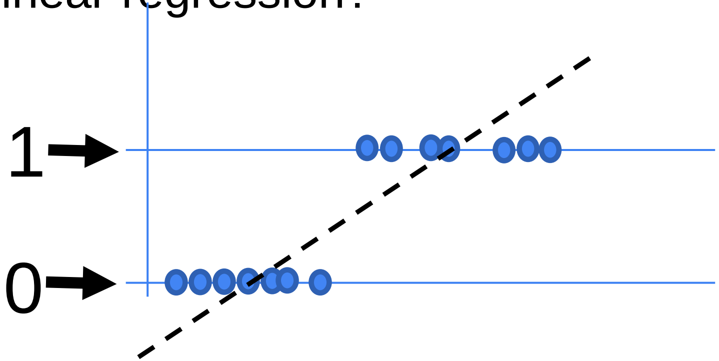
- If *x* has 2 attributes, we can do it with a line



- If *x* has 3 attributes, we can do it with a plane
- If *x* has more than 3 attributes, we can do it with a hyperplane (can't draw it anymore)
- If the classes are linearly separable, the training error will be 0.

#Code

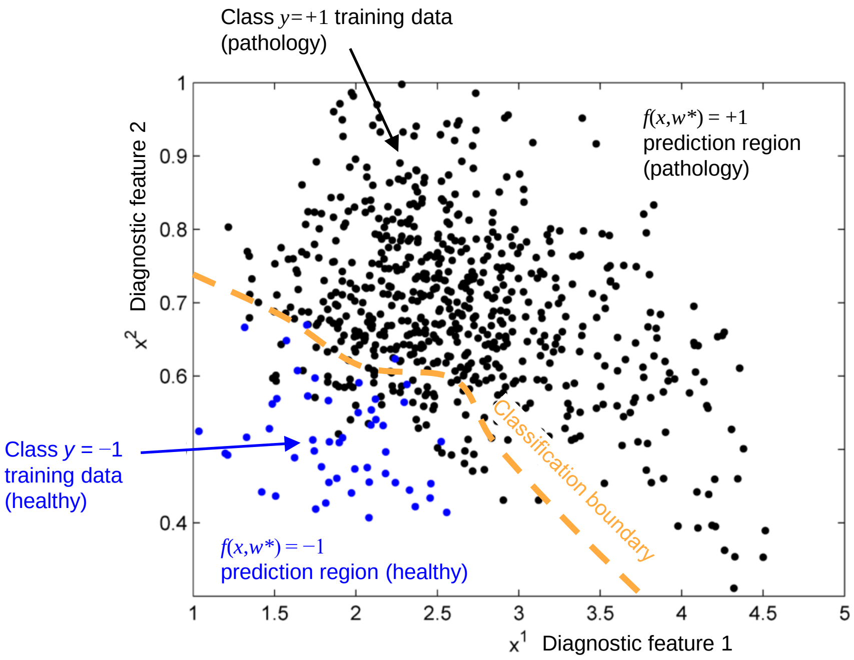
Q: Can you plug classification data into linear regression?



A: Yes. But it might not perform very well. No ordering between categories, like there is between real numbers. We need a better model

Attendance Code:
98446577

ML Classification: Model



- We need a **classification model**, $f(w,x)$, to predict class y_i
- Previous: regression model

$$f(w,x) = w^T x$$

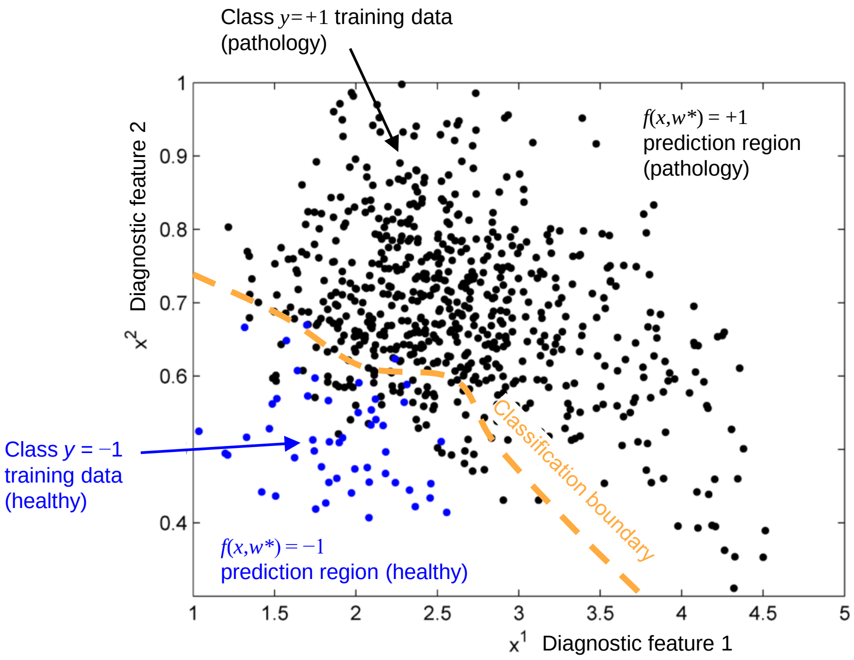
$$f: \mathbb{R}^D \rightarrow \mathbb{R}$$

Classification model:

AIML

Attendance Code:
98446577

ML Classification: Model



- We need a **classification model**, $f(w,x)$, to predict class y_i
- Previous: regression model

$$f(w,x) = w^T x$$

$$f: \mathbb{R}^D \to \mathbb{R}$$

- Classification model:

$$f(w,x) = \text{sign}(w^T x)$$

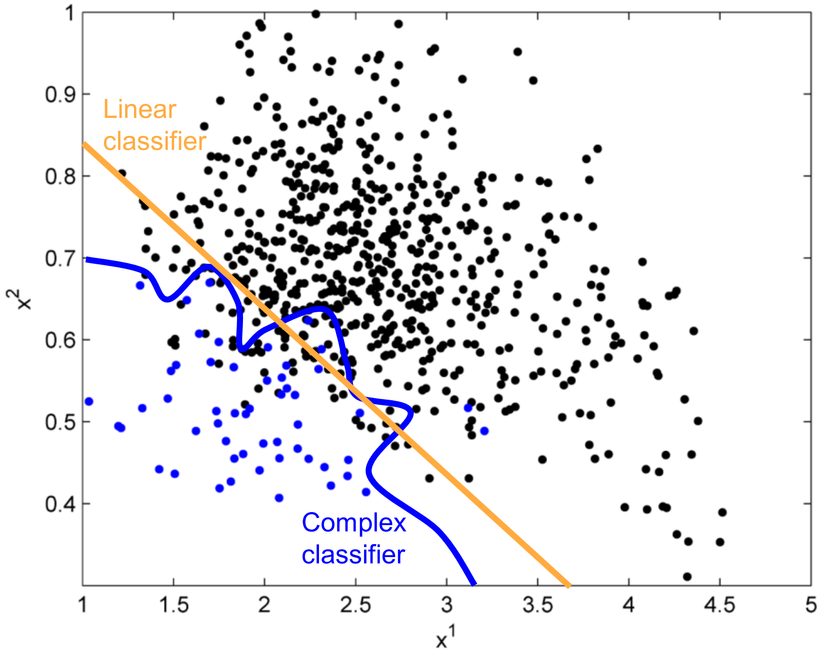
$$f: \mathbb{R}^D \to \{-1,0,1\}$$

- Decision boundary occurs where $w^T x = 0$

Attendance Code:
98446577

ML Classification: Model

A note about classifiers' complexity

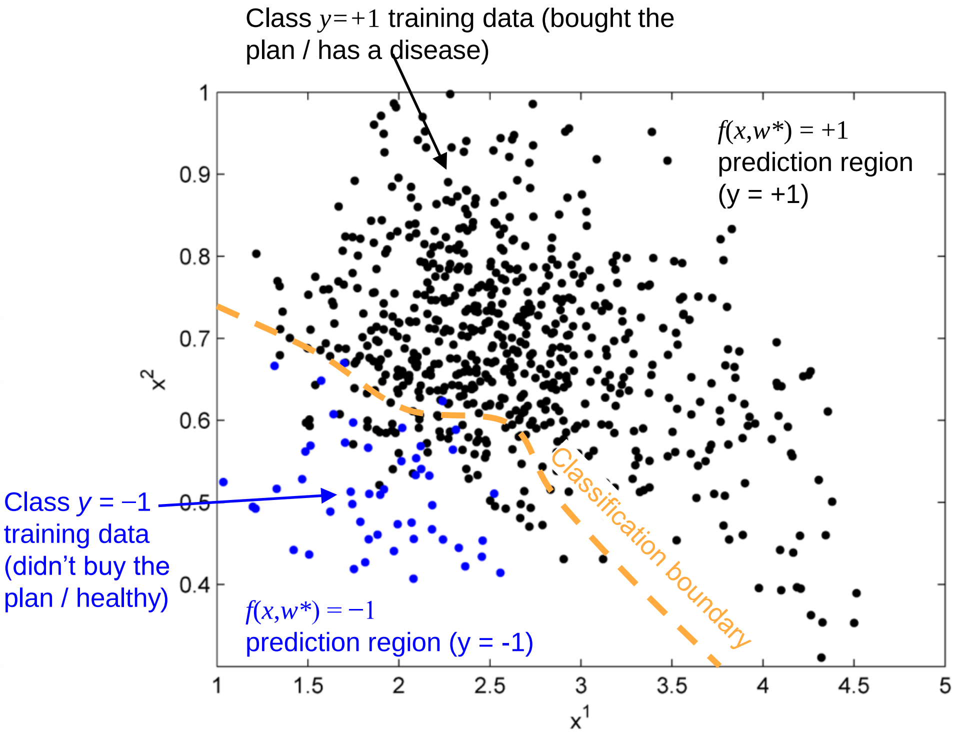


- Complex classifiers can achieve zero training set error but this is the wrong decision boundary if there is randomness to the data
- Linear classifiers may be too simple for most ML applications in the real world
- Best model is usually as simple as possible, but no simpler (Occam's razor)

AIML

Attendance Code:
98446577

ML Classification: Objective Function

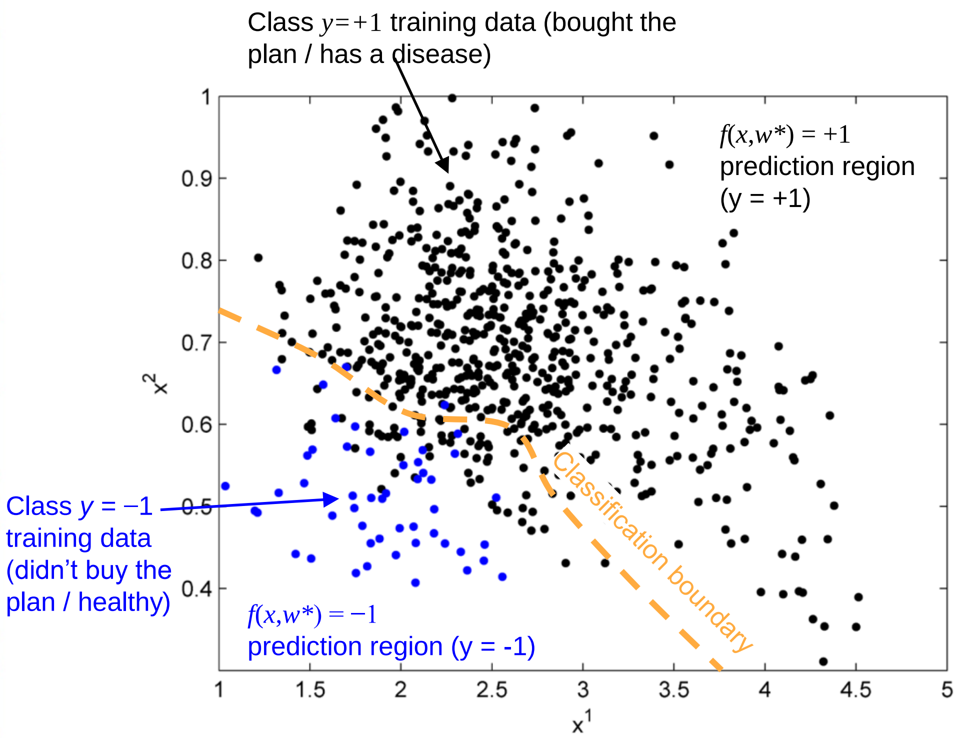


- ML problem to be solved:

$$w^\star = \arg\min_{w' \in \mathcal{W}} F(w')$$

Attendance Code:
98446577

ML Classification: Objective Function



- ML problem to be solved:

$$w^\star = \underset{w' \in \mathcal{W}}{\arg\min}\, F(w')$$

- The **misclassification error** can be expressed mathematically as

$$F(w) = \sum_{i=1}^{N} \mathbb{I}[f(w, x_i) \neq y_i]$$

where the indicator $\mathbb{I}[P] = 1$ if logical condition P is true, and 0 otherwise.

Attendance Code:
98446577

SGD: algorithm (Section 9 Lecture Notes)

- **Step 1**. *Initialization*: Select an initial guess for w_0, a convergence tolerance $\varepsilon > 0$, step size (learning rate) parameter $\alpha > 0$, set iteration number $n=0$
- **Step 2**. *Gradient descent step*: Compute new model parameters,

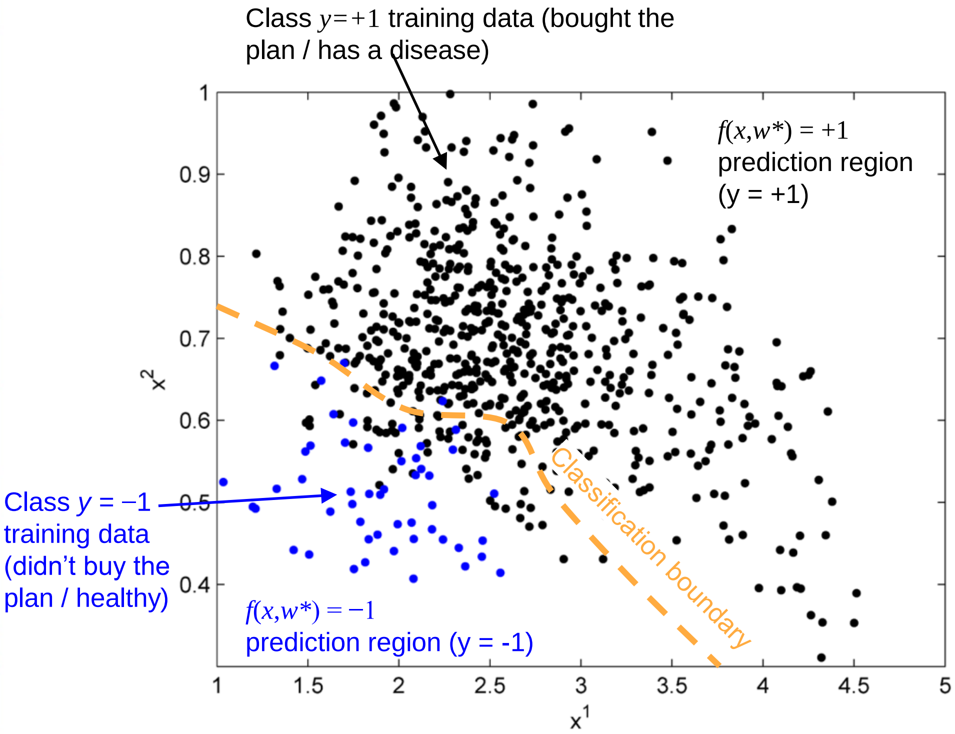
$$w_{n+1} = w_n - \alpha\, F_w(w_n)$$

- **Step 3**. *Convergence test*: Compute new loss function value $F(w_{n+1})$, and loss function improvement, $\Delta F = |F(w_{n+1}) - F(w_n)|$ and if $\Delta F < \varepsilon$, exit with solution $w^*=w_{n+1}$
- **Step 4**. *Iteration*: update $n=n+1$ and go to step 2.

AIML

Attendance Code: **98446577**

ML Classification: Objective Function



- ML problem to be solved:

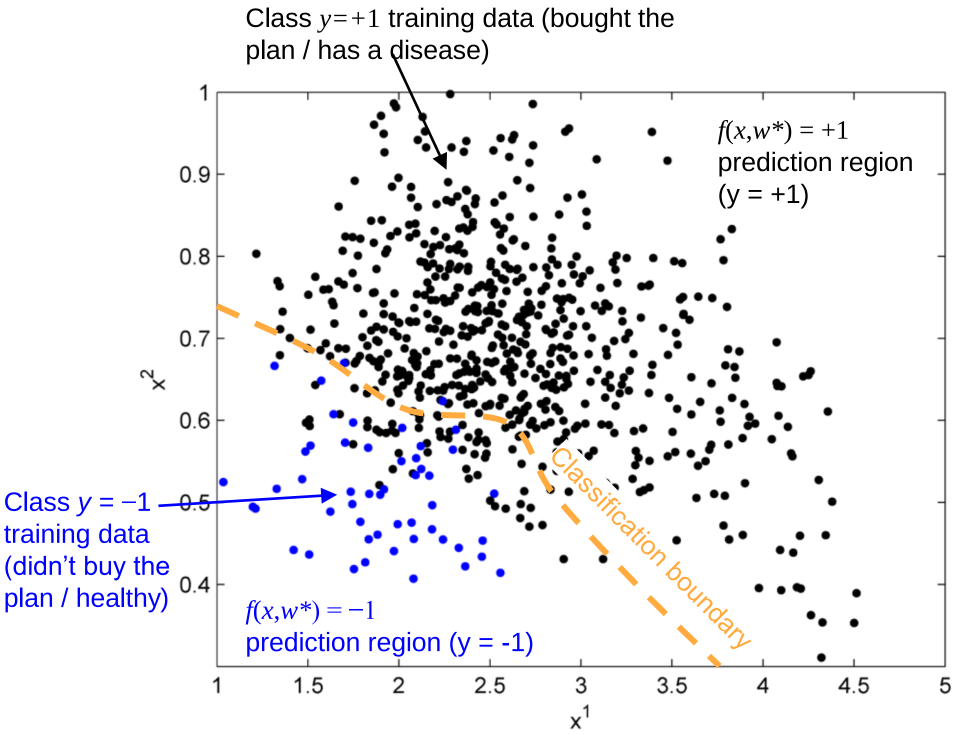
$$w^\star = \underset{w' \in \mathcal{W}}{\arg\min}\, F(w')$$

- Binary nature of classification: **misclassification error**, which can be expressed mathematically as

$$F(w) = \sum_{i=1}^{N} \mathbb{I}[f(w, x_i) \neq y_i]$$

where the indicator $\mathbb{I}[P] = 1$ if logical condition P is true, and 0 otherwise.

ML Classification: Objective Function



- ML problem to be solved:

$$w^\star = \arg\min_{w' \in \mathcal{W}} F(w')$$

- Binary nature of classification: **misclassification error**, which can be expressed mathematically as

$$F(w) = \sum_{i=1}^{N} \mathbb{I}[f(w, x_i) \neq y_i]$$

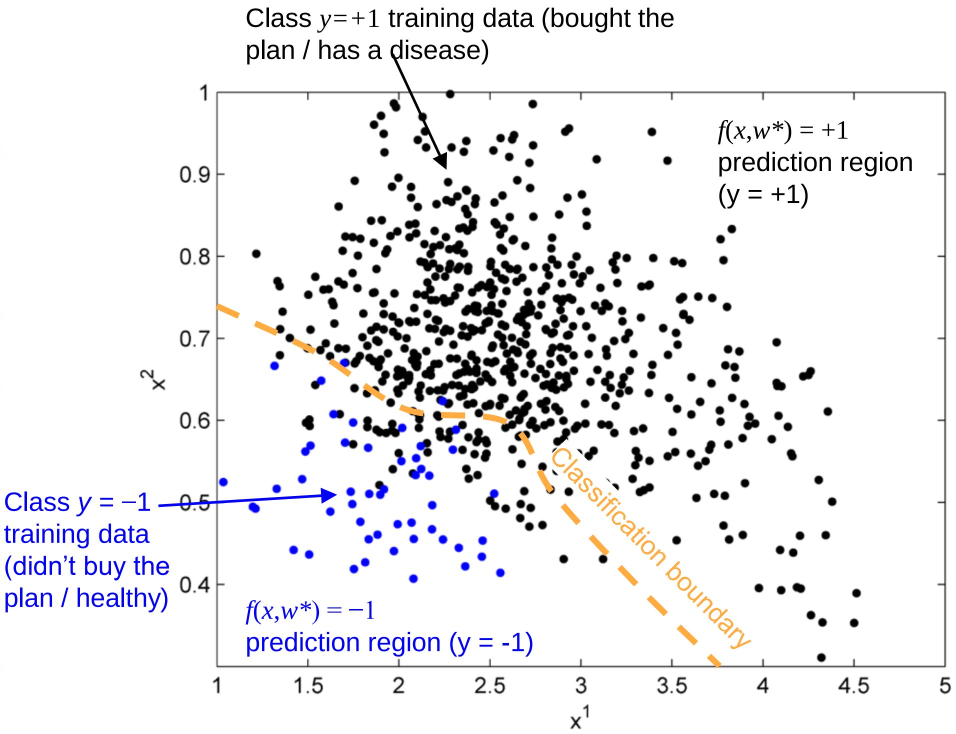
where the indicator $\mathbb{I}[P] = 1$ if logical condition P is true, and 0 otherwise.

- **Problem:** Very challenging to solve the optimal misclassification error problem (bad gradients: 0 or not defined!)

AIML

Attendance Code: **98446577**

ML Classification: Objective Function



ML problem to be solved:

$$w^\star = \arg\min_{w' \in \mathcal{W}} F(w')$$

Proxy or surrogate error (loss) that is easier to optimize:

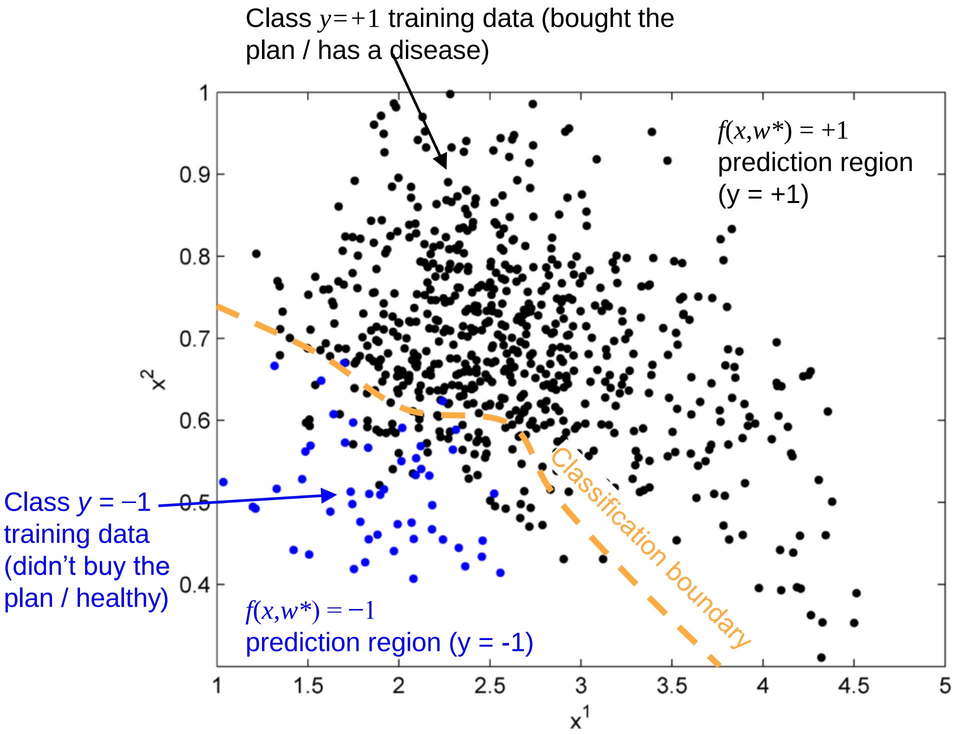
- **Perceptron loss:**

$$F(w) = \sum_{i=1}^{N} \max\left[0, -y_i f(w, x_i)\right]$$

AIML

Attendance Code:
98446577

ML Classification: Objective Function



ML problem to be solved:

$$w^\star = \arg\min_{w' \in \mathcal{W}} F(w')$$

Proxy or surrogate error (loss) that is easier to optimize:

- **Perceptron loss:**

$$F(w) = \sum_{i=1}^{N} \max\left[0, -y_i f(w, x_i)\right]$$

- **Logistic loss:**

$$F(w) = \sum_{i=1}^{N} \log\left[1 + e^{-y_i f(w, x_i)}\right]$$

- **Hinge loss:**

$$F(w) = \sum_{i=1}^{N} \max\left[0, 1 - y_i f(w, x_i)\right]$$

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w, x)]$

Regression	Classification
$w^T x$	$\text{sign}\ (w^T x)$
> 0 (positive)	
< 0 (negative)	

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w, x)]$

Regression	Classification
$w^T x$	$\text{sign}\ (w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w,x)]$

Regression	Classification
$w^T x$	$\text{sign}\,(w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

Misclassification error
$\mathbb{I}[f(w,x_i) \neq y_i]$

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w,x)]$

Regression	Classification
$w^T x$	$\text{sign}\ (w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error
y	$\mathbb{I}[f(w, x_i) \neq y_i]$
$+1$	
-1	
-1	
$+1$	

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w, x)]$

Regression	Classification
$w^T x$	$\text{sign}\,(w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error
y	$\mathbb{I}[f(w, x_i) \neq y_i]$
$+1$	0
-1	1
-1	0
$+1$	1

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w, x)]$

Regression	Classification
$w^T x$	$\text{sign}\ (w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error	Perceptron loss	
y	$\mathbb{I}[f(w, x_i) \neq y_i]$		$max\ (0, -yw^T x)$
$+1$	0		
-1	1		
-1	0		
$+1$	1		

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w, x)]$

Regression	Classification
$w^T x$	sign $(w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error	Perceptron loss	
y	$\mathbb{I}[f(w, x_i) \neq y_i]$	$-yw^T x$	$max\,(0, -yw^T x)$
$+1$	0		
-1	1		
-1	0		
$+1$	1		

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w,x)]$

Regression	Classification
$w^T x$	sign $(w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error	Perceptron loss	
y	$\mathbb{I}[f(w,x_i) \neq y_i]$	$-yw^T x$	$max\ (0, -yw^T x)$
$+1$	0	$-w^T x$	
-1	1	$w^T x$	
-1	0	$w^T x$	
$+1$	1	$-w^T x$	

AIML

Attendance Code:
98446577

ML Classification: Comparisons

Model $[f(w,x)]$

Regression	Classification
$w^T x$	$\text{sign}\,(w^T x)$
> 0 (positive)	$+1$
< 0 (negative)	-1

Objective Function $[F(w)]$

True Label	Misclassification error	Perceptron loss	
y	$\mathbb{I}[f(w,x_i) \neq y_i]$	$-yw^T x$	$max\,(0, -yw^T x)$
$+1$	0	$-w^T x$	0
-1	1	$w^T x$	$w^T x$
-1	0	$w^T x$	$w^T x$
$+1$	1	$-w^T x$	0

Attendance Code:
98446577

AIML

To recap

- We learned the **principles of classifier model** training and contrasted it with memorization.
- We appreciated the **difficulties** in using the **misclassification error** and analyzed one alternative of **surrogate loss** (perceptron loss).
  - Mathematically convenient but in general not guaranteed to find the classifier which **globally** minimizes the misclassification error.

Further Reading

- **R&N**, Section 18.8
- **PRML**, Section 2.5
- **H&T**, Section 13.3

Attendance Code:
98446577

To recap

- We learned the **principles of classifier model** training and contrasted it with memorization.
- We appreciated the **difficulties** in using the **misclassification error** and analyzed one alternative of **surrogate loss** (perceptron loss).
  - Mathematically convenient but in general not guaranteed to find the classifier which **globally** minimizes the misclassification error.
- **Next**: how to use this classification framework to solve ML classification problems

Further Reading

- **R&N**, Section 18.8
- **PRML**, Section 2.5
- **H&T**, Section 13.3