Day 4: Classification

Summer STEM: Machine Learning

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

August 6, 2020

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

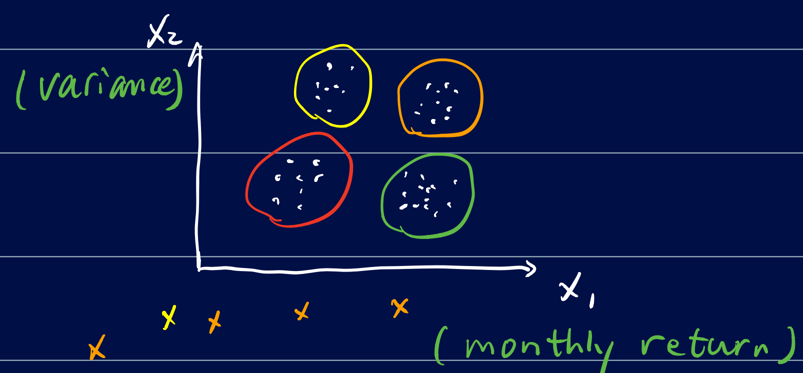
Machine Learning { unsupervised
supervised (regression / classification)

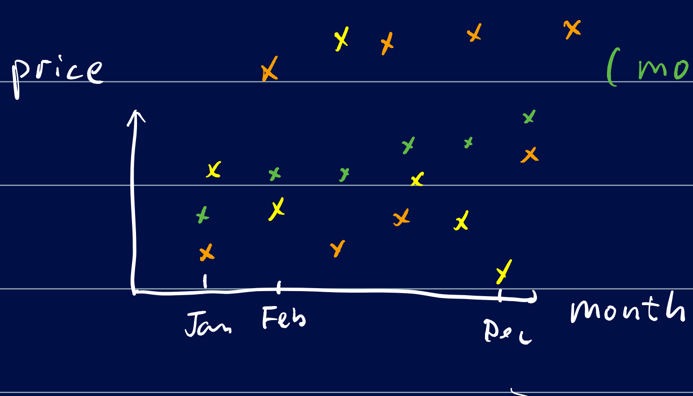
- learn patterns / trends from data
- without specific rules / instructions (autonomous)
- predict

Data features, labels

Supervised data with labels

unsupervised learning data without labels





Supervised Learning

- Classification labels: different classes
  discrete number

  binary $\{0, 1\}$
  $\uparrow$ $\uparrow$
  cats dogs

  $\{0, 1, 2\}$ multiclass classification
  $\uparrow$ $\uparrow$ $\uparrow$
  planes trains cars

- Regression labels: real numbers

$$\longrightarrow y$$

$$\xrightarrow{x} \boxed{f(x;w)} \xrightarrow{\hat{y}}$$

$$w: \quad f(x;w) = w_0 + w_1 x \qquad w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

<u>vector-valued features</u> $\quad x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}$

Remark: We use the subscript i in x_i to denote the i-th sample; but sometimes we use the same notation to denote the i-th component of a vector x.

$$f(x_i; w) = w_0 + w_1 x_{i1} + w_2 x_{i2}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

a model parameterized by w

$f(x)$

scalar-valued feature x Polynomial with an order of M

$$a_0 x^0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_M x^M$$

the order of this polynomial is the highest power $a_i x^i$ where $a_i \neq 0$

Ex: $x + x^4 \rightarrow$ 4-th order

1 0-th order

$x + 3x^4 + 5x^M$ M-th order if $M \geq 4$

$$f(x; w) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M \quad (w_M \neq 0)$$

$$w = [w_0 \ w_1 \ \dots \ w_M]^T$$

Matrix-vector representation of the model

$$f(x) = w_0 + w_1 x$$
$$= \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

let's do this for $i = 1, 2, \dots N$

$$\hat{y}_1 = \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$
$$\hat{y}_2 = \begin{bmatrix} 1 & x_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \Rightarrow \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{bmatrix}}_{X} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$
$$\vdots$$
$$\hat{y}_N = \begin{bmatrix} 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Vector-valued features

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \qquad f(x; w) = w_0 + w_1 x_{i1} + w_2 x_{i2}$$
$$= \begin{bmatrix} 1 & x_{i1} & x_{i2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & & \\ 1 & x_{N1} & x_{N2} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}}_{w}$$

The shape of X, w?

$X \quad (N, 3)$

$w \quad (3, 1)$

$\hat{y} \quad (N, 1)$

$$\boxed{\hat{y} = \boxed{X} w}$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{N1} & x_{N2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} \\ w_0 + w_1 x_{21} + w_2 x_{22} \\ w_0 + w_1 x_{N1} + w_2 x_{N2} \end{bmatrix}$$

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \qquad f(x) = w_0 + w_1 x_{i1} + w_2 x_{i2} = \hat{y}_i$$

$$\hat{y} = Xw \qquad \underline{w = (X^T X)^{-1} X^T y} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

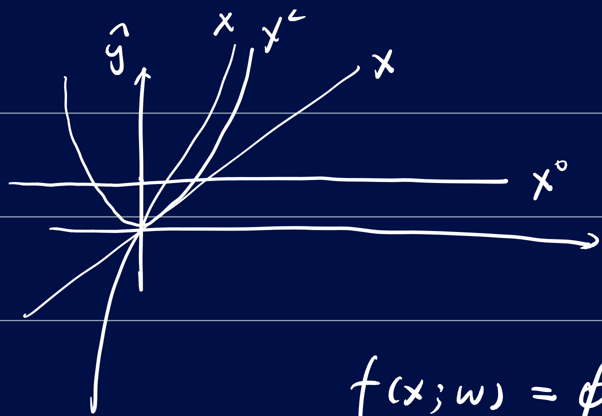
Scalar-valued features x_i:

$$f(x_i; w) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M$$

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^M \\ 1 & x_2 & x_2^2 & \dots & x_2^M \\ & & \vdots & & \\ 1 & x_N & x_N^2 & \dots & x_N^M \end{bmatrix} \qquad \underline{Xw}$$

$x \downarrow$ feature transformation $\phi(x) = [1 \; x \; x^2 \dots x^M]^T$

$\underline{[1 \;\; x \;\; x^2 \;\; \dots \;\; x^M]^T}$ $\qquad \phi_j(x) = x^j$ basis function



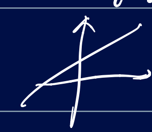
$$[1 \ x \ x^2 \ \dots \ x^M] \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}$$

$$f(x; w) = \phi^T(x)\, w \quad = \underbrace{w^T \phi(x)}$$

$$\left(a^T b = b^T a \quad (a, b \text{ vectors}) \right)$$

$$f(x_i; w) = w^T \phi(x_i)$$

$M = 1$



scalar-valued $\quad \underbrace{\phi(x_i)} = [1 \ x_i]^T$

$$\rightarrow \quad \underbrace{\phi(x_i)} = [1 \ x_i \ x_i^2 \dots x_i^M]^T$$

vector-valued $\quad \phi(x_i) = [1 \ x_{i1} \ x_{i2} \ \dots \ x_{iD}]^T$

$x_i = [x_{i1} \ x_{i2} \ \dots \ x_{iD}]^T$

$$f(x_i; w) = w^T \phi(x_i) \qquad \underbrace{M = 2}$$

$x_i = [x_{i1} \ x_{i2}]^T$

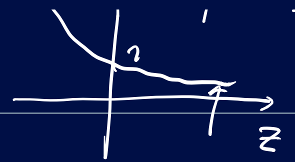
$$f(x_i; w) = w_{00} + w_{10} x_{i1} + w_{01} x_{i2}$$
$$+ w_{20} x_{i1}^2 + w_{02} x_{i2}^2 + w_{11} x_{i1} x_{i2}$$

$$= \sum_j \sum_k w_{jk} x_{i1}^j x_{i2}^k$$

$(j + k \leq 2)$

$\backslash \quad + \exp(-z)$

$$\phi_j(x_i) = \exp\left(-\frac{\|x_i - x_j\|^2}{2s^2}\right)$$



Now that we have data, models, how do we evaluate if a model is good or bad?

loss functions $\quad J(w) = \frac{1}{N}\|Y - \hat{Y}\|^2$ (MSE)

$$= \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2$$

For linear models $\quad \hat{y} = Xw$

$$X = \begin{bmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_N) \end{bmatrix}$$

(MAE) $\quad J(w) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

MAE of 0.7 lbs

MSE of 0.3 lbs^2 $\qquad$ $RMSE = \sqrt{MSE}$

$$w = (X^TX)^{-1}X^TY$$

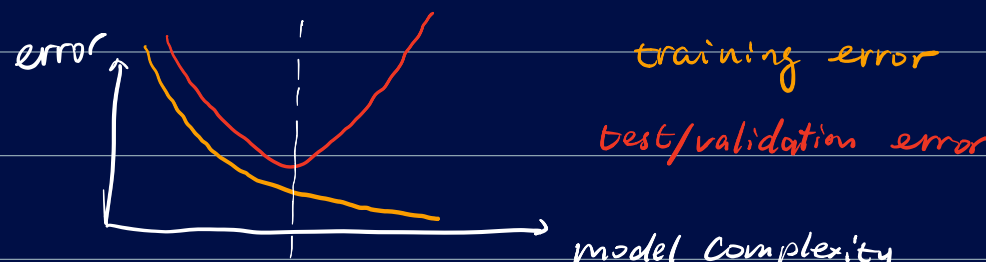
Overfitting and Generalization

- learn patterns/trends from the data to predict the labels for unseen features
- We say a model generalizes well if it has a good performance for unseen data.
- if a model performs well on the training set but poorly on unseen data → overfitting

→ How do we alleviate overfitting?

- Reduce M (Polynomials)
- weight-based regularization

⇒ limit the complexity of the model



- Hyper-parameters
  - M order of the polynomial
  - λ regularization coefficient

$$J(w) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} \| y_i - \underbrace{w^T\phi(x_i)}_{f(x_i;w)} \|^2}_{\text{MSE}} + \underbrace{\lambda \|w\|_2^2}_{\text{regularization term}}$$

$$\lambda \|w\|^2 = \lambda\left(\sqrt{w_0^2 + w_1^2 + \dots + w_M^2}\right)^2$$

$$= \lambda(w_0^2 + w_1^2 + \dots w_M^2)$$

$$= \lambda \sum_{j=0}^{M} w_j^2$$

Norm-based regularization

L2 - norm $\quad \|w\|_2 = \sqrt{w_0^2 + w_1^2 + \dots + w_M^2}$

L1 - norm $\quad \|w\|_1 = |w_0| + |w_1| + \dots + |w_M|$

$\lambda \|w\|_2^2 \longrightarrow$ Ridge regularization

$\lambda \|w\|_1 \longrightarrow$ Lasso regularization

- Ridge regularization $\quad \lambda = 0 \quad \lambda \uparrow \quad \|w^*\| \downarrow$

$w^* = (X_{train}^T X_{train} + \lambda I)^{-1} X_{train} Y_{train}$

- Lasso regularization $\quad \lambda \uparrow \rightarrow$ more components of w become 0

How would it be useful?

$$f(x; w) = w_0 + w_0 x + w_0 x^2 + \cdots + w_M x^M$$

Ex x_i: age, height, eye color (with x_{i1} above age, x_{i2} above height, x_{i3} above eye color)

y_i: weight

$$\hat{y}_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3}$$

Lasso $\quad w_3 \rightarrow 0$

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Classification Vs. Regression

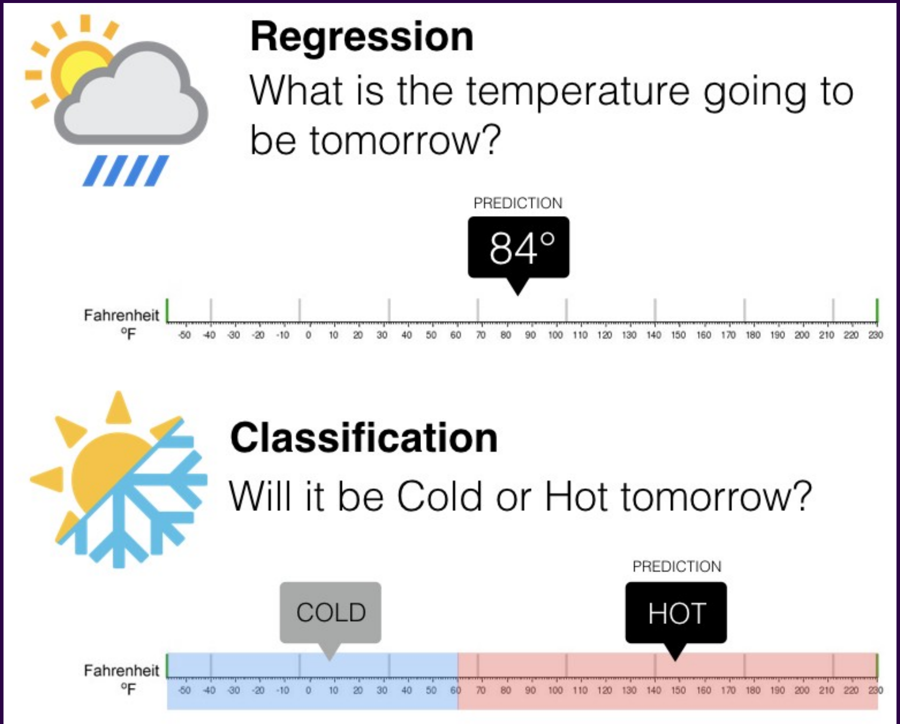
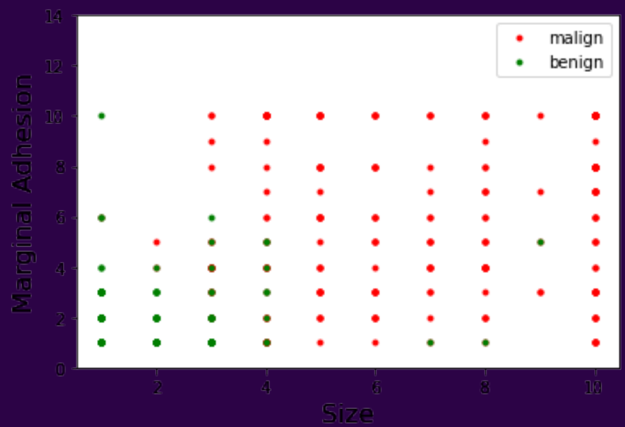
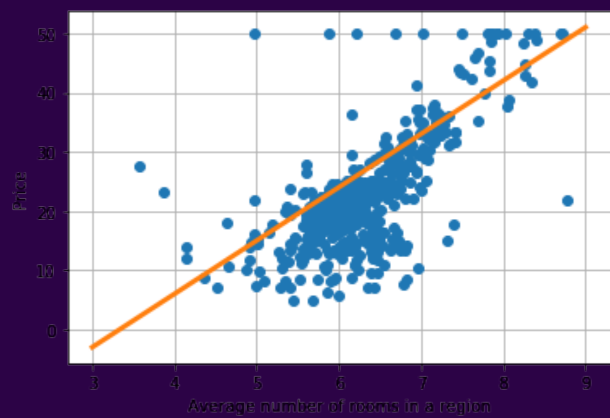


Figure: https://www.pinterest.com/pin/672232681855858622/?lp=true

Classification Vs. Regression



(a) Breast cancer dataset

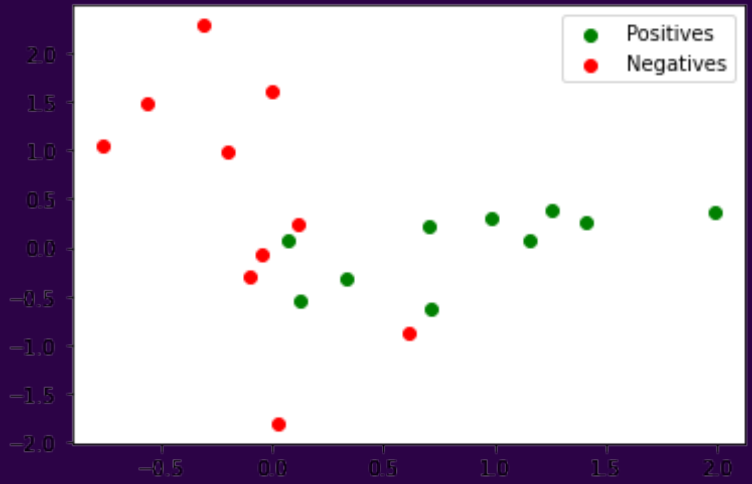


(b) Boston Housing dataset

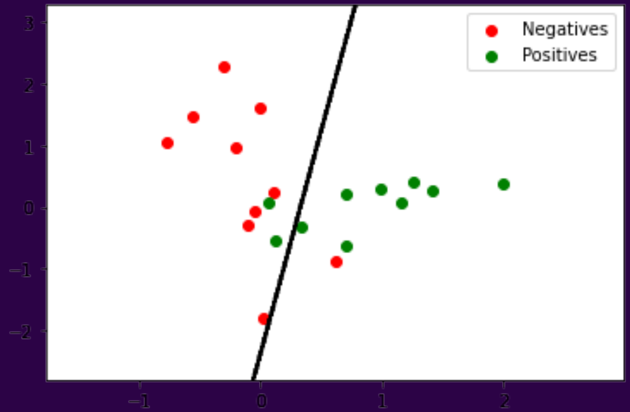
Classification

Given the dataset (x_i, y_i) for $i = 1, 2, \ldots, N$, find a function $f(x)$ (model) so that it can predict the label $\hat{y}$ for some input x, even if it is not in the dataset, i.e. $\hat{y} = f(x)$.

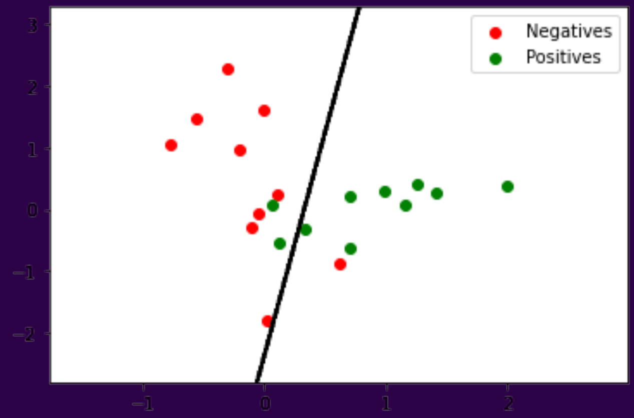
- Positive : $y = 1$
- Negative : $y = 0$



Decision Boundary



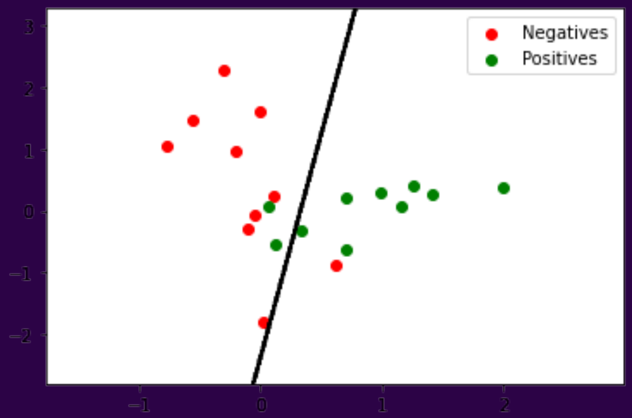
Decision Boundary



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}}$$

- What is the accuracy in this example ?



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}} = \frac{17}{20} = 0.85 = 85\%$$

Need for a new model

- What would happen if we used the linear regression model :

$$\hat{y} = w_0 + w_1 x$$

Need for a new model

- What would happen if we used the linear regression model :

$$\hat{y} = w_0 + w_1 x$$

- y is 0 or 1
- $\hat{y}$ will take any value between $-\infty$ and ∞

Need for a new model

- What would happen if we used the linear regression model :

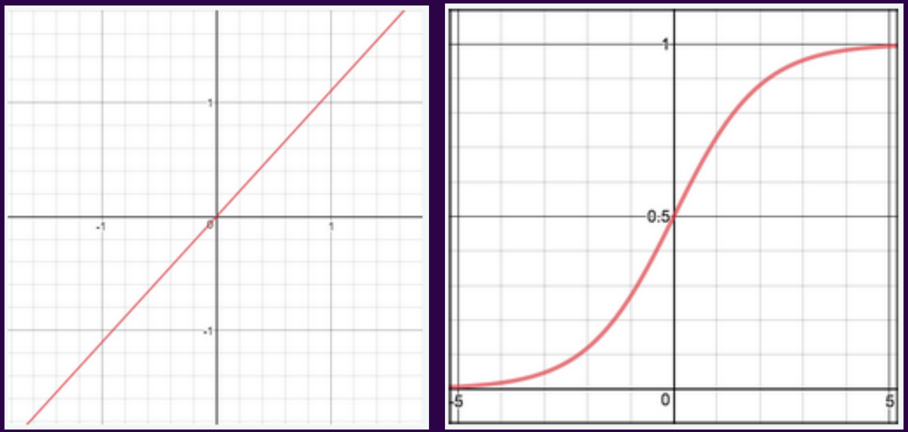
$$\hat{y} = w_0 + w_1 x$$

- y is 0 or 1
- $\hat{y}$ will take any value between $-\infty$ and ∞
- It will be hard to find w_0 and w_1 that make the prediction $\hat{y}$ match the label y.

Sigmoid Function

- By applying the sigmoid function, we enforce $0 \leq \hat{y} \leq 1$

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$



(a) Linear model (b) Sigmoid model

A new loss function

- Binary cross entropy loss :

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$

pause

- What happens if $y_i = 0$:
  $\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = ?$

A new loss function

- Binary cross entropy loss :

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$

- If $y_i = 0$:

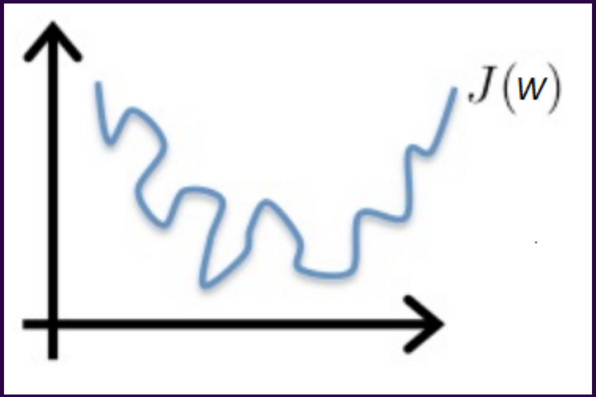
$$\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(1-\hat{y}_i)$$

A new loss function

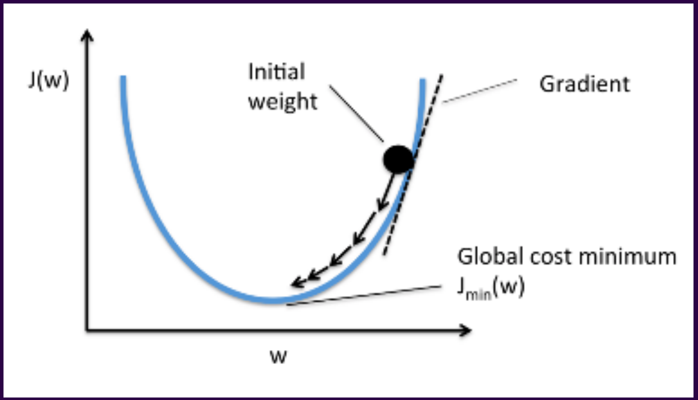
- Binary cross entropy loss :
$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$
- If $y_i = 0$:
$$\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(1-\hat{y}_i)$$
- If $y_i = 1$:
$$\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(\hat{y}_i)$$

MSE vs Binary cross entropy loss

- MSE of a logistic function has many local minima.
- The Binary cross entropy loss has only one minimum.



(a) MSE



(b) Binary cross entropy loss

Classifier

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

- How to deal with uncertainty ?
  - Thanks to the sigmoid, $\hat{y} = f(x)$ is between 0 and 1.

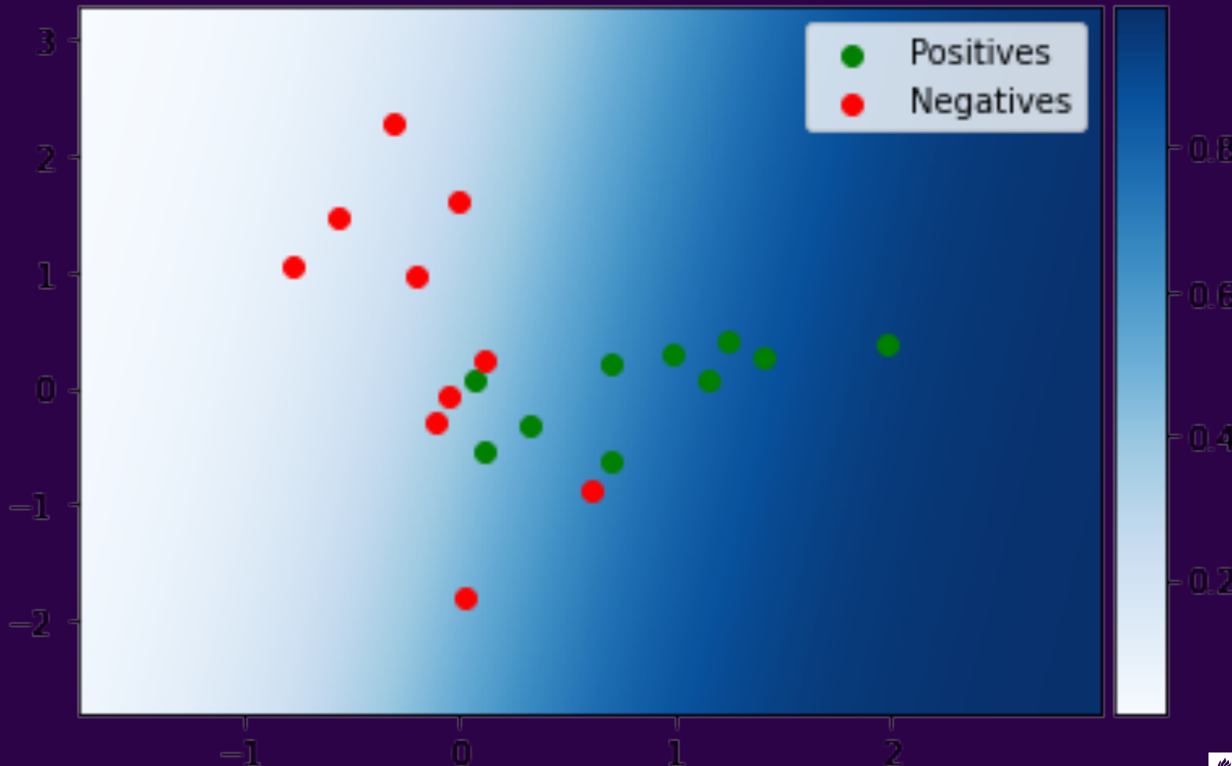
NYU TANDON SCHOOL OF ENGINEERING

Classifier

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

- How to deal with uncertainty ?
  - Thanks to the sigmoid, $\hat{y} = f(x)$ is between 0 and 1.
- If $\hat{y}$ is close to 0, the data is probably negative
- If $\hat{y}$ is close to 1, the data is probably positive
- If $\hat{y}$ is around 0.5, we are not sure.

Classifier



Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.

Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.
- Let $0 < t < 1$ be a Threshold :
  - If $\hat{y} > t$, $\hat{y}$ is classified as positive.
  - If $\hat{y} < t$, $\hat{y}$ is classified as negative.

Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.
- Let $0 < t < 1$ be a Threshold :
  - If $\hat{y} > t$, $\hat{y}$ is classified as positive.
  - If $\hat{y} < t$, $\hat{y}$ is classified as negative.
- How to choose t ?

Impact of the threshold

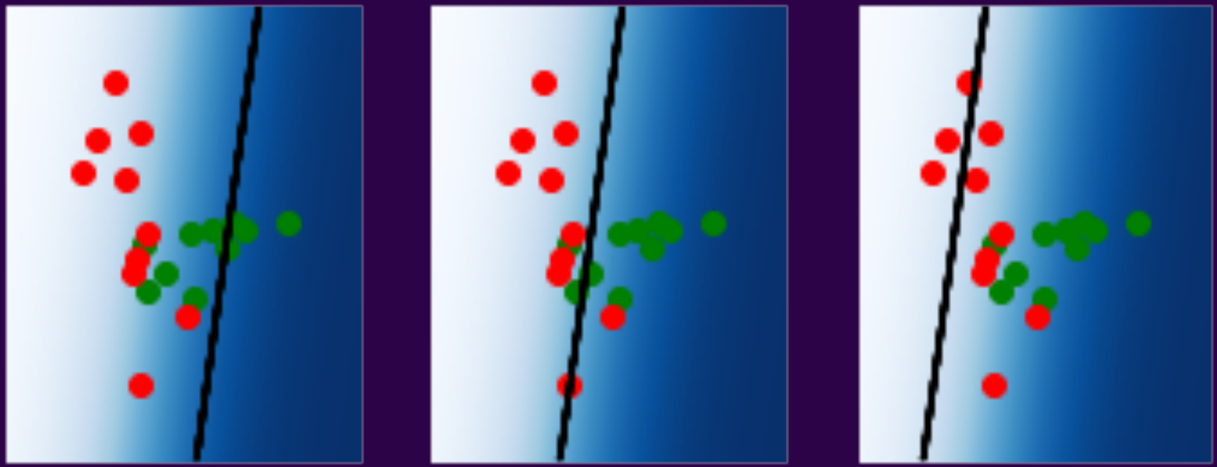


Figure: $t = 0.2, 0.5, 0.8$

Performance metrics for a classifier

- Accuracy of a classifier: percentage of correct classification
- Why accuracy alone is not a good measure for assessing the model ?

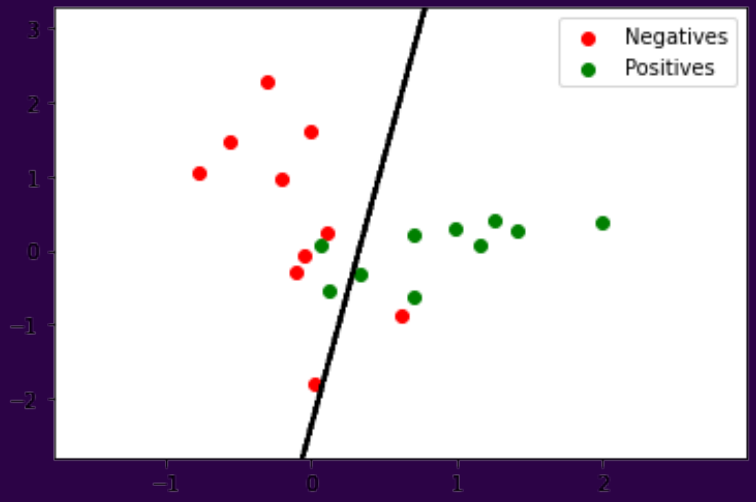
Performance metrics for a classifier

- Accuracy of a classifier: percentage of correct classification
- Why accuracy alone is not a good measure for assessing the model ?
  - Example: A rare disease occurs 1 in ten thousand people
  - A test that classifies everyone as free of the disease can achieve 99.999% accuracy when tested with people drawn randomly from the entire population

Types of Errors in Classification

- Correct predictions:
  - True Positive (TP) : Predict $\hat{y} = 1$ when $y = 1$
  - True Negative (TN) : Predict $\hat{y} = 0$ when $y = 0$
- Two types of errors:
  - False Positive/ False Alarm (FP): $\hat{y} = 1$ when $y = 0$
  - False Negative/ Missed Detection (FN): $\hat{y} = 0$ when $y = 1$

Example



- How many True Positive (TP) are there ?
- How many True Negative (TN) are there ?
- How many False Positive (FP) are there ?
- How many False Negative (FN) are there ?

Other metrics

- Sensitivity/Recall/TPR (How many positives are detected among all positive?)

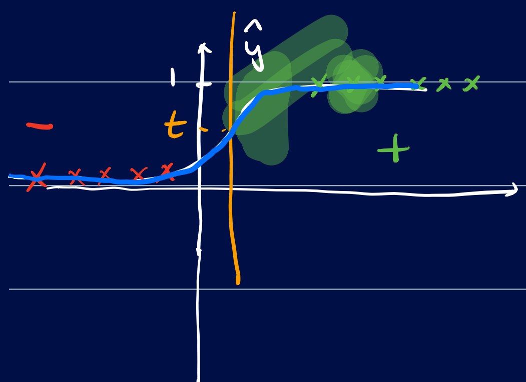
$$\frac{\text{TP}}{\text{TP} + \text{FN}}$$

- Precision (How many detected positives are actually positive?)

$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

Logistic Regression

- $\sigma(z) = \dfrac{1}{1+e^{-z}}$
- model $\quad \sigma(W^T \underline{\phi(x)})$
  - scalar-valued $\quad \sigma(w_0 + w_1 x)$

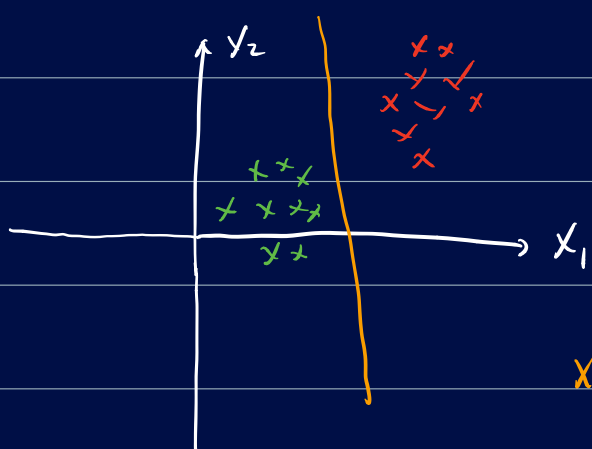


decision boundary

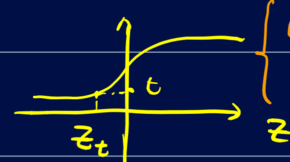
decision rule

$$\begin{cases} 1 & \text{if } \sigma(w_0 + w_1 x) > t \\ 0 & \text{if } \sigma(w_0 + w_1 x) \leq t \end{cases}$$

- vector-valued feature $\quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$



$t = \sigma(w_0 + w_1 x_1 + w_2 x_2)$



$\left\{ w_0 + w_1 x_1 + w_2 x_2 = z_t \right\}$

$$x_2 = \frac{1}{w_2}(z_t - w_1 x_1 - w_0)$$

$$x_2 = \frac{z_t}{w_2} - \frac{w_0}{w_2} \boxed{-\frac{w_1}{w_2}} x_1$$

$$= g(x_1)$$

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Lab: Diagnosing Breast Cancer

- We're going to use the breast cancer dataset to predict whether the patients' scans show a malignant tumour or a benign tumour.
- Let's try to find the best linear classifier using logistic regression.

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
    - One-hot / 1-of-K vectors, ex : 4 Class
    - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
    - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
    - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
    - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
  - One-hot / 1-of-K vectors, ex : 4 Class
  - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
  - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
  - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
  - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$
- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(W^T \phi(\mathbf{x}))$
- Shape of $W^T \phi(\mathbf{x})$: $(K, 1) = (K, D) \times (D, 1)$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{z_k}}{\sum_j e^{z_j}}$

Multiclass Classificaiton

- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(\mathbf{z})$ with $\mathbf{z} = W^T\phi(\mathbf{x})$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{\mathbf{z}_k}}{\sum_j e^{\mathbf{z}_j}}$
- Softmax example: If $\mathbf{z} = \begin{bmatrix} -1 \\ 2 \\ 1 \\ -4 \end{bmatrix}$ then,

$$\text{softmax}(z) = \begin{bmatrix} \frac{e^{-1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{2}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{-4}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \end{bmatrix} \approx \begin{bmatrix} 0.035 \\ 0.704 \\ 0.259 \\ 0.002 \end{bmatrix}$$

Cross-entropy

- Multiple outputs: $\hat{\mathbf{y}}_\mathbf{i} = \text{softmax}(W^T \phi(\mathbf{x}_i))$
- Cross-Entropy: $J(W) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik})$
- Example : $\text{K} = 4$

$$\text{If, } \mathbf{y}_i = [0,0,1,0] \text{ then, } \sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik}) = log(\hat{\mathbf{y}}_{i3})$$

NYU TANDON SCHOOL OF ENGINEERING

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Lab: Iris Dataset

- Open demo_iris.ipynb