Logistic Regression

COEN140

Santa Clara University

Classification Problem

- Purpose: to classify the input to a set of discrete classes
- Binary classification: 2 classes

- Estimate the probability that an input belongs to a particular class
  - e.g What is the probability that this email is spam?
  - If the estimated probability is greater than 50%
    - Label it as class “1”: C_1
  - If the estimated probability is less than 50%
    - Label it as class “0”: C_0

Binary Classification

- Given a set of training samples $\mathbf{x}_n \in \mathbb{R}^D$ and the corresponding target labels $t_n, n = 1, 2, \ldots, N$.
- $t_n \in \{0,1\}$
  - $t_n = 1$: $\mathbf{x}_n$ belongs to class-1
  - $t_n = 0$: $\mathbf{x}_n$ belongs to class-0

Logistic Regression: a classification method

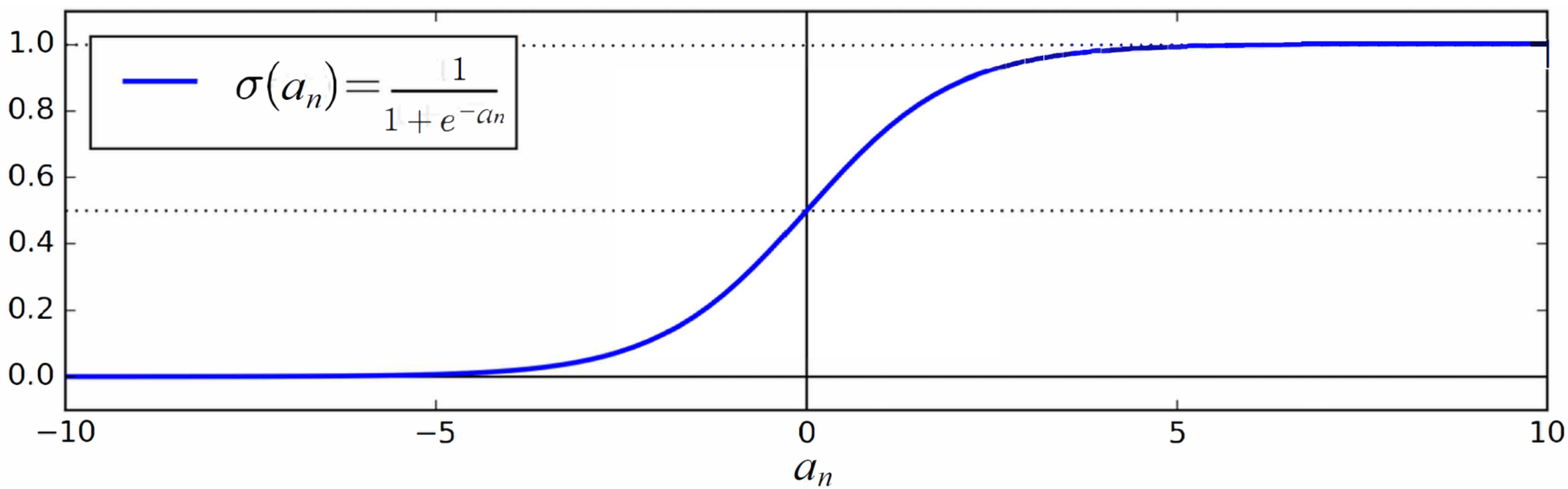
- Estimate Probabilities
- Compute a weighted sum of the input features
  - $\mathbf{w} = [w_1, \dots, w_D]^T$
  - $\mathbf{x}_n = [x_{n1}, x_{n2}, \dots, x_{nD}]^T$
  - $a_n = \mathbf{w}^T \mathbf{x}_n$
- Output the logistic of the result
  - $y_n = \sigma(a_n) = \sigma\left(\mathbf{w}^T \mathbf{x}_n\right)$

Logistic Regression

- Estimate Probabilities
- If we include a bias (intercept): b (same as w_0)
  - $\mathbf{w} = [b, w_1, \dots, w_D]^T$
  - $\mathbf{x}_n = [1, x_{n1}, x_{n2}, \dots, x_{nD}]^T$
  - $a_n = \mathbf{w}^T \mathbf{x}_n$
- Output the logistic of the result
  - $y_n = \sigma(a_n) = \sigma(\mathbf{w}^T \mathbf{x}_n)$

Logistic Regression

- Output the logistic of the result
  - $y_n = \sigma(a_n) = \sigma\left(\mathbf{w}^T\mathbf{x}_n\right) = P(C_1|\mathbf{x}_n)$
  - $1 - y_n = 1 - \sigma(a_n) = 1 - \sigma\left(\mathbf{w}^T\mathbf{x}_n\right) = P(C_0|\mathbf{x}_n)$
- Sigmoid: $\sigma(a_n) = \frac{1}{1+\exp(-a_n)}$, represents a probability

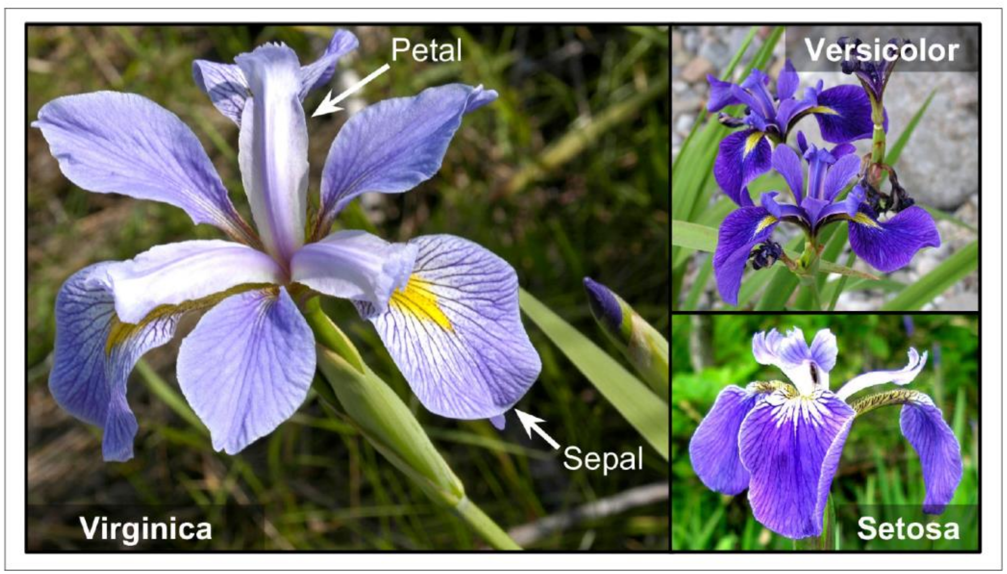


Make Decisions

- Decision Criterion
  - $y_n = \sigma\left(\mathbf{w}^T \mathbf{x}_n\right) \underset{C_0}{\overset{C_1}{\gtrless}} 0.5$
  - 0.5: a threshold
- model parameter: $\mathbf{w}$

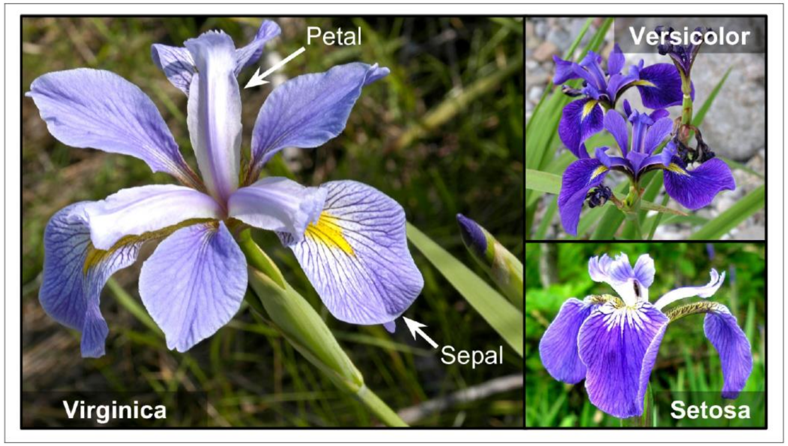
Example: Binary Classification

- Iris Plant
  - Virginica=1, Non-Virginica=0
- 1 feature
  - x_n=Petal width



Example: Binary Classification

- $\mathbf{x}_n = [1, x_n] = [1,\ 1.8]$
- $\mathbf{w} = [b, w_1] = [\text{-}6.58,\ 3.96]$
- $a_n = \mathbf{w}^T\mathbf{x}_n = -6.58 \times 1 + 3.96 \times 1.8 = 0.55$
- $y_n = \sigma(a_n) = \frac{1}{1+\exp(-0.55)} = 0.63 = P(C_1|\mathbf{x}_n) > 0.5$



Hence, $\mathbf{x}_n$ is classified as Virginica!

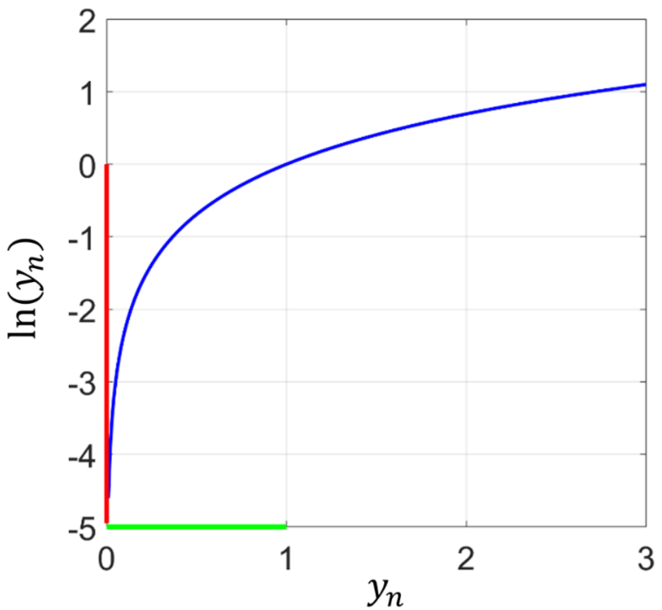
Make Decisions

- Decision Criterion
  - $y_n = \sigma(\mathbf{w}^T \mathbf{x}_n) \underset{C_0}{\overset{C_1}{\gtrless}} 0.5$
  - 0.5: a threshold
- How to find $\mathbf{w}$?
  - This is the model parameter

Binary Classification

- Cross-entropy error function
- For the n-th training sample
  - $E_n(\mathbf{w}) = -[t_n \ln y_n + (1 - t_n)\ln(1 - y_n)]$
  - $t_n \in \{0,1\}$ is the class label for $\mathbf{x}_n$
  - $y_n = \sigma(\mathbf{w}^T \mathbf{x}_n)$: the predicted class-1 probability for $\mathbf{x}_n$
  - If $t_n = 1$, then we want $y_n \to 1$
  - If $t_n = 0$, then we want $y_n \to 0$

$y_n \in (0,1]$



Binary Classification

- Cross-entropy error function
- For the n-th training sample
  - $E_n(\mathbf{w}) = -[t_n \ln y_n + (1 - t_n)\ln(1 - y_n)]$
- For all training samples
  - $E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$

How to find **w**? Minimize the error function

- For one data sample
- $E_n(\mathbf{w}) = -[t_n \ln y_n + (1 - t_n)\ln(1 - y_n)]$
  - $y_n = \sigma(a_n), a_n = \mathbf{w}^T \mathbf{x}_n$
- $\frac{dy_n}{da_n} = y_n(1 - y_n), \frac{da_n}{d\mathbf{w}} = \mathbf{x}_n$
- $\frac{dE_n}{dy_n} = -[t_n \times \frac{1}{y_n} + (1 - t_n) \times \frac{1}{1-y_n} \times (-1)]$
- $= -t_n \times \frac{1}{y_n} + (1 - t_n) \times \frac{1}{1-y_n}$
- $\nabla_{\mathbf{w}} E_n(\mathbf{w}) = \frac{dE_n}{dy_n} \times \frac{dy_n}{da_n} \times \frac{da_n}{d\mathbf{w}} = (y_n - t_n)\mathbf{x}_n$

How to find $\mathbf{w}$? Minimize the error function

- For all data samples
- $E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$
  $= -\sum_{n=1}^{N}[t_n \ln y_n + (1 - t_n)\ln(1 - y_n)]$
  - $\nabla_{\mathbf{w}} E_n(\mathbf{w}) = (y_n - t_n)\mathbf{x}_n$
  - $\nabla_{\mathbf{w}} E(\mathbf{w}) = \nabla_{\mathbf{w}} \sum_{n=1}^{N} E_n(\mathbf{w}) = \sum_{n=1}^{N} \nabla_{\mathbf{w}} E_n(\mathbf{w})$
  - $\nabla_{\mathbf{w}} E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\mathbf{x}_n$
- Gradient Descent
- $\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta \nabla_{\mathbf{w}} E\left(\mathbf{w}^{(\tau-1)}\right) = \mathbf{w}^{(\tau-1)} - \eta \sum_{n=1}^{N}(y_n - t_n)\mathbf{x}_n$

Make Decisions

- Let $\mathbf{x}_n$ be a test sample
- Assume $\mathbf{w}$ is already calculated
- Decision Criterion
    - $y_n = \sigma\left(\mathbf{w}^T \mathbf{x}_n\right) \underset{C_0}{\overset{C_1}{\gtrless}} 0.5$

Multi-Class Classification

- Given a set of training samples $\mathbf{x}_n \in \mathbb{R}^D$ and the corresponding target vectors $\mathbf{t}_n, n = 1, 2, \ldots, N$.
- K-class classification problem:
  - $\mathbf{t}_n \in \{0,1\}^K$
    - This is called: 1-of-K coding
    - $t_{nk} = 1$: $\mathbf{x}_n$ belongs to class-k
    - $t_{nk} = 0$: $\mathbf{x}_n$ does not belong to class-k

    - Example: $K = 5$, 5 classes
    - $\mathbf{t}_n = [0,0,0,1,0]^T$: $\mathbf{x}_n$ belongs to class-4

Model Parameters

- Need K weight vectors
- $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$
- $\mathbf{x}_n$: the n-th training sample
- $a_{nk} = \mathbf{w}_k^T \mathbf{x}_n, k = 1, 2, \ldots, K$

- Softmax function:
- $y_{nk} = \frac{\exp(a_{nk})}{\sum_{j=1}^{K} \exp(a_{nj})} = P(C_k | \mathbf{x}_n), k = 1, 2, \ldots, K$
- Outputs the probability of $\mathbf{x}_n$ belonging to the k-th class

Softmax Function

- $a_{nk} = \mathbf{w}_k^T \mathbf{x}_n, k = 1, 2, \dots, K$

- Property: all the probabilities sum up to 1

- $\sum_{k=1}^{K} y_{nk} = \sum_{k=1}^{K} P(C_k|\mathbf{x}_n) = \sum_{k=1}^{K} \frac{\exp(a_{nk})}{\sum_{j=1}^{K} \exp(a_{nj})}$

$$= \frac{\exp(a_{n1})}{\sum_{j=1}^{K} \exp(a_{nj})} + \frac{\exp(a_{n2})}{\sum_{j=1}^{K} \exp(a_{nj})} + \cdots + \frac{\exp(a_{nK})}{\sum_{j=1}^{K} \exp(a_{nj})}$$

$$= 1$$

Make Decisions

- Let $\mathbf{x}_n$ be a test sample
- Assume $\mathbf{w}_k, k = 1, \dots, K$ are already calculated
  - Compute $a_{nk} = \mathbf{w}_k^T \mathbf{x}_n, k = 1, 2, \dots, K$
  - Compute probabilities $y_{nk} = \frac{\exp(a_{nk})}{\sum_{j=1}^{K} \exp(a_{nj})}, k = 1, 2, \dots, K$

- Decision Criterion
- The predicted class label for $\mathbf{x}_n$ is

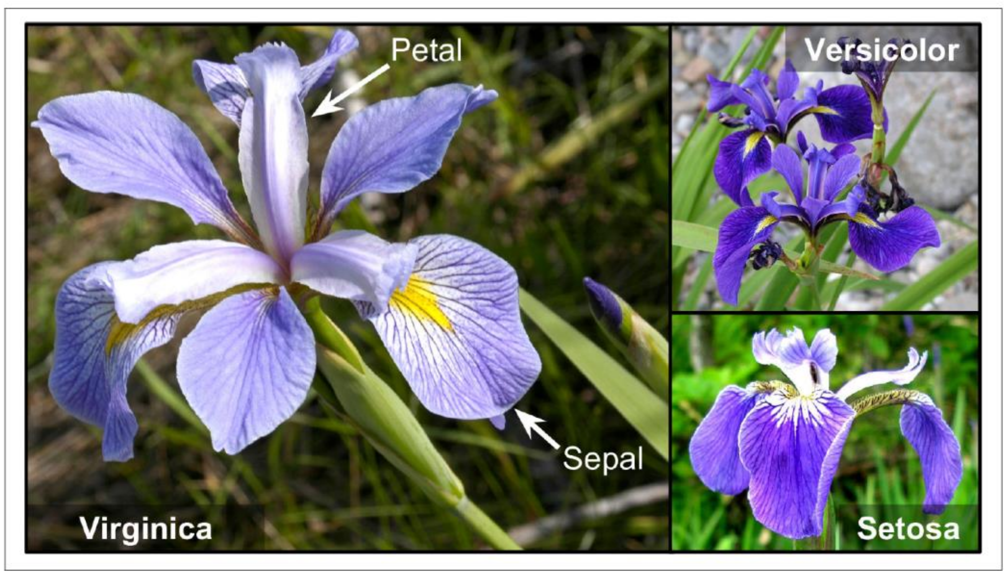
$$\hat{k}_n = \arg\max_k y_{nk}$$

- Or, equivalently,

$$\hat{k}_n = \arg\max_k a_{nk}$$

Example: $K = 3$

- Three Iris Plant Species
  - Setosa=0 , Versicolor=1, Virginica=2
- Four features
  - Sepal length, Sepal width, Petal length, Petal width
  - $\mathbf{x}_n = [x_{n1}, x_{n2}, x_{n3}, x_{n4}]^T$



Example: $K = 3$

- Calculated parameters (coefficients)
  - $\mathbf{w}_0 = [\text{-}0.53,\ 0.83, \text{-}2.34, \text{-}1.00]$
  - $\mathbf{w}_1 = [0.53\ \text{-}0.31\ \text{-}0.17\ \text{-}0.85]$
  - $\mathbf{w}_2 = [0.00, \text{-}0.52,\ 2.52\ \ 1.85]$
- Bias (Incercept)
- $b_0 = 10.12,$
- $b_1 = 1.81$
- $b_2 = -11.93$

- $\mathbf{x}_n = [4.4,\ \ 3.0,\ \ 1.3,\ \ \ 0.2]^T$
- Probabilities $y_{nk} =?, k = 0, 1, 2$

Example: $K = 3$

- Parameters (coefficients)
  - $\mathbf{w}_0 = [-0.53,\ 0.83, -2.34, -1.00]$
  - $\mathbf{w}_1 = [0.53\ -0.31\ -0.17\ -0.85]$
  - $\mathbf{w}_2 = [0.00, -0.52,\ 2.52\ 1.85]$
- Bias (Incercept)
- $b_0 = 10.12,\ \ b_1 = 1.81,\ \ \ b_2 = -11.93$

- $\mathbf{x}_n = [4.4,\ \ 3.0,\ \ 1.3,\ \ \ 0.2]^T$
- $a_{n0} = b_0 + \mathbf{w}_0^T \mathbf{x}_n$
- $= 10.12 - 0.53 \times 4.4 + 0.83 \times 3.0 - 2.34 \times 1.3 - 1 \times 0.2$
- $= 7.04$

Example: $K = 3$

- Parameters (coefficients)
  - $\mathbf{w}_0 = [-0.53,\ 0.83, -2.34, -1.00]$
  - $\mathbf{w}_1 = [0.53\ -0.31\ -0.17\ -0.85]$
  - $\mathbf{w}_2 = [0.00, -0.52,\ 2.52\ \ 1.85]$
- Bias (Incercept)
- $b_0 = 10.12,\ \ b_1 = 1.81,\ \ \ b_2 = -11.93$

- $\mathbf{x}_n = [4.4,\ \ 3.0,\ \ 1.3,\ \ \ 0.2]^T$
- $a_{n1} = b_1 + \mathbf{w}_1^T \mathbf{x}_n$
- $= 1.81 + 0.53 \times 4.4 - 0.31 \times 3.0 - 0.17 \times 1.3 - 0.85 \times 0.2$
- $= 2.82$

Example: $K = 3$

- Parameters (coefficients)
  - $\mathbf{w}_0 = [-0.53,\ 0.83, -2.34, -1.00]$
  - $\mathbf{w}_1 = [0.53\ -0.31\ -0.17\ -0.85]$
  - $\mathbf{w}_2 = [0.00, -0.52,\ 2.52\ 1.85]$
- Bias (Incercept)
- $b_0 = 10.12,\ \ b_1 = 1.81,\ \ \ b_2 = -11.93$

- $\mathbf{x}_n = [4.4,\ \ 3.0,\ \ 1.3,\ \ \ 0.2]^T$
- $a_{n2} = b_2 + \mathbf{w}_2^T \mathbf{x}_n$
- $= -11.93 + 0.00 \times 4.4 - 0.52 \times 3.0 + 2.52 \times 1.3 + 1.85 \times 0.2$
- $=$ -9.84

Example: $K = 3$

- $a_{n0} = 7.04, a_{n1} = 2.82, a_{n2} = -9.84$

- $y_{n0} = \frac{\exp(a_{n0})}{\sum_{j=1}^{3} \exp(a_{nj})} = 0.986$
- $y_{n1} = \frac{\exp(a_{n1})}{\sum_{j=1}^{3} \exp(a_{nj})} = 0.014$
- $y_{n2} = \frac{\exp(a_{n2})}{\sum_{j=1}^{3} \exp(a_{nj})} = 0.000$

- Hence, the predicted label of $\mathbf{x}_n$ is 0.

Cross-entropy error function

- For the n-th training sample
  - $E_n(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{k=1}^{K} t_{nk} \ln y_{nk}$
    - t_{nk} is the one-of-K coding class label for the n-th training sample
    - Only 1 of the K terms is non-zero.
    - y_{nk}: the predicted probability of $\mathbf{x}_n$ belonging to class-k
    - If $t_{nk} = 1$, then we want $y_{nk} \rightarrow 1$

- For all training samples
  - $E(\mathbf{w}_1, \dots, \mathbf{w}_K) = \sum_{n=1}^{N} E_n(\mathbf{w}_1, \dots, \mathbf{w}_K)$

How to find $\mathbf{w}_k$, $k = 1, \dots, K$?

- For the n-th training sample
- $E_n(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{k=1}^{K} t_{nk} \ln y_{nk}$
- $\nabla_{\mathbf{w}_k} E_n(\mathbf{w}_1, \dots, \mathbf{w}_K) = (y_{nk} - t_{nk})\mathbf{x}_n$

- For all training samples
- $\nabla_{\mathbf{w}_k} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nk} - t_{nk})\mathbf{x}_n$

- Gradient Descent
- $\mathbf{w}_k^{(\tau)} = \mathbf{w}_k^{(\tau-1)} - \eta \nabla_{\mathbf{w}_k} E\left(\mathbf{w}_1^{(\tau-1)}, \dots, \mathbf{w}_K^{(\tau-1)}\right)$
- $k = 1, \dots, K$
    - Need to update K weight vectors

How often to update the weights?

- 1. Batch Gradient Descent
  - In each iteration, compute the gradient based on the full training set

$$\begin{aligned}\mathbf{w}^{(\tau)} &= \mathbf{w}^{(\tau-1)} - \eta\nabla_{\mathbf{w}}E\left(\mathbf{w}^{(\tau-1)}\right)\\ &= \mathbf{w}^{(\tau-1)} - \eta\nabla_{\mathbf{w}}\sum_{n=1}^{N}E_n\left(\mathbf{w}^{(\tau-1)}\right)\\ &= \mathbf{w}^{(\tau-1)} - \eta\sum_{n=1}^{N}\nabla_{\mathbf{w}}E_n\left(\mathbf{w}^{(\tau-1)}\right)\end{aligned}$$

How often to update the weights?

- 2. Stochastic Gradient Descent
  - In each iteration, compute the gradient based on only one training sample

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta \nabla_{\mathbf{w}} E_n\left(\mathbf{w}^{(\tau-1)}\right)$$

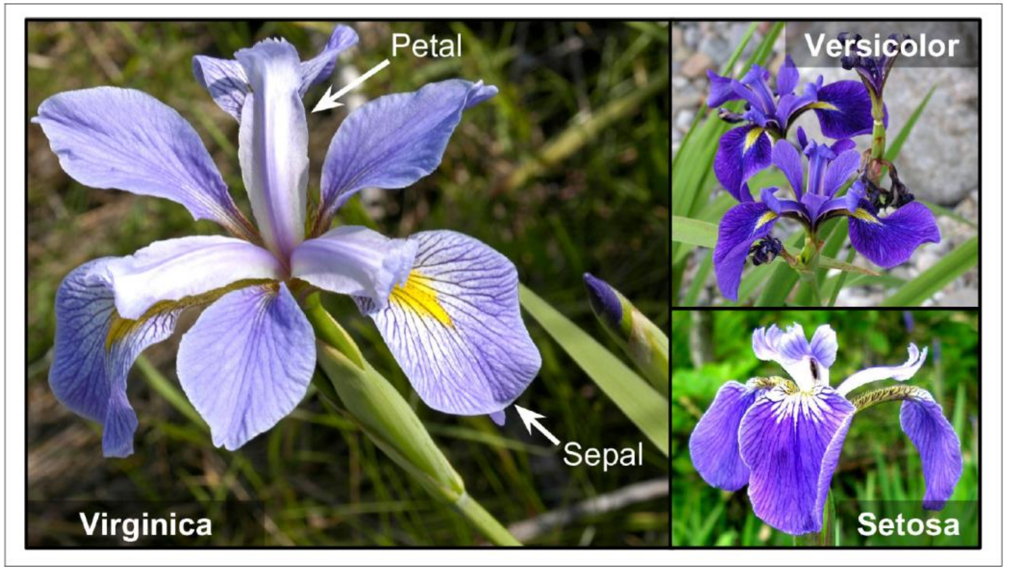
- 3. Mini-batch Gradient Descent
  - In each iteration, compute the gradient based on a small set of training samples
  - $\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta \sum_{n \in \mathcal{N}_i} \nabla_{\mathbf{w}} E_n\left(\mathbf{w}^{(\tau-1)}\right)$
  - $\mathcal{N}_i$: the set of indices for data samples in the i-th mini-batch

Concepts

- For example, I have $N = 10{,}000$ training samples
- Batch size N_b= 200: the number of samples in a mini-batch
- Number of iterations
$$= N/\, N_b \;= 10{,}000/200 \;= 50$$
  - The number of iterations to go through all training samples once
  - Equals to the number of mini-batches

Iris Plant Classification: K=3

- Three Iris Plant Species
  - Setosa=0 , Versicolor=1, Virginica=2
- Four features
  - Sepal length, Sepal width, Petal length, Petal width
  - $\mathbf{x}_n = [x_{n1}, x_{n2}, x_{n3}, x_{n4}]^T$

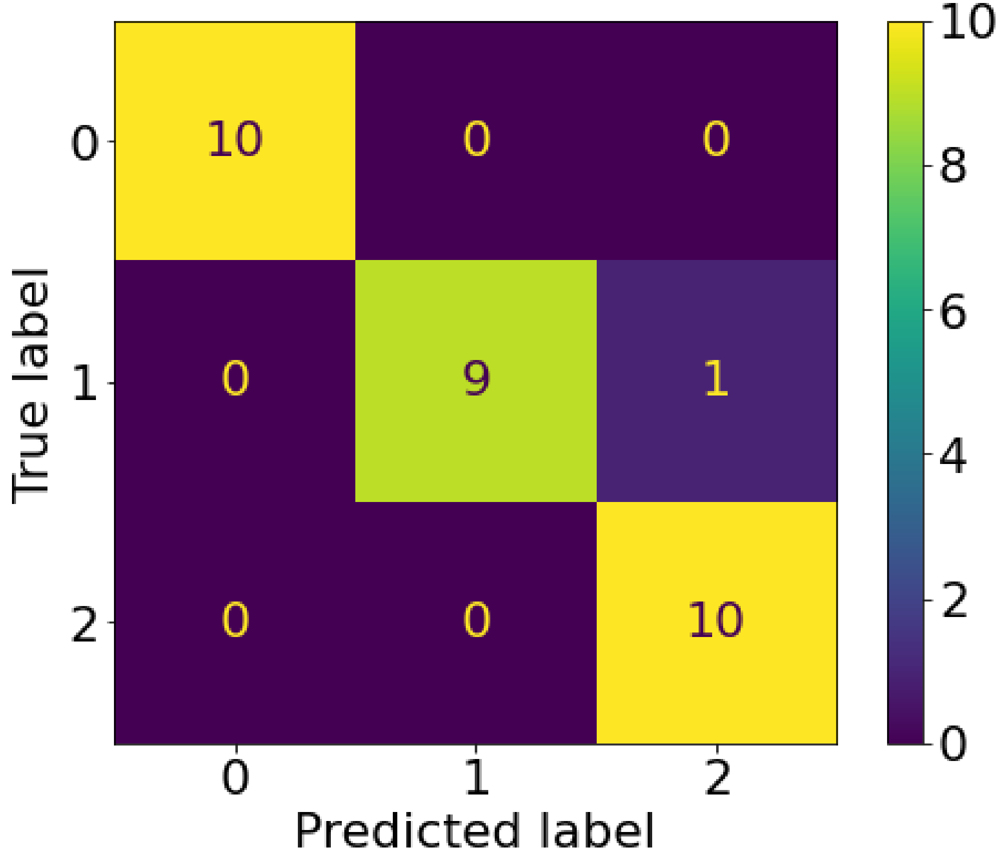


Iris Plant Classification

- 150 data samples
  - Class 0: 50 samples
  - Class 1: 50 samples
  - Class 2: 50 samples
- Split them into Training Set (80%) and Test Set (20%)
  - Training Set:
    - Class 0: 40 samples
    - Class 1: 40 samples
    - Class 2: 40 samples
  - Test Set:
    - Class 0: 10 samples
    - Class 1: 10 samples
    - Class 2: 10 samples

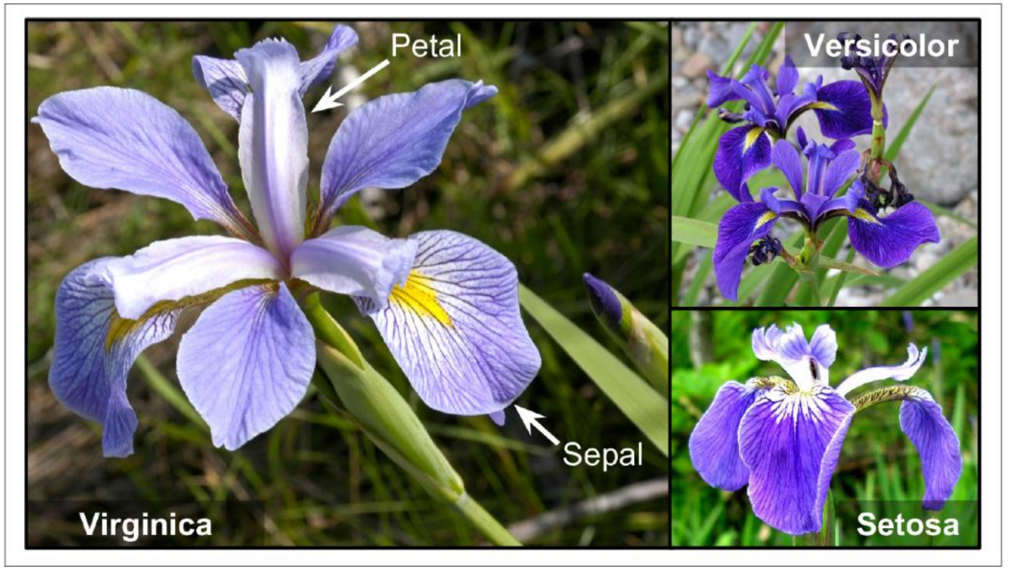
Iris Plant Classification

- Confusion Matrix



Binary Classification

- The same Iris dataset
  - Virginica=1, Non-Virginica=0
- 1 feature
  - Petal width
  - $\mathbf{x}_n = x_{n4}$

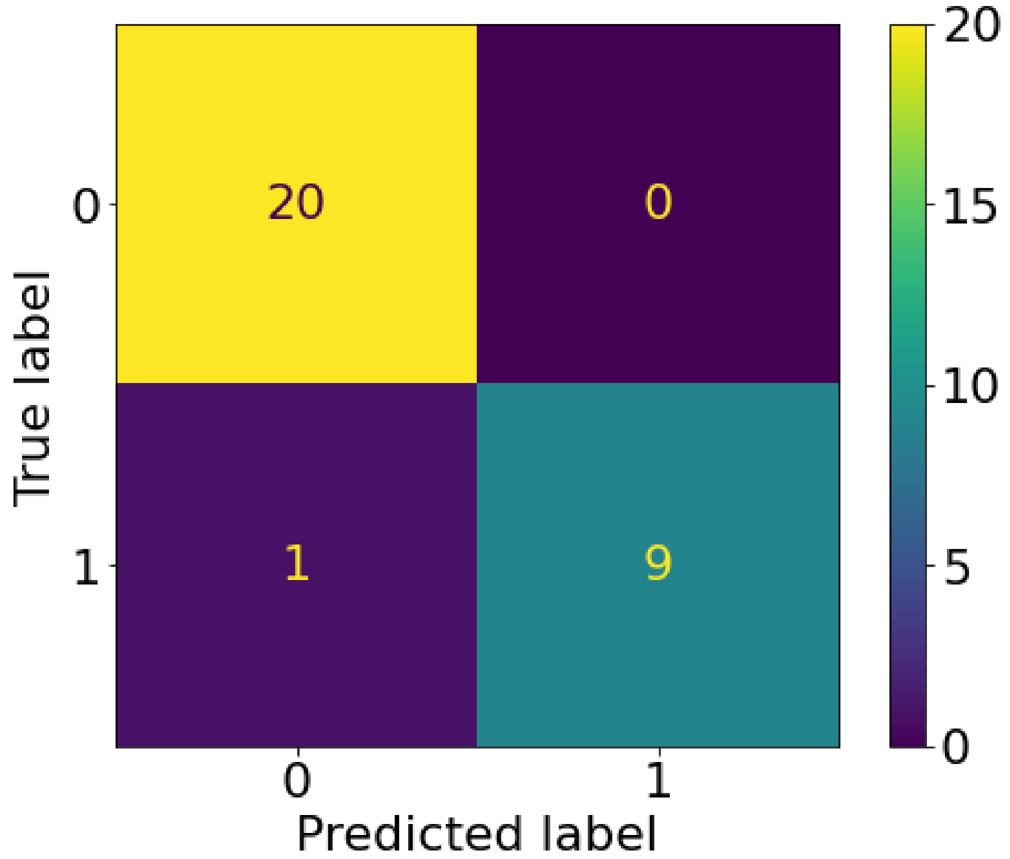


Iris Plant Classification: Binary Case

- 150 data samples
    - Class 0: 100 samples
    - Class 1: 50 samples
- Split them into Training Set (80%) and Test Set (20%)
    - Training Set:
        - Class 0: 80 samples
        - Class 1: 40 samples
    - Test Set:
        - Class 0: 20 samples
        - Class 1: 10 samples

Iris Plant Classification: Binary Case

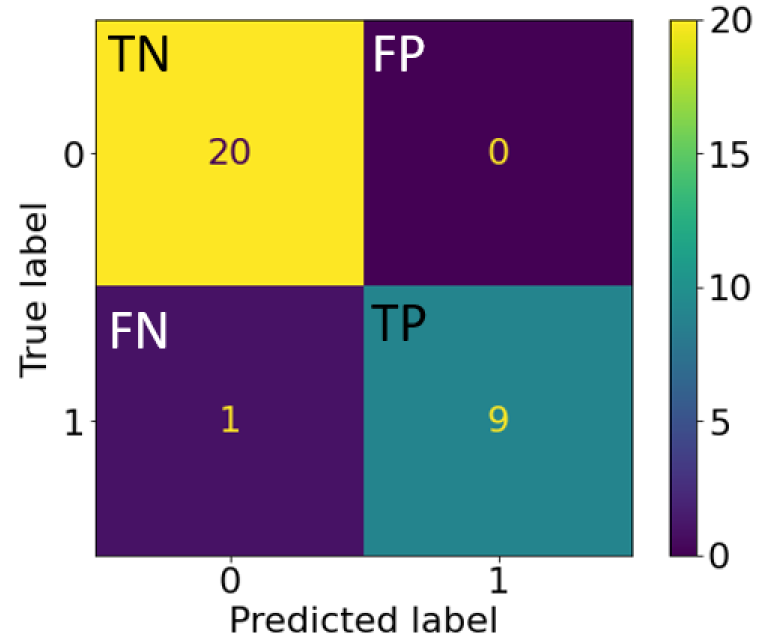
- Confusion Matrix



Binary Classification: Evaluation Metrics

- Class 1: positive class
- Class 0: negative class

- True Positive: TP
- False Positive: FP
- True Negative: TN
- False Negative: FN

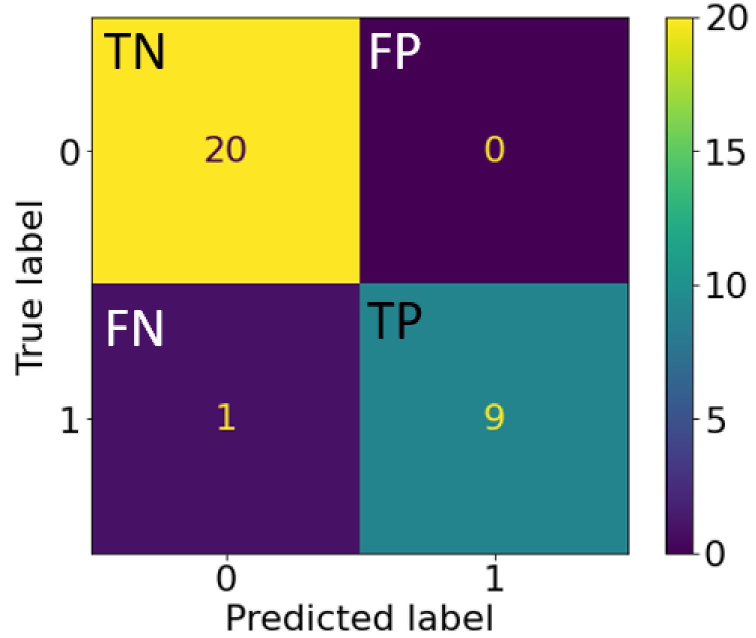


Binary Classification: Evaluation Metrics

- Class 1: positive class
- Class 0: negative class

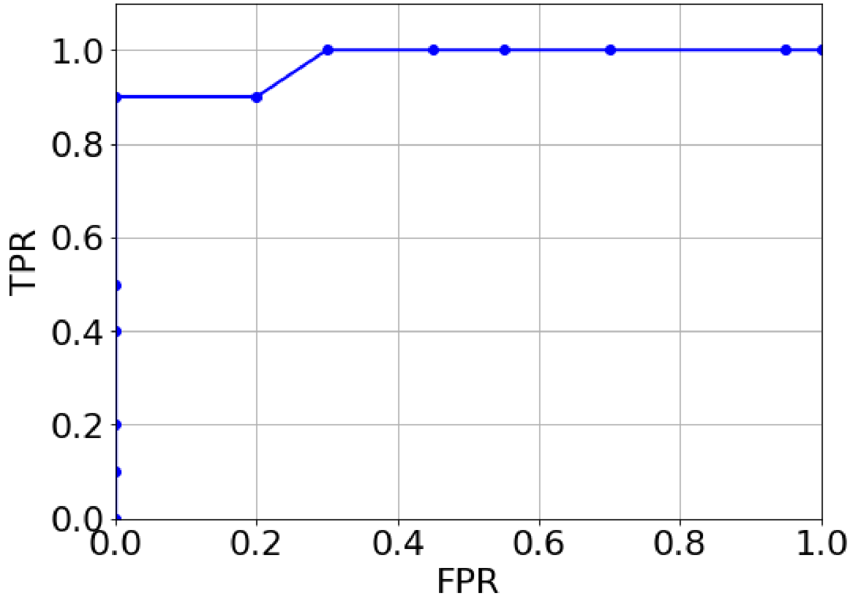
- True Positive Rate
  - $TPR = \frac{TP}{TP+FN}$
- False Positive Rate
  - $FPR = \frac{FP}{FP+TN}$

- We want high TPR, but low FPR



Binary Classification: Evaluation Metrics

- Receiver Operating Characteristic (ROC) Curve
  - Plot the TPR versus the FPR



Decision criterion

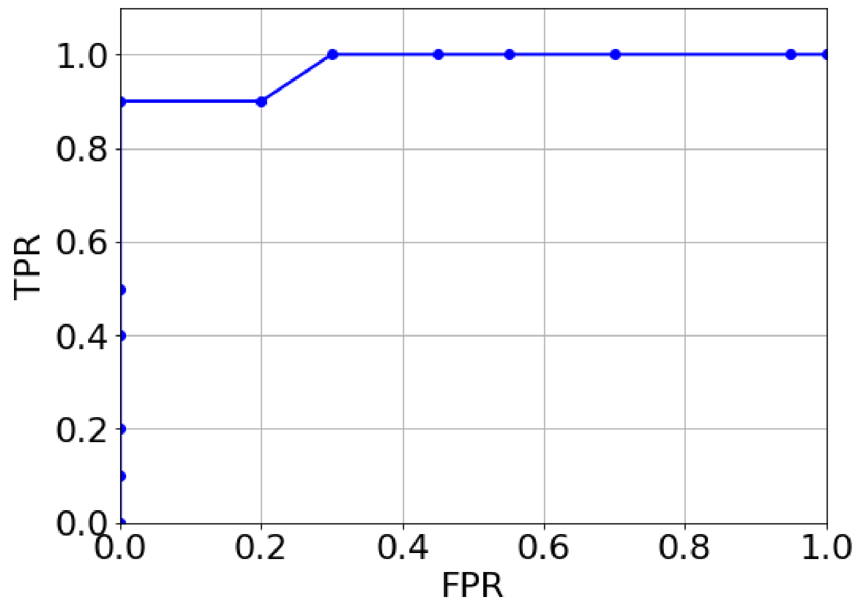
$$y_n = \sigma\left(\mathbf{w}^T \mathbf{x}_n\right) \underset{C_0}{\overset{C_1}{\gtrless}} threshold$$

Previously, $threshold = 0.5$

  - By setting a different $threshold$, you get different TP, TN, FP, FN
    - Hence, you get a different point on the ROC curve

Binary Classification: Evaluation Metrics

- Receiver Operating Characteristic (ROC) Curve
  - Plot the TPR versus the FPR



- AUC: area under the ROC curve
  - The larger the better
  - AUC ≤ 1