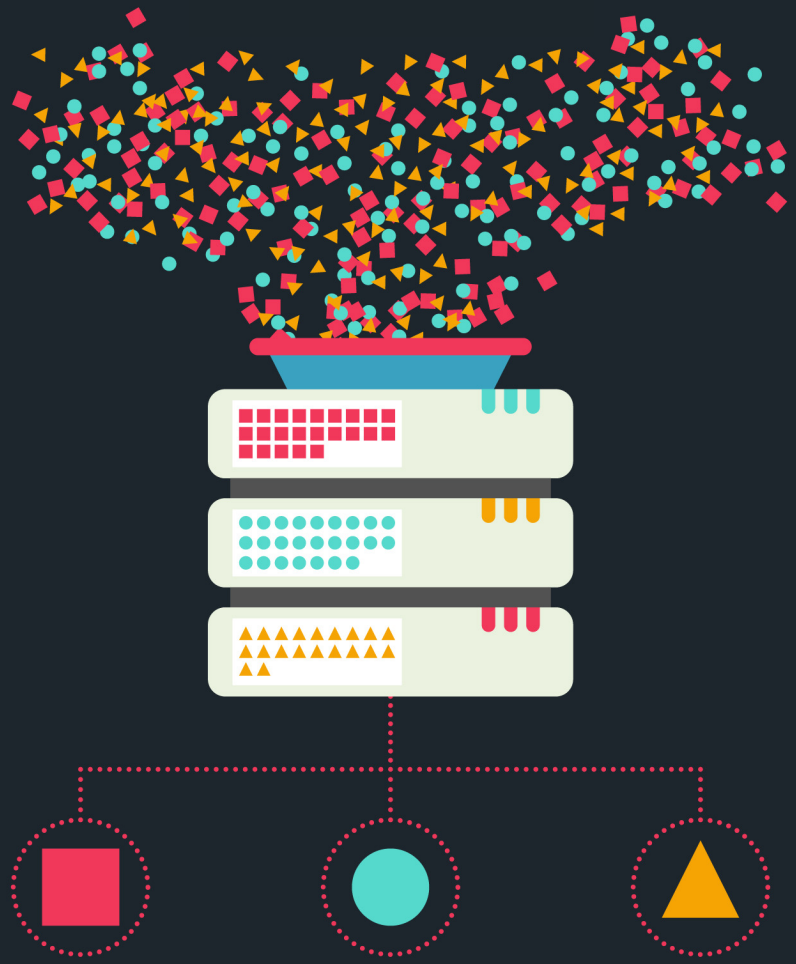
Georgia Tech

# Classification



**Mohsen Moghaddam, Ph.D.**

Gary C. Butler Family Associate Professor
H. Milton Stewart School of Industrial and Systems Engineering
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

SAIL
Symbiotic & Augmented Intelligence Laboratory

Spring 2026

# Learning Outcomes

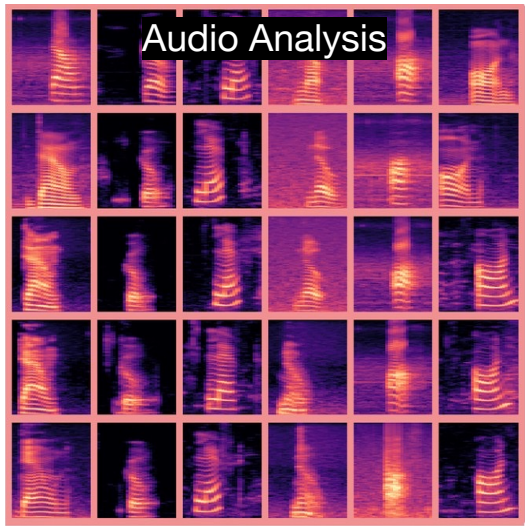
- Formulate supervised classification problems and interpret decision boundaries in feature space
- Apply Bayes decision theory and distinguish generative and discriminative classification models
- Analyze linear and quadratic classifiers arising from Gaussian assumptions (LDA and QDA)
- Apply Naïve Bayes, k-nearest neighbors, and logistic regression, and interpret their modeling assumptions
- Train logistic regression models using gradient-based optimization methods
- Evaluate classification performance using confusion matrices, precision, recall, and F1-score

# Motivation & Problem Statement

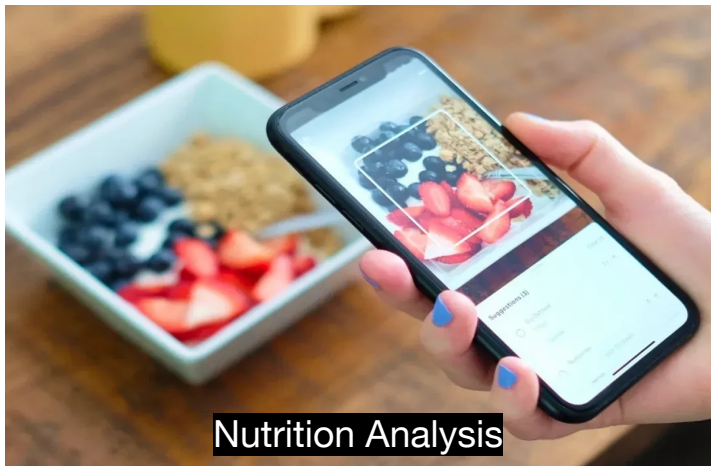
# Classification

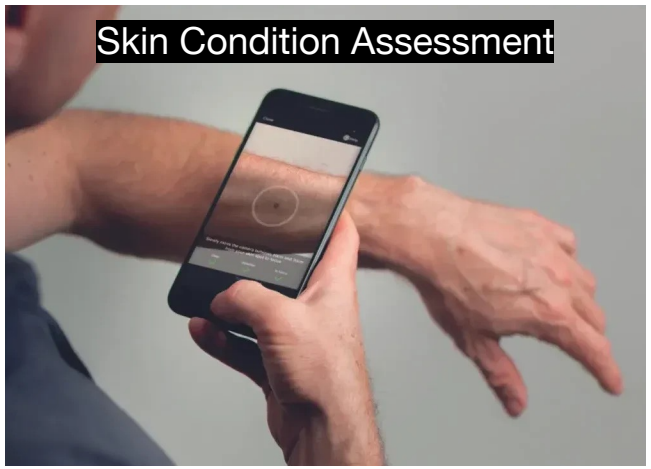
- Classification is a **predictive task**—given input, predict categorical output
- **Supervised learning:** Training data include both features (input) and the categorical response (output)

**How do we map high-dimensional feature vectors to discrete labels in a principled and generalizable way?**

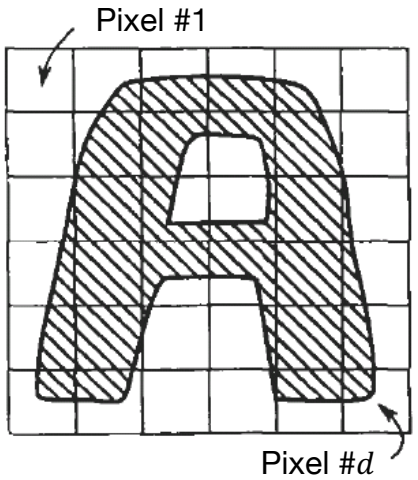




# Classification (cont.)

Each data point is represented as a **feature vector**:

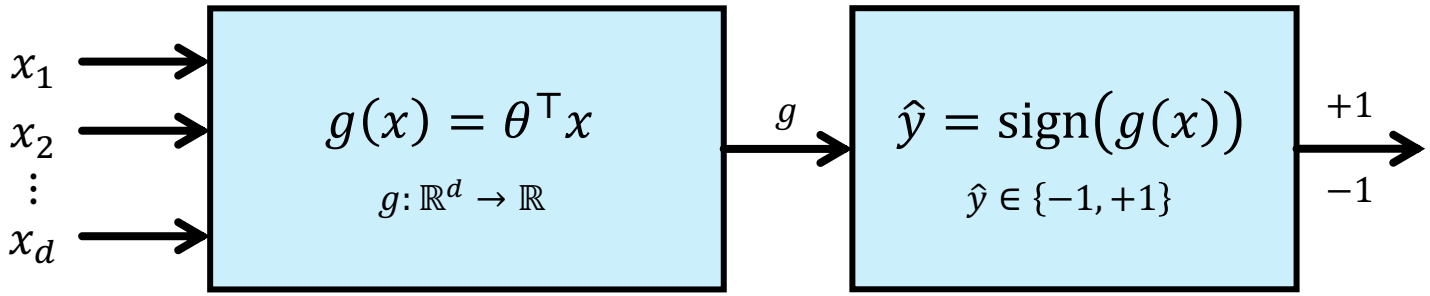



$$\Rightarrow \quad X = (x_1, x_2, \ldots, x_n)^\top \in \mathbb{R}^{n \times d}$$

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^\top \in \mathbb{R}^d$$

A label is provided for each data point: e.g., $y_i \in \{-1, +1\}$ or $\{0,1\}$

**Classifier:**



# Classification vs Clustering

**MNIST** example:

**Classification**

?

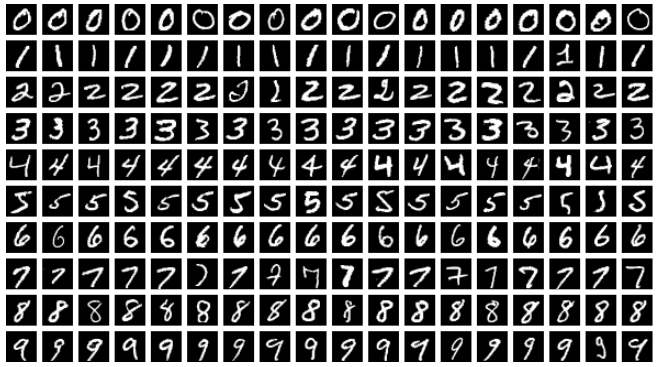
**5**

?

**7**

**Supervised learning**—labels are known during training

**Clustering**

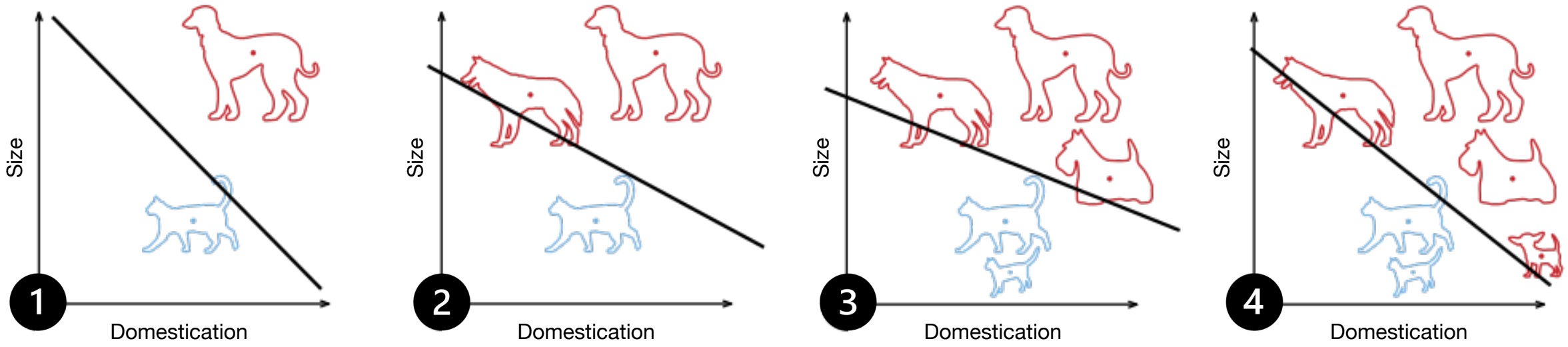


**Unsupervised learning**—labels are inferred from structure alone

**Classification learns a decision boundary—clustering discovers structure**

# Decision Boundary

A decision boundary separates regions of the **feature space** assigned to different labels—e.g., features like “size” and “domestication” to classify cats and dogs

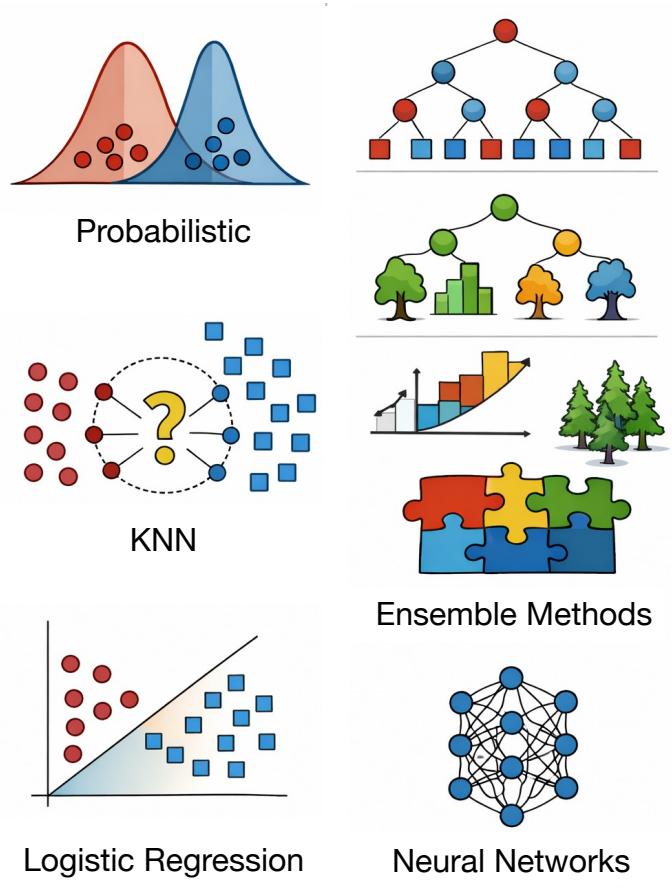



**Which decision boundary generalizes best to unseen data?**

# Classification Algorithms

Algorithms differ by **assumptions**, **model flexibility**, and **computational cost**

- Probabilistic, parametric; e.g., **Bayes classifier**, **naïve Bayes**
- Distance-based; e.g., K-nearest neighbors (**KNN**)
- Linear models determining how features map to labels; e.g., **logistic regression**
- Nonlinear; e.g., support vector machine (**SVM**) as a geometry-based method; **neural networks** that capture complex non-linear decision boundary
- **Ensemble methods**, **boosting**, and **decision tree**



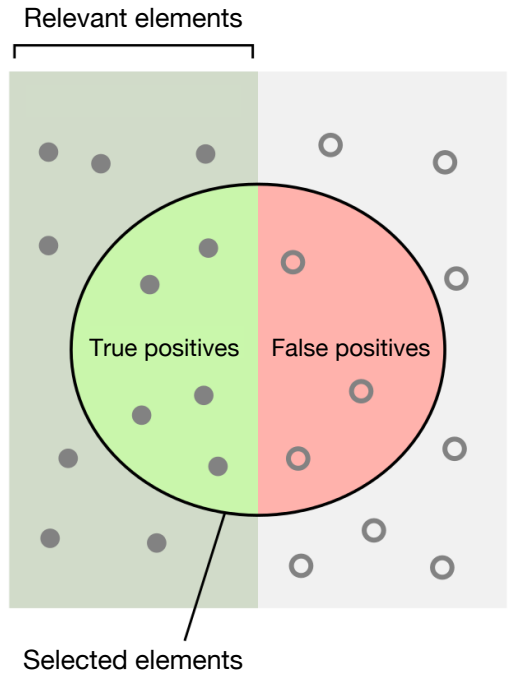
# Evaluating Classification Performance

# Classification Performance Measures

## Misclassification Error (Error Rate)

The fraction of samples whose predicted label does not match the true label:

**# of misclassified samples ÷ total # of samples**




- It treats all errors equally and does not distinguish between mistake types (e.g., false positives vs. false negatives)
- As a result, it can be misleading for imbalanced datasets, where one class is much more common than the other

**Why might misclassification error be misleading for imbalanced datasets?**

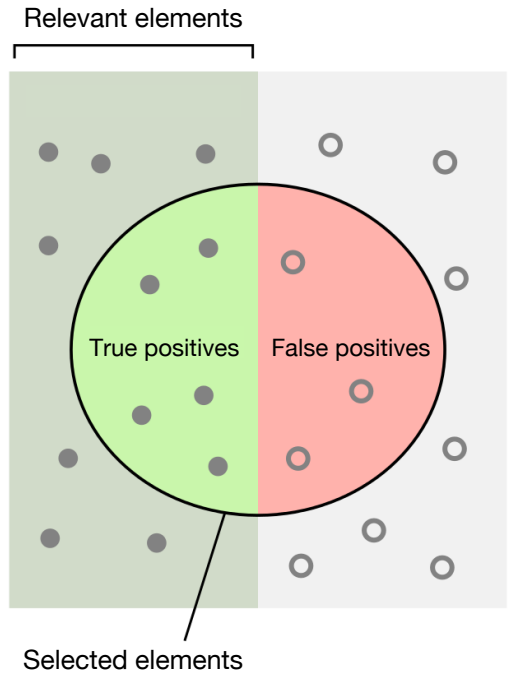
# Classification Performance Measures (cont.)

## Confusion Matrix

Explicitly summarizing how predictions are distributed across true and predicted classes

| Predicted Class \ Actual Class | 🐱 | 🐶 |
|---|---|---|
| 🐱 | 5 | 2 |
| 🐶 | 3 | 3 |

e.g., 2 Dogs were predicted as Cats




- E.g., showing which animals are **confused** with each other
- It is the basis for more informative metrics like precision, recall, and F1-score

**Which entries in the matrix represent false positives?**

# Classification Performance Measures (cont.)

## Precision

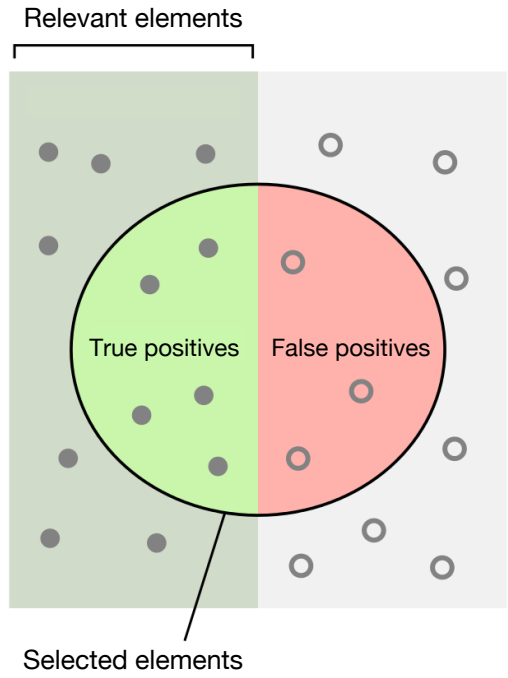
# of retrieved relevant instances ÷ total # of **retrieved** $= \frac{TP}{TP + FP}$

## Recall

# of retrieved relevant instances ÷ total # of **relevant** $= \frac{TP}{TP + FN}$

## F1 Score

$$\frac{2}{\text{precision}^{-1} + \text{recall}^{-1}} \in [0,1]$$




## Example

| | | Actual Class 🐱 | Actual Class 🐶 | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|
| Predicted Class | 🐱 | 5 | 2 | 0.714 | 0.625 | 0.666 |
| | 🐶 | 3 | 3 | 0.500 | 0.600 | 0.545 |

How many selected items are relevant?

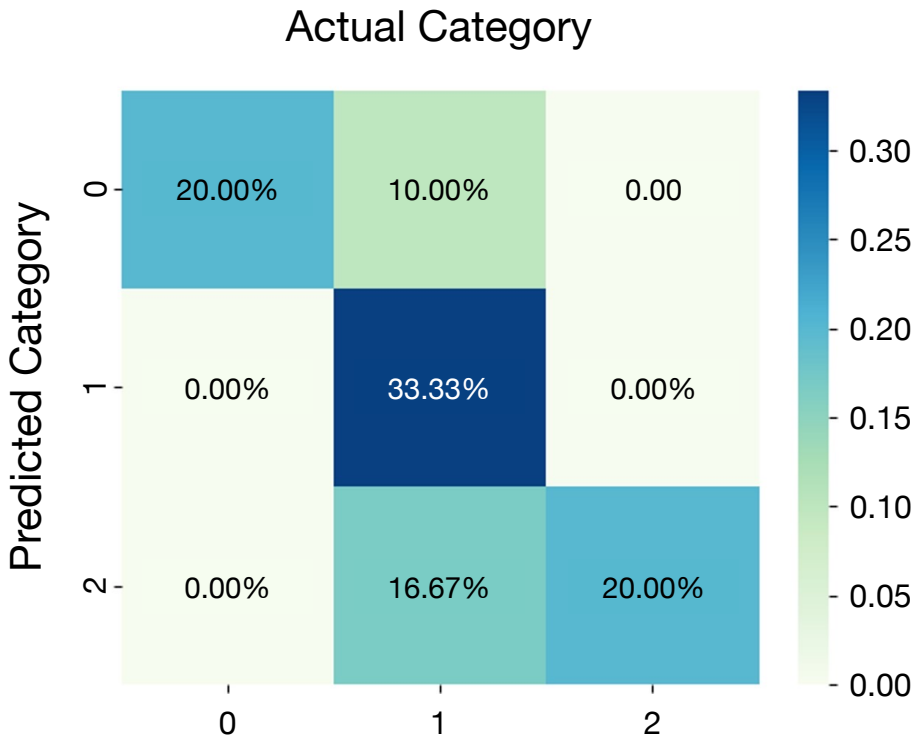
Precision = $\frac{\text{relevant selected}}{\text{selected}}$

How many relevant items are selected?

Recall = $\frac{\text{relevant selected}}{\text{relevant}}$

# Multi-Class Classification

- A confusion matrix is a table that summarizes classification performance by showing **how often predictions match actual (true) labels**
- Each **row** represents the **actual classes**, and each **column** represents the **predicted classes** (or vice versa)
- It helps identify where the model is making correct predictions and where it is **"confusing" one class for another**




**Confusion matrices generalize naturally from binary to multi-class problems**

# Example

Suppose we have a multi-class classifier that predicts animal types, 🐱 **Cat**, 🐶 **Dog**, 🐰 **Rabbit** —we have tested it on 15 images:

| | | Actual Class | | |
|---|---|---|---|---|
| | | 🐱 | 🐶 | 🐰 |
| Predicted Class | 🐱 | 4 | 1 | 0 |
| | 🐶 | 1 | 5 | 1 |
| | 🐰 | 0 | 1 | 2 |

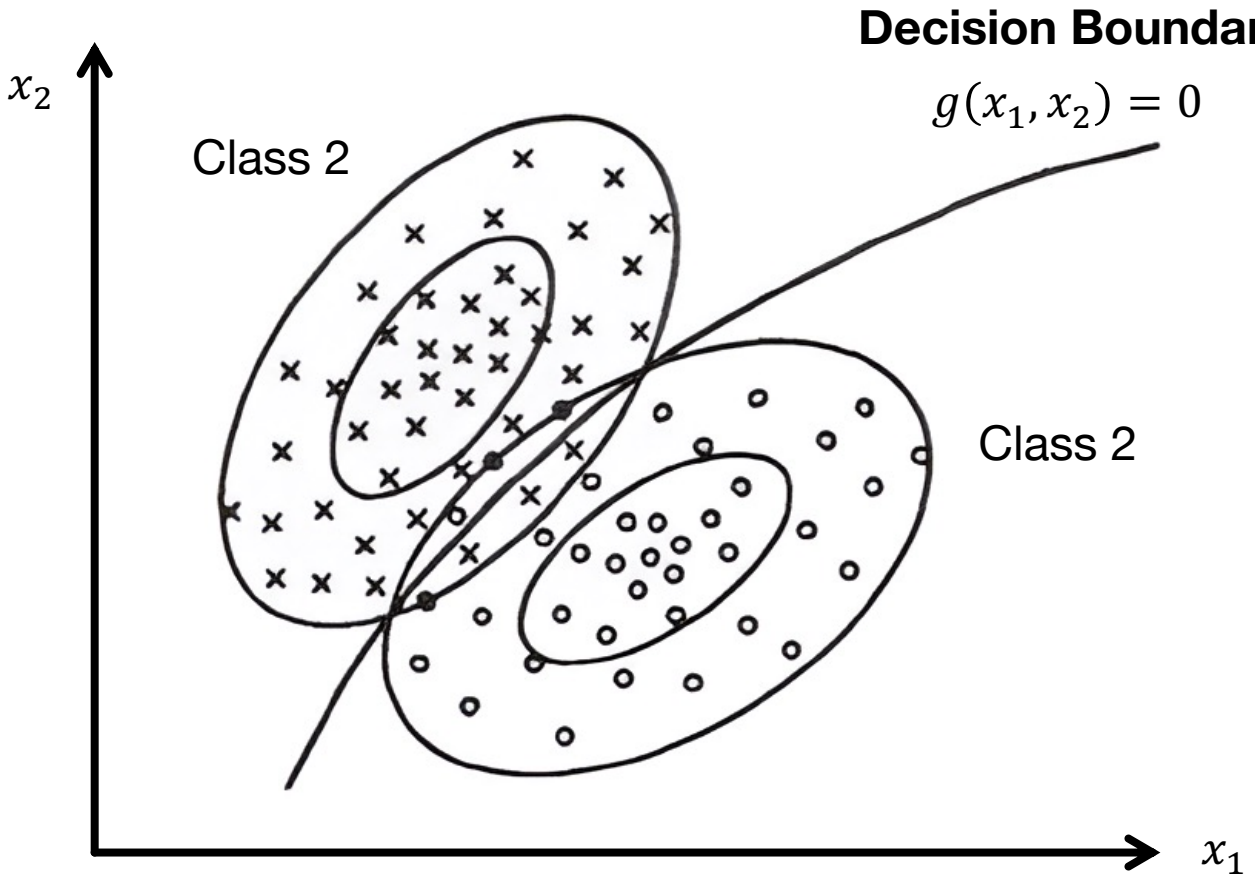
- **Diagonal elements** → correctly predicted (**true positives** for each class)
- **Off-diagonal elements** → misclassifications
  → One Cat was predicted as a Dog, one Dog was predicted as a Rabbit, etc.

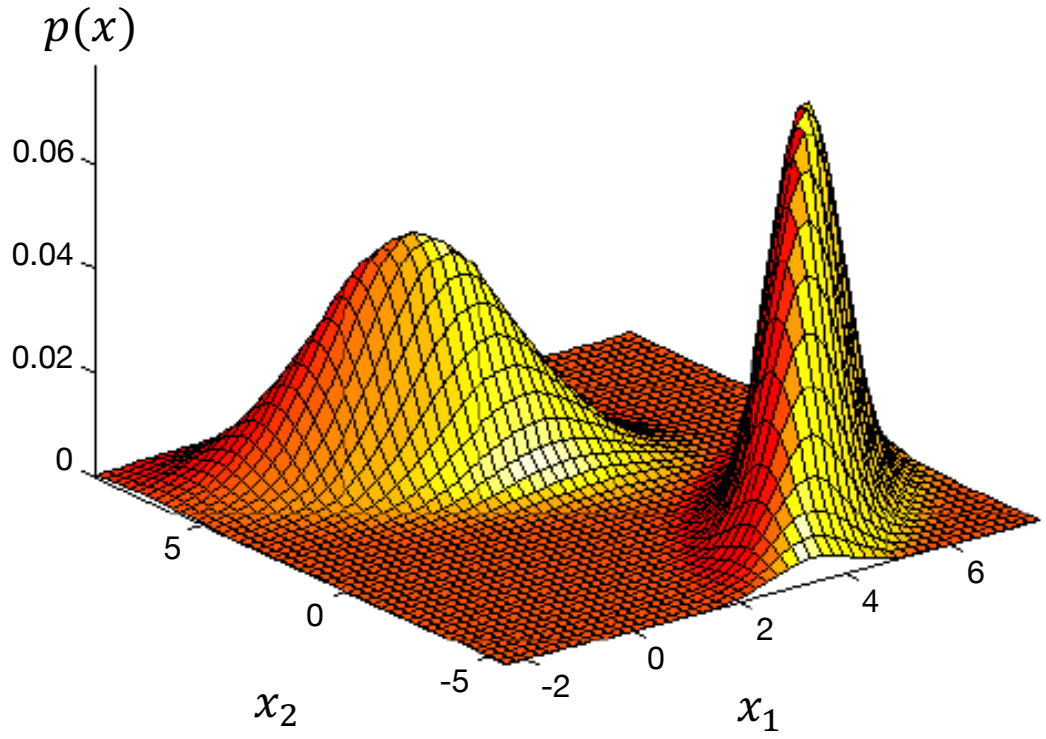
**Which class seems hardest for the model to predict correctly?**

# Bayes Decision Theory & Generative Models

# Divide High-Dimensional Space

Classification can be viewed as partitioning feature space into decision regions



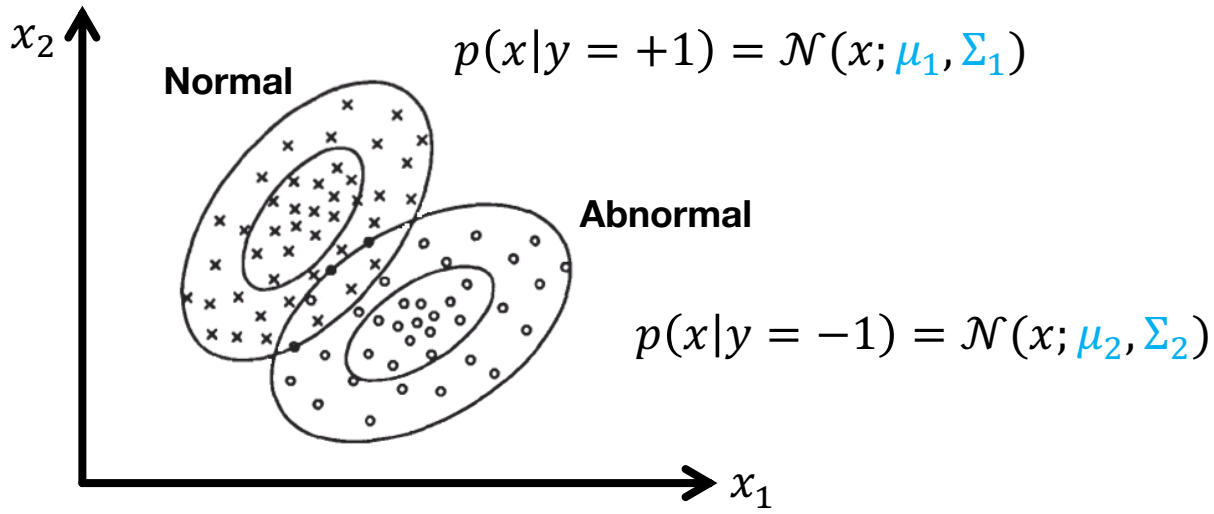
# Bayes Classifier

**Parametric approach:** Assume a parametric distribution (e.g., Gaussian) for the features conditioned on each class—model the class-conditional likelihood $p(x|y)$

$$p(x|y=+1) \qquad p(x|y=-1)$$

**Class prior:**

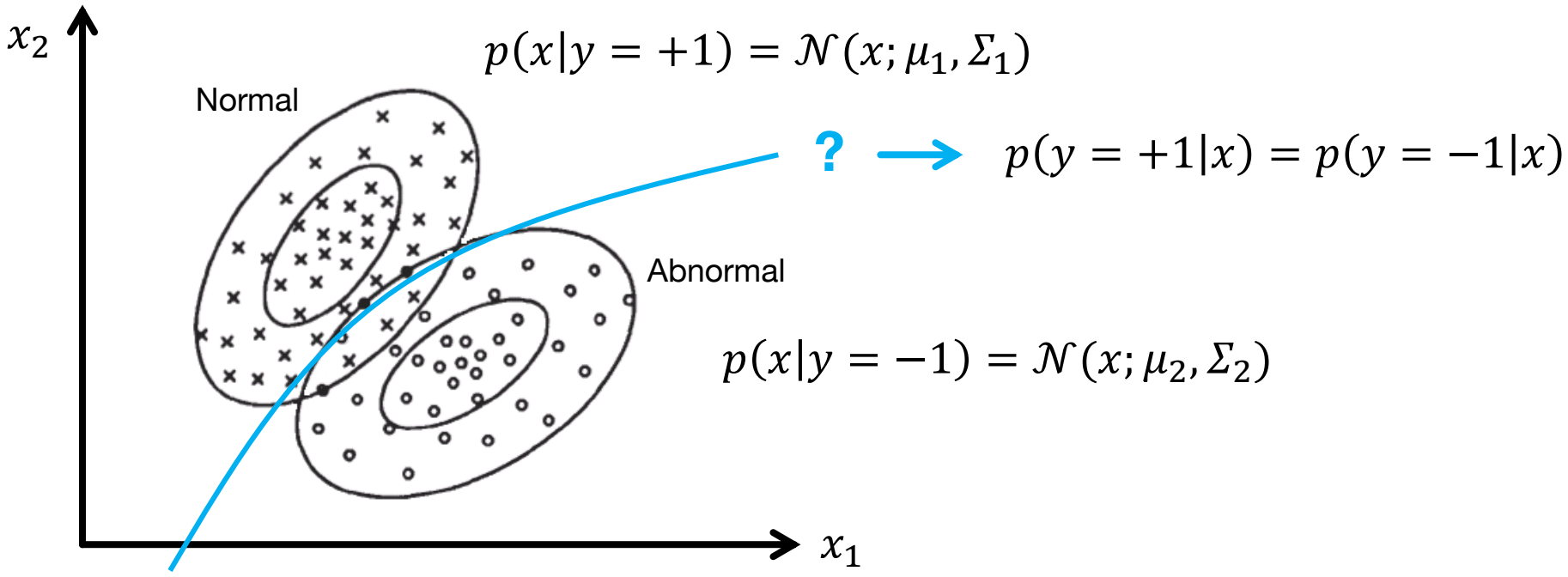
- Normal: $p(y=+1)$
- Abnormal: $p(y=-1)$




**What happens if the prior strongly favors one class?**

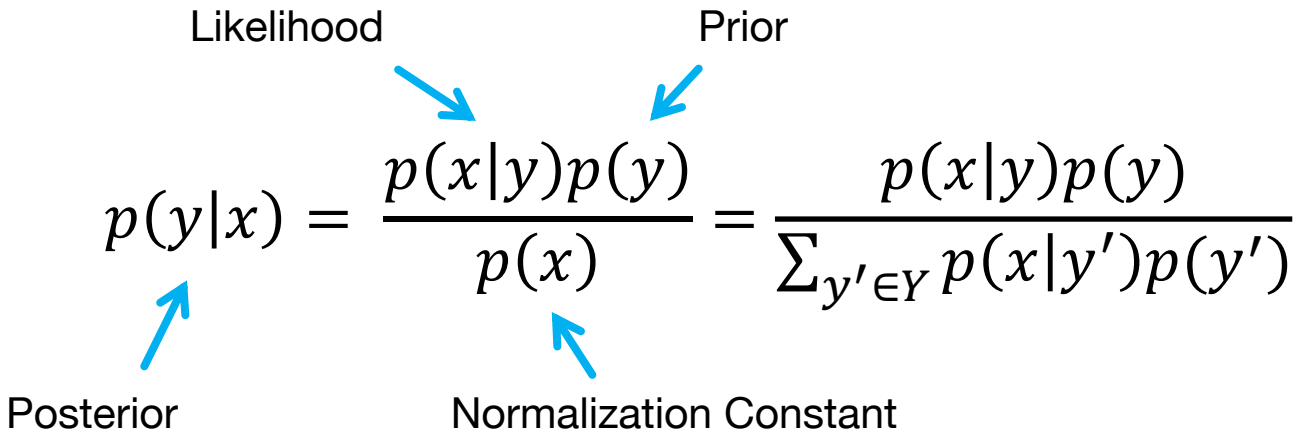
# Decision Boundary

Given class conditional distribution (likelihood) $p(x|y=+1), p(x|y=-1)$ and class prior $p(y=+1), p(y=-1)$, how can we produce a decision boundary?




**Decision boundary is defined by points where posterior probabilities are equal**

# Use Bayes Rule




**Example:** If $y \in \{+1, -1\}$, then

$$p(y|x) = \frac{p(x|y)p(y)}{p(x|y=+1)p(y=+1) + p(x|y=-1)p(y=-1)}$$

**Bayes decision rule combines data evidence and prior belief**

# Bayes Decision Rule

- Learn class **prior** $p(y)$ and **likelihood** $p(x|y)$
- Calculate **posterior** probability of a test sample $x$ belonging to class $i$

$$q_i(x) := p(y = i|x) = \frac{p(x|y = i)p(y = i)}{p(x)}$$

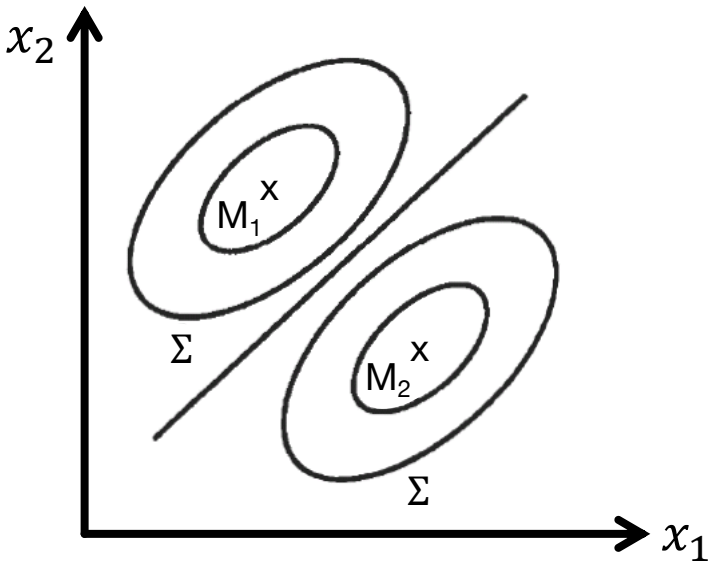
- **Bayes decision rule:** If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$
- We can also use $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$: if $h(x) < 0$, then $y = i$, otherwise $y = j$
- This is equivalent to the following decision boundary

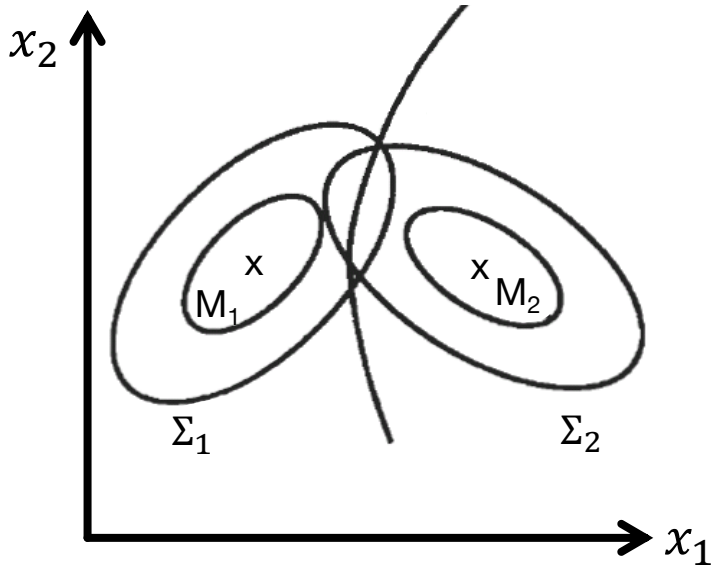
If **likelihood ratio** $l(x) = \frac{p(x|y=i)}{p(x|y=j)} > \frac{p(y=j)}{p(y=i)}$, then $y = i$, otherwise $y = j$

# Gaussian Class Conditional Distribution

Depending on assumptions made for the likelihood, the decision boundary can be very different (e.g., linear or quadratic)

**Linear Discriminant Analysis (LDA)**




**Quadratic Discriminant Analysis (QDA)**

**When might a quadratic boundary be necessary?**

# Linear Discriminant Analysis (LDA)

$$\left.\begin{array}{ll} p(x|y=1) = \mathcal{N}(x;\mu_1,\Sigma_1), & p(y=1)=\pi_1 \\ p(x|y=0) = \mathcal{N}(x;\mu_2,\Sigma_2), & p(y=0)=\pi_2 \end{array}\right\} \pi_1+\pi_2=1$$

**Assumption:** $\Sigma_i = \Sigma, i = 1,2$ (all classes share the same covariance matrix)

Gaussian density

$$h(x) = -\ln\frac{q_1(x)}{q_2(x)} = -\ln\frac{p(x|1)\pi_1}{p(x|2)\pi_2} = -\ln p(x|1) + \ln p(x|2) - \ln\pi_1 + \ln\pi_2$$

$$= \underbrace{-(\mu_1-\mu_2)^\top\Sigma^{-1}x}_{\text{How the feature vector aligns with class mean differences}} + \underbrace{\left(\frac{1}{2}\mu_1\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_2\Sigma^{-1}\mu_2\right)}_{\text{Constant offset from the Gaussian distribution (shift the boundary)}} \underbrace{-\ln\pi_1 + \ln\pi_2}_{\text{Prior adjustment (class imbalance)}}$$

# Quadratic Discriminant Analysis (QDA)

$$\left.\begin{array}{ll} p(x|y=1) = \mathcal{N}(x;\mu_1,\Sigma_1), & p(y=1)=\pi_1 \\ p(x|y=0) = \mathcal{N}(x;\mu_2,\Sigma_2), & p(y=0)=\pi_2 \end{array}\right\} \pi_1+\pi_2=1$$

**Assumption:** $\Sigma_i, i=1,2$ (each class has its own covariance matrix)

$$h(x) = -\ln\frac{q_1(x)}{q_2(x)}$$

Quadratic term (curved decision boundary)

$$= \frac{1}{2}x^\top(\Sigma_1^{-1}-\Sigma_2^{-1})x - (\mu_1^\top\Sigma_1^{-1}-\mu_2^\top\Sigma_2^{-1})x + \left(\frac{1}{2}\mu_1\Sigma_1^{-1}\mu_1 - \frac{1}{2}\mu_2\Sigma_2^{-1}\mu_2\right)$$

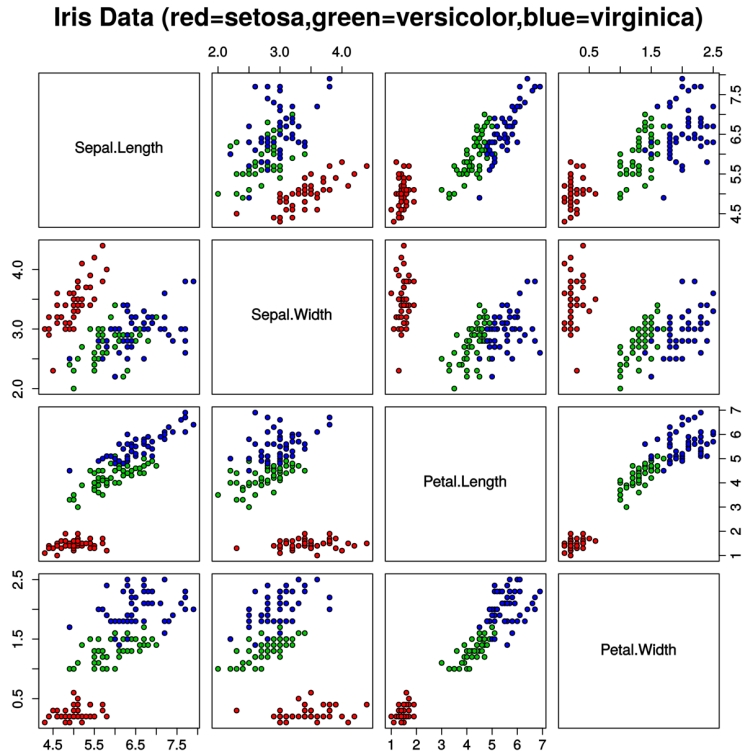
$$-\ln\pi_1 + \ln\pi_2 + \ln\frac{|\Sigma_1|}{|\Sigma_2|}$$

Determinant term (accounts for different covariance spreads)

**What is the statistical cost of this added flexibility?**

# Example: Fisher's Iris Dataset

- The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis
- The data set consists of 50 samples from each of three species of Iris: **Setosa**, **Virginica**, and **Versicolor**
- Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters
- Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other
- Demonstrate how linear combinations of features maximize class separation

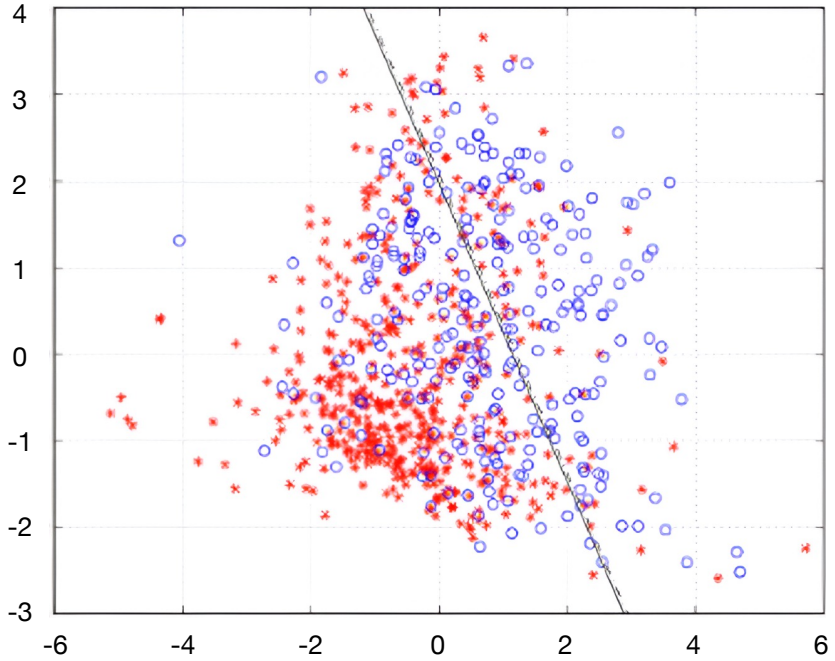



## What does LDA optimize that PCA ignores?

https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html#sphx-glr-auto-examples-decomposition-plot-pca-vs-lda-py

# Example: Diabetes Dataset

- Without diabetes: **stars** (class 1), with diabetes: **circles** (class 0)
- **Solid line:** classification boundary obtained by LDA
- **Dash dot line:** boundary obtained by linear regression of indicator matrix





**Real-world medical data with overlapping classes**

# Example: Diabetes Dataset (cont.)

Two input variables computed from the principal components of original 8 variables

**Prior probabilities:** $\hat{\pi}_1 = 0.651$ $\hat{\pi}_2 = 0.349$

**Mean vectors:** $\hat{\mu}_1 = [-0.4035, -0.1935]^\top$ $\hat{\mu}_2 = [0.7528, 0.3611]^\top$

$$\hat{\Sigma} = \begin{bmatrix} 1.7925 & -0.1461 \\ -0.1461 & 1.6634 \end{bmatrix}$$

**LDA decision rule:**

$$f^{\mathrm{LDA}}(x) = \begin{cases} 1 & 1.1443 - x_1 - 0.5802 x_2 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

**Within training data classification error rate: 28.26%**

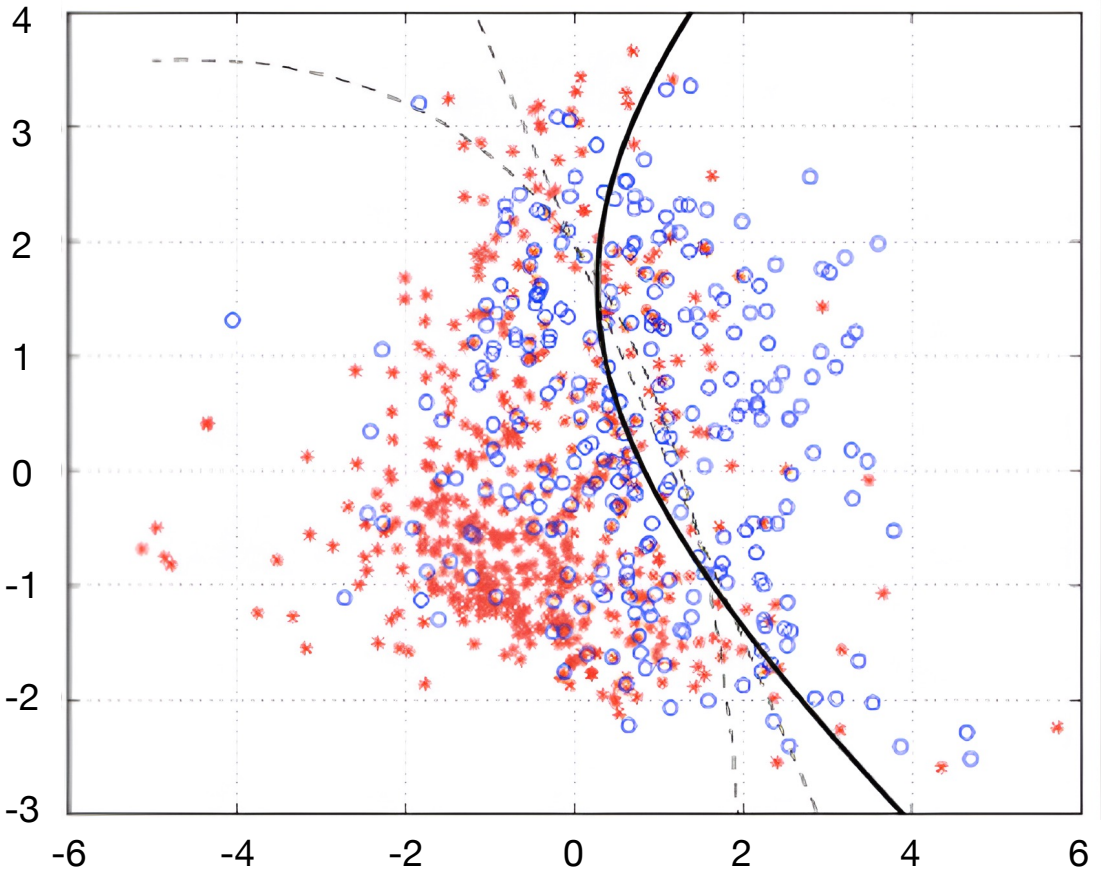
# Example: Diabetes Dataset (cont.)

**QDA:** Everything else stay the same, except that:

$$\hat{\Sigma}_1 = \begin{bmatrix} 1.6769 & -0.0461 \\ -0.0461 & 1.5964 \end{bmatrix}$$

$$\hat{\Sigma}_2 = \begin{bmatrix} 2.0087 & -0.3330 \\ -0.3330 & 1.7887 \end{bmatrix}$$

Increased flexibility does not always reduce error due to variance




**Within training data classification error rate: 29.04%**

# Naïve Bayes Classifier

# Naïve Bayes Likelihood Factorization

- Use Bayes decision rule for classification
- Simplify by assuming all features (dimensions of the vector $x$) are independent given the label
- Thus, likelihood $p(x|y=1)$ is fully factorized

$$p(x|y=1) = \prod_{i=1}^{n} p(x_i|y=1)$$

- Similarly, for $p(x|y=0)$

**What breaks if features are actually correlated?**

# Example: Email Spam Filtering

Suppose we want to train a classifier to identify an email as spam/non-spam

- **Bag-of-words** model converts email text into count-based feature vectors—independence across words is a reasonable approximation

- **Volcabulary:**

```
{secret, offer, low, price, valued, customer, today, dollar,
      million, sports, is, for, play, healthy, pizza}
```

- **Training data:**

Spam

- million dollar offer
- secret offer today
- secret is secret

Non-spam

- low price for value customer
- play secret sports today
- sports is healthy
- low price pizza

# Multinomial Naïve Bayes Likelihood

Assuming $n$ keywords are sampled from **multinomial** distribution, the **likelihood** of a sentence with its feature vector $x$ given class $y = c$:

$$p(x|y=c) = \frac{d!}{x_1! \cdots x_n!} \prod_{k=1}^{n} \theta_{c,k}^{x_k}, \qquad c = 0,1, \qquad d = x_1 + \cdots + x_n$$

- $\theta_{c,k} \in [0,1]$ is the probability of word $k$ appearing in class $c$
- **Constraint:** $\theta_{c,1} + \cdots + \theta_{c,n} = 1, \qquad c = 0,1$
- **Training:** Maximizing log likelihood function of the training data

$$L(\theta_{c,k}, k = 1, \ldots, n) = \sum_{i=1}^{m} \sum_{k=1}^{n} x_k^i \log \theta_{y^i,k}, \qquad c = 0,1$$

# Training and Classification with Naïve Bayes

Solving a constrained maximization problem to find the **estimator** $\{\hat{\theta}_{c,k}\}$

$$\max_{\theta} \; L\left(\theta_{c,k}, c = 0,1, k = 1, \dots, n\right) = \sum_{i=1}^{m} \sum_{k=1}^{n} x_k^i \log \theta_{y^i,k}$$

$$\text{subject to} \qquad \theta_{c,1} + \cdots + \theta_{c,n} = 1, \qquad c = 0,1$$

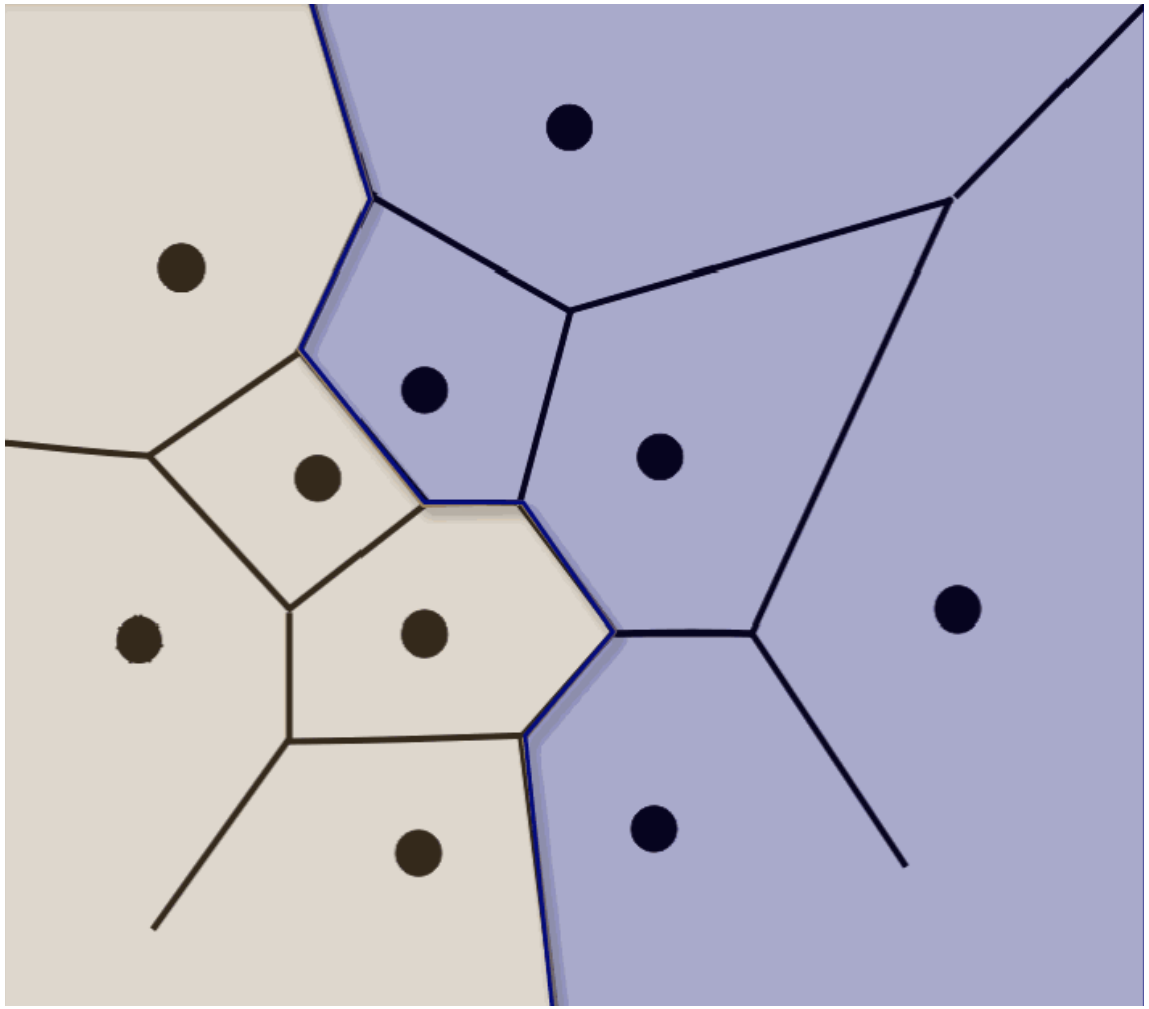
- Apply this to a new email: $z \in \mathbb{R}^n$
- Classification reduces to comparing **likelihood ratios** (compare with 1)

$$\frac{p(z|y=1)}{p(z|y=0)} = \prod_{k=1}^{n} \hat{\theta}_{1,k}^{z_k} / \hat{\theta}_{0,k}^{z_k} \quad \rightarrow \quad \begin{cases} \text{if ratio} > 1 & \Rightarrow \text{ class 1} \\ \text{if ratio} < 1 & \Rightarrow \text{ class 0} \end{cases}$$

# K-Nearest Neighbors

# Nearest Neighbor Classifier

- The nearest neighbor rule defines a **Voronoi partition** of the feature space
- **Nearest neighbor classifier:** Assigning $x$ the same label as the closest training samples $x^i$
- It is non-parametric, non-linear and easy to kernelize
- It is considered a **distance-based method**
- **Intuition:** Similar inputs should have similar labels



# K-Nearest Neighbor Classifier

Assign $x$ a label by taking a majority vote over the $K$ training points $x^i$ closest to $x$

Labels $\in \{\pm 1\}$

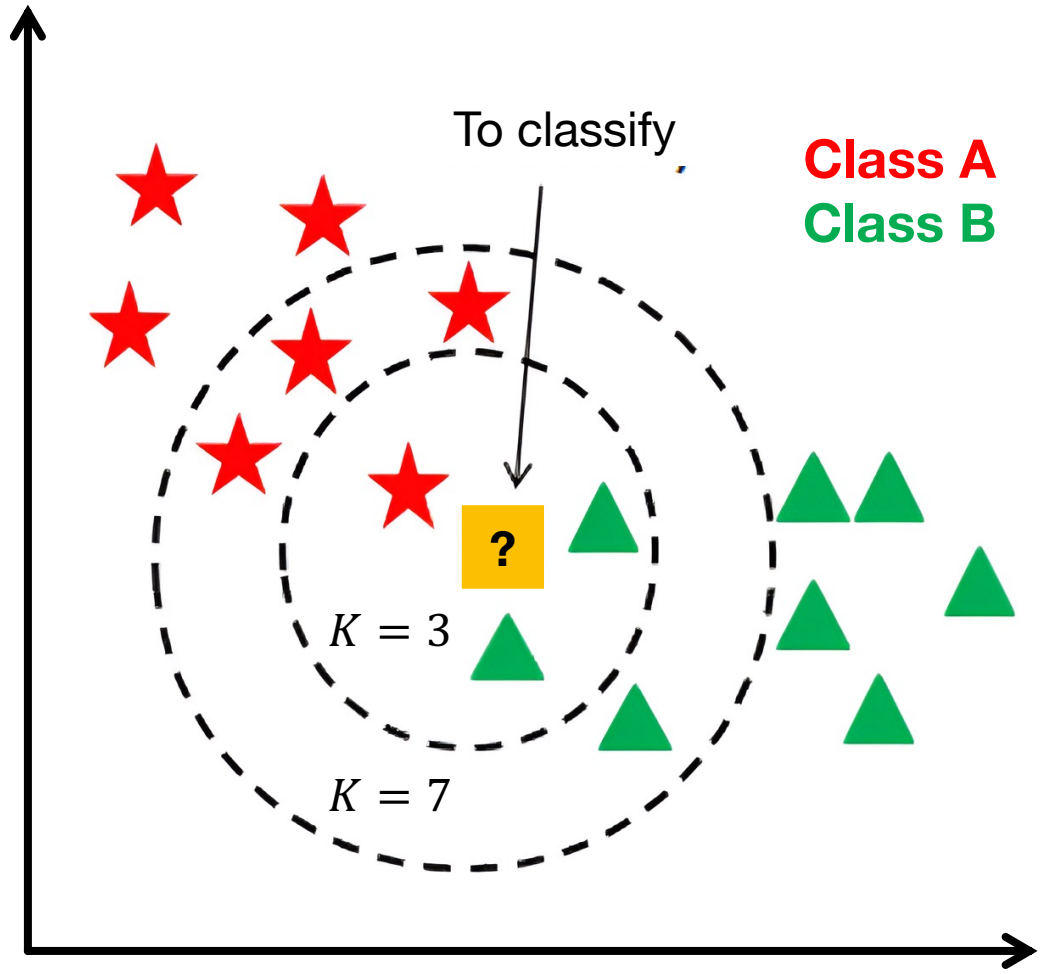
$$f(x) := \text{sign}\left(\sum_{i \in I_k(x)} y^i\right)$$

Predicted label for test point

Indices of the $k$ training points closest to $x$

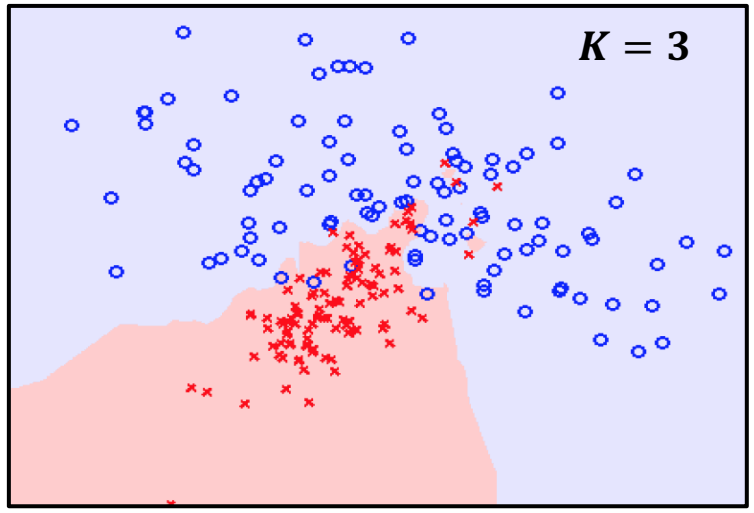
$K \uparrow \Rightarrow$ **bias ↑, variance ↓**

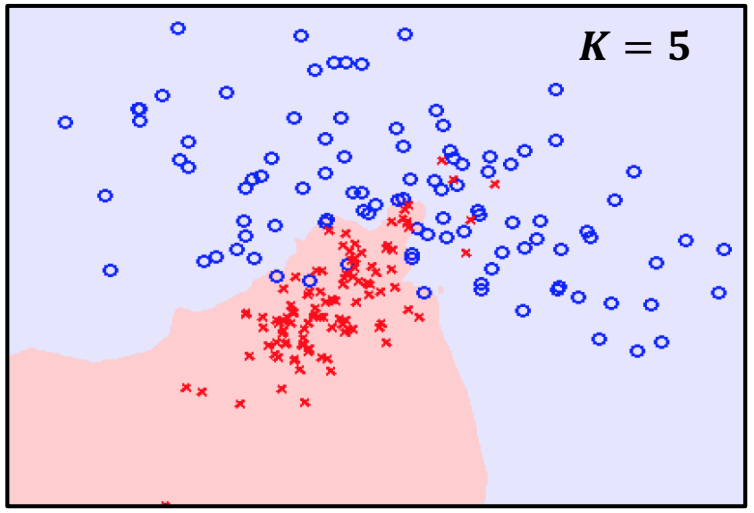
**How does $K$ affect the decision boundary?**

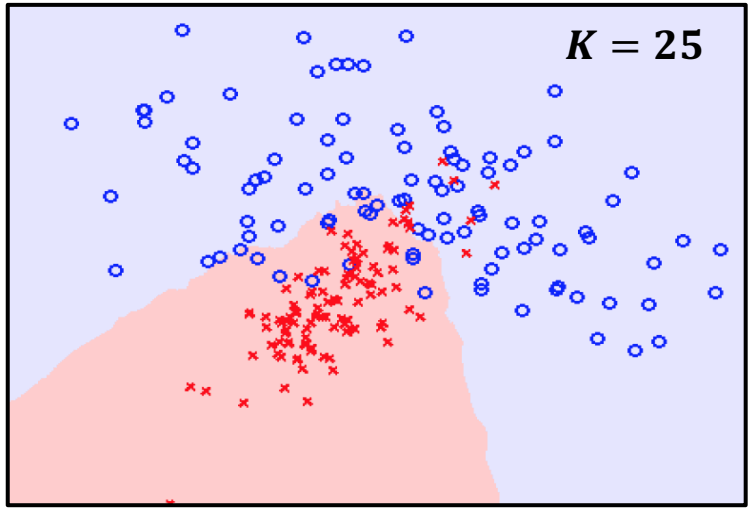


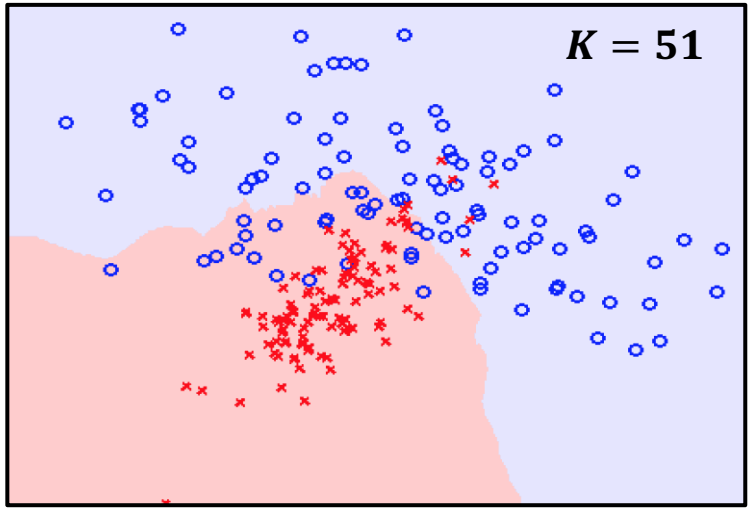

Credit: Sarang Anil Gotke

# Example

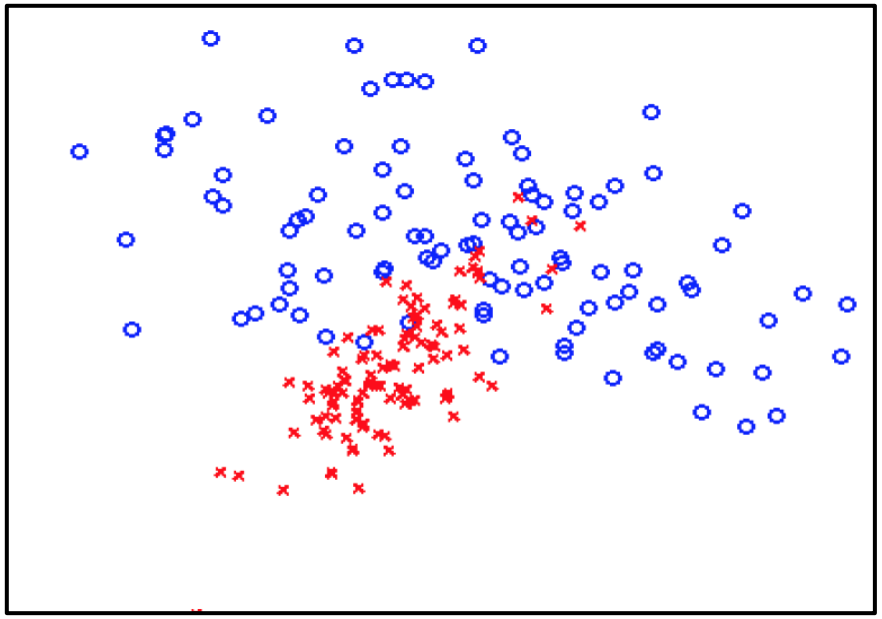



Increasing $K$ smooths the decision boundary and improves stability



# Logistic Regression

# Classification from a Generative Model

**Logistic regression:** Specify $p(y|x^i, \theta)$ using the logistic (sigmoid) function

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top x)}$$

$$p(y = 0|x, \theta) = 1 - p(y = 1|x, \theta) = \frac{\exp(-\theta^\top x)}{1 + \exp(-\theta^\top x)}$$

$$\rightarrow \begin{cases} \theta^\top x > 0 & \Rightarrow y = 1 \\ \theta^\top x < 0 & \Rightarrow y = 0 \end{cases}$$

- Assume data are from a statistical generative model (it is **discriminative** though)
- For each data point, sample the value of $x^i$ according to a prior $p(x)$, and sample a label $y^i$ from the conditional probability $p(y|x^i, \theta)$, $y = 0, 1$

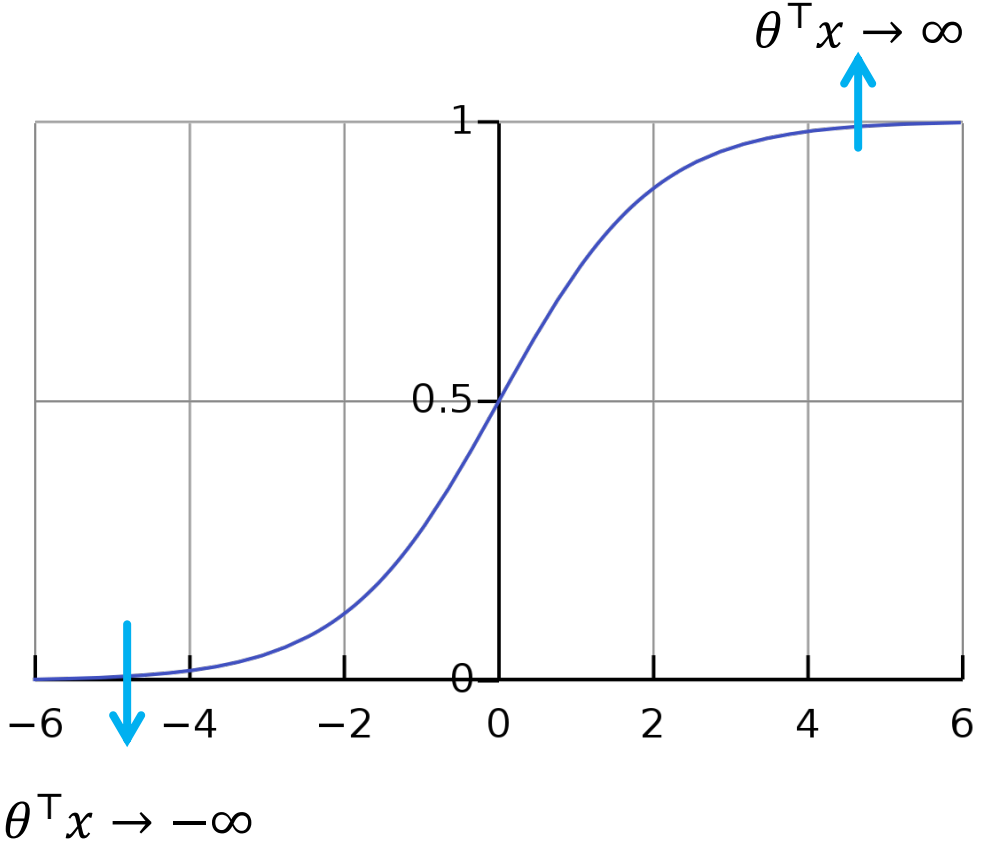
**Why might it be easier to model $p(y|x)$ than $p(x|y)$?**

# Logistic Regression Model

Assume the posterior distribution $p(y = 1|x)$ take a particular form (**link function**)

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top x)}$$

Logistic (sigmoid) function



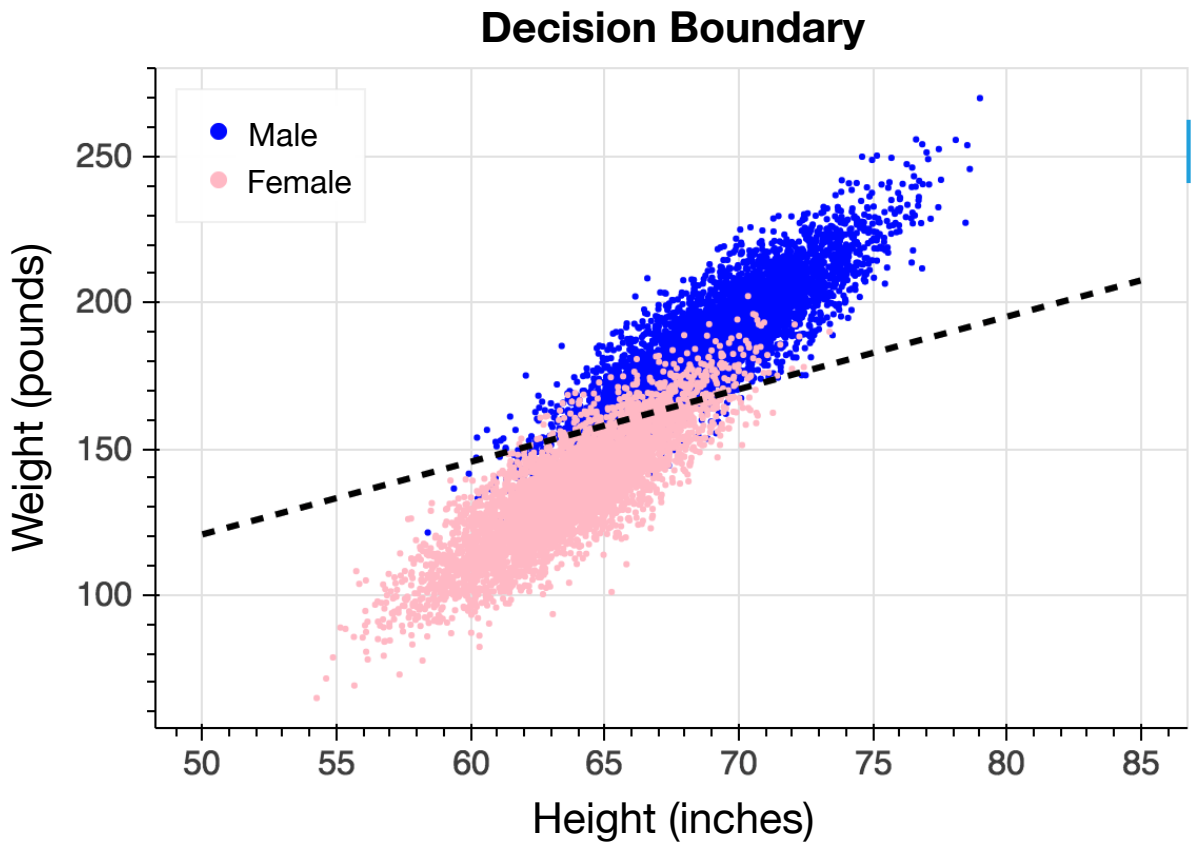

**Classification:** Given a new data $z$

$$\begin{cases} p(y = 1|z, \hat{\theta}) > \eta & \Rightarrow\ y = 1 \\ p(y = 1|z, \hat{\theta}) < \eta & \Rightarrow\ y = 0 \end{cases}$$

**How does changing the threshold $\eta$ prioritize FP versus FN?**

# Decision Boundary of Logistic Regression

- Boundary: $p\left(y = 1 \middle| z, \hat{\theta}\right) = \eta$
- Decision boundary is linear in feature space
- Learning in logistic regression is to find $\theta$ to optimally separate the classes in training data
- Changing $\theta$ will change the decision boundary




**Does logistic regression become nonlinear if I add nonlinear features?**

# Learning Parameters in Logistic Regression

Find $\theta$, such that the **conditional likelihood of the labels** is maximized

$$\max_{\theta} l(\theta) \coloneqq \log \prod_{i} p\left(y^i \middle| x^i, \theta\right) = \sum_{i} \log p\left(y^i \middle| x^i, \theta\right)$$

$$p\left(y^i \middle| x^i, \theta\right) = \left(\frac{1}{1+\exp(-\theta^\top x)}\right)^{y^i} \left(\frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}\right)^{1-y^i}$$

$$\Rightarrow \quad \max_{\theta} l(\theta) \coloneqq \sum_{i} (y^i - 1)\, \theta^\top x^i - \log\left(1+\exp\left(-\theta^\top x^i\right)\right)$$

**$l(\theta)$ is concave (single global optimum), but there is no closed form solution**

# Gradient Ascent/Descent Algorithm

The algorithm iteratively improves the model by moving parameters in the direction that increases the log-likelihood

**Initialize:** $\theta^0$, $\epsilon$

**Repeat for** $t = 0,1,2, \ldots$

$$\theta^{t+1} \leftarrow \theta^t + \gamma_t \sum_i (y^i - 1)\, x^i + \frac{\exp\left(-\theta^{t\top} x^i\right) x^i}{1 + \exp\left(-\theta^{t\top} x^i\right)}$$

$$\frac{\partial l(\theta)}{\partial \theta}$$

(solving maximization)

**until** $\|\theta^{t+1} - \theta^t\| \leq \epsilon$

**Output:** $\theta^{t+1}$

**Step size (learning rate) $\gamma_t$ controls how fast the parameters are updated**

# Models Comparison

How do the three classification models compare?

| Aspect | Bayes Classifier | KNN | Logistic |
|---|---|---|---|
| Number of parameters | $O(n^2)$ | 1 (non-parametric) | $O(n)$ |
| Robustness to model mismatch | No | Yes | Relatively |
| Noisy prediction? | No | Yes | Yes (when logistic function value ~0.5) |
| Decision boundary | Linear (LDA) or quadratic (QDA) | Highly nonlinear, data-dependent | Linear |
| Bias–variance behavior | Higher bias, lower variance | Low bias, high variance (small $K$) | Moderate bias and variance |

# Example: Statlog (Heart) Dataset

- 13 features (e.g., chest pain type, resting blood pressure, etc.)
- 2 demographic
- 11 clinical measures of cardiovascular status and performance
- 2 classes: **absence** (1) or **presence** (2) of heart disease
- 270 samples
- Dataset taken from UC Irvine Machine Learning Repository
- Dataset characteristics: Multivariate

https://archive.ics.uci.edu/dataset/145/statlog+heart

# Multiclass Logistic Regression

Assign input vector $x^i, i = 1, \ldots, m$ into one class $c$, $c = 1, \ldots, C$, assuming that the posterior distribution take a particular form:

$$p\left(y^i = c | x^i, \theta_1, \ldots, \theta_C\right) = \frac{\exp\left(\theta_c^\top x^i\right)}{\sum_{c'} \exp\left(\theta_{c'}^\top x^i\right)}$$

Softmax function

- Fit parameters $\{\theta_c\}$, $c = 1, \ldots, C$
- Solving maximum likelihood
- Also known as softmax regression

**If $C = 2$, does the softmax model reduce to logistic regression?**

# Key Takeaways

# What We Learned This Week

- Classification is a supervised learning task that assigns discrete labels by learning decision boundaries in feature space
- Bayes decision theory provides a unifying framework: optimal classification compares posterior probabilities using priors and likelihoods
- Naive Bayes trades strong independence assumptions for speed and robustness, often performing well in high-dimensional problems
- K-nearest neighbors makes predictions by similarity, illustrating the bias–variance tradeoff through the choice of $K$
- Logistic regression directly models class probabilities and learns a linear decision boundary via maximum likelihood and gradient-based optimization
- Classification performance must be evaluated using confusion matrices, precision, recall, and F1-score, not accuracy alone