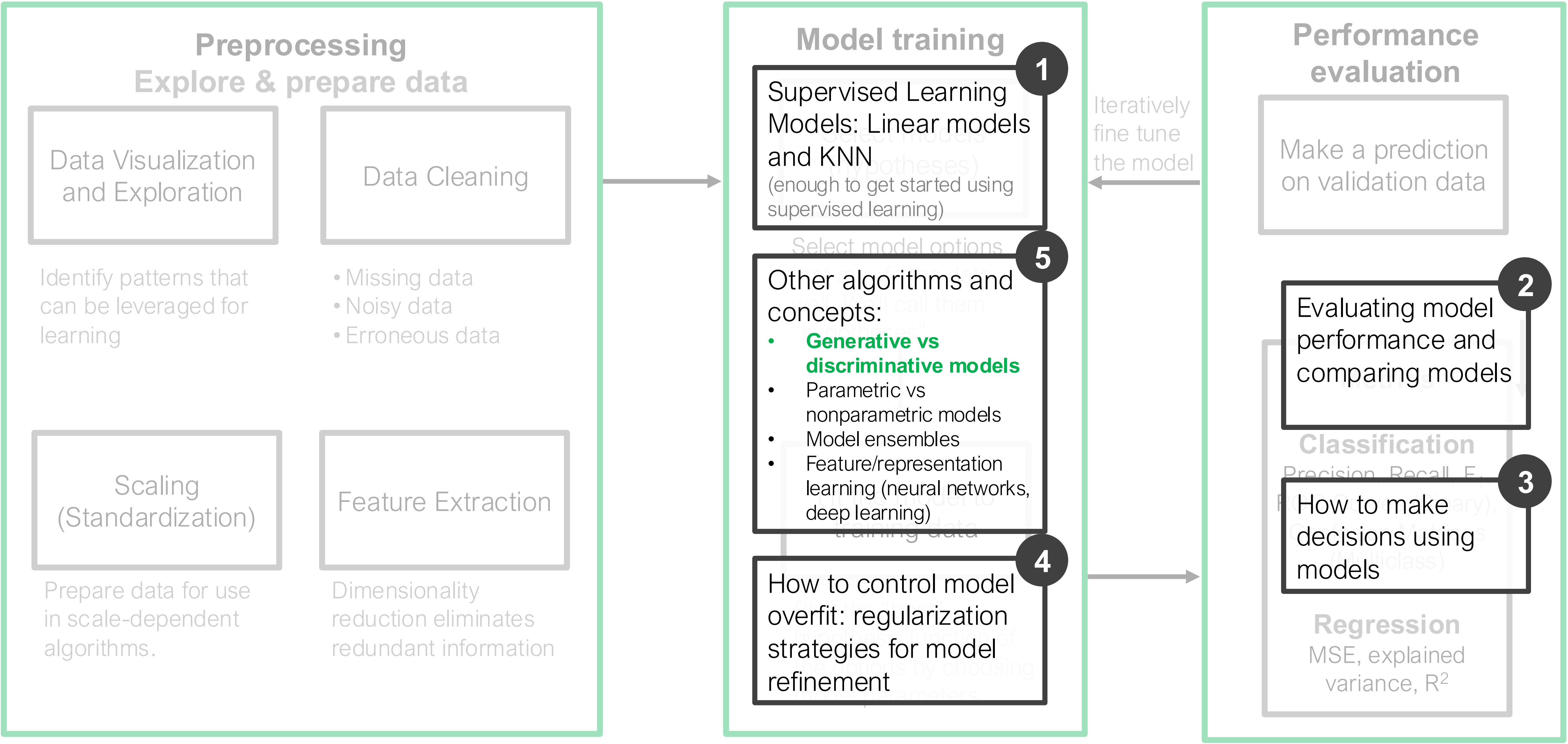
Generative Models for Classification

Supervised learning in practice



Preprocessing
Explore & prepare data

Data Visualization and Exploration

Identify patterns that can be leveraged for learning

Data Cleaning

- Missing data
- Noisy data
- Erroneous data

Scaling (Standardization)

Prepare data for use in scale-dependent algorithms.

Feature Extraction

Dimensionality reduction eliminates redundant information

Model training

1 Supervised Learning Models: Linear models and KNN (enough to get started using supervised learning)

Select model options

5 Other algorithms and concepts:

- **Generative vs discriminative models**
- Parametric vs nonparametric models
- Model ensembles
- Feature/representation learning (neural networks, deep learning)

4 How to control model overfit: regularization strategies for model refinement

Iteratively fine tune the model

Performance evaluation

Make a prediction on validation data

2 Evaluating model performance and comparing models

Classification

3 How to make decisions using models

Regression

MSE, explained variance, R^2

Classifiers

Covered so far

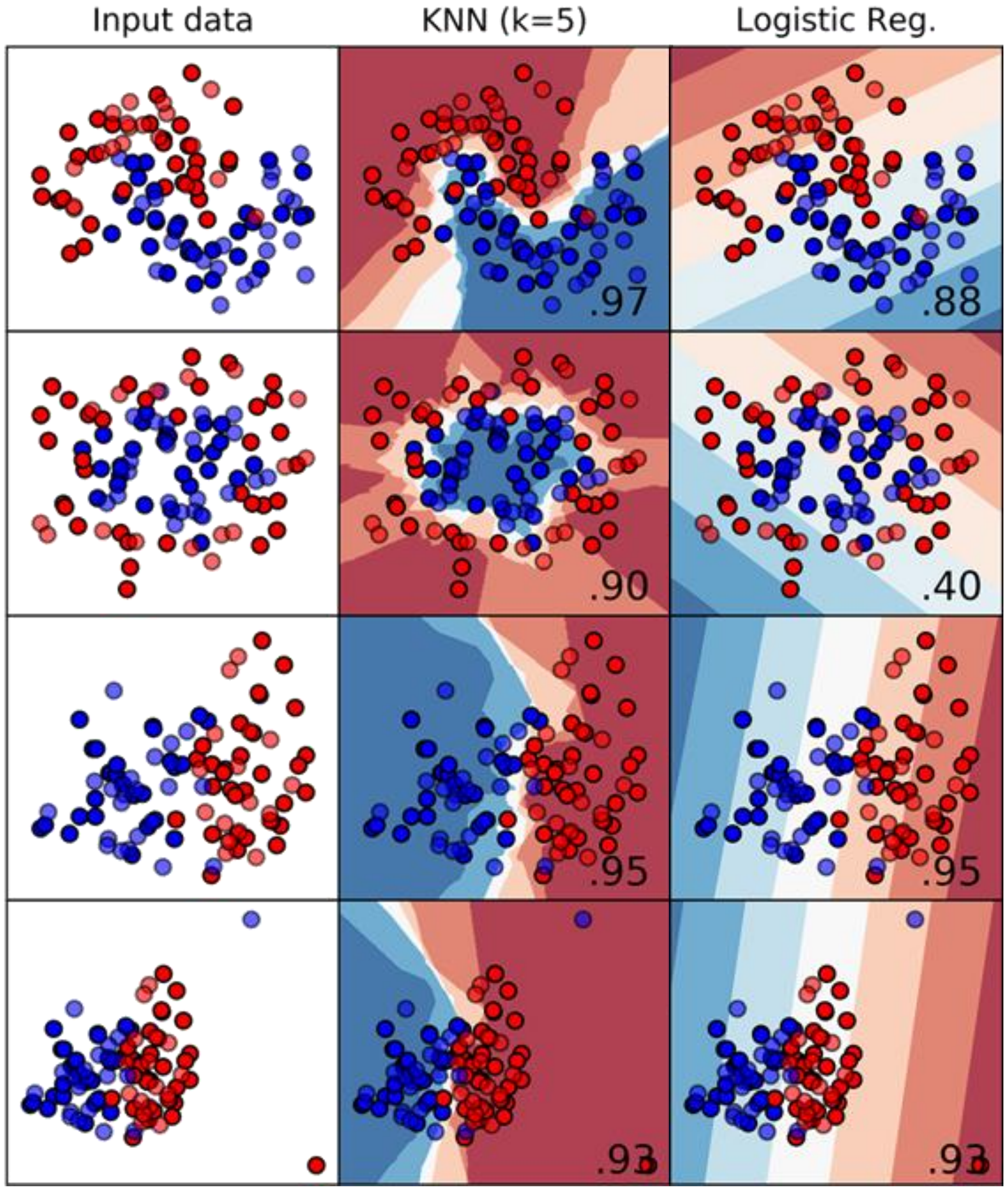
K-Nearest Neighbors

Logistic Regression

Linear/Quadratic Discriminant Analysis

Naïve Bayes

Comparison of classifiers we have seen so far



The color gradient shows the confidence scores

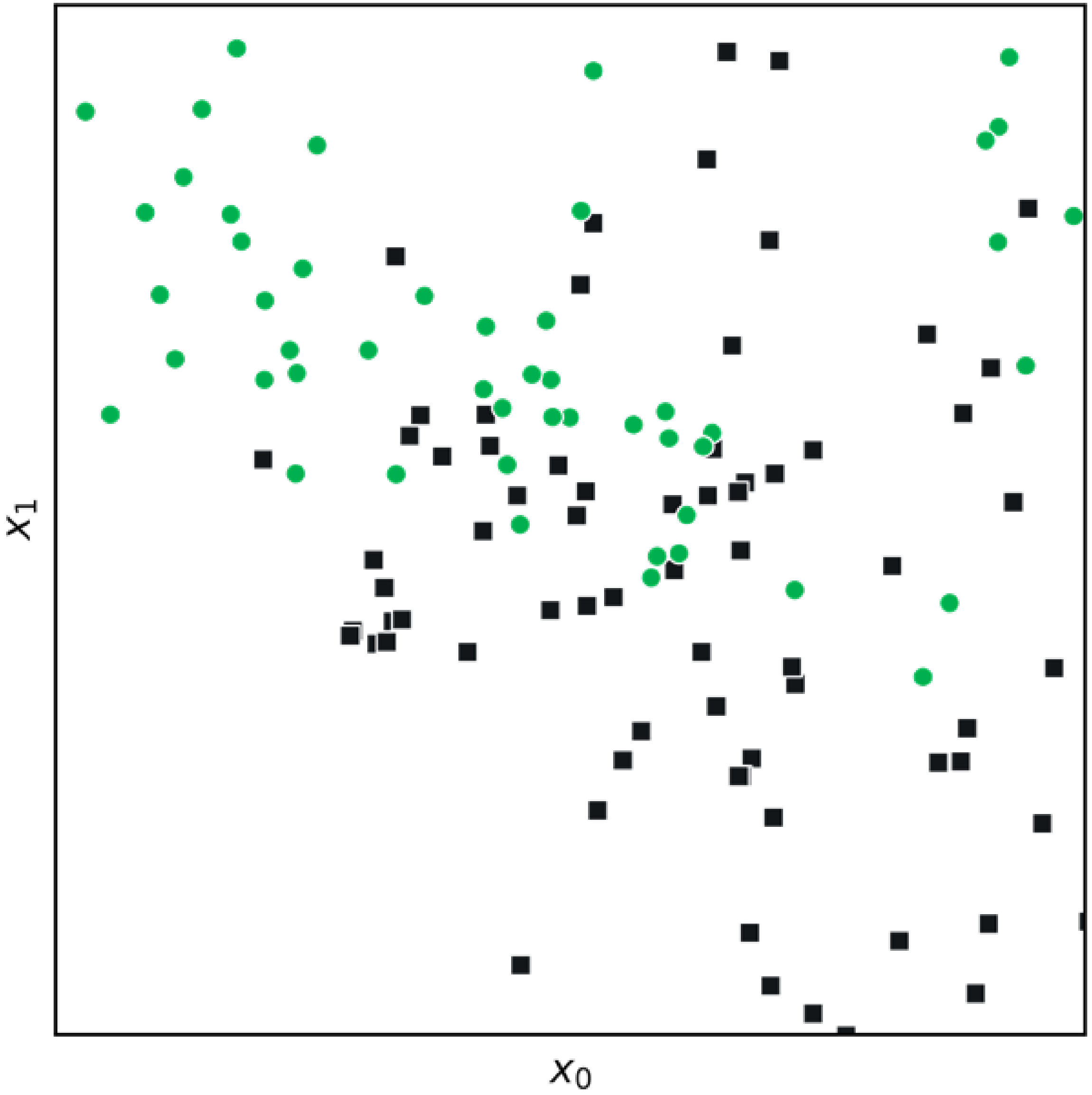
Test data accuracy

What’s the best possible classification model?

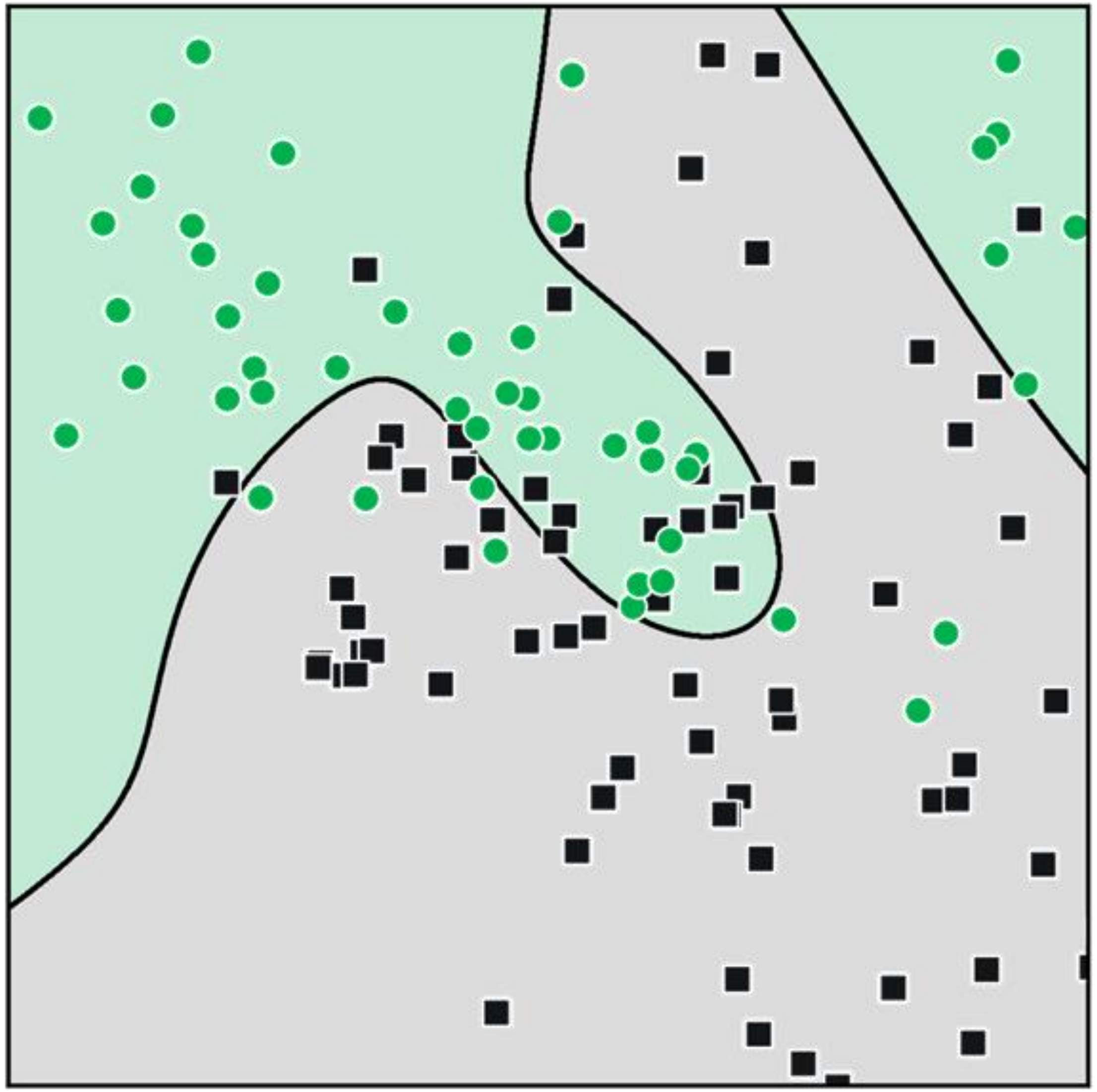
(“best” in the sense of minimum average misclassification rate)

Bayes’ Classifier

Classification feature space



Bayes Decision Boundary



Bayes’ rule in the context of classification

Bayes' Rule

$$\underbrace{P(Y|\boldsymbol{X})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{X}|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{X})}_{\text{Evidence}}}$$

$\boldsymbol{X}$ Features

Y Class label

i.e. $\mathrm{Y} \in \{0,1\}$ for the binary case

Bayes' Decision Rule:

choose the most probable class given the data

$$\hat{y} = \underset{y \in \{0,1,2,\ldots K\}}{\operatorname{argmax}} P(Y = y|\boldsymbol{X})$$

- If the distributions are correct, this decision rule is **optimal**
- Rarely do we have enough information to use this in practice

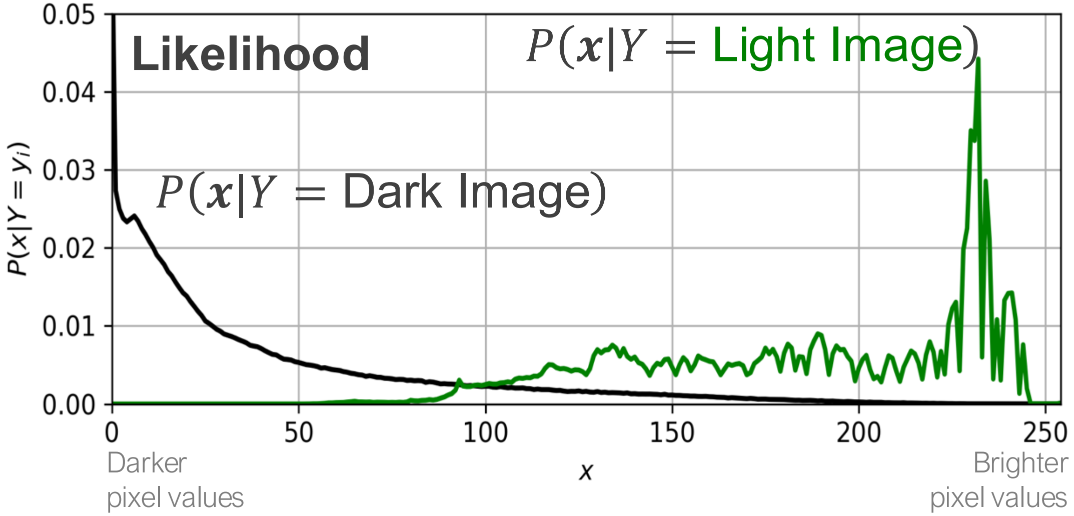
Class 1: Light Image

Class 0: Dark Image

Randomly draw a pixel from either of the images: $x_i = \boxed{149}$

Darker pixel values are lower numbers (closer to 0), brighter pixels are higher numbers (closer to 255)

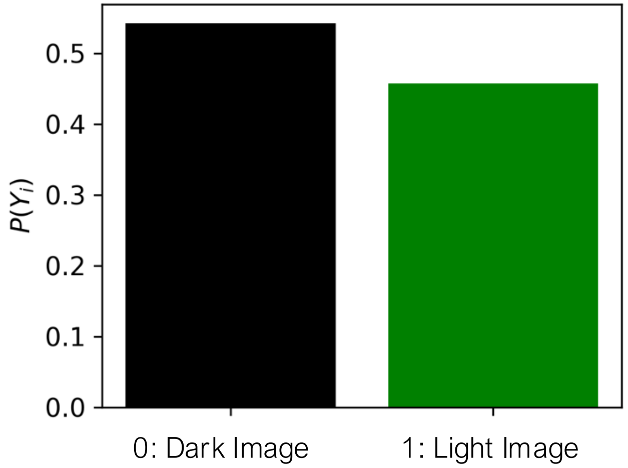
How do we determine which image the sample was most likely to have come from?



Class 1: Light Image y_1

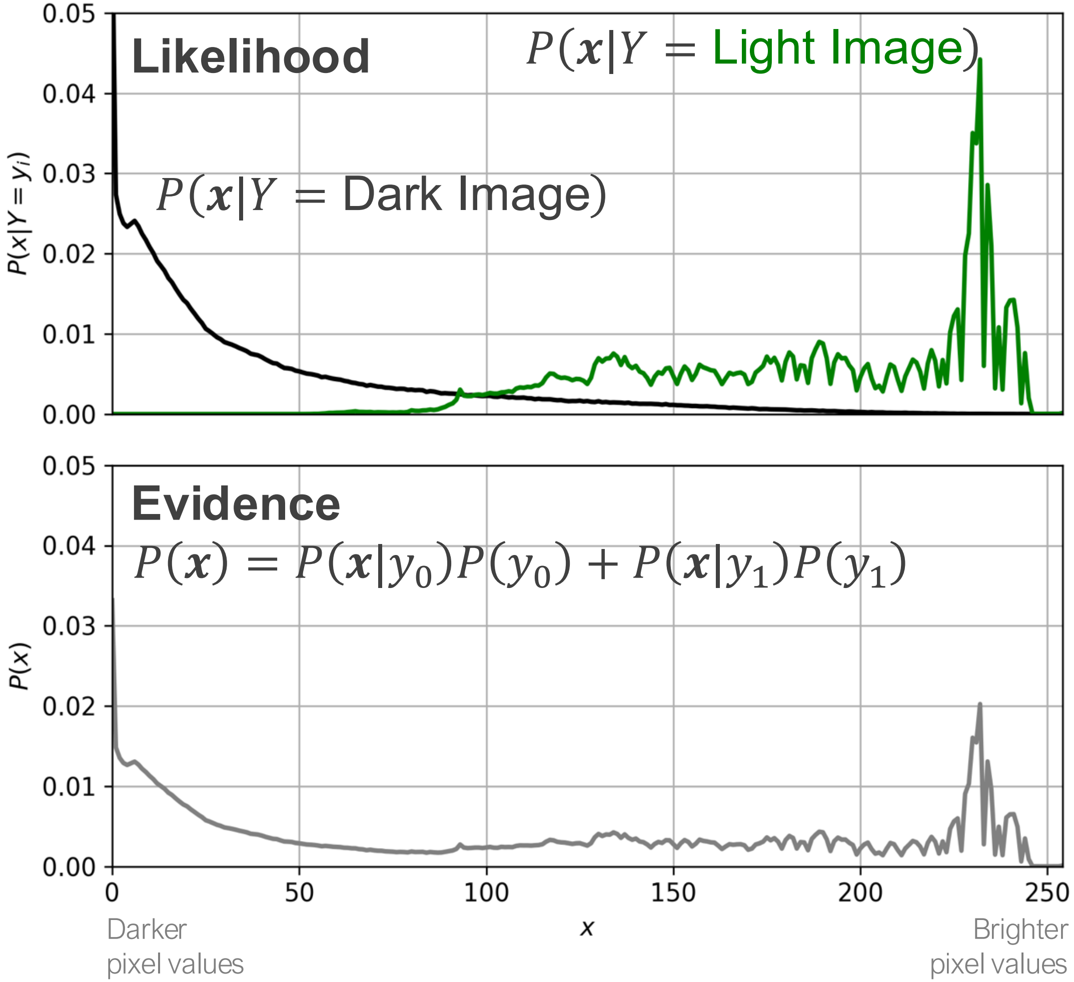
Class 0: Dark Image y_0

Prior: $P(Y = y)$



Bayes' Rule

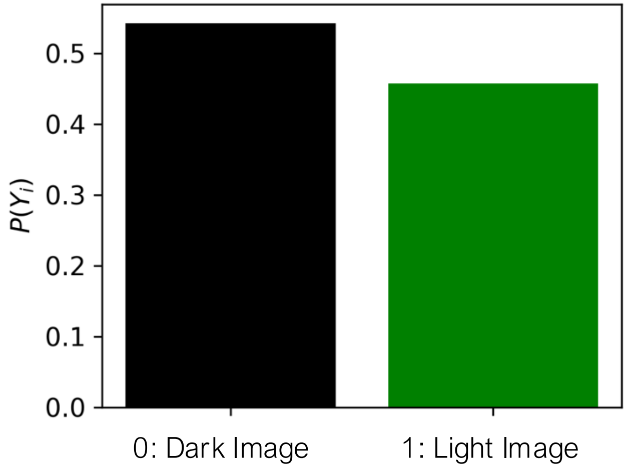
$$\underbrace{P(Y = y|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y)}^{\text{Likelihood}}\overbrace{P(Y = y)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$



Class 1: Light Image y_1

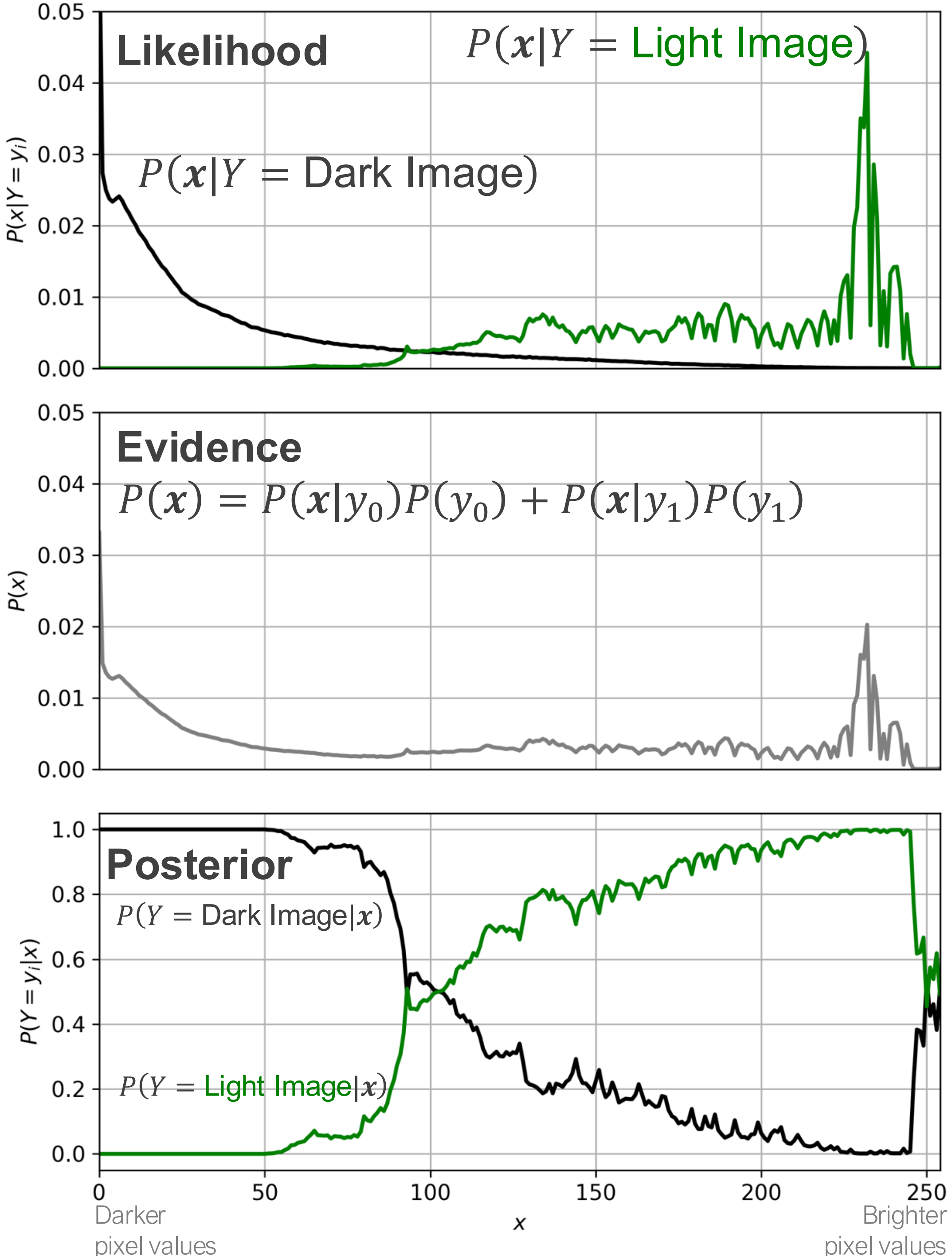
Class 0: Dark Image y_0

Prior: $P(Y=y)$



Bayes' Rule

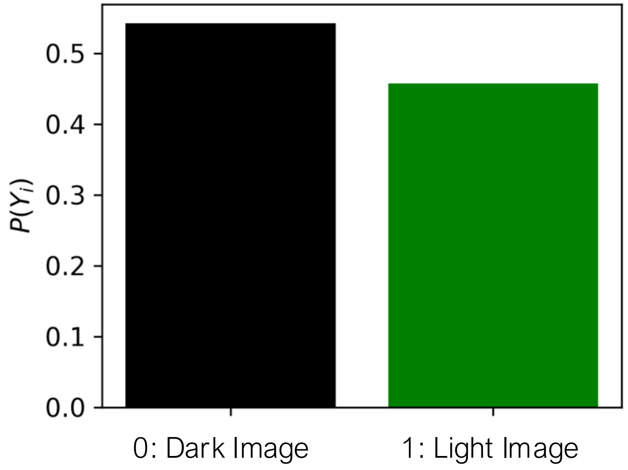
$$\overset{\textbf{Posterior}}{P(Y=y|\boldsymbol{x})} = \frac{\overset{\textbf{Likelihood}}{P(\boldsymbol{x}|Y=y)}\overset{\textbf{Prior}}{P(Y=y)}}{P(\boldsymbol{x})\ \textbf{Evidence}}$$



Class 1: Light Image y_1

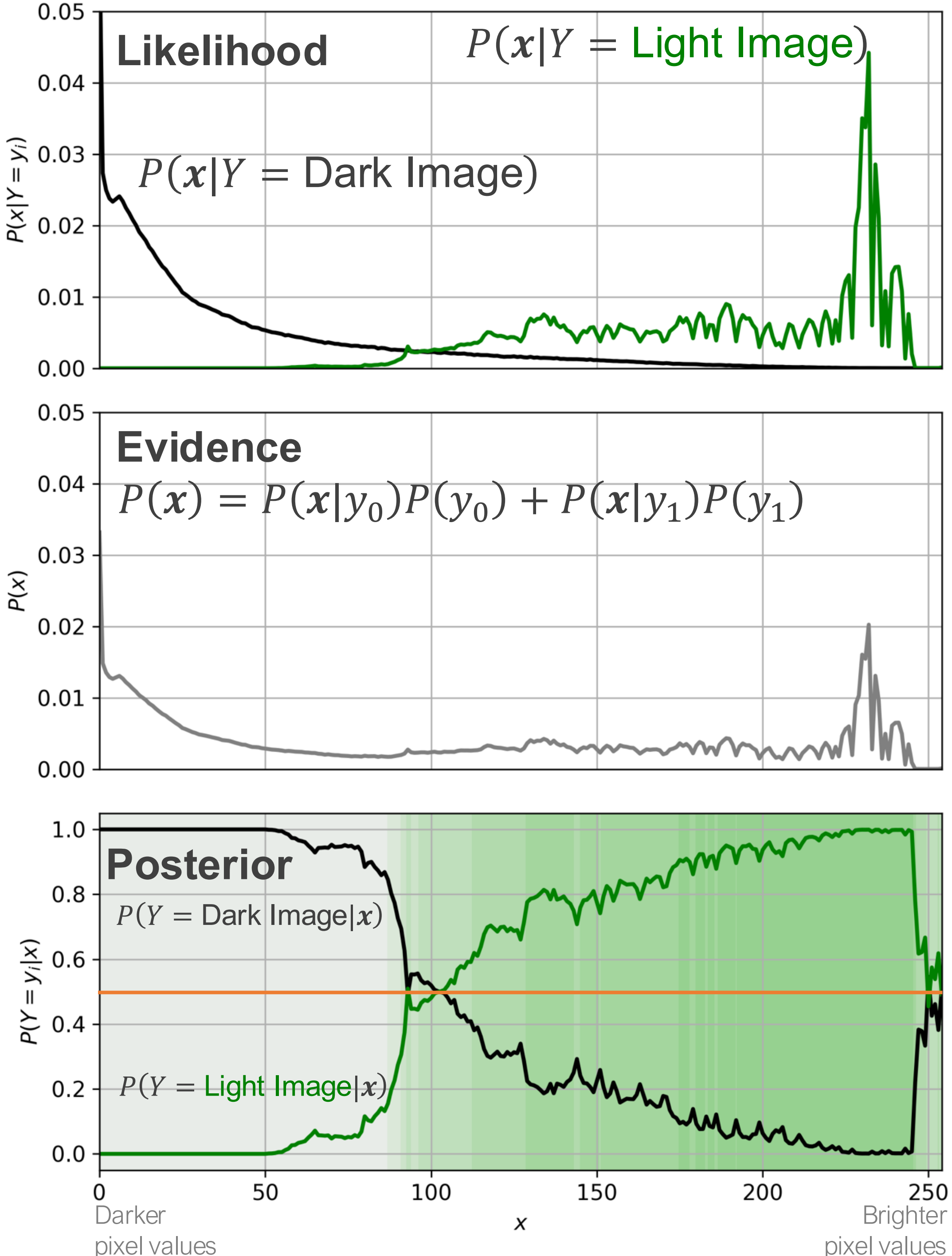
Class 0: Dark Image y_0

Prior: $P(Y = y)$



Bayes' Rule

$$\overset{\text{Posterior}}{P(Y = y|\boldsymbol{x})} = \frac{\overset{\text{Likelihood}}{P(\boldsymbol{x}|Y = y)}\overset{\text{Prior}}{P(Y = y)}}{P(\boldsymbol{x})\ \text{Evidence}}$$



Likelihood

$P(\boldsymbol{x}|Y=$ Light Image$)$

$P(\boldsymbol{x}|Y=$ Dark Image$)$

Evidence

$P(\boldsymbol{x}) = P(\boldsymbol{x}|y_0)P(y_0) + P(\boldsymbol{x}|y_1)P(y_1)$

Posterior

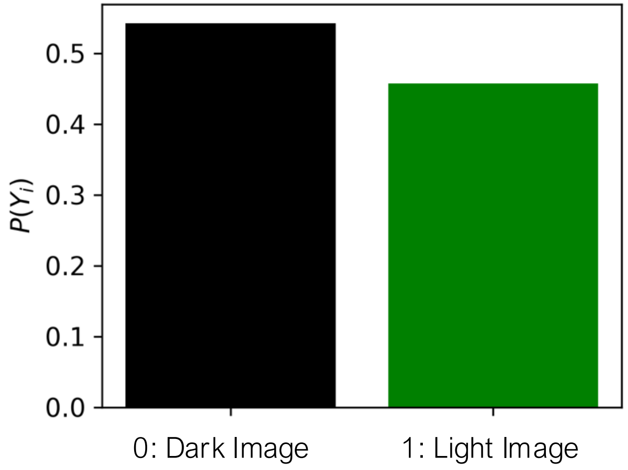
$P(Y=$ Dark Image$|\boldsymbol{x})$

$P(Y=$ Light Image$|\boldsymbol{x})$

Class 1: Light Image y_1

Class 0: Dark Image y_0

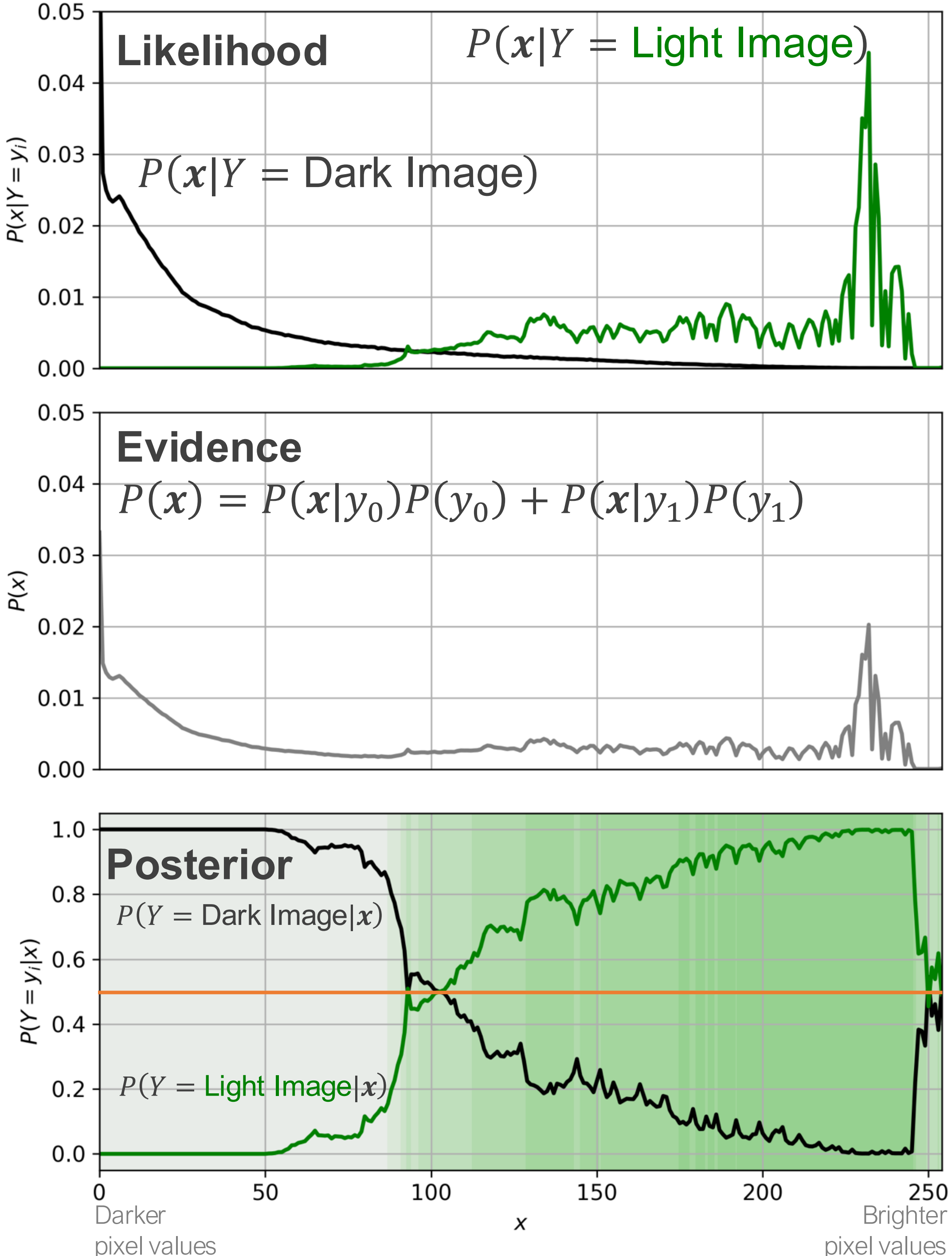
Prior: $P(Y=y)$



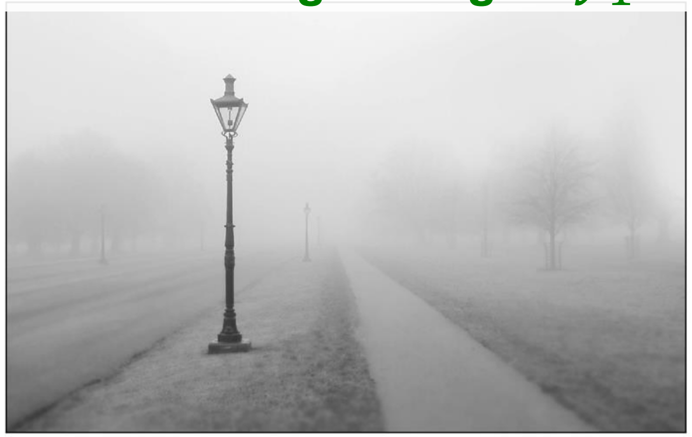
Decision rule:

If $P(Y=$ Light Image$|\boldsymbol{x}) > P(Y=$ Dark Image$|\boldsymbol{x})$ then Light Image

else Dark Image



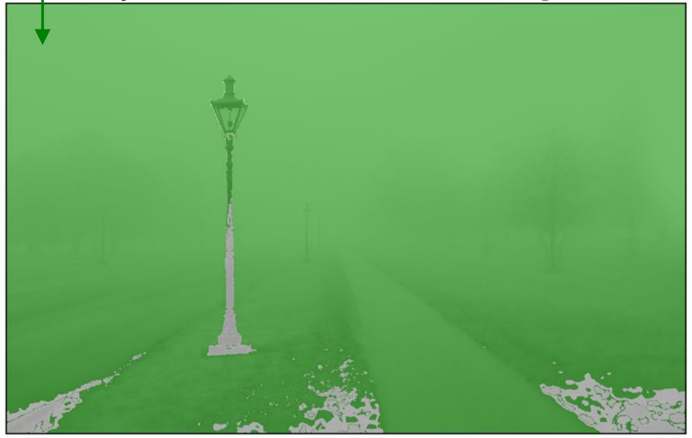
Class 1: Light Image y_1

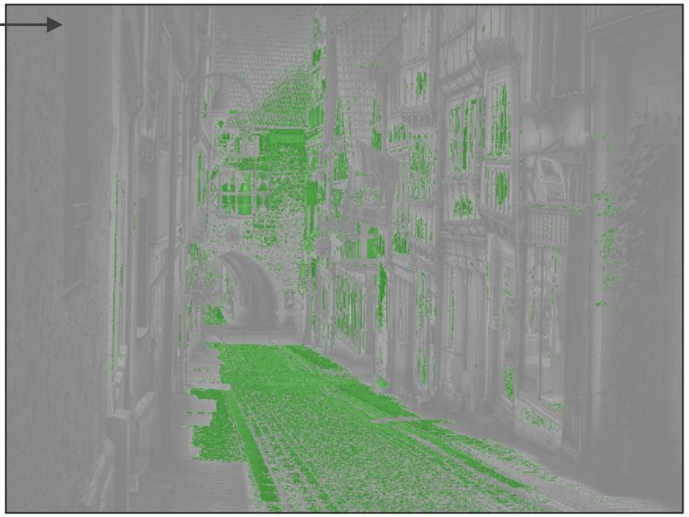


Class 0: Dark Image y_0

Green = classified as from Light Image

Grey = classified as from Dark Image



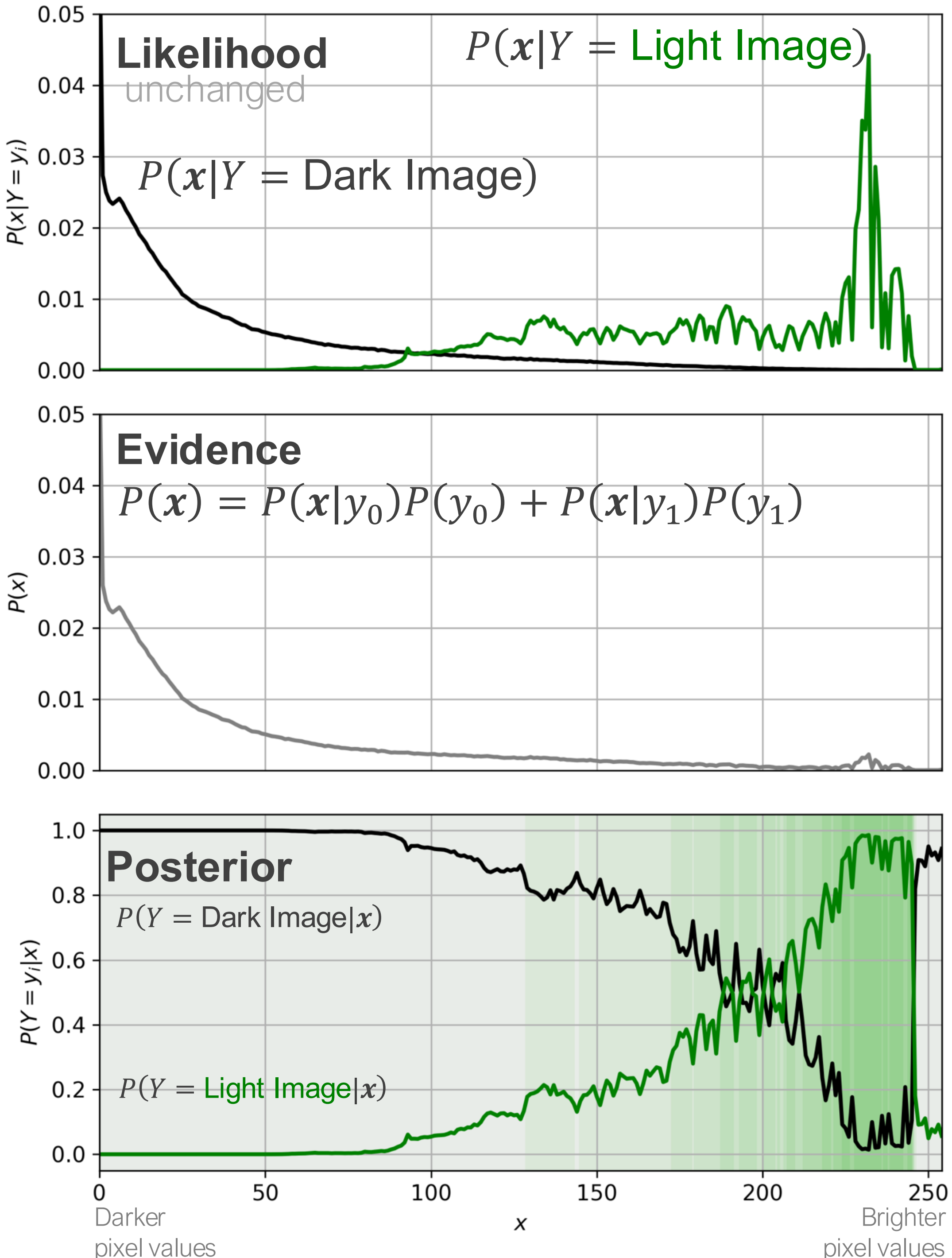


Classifying each of the individual pixels as being either from **Light Image** or **Dark Image** results in classification above

Decision rule:

If $P(Y = \text{Light Image}|\boldsymbol{x}) > P(Y = \text{Dark Image}|\boldsymbol{x})$ then Light Image

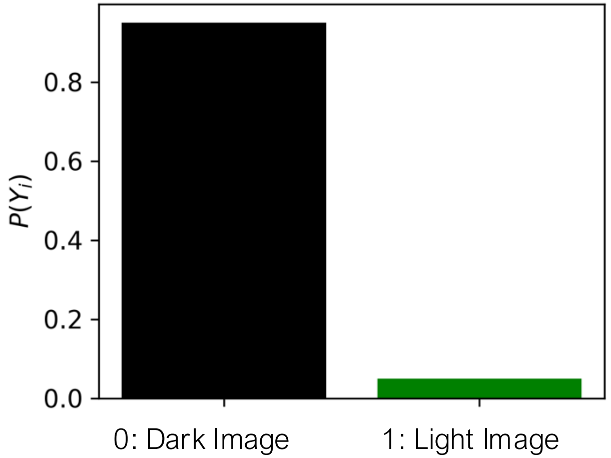
else Dark Image



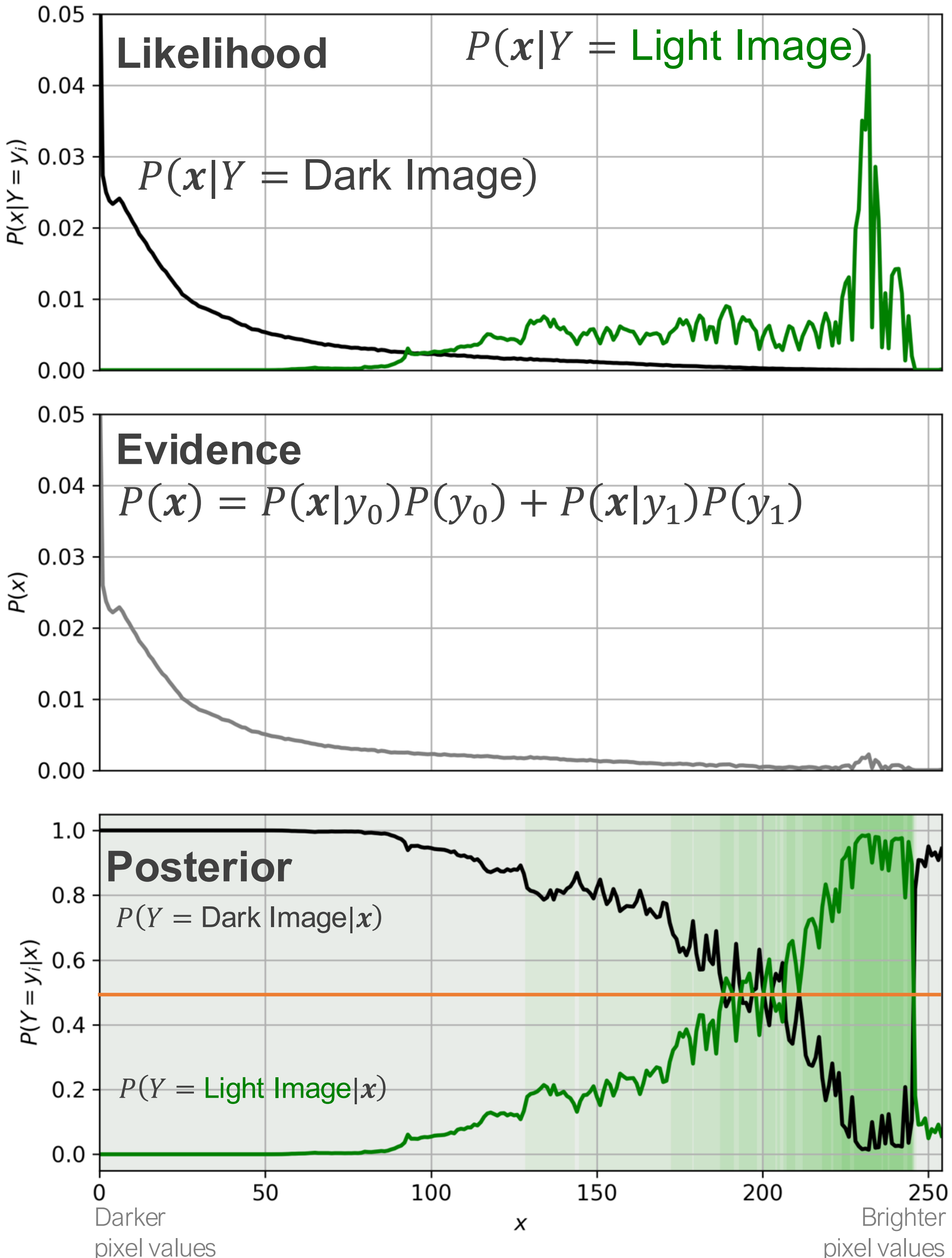
Class 1: Light Image y_1

Class 0: Dark Image y_0

Prior: $P(Y=y)$



Let's assume the sampling of pixels occurred more from the **Dark Image**



Likelihood

$P(\boldsymbol{x}|Y =$ Light Image$)$

$P(\boldsymbol{x}|Y =$ Dark Image$)$

Evidence

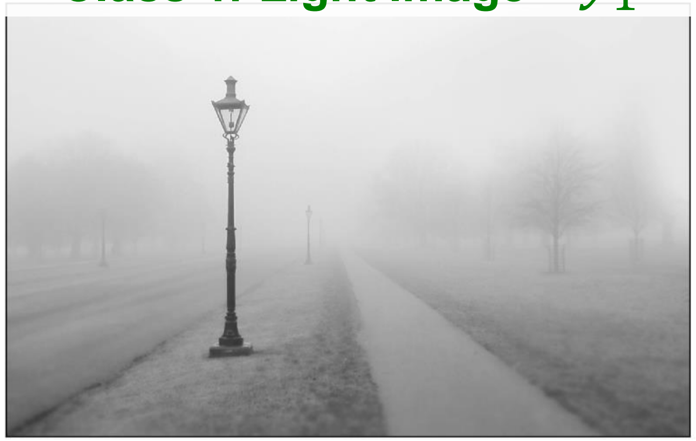
$P(\boldsymbol{x}) = P(\boldsymbol{x}|y_0)P(y_0) + P(\boldsymbol{x}|y_1)P(y_1)$

Posterior

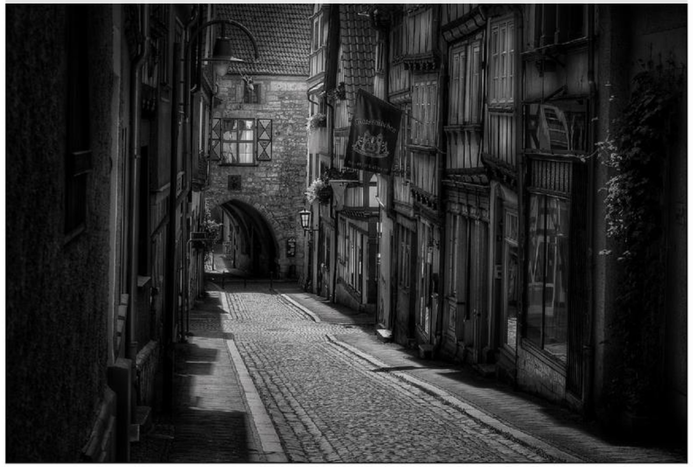
$P(Y =$ Dark Image$|\boldsymbol{x})$

$P(Y =$ Light Image$|\boldsymbol{x})$

Class 1: Light Image y_1

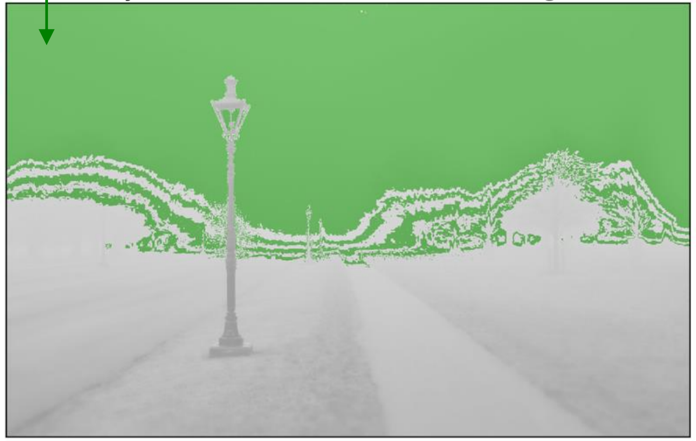


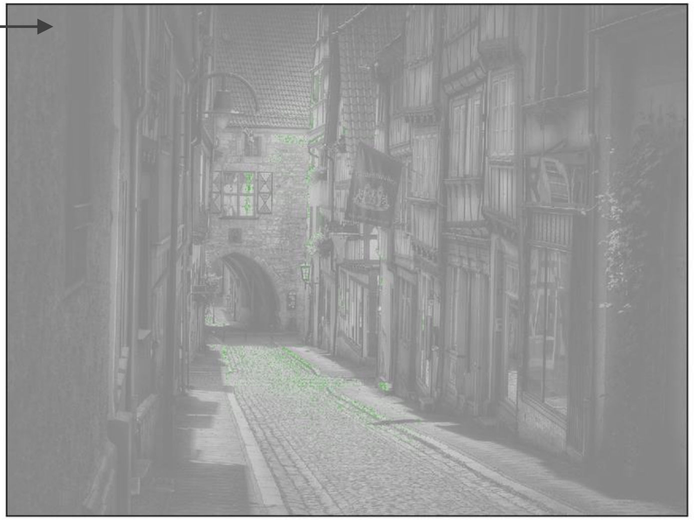
Class 0: Dark Image y_0



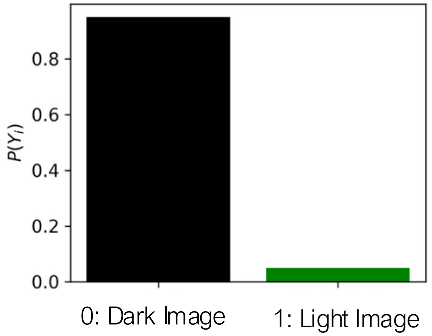
Green = classified as from Light Image

Grey = classified as from Dark Image

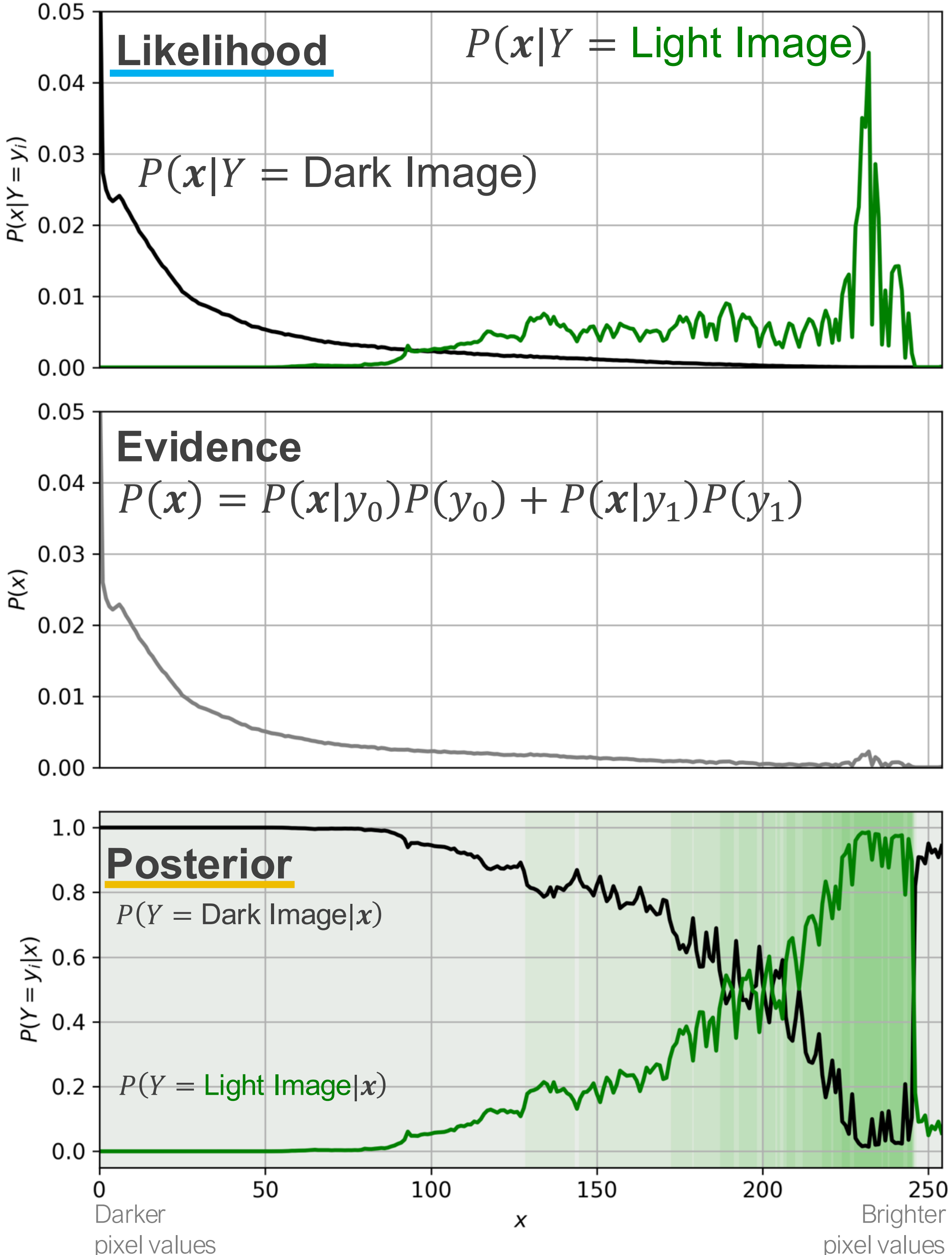




Prior: $P(Y = y)$



Assuming we the sampling of pixels occurred more from the **Dark Image**



Generative models model the **data distributions**

- These can also be used to generate synthetic data
- Can be useful with very small sample sizes

Examples: linear discriminant analysis, naïve Bayes, Gaussian mixture models

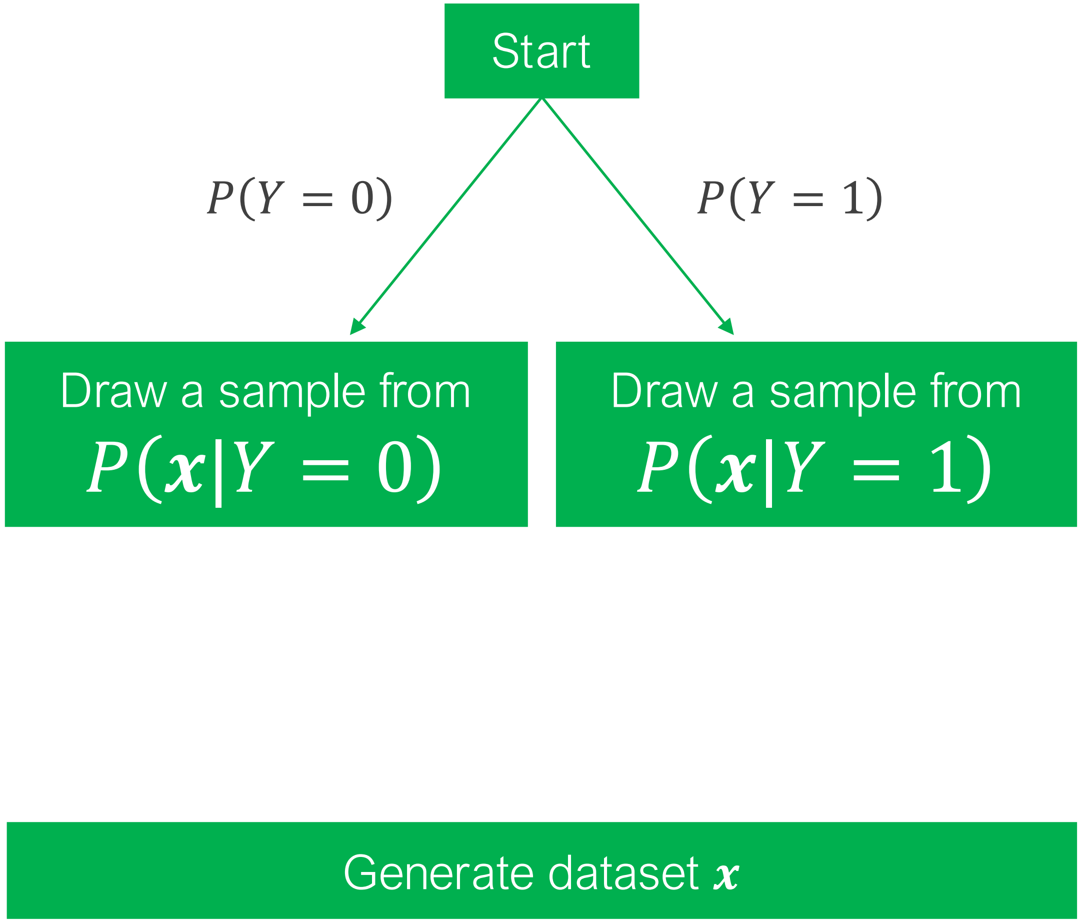
$$\overset{\text{Posterior}}{P(Y=y|\boldsymbol{x})} = \frac{\overset{\text{Likelihood}}{P(\boldsymbol{x}|Y=y)}\,\overset{\text{Prior}}{P(Y=y)}}{\underset{\text{Evidence}}{P(\boldsymbol{x})}}$$

Discriminative models model the **posterior**

- Or they just directly estimate labels without a probabilistic interpretation, $f(\boldsymbol{x}) \rightarrow \boldsymbol{y}$
- Often better performance for large sample sizes

Examples: logistic regression, k nearest neighbors, neural networks

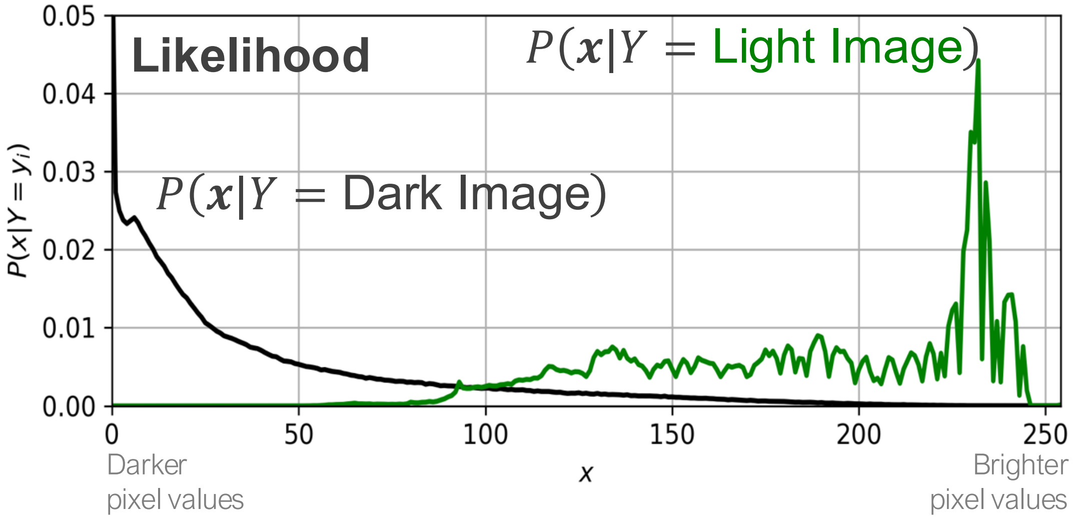
What’s “generative” about this?


Start
$P(Y = 0)$
$P(Y = 1)$
Draw a sample from $P(\boldsymbol{x}|Y = 0)$
Draw a sample from $P(\boldsymbol{x}|Y = 1)$
Generate dataset $\boldsymbol{x}$

1. Randomly choose a class to generate

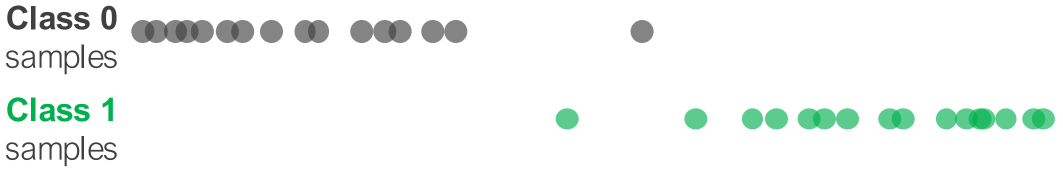
2. Use the class-conditional probability to generate a sample from that class
(this is the critical component for generative models)

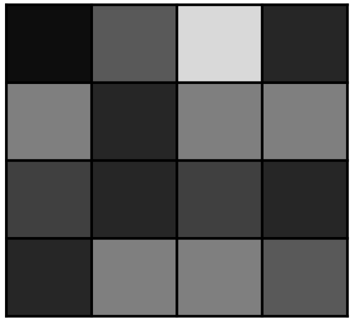
3. Repeat steps 1 and 2 to generate a synthetically-generated dataset



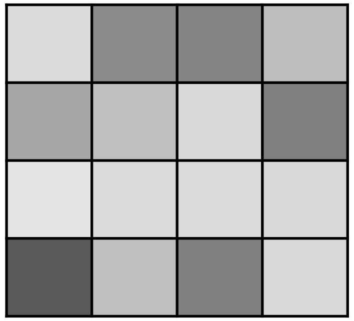
Class 1: Light Image y_1

Class 0: Dark Image y_0



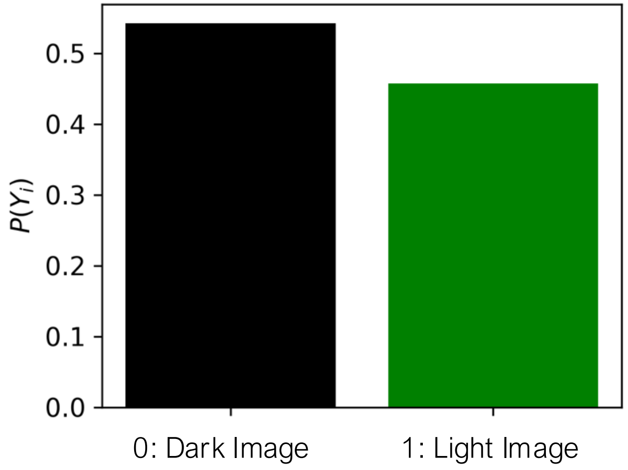


Synthetic "Image" sampled from **Class 0**



Synthetic "Image" sampled from **Class 1**

Prior: $P(Y = y)$



Bayes' Rule

$$\underbrace{P(Y = y|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overset{\text{Likelihood}}{P(\boldsymbol{x}|Y = y)}\overset{\text{Prior}}{P(Y = y)}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$

How to "Generate" Data?

Generative models for classification

Assume we have c different classes

For a new sample, classify it as the class with the **largest posterior** $P(Y = i|\boldsymbol{X} = \boldsymbol{x})$

$$i = 1, \dots, c$$

Generative modeling for classification

If we have c different classes, we define a discriminant function, $d_i(\boldsymbol{x})$ for $i = 1, \dots, c$

If $P(Y = i|\boldsymbol{X} = \boldsymbol{x}) > P(Y = j|X = \boldsymbol{x})$ for all $i \neq j$, then we classify feature $\boldsymbol{x}$ as class i

$$P(Y = i|X = \boldsymbol{x}) = \frac{P(\boldsymbol{X} = \boldsymbol{x}|Y = i)P(Y = i)}{P(\boldsymbol{X} = \boldsymbol{x})}$$

Bayes' Rule: $$\underset{\text{Posterior}}{P(Y|\boldsymbol{X})} = \frac{\overset{\text{Likelihood}}{P(\boldsymbol{X}|Y)}\overset{\text{Prior}}{P(Y)}}{\underset{\text{Evidence}}{P(\boldsymbol{X})}}$$

Denominator is the same for all classes i, so it won't help us tell which class's posterior is higher relative to other classes, so we can leave it out of our discriminant function

We can define the discriminant function as:

$$d_i(x) = P(\boldsymbol{X} = \boldsymbol{x}|Y = i)P(Y = i)$$

If we know the **true likelihood and prior** for our data, this process yields our **Bayes' classifier** (minimum misclassification error classifier)

Generative modeling for classification

$$d_i(x) = P(\boldsymbol{X} = \boldsymbol{x}|Y = i)P(Y = i)$$

We **rarely** know our **true likelihood** for our data so we need to assume a form for the distributions and approximate

1 Assume a form for $P(\boldsymbol{X} = \boldsymbol{x}|Y = i)$

Gaussian → Linear and Quadratic Discriminant Analysis

Gaussian mixture models

Nonparametric density estimates

If we assume **independent features** → **Naïve Bayes**

(remember $d_i(x)$ proportional to the posterior)

2 Assign the class, i, for which $d_i(x)$ is largest

Applies to both binary and multiclass problems

Example: Linear and Quadratic Discriminant Analysis

We build a classifier that models each class as a normal distribution

$$d_i(x) = P(\boldsymbol{X} = \boldsymbol{x} | Y = i) P(Y = i)$$

→ $N(\mu_1, \Sigma_1)$ Assumes the class-conditional likelihoods are normal

→ $N(\mu_0, \Sigma_0)$

For each class, we estimate the class-conditional mean and covariance matrix from the data

Predict the class with the highest $d_i(x)$ over all i for unseen data

Example: Linear and Quadratic Discriminant Analysis

We build a classifier that models each class as a normal distribution

$$d_i(x) = P(\boldsymbol{X} = \boldsymbol{x} | Y = i) P(Y = i)$$

$\longrightarrow N(\mu_1, \Sigma_1)$ Assumes the class-conditional likelihoods are normal

$\longrightarrow N(\mu_0, \Sigma_0)$

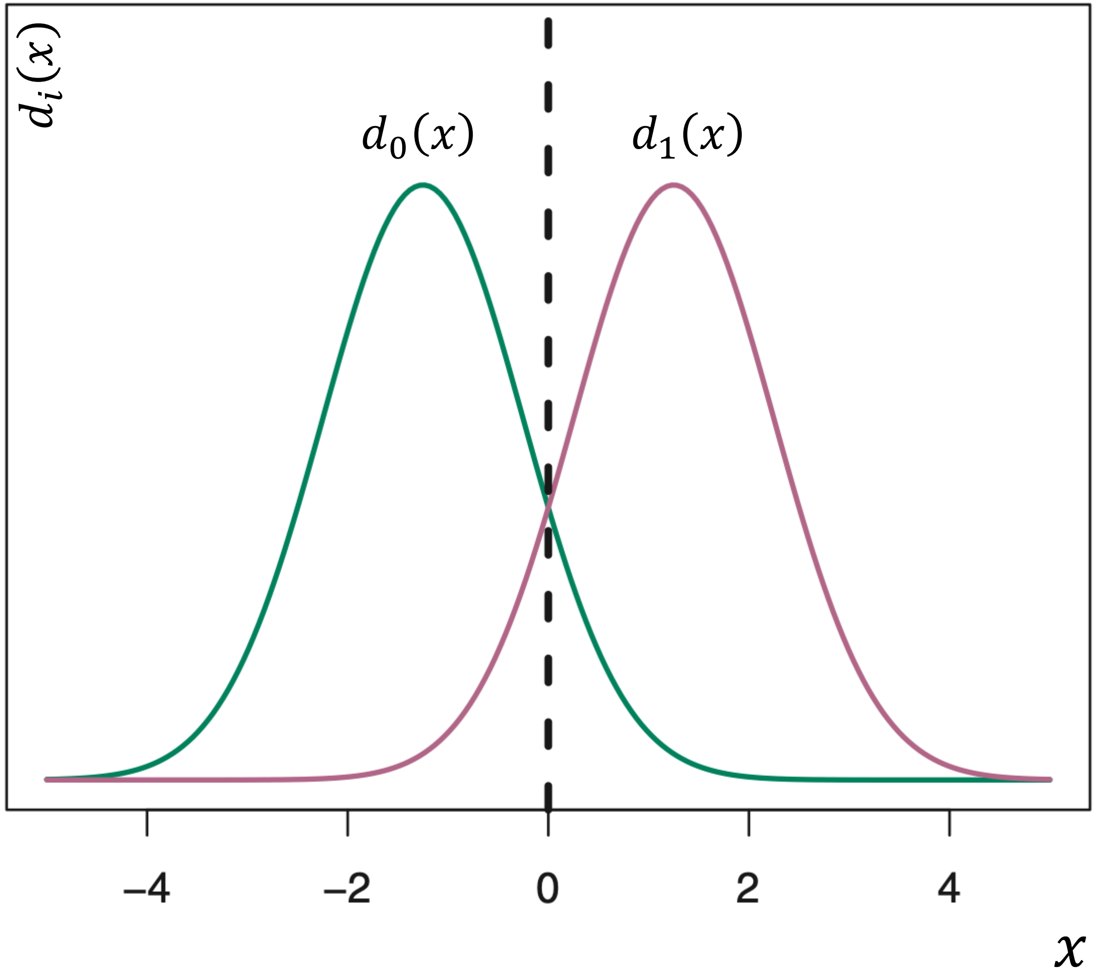
By assuming the class conditional distributions are **Gaussian**, this represents

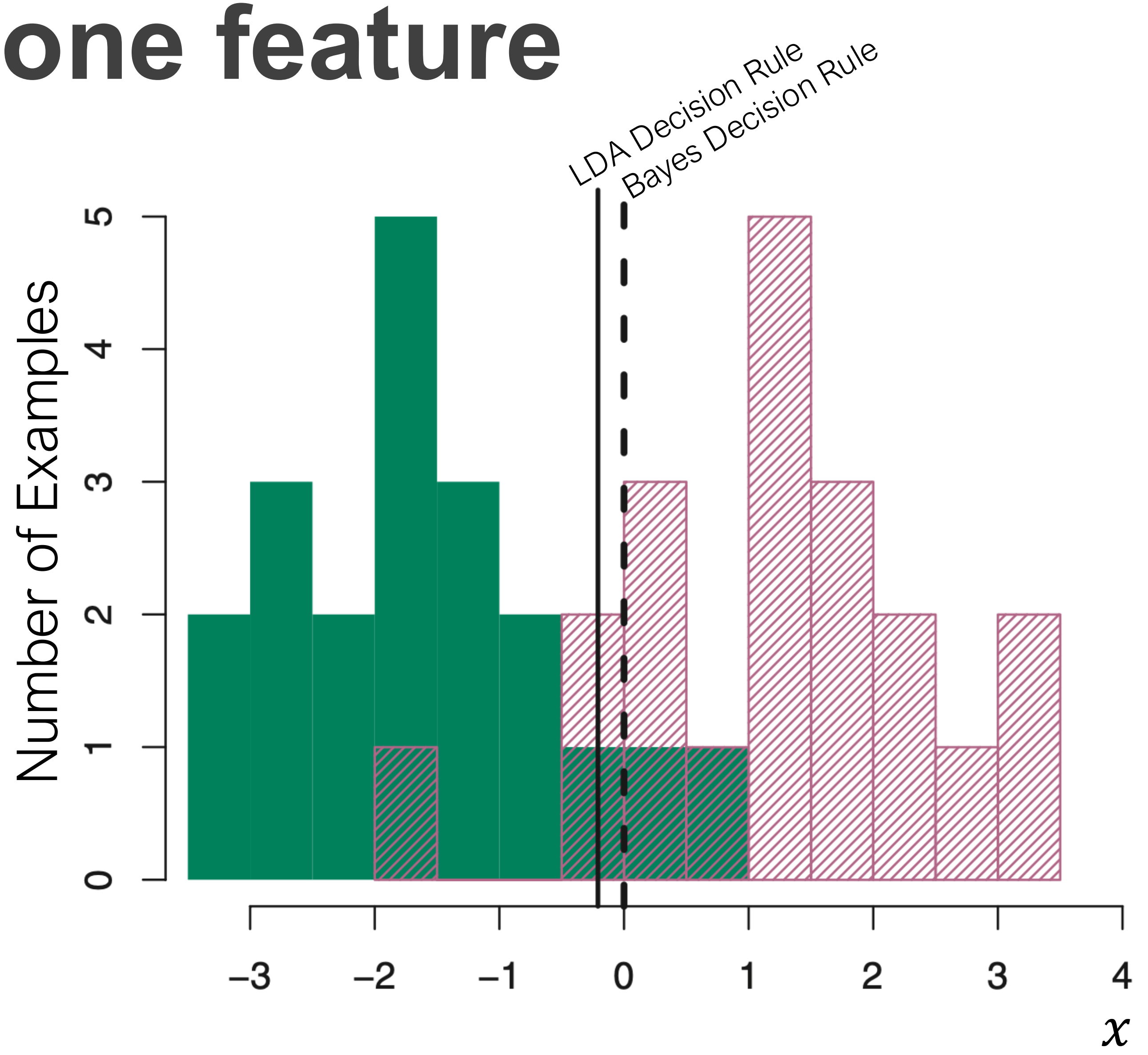
Quadratic Discriminant Analysis

If we further assume the **covariance matrices for each class are the same**, $\Sigma_0 = \Sigma_1$, this represents

Linear Discriminant Analysis

Simple example with one feature





Figures from James et al. - Introduction to Statistical Learning

Example: Linear and Quadratic Discriminant Analysis

We build a classifier that models each class as a normal distribution

$$d_i(x) = P(\boldsymbol{X} = \boldsymbol{x} | Y = i) P(Y = i)$$

$$\rightarrow N(\mu_1, \Sigma_1)$$

$$\rightarrow N(\mu_0, \Sigma_0)$$

Assumes the class-conditional likelihoods are normal

If we assume the class conditional distributions are **Gaussian**, this represents

Quadratic Discriminant Analysis

If we further assume the **covariance matrices for each class are the same**, $\Sigma_0 = \Sigma_1$, this represents

Linear Discriminant Analysis

Covariance matrix

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$$

Vector of observation i

$$x_{ij}$$

Observation index

Predictor index

Predictors →

$$\mathbf{X} = \begin{array}{c} \text{Observations} \end{array} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \rightarrow [D \times D]$$

$$[D \times 1][1 \times D]$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \Sigma_{DD} \end{bmatrix}$$

$$\Sigma_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

$$= \mathrm{cov}(X_j, X_k)$$

$$= E[(X_j - \mu_j)(X_k - \mu_k)]$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \sigma_2^2 & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \sigma_D^2 \end{bmatrix}$$

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2$$

$$= E[(X_j - \mu_j)^2]$$

Mean of each predictor

If $\bar{x}_j = 0$ for all j

This will be the case IF the data are standardized

$$\Sigma_{jk} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}x_{ik}$$

$$= \frac{1}{N}\mathbf{x}_j^T\mathbf{x}_k$$

$$= E[X_j X_k]$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}$$

Covariance and Correlation

Relationship between covariance and correlation

$$\mathrm{corr}(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}$$

When $\mathrm{var}(X) = \mathrm{var}(Y) = 1$, then:

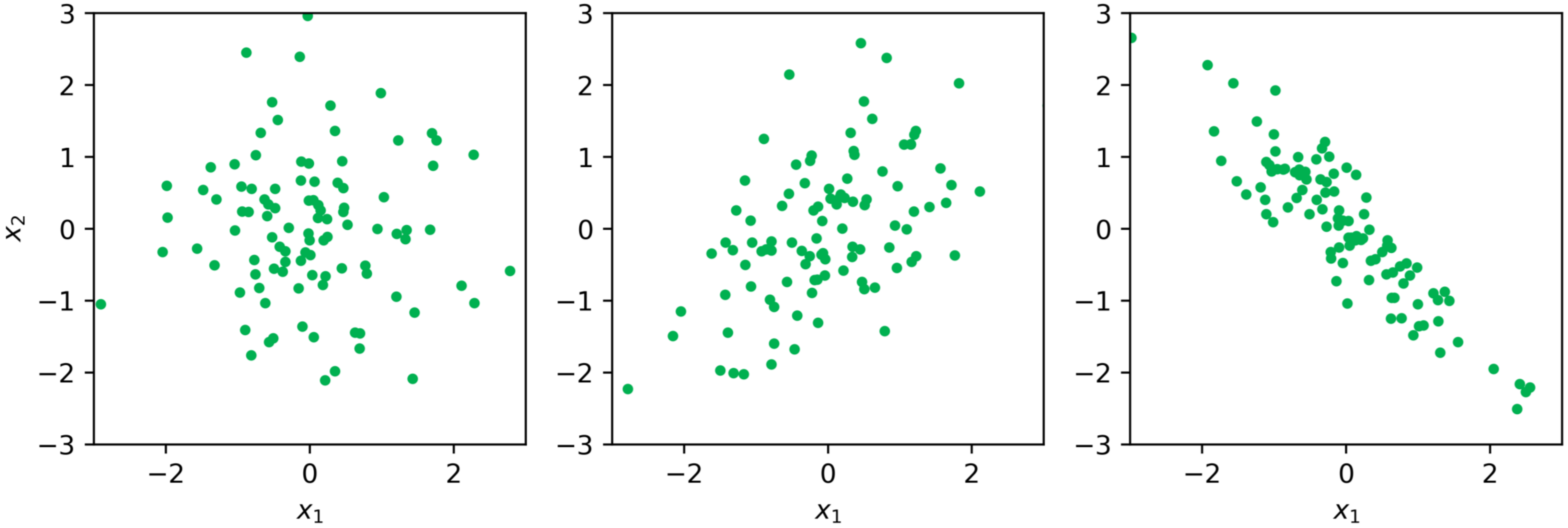
$$\mathrm{corr}(X,Y) = \mathrm{cov}(X,Y)$$

If each of the features have been standardized, this means this matrix is:

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1D} \\ \rho_{21} & 1 & \cdots & \rho_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{D1} & \rho_{D2} & \cdots & 1 \end{bmatrix}$$

Covariance Matrix Examples

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}$$



Key model differences

Quadratic Discriminant Analysis (QDA)

$$\mathbf{\Sigma}_0 \neq \mathbf{\Sigma}_1$$

Each class (e.g., 0 or 1) may have a **unique** covariance matrix

Linear Discriminant Analysis (LDA)

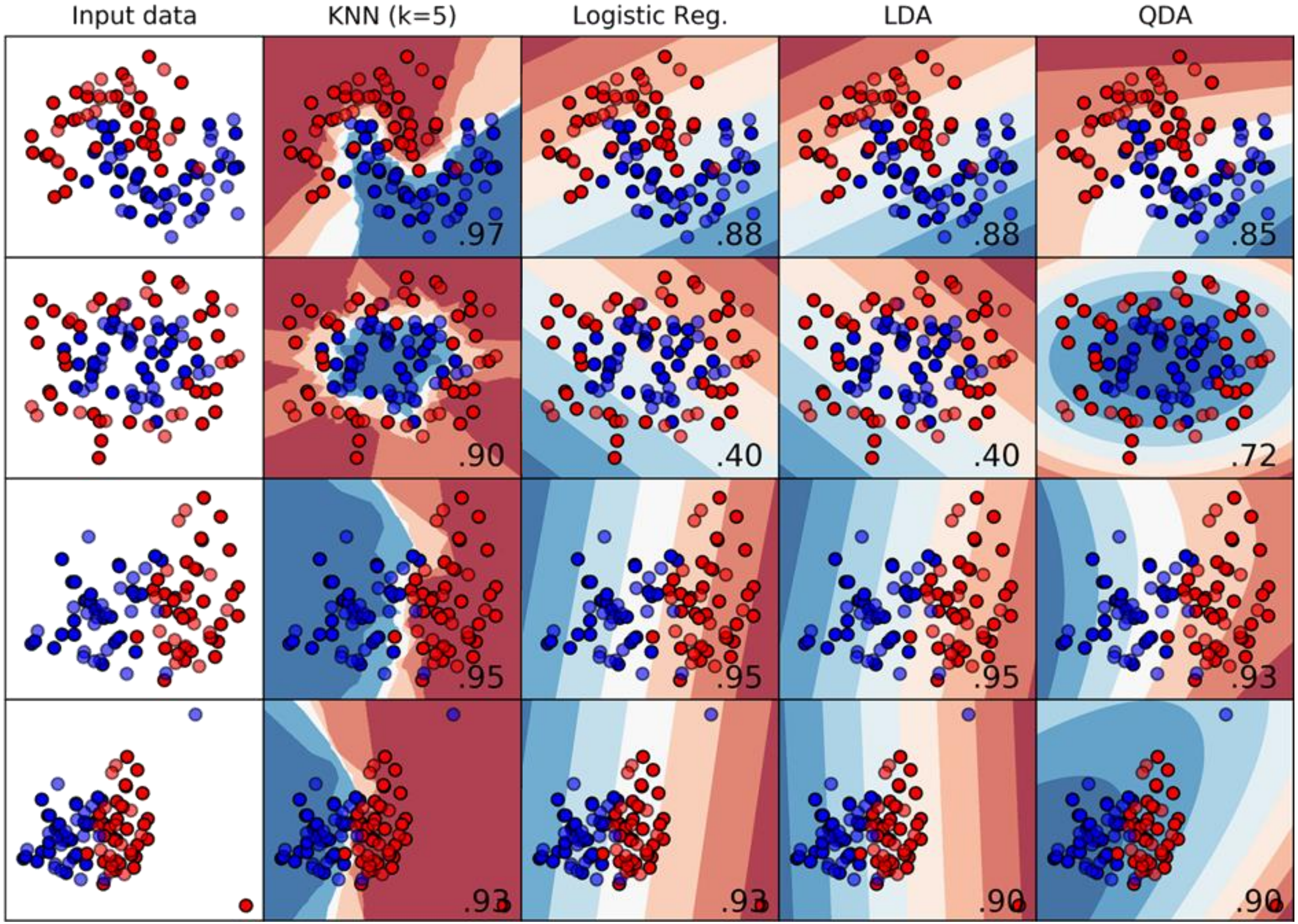
$$\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1$$

Every class (e.g., 0 or 1) has an **identical** covariance matrix

Naïve Bayes with Gaussian Likelihoods

$$\mathbf{\Sigma}_i = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix}$$

Every class (e.g., $i = 0$ or 1) has a **diagonal** covariance matrix


Input data
KNN (k=5)
Logistic Reg.
LDA
QDA
.97
.88
.88
.85
.90
.40
.40
.72
.95
.95
.95
.93
.93
.93
.90
.90

Joint vs Marginal Densities

The marginal densities don't factor in relationships between features

What if the joint density is too hard to estimate?

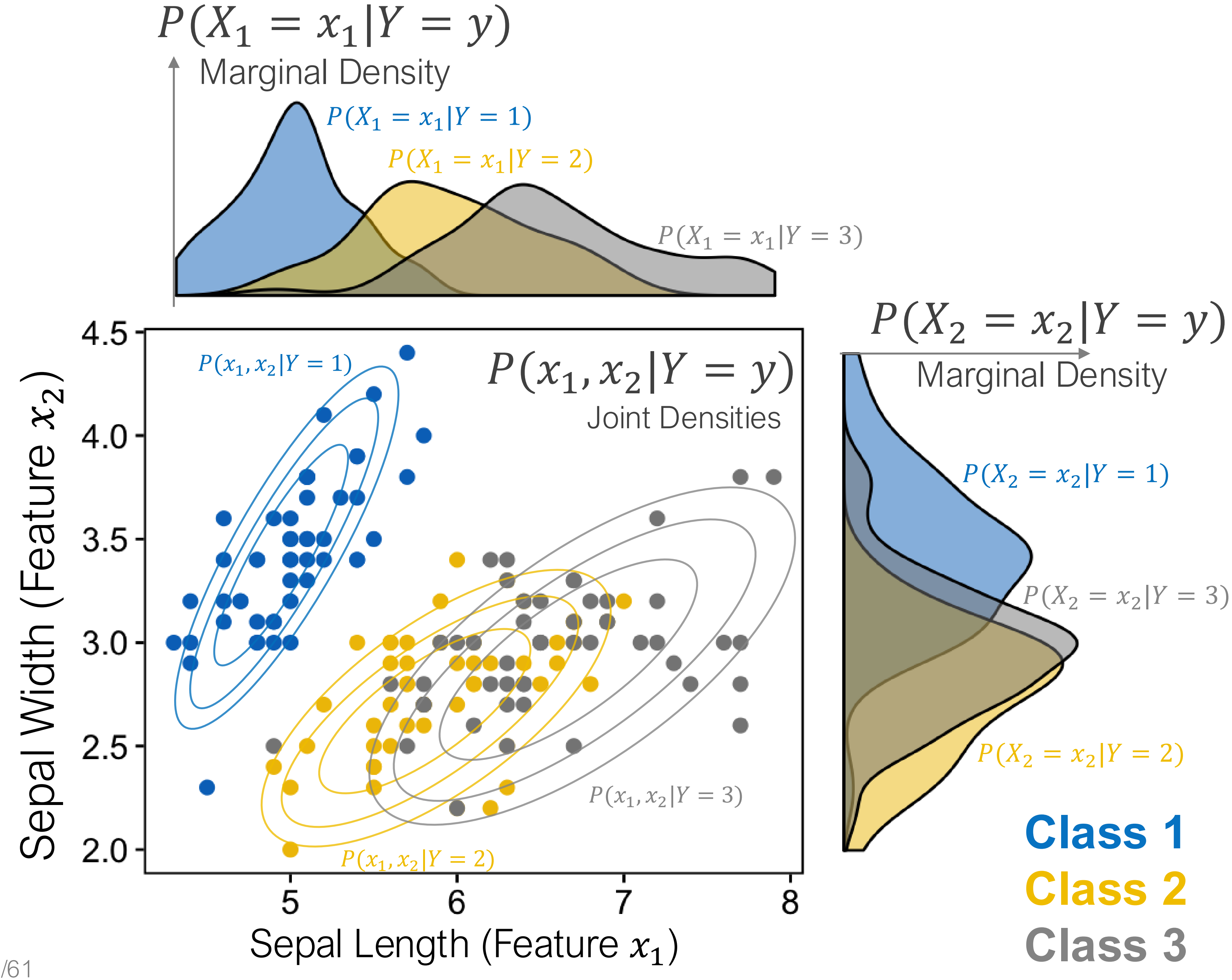


Image adapted from: https://github.com/daattali/ggExtra/issues/61

Naïve Bayes

For independent events: A, B, and C

$$P(A \text{ and } B \text{ and } C) = P(A)P(B)P(C)$$

Start with our original expression for our discriminant function

(proportional to the posterior distribution)

$$d_i(\boldsymbol{x}) = P(\boldsymbol{X} = \boldsymbol{x} | Y = i)P(Y = i)$$

Write out the full expression with all the terms in $\boldsymbol{x}$

(assume p predictors/features)

$$d_i(x_1, x_2, \dots, x_p) = P\left(X_1 = x_1, X_2 = x_2, \dots, X_p = x_p \middle| Y = i\right)P(Y = i)$$

Assumption: Given the class, the features are independent

$$d_i(x_1, x_2, \dots, x_p) = P(Y = i) \prod_{j=1}^{p} P\left(X_j = x_j \middle| Y = i\right)$$

Predict the class with the largest discriminant function
(i.e., largest posterior probability)

Naïve Bayes

We assign the class that has the largest discriminant (i.e. posterior probability)

$$d_i(x_1, x_2, \ldots, x_p) = P(Y = i) \prod_{j=1}^{p} P\left(X_j = x_j \middle| Y = i\right)$$

This implies we estimate the density of each feature **separately**

Considerably simplifies computation and data needs

Is flexible to allow for different distributional forms (i.e. Gaussian) or nonparametric techniques for the likelihood $P\left(X_j = x_j \middle| Y = i\right)$

(assume any form for the distribution you'd like and fit it to each class of your data!)

This independence assumption is a strong assumption that is rarely valid

Naïve Bayes: Gaussian example

We assign the class that has the largest discriminant (i.e. posterior probability)

$$d_i(x_1, x_2, \dots, x_p) = P(Y = i) \prod_{j=1}^{p} P\left(X_j = x_j \middle| Y = i\right)$$

This implies we estimate the density of each feature **separately**

If $P\left(X_j = x_j \middle| Y = i\right)$ is $N(\mu_{ji}, \sigma_{ji}^2)$, for each class we estimate one mean and variance for each of the p features and for each class. We multiply **univariate** distributions together:

$$d_i(x_1, x_2, \dots, x_p) = P(Y = i) \prod_{j=1}^{p} N(\mu_{ji}, \sigma_{ji}^2)$$

Naïve Bayes: Parameters

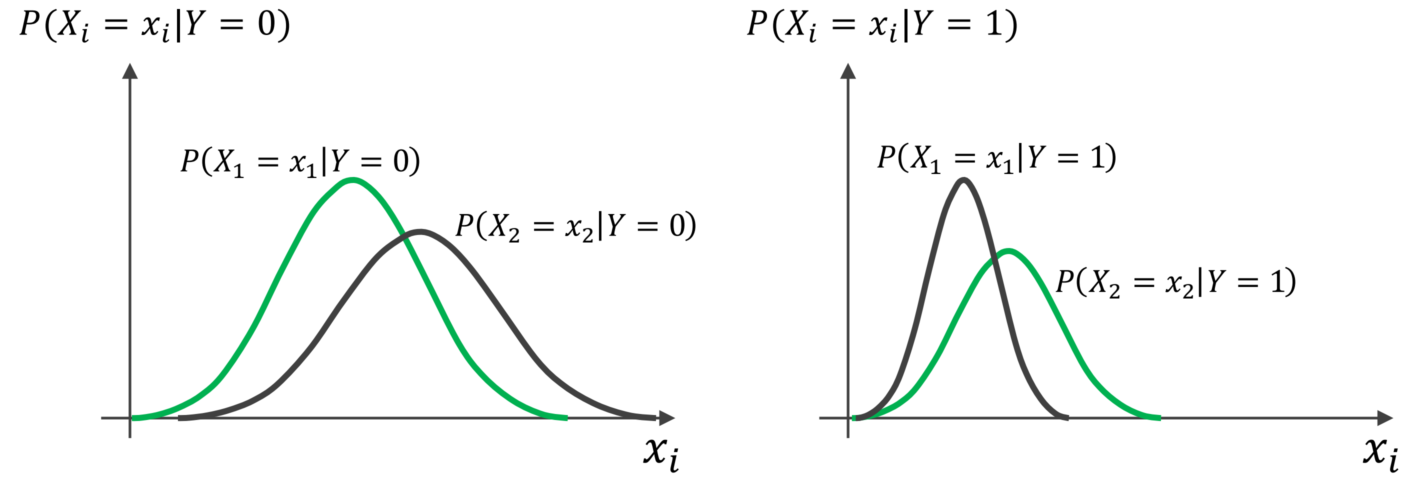
p predictors, c classes

For each predictor, x_i, and class, y_j:

$$(\mu_{ij}, \sigma_{ij}^2)$$

Total parameters = $2cp$

Gaussian Naïve Bayes ($p = 2$)



Without the Naïve Bayes independence assumption, each class would be a multivariate Gaussian with $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

$$\boldsymbol{\mu}_j = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \boldsymbol{\Sigma}_j = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1p}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2p}^2 \\ \vdots & \vdots & & \vdots \\ \sigma_{p1}^2 & \sigma_{p2}^2 & \cdots & \sigma_{pp}^2 \end{bmatrix}$$

Total parameters = $c\left(p + \frac{p^2-p}{2} + p\right)$

Multivariate Gaussian ($p = 2$)

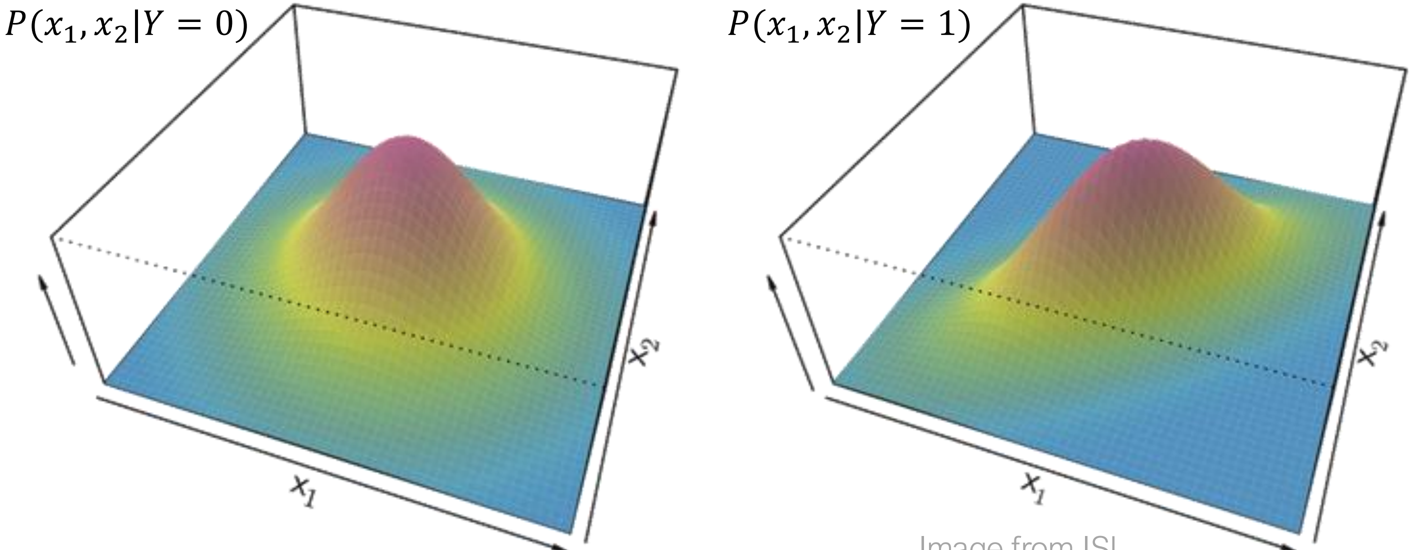
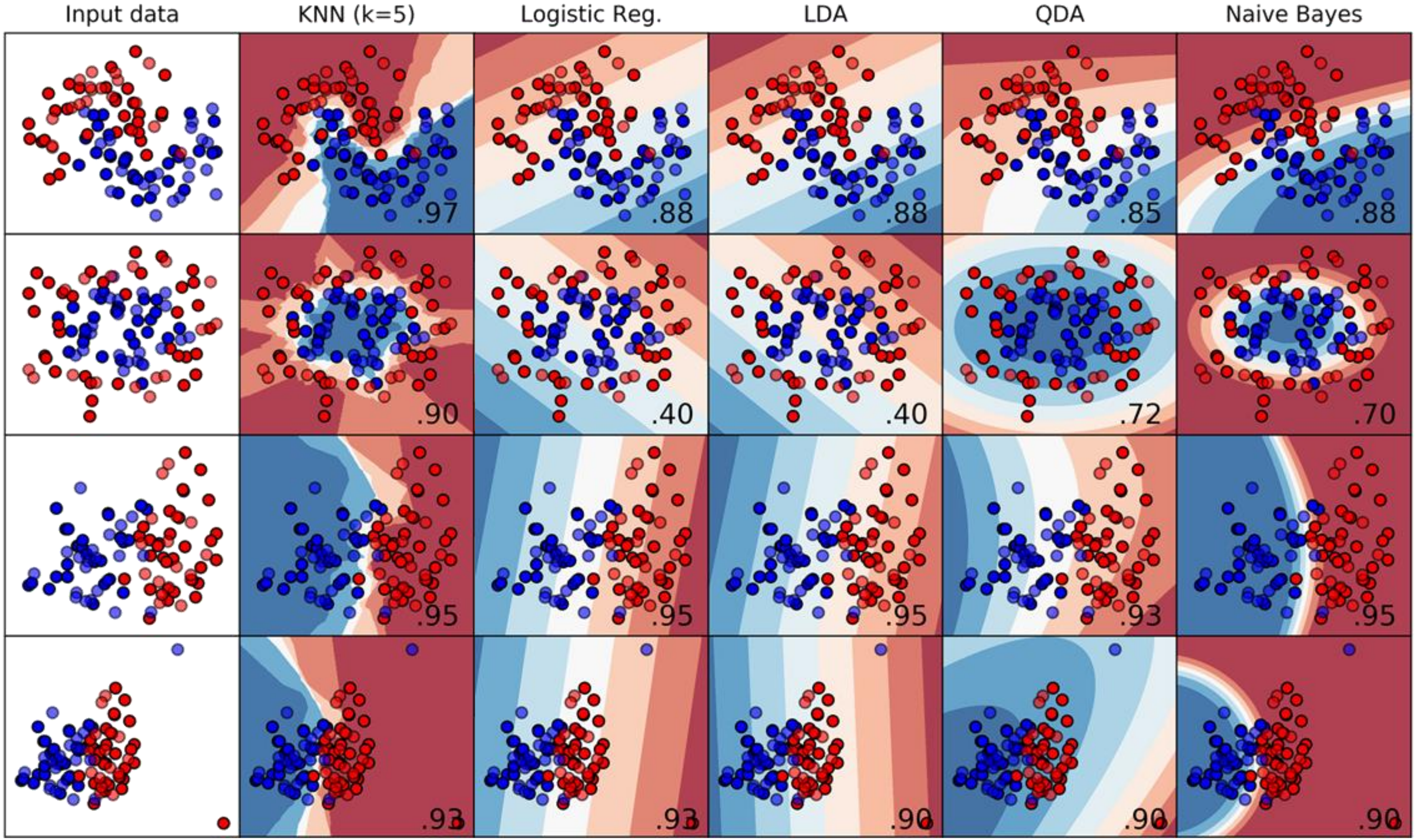


Image from ISL


Input data
KNN (k=5)
Logistic Reg.
LDA
QDA
Naive Bayes
.97
.88
.88
.85
.88
.90
.40
.40
.72
.70
.95
.95
.95
.93
.95
.93
.93
.90
.90
.90

Classifiers

Covered so far

- K-Nearest Neighbors
- Logistic Regression
- Linear/Quadratic Discriminant Analysis
- Naïve Bayes

Often a component of other ML systems
(e.g., finding nearest neighbors in embedding space
finding similar users, data imputation)

Often used as the “end model” after feature engineering or representation learning or as a baseline

Requires small amounts of training data (shines in highly data limited scenarios)

Only choice is the form of P(X|Y) (otherwise no parameter choices)

These each provide foundational context for understanding more advanced ML methods

HOWEVER, these models are limited in their applicability (often find them as baselines) and often used in conjunction with other models

Deep generative models…

Variational Auto Encoders
Normalizing Flows
Generative Adversarial Networks (GANs)
Diffusion Models

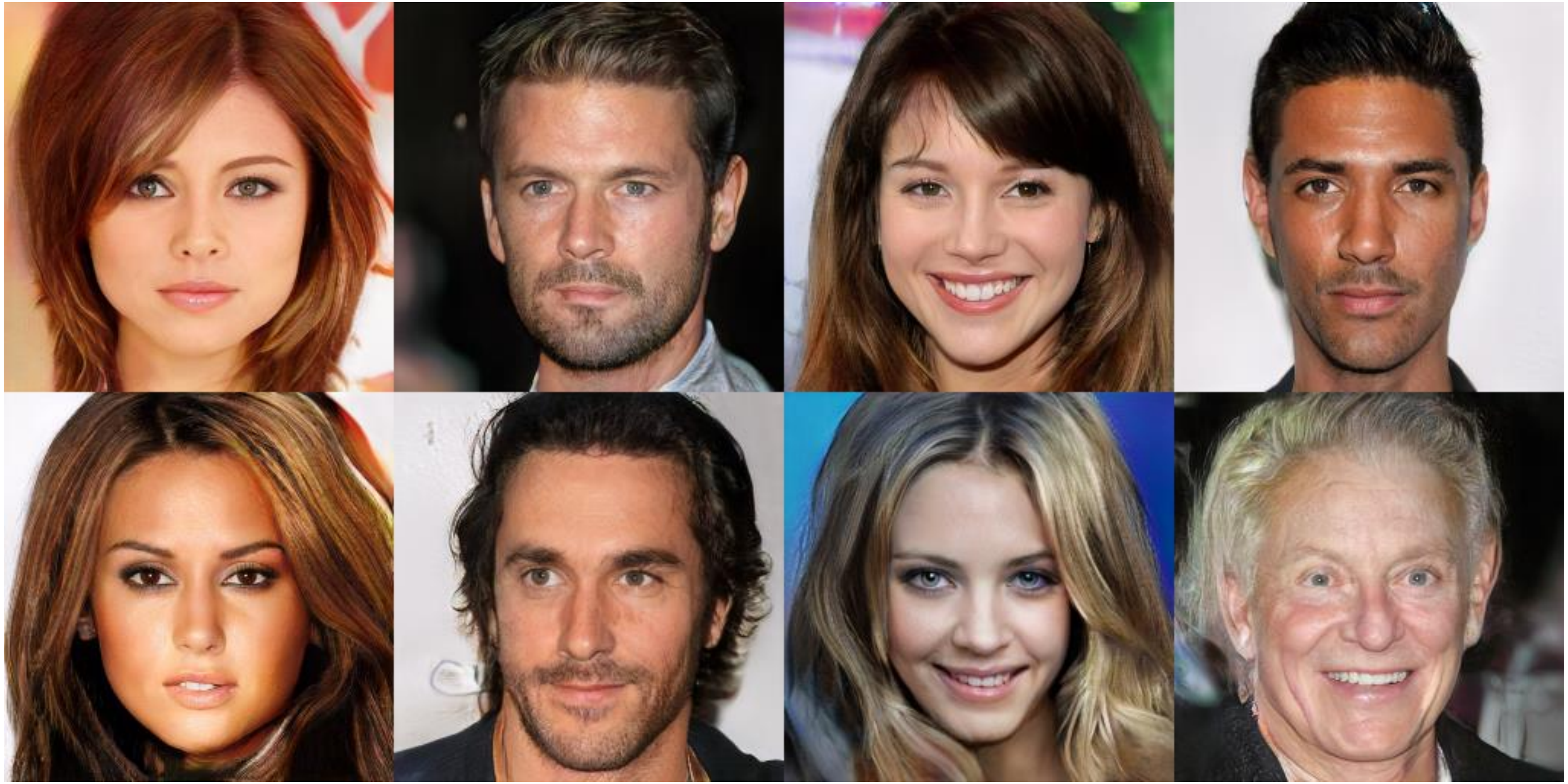
Note: these are not models for classification, purely for generation.
Today's topic builds a foundation for understanding these topics.

Face Synthesis

Image Synthesis (link)

Karras et al. 2018, NVIDIA: Progressive growing of GANS for improved quality, stability, and variation

These images are all synthetic

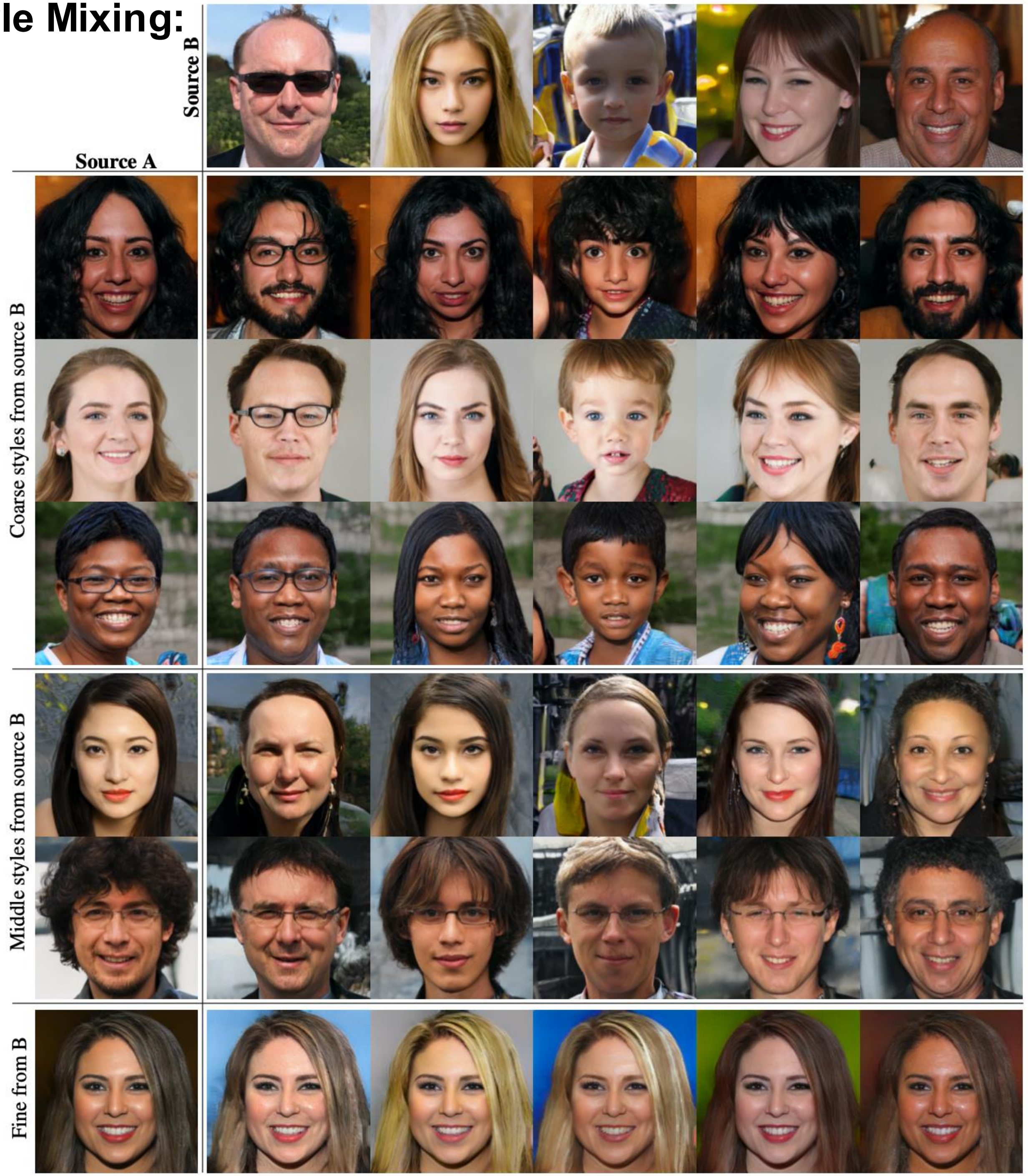


Synthetic Generation

Karras, T., Laine, S. and Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).

Synthetic Images

Style Mixing:

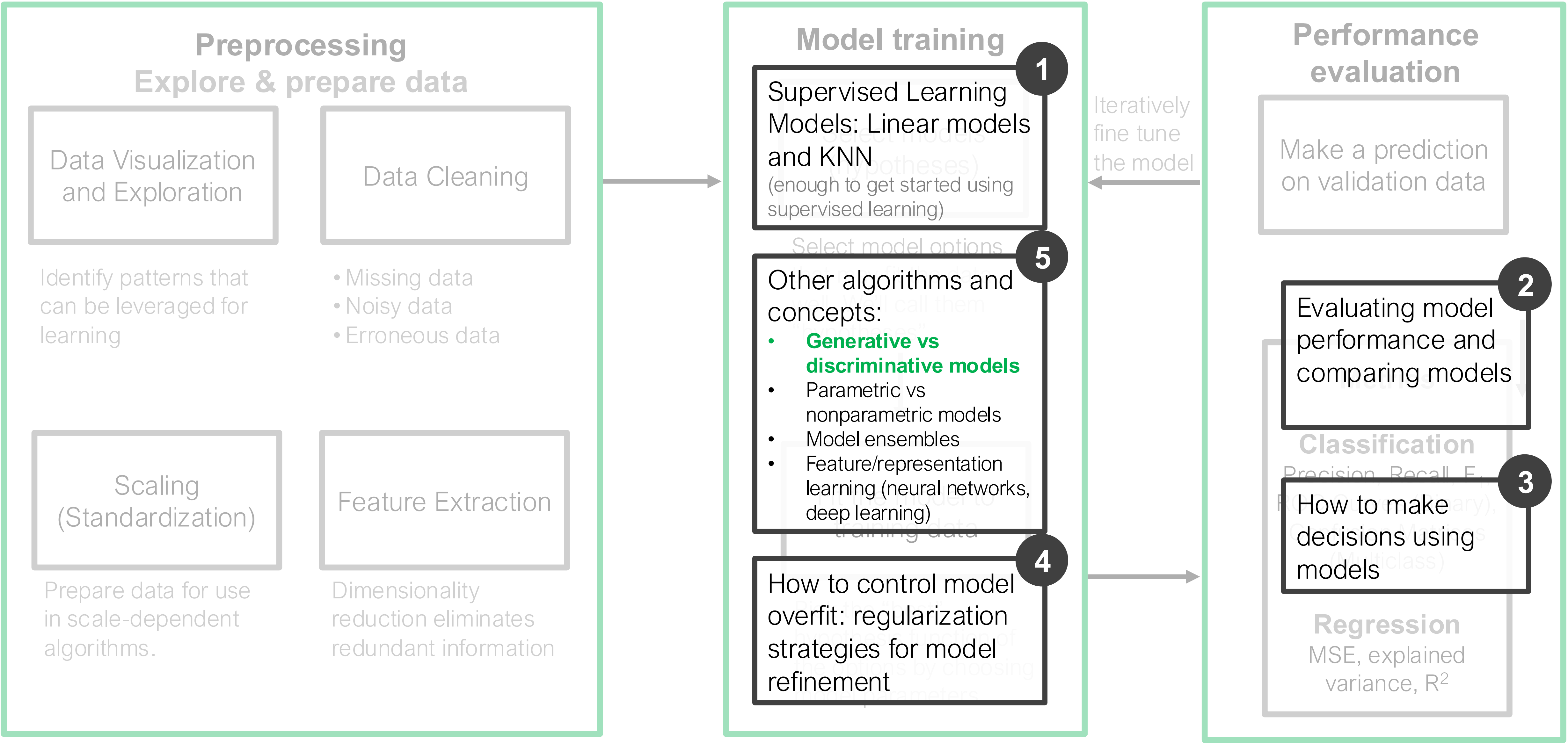


From source B:
Pose, general hair style, face shape, eyeglasses

Hair style, eyes open/closed

Color scheme and microstructure

Supervised learning in practice



Preprocessing
Explore & prepare data

Data Visualization and Exploration

Identify patterns that can be leveraged for learning

Data Cleaning

- Missing data
- Noisy data
- Erroneous data

Scaling (Standardization)

Prepare data for use in scale-dependent algorithms.

Feature Extraction

Dimensionality reduction eliminates redundant information

Model training

1 Supervised Learning Models: Linear models and KNN (enough to get started using supervised learning)

Select model options

5 Other algorithms and concepts:

- **Generative vs discriminative models**
- Parametric vs nonparametric models
- Model ensembles
- Feature/representation learning (neural networks, deep learning)

4 How to control model overfit: regularization strategies for model refinement

Iteratively fine tune the model

Performance evaluation

Make a prediction on validation data

2 Evaluating model performance and comparing models

Classification

3 How to make decisions using models

Regression

MSE, explained variance, R^2