EECS 553 Lecture 03
Machine Learning (ECE)
[class version]

Jeff Fessler

University of Michigan
©Laura Balzano and Jeff Fessler, UM, September 8, 2026, 13:03

F26

Logistics

© 2026 Fessler

- HW1 due Wed. Sep. 9, 11PM
  (24 hour grace period for gradescope.com upload)
- HW1#8a must be submitted to autograder.io
  by Sat. Sep. 12, 11PM (3-day grace period this time).
- HW2 released on my web site. On gradescope soon.
- Piazza: learn by teaching :)
- Sit near (at least) one other classmate!
- CSP seminar by Prof. Lei Ying on Reinforcement Learning
  (Thu. Sep. 10, 3:30pm, 1003 EECS)
- Oct. 20 AI Symposium here in Leinweber

Recap / Outline

© 2026 Fessler

- Recap
  - Probabilistic methods for classification
  - Loss, risk, 0-1 loss, error probability
  - Bayes risk
  - Bayes classifier
- Today
  - Bayes classifier - recap
  - Plug-in classifiers
    - Linear discriminant analysis (LDA)
    - Generative vs discriminative
    - Parametric vs nonparametric
    - Quadratic discriminant analysis (QDA)
    - Naïve Bayes classifier
  - Empirical risk minimization

Bayes classifier illustration

© 2026 Fessler

When $\mathcal{Y} = \{0, 1\}$, the Bayes classifier is:

$$f(\boldsymbol{x}) = \underset{k \in \{0,1\}}{\arg\max}\, \eta_k(\boldsymbol{x}) = \mathbb{I}_{\{\eta_1(\boldsymbol{x}) \geq \eta_0(\boldsymbol{x})\}} = \mathbb{I}_{\{\pi_1 g_1(\boldsymbol{x}) \geq \pi_0 g_0(\boldsymbol{x})\}}$$

$$= \quad$$

Picture:

CQ Bayes classifier: risk examples

© 2026 Fessler

Consider four possible data distributions for labels $\mathcal{Y} = \{0, 1\}$:

1: $g_0 = U[3, 5],\ g_1 = U[3, 5]$

2: $g_0 = U[3, 6],\ g_1 = U[3, 7]$

3: $g_0 = U[3, 5],\ g_1 = U[6, 7]$

4: $g_0 = U[3, 7],\ g_1 = U[4, 5]$

Q03.1 When $\pi_0 = \pi_1 = 1/2$, which data distribution leads to **Bayes risk** ($R^* = 0,\ R^* = \frac{1}{2}$)?
Enter one of these 2-digit numbers: 11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34, 41, 42, 43, 44, or "99" for none of these.

Plug-in classifiers

© 2026 Fessler

- In ML, generally the quantities $\{\pi_k, g_k(\boldsymbol{x}), \eta_k(\boldsymbol{x})\}$ are unknown, so the **Bayes classifier** is unknown.
- Instead, we **plug in** estimates *learned* from training data.
- We plug those estimates into the Bayes classifier formula to get a

Forthcoming examples:

- **linear discriminant analysis** (**LDA**) and **Naïve Bayes** classifiers have the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \hat{\pi}_k \hat{g}_k(\boldsymbol{x}).$$

(Naïve Bayes assumes the features are independent given the label Y.)

- **logistic regression** is essentially a **classifier** that has the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \hat{\eta}_k(\boldsymbol{x}).$$

- Other estimators are also possible.

Linear discriminant analysis: History

© 2026 Fessler

- **Fisher's linear discriminant** (Fisher, 1936) [2] seeks a "linear" classifier between two classes ($K = 2$), *i.e.*, $f(\boldsymbol{x}) = \text{sign}\left(\tilde{f}(\boldsymbol{x})\right)$, where $\tilde{f}$ is a linear (or affine) function of $\boldsymbol{x}$ [1, §9.2.6].
- "Note on discriminant functions" (Welch, 1939) [3] derives Bayes binary classifier when the distributions are known and specifies results for multivariate normal with the same covariance, where it coincides with Fisher's.
- Extensions for more classes with varying assumptions were made in Fisher (1938), Day and Sandomire (1942), Brown (1947), Rao (1952), Anderson (1958), Rao (1963) [4]

Linear discriminant analysis: Perspectives

© 2026 Fessler

- In early 20th century "discriminant" was used in two ways:
  - for generic classification problems without distributional assumptions
  - for hypothesis testing problems between multiple populations from known distributions.
- Today, **linear discriminant analysis** (**LDA**) is sometimes described either as
  - **Fisher's discriminant**
    (a heuristic rule with guarantees in certain Gaussian cases)
  - or as a **plug-in classifier**
    (our treatment) [1, §9.1].

Keep this in mind when reading about LDA elsewhere.

- Supervised classifier, starting with training data of feature vectors and labels: $\mathcal{D}_{\text{train}} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$.
- **LDA model** uses the **categorical distribution** (aka generalized Bernoulli) and conditional **multivariate Gaussian distributions**:

$$Y_1, \ldots, Y_n \overset{iid}{\sim}$$

$$\boldsymbol{X} \mid Y = k \sim$$

for some unknown class mean vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and

- Full data probabilistic model:

$$\mathsf{p}(\mathcal{D}_{\text{train}}) = \underbrace{\prod_{i=1}^{n}}_{} \mathsf{p}(\boldsymbol{x}_i, Y = y_i) = \prod_{i=1}^{n} \mathsf{p}(\boldsymbol{x}_i \mid Y = y_i) \, \mathsf{P}\{Y = y_i\}$$

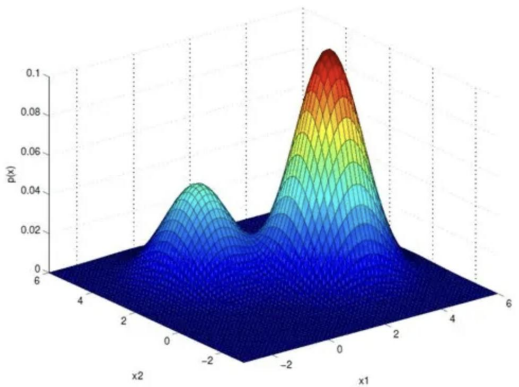
$$= \prod_{i=1}^{n} \mathsf{p}(\boldsymbol{x}_i \mid Y = y_i) \, \pi_{y_i}$$

Specifically, letting $|\boldsymbol{\Sigma}|$ denote the **determinant** of $\boldsymbol{\Sigma}$:

$$g_k(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x} \mid Y = k) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right).$$

- Thus $\mathrm{p}(\mathcal{D}_{\text{train}})$ is a **mixture of gaussians** (**MoG**) model.
- LDA is the **plug-in classifier** based on this model.
- We use training data to estimate
- Figure:

$$\mathrm{p}(\boldsymbol{x}) = \sum_k \pi_k g_k(\boldsymbol{x})$$



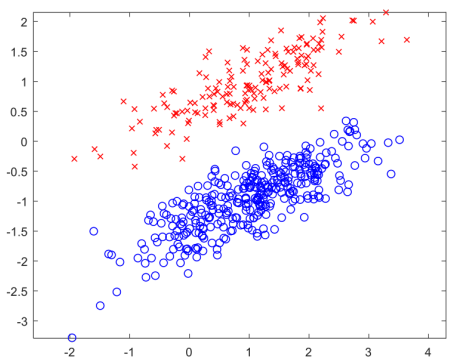


Figure courtesy Farkhod Khushaktov

- **LDA** is the classifier obtained by plugging the following parameter estimates into the **Bayes classifier** formula:

$$\hat{\pi}_k =$$

$$\hat{\boldsymbol{\mu}}_k =$$

$$\widehat{\boldsymbol{\Sigma}} =$$

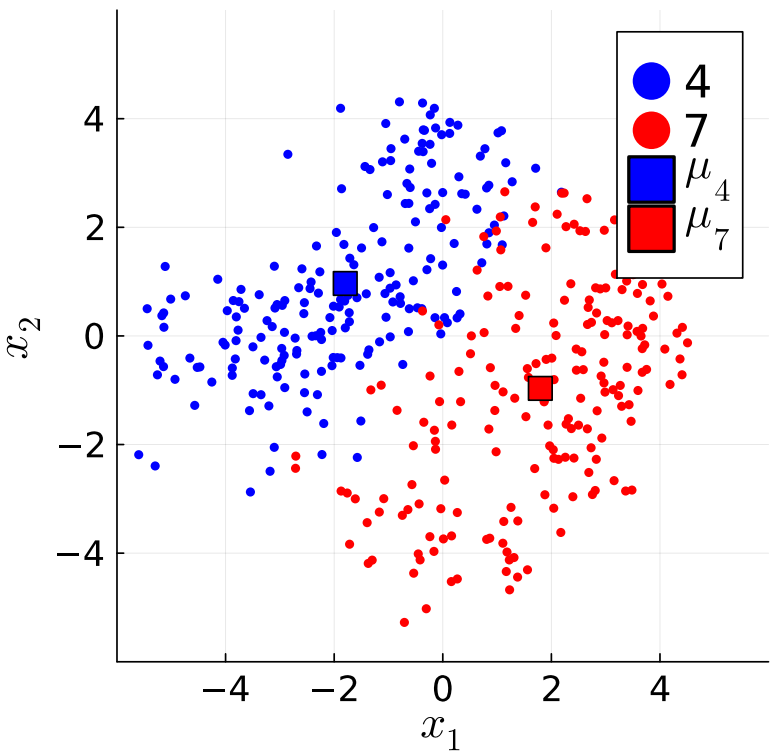
- These estimates are all **maximum likelihood estimates**
  - $\hat{\pi}_k$ is simply proportions (fractions or frequencies) of each class
  - $\hat{\boldsymbol{\mu}}_k$ is the
  - $\widehat{\boldsymbol{\Sigma}}$ is the

LDA: Estimates visualized

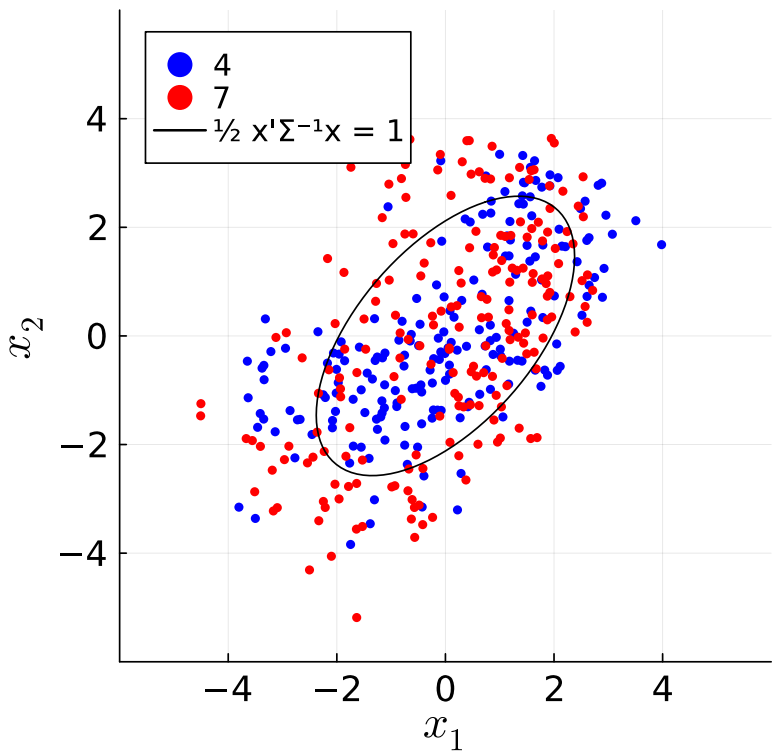
© 2026 Fessler

- Julia LDA demo with same data as k-NN demo.
- Visualize sample means $\{\hat{\boldsymbol{\mu}}_k\}$ and
- Ellipse is one contour line of $\phi(\boldsymbol{x}; \mathbf{0}, \widehat{\boldsymbol{\Sigma}})$, where $\frac{1}{2}\boldsymbol{x}^\top \widehat{\boldsymbol{\Sigma}} \boldsymbol{x} = 1$.

Train data with sample means



Train data de-meaned



- Recall the **Bayes classifier** is

$$f(\boldsymbol{x}) = \underset{k \in \{1,\dots,K\}}{\arg\max} \; \mathrm{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\} = \underset{k}{\arg\max} \, \pi_k \, g_k(\boldsymbol{x}).$$

- Plug-in classifier:

$$f(\boldsymbol{x}) \triangleq \underset{k}{\arg\max} \, \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x})$$

$$= \underset{1 \le k \le K}{\arg\max} \qquad\qquad (1)$$

where LDA's conditional generative model is based on

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^\top \qquad (\boldsymbol{x} - \boldsymbol{\mu}) \right).$$

- One could classify $\boldsymbol{x}$ by evaluating (1) numerically.

CQ sample covariance

© 2026 Fessler

- Plug-in covariance estimate:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{y_i})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{y_i})^\top$$

- Gaussian pdf uses the *inverse* of the covariance matrix.

Q03.2 Having $n \geq d$ is (necessary, sufficient) for $\widehat{\boldsymbol{\Sigma}}$ to be invertible.

A: (T,T) B: (T,F) C: (F,T) D: (F,F) **??**

LDA demo: Linear decision boundary

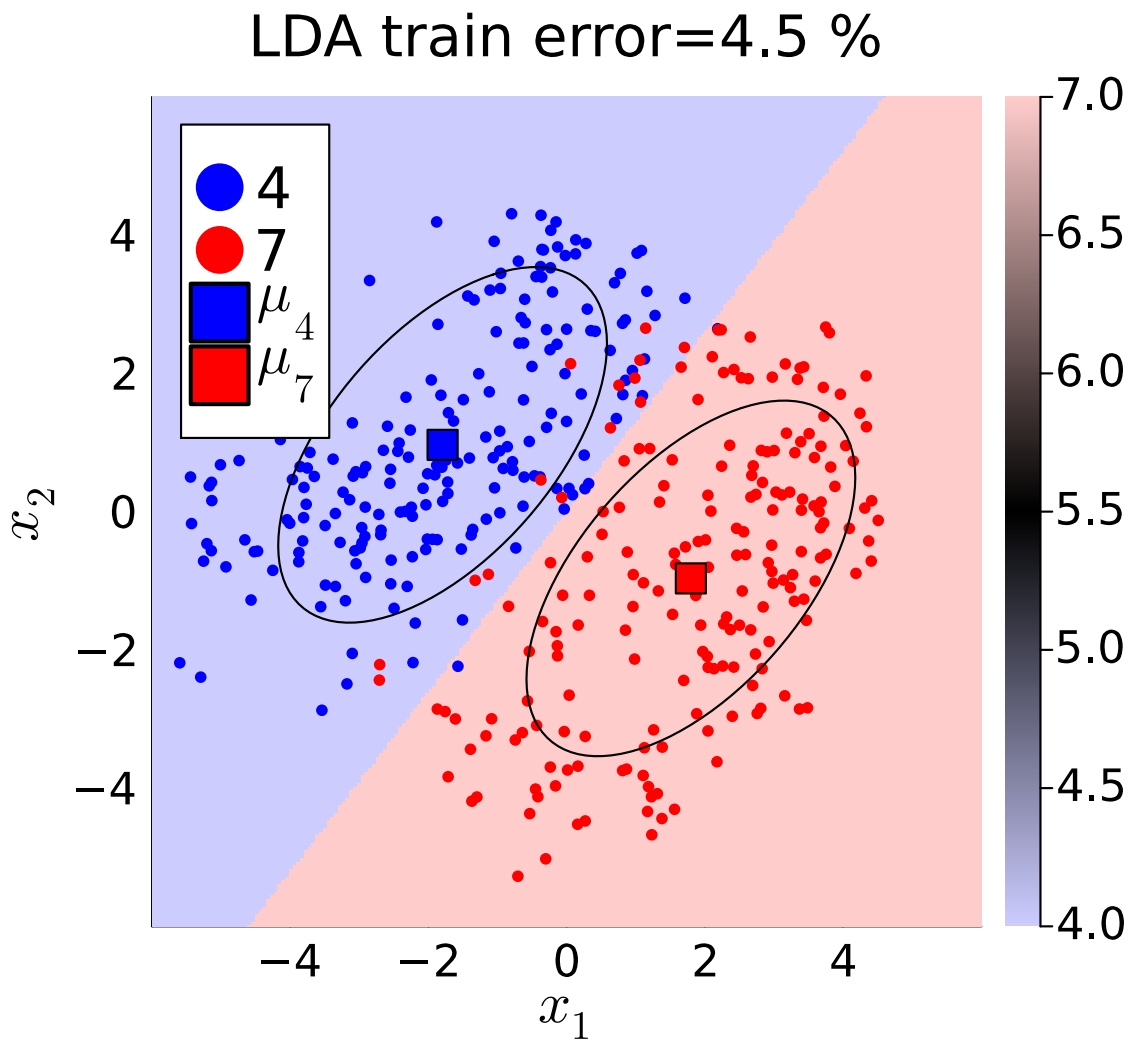
© 2026 Fessler

LDA classifier:

$f(\boldsymbol{x}) = \arg\max_k \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x})$

Julia LDA demo

Next we derive this property for general K.



LDA is a "linear" classifier: General K

© 2026 Fessler

$$f(\boldsymbol{x}) = \arg\max_k \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x}) = \arg\max_k \; \log \hat{\pi}_k + \log \phi(\boldsymbol{x}; \hat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}})$$

where

LDA is a linear classifier: Binary case

© 2026 Fessler

- Binary setting with $Y \in \mathcal{Y} = \{-1, 1\}$.
- Now the LDA classifier simplifies greatly to

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

where

$\boldsymbol{w} =$

$b =$

- Exercise: solve for these formulae by changing argmax to the sign of the difference.

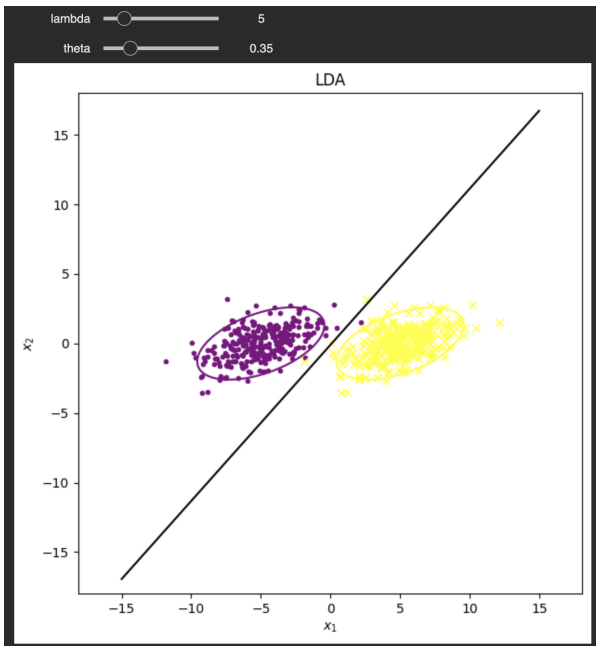
Q03.3 The number of hyperparameters for LDA is:

A: 0 B: 1 C: d D: n E: K

Synthetic data with $\boldsymbol{\mu}_{-1} = \left[\begin{smallmatrix} -5 \\ 0 \end{smallmatrix}\right]$ and $\boldsymbol{\mu}_1 = \left[\begin{smallmatrix} 5 \\ 0 \end{smallmatrix}\right]$ and covariance:

$$\boldsymbol{\Sigma} = \boldsymbol{R} \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{R}^\top, \qquad \boldsymbol{R} \triangleq \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

- If $\boldsymbol{\Sigma}$ is multiple of identity, decision boundary is orthogonal (with respect to the dot product) to $\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_{-1}$.
- Otherwise, decision boundary is skewed by covariance matrix.



Python colab demo with interactive λ and θ.

Generative vs discriminative

© 2026 Fessler

- A machine learning algorithm is **generative** if it is based on a full probabilistic model of the observed data:
  - Assume some functional form or probabilistic model for $\{\pi_k, g_k(\boldsymbol{x})\}$.
  - Estimate parameters of $\pi_k, g_k(\boldsymbol{x})$ from training data.
  - Use plug-in and **Bayes' rule** to calculate/estimate $\eta_k(\boldsymbol{x})$ from $\{\hat{\pi}_k, \hat{g}_k\}$.
- An algorithm is **discriminative** if it assumes a partial or no probabilistic model.
  - Assume some functional form or probabilistic model for $\eta_k(\boldsymbol{x})$.
  - Estimate (*i.e.*, learn) $\eta_k(\boldsymbol{x})$ directly from training data.
- In both cases, classify $\boldsymbol{x}$ using $\arg\max_k \hat{\eta}_k(\boldsymbol{x})$.
- See [1, §9.4] [5] for discussions.

Parametric vs nonparametric

© 2026 Fessler

- A learning algorithm is **parametric** if the number of model parameters* is independent of sample size n.
- An algorithm is **nonparametric** if the number of model parameters* grows with sample size n.
- *Model parameters = parameters needed to represent the function (=model) produced by the learning algorithm
- Example.
  Using a histogram as a probability **density estimator** is:
  - **parametric** if the number of bins is
  - **nonparametric** if the **number of bins**

CQ: LDA properties

© 2026 Fessler

Q03.4 The LDA classifier is

A: generative & parametric

B: generative & nonparametric

C: discriminative & parametric

D: discriminative & nonparametric

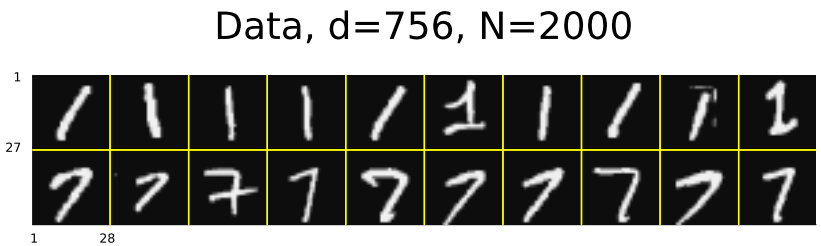
E: none of these

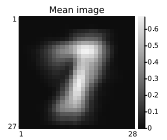
Q03.5 The LDA classifier always exists for any $\mathcal{D}_{\text{train}}$ with $n > d$.

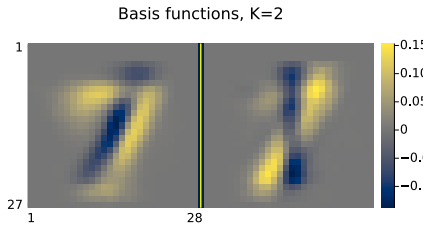
A: True B: False **??**

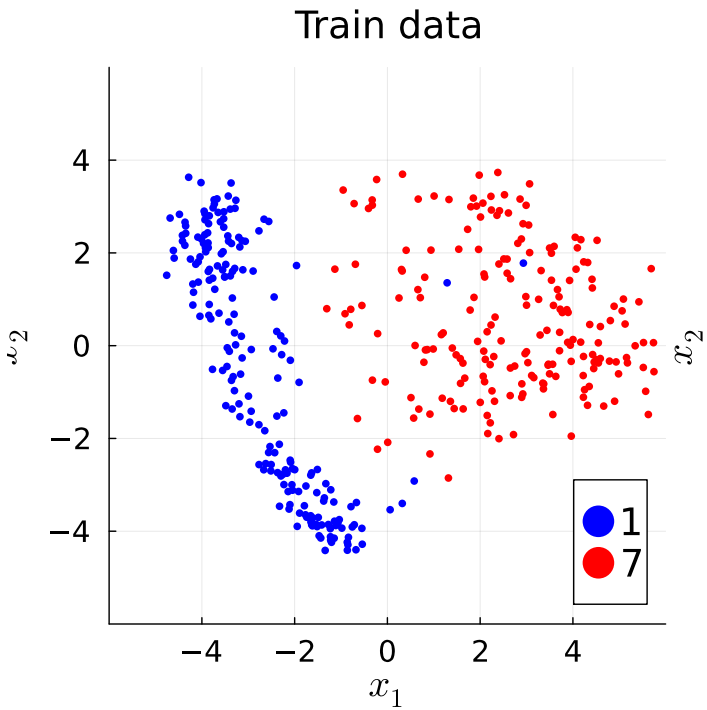
Classifier demo for digits 1,7: Data

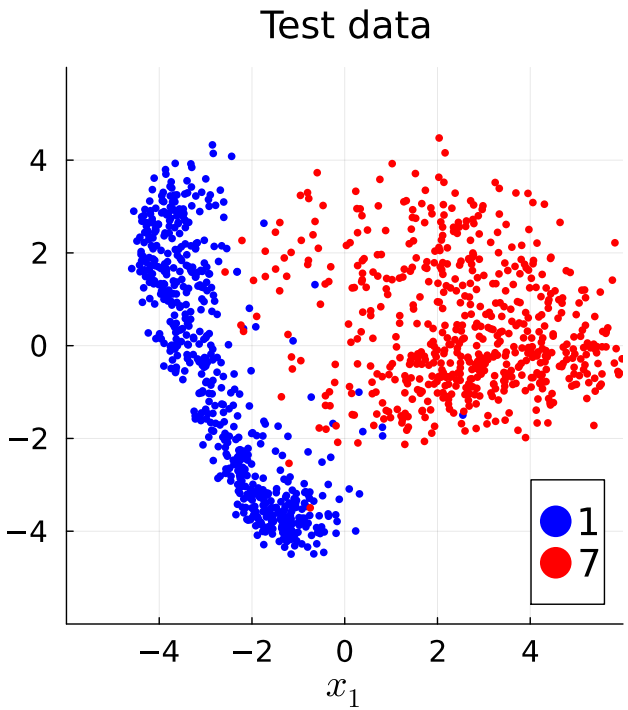
© 2026 Fessler



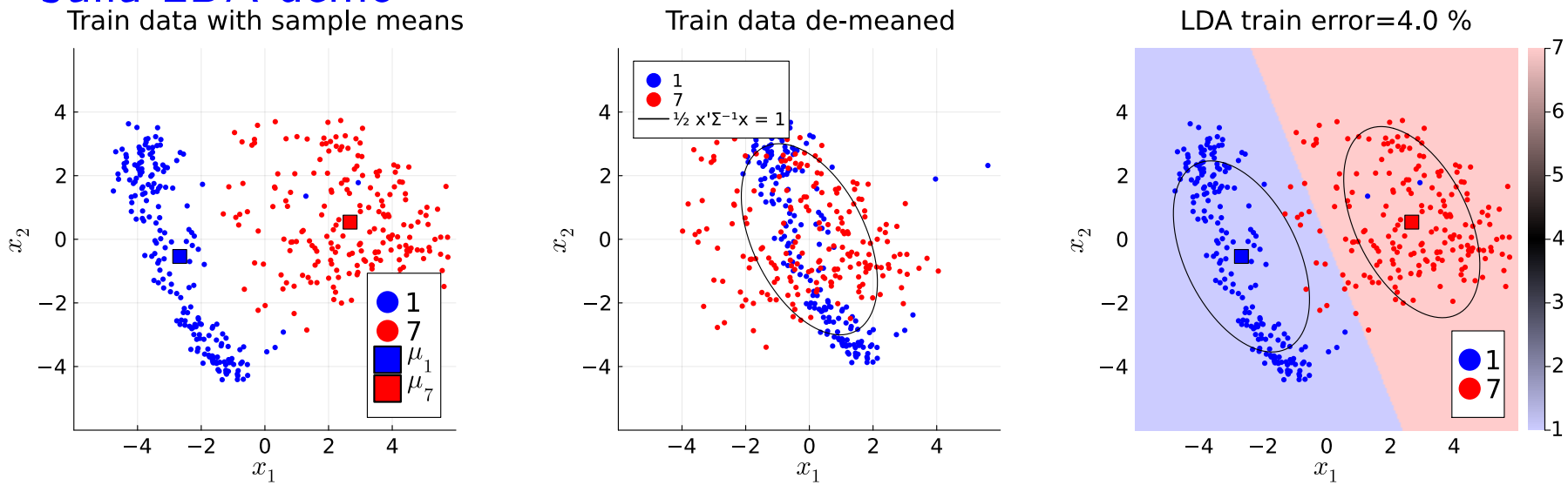








Julia LDA demo



The covariance matrices $\boldsymbol{\Sigma}_k$ differ *greatly* between the two classes.

Quadratic discriminant analysis: Data model

© 2026 Fessler

- Training data $\mathcal{D}_{\text{train}} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$.
- **quadratic discriminant analysis** (**QDA**) probability model:

$$\begin{aligned} \text{labels:} \quad & Y_1, \ldots, Y_n \overset{iid}{\sim} \text{discrete}(\pi_1, \ldots, \pi_K) \\ \text{features:} \quad & \boldsymbol{X} \mid Y = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \quad), \quad k = 1, \ldots, K \end{aligned}$$

for some unknown $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and Equivalently:

$$g_k(\boldsymbol{x}) = |2\pi \quad|^{-1/2} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \quad (\boldsymbol{x} - \boldsymbol{\mu}_k) \right).$$

- QDA is the plug-in rule based on this model.
- We use training data $\mathcal{D}_{\text{train}}$ to estimate

Quadratic discriminant analysis: Estimates

© 2026 Fessler

- QDA is the classifier obtained by plugging the following parameter estimates into the Bayes classifier formula:

$$\hat{\pi}_k = \frac{n_k}{n}, \qquad n_k = |\{i : y_i = k\}|$$

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i \,:\, y_i = k} \boldsymbol{x}_i$$

$$\widehat{\boldsymbol{\Sigma}}_k =$$

- These estimates are all **maximum likelihood estimates**.
- $\hat{\boldsymbol{\mu}}_k$ is the **sample mean** for class k.
- $\widehat{\boldsymbol{\Sigma}}_k$ is the
- All model parameters for $g_k(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x} \mid Y = k)$ are estimated from *only* the samples with label k, as is typical in many Bayesian methods [6, p. 91].

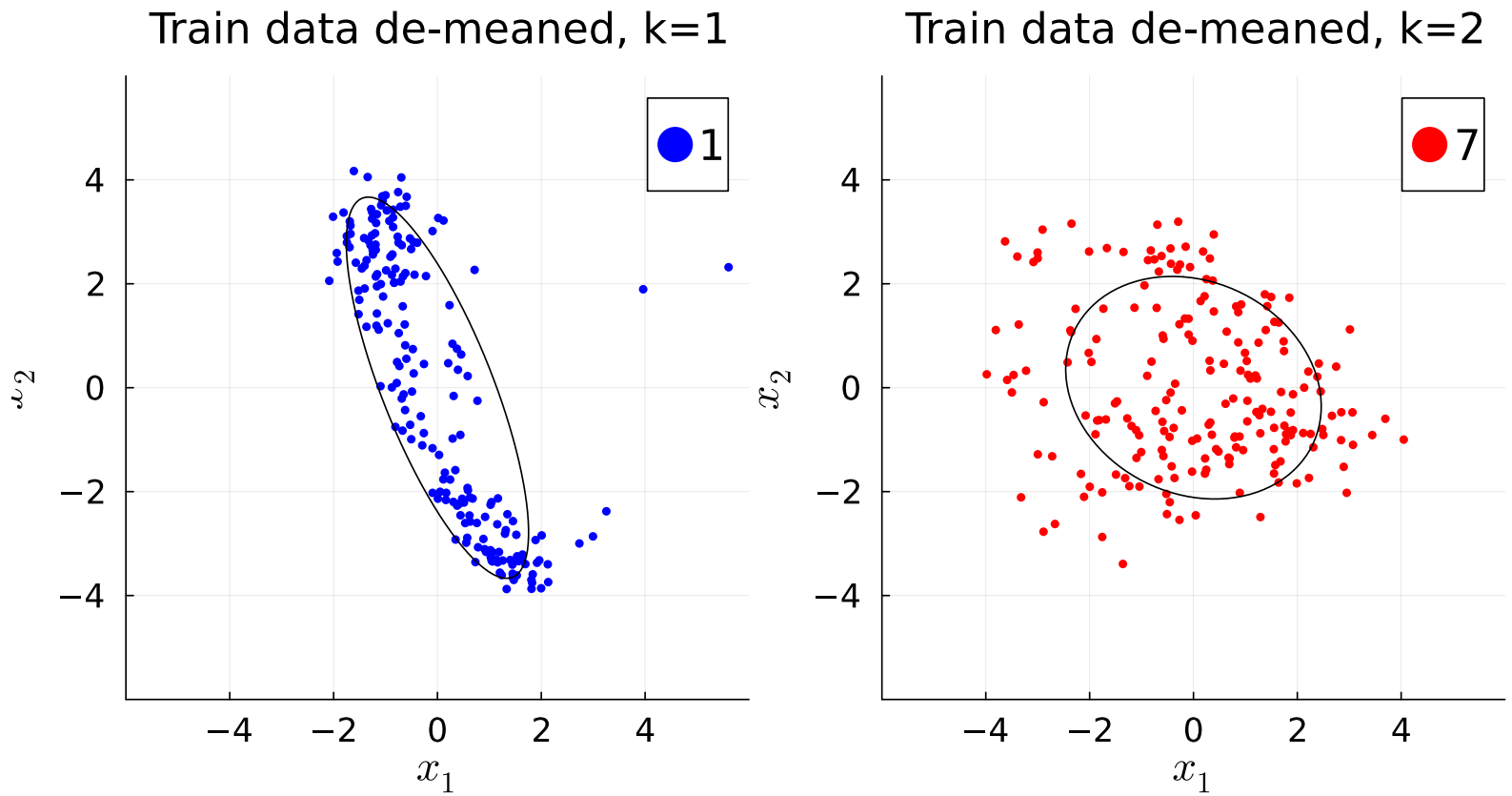
QDA estimates illustrated

© 2026 Fessler

Within-class sample covariance matrix:

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k} \sum_{i \,:\, y_i = k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^\top.$$

Julia QDA demo



Quadratic discriminant analysis: Binary case

© 2026 Fessler

- Binary setting, $Y \in \{-1, 1\}$.
- The QDA classifier is

$$f(\boldsymbol{x}) = \underbrace{\arg\max_k \hat{\pi}_k \hat{g}_k(\boldsymbol{x})}_{} = \underbrace{\operatorname{sign}\left(\frac{1}{2} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c \right)}_{K=2},$$

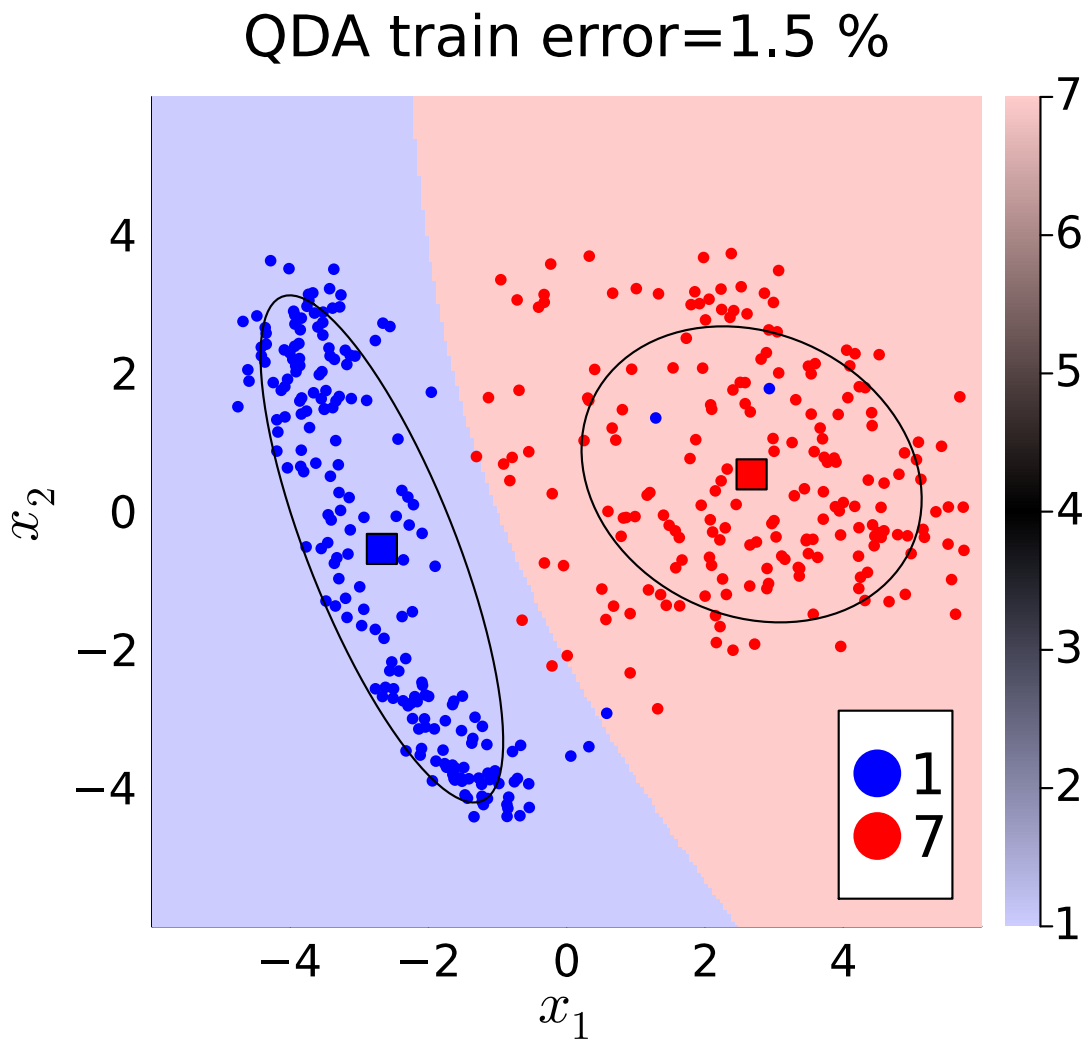
  for certain matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, scalar c.
- The decision boundaries are quadratics.

- Exercise: Determine $\boldsymbol{A}, \boldsymbol{b}, c$.
  You will again use sign of the difference instead of argmax, and then you manipulate the relevant terms of the Gaussian likelihoods.

QDA hand-written digit application

© 2026 Fessler

Julia QDA demo



Fisher's original iris data (application)

© 2026 Fessler





Image attributions:
setosa: CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=170298,
versicolor: CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=248095, and
virginica: By Frank Mayfield - originally posted to Flickr as Iris virginica shrevei BLUE FLAG, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=9805580

Fisher's original iris data

© 2026 Fessler

- 150 observations made on $K = 3$ classes of the iris flower: Iris setosa, Iris versicolor, and Iris virginica.
- $d = 4$ features:
  - **sepal** length
  - sepal width
  - petal length
  - petal width.

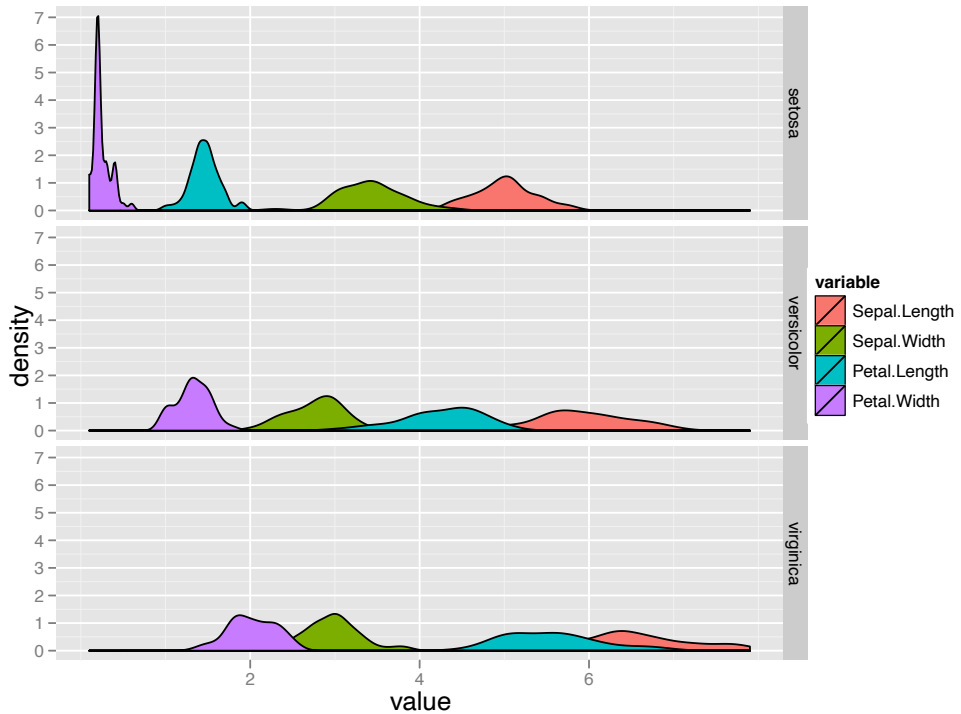


Figure courtesy Larry Wasserman

https://www.stat.cmu.edu/~larry/=sml/linearclassification.pdf

Q03.6 We could classify accurately based just on petal length.

A: True B: False **??**

CQ: LDA accuracy for $d = 2$

© 2026 Fessler

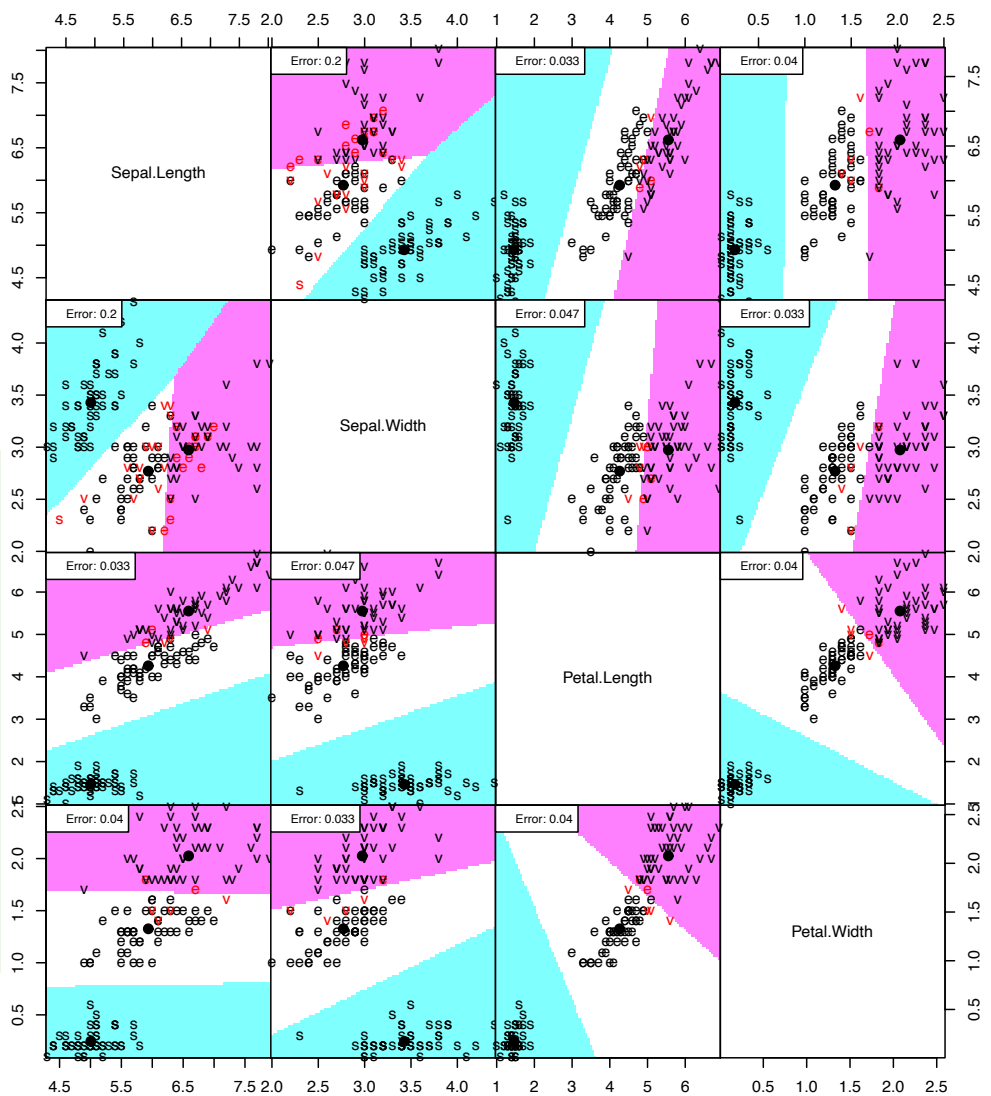
Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the model. In these plots, "s" represents the class label Iris setosa, "e" represents the class label Iris versicolor, and "v" represents the class label Iris virginica. The red letters illustrate the misclassified observations.

Q03.7 Which class of flower is most accurately classified by LDA per these plots?

A: Setosa ("s")

B: Versicolor ("e")

C: Virginica ("v")



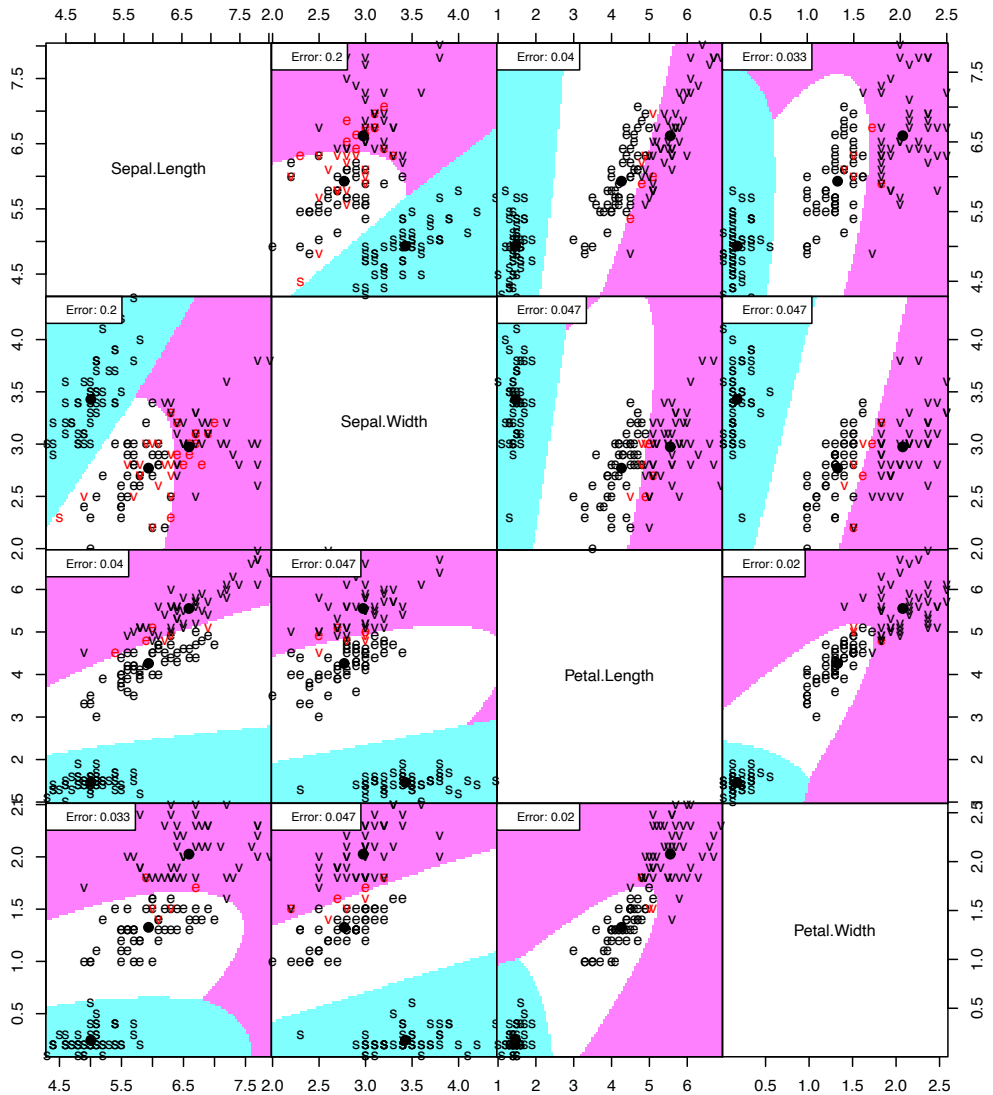
QDA on Fisher's original iris data

© 2026 Fessler

Classifying the Iris data using QDA. The multiple figure array illustrates the classification of observations based on QDA for every combination of two features. The classification boundaries are displayed and the classification error by simply casting the data onto these two features are calculated. In these plots, "s" represents the class label Iris setosa, "e" represents the class label Iris versicolor, and "v" represents the class label Iris virginica. The red letters illustrate the misclassified observations. Overall training error is 0.02.

Figure & text courtesy Larry Wasserman

https://www.stat.cmu.edu/~larry/=sml/linearclassification.pdf



Linear discriminant analysis: Summary

© 2026 Fessler

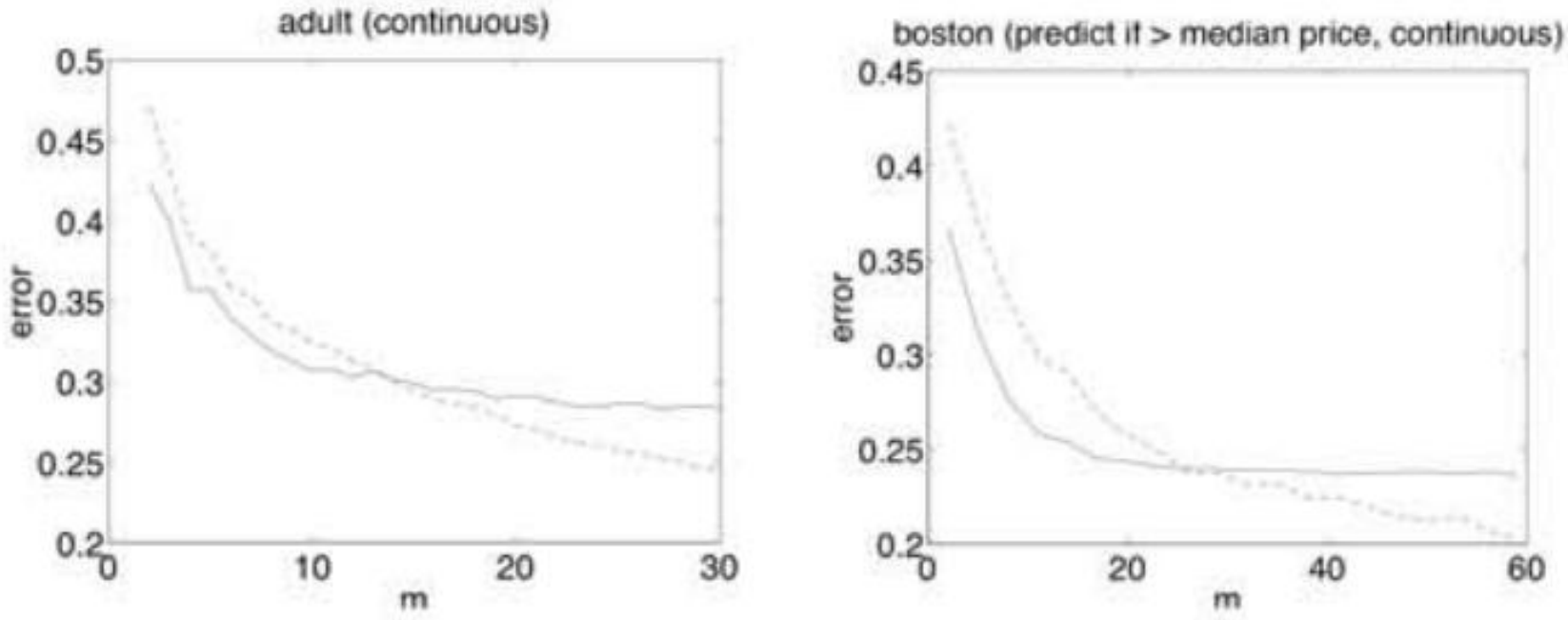
- One of the very earliest ML algorithms invented (c. 1930s)
- Binary classifier is "linear:"

$$f(\boldsymbol{x}) = \operatorname{sign}\Big(\tilde{f}(\boldsymbol{x})\Big), \quad \tilde{f}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

- $\boldsymbol{w}$ and b are functions of parameter estimates $\{\hat{\pi}_k\}$, $\{\hat{\boldsymbol{\mu}}_k\}$, $\widehat{\boldsymbol{\Sigma}}^{-1}$
- Requires $n \geq d$ although extensions are possible. In particular, MAP estimates of $\boldsymbol{\Sigma}$ based on priors can help [1, §9.2.4.3].
- Many applications throughout history, although better methods now available.
- **Logistic regression** often preferable, depending on n [7]
  - makes weaker assumptions
  - learns $f(\boldsymbol{x}) = \operatorname{sign}\big(\boldsymbol{w}^\top \boldsymbol{x} + b\big)$ directly from training data
- Optional: read about **Mahalanobis distance** in Prof. Scott's notes on Canvas.

Logistic regression vs Naïve Bayes

© 2026 Fessler



- Dashed line: logistic (discriminative)
  - lower asymptotic generalization error
- Dotted (solid?) line: Naïve Bayes (generative)
  - faster descent to higher error

Figure from [7, Fig. 1].

- Training data:

$$(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n) \overset{iid}{\sim} \mathrm{p}(\boldsymbol{x}, y).$$

- Notation for each training vector's elements:

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_d \end{bmatrix}$$

- *Naïve Bayes assumption*: Given Y, the features $X_1, \ldots, X_d$ are
- **Naïve Bayes** is a **plug-in**, **generative** method [7].
- It could be and **parametric** or **nonparametric**, depending on

CQ: Naïve Bayes model

© 2026 Fessler

Q03.8 Let $g_k(\boldsymbol{x})$ be the pmf/pdf of $\boldsymbol{X} \mid Y = k$, and $g_{kj}(x_j)$ the marginal pmf/pdf of $X_j \mid Y = k$. By the **Naïve Bayes** assumption:

A: $g_k(\boldsymbol{x}) = \sum_{j=1}^{d} \pi_j g_{kj}(\boldsymbol{x})$

B: $g_k(\boldsymbol{x}) = \sum_{j=1}^{d} g_{kj}(x_j)$

C: $g_k(\boldsymbol{x}) = \prod_{j=1}^{d} \pi_j g_{kj}(x_j)$

D: $g_k(\boldsymbol{x}) = \prod_{j=1}^{d} g_{kj}(x_j)$

E: none of these

- By the Naïve Bayes assumption:

$$g_k(\boldsymbol{x}) =$$

- Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be training data, and let

$$\hat{\pi}_k =$$

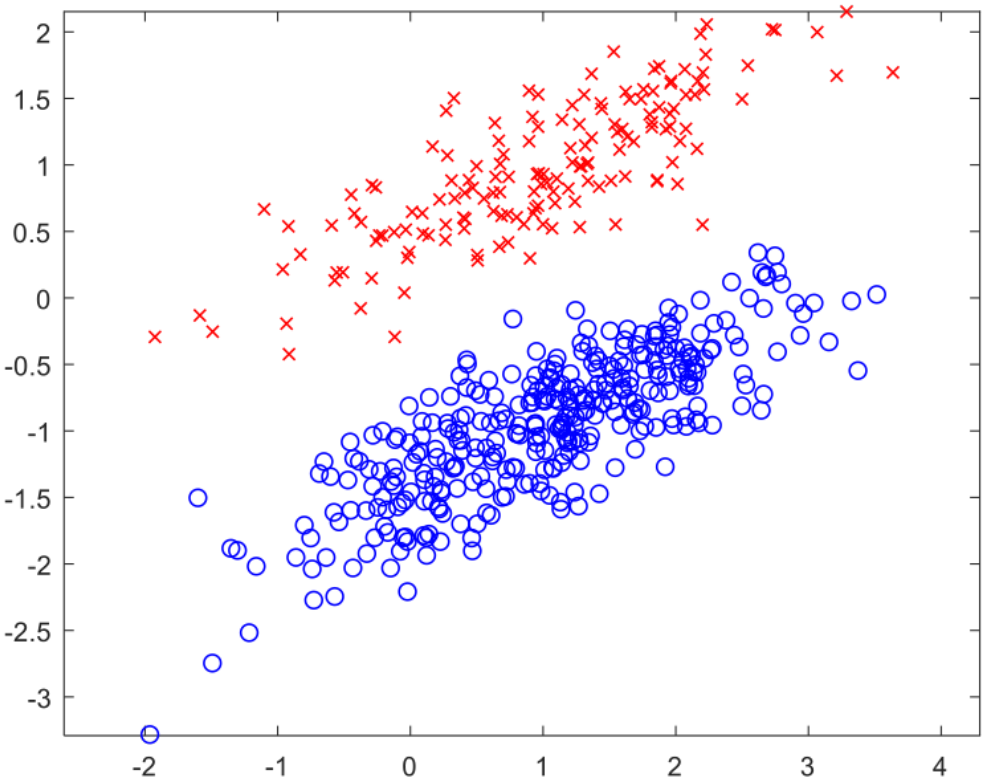
$$\hat{g}_{kj} =$$

- Then the Naïve Bayes classifier is

$$\widehat{f}(\boldsymbol{x}) =$$

Naïve Bayes: Gaussian data model

© 2026 Fessler



Naïve Bayes Example: Doc classification

© 2026 Fessler

- We wish to classify documents into categories like "business," "politics," "sports," etc.
- A simple yet popular feature representation is the

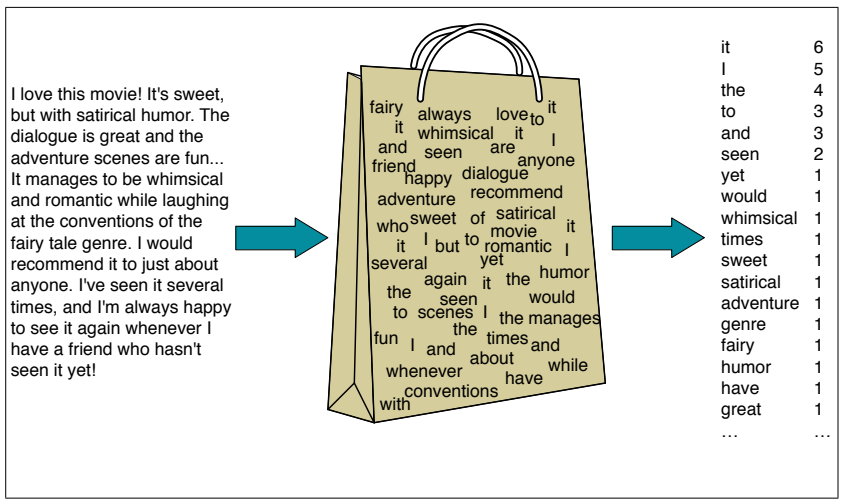


Figure 4.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Image courtesy [8].

Naïve Bayes Example: Doc classification

© 2026 Fessler

- A document is represented as a vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix},$$

where

$d =$

$x_j =$

- $\hat{\pi}_k =$
- $\hat{g}_{kj}(\ell) =$

Naïve Bayes Example: Doc classification

© 2026 Fessler

- `sklearn` dataset "20 newsgroups" studied in [9]
- Emails between participants of a newsgroup focusing on specific topics.
- $K = 20$ topics (these are the classes)
- $n = 11314$ training samples, 7532 test samples
- $d \approx 100,000$ word vocab, depending on vectorizer
- Example uses `TfidfVectorizer`

```
# Build the model
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
# Train the model using the training data
model.fit(train_data.data, train_data.target)
# Predict the categories of the test data
predicted_categories = model.predict(test_data.data)
```

Naïve Bayes Example: Doc classification

© 2026 Fessler

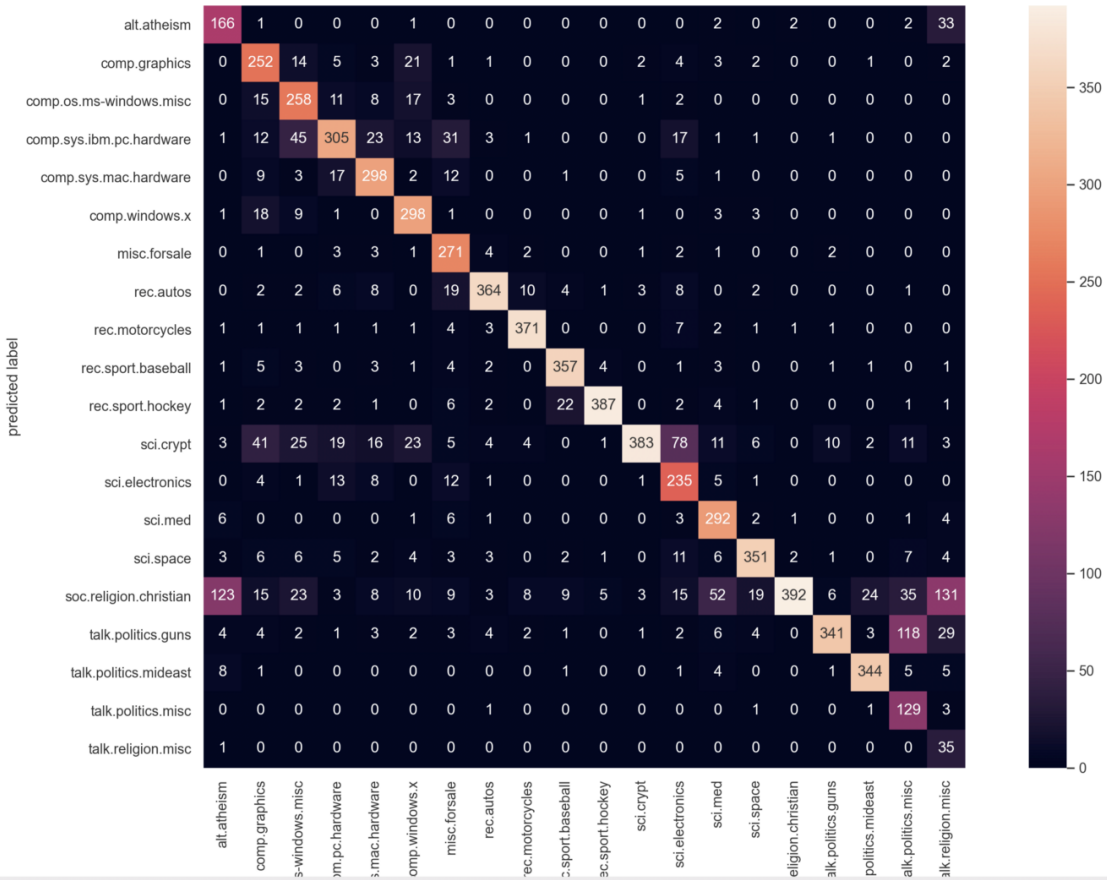


Figure courtesy [9]

Big picture: Toward ERM

© 2026 Fessler

- *(Regularized) empirical risk minimization* learns a prediction function f by solving

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i)) + \lambda \Omega(f),$$

- Different choices of $L, \mathcal{F}, \Omega$ lead to different methods.
- We will see several examples including support vector machines, boosting, decision trees, neural networks, and sparse linear regression
- Advantages of such a framework:
  - Makes it easier to compare and contrast different methods.
  - There are optimization strategies that can be used to solve large classes of ERM methods. Some will be covered in future lectures.
  - This framework facilitates theoretical analysis and understanding, which in turn facilitates progress.

References I

The Annals of Eugenics, established 1925, is now named The Annals of Human Genetics. While **Fisher** was an undisputed statistical giant, he also was part of an awful history that we should not ignore [10, 11].

[1] K. P. Murphy.
Probabilistic machine learning: An introduction.
MIT Press, 2022.

[2] Ronald A Fisher.
The use of multiple measurements in taxonomic problems.
Annals of eugenics, 7(2):179–188, 1936.

[3] Bernard L Welch.
Note on discriminant functions.
Biometrika, 31(1/2):218–220, 1939.

[4] Somesh Das Gupta.
Theories and methods in classification: a review.
Discriminant analysis and applications, pages 77–137, 1973.

[5] Andrew Ng and Michael Jordan.
On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes.
In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[6] R. O. Duda, P. E. Hart, and D. G. Stork.
Pattern classification.
Wiley, New York, 2001.

References II

[7] A. Ng and M. Jordan.
On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes.
In *NeurIPS*, volume 14, 2001.

[8] Dan Jurafsky and James H Martin.
Speech and language processing. Vol. 3.
TBD, 2022.
https://web.stanford.edu/~jurafsky/slp3/.

[9] Serafeim Loukas.
Text classification using Naive Bayes: Theory & a working example.
towardsdatascience. com, 2020.
https://towardsdatascience.com/
text-classification-using-naive-bayes-theory-a-working-example-2ef4b7eb7d5a.

[10] Walter Bodmer, RA Bailey, Brian Charlesworth, Adam Eyre-Walker, Vernon Farewell, Andrew Mead, and Stephen Senn.
The outstanding scientist, RA Fisher: his views on eugenics and race.
Heredity, 126(4):565–576, 2021.

[11] Eric Michael Johnson.
Ronald Fisher is not being 'cancelled', but his eugenic advocacy should have consequences, 2021.
https://thisviewoflife.com/
ronald-fisher-is-not-being-cancelled-but-his-eugenic-advocacy-should-have-consequences/.