EECS 553 Lecture 04
Machine Learning (ECE)
[class version]

Jeff Fessler

University of Michigan
©Laura Balzano and Jeff Fessler, UM, September 9, 2026, 14:56

F26

Recap / Outline

© 2026 Fessler

- Recap
  - Plug-in classifiers, LDA
- Today
  - Project overview
    - Project deadlines
  - Plug-in classifiers (continued)
    - Quadratic discriminant analysis (QDA)
    - Naïve Bayes classifier
  - Regression
    - Minimum MSE estimator
    - Generative approach
    - Plug-in regression
    - Discriminative approach (OLS)

Project Intro: ML Reproducibility Challenge

© 2026 Fessler

The goal of the (roughly annual) Machine Learning reproducibility challenge *"is to encourage the publishing and sharing of scientific results that are reliable and reproducible. In support of this, the objective of this challenge is to investigate reproducibility of papers accepted for publication at top conferences by inviting members of the community at large to select a paper, and verify the empirical results and claims in the paper by reproducing the computational experiments, either via a new implementation or using code/data or other information provided by the authors."*

Quote from paperswithcode.com.

- Syllabus link
- Groups of 3-4 students (self selected) entered into Canvas
  Use Piazza to help find group members
- Select any paper to reproduce from NeurIPS, ICML, ICLR, AAAI and AI Stats, as well as the journals JMLR and TMLR from
- You should highlight the ML aspects in your initial justification when selecting these papers and note that your selection is subject to approval by the instructional staff.

Project guidelines I

- As highlighted in the challenge description, "reproducing" a paper for the challenge involves investigating their *claims and conclusions* (not just the plots) and evaluating whether those claims and conclusions are reproducible.
- That is likely to include reproducing their experimental results on both synthetic and real data.
- Claims often are made broadly, yet in the paper they may be only illustrated on a few datasets, so reproducing their *claims* could involve applying their algorithm with different hyperparameters, to new datasets, and/or to synthetic data with realistic variations from the original paper.
- If computational efficiency is a focus of the original paper, students may want to time the algorithms using their own computing resources to see if they align with the author's numbers. If the algorithm has several components, (additional) ablation studies may be carried out.

Project guidelines II

- Claims may also involve mathematical statements that should be carefully checked, but simply saying “We checked the proof and it is correct" is insufficient – one must look at parts of the proofs or derivations where steps are missing, flesh out those steps, include additional references for justification, etc. Interaction with the original authors is encouraged.
- Groups may use code that authors share online or by email. Students may also choose to implement the algorithms from scratch, especially if the paper claims that the algorithms are simple to implement.

GenAI policy I

- You do not want to listen to AI slop
- I do not want to read AI slop.
- NY Times article: Writing with AI degrades your brain
- Guiding principle: *You can (and should) use GenAI in ways that enhance your learning, but you must not use GenAI in any way that hinders your learning.*
- Read the syllabus for more details
- *GenAI for the project assignments is permitted with disclosure. For example, you may ask it to help rewrite a paragraph, find bugs in your already written code, suggest an outline for a report, etc. If you use GenAI in submitted project work, you must include all the details in a "Project GenAI Disclosure Report" (see the project submission assignments for details). Save the transcript of your sessions and track changes as you*

GenAI policy II

edit, so you can create a complete disclosure report when the time comes. Using GenAI on the project is NOT required.

- Reports (both draft and final) must be submitted as a pdf to overleaf, but also shared with me and GSIs as a GoogleDoc or Overleaf where *version history* is visible. (No Word, no Pages...) A version history that shows signs of bulk cutting and pasting will be problematic.
- You are each responsible for every word in your reports, including all words written by other group members.
- I may use AI detectors like GPTZero if needed.
- Some ML conferences are strongly rejecting AI slop...

Project deadlines

Deadlines for project assignments (PA) are as follows:

- PA1 - Sep 25. Form your group and give 2-3 paper ideas, worth 5% of the total project grade.
  Provide links to official abstract, arXiv abstract, project/code.
- PA2 - Oct 9. Select your paper and give justification, worth 5% of the total project grade.
- PA3 - Nov 6. Draft reproducibility report due, worth 30% of the total project grade. Detailed expectations on Canvas.
- PA4 - Dec 11. Final report due, worth 40% of the total project grade.
- PA5 - TBD (Dec. 3 - 17) oral or poster presentation, worth 20% of the total project grade.
- Oral questions about your project work will count toward your project grade.
- There will also be points for attendance at project presentations and group assessment.

Feedback on papers for projects in W26

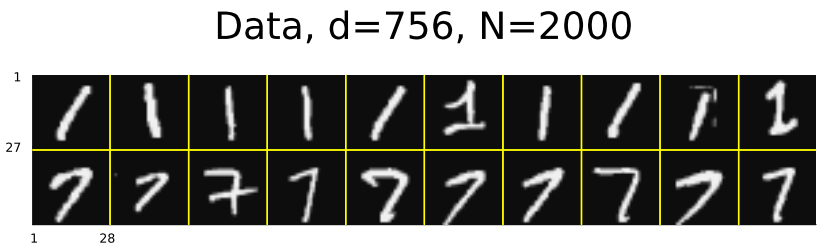
© 2026 Fessler

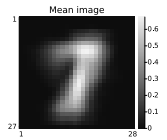
Team 1 - 1:00pm
Team 2 - 1:05
Team 3 - 1:10
Team 4 - 1:15
Team 5 - 1:20
Team 6 - 1:25
Team 7 - 1:30
Team 8 - 1:35
Break
Team 9 - 1:45
Team 10 - 1:50
Team 11 - 1:55
Team 12 - 2:00
Team 13 - 2:05
Team 14 - 2:10
...

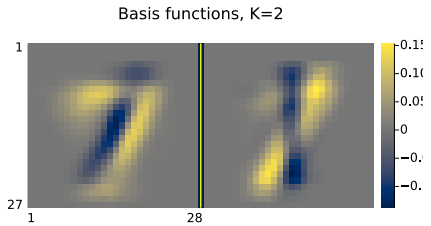
- Most paper ideas included experiments, which is good!
- Many papers were by authors at Facebook AI Research, DeepSeek, Google Brain, Google Research, DeepMind. It would be hard (or impossible) to truly replicate even parts of these papers, given the resource constraints.
- If a paper used 1 GPU of type P100, RTX 2080TI, V100, RTX 3090 or 4090, or A40, or equivalent, you can probably accomplish it on Great Lakes.
- If a paper used 8+ GPUs or one GPU of type A100, A800, H100, B100, you are would be unable to replicate those results.
- Replicating only part of the results is fine. More importantly, we want you to think about their claims and run new adjacent experiments to investigate whether their claims hold.
- Check whether the paper has been replicated already, whether directly through a Reproducibility Challenge report or through follow-on papers that repeat similar experiments. Papers cited 100s or 1000s of times certainly have had experiments replicated, and we will expect you to discuss at least a couple of those follow-on papers as well.

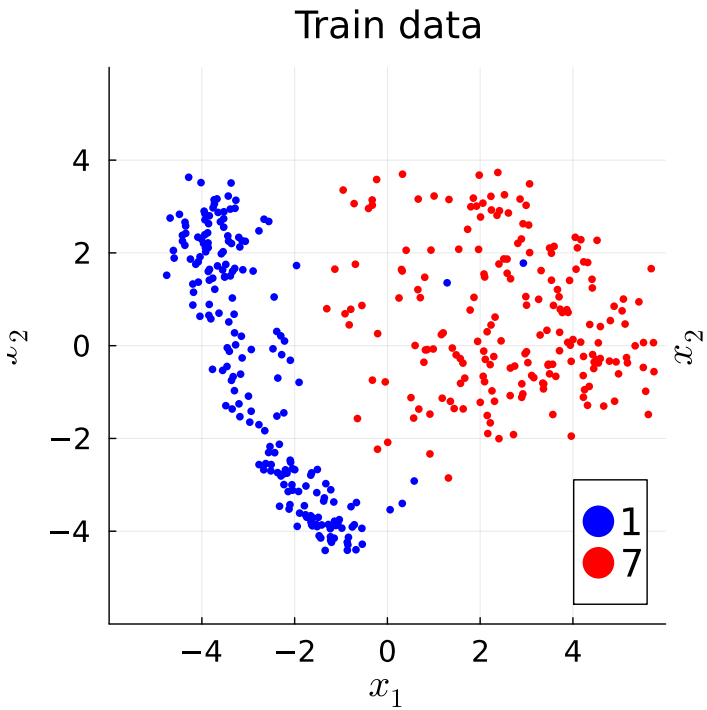
Classifier demo for digits 1,7: Data

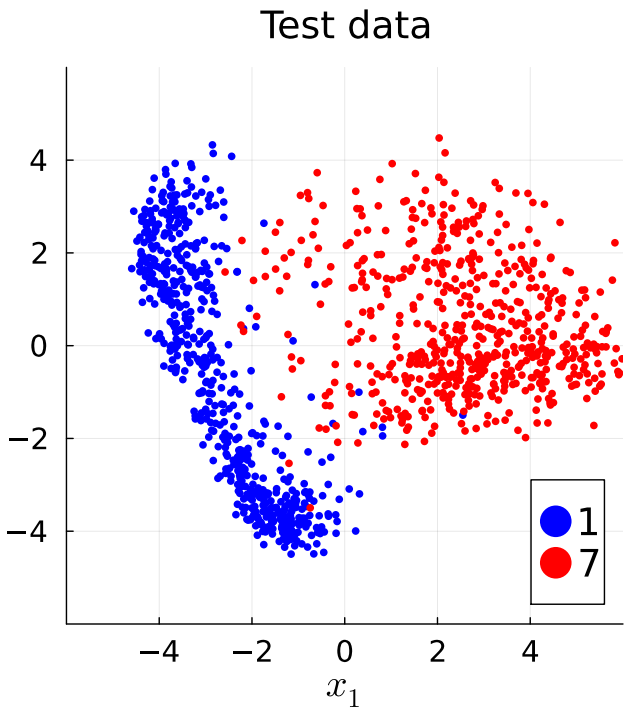
© 2026 Fessler



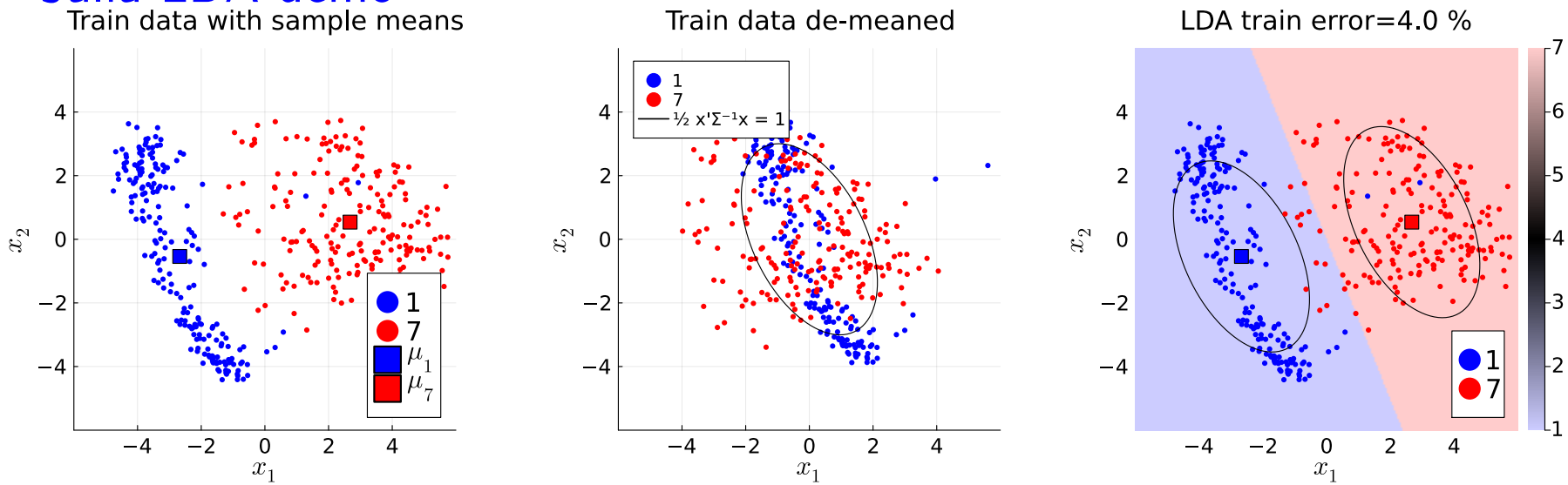








Julia LDA demo



The covariance matrices $\boldsymbol{\Sigma}_k$ differ *greatly* between the two classes.
Recall LDA plug-in classifier:

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

$$= \arg\max_k \; \boldsymbol{w}_k^\top \boldsymbol{x} + b_k, \qquad \boldsymbol{w}_k \triangleq \widehat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k, \quad b_k \triangleq \log \hat{\pi}_k - \tfrac{1}{2} \hat{\boldsymbol{\mu}}_k^\top \widehat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k$$

$$= \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b), \quad \mathcal{Y} = \{+1, -1\}$$

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 compute maximum-likelihood estimates of QDA model parameters
from training data $\mathcal{D}_{\text{train}}$ via $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \text{p}(\mathcal{D}_{\text{train}}; \boldsymbol{\theta})$.

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 [2, p. 91].

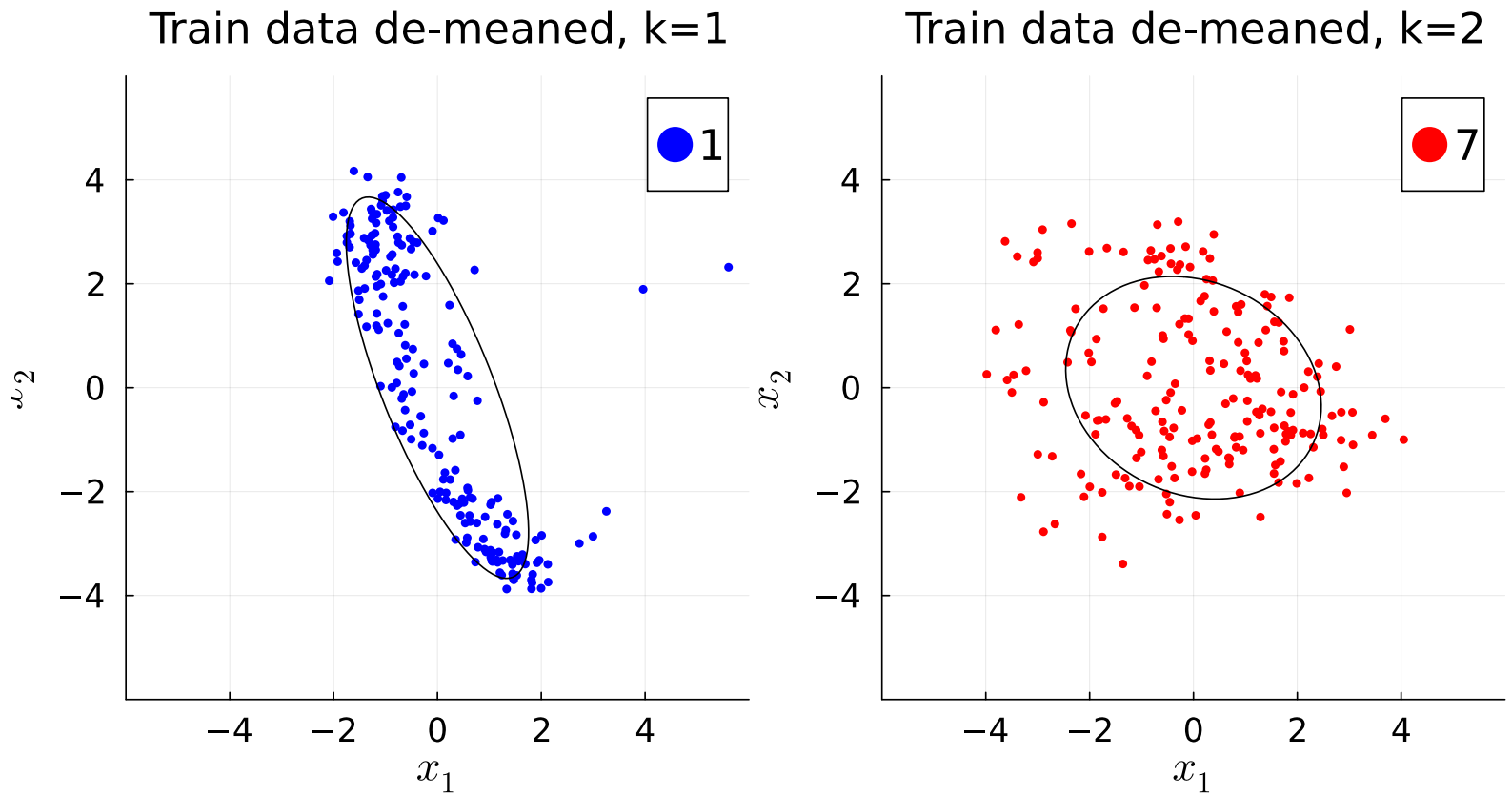
QDA estimates illustrated

© 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

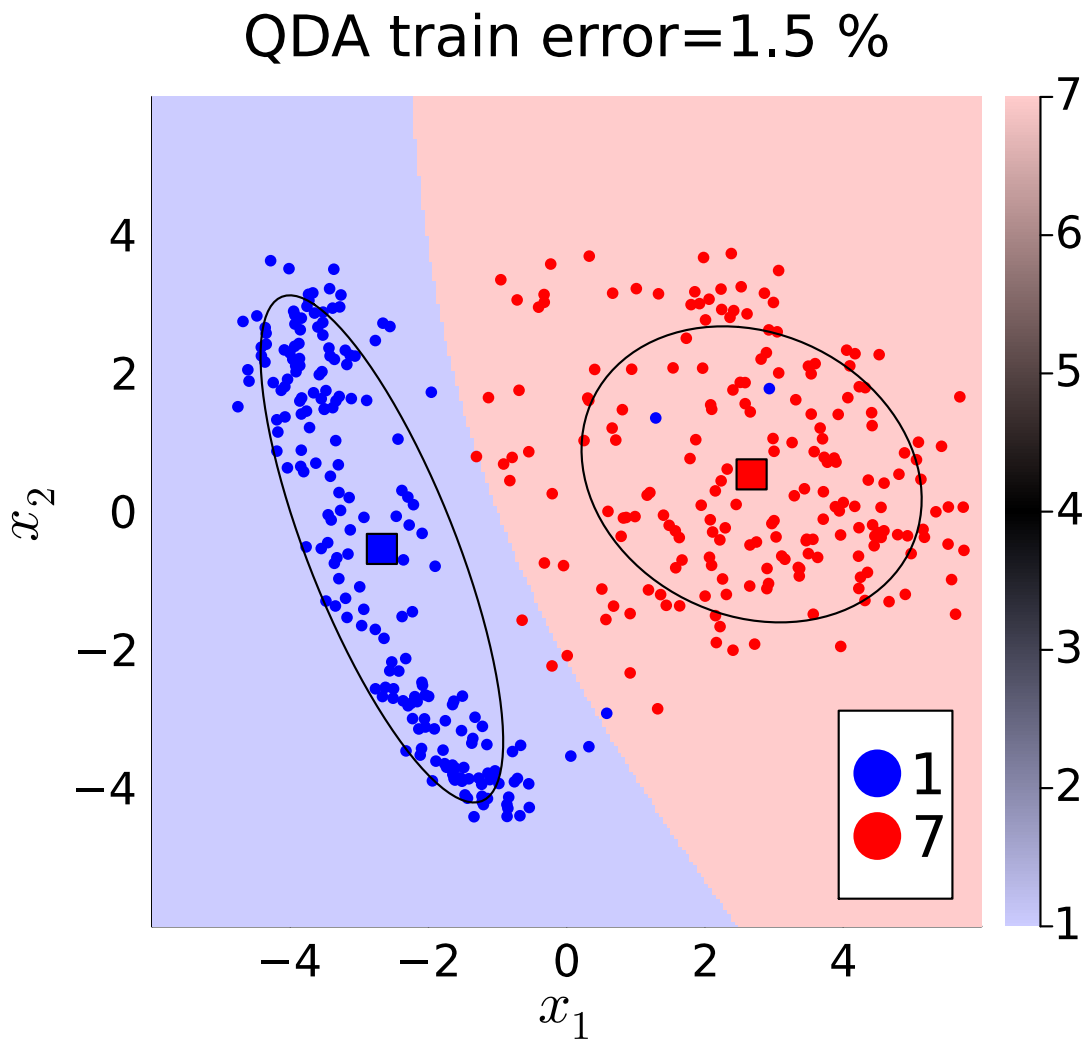
- Binary setting, $Y \in \{-1, 1\}$.
- The QDA classifier is

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

  for certain matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, scalar c.
- The decision boundaries are quadratics due to $\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}$ term.

- 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.

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.

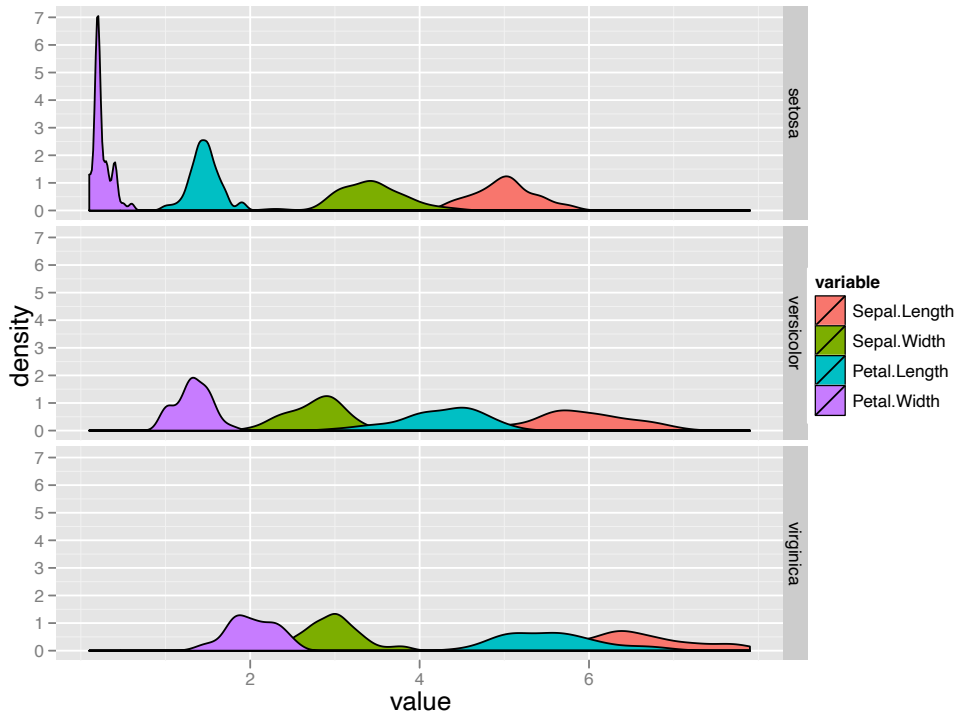


Figure courtesy Larry Wasserman

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

Q04.1 We could classify accurately based just on petal length.

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

CQ: LDA accuracy for $d = 2$ (of 4)

© 2026 Fessler

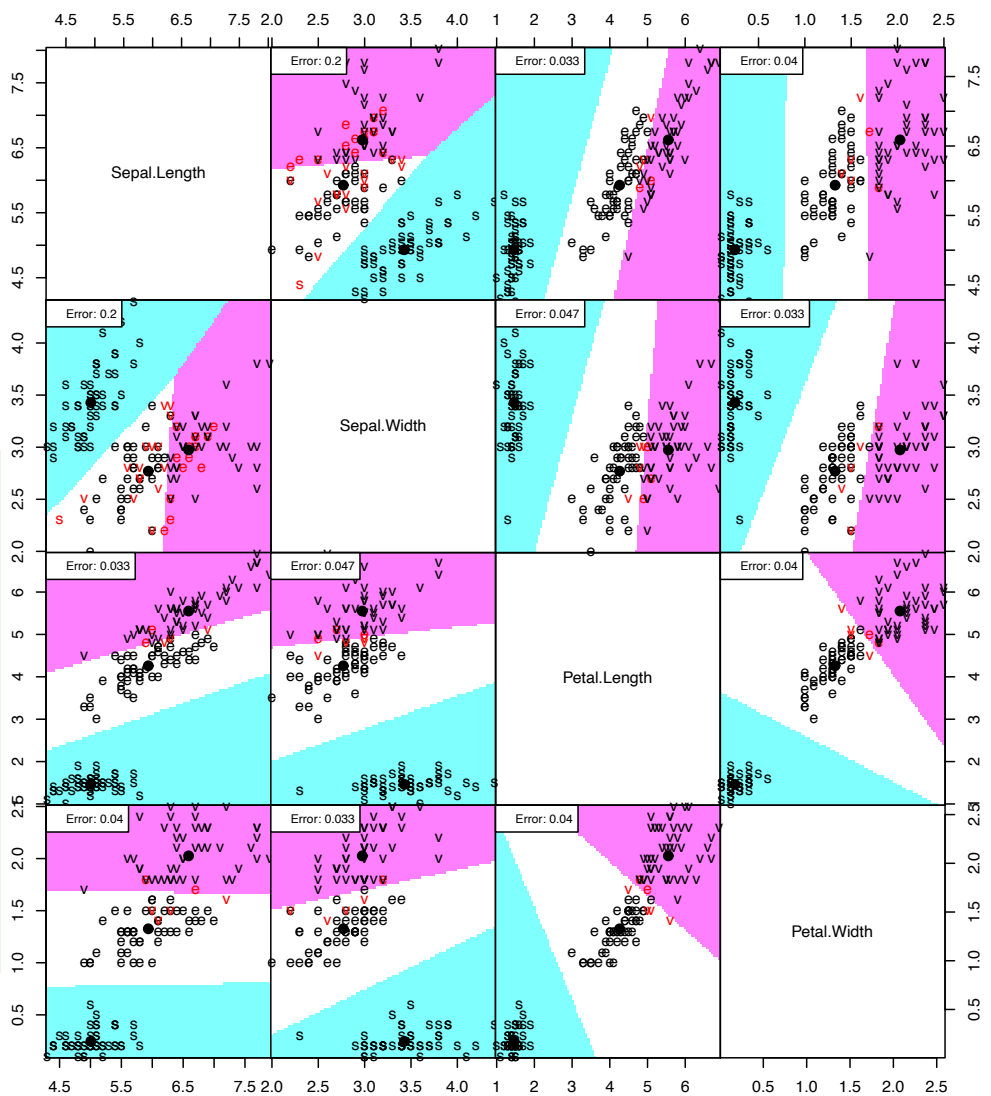
Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the model. In 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.

Q04.2 Which class of flower is most accurately classified by LDA per these plots?

A: Setosa ("s")

B: Versicolor ("e")

C: Virginica ("v")



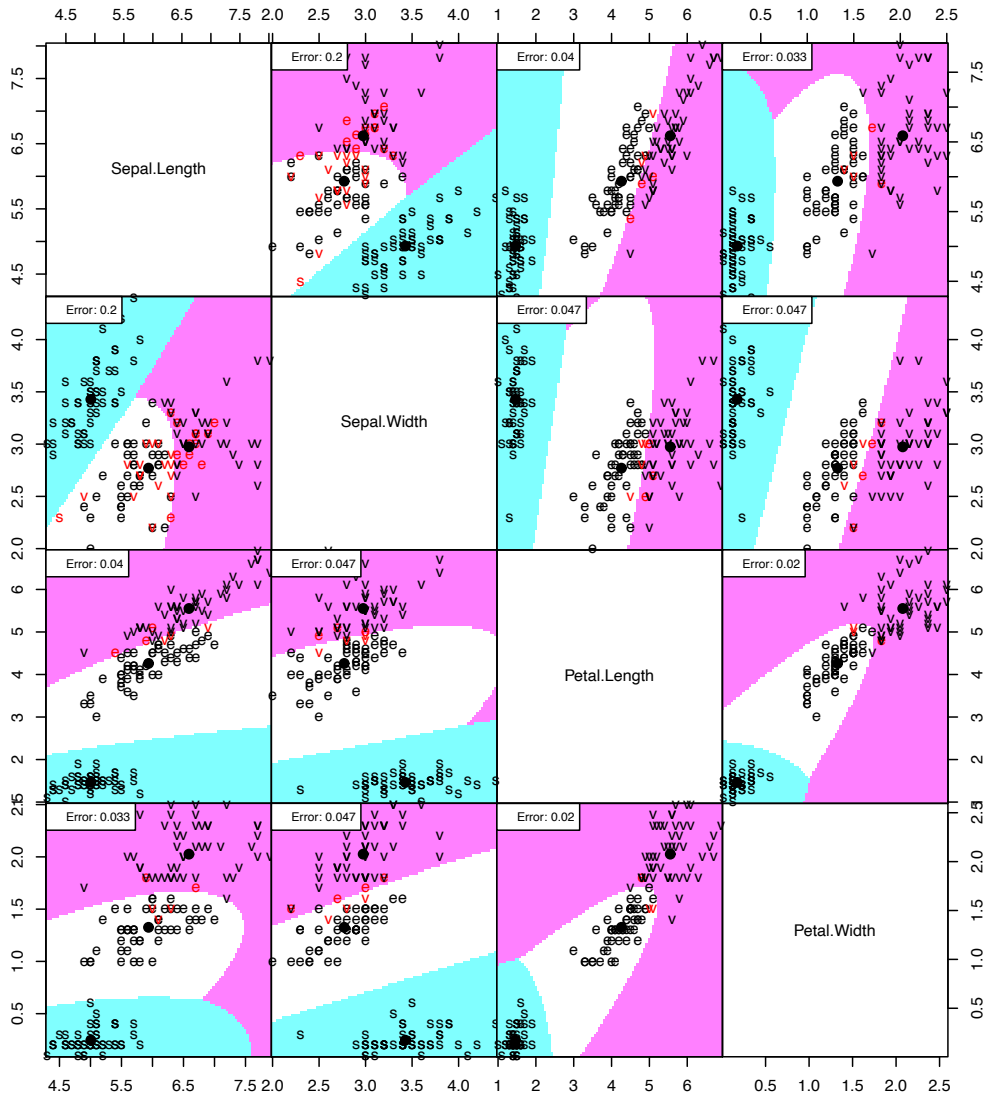
QDA on Fisher's original iris data

© 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

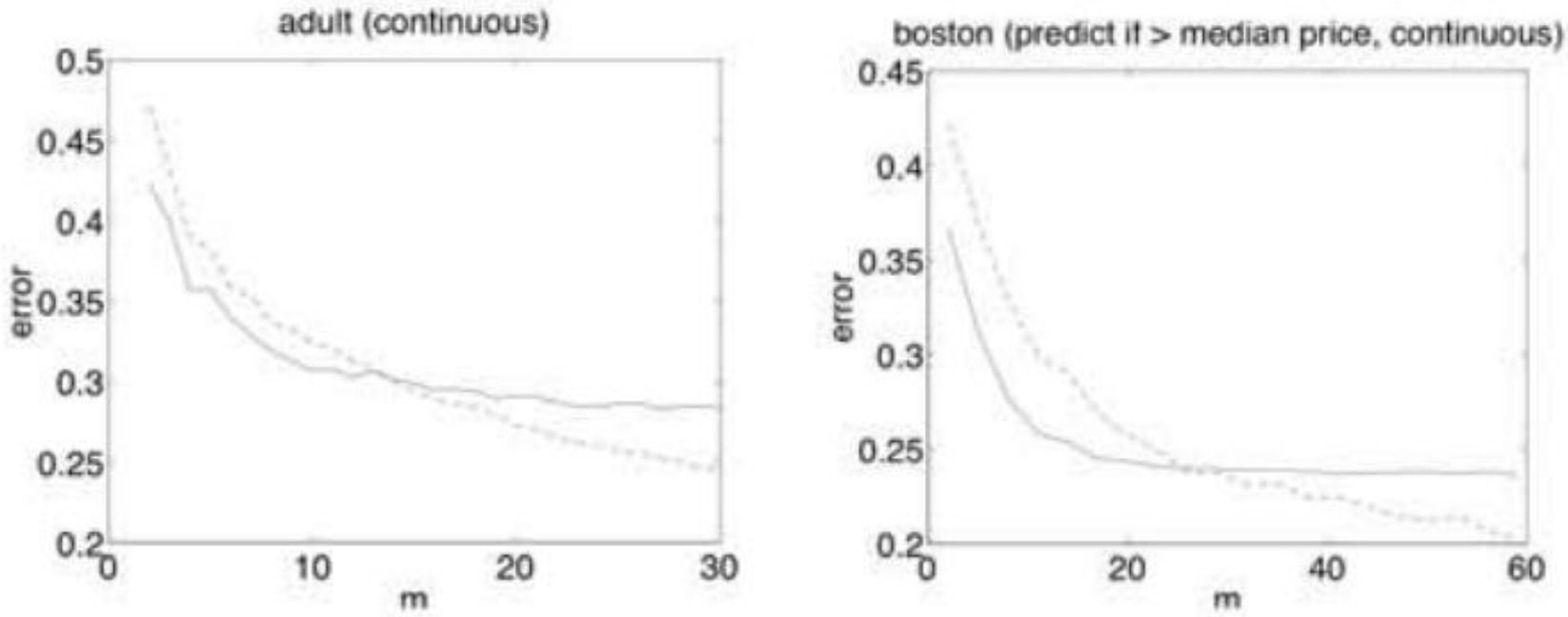
- One of the very earliest ML algorithms invented (c. 1930s)
- Binary classifier is "linear:"

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

- $\boldsymbol{w}$ 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 [3]
  - 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



- Horizontal axis "m" is our "n" (number of training pairs)
- Dashed line: logistic (discriminative)
  - lower asymptotic generalization error
- Dotted (solid?) line: Naïve Bayes (generative)
  - faster descent to higher error

Figure from [3, 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 [3].
- It could be and **parametric** or **nonparametric**, depending on

CQ: Naïve Bayes model

© 2026 Fessler

Q04.3 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:

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

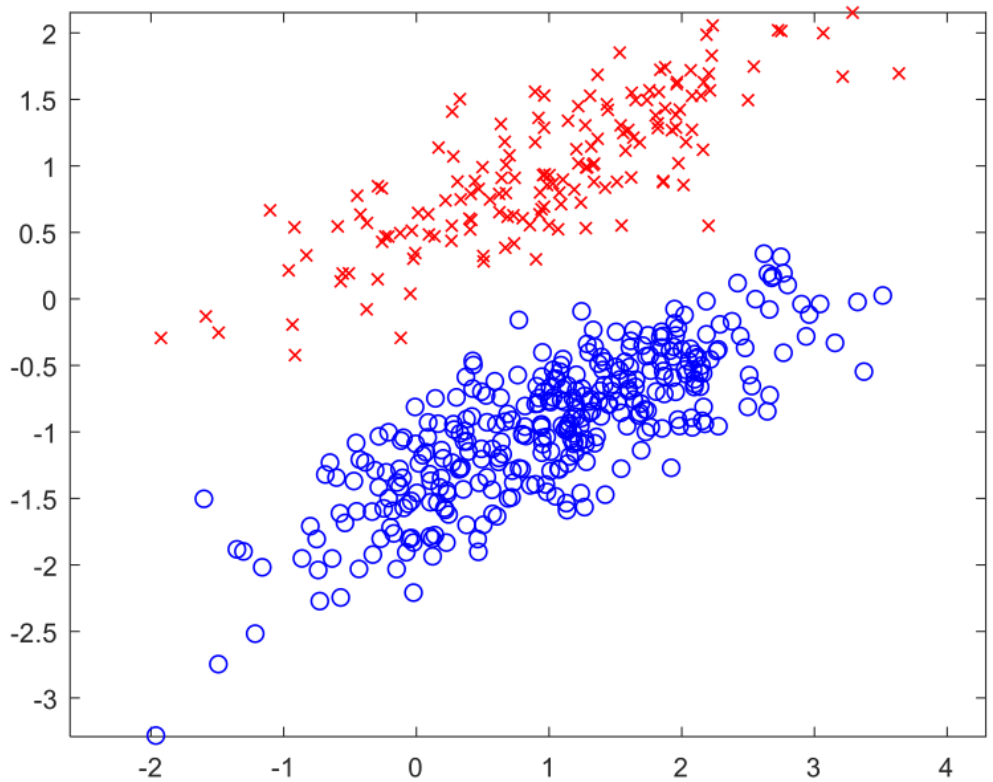
- Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be training data, and let

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

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

- Then the Naïve Bayes classifier is

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



HW3 will apply to handwritten digits.

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

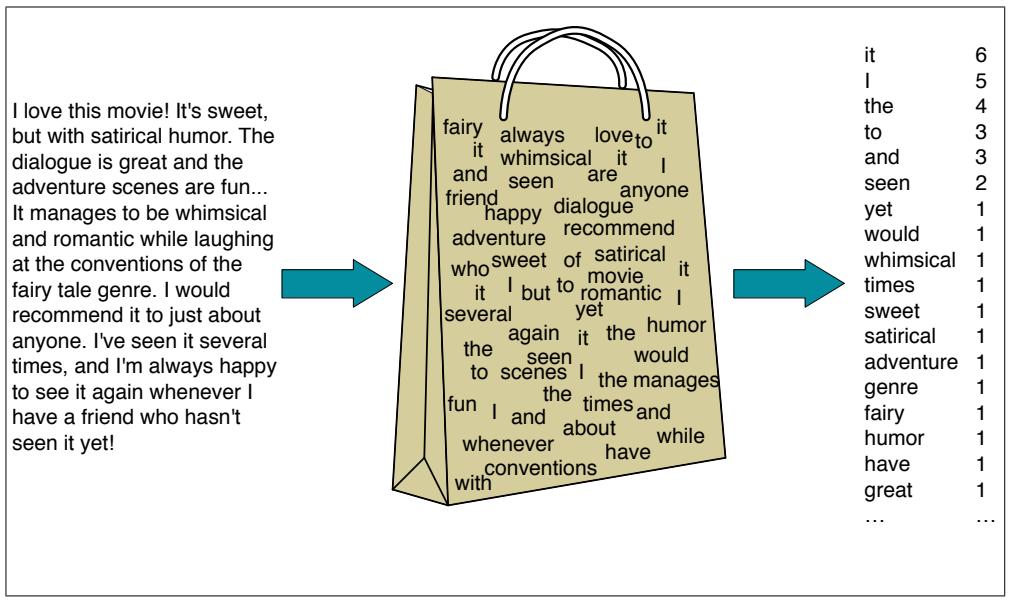


Figure 4.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Image courtesy [4].

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

$d =$

$x_j =$

- $\hat{\pi}_k =$
- $\hat{g}_k(\boldsymbol{x}) = \prod_{j=1}^{d} \hat{g}_{kj}(x_j)$, where $\hat{g}_{kj}(\ell) =$

Naïve Bayes Example: Doc classification

© 2026 Fessler

- `sklearn` dataset "20 newsgroups" studied in [5]
- 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)
```

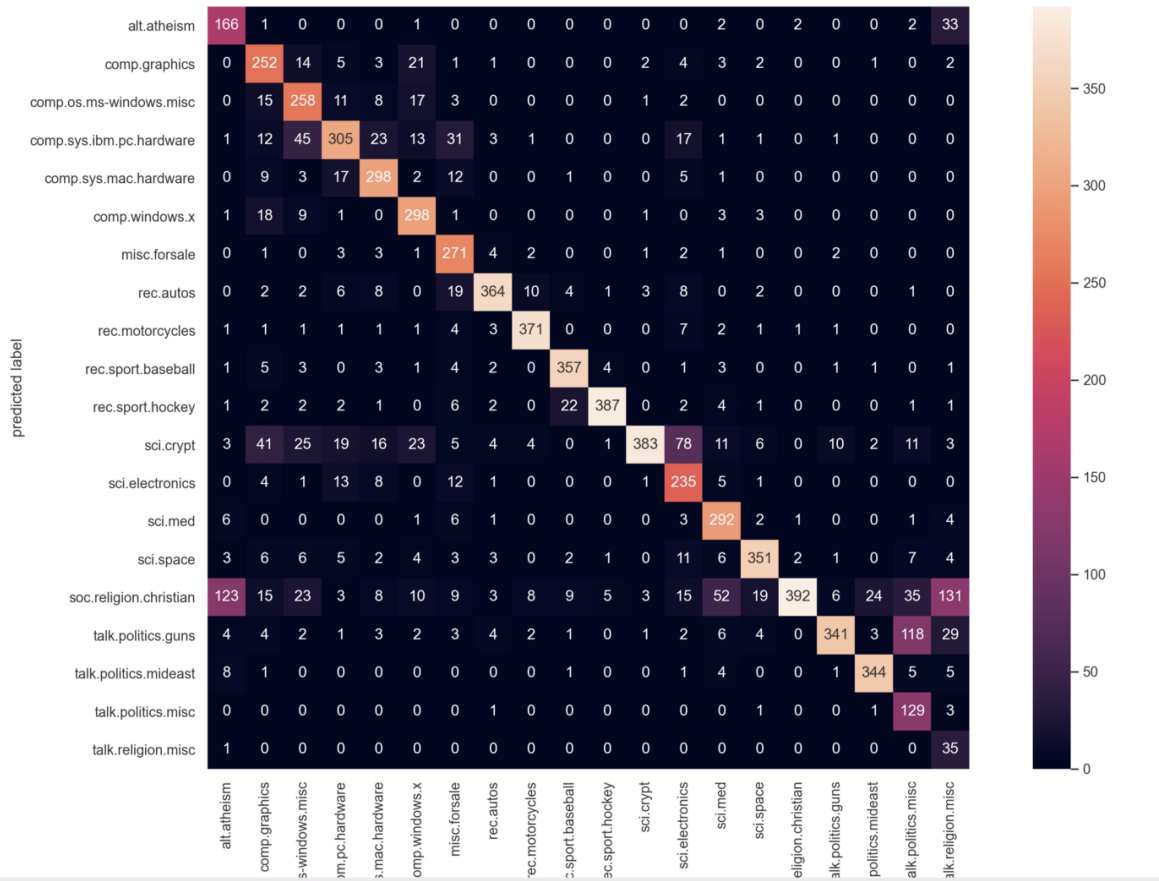
Confusion matrix

$K \times K$

here $K = 20$

count successes

Figure courtesy [5]



- Regression and classification are the two classical **supervised learning** problems.
- Every feature vector $\boldsymbol{x}$ is associated to a response variable y, and the goal is to predict y from $\boldsymbol{x}$.
- As in classification, this prediction function must be learned from training data.

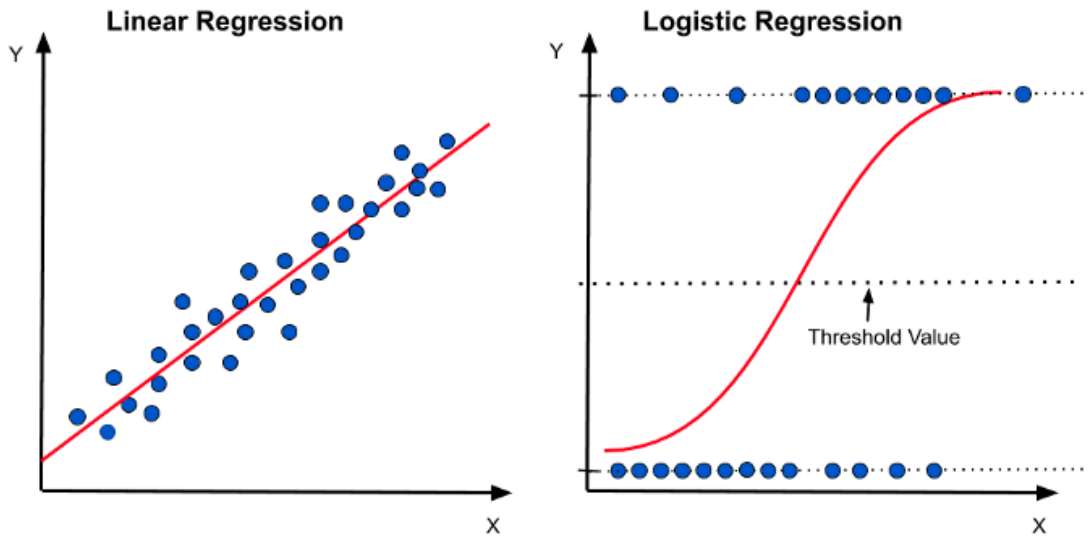


Figure courtesy
https://ai.plainenglish.io/why-is-logistic-regression-called-regression-if-it-is-a-classification-algorithm-9c2a166e7b74

Regression setting

© 2026 Fessler

- The main difference between **classification** and **regression** is

$$Y \in$$

- Probabilistic setting: We assume $(\boldsymbol{X}, Y)$ are jointly distributed random variables with some $\mathrm{p}(\boldsymbol{x}, y)$ [1, §11.2.3.5].
- The goal is to predict the response or "label" $Y \in \mathbb{R}$ from features $\boldsymbol{X} \in \mathbb{R}^d$ using a **regression function**

$$f : \mathbb{R}^d \to \mathbb{R}.$$

- We use different performance measures for regression because Y is "continuous." (The 0-1 loss would be)
- In regression problems, the most common **loss** function is **squared error**:

$$L(y, \hat{y}) = (\hat{y} - y)^2.$$

- The corresponding **risk** is the **mean squared error** (**MSE**):

$$R(f) \triangleq$$

Conditional mean predictor

© 2026 Fessler

- Just like classification, there is a regression function f^* that achieves the minimum value R^* of the MSE.
- **Theorem:** The function that minimizes the :

$$f^*(\boldsymbol{x}) \triangleq$$

  Proof in Prof. Scott's lecture notes on Canvas.
- This function is called the **conditional mean** predictor or minimum MSE (**MMSE**) estimator.
- Awesome theorem in theory.
- MMSE perspective is useful for modern diffusion generative models, *e.g.*, [6, 7].
- Challenges in practice:
  - The expectation/integral can be difficult except in special cases.
  - Often we do not know $\mathrm{p}(y \mid \boldsymbol{x})$.

- One approach is to assume $(\boldsymbol{X}, Y)$ have a joint multivariate Gaussian distribution:

$$\begin{bmatrix} \boldsymbol{X} \\ Y \end{bmatrix} \sim$$

where the **cross covariance** is $\boldsymbol{\Sigma}_{xy} = \mathbb{E}\{\boldsymbol{X}Y\} \in \mathbb{R}^d$.

- Under this (strong) Gaussian assumption, the MMSE estimator is analytically tractable [1, §11.2.3.5]:

$$f_*(\boldsymbol{x}) = \mathbb{E}\{Y \mid \boldsymbol{X} = \boldsymbol{x}\} = \mu_y + \boldsymbol{\Sigma}_{xy}^\top \boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$

$$= \quad$$

$$\boldsymbol{w} \triangleq \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}, \quad b \triangleq \mu_y - \boldsymbol{\Sigma}_{xy}^\top \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\mu}_x$$

- The estimator is !

- For a practical plug-in method, we estimate the means, the covariances, and the cross-covariances from training data:

$$\hat{\boldsymbol{\mu}}_x = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \qquad \hat{\mu}_y = \frac{1}{n}\sum_{i=1}^{n} y_i$$

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

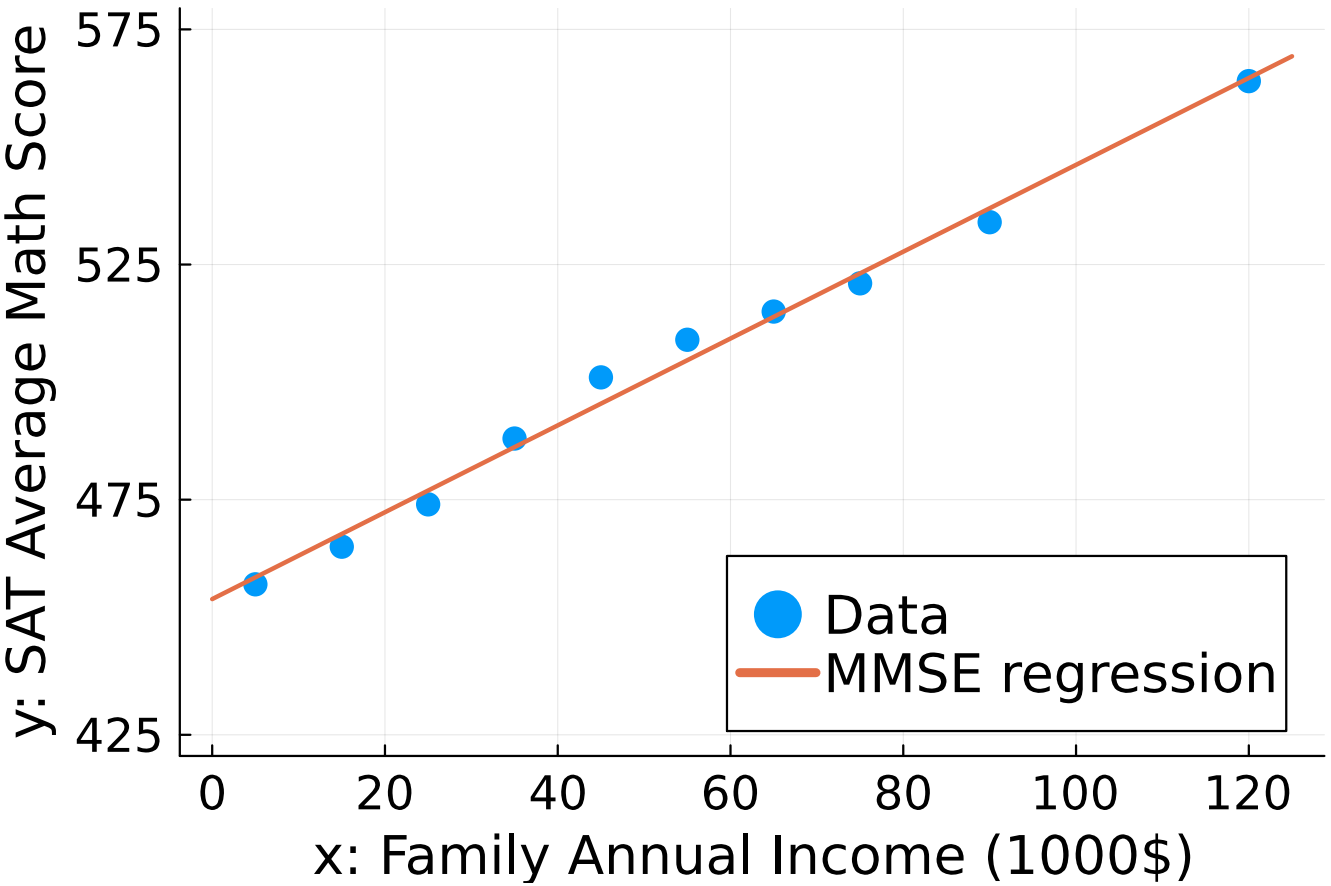
$$\widehat{\boldsymbol{\Sigma}}_{xy} = \frac{1}{n}\sum_{i=1}^{n}$$

- Then the plug-in estimator is $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$,
$\boldsymbol{w} = \hat{\boldsymbol{\Sigma}}_{xx}^{-1} \hat{\boldsymbol{\Sigma}}_{xy}, \quad b = \hat{\mu}_y - \hat{\boldsymbol{\Sigma}}_{xy}^\top \hat{\boldsymbol{\Sigma}}_{xx}^{-1} \hat{\boldsymbol{\mu}}_x$
- Again need $n \geq d$ (at least) for invertibility of $\widehat{\boldsymbol{\Sigma}}_{xx}$.
- Can work well *if* joint Gaussian distribution is reasonable and n is large enough.

MMSE regression example

© 2026 Fessler

MMSE regression demo using data from [8]



- In practice the joint distribution of $(\boldsymbol{X}, Y)$ is unknown, so we must estimate f^* using training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$.
- Choose f to minimize the

$$\underbrace{R(f) = \mathbb{E}_{\boldsymbol{X},Y}\left\{(f(\boldsymbol{X}) - Y)^2\right\}}_{\text{ideal}} \quad \overset{?}{\approx} \quad \underbrace{\widehat{R}(f) =}_{\text{practical}}$$

- To avoid overfitting, we must restrict f to belong to some **regression model**, *i.e.*,
- For now, we focus on the **linear model** (actually **affine**):

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

where $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$.

- This is a (typical) discriminative approach to regression.

Let

$$\boldsymbol{y} \triangleq \quad , \quad \bar{\boldsymbol{X}} \triangleq \quad , \quad \boldsymbol{\theta} \triangleq$$

We solve the following optimization problem known as **least-squares linear regression** or **ordinary least squares**:

$$\underset{b \in \mathbb{R},\, \boldsymbol{w} \in \mathbb{R}^d}{\arg\min} \; \frac{1}{n} \sum_{i=1}^{n} (\underbrace{\boldsymbol{w}^\top \boldsymbol{x}_i + b}_{} - y_i)^2 \equiv$$

Q04.4 Assume $d+1 \leq n$ and $\bar{\boldsymbol{X}} \in \mathbb{R}^{n \times (d+1)}$ has full **rank**. What is

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{d+1}}{\arg\min} \frac{1}{n} \left\| \bar{\boldsymbol{X}} \boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 \qquad ?$$

A: $\bar{\boldsymbol{X}}^\top (\bar{\boldsymbol{X}} \bar{\boldsymbol{X}}^\top)^{-1} \boldsymbol{y}$

B: $\bar{\boldsymbol{X}}^{-1} \boldsymbol{y}$

C: $(\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}})^{-1} \bar{\boldsymbol{X}}^\top \boldsymbol{y}$

D: $\bar{\boldsymbol{X}} (\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}})^{-1} \bar{\boldsymbol{X}}^\top \boldsymbol{y}$

E: minimizer need not be unique

For $\bar{\boldsymbol{X}} \in \mathbb{R}^{n \times (d+1)}$ and $\boldsymbol{y} \in \mathbb{R}^n$, **ordinary least squares** (**OLS**) seeks the minimizer

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^{d+1}}{\arg\min} \frac{1}{n} \left\| \bar{\boldsymbol{X}} \boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 .$$

Q04.5 What happens when $d \geq n$?

A: $\bar{\boldsymbol{X}} \bar{\boldsymbol{X}}^\top$ is never invertible, so we cannot compute the OLS solution

B: $\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}}$ is never invertible, so the OLS solution is not unique

C: No OLS solution exists

D: The unique solution is $\bar{\boldsymbol{X}}^+ \boldsymbol{y}$

E: None of these

Regularized least squares linear regression I

- For greater generality, especially to handle cases where $\bar{\boldsymbol{X}}$ is rank deficient or when $\bar{\boldsymbol{X}}'\bar{\boldsymbol{X}}$ is poorly conditined, we can add a

$$\underset{b\in\mathbb{R},\, \boldsymbol{w}\in\mathbb{R}^d}{\arg\min} \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 +$$

- $\lambda \geq 0$ is the **regularization parameter** or sometimes **hyperparameter**.
- If we choose this method is known as
- To avoid a systematic bias, often we choose [9, p. 10]

$$\Omega(\boldsymbol{w}, b) = \|\boldsymbol{w}\|_2^2 .$$

Regularized least squares linear regression II

- If we choose
  this method is known as [10] the
- The "selection" here is important: The resulting $\hat{\boldsymbol{\theta}}$ is typically sparse, so the method *selects* a subset of the features for prediction, improving interpetability.
- If we choose

$$\lambda\Omega(\boldsymbol{\theta}) =$$

  this method is known as [11] the

References I

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

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

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

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

[5] 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.

[6] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov.
Provable convergence of plug-and-play priors with MMSE denoisers.
IEEE Signal Proc. Letters, 27:1280–4, 2020.

[7] J. Liu, R. Anirudh, J. J. Thiagarajan, S. He, K. A. Mohan, U. S. Kamilov, and H. Kim.
DOLCE: A model-based probabilistic diffusion framework for limited-angle CT reconstruction.
In *Proc. Intl. Conf. Comp. Vision*, pages 10464–74, 2023.

References II

[8] L. M. Lesser.
Critical values and transforming data: teaching statistics with social justice.
J. of Statistics Education, 15(1), August 2017.

[9] C. Bishop.
Pattern recognition and machine learning.
Springer, 2006.

[10] R. Tibshirani.
Regression shrinkage and selection via the LASSO.
J. Royal Stat. Soc. Ser. B, 58(1):267–88, 1996.

[11] H. Zou and T. Hastie.
Regularization and variable selection via the elastic net.
J. Royal Stat. Soc. Ser. B, 67(2):301–20, 2005.