

# Data Analytics

## EEE 4774 & 6777

### Module 4 - Classification

Logistic Regression (LR), k Nearest Neighbor (kNN), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA)

Spring 2022

# Logistic Regression for Classification

$$p(C = 1|\mathbf{x}) = y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$

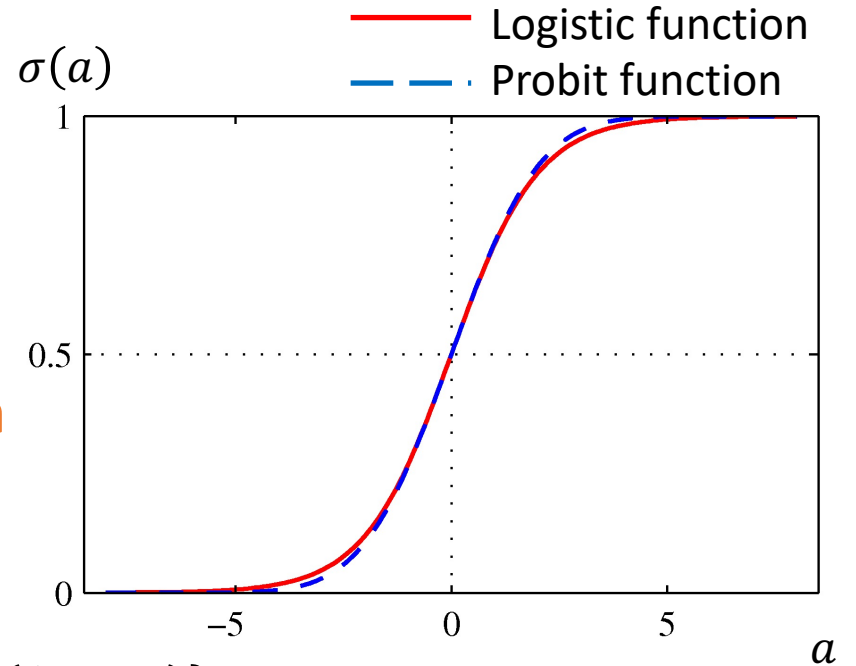
$$\hat{C}_n = \begin{cases} 1 & \text{if } \mathbf{w}^T \mathbf{x} \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

Logistic sigmoid function

$$E(\mathbf{w}) = -\log p(\mathbf{C}|\mathbf{w}) = -\sum_{n=1}^N \{C_n \log y_n + (1 - C_n) \log (1 - y_n)\}$$

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - C_n) \mathbf{x}_n$$


$$\mathbf{w}_{i+1} = \mathbf{w}_i - \eta \nabla E(\mathbf{w})$$



# Multiclass Logistic Regression

$$p(C_k|\mathbf{x}) = y_k(\mathbf{x}) = \frac{e^{w_k^T \mathbf{x}}}{\sum_{j=1}^K e^{w_j^T \mathbf{x}}} = \frac{e^{(w_k - w_K)^T \mathbf{x}}}{\sum_{j=1}^K e^{(w_j - w_K)^T \mathbf{x}}} = \begin{cases} \frac{e^{\tilde{w}_k^T \mathbf{x}}}{1 + \sum_{j=1}^{K-1} e^{\tilde{w}_j^T \mathbf{x}}}, & k = 1, \dots, K-1 \\ \frac{1}{1 + \sum_{j=1}^{K-1} e^{\tilde{w}_j^T \mathbf{x}}}, & k = K \end{cases}$$

Softmax function



$$\sum_{k=1}^K p(C_k|\mathbf{x}) = 1 \Rightarrow p(C_K|\mathbf{x}) = 1 - \sum_{k=1}^{K-1} p(C_k|\mathbf{x})$$

# Probit Regression

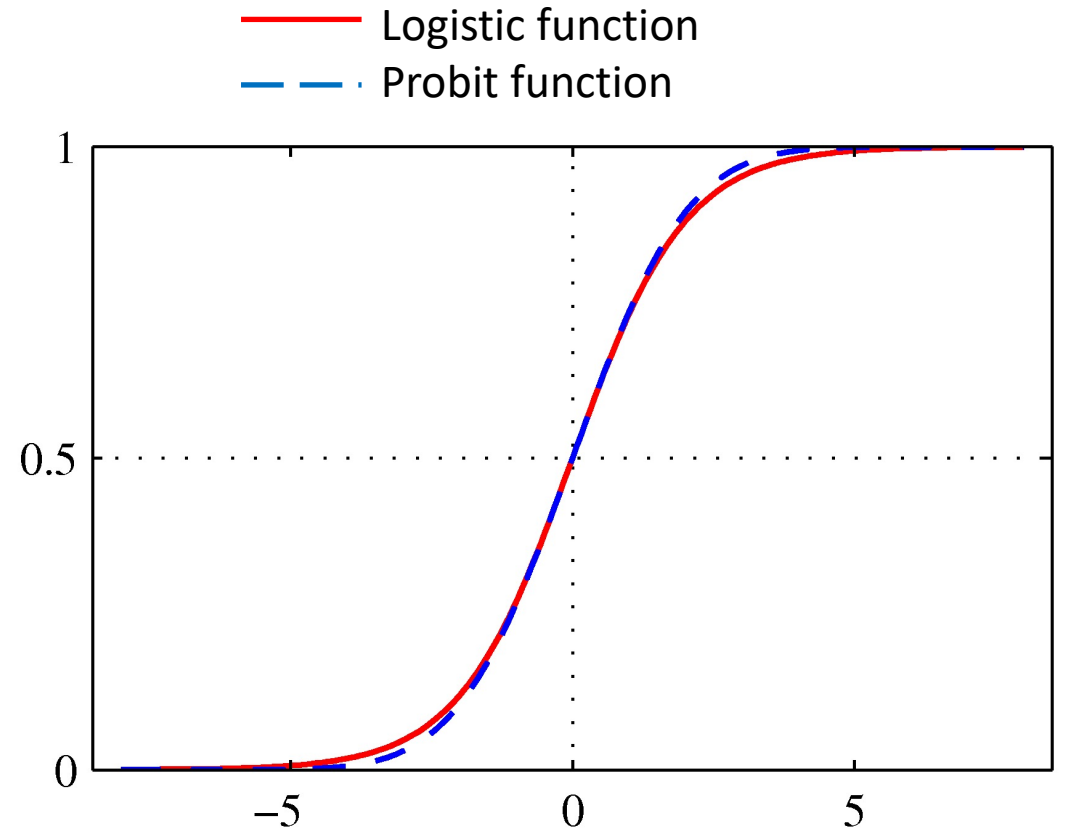
$$p(C_1|\mathbf{x}) = y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$

Logistic sigmoid function

$$p(C_1|\mathbf{x}) = y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x}) = \int_{-\infty}^{\mathbf{w}^T \mathbf{x}} \mathcal{N}(\theta|0,1) d\theta$$

Probit function

$$\hat{C}_n = \begin{cases} 1 & \text{if } \mathbf{w}^T \mathbf{x} \geq 0 \\ 0 & \text{o. w.} \end{cases}$$



# k-Nearest Neighbor (kNN) Classification

- **Training:** store the feature vectors and class labels of the training samples
- **Testing:** assign a test sample to the label which is most frequent among the k training samples nearest to the test sample

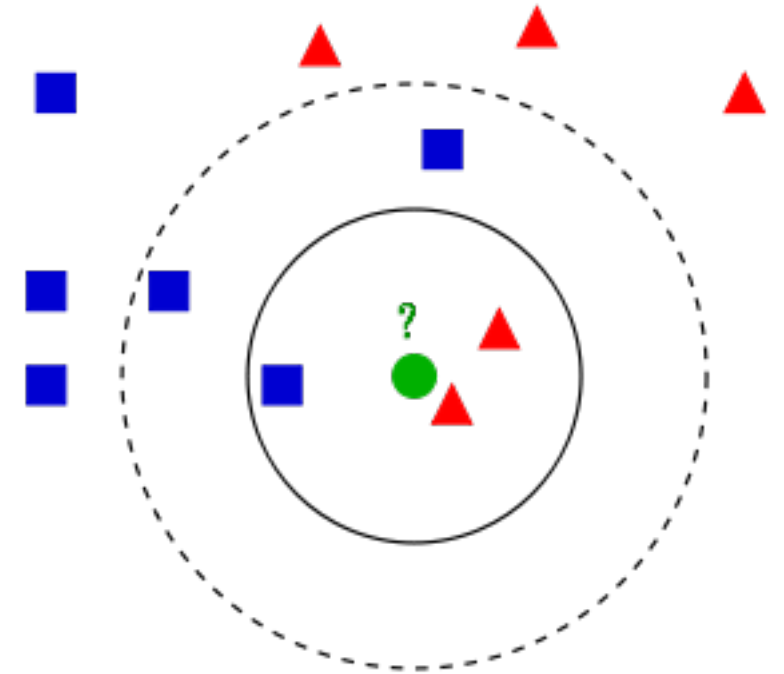
- A commonly used distance metric for continuous

variables is **Euclidean distance**:  $\|\tilde{\mathbf{x}}_j - \mathbf{x}_i\|_2 = \sqrt{(\tilde{\mathbf{x}}_j - \mathbf{x}_i)^T (\tilde{\mathbf{x}}_j - \mathbf{x}_i)}$

**Manhattan distance**:  $\|\tilde{\mathbf{x}}_j - \mathbf{x}_i\|_1 = \sum_{k=1}^d |\tilde{x}_{jk} - x_{ik}|$

- For **discrete** variables, such as for text classification, another metric can be used, e.g., **Hamming distance**

- For **spherical** variables,  $\sum_{k=1}^d x_{ik}^2 = a$ , **cosine distance**:  $1 - \cos \theta_{ji} = 1 - \frac{\tilde{\mathbf{x}}_j^T \mathbf{x}_i}{a}$



- k is an user-defined parameter: large values suppress noise, but provides less modeling power

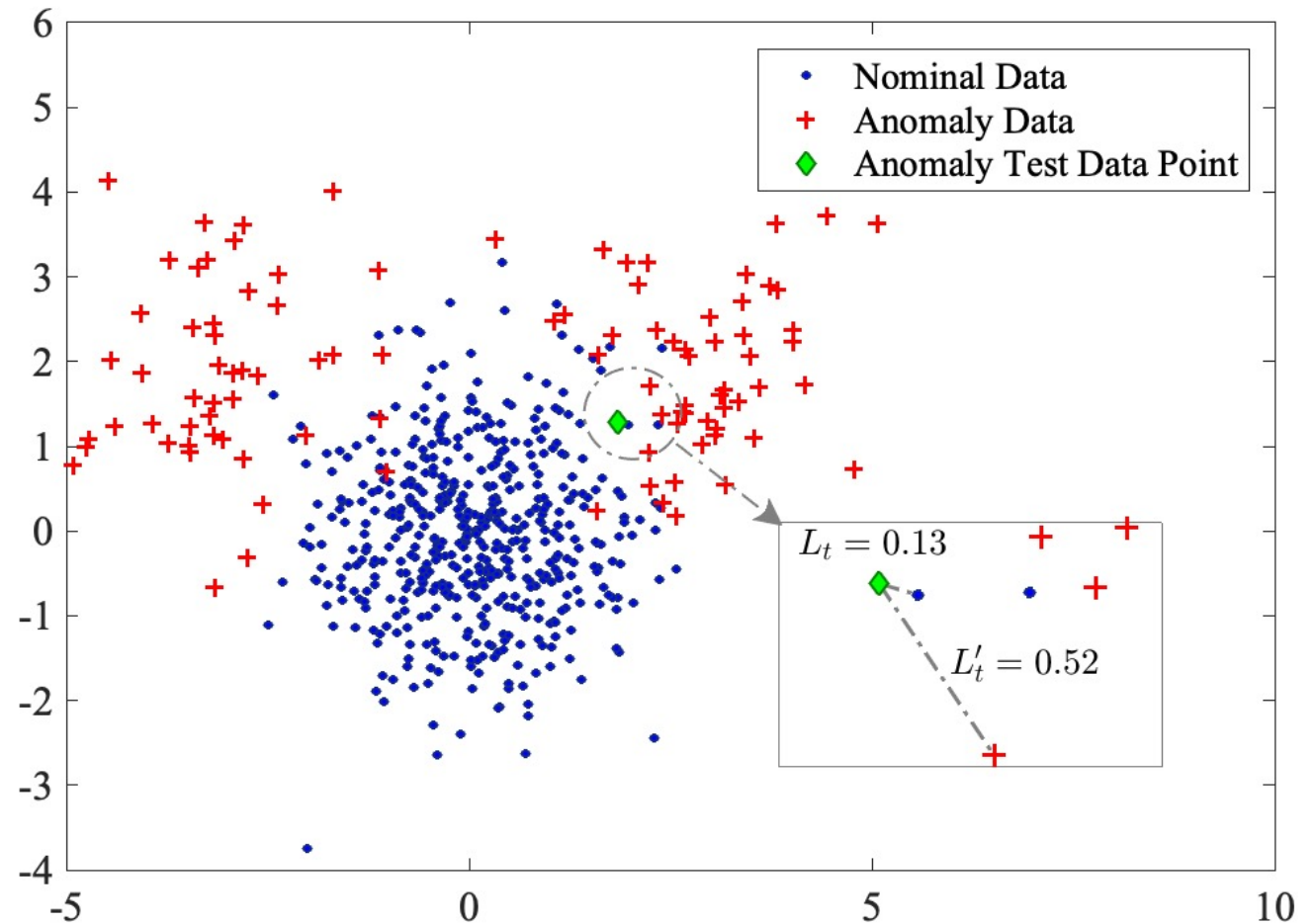
# kNN-Distance Classification

- Imbalanced datasets from each class
- More probable to have nearest neighbors from the class with more points
- Consider kNN distances with a correction factor

$$\log \frac{\frac{k/M}{V_d q_k(x)^d}}{\frac{k/N}{V_d r_k(x)^d}} =$$

$$d [\log r_k(x) - \log q_k(x)] + \log(N/M)$$

$$\rightarrow \log \frac{f_1(x)}{f_0(x)} \quad \text{as } M, N \rightarrow \infty$$



$$2 * [\log(.13) - \log(.52)] + \log(20) = 0.2231$$

# Linear Discriminant Analysis (LDA)

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

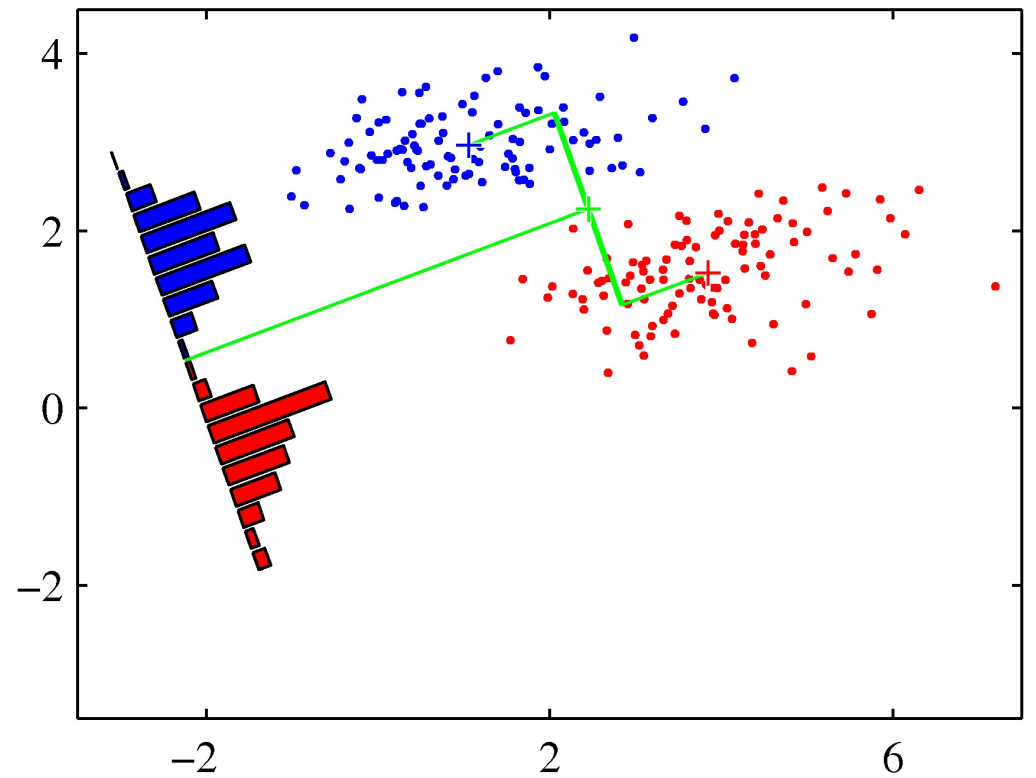
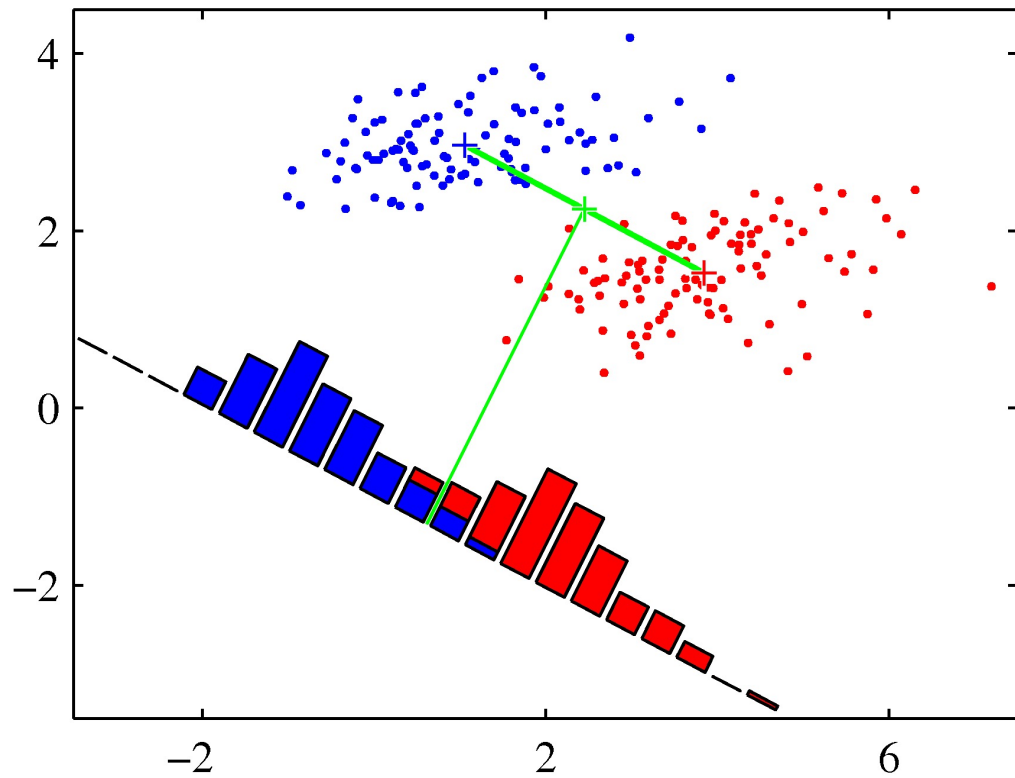
- Binary classification: choose  $C_1$  if  $y > 0$ , otherwise  $C_2$
- Projection onto 1-dimensional  $y(\mathbf{x})$  from multi-dimensional  $\mathbf{x}$
- Choose  $\mathbf{w}$  such that the class separation is maximized

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \mathbf{x}_n$$

- Maximize  $m_2 - m_1 = \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)$ , where in general  $m_k = \mathbf{w}^T \mathbf{m}_k$ , s.t.  $\sum_i w_i^2 = 1$

$$\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$$

# Linear Discriminant Analysis (LDA)



- Within-class variance  $s_k^2 = \sum_{n \in C_k} (y_n - m_n)^2$   $\longrightarrow$  total within-class variance  $s_1^2 + s_2^2$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

$$\begin{aligned} \max_{\mathbf{w}} J(\mathbf{w}) &= \max_{\mathbf{w}} \frac{(\mathbf{m}_2 - \mathbf{m}_1)^2}{s_1^2 + s_2^2} \\ &= \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \end{aligned}$$

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$



# Quadratic Discriminant Analysis (QDA)

$$y(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}$$

- Binary classification: choose  $C_1$  if  $y > 0$ , otherwise  $C_2$
- Projection onto 1-dimensional  $y(\mathbf{x})$  from multi-dimensional  $\mathbf{x}$
- Corresponds to a Normality (Gaussian) assumption like LDA, but now quadratic terms are also involved
- Closely related to Gaussian Generative Model Classifier – remember the Iris data example in Quiz 2!

$$\text{Likelihood ratio} = \frac{\sqrt{|2\pi\Sigma_{y=1}|}^{-1} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_{y=1})^T \Sigma_{y=1}^{-1} (\mathbf{x} - \mu_{y=1})\right)}{\sqrt{|2\pi\Sigma_{y=0}|}^{-1} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_{y=0})^T \Sigma_{y=0}^{-1} (\mathbf{x} - \mu_{y=0})\right)} < t$$

# LDA vs. QDA

Both LDA and QDA can be derived from simple probabilistic models which model the class conditional distribution of the data  $P(X|y = k)$  for each class  $k$ . Predictions can then be obtained by using Bayes' rule, for each training sample  $x \in \mathcal{R}^d$ :

$$P(y = k|x) = \frac{P(x|y = k)P(y = k)}{P(x)} = \frac{P(x|y = k)P(y = k)}{\sum_l P(x|y = l) \cdot P(y = l)}$$

and we select the class  $k$  which maximizes this posterior probability.

More specifically, for linear and quadratic discriminant analysis,  $P(x|y)$  is modeled as a multivariate Gaussian distribution with density:

$$P(x|y = k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) \right)$$

where  $d$  is the number of features.

# LDA vs. QDA

According to the model above, the log of the posterior is:

$$\begin{aligned}\log P(y = k|x) &= \log P(x|y = k) + \log P(y = k) + Cst \\ &= -\frac{1}{2}\log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \log P(y = k) + Cst,\end{aligned}$$

where the constant term  $Cst$  corresponds to the denominator  $P(x)$ , in addition to other constant terms from the Gaussian. The predicted class is the one that maximises this log-posterior.

## **Note: Relation with Gaussian Naive Bayes**

If in the QDA model one assumes that the covariance matrices are diagonal, then the inputs are assumed to be conditionally independent in each class, and the resulting classifier is equivalent to the Gaussian Naive Bayes classifier

[`naive\_bayes.GaussianNB`](#).

LDA is a special case of QDA, where the Gaussians for each class are assumed to share the same covariance matrix:  $\Sigma_k = \Sigma$  for all  $k$ . This reduces the log posterior to:

$$\log P(y = k|x) = -\frac{1}{2}(x - \mu_k)^t \Sigma^{-1} (x - \mu_k) + \log P(y = k) + Cst.$$