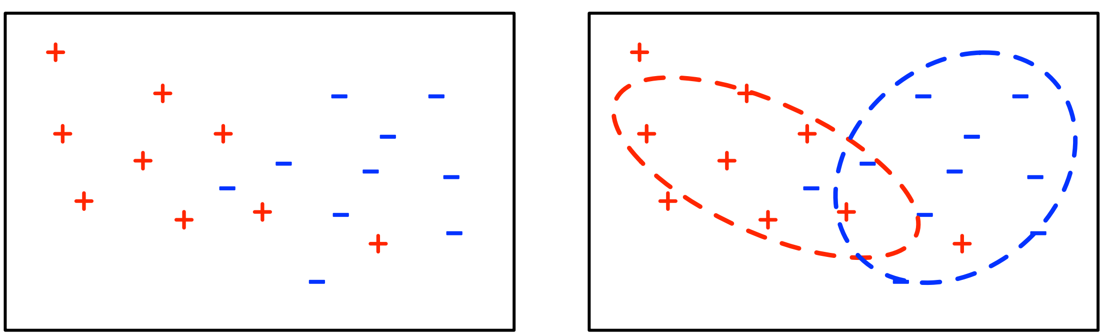
The generative approach to classification

The generative approach to classification



The learning process:

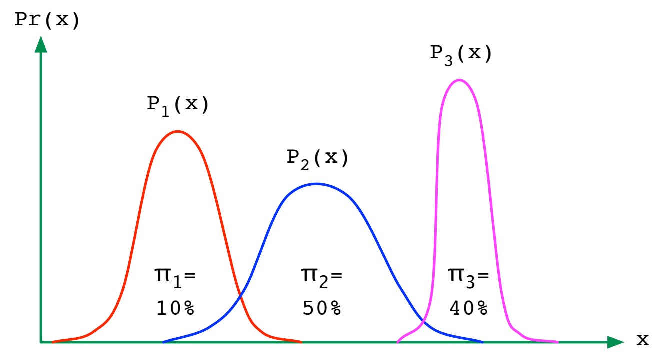
- Fit a probability distribution to each class, individually

To classify a new point:

- Which of these distributions was it most likely to have come from?

Generative models

Example:
Data space $\mathcal{X} = \mathbb{R}$
Classes/labels $\mathcal{Y} = \{1, 2, 3\}$



For each class j, we have:

- the probability of that class, $\pi_j = \Pr(y = j)$
- the distribution of data in that class, $P_j(x)$

Overall **joint distribution**: $\Pr(x, y) = \Pr(y)\Pr(x|y) = \pi_y P_y(x)$.

To classify a new x: pick the label y with largest $\Pr(x, y)$

A classification problem

You have a bottle of wine whose label is missing.

Which winery is it from, 1, 2, or 3?

Solve this problem using visual and chemical features of the wine.

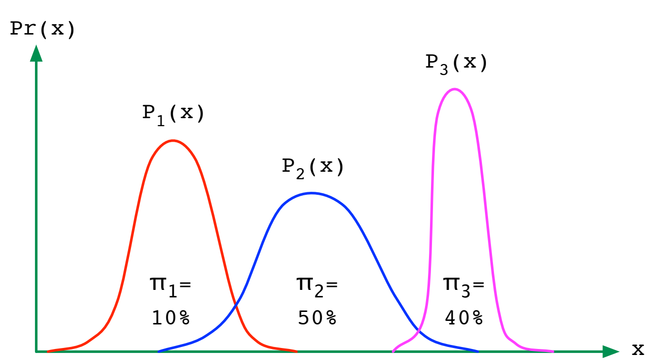
The data set

Training set obtained from 130 bottles

- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles
- For each bottle, 13 features:
  'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium',
  'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
  'Proanthocyanins',
  'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

Also, a separate test set of 48 labeled points.

Recall: the generative approach



For any data point $x \in \mathcal{X}$ and any candidate label j,

$$\Pr(y=j|x) = \frac{\Pr(y=j)\Pr(x|y=j)}{\Pr(x)} = \frac{\pi_j P_j(x)}{\Pr(x)}$$

Optimal prediction: the class j with largest $\pi_j P_j(x)$.

Fitting a generative model

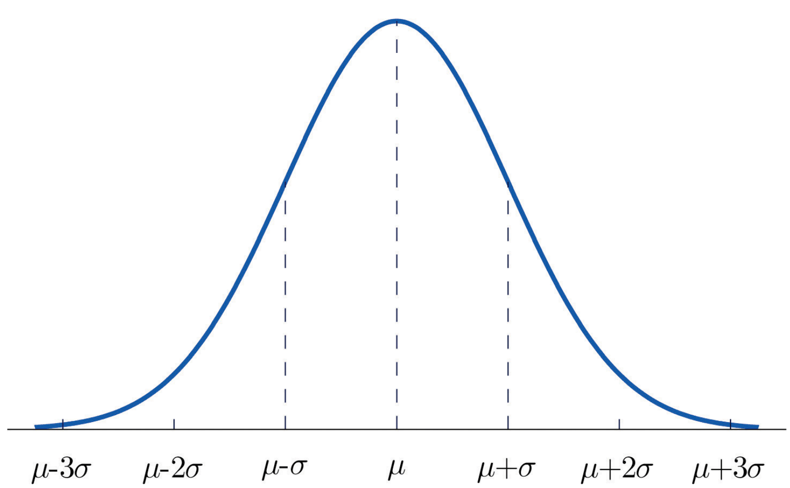
Training set of 130 bottles:

- Winery 1: 43 bottles, winery 2: 51 bottles, winery 3: 36 bottles
- For each bottle, 13 features: 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash','Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

Class weights:
$\pi_1 = 43/130 = 0.33, \quad \pi_2 = 51/130 = 0.39, \quad \pi_3 = 36/130 = 0.28$

Need distributions P_1, P_2, P_3, one per class.
Base these on a single feature: 'Alcohol'.

The univariate Gaussian

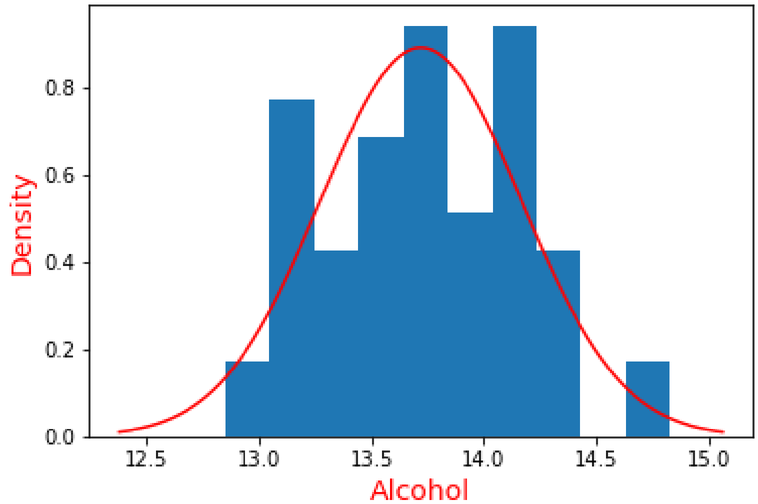


The Gaussian $N(\mu, \sigma^2)$ has mean μ, variance σ^2, and density function

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

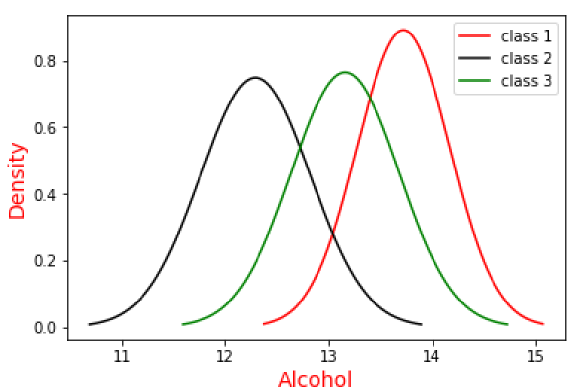
The distribution for winery 1

Single feature: 'Alcohol'



Mean $\mu = 13.72$, Standard deviation $\sigma = 0.44$ (variance 0.20)

All three wineries



- $\pi_1 = 0.33,\ P_1 = N(13.7, 0.20)$
- $\pi_2 = 0.39,\ P_2 = N(12.3, 0.28)$
- $\pi_3 = 0.28,\ P_3 = N(13.2, 0.27)$

To classify x: Pick the j with highest $\pi_j P_j(x)$

Test error: 14/48 = 29%

What if we use **two** features?

The data set, again

Training set obtained from 130 bottles

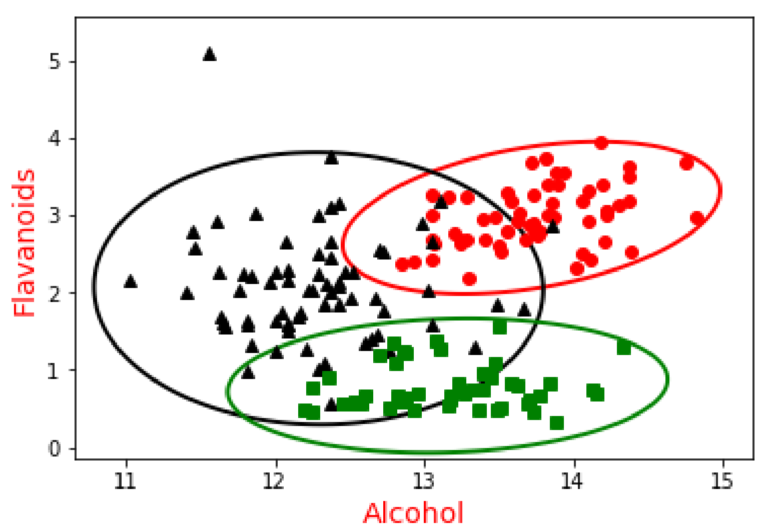
- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles
- For each bottle, 13 features:
  'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash','Magnesium',
  'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
  'Proanthocyanins',
  'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

Also, a separate test set of 48 labeled points.

This time: 'Alcohol' and 'Flavanoids'.

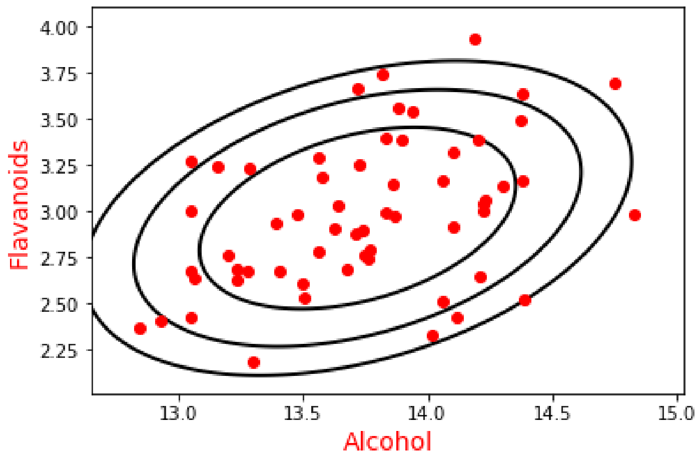
Why it helps to add features

Better **separation** between the classes!



Error rate drops from 29% to 8%.

The bivariate Gaussian



Model class 1 by a bivariate Gaussian, parametrized by:

$$\text{mean } \mu = \begin{pmatrix} 13.7 \\ 3.0 \end{pmatrix} \text{ and covariance matrix } \Sigma = \begin{pmatrix} 0.20 & 0.06 \\ 0.06 & 0.12 \end{pmatrix}$$

Dependence between two random variables

Suppose X_1 has mean μ_1 and X_2 has mean μ_2.

Can measure dependence between them by their **covariance**:

- $\mathrm{cov}(X_1, X_2) = \mathbb{E}[(X_1 - \mu_1)(X_2 - \mu_2)] = \mathbb{E}[X_1 X_2] - \mu_1 \mu_2$
- Maximized when $X_1 = X_2$, in which case it is $\mathrm{var}(X_1)$.
- It is at most $\mathrm{std}(X_1)\mathrm{std}(X_2)$.

The bivariate (2-d) Gaussian

A distribution over $(x_1, x_2) \in \mathbb{R}^2$, parametrized by:

- **Mean** $(\mu_1, \mu_2) \in \mathbb{R}^2$, where $\mu_1 = \mathbb{E}(X_1)$ and $\mu_2 = \mathbb{E}(X_2)$
- **Covariance matrix** $\Sigma = \left[\begin{array}{cc} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{array} \right]$ where

$$\left\{ \begin{array}{c} \Sigma_{11} = \text{var}(X_1) \\ \Sigma_{22} = \text{var}(X_2) \\ \Sigma_{12} = \Sigma_{21} = \text{cov}(X_1, X_2) \end{array} \right\}$$

Density is highest at the mean, falls off in ellipsoidal contours.

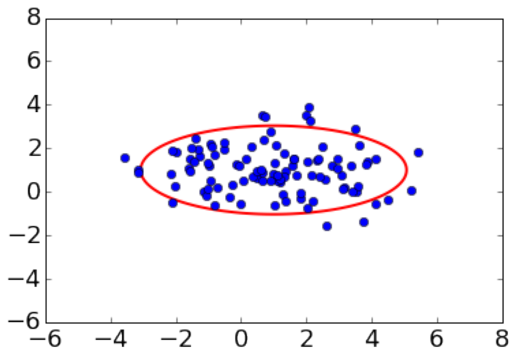
Density of the bivariate Gaussian

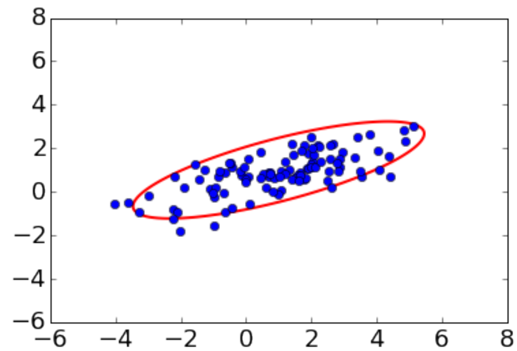
- **Mean** $(\mu_1, \mu_2) \in \mathbb{R}^2$, where $\mu_1 = \mathbb{E}(X_1)$ and $\mu_2 = \mathbb{E}(X_2)$
- **Covariance matrix** $\Sigma = \left[\begin{array}{cc} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{array} \right]$

Density $p(x_1, x_2) = \dfrac{1}{2\pi |\Sigma|^{1/2}} \exp\left(-\dfrac{1}{2} \left[\begin{array}{c} x_1 - \mu_1 \\ x_2 - \mu_2 \end{array} \right]^T \Sigma^{-1} \left[\begin{array}{c} x_1 - \mu_1 \\ x_2 - \mu_2 \end{array} \right] \right)$

Bivariate Gaussian: examples

In either case, the mean is $(1, 1)$.

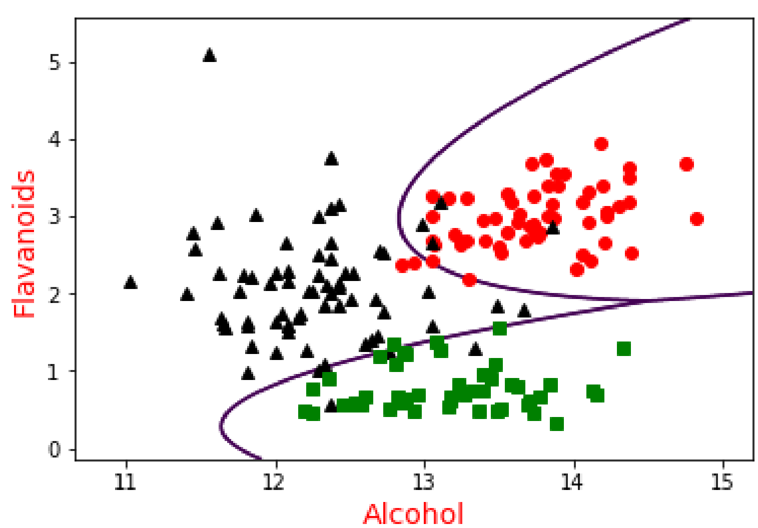




$$\Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 4 & 1.5 \\ 1.5 & 1 \end{bmatrix}$$

The decision boundary

Go from 1 to 2 features: error rate goes from 29% to 8%.



What kind of function is this? And, can we use more features?