**PS5841**

# Data Science in Finance & Insurance

# Naïve Bayes

Yubo Wang

Autumn 2022

Columbia University
in the City of New York

# Bayes Classifier

- Classifies an observation to the most likely class
  - Discrete $Y$; Continuous & Discrete features $X$

$$\underset{k}{\operatorname{argmax}} \Pr(Y = k | X = \boldsymbol{x})$$

- Has the lowest possible test error rate out of all classifiers
- Bayes error rate at $\boldsymbol{x}$

$$1 - \max_k \Pr(Y = k | X = \boldsymbol{x})$$

- Overall Bayes error rate

$$1 - E\left(\max_k \Pr(Y = k | X = \boldsymbol{x})\right)$$

# Conditional Class Probabilities

- Use Bayes Theorem

$$\Pr(Y = k | X = \boldsymbol{x}) = \frac{\Pr(X = \boldsymbol{x} | Y = k) \Pr(Y = k)}{\sum_{l=1}^{K} \Pr(X = \boldsymbol{x} | Y = l) \Pr(Y = l)}$$

$$p_k(\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{l=1}^{K} f_l(\boldsymbol{x})\pi_l}$$

# Bayes Classifier

- Assigned class maximizes

$$p_k(\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{l=1}^{K} f_l(\boldsymbol{x})\,\pi_l}$$

- Estimation
  - prior

$$\hat{\pi}_k = \frac{n_k}{N}$$

  - likelihood

$$\hat{f}_k(\boldsymbol{x}) = ?$$

# Naïve Bayes Classifier

- Assumption

$$f_k(\boldsymbol{x}) = \prod_{j=1}^{p} f_{kj}(x_j), \qquad \boldsymbol{x} = \left(x_1, \ldots, x_p\right)^T$$

- posterior

$$p_k(\boldsymbol{x}) = \frac{\pi_k \prod_{j=1}^{p} f_{kj}(x_j)}{\sum_{l=1}^{K} \pi_l \prod_{j=1}^{p} f_{lj}(x_j)}$$

# Ex: 2 quantitative features & 2 classes

- For a particular observation $(x_1, x_2)$

$$p_1[(x_1, x_2)] = \frac{\pi_1 f_{11}(x_1) f_{12}(x_2)}{\pi_1 f_{11}(x_1) f_{12}(x_2) + \pi_2 f_{21}(x_1) f_{22}(x_2)}$$

$$p_2[(x_1, x_2)] = \frac{\pi_2 f_{21}(x_1) f_{22}(x_2)}{\pi_1 f_{11}(x_1) f_{12}(x_2) + \pi_2 f_{21}(x_1) f_{22}(x_2)}$$

- Classify the observation to class 1 if

$$p_1[(x_1, x_2)] > p_2[(x_1, x_2)]$$

or

$$p_1[(x_1, x_2)] > \text{custom threshold}$$

# Applications

- Insurance
  - Insurable Risk classification
- Finance
  - Credit risk classification
  - Fraud detection
  - Market timing
- Text
  - Document classification
  - Email spam filter
- Healthcare

# Gaussian Naïve Bayes

- Quantitative features
- The $j$-th feature in the $k$-th class $\sim N(\mu_{kj}, \sigma_{kj}^2)$
- Estimation

$$\hat{\mu}_{kj} = \frac{1}{n_k} \sum_{i:y_i=k} x_{ij}$$

$$\hat{\sigma}_{kj}^2 = \frac{1}{n_k - 1} \sum_{i:y_i=k} \left(x_{ij} - \hat{\mu}_{kj}\right)^2$$

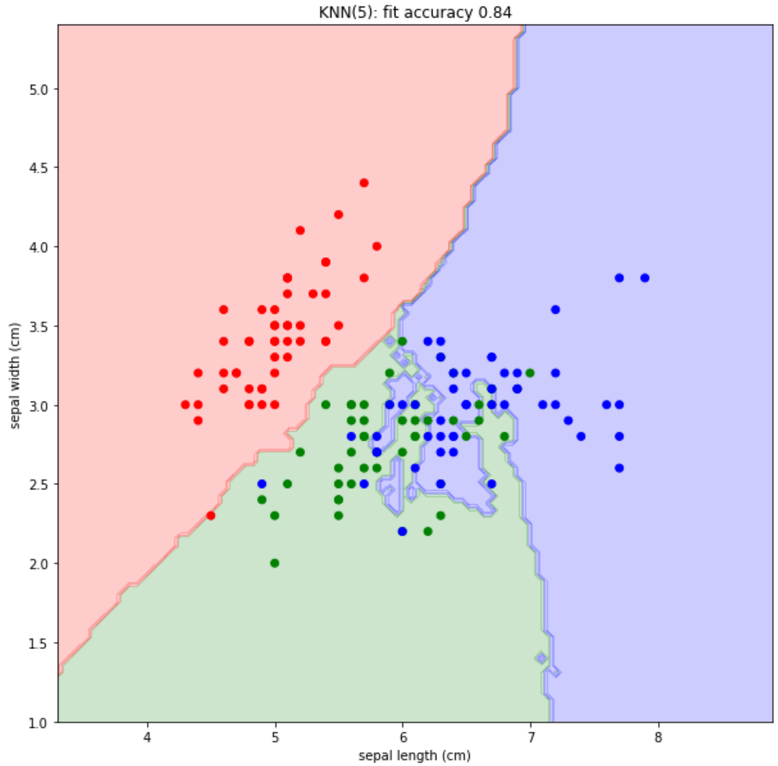
- Likelihood

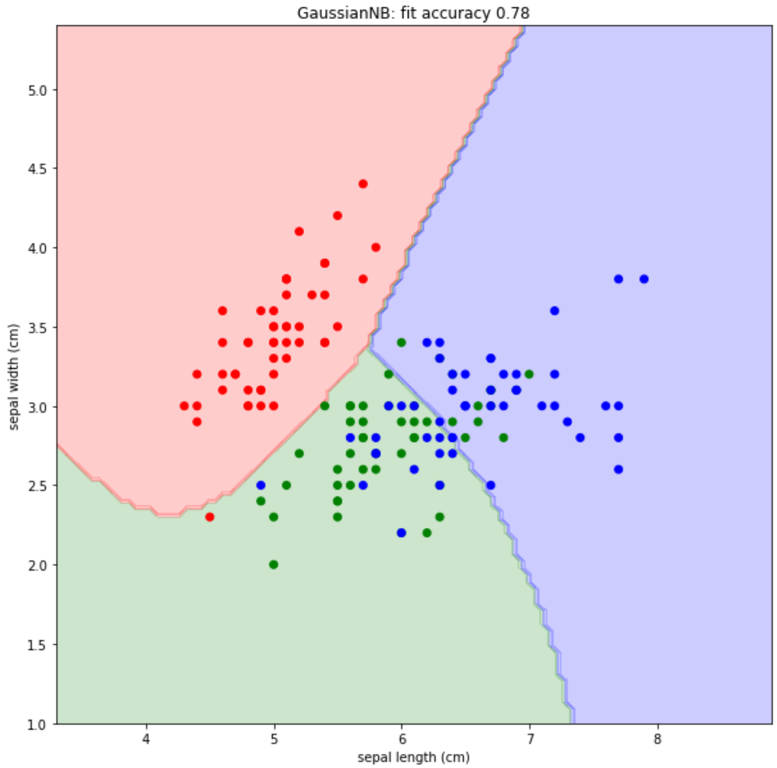
$$\hat{f}_k(x_j) = \frac{1}{\hat{\sigma}_{kj}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_j - \hat{\mu}_{kj}}{\hat{\sigma}_{kj}}\right)^2\right)$$

# Example

| | | | | | |
|---|---|---|---|---|---|
| Category | Value | | | | |
| Label | Feature 1 | | A-priori probabilities | | |
| C1 | 86 | | Y | cat_1 | cat_2 |
| C1 | 15 | | | 0.4 | 0.6 |
| C1 | 40 | | | | |
| C1 | 33 | | Conditional probabilities | | |
| C2 | 25 | | | Value | |
| C2 | 38 | | | mean | sd |
| C2 | 73 | | cat_1 | 43.50000 | 30.22692 |
| C2 | 79 | | cat_2 | 48.83333 | 22.86846 |
| C2 | 28 | | | | |
| C2 | 50 | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Feature 1 | | likelihood | prior | posterior | |
| 15 | cat_1 | 0.00846 | 0.4 | 0.491372 | |
| 15 | cat_2 | 0.00584 | 0.6 | 0.508628 | |

# Decision Boundary: GaussianNB vs KNN



# Coding: Gaussian Naïve Bayes

- R

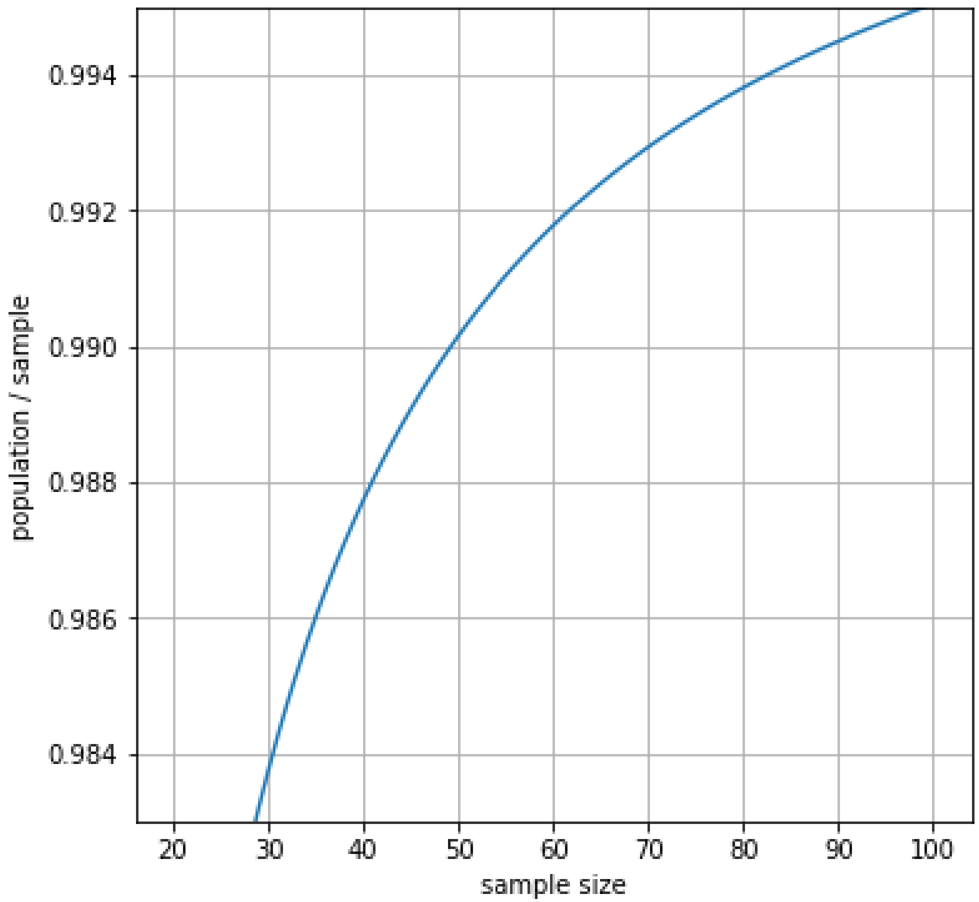
  e1071::naiveBayes()

- Python

  sklearn.naive_bayes.GaussianNB()

  – (version 1.1.2) used population stdev formula for $\hat{\sigma}_{kj}^2$

# Population vs Sample SD



# Multinomial Naïve Bayes

- Qualitative features
- The $j$-th feature (having L levels) in the $k$-th class

$$\sim MultiN\left(n_k = \sum_{l=1}^{L} n_{kl}\,, \boldsymbol{\theta} = (\theta_1, \dots, \theta_L)^T\right)$$

- Estimation

$$\widehat{\boldsymbol{\theta}} = \left(\frac{n_{k1}}{n_k}, \dots, \frac{n_{kL}}{n_k}\right)$$

- Likelihood

$$\hat{f}_k(x_j) = \frac{n_k!}{n_{k1}! \dots n_{kL}!} \hat{\theta}_1^{n_{k1}} \dots \hat{\theta}_L^{n_{kL}}$$

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| Category | Quality | Temp | | A-priori probabilities | | |
| Label | Feature 1 | Feature 2 | | Y | cat_1 | cat_2 |
| C1 | B | H | | | 0.4 | 0.6 |
| C1 | G | C | | | | |
| C1 | G | H | | Conditional probabilities | | |
| C1 | G | H | | | Quality | |
| C2 | B | C | | | B | G |
| C2 | B | C | | cat_1 | 0.25000 | 0.75000 |
| C2 | G | H | | cat_2 | 0.33333 | 0.66667 |
| C2 | G | H | | | | |
| C2 | G | H | | Conditional probabilities | | |
| C2 | G | C | | | Temp | |
| | | | | | C | H |
| | | | | cat_1 | 0.25000 | 0.75000 |
| | | | | cat_2 | 0.50000 | 0.50000 |
| | | | | | | |
| Feature 1 | Feature 2 | | likelihood | prior | posterior | |
| B | H | cat_1 | 0.187500 | 0.4 | 0.4285714 | |
| B | H | cat_2 | 0.166667 | 0.6 | 0.5714286 | |
| | | | | | | |
| Feature 1 | Feature 2 | | likelihood | prior | posterior | |
| G | C | cat_1 | 0.187500 | 0.4 | 0.2727273 | |
| G | C | cat_2 | 0.333333 | 0.6 | 0.7272727 | |

# Coding: Multinomial Naïve Bayes

- R

  e1071::naiveBayes()

- Python

  sklearn.naive_bayes.MultinomialNB()

  – Encode feature using a one-hot (aka ‘one-of-K’ or ‘dummy’) encoding scheme.

# Coding: Naïve Bayes w/ Q & C Features

- R

e1071::naiveBayes()

- Python

use estimates from

sklearn.naive_bayes.GaussianNB()*

sklearn.naive_bayes.MultinomialNB()

to calculation posterior class probabilities

# That was