Probabilistic Classification Models

Peerapon S.

Machine Learning (67/2)

Topics

- ☐ Review of Bayes Theorem
- ☐ Linear Discriminant Analysis
- ☐ Quadratic Discriminant Analysis
- ☐ Naives' Bayes classifier

Example

- Consider four boxes with percentage of defected components as 5%, 40%, 10%, and 10%.
  - A component is sampled from a randomly selected box .
  - What is Pr{Box 2 is selected if the selected component is defective} ?
- Evidence = Selected component is defective.
- Event of interest = Box 2 is selected.

Review of Bayes Theorem

- For an event of interest Y and some evidence event X,
  - $Pr\{Y\}$ is called ***prior probability.***
  - $Pr\{X\}$ is called ***evidence probability.***
  - $Pr\{X|Y\}$ is called ***likelihood probability.***

- Given that the evidence event X occurs, the ***posterior probability*** $Pr\{Y|X\}$ can be determined by

$$\begin{aligned}\mathbb{P}\{Y \mid X\} &= \frac{\mathbb{P}\{Y \cap X\}}{\mathbb{P}\{X\}} \\ &= \frac{\mathbb{P}\{X \mid Y\}\mathbb{P}\{Y\}}{\mathbb{P}\{X\}}\end{aligned}$$

(Generalized) Bayes Theorem

- Let $B_j's$, $1 \leq j \leq n$, be a set of exhaustively mutually exclusive events.
- For an arbitrary event A,
  - $Pr\{B_k\}$ is called ***prior probability***, e.g, target class k.
  - $Pr\{A\}$ is called ***evidence probability***, e.g, features.
  - $Pr\{A \mid B_k\}$ is called ***likelihood probability.***
- Given that event A occurs, the ***posterior probability*** $Pr\{B_k \mid A\}$ can be determined by

$$\mathbb{P}\{B_k \mid A\} = \frac{\mathbb{P}\{A \cap B_k\}}{\mathbb{P}\{A\}} = \frac{\mathbb{P}\{A \mid B_k\}\mathbb{P}\{B_k\}}{\sum_{i=1}^{n} \mathbb{P}\{A \mid B_i\}\mathbb{P}\{B_i\}}$$

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

Example

☐ A certain cancer test is 95% reliable.

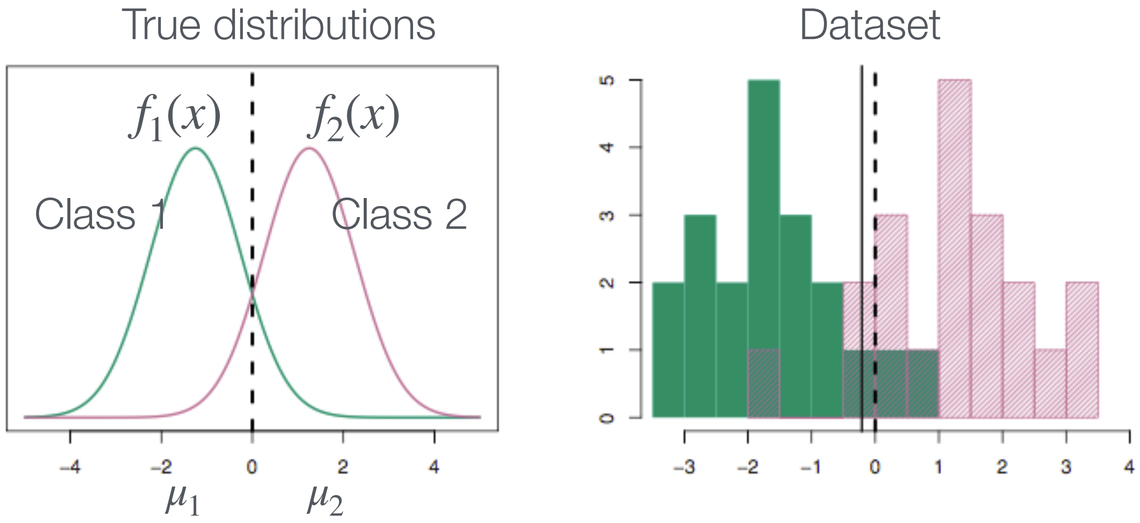
- For a sick person, the test is positive 95% of the time.
- For a healthy person, the test is negative 95% of the time.

A person randomly selected from a large population submits to the test and the result is positive.
Suppose we know that 2% of the population actually has the cancer.
What is the probability that the person actually has the cancer ?

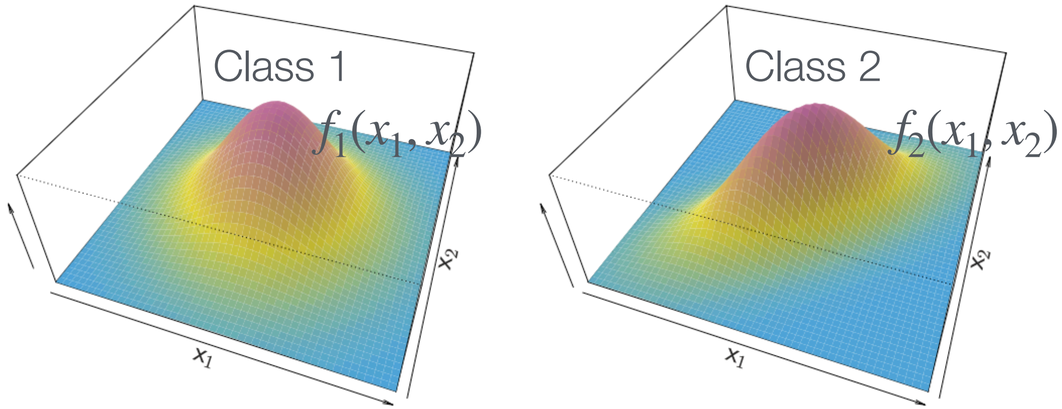
Class-Likelihood Distributions

Distribution of features by class $f_k(\boldsymbol{x})$

- One-dimensional (Univariate) feature
- Two-dimensional (Bivariate) features
- Multi-dimensional (Multivariate) features



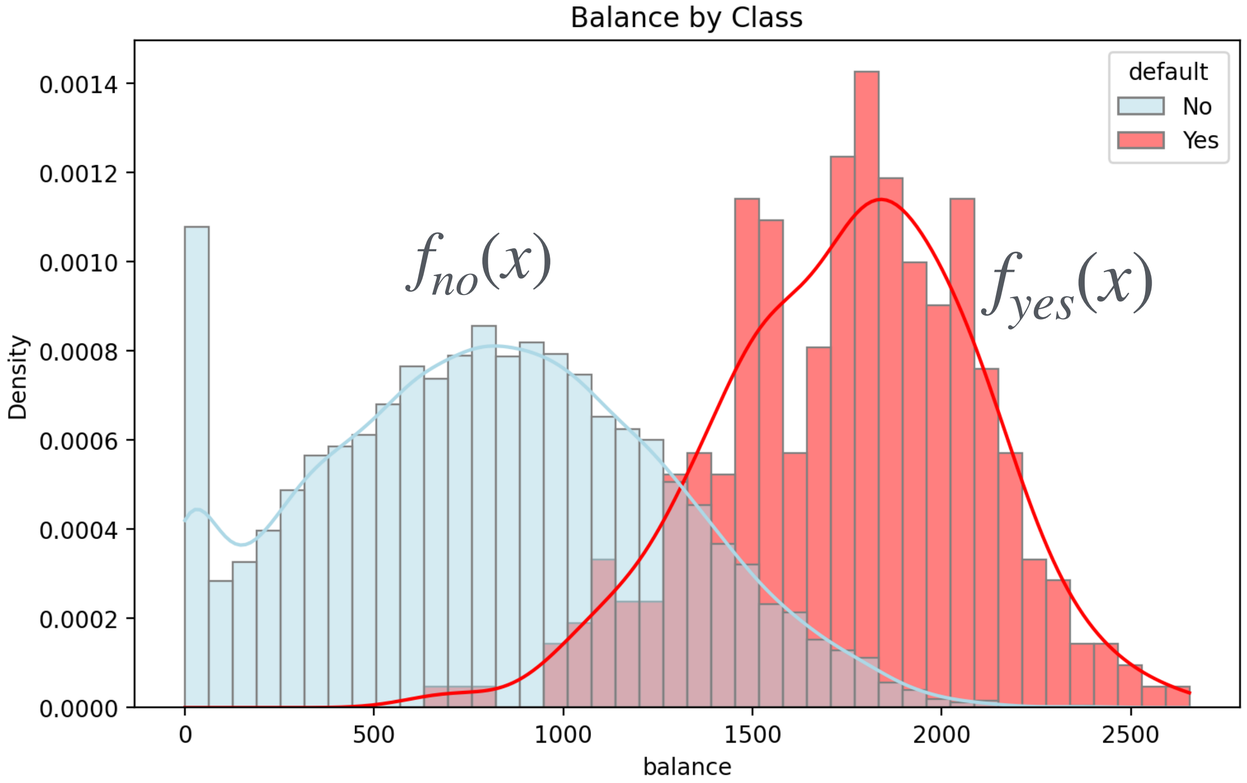
Univariate data

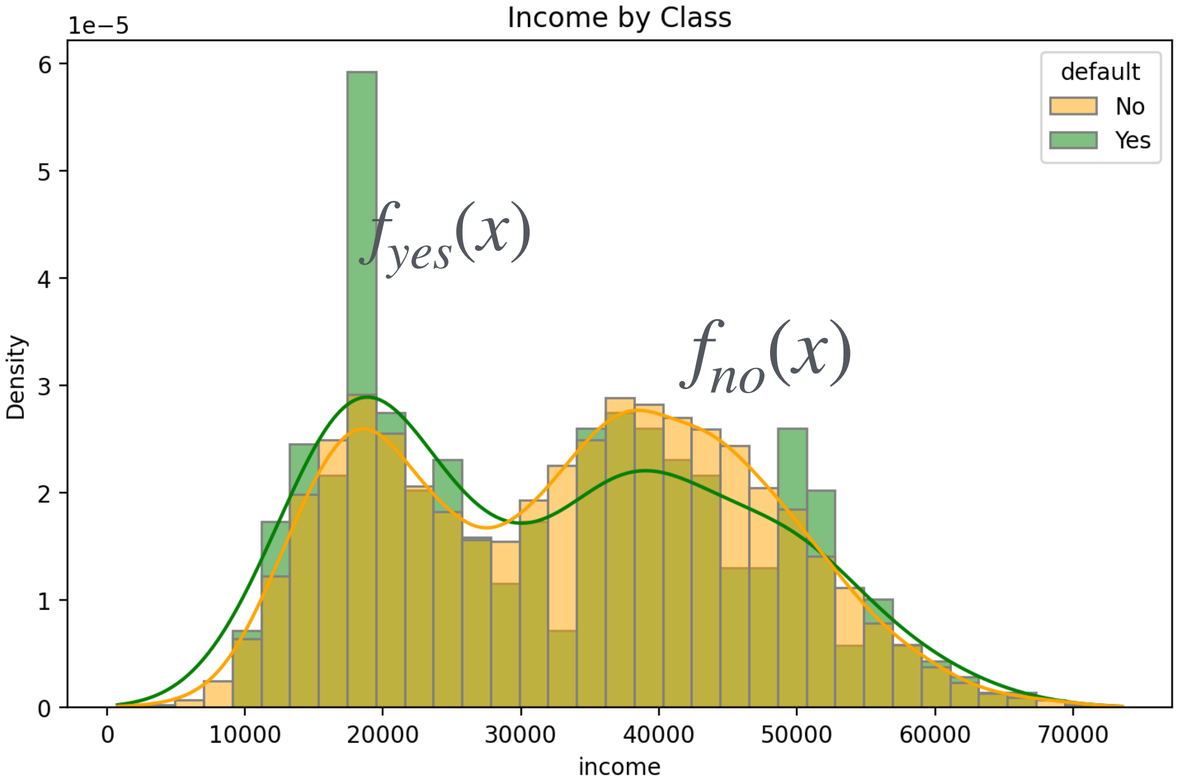


Bivariate data

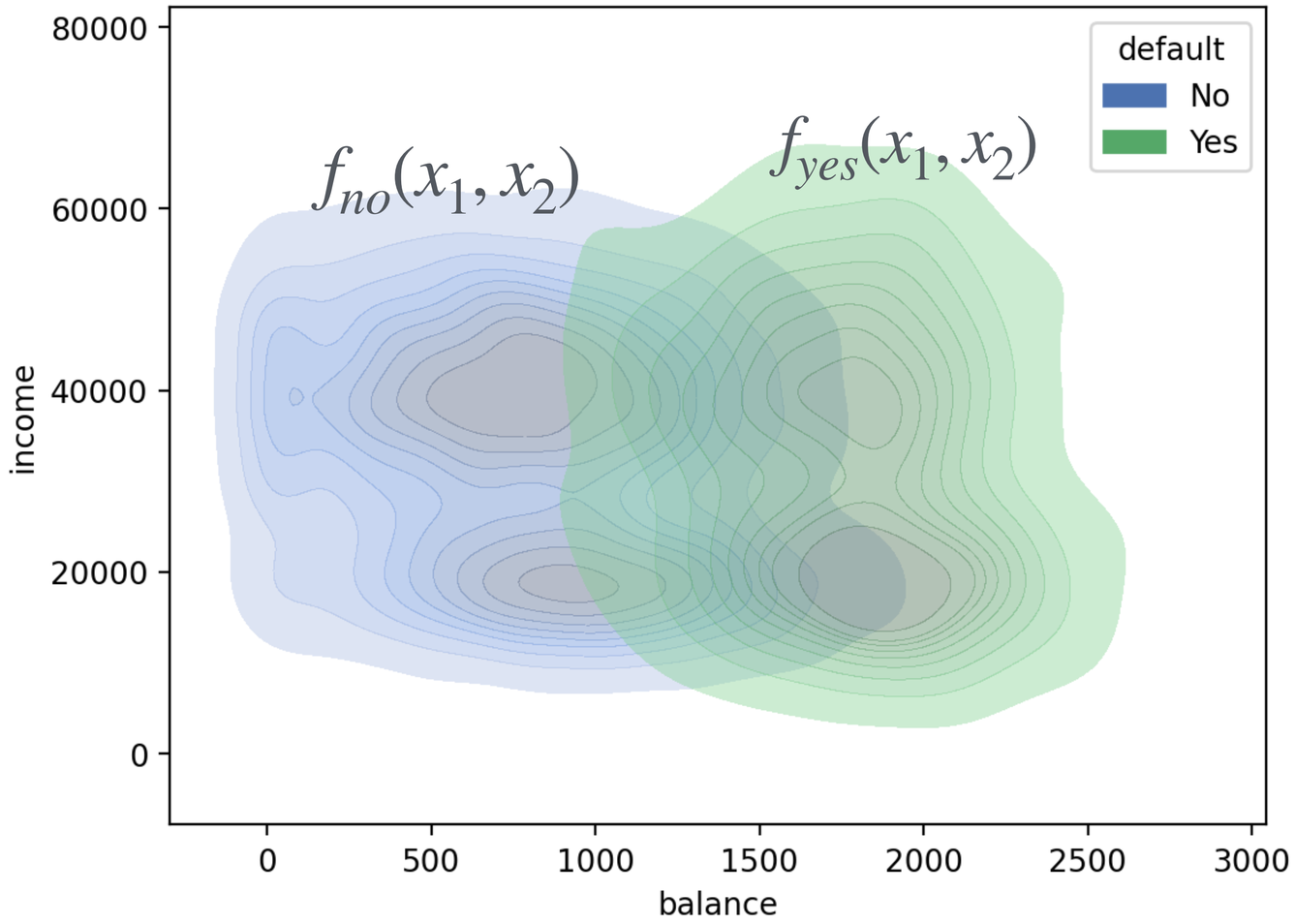
Example

	No.	default	student	balance	income
0	1	No	No	729.526495	44361.625074
1	2	No	Yes	817.180407	12106.134700
2	3	No	No	1073.549164	31767.138947
3	4	No	No	529.250605	35704.493935
4	5	No	No	785.655883	38463.495879
5	6	No	Yes	919.588530	7491.558572
6	7	No	No	825.513331	24905.226578
7	8	No	Yes	808.667504	17600.451344
8	9	No	No	1161.057854	37468.529288
...					





Joint distribution of income and balance for each default status



Generative Model for Classification

- In logistic regression, the posterior probability $Pr\{Y = k | \boldsymbol{X} = \boldsymbol{x}\}$ is estimated directly by fitting data to a logistic function, hence a ***discriminative model***.
- Alternatively, the posterior probability can be determined by using Bayes' Theorem:

$$\begin{aligned}\mathbb{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\} &= \frac{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\}\mathbb{P}\{Y = k\}}{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x}\}} \\ &= \frac{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\}\mathbb{P}\{Y = k\}}{\sum_j \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = j\}\mathbb{P}\{Y = j\}} \\ &= \frac{f_k(\boldsymbol{x})\pi_k}{\sum_j \pi_j f_j(\boldsymbol{x})}\end{aligned}$$

- Knowing $f_k(\boldsymbol{x})$, the joint distribution $Pr\{Y = k, X = \boldsymbol{x}\}$ can be determined, hence a ***generative model***.

Discriminant Analysis Classification

Let π_k be the proportion of class k from the binary target variable (*class prior probability*).

Given input $\boldsymbol{x}$, the probability that Y will come from class 1 (or 2) can be derived from ***Bayes theorem*** as

Class 1 $$p_1(\boldsymbol{x}) = \frac{\pi_1 f_1(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x}) + \pi_2 f_2(\boldsymbol{x})} = \frac{\delta_1(\boldsymbol{x})}{\delta_1(\boldsymbol{x}) + \delta_2(\boldsymbol{x})}$$

Class 2 $$p_2(\boldsymbol{x}) = \frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x}) + \pi_2 f_2(\boldsymbol{x})} = \frac{\delta_2(\boldsymbol{x})}{\delta_1(\boldsymbol{x}) + \delta_2(\boldsymbol{x})}$$

Classification usually done by selecting the most probable class, i.e., ***largest discriminant score*** $\delta(\boldsymbol{x})$.

The ***discrimination function*** $\log \delta(\boldsymbol{x})$ is typically used.

The same concept extends to more than two classes.

Three Approaches to Estimating Class-likelihood Distributions

Linear discriminant analysis (LDA)

Quadratic discriminant analysis (QDA)

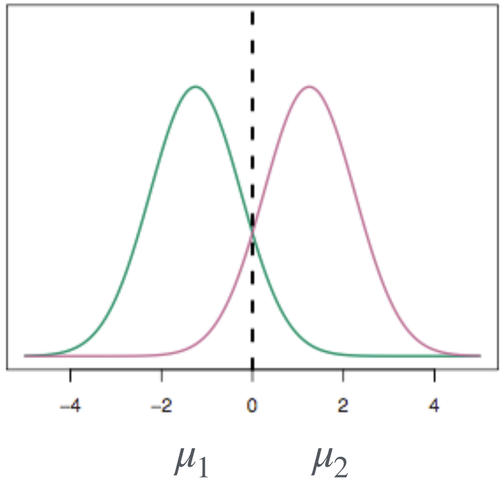
Naives Bayes (NB)

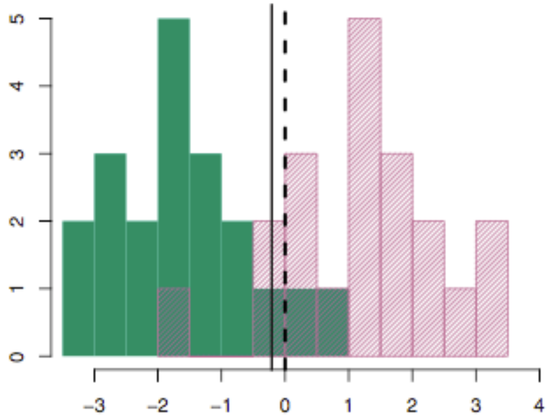
Linear Discriminant Analysis (LDA) Classifier

- Consider the case of a single predictor ($p = 1$).
- Assume that observations in class $k \sim \text{Normal}(\mu_k, \sigma^2)$, $1 \leq k \leq K$

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2}$$
$$= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu_k)^2}$$

$K = 2$





☐ So, the LDA for a single predictor has the discriminant function:

$$\begin{aligned}\delta_k(x) &= \log f_k(x) + \log \pi_k \\ &= -\frac{1}{2\sigma^2}(x-\mu_k)^2 - \log(\sqrt{2\pi}\sigma) + \log \pi_k\end{aligned}$$

☐ The distribution mean and standard deviation can be estimated by

$$\begin{aligned}\hat{\mu}_k &= \frac{1}{n_k}\sum_{i:y_i=k} x_i \\ \hat{\sigma}^2 &= \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)^2\end{aligned}$$

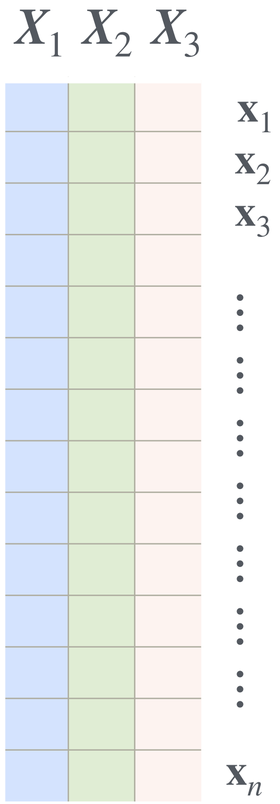
☐ Plugging in $\hat{\mu}_k$, $\hat{\sigma}$, $\hat{\pi}_k$ and removing constant terms, the LDA classifier assigns the observation x to class k for which the following ***linear discriminant function*** is largest:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} + \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Sample Covariance Matrix of a Random Vector

- Represent all pair-wise linear relationships in dataset.
- Let $\mathbf{X}$ be an $(n \times p)$ data matrix with columns $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p\}$ and rows $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

$$\boldsymbol{X} =$$



$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbb{C}\text{ov}(X_1, X_1) & \mathbb{C}\text{ov}(X_1, X_2) & \mathbb{C}\text{ov}(X_1, X_3) \\ \mathbb{C}\text{ov}(X_2, X_1) & \mathbb{C}\text{ov}(X_2, X_2) & \mathbb{C}\text{ov}(X_2, X_2) \\ \mathbb{C}\text{ov}(X_3, X_1) & \mathbb{C}\text{ov}(X_3, X_2) & \mathbb{C}\text{ov}(X_3, X_3) \end{bmatrix}$$

$$\mathbb{C}\text{ov}(X_i, X_j) = \frac{\sum_{i=1}^{n}(X_i - \overline{X}_i)(X_j - \overline{X}_j)}{n-1} \quad \text{(Sample covariance)}$$

$$\mathbb{C}\text{ov}(X_i, X_i) = \mathbb{V}\text{ar}\{X_i\}$$

Covariance Matrix

X_1	X_2	X_3
11.01	10	2.7
11.45	13.7	3.7
7.32	12.2	3.29
8.86	6.6	1.78
8.41	10.1	2.73
9.18	6.3	1.7
9.57	7.5	2.03
9.53	7.1	1.92
9.71	8.7	2.35
14.69	11.2	3.02
9.4	12.7	3.43
8.82	9.1	2.46
6.35	11.2	3.02
8.65	12.3	3.32
9.14	13.1	3.54
7.23	5.4	1.46
4.02	13.8	3.73
5.83	8.1	2.19
7.46	10	2.7
8.68	13	3.51
9.4	10.6	2.86
9.43	5.7	1.54
8.18	9.5	2.57
8.53	7.7	2.08
10.67	10.3	2.78
11.28	15.5	4.19
8.22	9.5	2.57
8.93	10.03	2.71
4.01	7.28	0.53

Mean
Variance

$$\mathbb{C}\text{ov}\{X_1, X_2\} \approx \frac{\sum_{j=1}^{n}(X_{1j} - 8.93)(X_{2j} - 10.03)}{n-1} = 0.516$$

$$\mathbb{C}\text{ov}\{X_1, X_3\} \approx \frac{\sum_{j=1}^{n}(X_{1j} - 8.93)(X_{3j} - 2.71)}{n-1} = 0.138$$

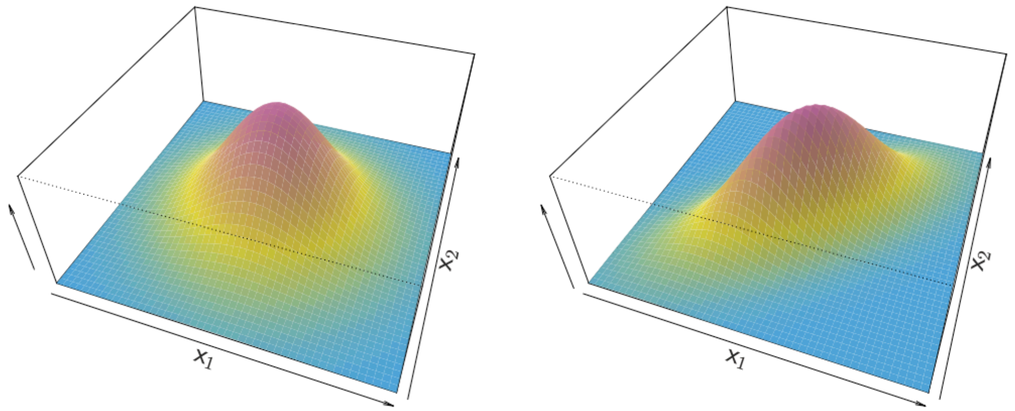
$$\mathbb{C}\text{ov}\{X_2, X_3\} \approx \frac{\sum_{j=1}^{n}(X_{2j} - 10.03)(X_{3j} - 0.53)}{n-1} = 1.964$$

$$\text{Covariance matrix } \mathbf{\Sigma} \approx \begin{bmatrix} 4.01 & 0.516 & 0.138 \\ 0.516 & 7.28 & 1.964 \\ 0.138 & 1.964 & 0.53 \end{bmatrix}$$

LDA with $p \geq 2$ Features

- Observations in class k come from a ***multivariate normal*** distribution with a class-specific mean vector $\boldsymbol{\mu}_k$ and a common variance matrix $\boldsymbol{\Sigma}$.

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$



Class 1 Class 2

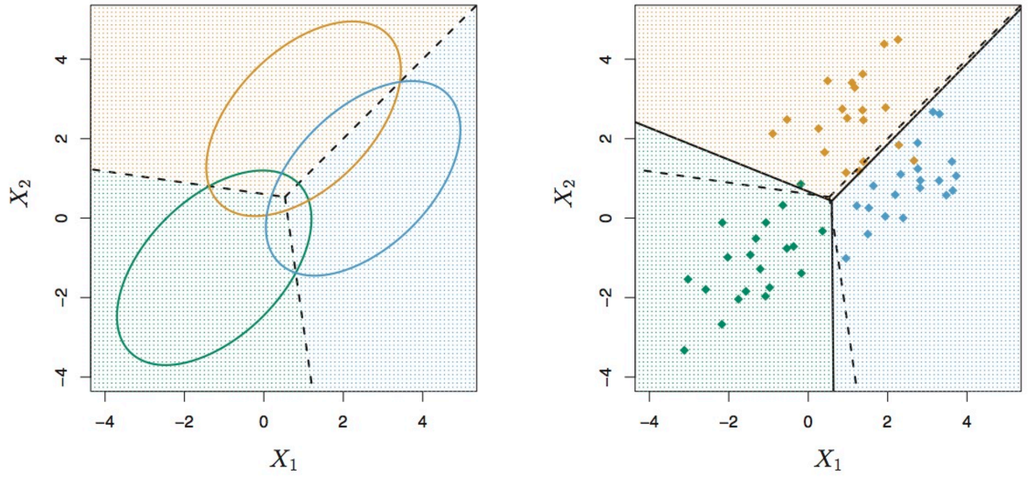
☐ It can be shown that the LDA classifier has the following discrimination function (after removing constant terms):

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

☐ The parameters in $\delta_k(\boldsymbol{x})$ can be estimated from $\hat{\pi}_k = n_k/N, \quad \hat{\boldsymbol{\mu}}_k = (1/n_k) \sum_{i:y_i=k} \boldsymbol{x}_i$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^T$$

☐ Ex: $p = 2, K = 3$



class `sklearn.discriminant_analysis.`**`LinearDiscriminantAnalysis`**(*solver='svd', shrinkage=None, priors=None, n_components=None, store_covariance=False, tol=0.0001, covariance_estimator=None*) [source]

```
#Load financial data
fin_df = pd.read_excel('data/supervised-learning.xlsx',
                       sheet_name='Financial',
                       usecols=['Status','X1','X2','X3'], header=0)



# LDA function from scikit-learn
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

# Prepare the input data
y = fin_df['Status']
X_fin = fin_df.drop(['Status'], axis=1)

# Fit model to data
lda_clf = LDA()
lda_clf.fit(X_fin, y)
```

Status	X1	X2	X3
C1-Solvent	17.4	12.6	1.3
C1-Solvent	54.7	14.6	1.7
C1-Solvent	53.5	20.6	1.1
C1-Solvent	35.9	26.4	2
C1-Solvent	39.4	30.5	1.9
C1-Solvent	53.1	7.1	1.9
C1-Solvent	39.8	13.8	1.2
C1-Solvent	59.5	7	2
C1-Solvent	16.3	20.4	1
C1-Solvent	21.7	-7.8	1.6
C2-Bankrupt	-62.8	-89.5	1.7
C2-Bankrupt	3.3	-3.5	1.1
C2-Bankrupt	-120.8	-103.2	2.5
C2-Bankrupt	-18.1	-28.8	1.1
C2-Bankrupt	-3.8	-50.6	0.9
C2-Bankrupt	-61.2	-56.2	1.7
C2-Bankrupt	-20.3	-17.4	1
C2-Bankrupt	-194.5	-25.8	0.5
C2-Bankrupt	20.8	-4.3	1
C2-Bankrupt	-106.1	-22.9	1.5

```python
#Determine the predicted values and their probabilites
y_pred = lda_clf.predict(X_fin)
y_pred
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0])
```

```python
y_pred_probs = lda_clf.predict_proba(X_fin)
y_pred_probs.round(3)
```

```
array([[0.122, 0.878],
       [0.053, 0.947],
       [0.112, 0.888],
       [0.062, 0.938],
       [0.198, 0.802],
       [0.065, 0.935],
       [0.097, 0.903],
       [0.088, 0.912],
       [0.048, 0.952],
       [0.004, 0.996],
       [0.   , 1.   ],
       [0.037, 0.963],
       [0.194, 0.806],
       [0.06 , 0.94 ],
       [0.32 , 0.68 ],
       [0.014, 0.986],
       [0.008, 0.992],
         ....
       [0.631, 0.369]])
```

```python
#Evaluate model performance on train data
from sklearn.metrics import confusion_matrix, classification_report

print(confusion_matrix(y, y_pred))
print(classification_report(y, y_pred))
```

```
[[32  1]
 [ 5 28]]
              precision    recall  f1-score   support

  C1-Solvent       0.86      0.97      0.91        33
 C2-Bankrupt       0.97      0.85      0.90        33

    accuracy                           0.91        66
   macro avg       0.92      0.91      0.91        66
weighted avg       0.92      0.91      0.91        66
```

Quadratic Discriminant Analysis (QDA)

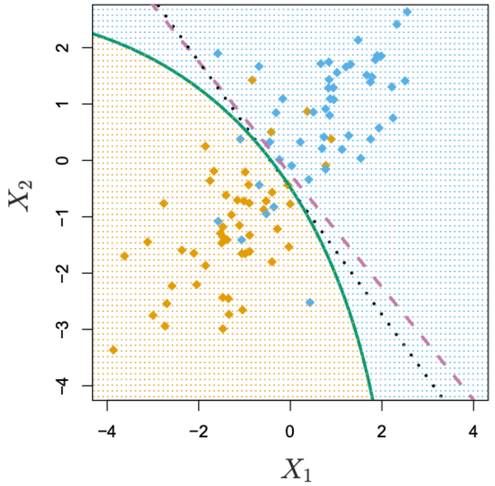
- Assume that observations $\boldsymbol{X}$ from each class are drawn from a Gaussian distribution with its own covariance matrix, or $\boldsymbol{X} \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- The (Bayes) classifier assigns an observation $\boldsymbol{X} = \boldsymbol{x}$ to the class with the largest value of

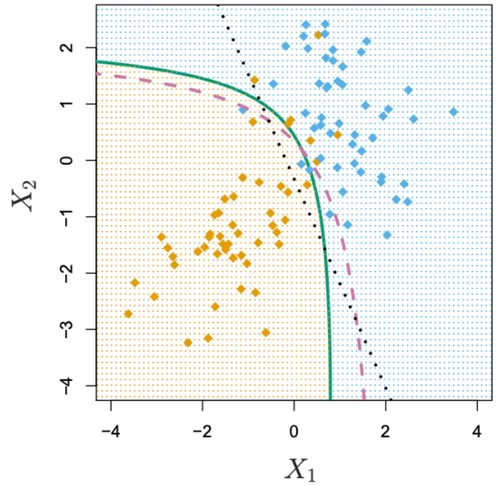
$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\boldsymbol{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\boldsymbol{\Sigma}_k^{-1}x + x^T\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k\end{aligned}$$

Probably use x_1x_2, x_1^2, x_2^2 terms ?

If $\Sigma_1 = \Sigma_2$, the decision boundary should be linear.

$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$





$\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$

Box's M Test for Homogeneity of Variance-Covariance Matrices

$$H0 : \Sigma_i = \Sigma_j \text{ for all i and j}$$
$$H1 : \Sigma_i \neq \Sigma_j \text{ for at least one i and j}$$

pingouin.box_m

pingouin.box_m(*data*, *dvs*, *group*, *alpha=0.001*)

Test equality of covariance matrices using the Box's M test.

Parameters:

data : *pandas.DataFrame*

Long-format dataframe.

dvs : *list*

Dependent variables.

group : *str*

Grouping variable.

alpha : *float*

Significance level. Default is 0.001 as recommended in [2]. A non-significant p-value (higher than alpha) indicates that the covariance matrices are homogenous (= equal).

Returns:

stats : *pandas.DataFrame*

- `'Chi2'` : Test statistic
- `'pval'` : p-value
- `'df'` : The Chi-Square statistic's degree of freedom
- `'equal_cov'` : True if `data` has equal covariance

1. Box M test with 3 dependent variables of 4 groups (equal sample size)

```
>>> import pandas as pd
>>> import pingouin as pg
>>> from scipy.stats import multivariate_normal as mvn
>>> data = pd.DataFrame(mvn.rvs(size=(100, 3), random_state=42),
...                     columns=['A', 'B', 'C'])
>>> data['group'] = [1] * 25 + [2] * 25 + [3] * 25 + [4] * 25
>>> data.head()
          A         B         C  group
0  0.496714 -0.138264  0.647689      1
1  1.523030 -0.234153 -0.234137      1
2  1.579213  0.767435 -0.469474      1
3  0.542560 -0.463418 -0.465730      1
4  0.241962 -1.913280 -1.724918      1
```

```
>>> pg.box_m(data, dvs=['A', 'B', 'C'], group='group')
          Chi2    df      pval  equal_cov
box  11.634185  18.0  0.865537       True
```

2. Box M test with 3 dependent variables of 2 groups (unequal sample size)

```
>>> data = pd.DataFrame(mvn.rvs(size=(30, 2), random_state=42),
...                     columns=['A', 'B'])
>>> data['group'] = [1] * 20 + [2] * 10
>>> pg.box_m(data, dvs=['A', 'B'], group='group')
         Chi2   df      pval  equal_cov
box  0.706709  3.0  0.871625       True
```

```
fin_df = pd.read_excel('data/supervised-learning.xlsx',
                       sheet_name='Financial', usecols=['Status','X1','X2','X3'], header=0)

#Check Covariance Homogeneity
import pingouin as pg

pg.box_m(fin_df, dvs=['X1', 'X2', 'X3'], group='Y')
```

	Chi2	df	pval	equal_cov
box	111.980019	6.0	7.844208e-22	False

```python
#Fit QDA model and evaluation result
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as QDA

y = fin_df['Y']
X = fin_df.drop('Y', axis=1)

qda_clf = QDA()
qda_clf.fit(X, y)

from sklearn.metrics import confusion_matrix, classification_report
y_pred = qda_clf.predict(X)
print(confusion_matrix(y, y_pred))
print(classification_report(y, y_pred))
```

```
[[31  2]
 [ 1 32]]
              precision    recall  f1-score   support

           0       0.97      0.94      0.95        33
           1       0.94      0.97      0.96        33

    accuracy                           0.95        66
   macro avg       0.95      0.95      0.95        66
weighted avg       0.95      0.95      0.95        66
```

Naive Bayes (NB) Classifier

☐ From

$$
\begin{aligned}
p_k(\boldsymbol{x}) \triangleq \mathbb{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\} &= \frac{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\}\mathbb{P}\{Y = k\}}{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x}\}} \\
&= \frac{\mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\}\mathbb{P}\{Y = k\}}{\sum_j \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = j\}\mathbb{P}\{Y = j\}} \\
&= \frac{f_k(\boldsymbol{x})\pi_k}{\sum_j \pi_j f_j(\boldsymbol{x})}
\end{aligned}
$$

☐ Estimating $f_k(\boldsymbol{x})$ can be simplified by assuming conditional independence of p predictors $X_1, X_2, \cdots, X_p$:

$$
f_k(\boldsymbol{x}) = f_{k1}(x_1) f_{k2}(x_2) \cdots f_{kp}(x_p)
$$

☐ Each attribute independently affects the class label.

- Still provide sufficiently good results.
- Hence, the name “Naive Bayes”.

Naive Bayes for Discrete Features

- If X_j is qualitative, we can estimate the class likelihood pmf $f_{kj}(x) = Pr\{X_j = x \,|\, Y = k\}$ and multiply them directly.
- For $\boldsymbol{X} = [X_1, X_2, X_3]$ and $\boldsymbol{x} = [x_1, x_2, x_3]$, the class likelihood pmf for class k is determined by

$$\begin{aligned} f_k(\boldsymbol{x}) &= \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\} \\ &= \mathbb{P}\{X_1 = x_1 \mid Y = k\}\,\mathbb{P}\{X_2 = x_2 \mid Y = k\}\,\mathbb{P}\{X_3 = x_3 \mid Y = k\} \end{aligned}$$

$$\mathbb{P}\{X_j = x_j \mid Y = k\} = \frac{\text{Count}(X_j = x_j, Y = k)}{\text{Count}(Y = k)}$$

- The discriminant score $\delta_k(\boldsymbol{x})$ and predicted class c are then determined by

$$\delta_k(\boldsymbol{x}) = f_k(\boldsymbol{x})\pi_k = \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x} \mid Y = k\}\mathbb{P}\{Y = k\}$$

$$c = \arg\max_k \delta_k(\boldsymbol{x})$$

Example

- Consider the Fraud dataset with three features Credit History (X_1), Support (X_2), Accommodation (X_3).
- What is $f_k(\boldsymbol{x})$ when $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{none}, x_3 = \text{rent}\}$?

ID	CreditHistory	Support	Accommodation	Fraud
1	current	none	own	yes
2	paid	none	own	no
3	paid	none	own	no
4	paid	guarantor	rent	yes
5	arrears	none	own	no
6	arrears	none	own	yes
7	current	none	own	no
8	arrears	none	own	no
9	current	none	rent	no
10	none	none	own	yes
11	current	coapplicant	own	no
12	current	none	own	yes
13	current	none	rent	yes
14	paid	none	own	no
15	arrears	none	own	no
16	current	none	own	no
17	arrears	coapplicant	rent	no
18	arrears	none	free	no
19	arrears	none	own	no
20	paid	none	own	no

Data from sheet 'Fraud' in supervised-learning.xlsx

☐ Consider the fraud status = 'Yes', $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{none}, x_3 = \text{rent}\}$

$$f_{credit}(x_1) = Pr\{X_1 = paid \,|\, Y = yes\} =$$
$$f_{support}(x_2) = Pr\{X_2 = none \,|\, Y = yes\} =$$
$$f_{accmd}(x_3) = Pr\{X_3 = rent \,|\, Y = yes\} =$$
$$f_{yes}(\boldsymbol{x}) =$$
$$\pi_{yes} =$$
$$P\{\text{Fraud} = \text{Yes} \,|\, \boldsymbol{x}\} =$$

☐ Consider the fraud status = 'No', $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{none}, x_3 = \text{rent}\}$

$$f_{credit}(x_1) = Pr\{X_1 = paid \,|\, Y = no\}$$
$$f_{support}(x_2) = Pr\{X_2 = none \,|\, Y = no\}$$
$$f_{accmd}(x_3) = Pr\{X_3 = rent \,|\, Y = no\}$$
$$f_{no}(\boldsymbol{x}) =$$
$$\pi_{no} =$$
$$P\{\text{Fraud} = \text{No} \,|\, \boldsymbol{x}\} =$$

	X_1	X_2	X_3	Y
ID	CreditHistory	Support	Accommodation	Fraud
1	current	none	own	yes
2	paid	none	own	no
3	paid	none	own	no
4	paid	guarantor	rent	yes
5	arrears	none	own	no
6	arrears	none	own	yes
7	current	none	own	no
8	arrears	none	own	no
9	current	none	rent	no
10	none	none	own	yes
11	current	coapplicant	own	no
12	current	none	own	yes
13	current	none	rent	yes
14	paid	none	own	no
15	arrears	none	own	no
16	current	none	own	no
17	arrears	coapplicant	rent	no
18	arrears	none	free	no
19	arrears	none	own	no
20	paid	none	own	no

$$|X_1| = 4, |X_2| = 3, |X_3| = 3$$

Solution

☐ Consider the fraud status = 'Yes', $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{none}, x_3 = \text{rent}\}$

$$f_1(x) = Pr\{X_1 = paid \,|\, Y = yes\} = 1/6$$
$$f_2(x) = Pr\{X_2 = none \,|\, Y = yes\} = 5/6$$
$$f_3(x) = Pr\{X_3 = rent \,|\, Y = yes\} = 2/6$$
$$\pi_1 = 6/20 = 0.3$$

ID	CreditHistory	Support	Accommodation	Fraud
1	current	none	own	yes
2	paid	none	own	no
3	paid	none	own	no
4	paid	guarantor	rent	yes
5	arrears	none	own	no
6	arrears	none	own	yes
7	current	none	own	no
8	arrears	none	own	no
9	current	none	rent	no
10	none	none	own	yes
11	current	coapplicant	own	no
12	current	none	own	yes
13	current	none	rent	yes
14	paid	none	own	no
15	arrears	none	own	no
16	current	none	own	no
17	arrears	coapplicant	rent	no
18	arrears	none	free	no
19	arrears	none	own	no
20	paid	none	own	no

☐ Consider the fraud status = 'No', $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{none}, x_3 = \text{rent}\}$

$$f_1(x) = Pr\{X_1 = paid \,|\, Y = no\} = 4/14$$
$$f_2(x) = Pr\{X_2 = none \,|\, Y = no\} = 12/14$$
$$f_3(x) = Pr\{X_3 = rent \,|\, Y = no\} = 2/14$$
$$\pi_2 = 14/20 = 0.7$$

$$|X_1| = 4, |X_2| = 3, |X_3| = 3$$

Laplace Smoothing

For the fraud status = 'Yes' and $\boldsymbol{x} = \{x_1 = \text{paid}, x_2 = \text{coapplicant}, x_3 = \text{rent}\}$, what is $f(\boldsymbol{x})$?

Feature values that do not or rarely occur in a certain class can distort or render the class likelihood incalculable.

For a feature X_j with $|X_j|$ levels, its class likelihood function is modified by using the ***smoothing parameter*** α as

$$\mathbb{P}\{X_j = x_j \mid Y = k\} = \frac{\text{Count}(X_j = x_j, Y = k) + \alpha}{\text{Count}(Y = k) + \alpha|X_j|}, \quad \alpha > 0$$

NB Classifier using Scikit Learn

- Each feature assumes its own discrete distribution whose class-conditional probability can be estimated separately.
- Feature values must be encoded to numbers by using ordinal or label encoders.

```
class sklearn.naive_bayes.CategoricalNB(*, alpha=1.0, force_alpha=True,
fit_prior=True, class_prior=None, min_categories=None)                [source]
```

Naive Bayes for Numeric Features

- Assume an arbitrary distribution whose parameters conveniently estimated (e.g., normal, student-t, exponential/gamma/beta, mix of normals) or kernel density estimation.
- For example, if all variables have $f_{kj}(x) \sim \mathrm{N}(\mu_k, \sigma_k^2)$, the discriminant function becomes

$$\begin{aligned}\delta_k(\boldsymbol{x}) &= \log f_k(\boldsymbol{x}) + \log \pi_k \\ &= \sum_{i=1}^{p} \log f_{ki}(x_i) + \log \pi_k \\ &= \sum_{i=1}^{p} \left(-\frac{x_i^2}{2\sigma_{ik}^2} + x_i \frac{\mu_{ik}}{\sigma_{ik}^2} - \frac{\mu_{ik}^2}{2\sigma_{ik}^2} - \log(\sqrt{2\pi}\sigma_{ik}) \right) + \log \pi_k\end{aligned}$$

  - Known as ***Gaussian Naive Bayes (GNB)***
  - Explore individual features to confirm the assumption

class sklearn.naive_bayes.GaussianNB(, priors=None, var_smoothing=1e-09)*

- Alternatively, use feature (equal-frequency) binning and proceed with Categorical NB.

Naive Bayes for Mixed Numeric and Categorical Features

- ☐ Feature preprocessing
  - Correlated features should be removed.
  - Use chi-square test to check dependence of categorical features.
- ☐ Assume or fit pdf to numeric features
  - EDA should help identifying candidate distributions.
  - Values of discrete pmf, e.g., Poisson, can be plugged directly.
- ☐ For continuous pdf, compute the probability over a small range.
  - Ex: For temperature 70C, compute the probability over [69.5, 70.5].
  - All classes must use the same range.

Your Turn: Default Data set with LDA and QDA Classifiers

- ☐ From the credit default data set, fit LDA and QDA classifiers on the whole data set. Then, evaluate the accuracy, precision, recall, and F1-score.
- ☐ Is one of the models significantly better than the other ?
- ☐ Explain how the precision and recall change if the decision threshold is reduced.

Data from sheet 'Default' in supervised-learning.xlsx