NYC DATA SCIENCE ACADEMY

# Python Machine Learning

## Discriminant Analysis & Naive Bayes

NYC Data Science Academy

# Outline

- ❖ **Discriminant Analysis: Motivation**
  - ➢ Conditional Probability and Bayes Theorem
- ❖ **Discriminant Analysis: Models**
  - ➢ **One Dimensional Cases**
  - ➢ **Higher Dimensional Cases**
- ❖ **Naive Bayes**

## Conditional Probability

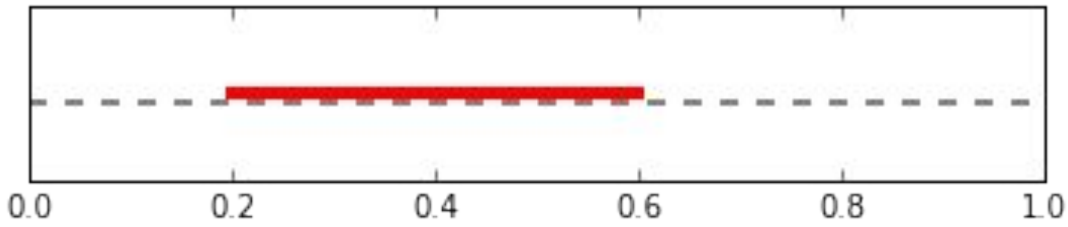
❖ Let *Y* be an event with probability *P*(*Y*) > 0, the ***conditional probability*** of observing *X* given that *Y* has occurred is defined as:

$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

- ➢ *P*(*X*, *Y*) refers to the **joint probability** that both *X* and *Y* occur.
- ➢ *P*(*X*| *Y*) is the probability of *X* after insuring *Y*'s occurrence.
- ➢ P(X) may be different from P(X|Y)

## Conditional Probability

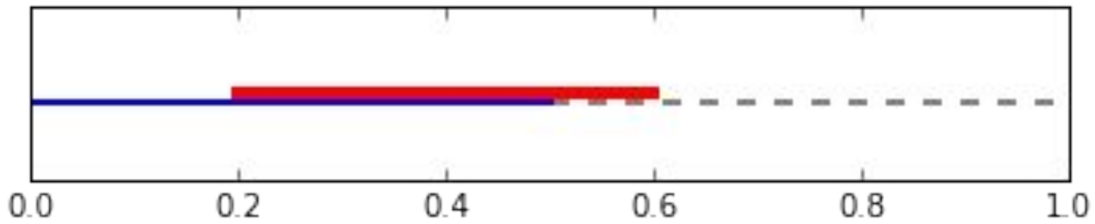
- Suppose that we draw a random number uniformly distributed in the unit interval [0,1]
- What is the probability of drawing a number in the red region?




$$0.4 \div 1 = 0.4$$

# Conditional Probability

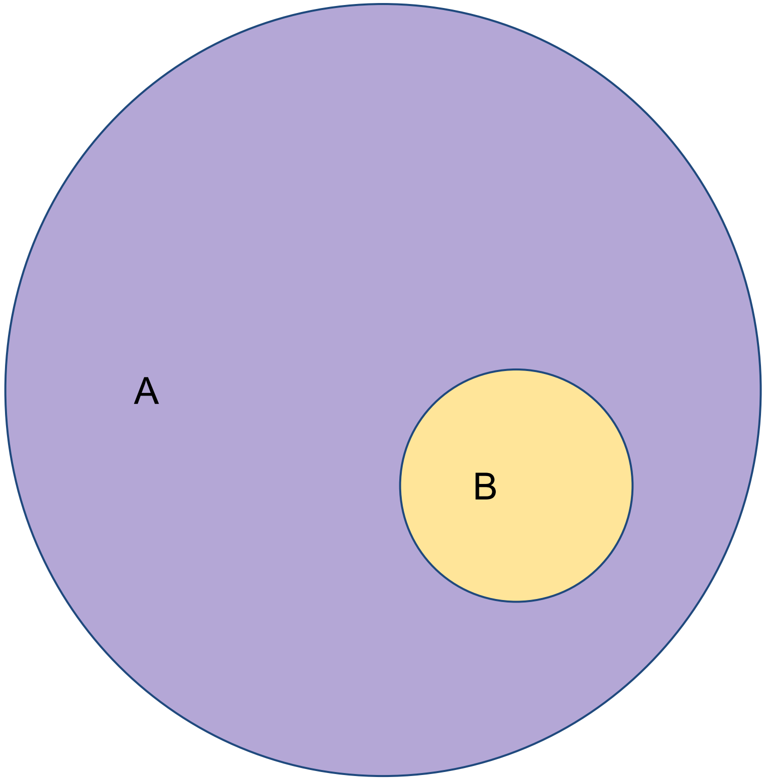
- How about the probability of obtaining a number in the red region when restricted in the blue region?




$$\frac{P(\text{red, blue})}{P(\text{blue})} = \frac{0.3}{0.5} = 0.6$$

## Venn Diagram

The set universe




P(A|B) = 1, but P(A) < 1
In this example P(A|B) != P(A)

## Independent Events

- If $X$ and $Y$ are independent, $P(X, Y) = P(X)P(Y)$:
  - Then the conditional probability is

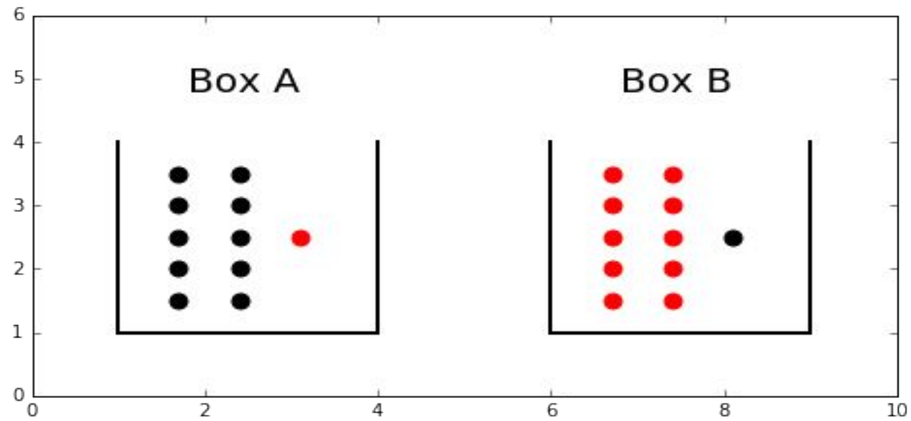
$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{P(X)P(Y)}{P(Y)} = P(X)$$

  - This implies that the occurrence of $Y$ does not have any impact on the occurrence of $X$.

## Conditional Probability Example

Consider an experiment of picking balls of two colors, red and black, from two boxes labeled A and B.

1. There are 10 black balls and 1 red ball in box A, and 1 black ball and 10 red balls in box B.
2. We randomly choose a box (with equal chance) and then pick a ball randomly from it.
3. What is the probability that we draw a red ball finally?



## Conditional Probability Example

When choosing a box to pick, we have:

1. $P(A) = P(B) = 0.5$.
2. If we choose $A$, $P(red|A) = 1/11$.
3. If we choose $B$, $P(red|B) = 10/11$.

So the probability to get one red ball from either box A or box B is:

$$\begin{aligned} P(red) &= P(red|A) \cdot P(A) + P(red|B) \cdot P(B) \\ &= \frac{1}{11} \times 0.5 + \frac{10}{11} \times 0.5 \\ &= \frac{1}{2} \end{aligned}$$

❖ Is there a more intuitive way to figure this out?

## Bayes Theorem

- **Bayes theorem** is named after **Thomas Bayes**.
- It describes the probability of an event, based on conditions that might be related to the event.
- **Bayes theorem** states (assuming Y is of discrete valued):

$$Pr(Y|X) = \frac{Pr(X|Y) \cdot Pr(Y)}{Pr(X)} = \frac{Pr(X|Y) \cdot Pr(Y)}{\sum_l Pr(X|Y=l) \cdot Pr(Y=l)}$$

- It allows us to **swap** the conditional probability Pr(Y|X) into Pr(X|Y), up to the rescaling factor--the prior probabilities ratio Pr(Y)/Pr(X).

# Thomas Bayes (1701-1761)

## Bayes Theorem Example

- Consider the same experiment of picking colored balls from two boxes.
- If the ball we picked is red, then what is the probability that the ball was from box *A*?

According to Bayes' theorem, we have:

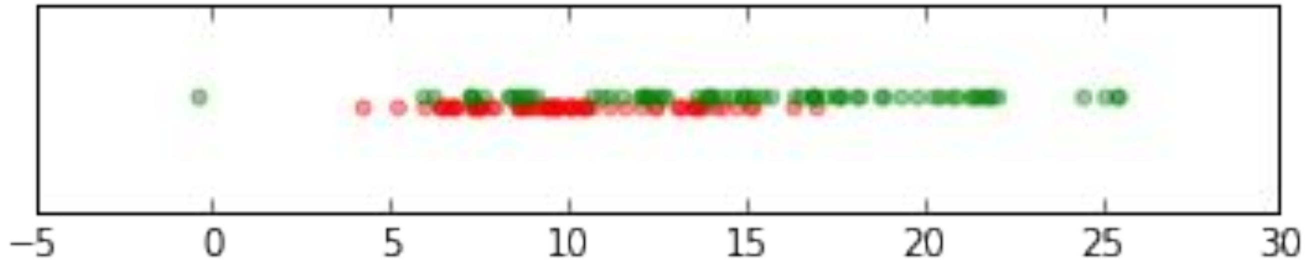
$$\begin{aligned} P(A|red) &= \frac{P(red|A) \cdot P(A)}{P(red)} \\ &= \frac{P(red|A) \cdot P(A)}{P(red|A)P(A) + P(red|B)P(B)} \\ &= \frac{\frac{1}{11} \times 0.5}{(\frac{1}{11} \times 0.5 + \frac{10}{11} \times 0.5)} \\ &= \frac{1}{11} \end{aligned}$$

## Bayes Theorem Example

- How does this relate to our classification problem? Consider from the train set we realize that for a red ball:
  - the probability that the red ball was from box A is 1/11
  - and the probability that the red ball was from box B is 10/11
- Next time if we get a red ball, shouldn't we be more certain that the ball was from box B?

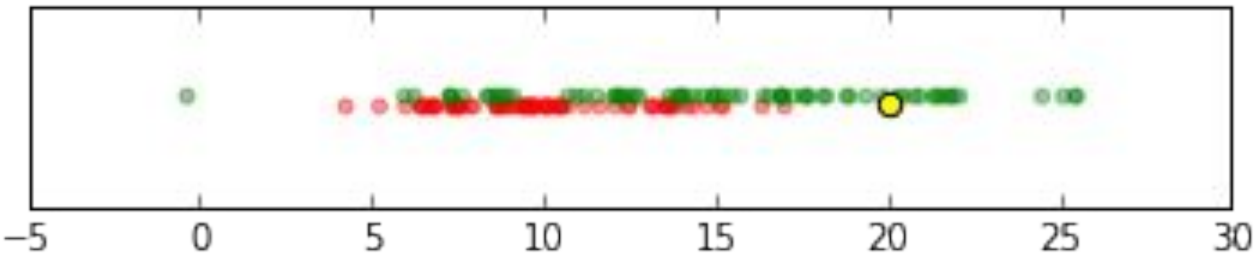
## Discriminant Analysis

- *Discriminant analysis* is a statistical analysis technique which classifies based on hypothesizing the per class conditional probability distribution to be normal and pinning down these parameters by data fitting.
- **Motivation**: To be more precise, let's consider binary classification based on a numerical feature with a simulated data.



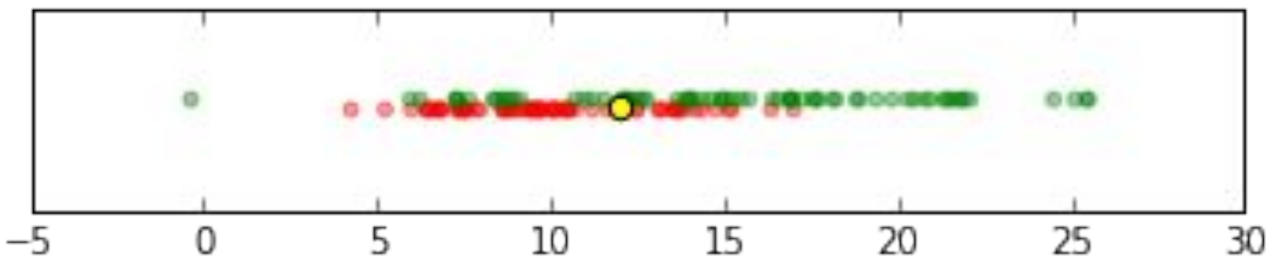
# Discriminant Analysis

- If we add a new observation, which class do you think it belongs to?



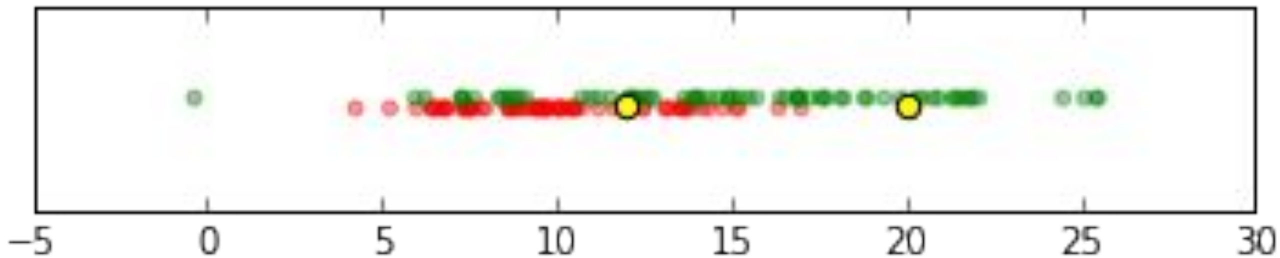
# Discriminant Analysis

- What about this one?



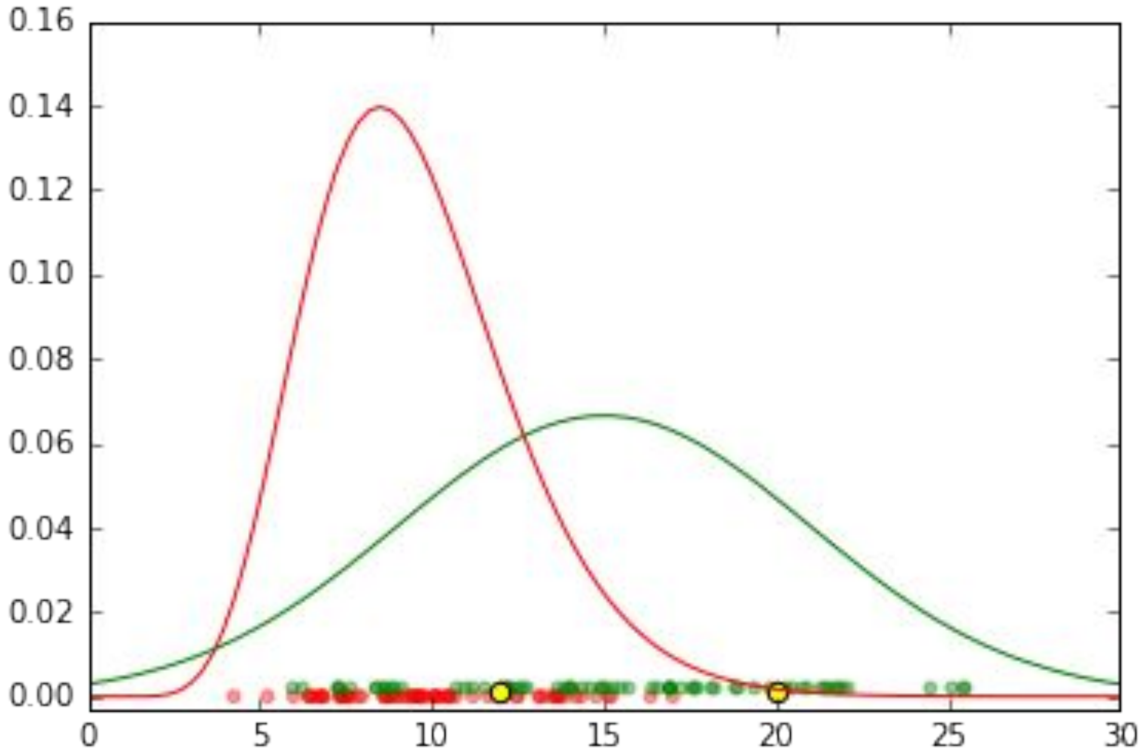
## Discriminant Analysis

- What makes us feel differently?
  - If there is some other information tacitly guiding us to the conclusion, can we somehow name it? or visualize it?



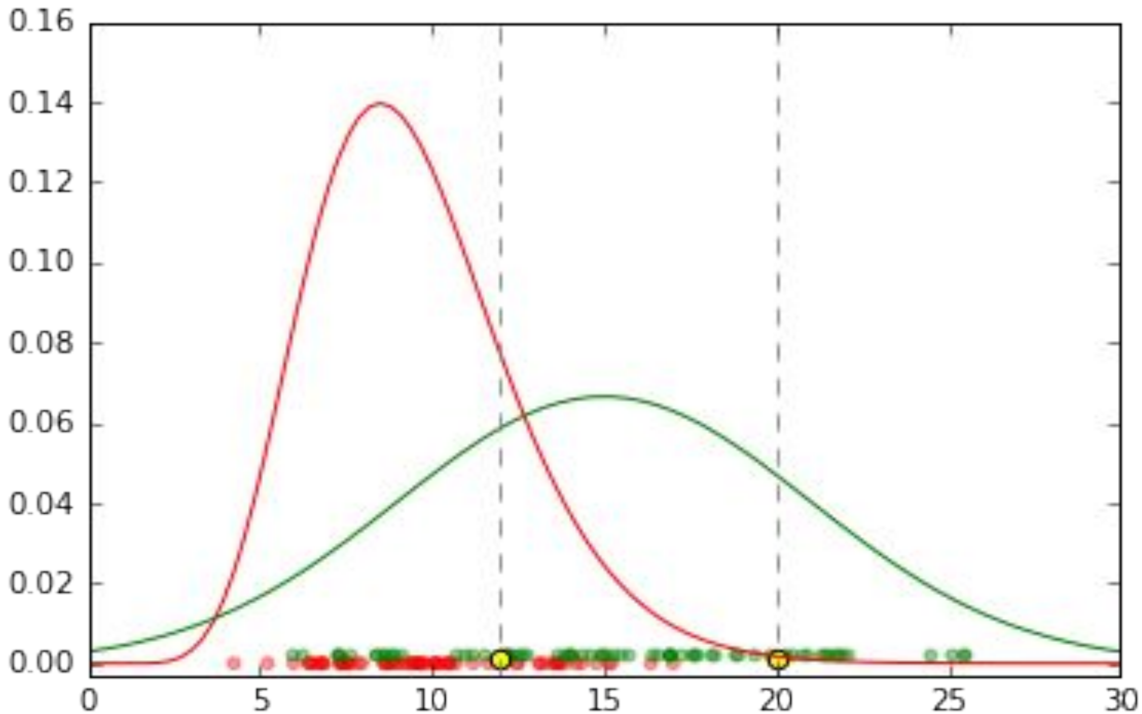
## Discriminant Analysis

- How about density plot for each class?



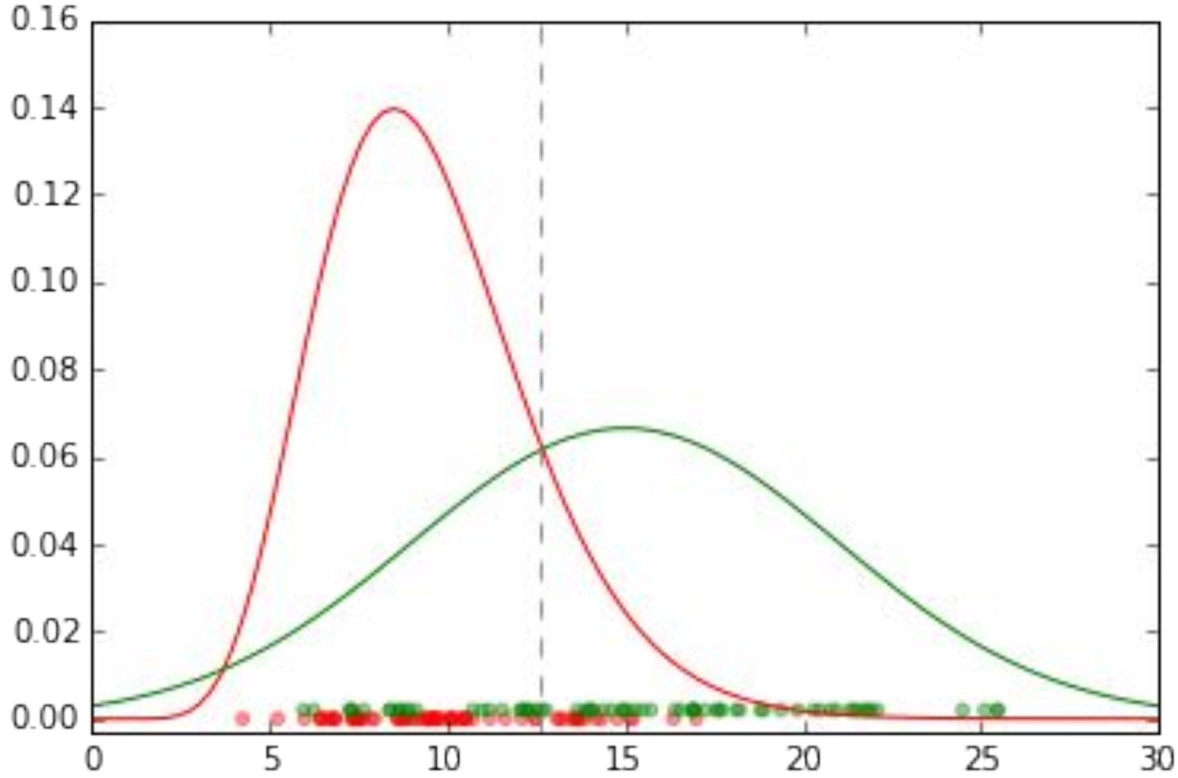
## Discriminant Analysis

- What happens to the density plots at the two yellow observations?
- Both density curves are of different widths/heights, due to different standard deviations.



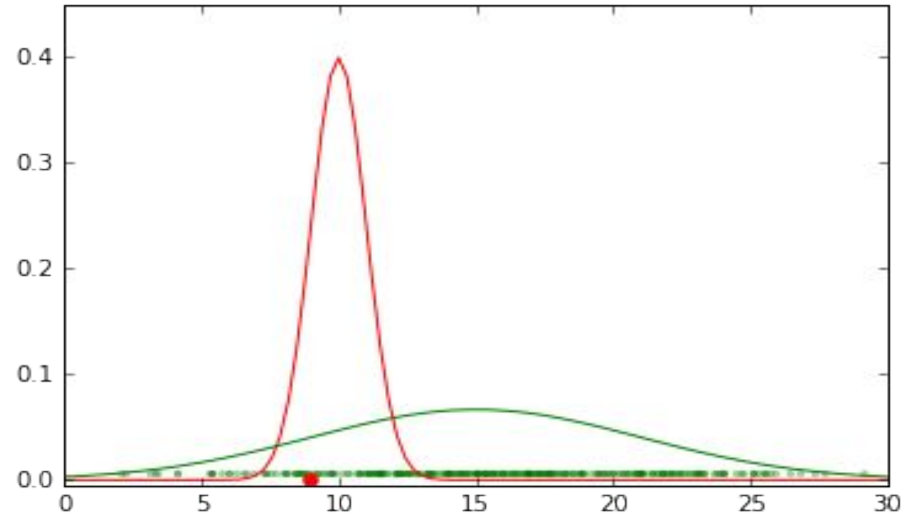
# Bayes Classifier

- So is this how we classify?



## Bayes Classifier

- **Caution:**
  - To emphasize the effect of the density within each class, we intentionally created two classes with the same size. When the sizes are different, **missing the prior probability would cause a big trouble, especially when the class labels are unbalanced**.
  - Below is an extreme case:



## Discriminant Analysis

- Note that the final goal of classification is to determine

$$P(Y = k \mid X = x) \text{ for each class k}$$

- But we just found that

$$p(X = x \mid Y = k) \text{ for each class k}$$

Is helpful! How do we relate the two types of conditional probabilities?

## Discriminant Analysis and Bayes Theorem

- Bayes theorem comes into play because we want to relate the two conditional probabilities above.

$$P(Y = k \mid X = x) = \frac{p(X = x \mid Y = k)P(Y = k)}{\sum_l p(X = x \mid Y = l)P(Y = l)}$$

- **Questions:**
  - How do we model $P(Y = k)$ (this is called the prior probability for class k)?
  - How do we model $p(X = x \mid Y = k)$?

## Discriminant Analysis and Bayes Theorem

- **Answers:**

➢ $P(Y = k)$ can be estimated by $\frac{n_k}{n}$, the sample class probability.

Where:

- $n_k$ = the number of observations in class *k*.
- *n* = the total count of observations.

➢ Modeling $p(X = x \mid Y = k)$ is nontrivial. Different models result in different classifiers as we will see.

## Bayes Classifier

- Now that we can estimate the probability of belonging to each class, we can then assign the observation to the class with the highest probability.
  - This is known as Bayes classifier. It minimises the probability of misclassification (the probability of false positive and false negative for binary classification).
  - The boundary of classification (decision boundary) in the feature space is simply where the probabilities of different classes happen to be the same.
  - This rule works when the different classes are more or less 'balanced'.
- For unbalanced classes, the classical **Bayesian decision theory** allows us to handle the scenario when the minority class is of particular interest.

# Outline

- Discriminant Analysis: Motivation
- **Discriminant Analysis: Models**
  - One Dimensional Cases
  - Higher Dimensional Cases
- Naive Bayes

## Discriminant Analysis: Models

- To build a Bayes classifier, the only thing we miss is, for all k,

$$p(X = x \mid Y = k)$$

- The Gaussian distribution is widely used to model it, due to its tractability. Different kinds of Gaussian distribution result in different kind of classifiers. The following three are most common:
  - **Linear Discriminant Analysis** (LDA)
  - **Quadratic Discriminant Analysis** (QDA)
  - **Gaussian Naive Bayes** (This is the same as QDA in a one dimensional case applying to a product space)

## Remark:

- For empirical data which is multimodal, there exists extensions of Gaussian DA to mixture of Gaussian discriminant analysis, MDA.

  The discussion of **MDA** is beyond our scope.

- There exists Kernel based **DA** extending **LDA** using kernel trick.
- Discriminant analysis (supervised learning) and Kmeans clustering (unsupervised learning) are intimately related to each other. There exists hybrid model combining discriminant analysis with cluster analysis--called discriminative cluster analysis.
- Multiclass discriminant analysis can be used as a dimensional reduction technique.

# Outline

- Discriminant Analysis: Motivation
- Discriminant Analysis: Models
  - **One Dimensional Cases**
  - Higher Dimensional Cases
- Naive Bayes

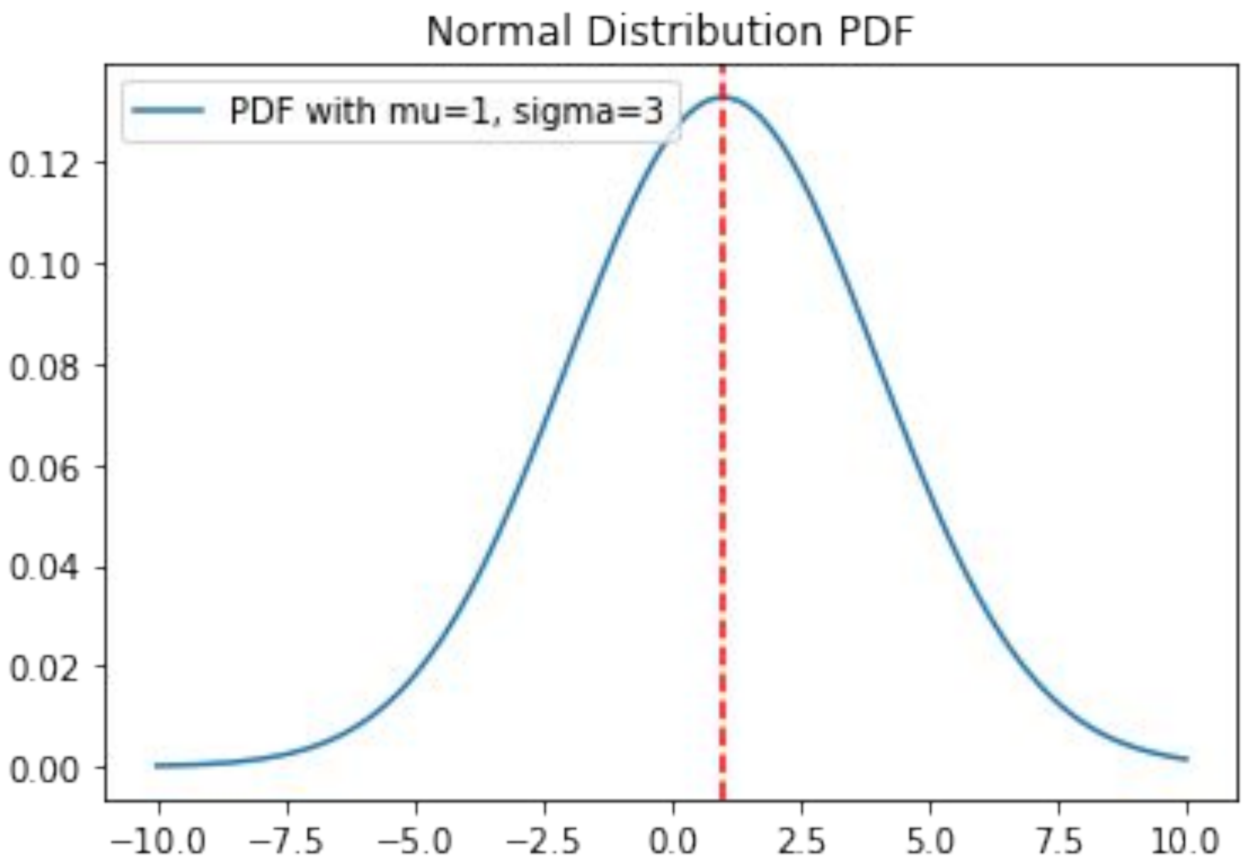
## One Dimensional Cases

- When we have only one feature, we use one dimensional Gaussian distribution pdf (probability density function).

$$N(\mu,\sigma)(x) = \frac{1}{\sqrt{2\pi\sigma}} exp[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2]$$

  - Note that it is sufficient to specify the mean and the standard deviation to specify a Gaussian distribution.

# Normal Distribution’s Probability Density Function Example



## One Dimensional Cases

- We always allow **different means** among different classes, but....

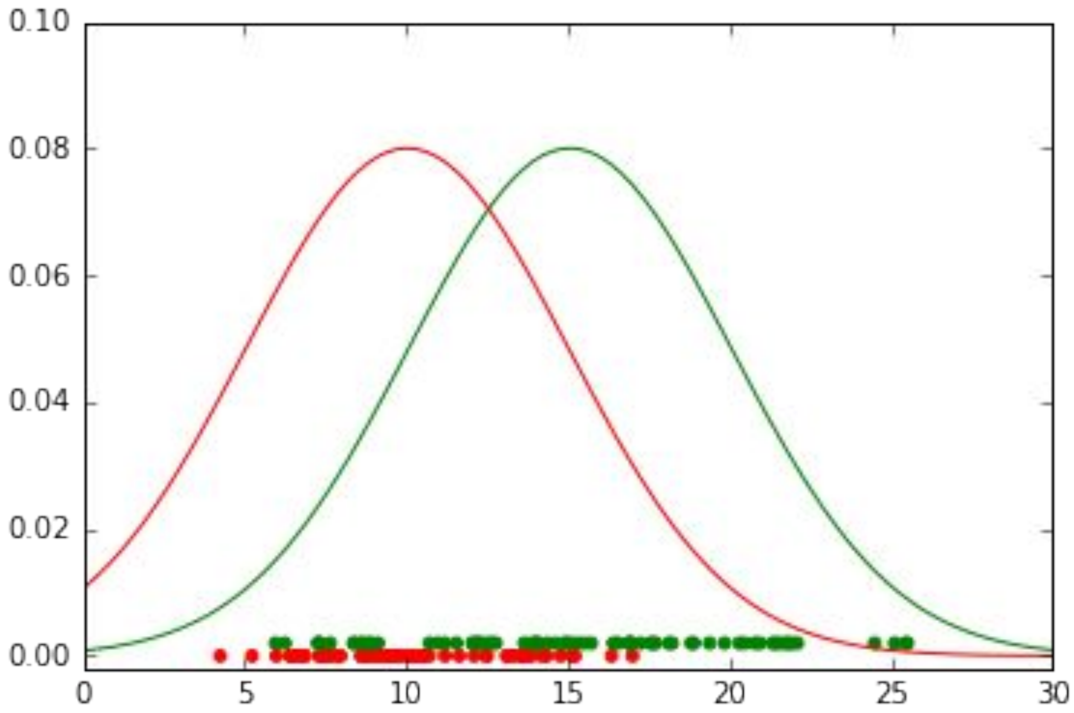
## Linear Discriminant Analysis

- For LDA, we assume that the standard deviation/variance is the same for every class. In one dimensional case, this means that the distribution density function for each class k is:

$$p(X = x \mid Y = k) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[-\frac{1}{2}\left(\frac{x - \mu_k}{\sigma}\right)^2\right]$$

# Linear Discriminant Analysis

- With visualization, this means the width of the normal distribution for every class is unchanged.



# Linear Discriminant Analysis

- **Question:** Now we assume that with **LDA** the distribution for each class k is:

$$p(X = x \mid Y = k) = \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2}(\frac{x - \mu_k}{\sigma})^2]$$

  - How do we decide $\mu_k$ and $\sigma$?

## Linear Discriminant Analysis

- **Answer (by maximal likelihood estimation):**

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i;y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i;y_i=k} (x_i - \hat{\mu}_k)^2$$

$$= \sum_{k=1}^{K} \frac{n_k - 1}{n - K} \cdot \hat{\sigma}_k^2$$

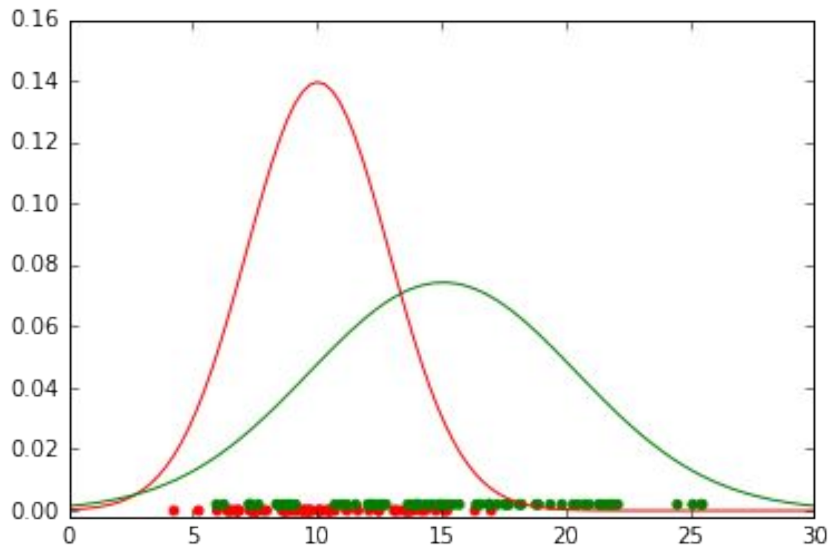
where

- K is the total number of classes.
- $\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{i;y_i=k} (x_i - \hat{\mu}_k)^2$ is the sample variance of class k.

# Quadratic Discriminant Analysis

- For **QDA**, the standard deviation can vary among the classes. In one dimensional case, this means the width of the distribution for every class can be different. Therefore:

$$p(X = x \mid Y = k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp\left[-\frac{1}{2}\left(\frac{x - \mu_k}{\sigma_k}\right)^2\right]$$



## Quadratic Discriminant Analysis

- **Question:** Now we assume that with **QDA** the distribution for each class k is:

$$p(X = x \mid Y = k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp[-\frac{1}{2}(\frac{x - \mu_k}{\sigma_k})^2]$$

- How do we estimate $\hat{\mu}_k$ and $\hat{\sigma}_k$?
- The maximal likelihood estimation on each "cluster" of data points

$\hat{\mu}_k$ , $\hat{\sigma}_k$ is the same as before.

# Outline

- Discriminant Analysis: Motivation
- Discriminant Analysis: Models
  - One Dimensional Cases
  - **Higher Dimensional Cases**
- Naive Bayes

# Higher Dimensional Cases

- We start with the discussion on **higher dimensional** Gaussian distribution. This is essentially the only difference in higher dimensional discriminant analysis.

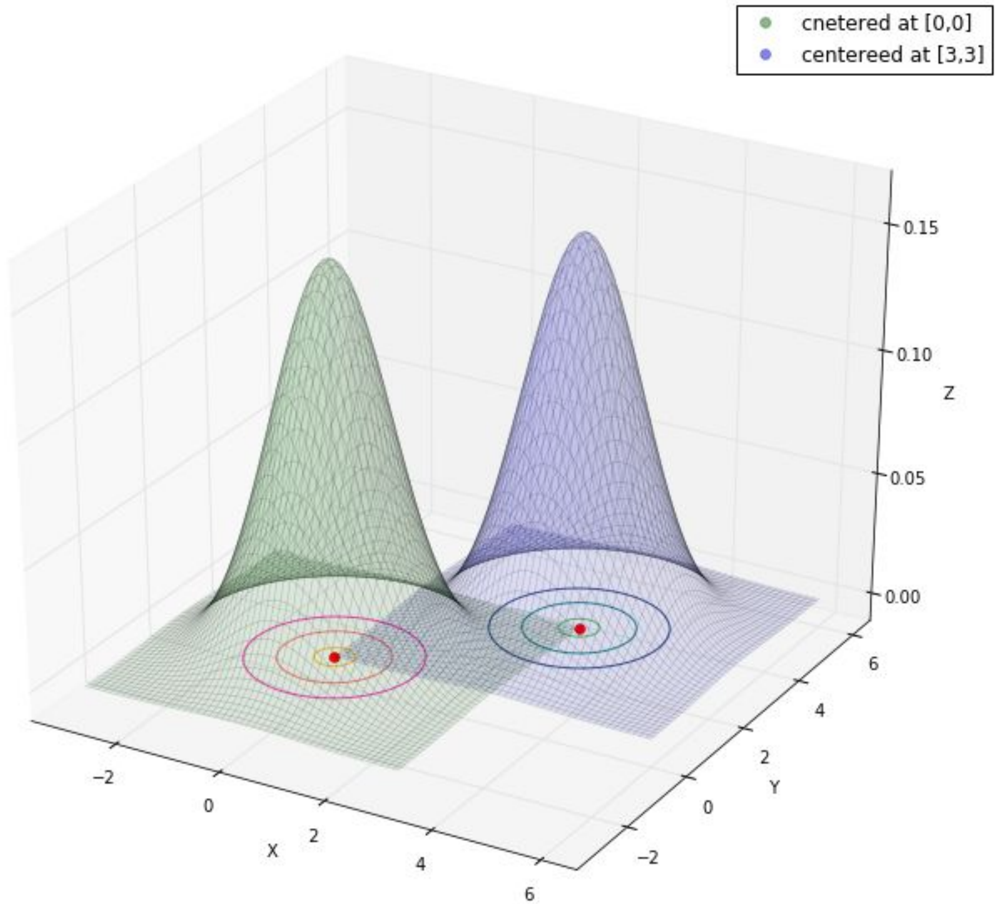
# Higher Dimensional Gaussian Distribution

- We still need only "two" sets of parameters to specify higher dimensional Gaussian distribution: the mean and the covariance. However, for a p dimensional case (with p features):
  - the mean is a p-dimensional vector.
  - the covariance is a $p \times p$ **symmetric** matrix.
- The distribution becomes:

$$N(\mu, \Sigma)(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

# Higher Dimensional Gaussian Distribution

- **mean**: The mean still decides the location where the "bell" is centered at.

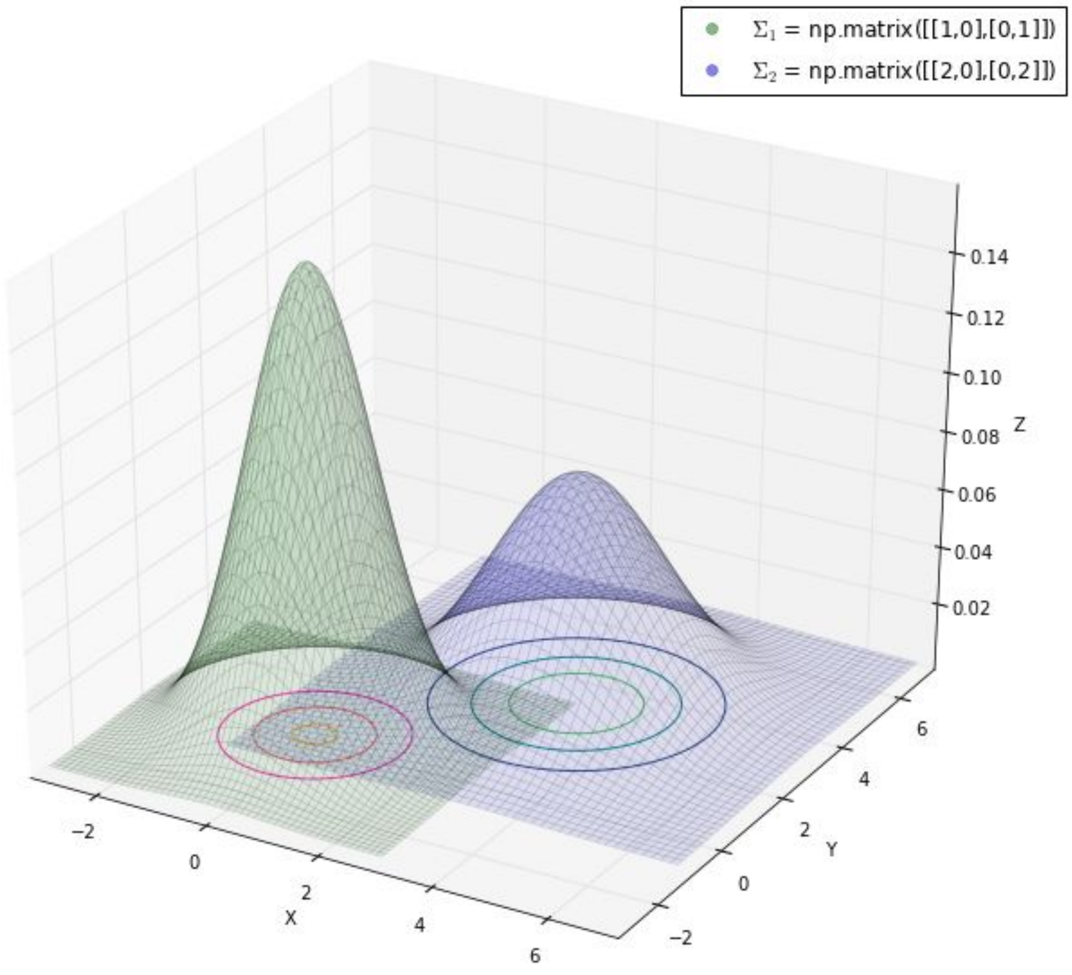


## Higher Dimensional Gaussian Distribution

- **Covariance Matrix**: The covariance matrix is a $p \times p$ **symmetric** matrix. The covariance matrix, one of whose special cases is the square of standard deviation in one dimensional space, decides the shape of the "bell". However, the shape of a high dimensional object means more than just the width.
- **Width**: Let's compare two Gaussian distributions with different covariance matrices in a two dimensional space.

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

# Higher Dimensional Gaussian Distribution

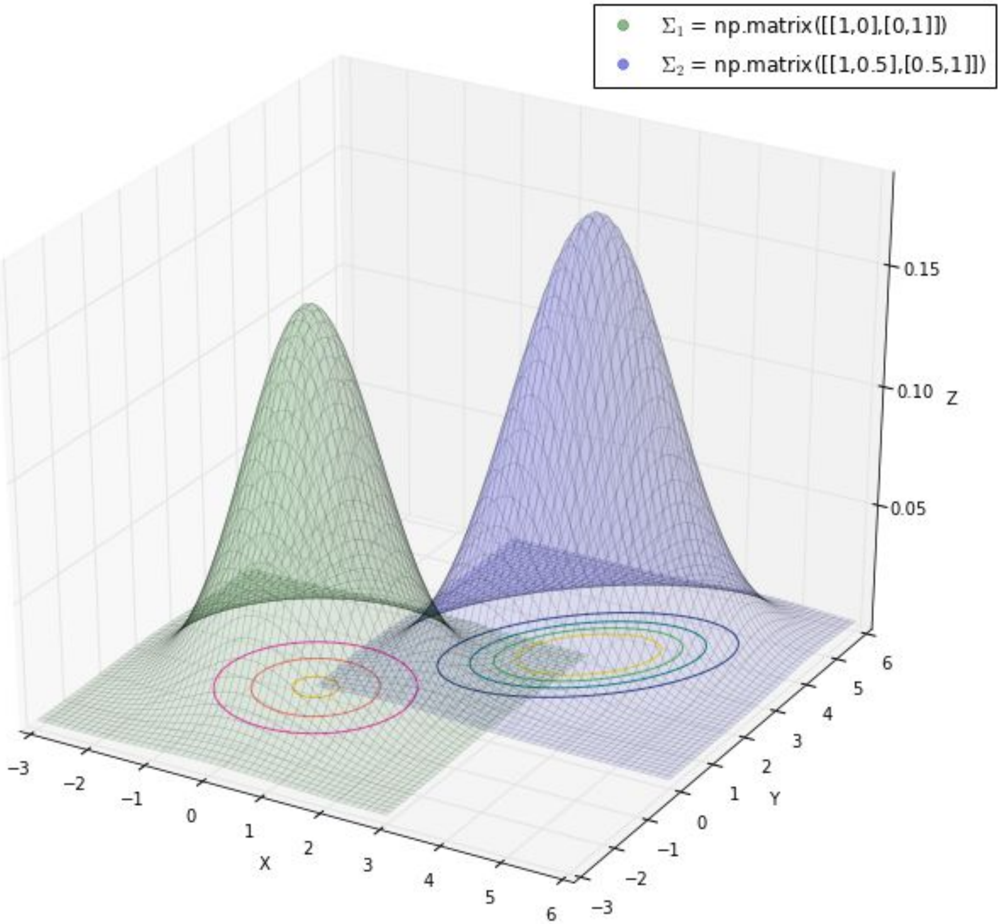


# Higher Dimensional Gaussian Distribution

- **Correlation**: Let's compare two Gaussian distributions with different covariance matrices in a two dimensional space.

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

# Higher Dimensional Gaussian Distribution



# Models in Higher Dimension

- So we only need to decide:
  - prior probability.
  - probability distribution of the features in each class.
- Since the prior probabilities are always estimated in the same way, the difference among **LDA**, **QDA** and **GNB** are stemmed from the different assumptions on the Gaussian distribution.

# Models in Higher Dimension: LDA

- **LDA** assumes the identical covariance matrix across all the classes. In the formula, we see that the mean depends on k, but the covariance matrix does not.

$$\mathrm{P}(\mathrm{X} = \mathrm{x}|Y = k) = \frac{1}{(2\pi)^{\frac{p}{2}} \cdot det(\Sigma)} exp[-(x - \mu_k)^T \Sigma^{-1} (x - \mu_k)]$$

# Models in Higher Dimension: QDA

- **QDA** allows different covariance matrices for different classes. In the formula, we see that the covariance matrix now depends on k as well.

$$p(X = x \mid Y = k) = \frac{1}{(2\pi)|\Sigma_k|^{1/2}} exp\Big[-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\Big]$$

## Models in Higher Dimension: GNB

- **GNB** also allows different covariance matrices for different classes.

$$p(X = x \mid Y = k) = \frac{1}{(2\pi)|\Sigma_k|^{1/2}} exp\left[-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right]$$

- The difference from **QDA** is that **GNB** assumes **no conditional correlation** among the features, so

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_p^2 \end{bmatrix}$$

## Models in Higher Dimension: GNB

- The assumption of zero correlation simplifies the conditional distribution. Within each class, the multivariate normal distribution can be written as the product of univariate normal distributions.

$$\prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma_j}} exp\left[-\frac{1}{2}\left(\frac{x_j - \mu_j}{\sigma_j}\right)^2\right]$$

  - Here each j indicates a feature subscript.

# Hands-on Session

- Please go to the **"Discriminant Analysis in Scikit-Learn"** in the lecture code.

# Outline

- ❖ Discriminant Analysis: Motivation
- ❖ Discriminant Analysis: Models
  - ➢ One Dimensional Cases
  - ➢ Higher Dimensional Cases
- ❖ **Naive Bayes Models**

## Naive Bayes

- Recall that Bayes theorem asserts the probability that the output is in class *k*, given *X* = *x*, can be estimated by the following form:

$$P(Y = k \mid X = x) = \frac{p(X = x \mid Y = k)P(Y = k)}{\sum_l p(X = x \mid Y = l)P(Y = l)}$$

- **LDA** and **QDA** use multivariate Gaussian densities (but with different assumptions on the covariance matrices). These do not work well when the number of features is large compared to the sample size (why?).
- ***Naive Bayes*** models make a simplifying assumption that the features are conditionally independent within each class so it works with dataset of a large number of features.

# Naive Bayes

- The naive Bayes classifier is based on Bayes theorem with conditional independence assumptions between predictors.
- The assumption of conditional independence requires the factorization:

$$P(X = x | Y = k) \equiv f_k(x), \qquad f_k(x) = \prod_{j=1}^{p} f_{jk}(x)$$

where $f_{jk}(x)$ is the probability density for the $j^{th}$ feature $X_j$ in class $k$.

- We will introduce three kinds of Naive Bayesian models:
  - Gaussian Naive Bayes
  - Multinomial Naive Bayes
  - Bernoulli Naive Bayes

# Gaussian Naive Bayes

- Gaussian Naive Bayes assumes each feature follows a gaussian distribution ($\boldsymbol{\Sigma}_k$ is diagonal):

$$f_{jk}(x) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} exp\left[ - \frac{(x_j - \mu_{jk})^2}{2\sigma_{jk}^2} \right]$$

where:

  - $\mu_{jk}$: the mean of the $j^{th}$ feature $X_j$ in class $k$;
  - $\sigma_{jk}^2$: the variance of the $j^{th}$ feature $X_j$ in class $k$.

- Since we assume Gaussian densities, Gaussian Naive Bayes is best suited for continuous features.

- In `scikit-learn` `GaussianNB` implements the Gaussian Naive Bayes algorithm for classification.

# Hands-on Session

- Please go to the **"Gaussian Naive Bayes in Scikit-Learn"** in the lecture code.

# Coin Flipping

# Rolling Dices

## Some terminologies on Bernoulli and Multinomial Distributions

- **Bernoulli distribution** models an unfair coin flip, head & tail, once of probabilities p and 1-p, respectively.
- **Binomial distribution** models an unfair coin-flip N times independently
- **Multinomial distribution** models an M-sided unfair dice rolling N times independently.
- If we take N=1, a binomial distribution reduces to a **Bernoulli distribution**
- If we take N=1, a multinomial distribution reduces to a **categorical distribution**
- Suppose that we flip an unfair coin N times, the result is a long sequence

  H, T, T, T, H, H, .........H, T, .....T.

- This is a text (long sentence) with two words 'H' and 'T'.
- **binomial distribution** models how many times do the head and the tail occur in a sample sequence.

# Continued

- Suppose that the faces of an unfair dice is coded by M distinct symbols, S1, S2, …., SM, then the result of N independent flips of the dice is nothing but a long sequence, e.g.:  S1, S1, S1, S2, S1, S3, …..SM, S3, ….S2.
- This is nothing but a long sentence formed by the vocabulary

S1, S2, ….., SM.

- We model on the counts S1 occurring in this 'sentence', S2 occurring in the sentence, …., SM occurring in the sentence. This is what multinomial distribution tries to capture.
- In general a multinomial distribution is determined by the probabilities of rolling the dice with symbols S1, S2, S3, ….SM, summing to 1.
- Alternatively, we may picture multinomial distribution as modeling the repeated drawings from a bag of symbols.

# Bag of Words

## Bag of Words Continued

- This is known as the '**bag of words**' model in **NLP**.
- Suppose that we are given a collection of documents (corpus), where each document is a long string of English words. Instead of working with this long token string, the multinomial model focuses on the count each token class occurs discarding the relative positions of the tokens.  It resolved in a much simplified feature space than the original text string.
- If a document, say a novel, contains 2 million words, but the corpus contains 10 thousands distinct word-tokens, we use the count vector of 10000 features to represent this document.
- Given each word-token, its value counts how many times a word-token occurs in this particular document.

# Bag of Words Model Continued

- This is like ignoring the order that the words occur and draw them independently and repeatedly from the '**bag**' of words to generate the **random** sentences.
- In a over-simplifying example, consider a short sentence: I love data science, and you love data science, too.
- We view 'and', 'I', 'love', 'you', 'data', 'science', 'too' as our features.
- Instead of training a model to analyze the raw text string, we represent the text string by a one-sample data frame

| and | i | love | you | data | science | too |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 2 | 2 | 1 |

## Multinomial Naive Bayes

- If all the columns of the raw data are categorical within the same value range, we may form the secondary table counting the times each value (our new feature) occurs.
- we can parameterize the multinomial distribution by vectors $\theta_k = (\theta_{k1}, \theta_{k2}, \ldots, \theta_{kn})$ for each class $k$,
  where:
  - $n$: the number of features ( the different values of the raw columns).
  - $\theta_{ki}$: probability $P(x_i | k)$ of feature $i$ appearing in a sample labelled to class $k$.
- `In scikit-learn MultinomialNB` implements the naive Bayes algorithm for multinomial distributed data, and is widely used in text classification/categorization.

# Multinomial Naive Bayes Example

- In our spam email data, we choose three words to build the model: "sale", "money", "work", denoted by $x_1$, $x_2$, $x_3$.
  - Among all the spams, "sale" appears 48 times,"money" appears 50 times, "work" 2 times, 100 in total.

    Thus we estimate:

    - $\theta_1$ = {0.48, 0.50, 0.02}
  - Among the non-spam emails, the frequency count of $x_1$, $x_2$, $x_3$ are 5, 10, 85, respectively.

    Thus we estimate:

    - $\theta_0$ = {0.05, 0.10, 0.85}

# Hands-on Session

- Please go to the **"Multinomial Naive Bayes in Scikit-Learn"** in the lecture code.

NYC DATA SCIENCE ACADEMY

## Bernoulli Naive Bayes

- Bernoulli Naive Bayes is used for data that:
  - is distributed according to a multivariate Bernoulli distribution;
  - each feature is assumed to be a binary-valued variable.
- `BernoulliNB` implements the naive Bayes training and classification algorithms.

## Bernoulli Naive Bayes

- Consider the spam filter problem.  With Bernoulli naive Bayes we do not care about the frequency count of a feature. We are just interested in whether it appears or not.
- Given a feature $x_k$ which denotes a word, does it appear in an email or not? What is the probability of its appearance among different emails?

# Bernoulli Naive Bayes Example

- Suppose we have 80 non-spams, and the word "sale" (denoted by $x_k$) appears in 10 of them; we also have 20 spams, and $x_k$ appears in 16 of them. We use $y$ = 1 to label a spam email. Then:

$$p(x_k = 1|y = 1) = \frac{16}{20} = \frac{4}{5}, \quad p(x_k = 0|y = 1) = \frac{1}{5}$$
$$p(x_k = 1|y = 0) = \frac{10}{80} = \frac{1}{8}, \quad p(x_k = 0|y = 0) = \frac{7}{8}$$

- Given a new email which contains the word “sale”, we have class = 1. If we use this single feature to predict:

$$p(y = 1|x_k = 1) = \frac{p(y = 1)p(x_k = 1|y = 1)}{p(x_k = 1)} = \frac{\frac{20}{100} \times \frac{4}{5}}{p(x_k = 1)} = \frac{0.16}{p(x_k = 1)}$$
$$p(y = 0|x_k = 1) = \frac{p(y = 0)p(x_k = 1|y = 0)}{p(x_k = 1)} = \frac{\frac{80}{100} \times \frac{1}{8}}{p(x_k = 1)} = \frac{0.1}{p(x_k = 1)}$$

then we will label this email to be spam.

## Hands-on Session

- Please go to the **"Bernoulli Naive Bayes in Scikit-Learn"** in the lecture code.