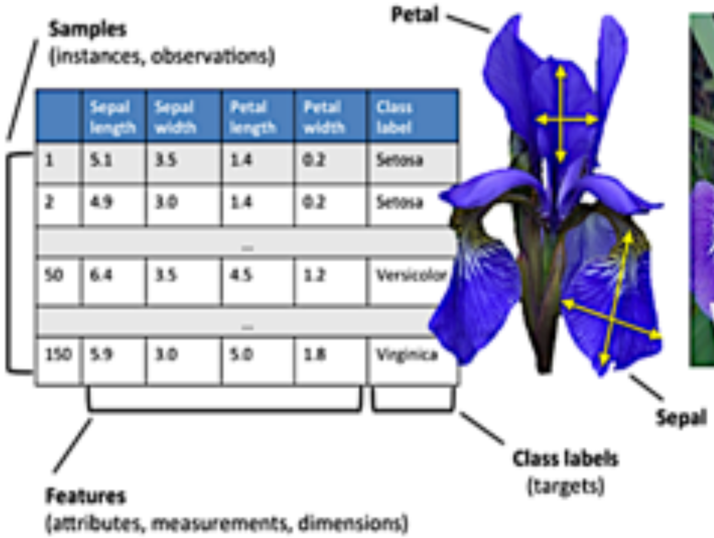
Classification
Generative Learning Algorithm
Naïve Bayes

Eun Soo Choi, Ph.D.

Classification:



- *Iris* flower data set

  $x = (x_1, x_2, x_3, x_4)$

  = (Sepal width, Sepal length, Petal width and Petal length)

  classify x as Versicolor or Setosa (or Virginica)

- Text Classification

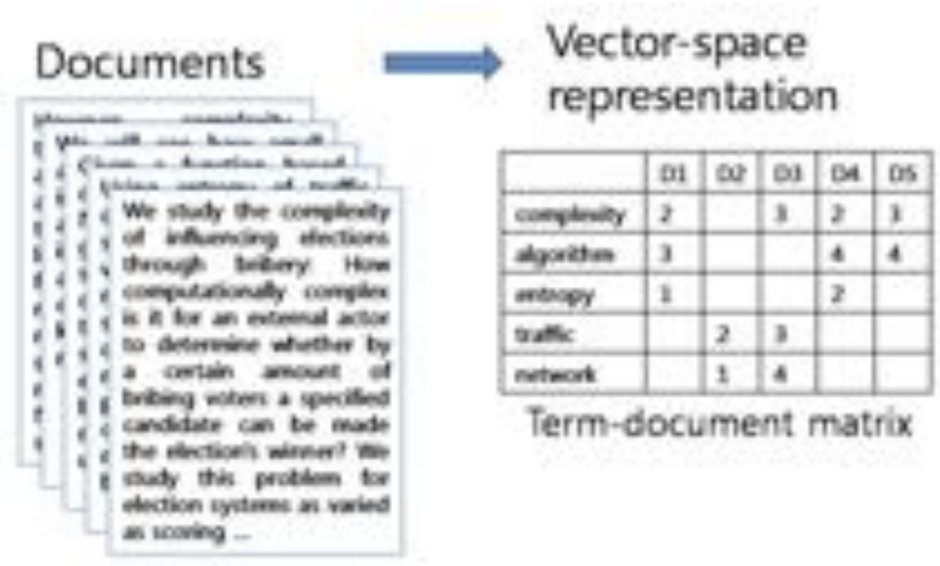
  x = word occurrence vector

  $$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{array}$$

  x = word count vector

  $$x = \begin{bmatrix} 2 \\ 5 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ 2 \end{bmatrix}$$

  classify x as Spam y=1 or Non-spam y=0



Classification in probabilistic perspectives

	Approach 1	Approach2
Training	Based on some features of flowers, we want to learn to distinguish between Versicolor or Setosa (= find a decision boundary that separates the Versicolor or Setosa)	Look at Versicolor and build a model of what Versicolor look like. Look at Setosa and build a separate model of what Setosa look like.
Test	Classify a new flower as either an Versicolor or Setosa (= predict depending on which side of the decision boundary it falls)	Match a new flower feature against the Versicolor model and against the Setosa model, and then classify whether the new flower feature looks more like the Versicolor or Setosa.

Classification in probabilistic perspectives

	Discriminative Learning Algorithm	Generative Learning Algorithms
Train	p(y|x; θ) Maximum Likelihood Estimation (MLE) arg $\max_{\theta} \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta)$	p(x|y=1) and p(x|y=0), and p(y) $p(y=k \mid x) = \frac{p(x \mid y=k)p(y=k)}{p(x)}$ Maximum Likelihood Estimation (MLE) arg max $\prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)})\, p(y^{(i)})$
Test	$\hat{y}$ = p(y=1|x) > or < threshold (0.5)	Maximum A Posterior (MAP) Estimation $\hat{y} = \arg\max_y p(y=k \mid x) = \arg\max_y \frac{p(x \mid y=k)p(y=k)}{p(x)}$ $= \arg\max_y p(x \mid y=k)p(y=k)$
Model	Everything else other than Naïve Bayes Logistic Regression *p*(y|x; θ) Linear Regression *f*(y|x; θ) …	Naïve Bayes • Gaussian NB: Gaussian Discriminant Analysis (GDA) • Bernoulli NB • Multinomial NB • Complement NB ...

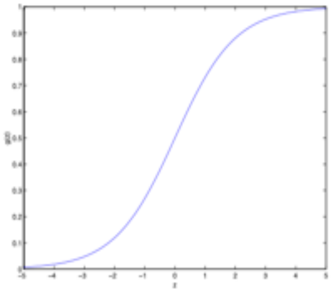
Discriminative Learning Algorithm: Logistic Regression

- Feature vector x = (x_1, x_2, x_3, x_4), class variable y ={0, 1}
- Step1 linearly transform x using *weight* or *coefficient* θ

  $z = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$
- Step2 Non-linear transform between 0 and 1 (for default class y=1)

  $\hat{y}$ = p(y=1|x; θ) = sigmoid(z) = $\frac{1}{1+e^{-z}}$

  p(y=0|x; θ) = 1 - $\hat{y}$



- Step 3 Define conditional probability p(y|x; θ) = $\hat{y}^y(1-\hat{y})^{1-y} = \begin{cases} p(y=1|x;\theta) = \hat{y} \\ p(y=0|x;\theta) = 1-\hat{y} \end{cases}$

  -> We want to maximize the likelihood of the parameter θ for the entire training set!
- Maximum Likelihood Estimation (MLE): Choose the parameter θ that maximizes the likelihood of θ

  $\mathcal{L}(\theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta)$
- Log likelihood: Easier to maximize

  $\ell(\theta) = \log \mathcal{L}(\theta) = \log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)};\theta) = \sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})$
- Loss/Cost function = Negative log likelihood to be minimized

  $L(\theta) = -\ell(\theta) = -\sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)};\theta)$
- How to minimize the loss/cost function? Use gradient descent algorithm
  - $\frac{\partial L}{\partial \theta_1}$ and $\theta_1 = \theta_1 - \alpha \frac{\partial L}{\partial \theta_1}$
- Prediction: If $\hat{y}$ = p(y=1|x) >= threshold(0.5), classify as 1 else 0

Review: Probability Theory 1

- Joint Probability and Conditional Probability

$$p(y|x) = \frac{p(x, y)}{p(x)} \qquad p(x, y) = p(x|y)p(y)$$

$$p(x|y) = \frac{p(x, y)}{p(y)} \qquad = p(y|x)p(x)$$

- Bayes' Theorem

$$p(y|x) = \frac{p(x, y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$$

- Chain Rule

$$p(x_1, x_2) = p(x_1)p(x_2|x_1)$$

$$p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)$$

$$p(y, x_1, x_2) = p(y)p(x_1|y)p(x_2|y, x_1)$$

- Independence

$$p(x_1, x_2) = p(x_1)p(x_2)$$

$$p(x_1, \ldots, x_n) = p(x_1)\ldots p(x_n) = \prod_{i=1}^{n} p(x_i)$$

- Naïve Bayes assumption: Conditional mutual independence of x's given y

$$p(x_1, x_2|y) = p(x_1|y)p(x_2|y, x_1) = p(x_1|y)p(x_2|y)$$

$$p(x_1, \ldots, x_n|y) = p(x_1|y)\ldots p(x_n|y, x_1,\ldots,x_{n-1}) = p(x_1|y)\ldots p(x_n|y) = \prod_{i=1}^{n} p(x_i|y)$$

$$\begin{aligned} p(y|x_1, \ldots, x_n) &= \frac{p(y, x_1, \ldots, x_n)}{p(x_1, \ldots, x_n)} \\ &= \frac{p(y)p(x_1, \ldots, x_n|y)}{p(x)} \\ &= \frac{p(y)p(x_1|y)\ldots p(x_n|y, x_1,\ldots,x_{n-1})}{p(x)} \\ &= \frac{p(y)p(x_1|y)\ldots p(x_n|y)}{p(x)} \\ &= \frac{p(y)\prod_{i=1}^{n} p(x_i|y)}{p(x)} \end{aligned}$$

Review: Probability Theory 2

- Bernoulli distribution: x = (0, 1) 2 categories # flipping a coin

$$p(\text{x}; \phi) = \begin{cases} \phi \ if \ x = 1 \\ 1 - \phi \ if \ x = 0 \end{cases} = \phi^{x}(1-\phi)^{1-x}$$

- Multinomial distribution: x = (1, …, k) k categories # rolling a die

$$p(\text{x} = \text{i}) = \theta_i \qquad (\theta_1, \theta_2, \ldots, \theta_k) \quad \Sigma\theta_i = 1$$

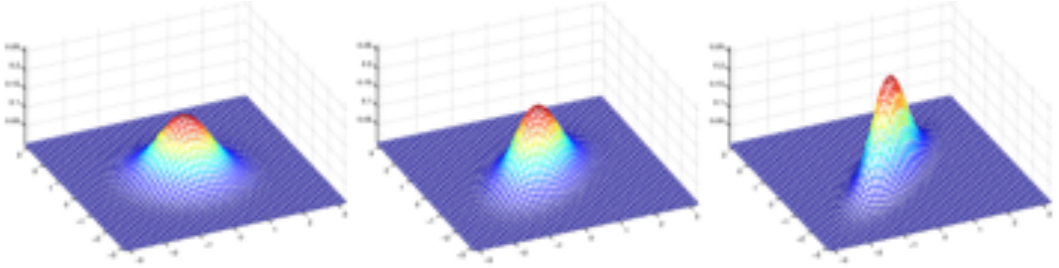
$$p(\text{x}) = \theta_1^{[x=1]}\, \theta_2^{[x=2]} \ldots \theta_k^{[x=k]}$$

- Gaussian distribution:

$$p(\text{x}; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multivariate Gaussian distribution:

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



The figures above show Gaussians with mean 0, and with covariance matrices respectively

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \ \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \ .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

Naïve Bayes

- Feature vector $x = (x_1, \ldots, x_n)$, class variable $y = \{0, 1\}$
- Conditional probability + Bayes' Theorem + Chain Rule + Naïve Bayes assumption (conditional mutual independence x's given y)

$$p(y|x) = \frac{p(x,y)}{p(x)}$$
$$= \frac{p(y)p(x|y)}{p(x)}$$
$$= \frac{p(y)\prod_{i=1}^{n} p(x_i|y)}{p(x_1, \ldots, x_n)}$$
$$p(y|x) \propto p(y)p(x|y)$$

$p(y)$: prior probability
$p(x|y)$: likelihood
$p(x)$: evidence
$p(y|x)$: posterior probability

  - Gaussian Discriminant Analysis (GDA): ϕ, μ_0, μ_1, Σ
  - Bernoulli NB: ϕ, $\phi_{j|y=1}$, $\phi_{j|y=0}$
  - Multinomial NB: ϕ, $\theta_{j|y=1}$, $\theta_{j|y=0}$

- Maximum Likelihood Estimation(MLE): Choose the parameters that maximizes the probability of observed data *p*(x, y)

$$\mathcal{L} = \prod_{i=1}^{m} p(y^{(i)})p(x^{(i)}|\, y^{(i)})$$
$$\ell = \log \mathcal{L}(\phi, \mu0, \mu1, \Sigma)$$

- Predict: Maximum A Posterior (MAP), a Bayesian method

$$p(y|x) \propto p(y)p(x|y)$$
$$\hat{y} = \arg\max_{y}\; p(y)p(x|y)$$

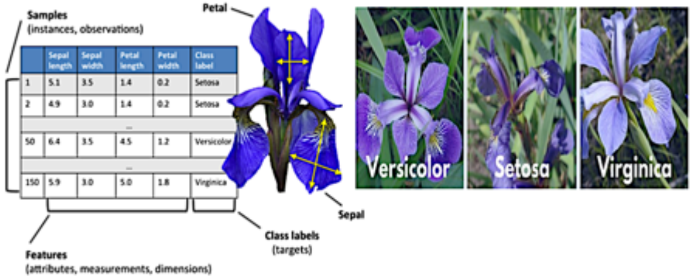
Naïve Bayes

- **Gaussian Discriminant Analysis** (GDA aka Gaussian NB): Input features x are continuous random variable

  $p(y = 1) = \phi_y$

  $p(y = 1) = 1 - \phi_y$

$$
\begin{aligned}
p(y) &= \phi^y (1-\phi)^{1-y} \\
p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right) \\
p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)
\end{aligned}
$$



- **Bernoulli Naïve Bayes**: Input features x are discrete as {0, 1}

  $\phi_y = p(y = 1)$ $p(y = 1) = 1 - \phi_y$

  $\phi_{j|y=1} = p(x_j = 1 | y = 1)$ $p(x_j = 0 | y = 1) = 1 - \phi_{j|y=1}$

  $\phi_{j|y=0} = p(x_j = 1 | y = 0)$ $p(x_j = 0 | y = 0) = 1 - \phi_{j|y=0}$

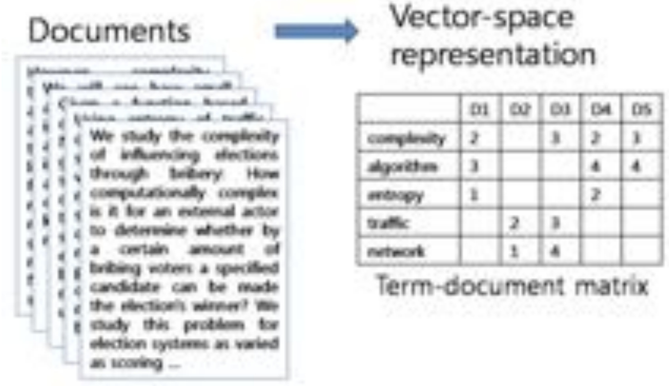
$$
x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}
\begin{array}{l} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{array}
$$

- **Multinomial Naïve Bayes**: Input features x are discrete

  $\phi_y = p(y = 1)$ $p(y = 1) = 1 - \phi_y$

  $\theta_{j|y=1} = p(x_j | y = 1)$

  $\theta_{j|y=0} = p(x_j | y = 0)$



Naïve Bayes

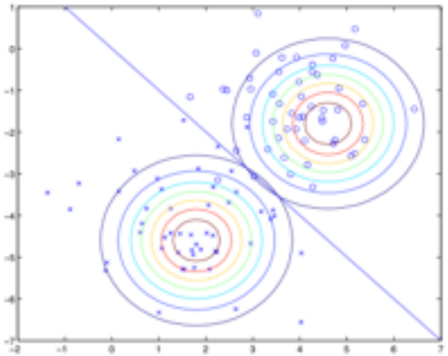
- **Gaussian Discriminant Analysis** (GDA aka Gaussian NB): Input features x are continuo

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T.$$



- **Bernoulli Naïve Bayes**: Input features x are discrete as {0, 1}

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

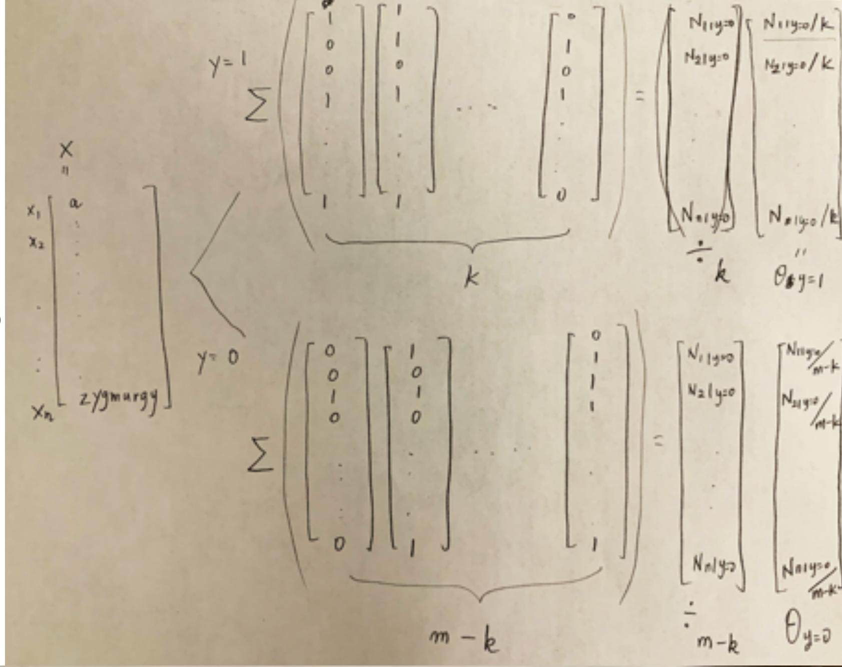
$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}$$

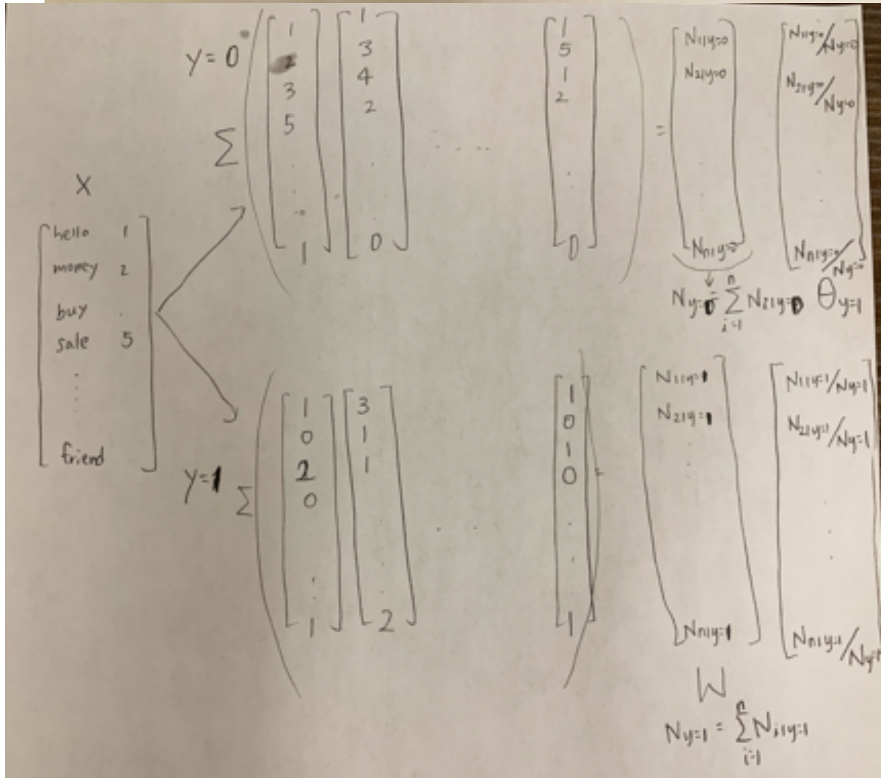
- **Multinomial Naïve Bayes**: Input features x are discrete

$$\theta_{j|y=1} = \frac{N_{j|y=0}}{N_{y=0}}$$

$$\theta_{j|y=0} = \frac{N_{j|y=1}}{N_{y=1}}$$

$$\phi_y = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

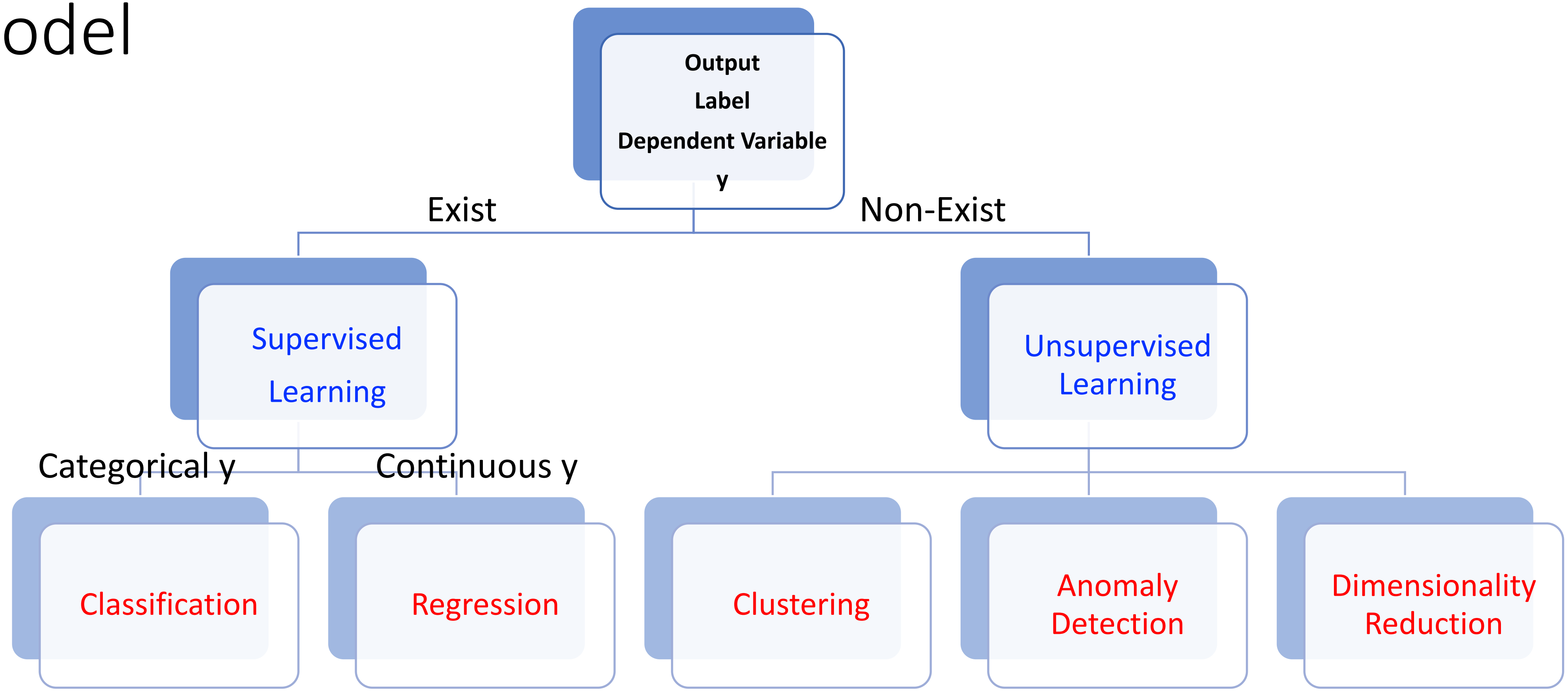




Classification in probabilistic perspectives

	Discriminative Learning Algorithm	Generative Learning Algorithms
Train	$p(y\|x; \theta)$ Maximum Likelihood Estimation (MLE) $\arg\max_\theta \prod_{i=1}^{m} p(y^{(i)}\|x^{(i)}; \theta)$	$p(x\|y=1)$ and $p(x\|y=0)$, and $p(y)$ $p(y=k\|x) = \frac{p(x\|y=k)p(y=k)}{p(x)}$ Maximum Likelihood Estimation (MLE) $\arg\max \prod_{i=1}^{m} p(y^{(i)}\|x^{(i)})\, p(y^{(i)})$
Test	$\hat{y} = p(y=1\|x)$ > or < threshold (0.5)	Maximum A Posterior (MAP) Estimation $\hat{y} = \arg\max_y p(y=k\|x) = \arg\max_y \frac{p(x\|y=k)p(y=k)}{p(x)}$ $= \arg\max_y p(x\|y=k)p(y=k)$
Model	Everything else other than Naïve Bayes Logistic Regression $p(y\|x; \theta)$ Linear Regression $f(y\|x; \theta)$ …	Naïve Bayes • Gaussian NB: Gaussian Discriminant Analysis (GDA) • Bernoulli NB • Multinomial NB • Complement NB ...

ML model



- Logistic Regression (two classes)
- Softmax Regression (Multinomial Logistic Regression)
- K-Nearest Neighbors (KNN)
- Support Vector Classification
- Decision Tree Classification
- Random Forest Classification
- Naïve Bayes

- Linear Regression
- Polynomial Linear Regression
  - Ridge Regression (L1 Regularization)
  - Lasso Regression (L2 Regularization)
  - Elastic Net (L1 + L2 Regularization)
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

ML model in Probabilistic Perspective

<table>
<tr><th></th><th>Classification</th><th>Regression</th></tr>
<tr><td rowspan="4">Discriminative Learning Algorithm</td><td>Logistic Regression
Softmax Regression (Multinomial Logistic Regression)
K-Nearest Neighbors (KNN)</td><td>Linear Regression
Simple Linear Regression
Multiple Linear Regression</td></tr>
<tr><td></td><td>Polynomial Regression
Ridge Regression (L1 Regularization)
Lasso Regression (L2 Regularization)
Elastic Net Regression (L1 + L2 Regularization)</td></tr>
<tr><td colspan="2">Support Vector Machine Model
Maximal Margin Classification
Support Vector Classification/ Regression
Linear Kernel
Polynomial Kernel
Radial Basis Kernel (RBS)</td></tr>
<tr><td colspan="2">Tree-Based Model
Decision Tree Classification/ Regression
Random Forest Classification/ Regression</td></tr>
<tr><td>Generative Learning Algorithm</td><td>Naïve Bayes
Gaussian NB: Gaussian Discriminant Analysis (GDA)
Bernoulli NB
Multinomial NB
Complement NB
Categorical NB</td><td></td></tr>
</table>