Support Vector Machine

MSB881

October 6, 2020

Youngsun Kwon

yokwon@kaist.ac.kr

Business and Technology Management, KAIST

http://itip.kaist.ac.kr

1. Capacity, Overfitting and Underfitting

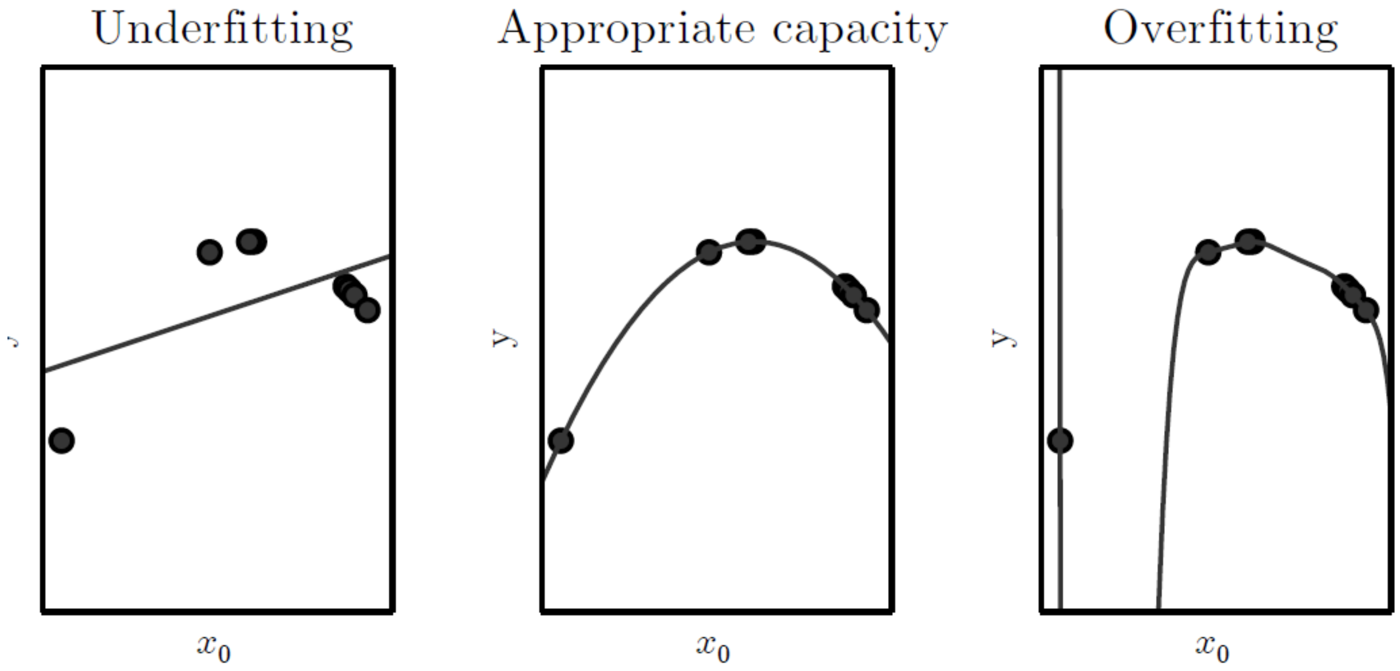
- Issues in relation with performance of ML algorithms
  - **Generalization**: ability to perform well on unseen (new) inputs.
    - Training error: error measured on the training set.
    - Generalization error (test error): Expected value of the error on a new input.
    - How to measure? With test set!
  - Expected test error is greater than or equal to the expected value of training error. Why?
  - Two concerns in machine learning
    - Making the training error small
    - Making the gap between training and test error small

1. Capacity, Overfitting and Underfitting

- Issues in relation with performance of ML algorithms
  - **Underfitting**: the model is not able to obtain a sufficiently low error value on the training set.
  - **Overfitting**: the gap between training and test error is too large.
  - **Capacity**: ability to fit a wide variety of functions.
    - Models with low capacity is likely to face underfitting and those with high capacity overfitting.
    - Ex) Linear models with high degree polynomials is an example of high capacity

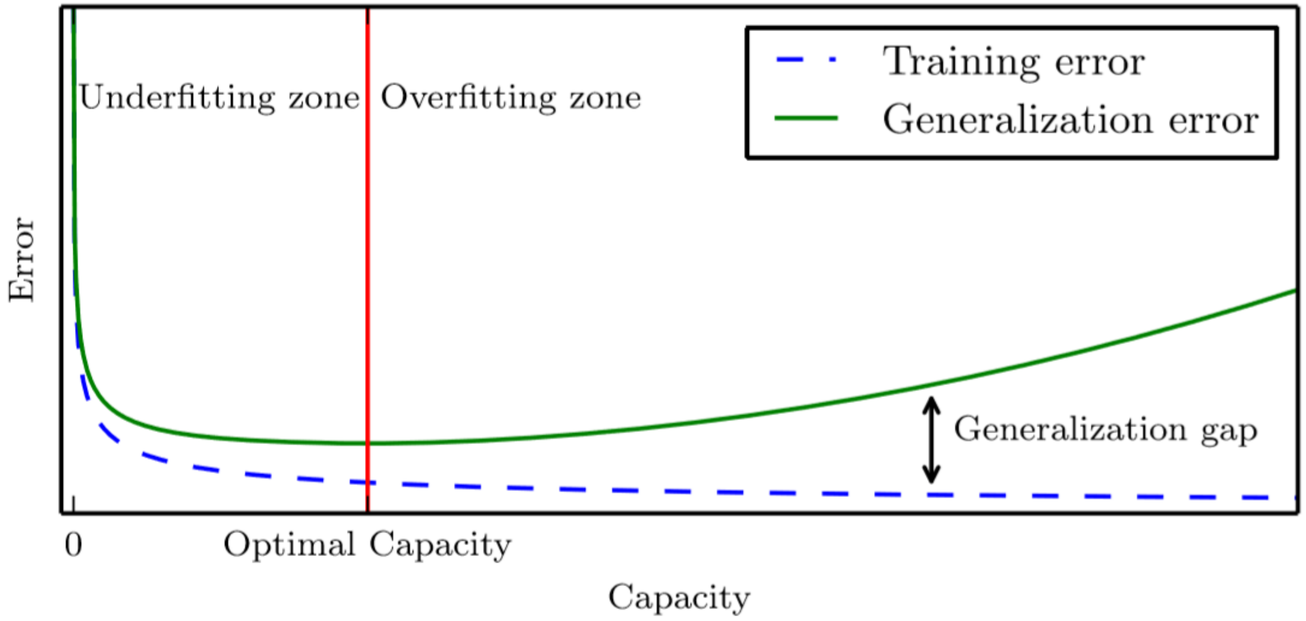
1. Capacity, Overfitting and Underfitting

- Issues in relation with performance of ML algorithms
  - Three models for underfitting and overfitting



1. Capacity, Overfitting and Underfitting

- Issues in relation with performance of ML algorithms
  - Among competing models, which one need to be chosen?
    - Rule of thumb(Occam's razor): choose the simplest one.
    - Choosing a model with appropriate capacity: optimal capacity



1. Capacity, Overfitting and Underfitting

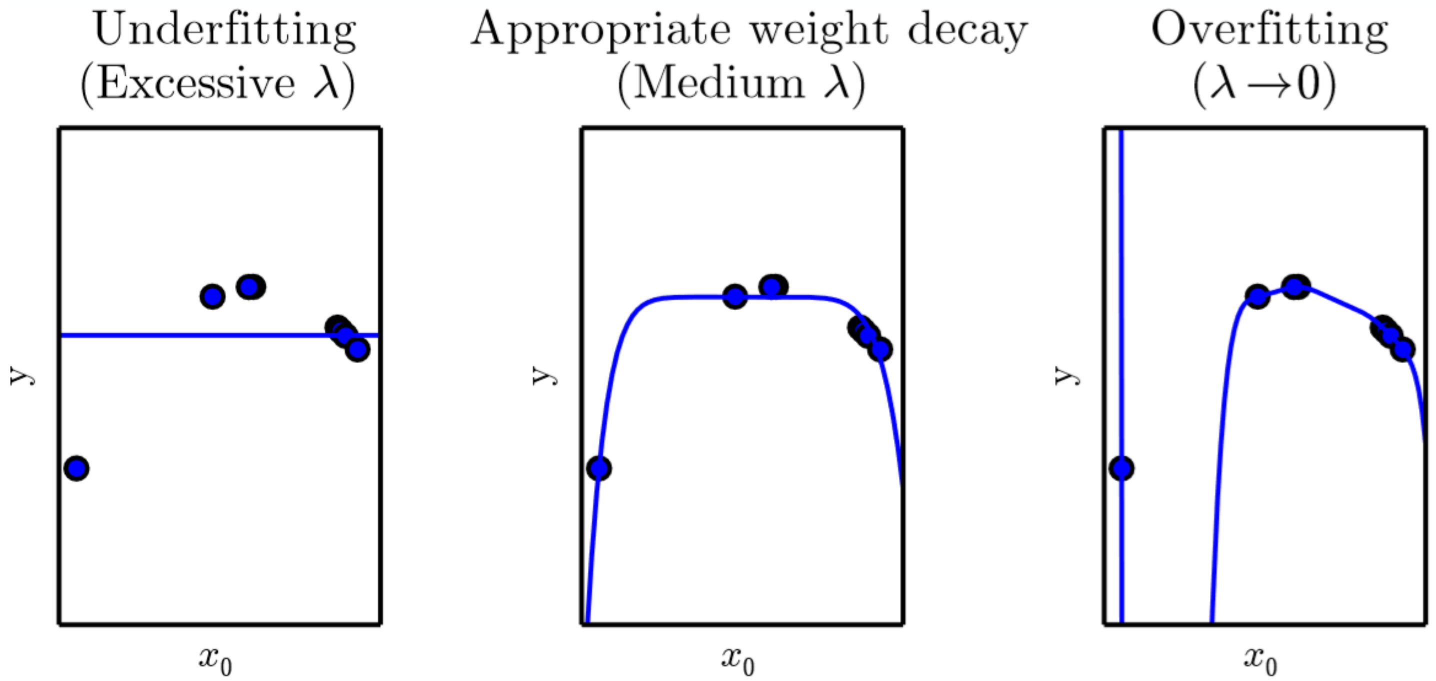
- Issues in relation with performance of ML algorithms
  - The size of training set matters!
    - Expected generalization error can never increase as the number of training examples increases!
  - No machine learning algorithm is UNIVERSALLY any better than any other!
    - The goal of ML research is not to seek a universal learning algorithm, nor the absolute best learning algorithm.

1. Capacity, Overfitting and Underfitting

- Issues in relation with performance of ML algorithms
  - Regularization
    - Any modification to a learning algorithm that is intended to reduce its generalization error but not its training error
    - An example is weight decay
    - Minimize $J(\mathbf{w}) = \mathrm{MSE}_{train} + \lambda\mathbf{w}^{\mathrm{T}}\mathbf{w}$
    - Trade off between fitting the training data and choosing a small model. The second term is a regularizer, a penalty of adding more features. $\boldsymbol{\lambda}$ is a capacity hyperparameter.

1. Capacity, Overfitting and Underfitting

- Issues in relation with performance of ML algorithms
  - Regularization
    - **Choosing λ**

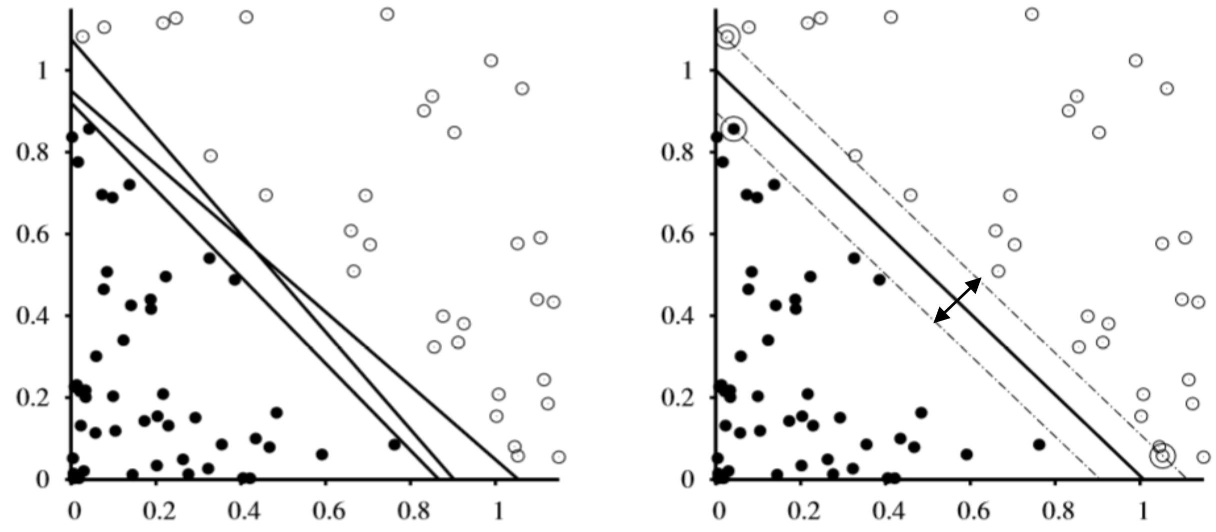


2. Support Vector Machine (SVM)

- A **popular supervised machine learning** algorithm.
- Used for **classification and regression**, but **mostly for classification**.
- Applications
  - Face detection
  - Classification of images
  - Bioinformatics – protein and cancer classification
  - Handwriting recognition
  - Etc.

2. Support Vector Machine (SVM)

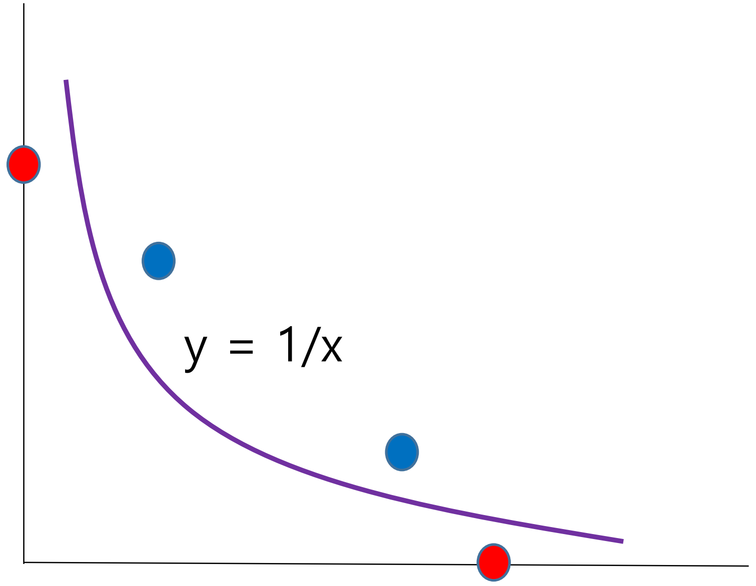
- **SVM** maps **a linear function to two classes** (with labels, +1 and -1 (or 0)), using **a linear separating hyperplane**. Search the space of **w** and b ($\mathbf{w} \cdot \mathbf{x} + b = 0$) to find the separator.
  - SVM construct a **maximum margin separator**. Which separator?
  - **Logistic regression** can also do the task.
  - Then **why SVM**?
    - Non-parametric method!



2. Support Vector Machine (SVM)

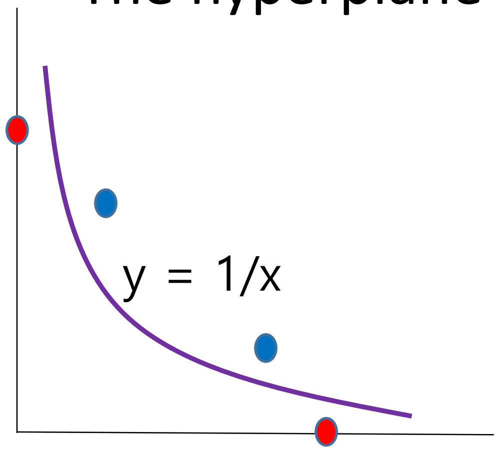
- **SVM** can present (imbed) the data into a **higher-dimensional space** using **kernel trick**.
  - **Data**, not linearly separable, can be **separable in the higher dimensional space**.
  - In the figure, linear separation is not possible.
    - $\mathbf{w} \cdot \mathbf{x} + b = 0$
  - Three functions: $x + y$, xy, x^2.

	(0,4)	(1,3)	(3,1)	(4,0)
x+y	4	4	4	4
xy	0	3	3	0
x^2	0	1	9	16

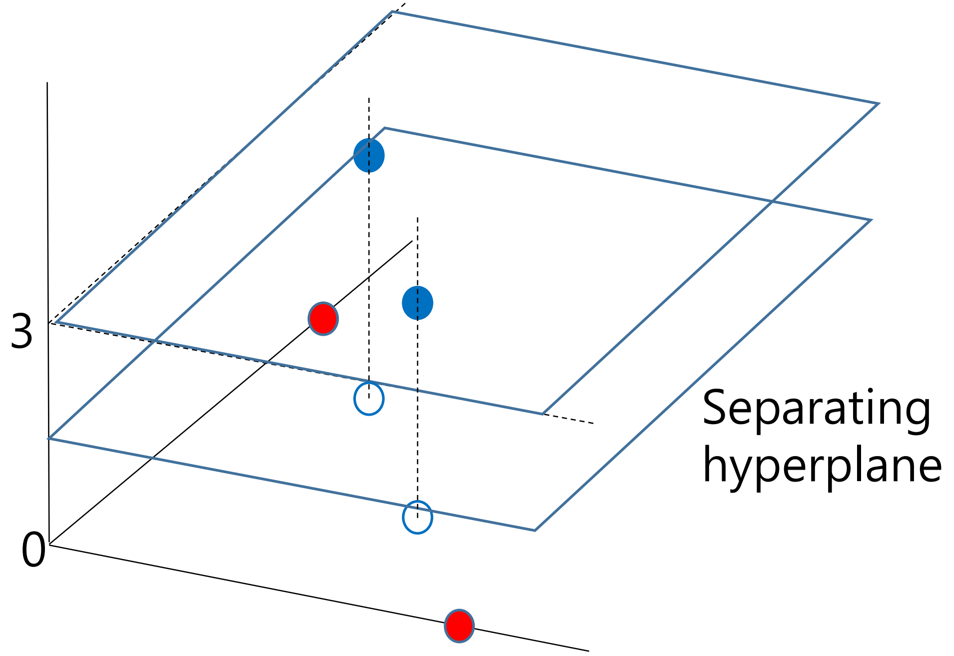


2. Support Vector Machine (SVM)

- **Data**, not linearly separable, can be **separable in the higher dimensional space**.
  - The hyperplane is actually nonlinear in the original space.



	(0,4)	(1,3)	(3,1)	(4,0)
x+y	4	4	4	4
xy	0	3	3	0
x^2	0	1	9	16



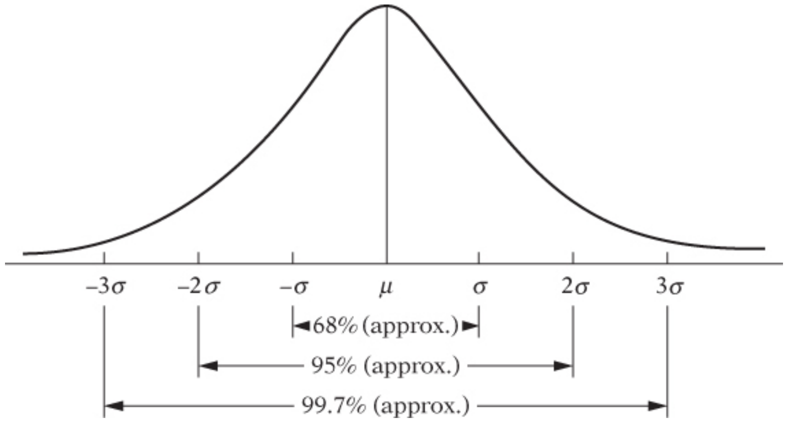
2. Support Vector Machine (SVM)

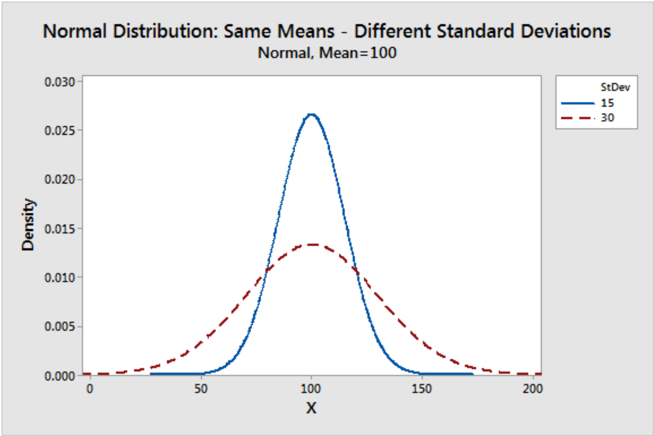
- **SVM kernels**.
  - Dual representation: $\mathbf{w} \cdot \mathbf{x} + b = b + \sum_{i=1}^{m} \alpha_i \mathbf{x} \cdot \mathbf{x}_i$
  - Polynomial kernel: $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \mathbf{x}_j + 1)^d$, d is the degree of the polynomial.
  - Gaussian kernel: $k(x, y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$
  - RBF(radial basis function): $k(x, y) = exp(-\gamma\|x-y\|^2)$, $\gamma > 0$.

2. Support Vector Machine (SVM)

- **Normal distribution (Gaussian distribution)**
  - One variable probability density function (pdf) with two parameters, μ and σ^2 .

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \qquad -\infty < x < \infty$$





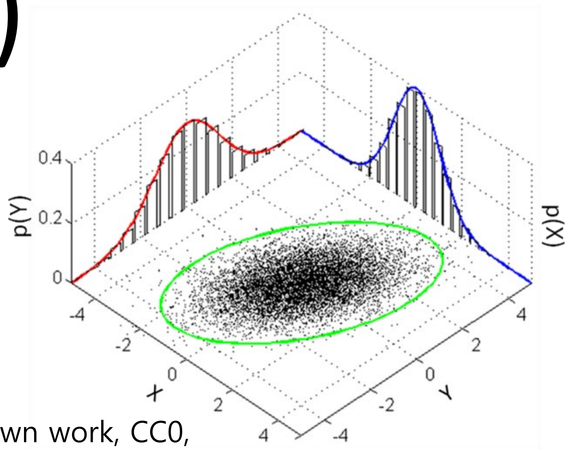
2. Support Vector Machine (SVM)

- **Normal distribution (Gaussian distribution)**
  - Multivariate Normal distribution

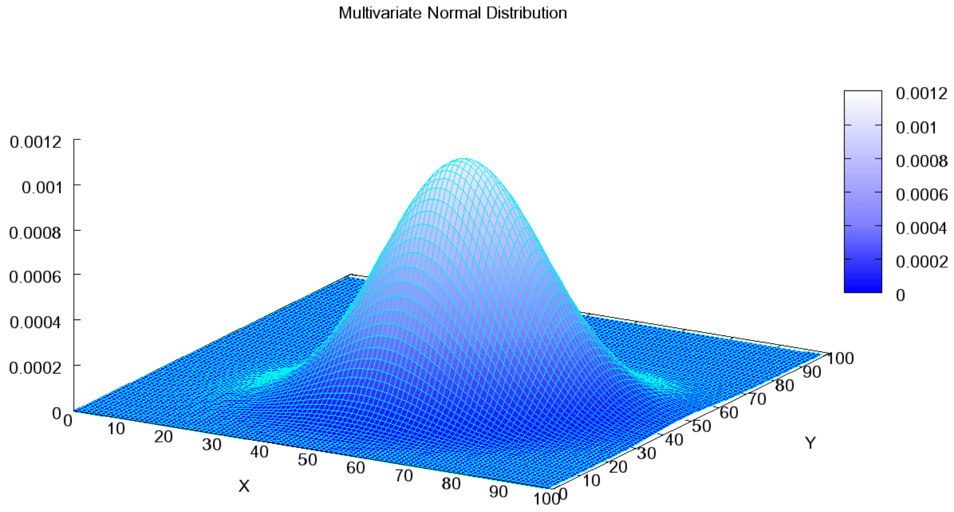
$$\boldsymbol{f(x)} = \frac{\mathbf{1}}{\sigma\sqrt{2\pi}}\,\boldsymbol{exp}\left(-\frac{\mathbf{1}}{\mathbf{2}}\frac{(\boldsymbol{x}-\mu)^{\mathbf{2}}}{\sigma^2}\right) \qquad -\infty < \mathrm{x} < \infty$$

$$f_{\mathbf{X}}(x_1,\ldots,x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}.$$

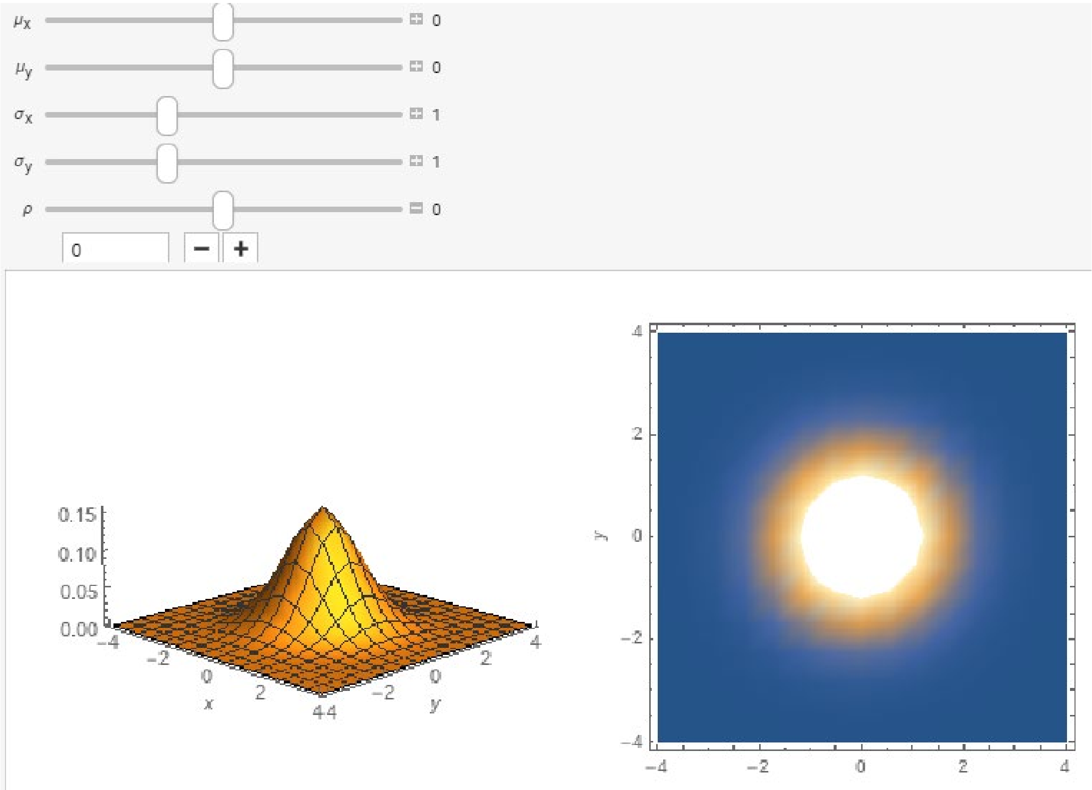


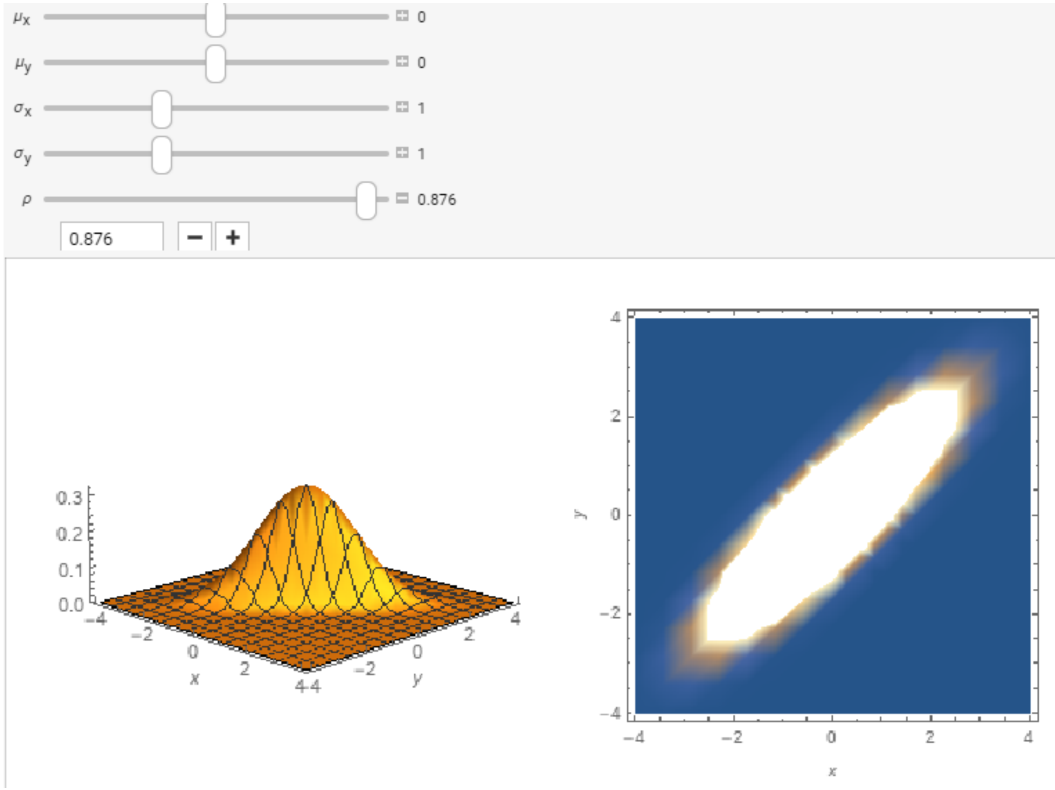
By Bscan - Own work, CC0,
https://commons.wikimedia.org/w/index.php?curid=25235145



CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1260349

2. Support Vector Machine (SVM)





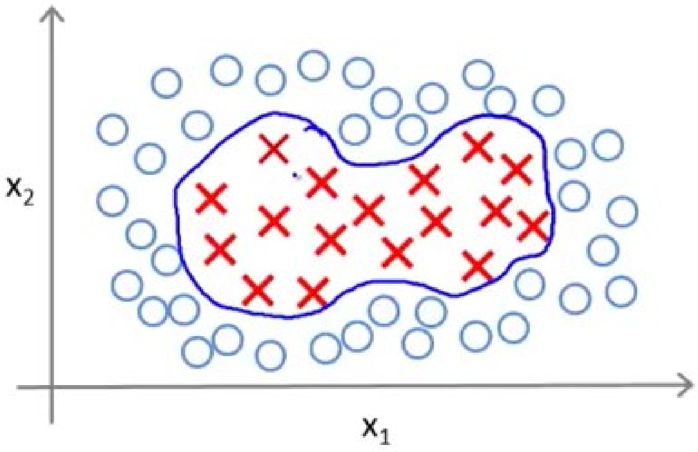
Contributed by: Chris Boucher (March 2011)

https://demonstrations.wolfram.com/TheBivariateNormalDistribution/

2. Support Vector Machine (SVM)

- **SVM kernels**.
  - Professor Andrew Ng
    - Each polynomial variable can be seen as a feature: choosing a better choice of the features.

Non-linear Decision Boundary



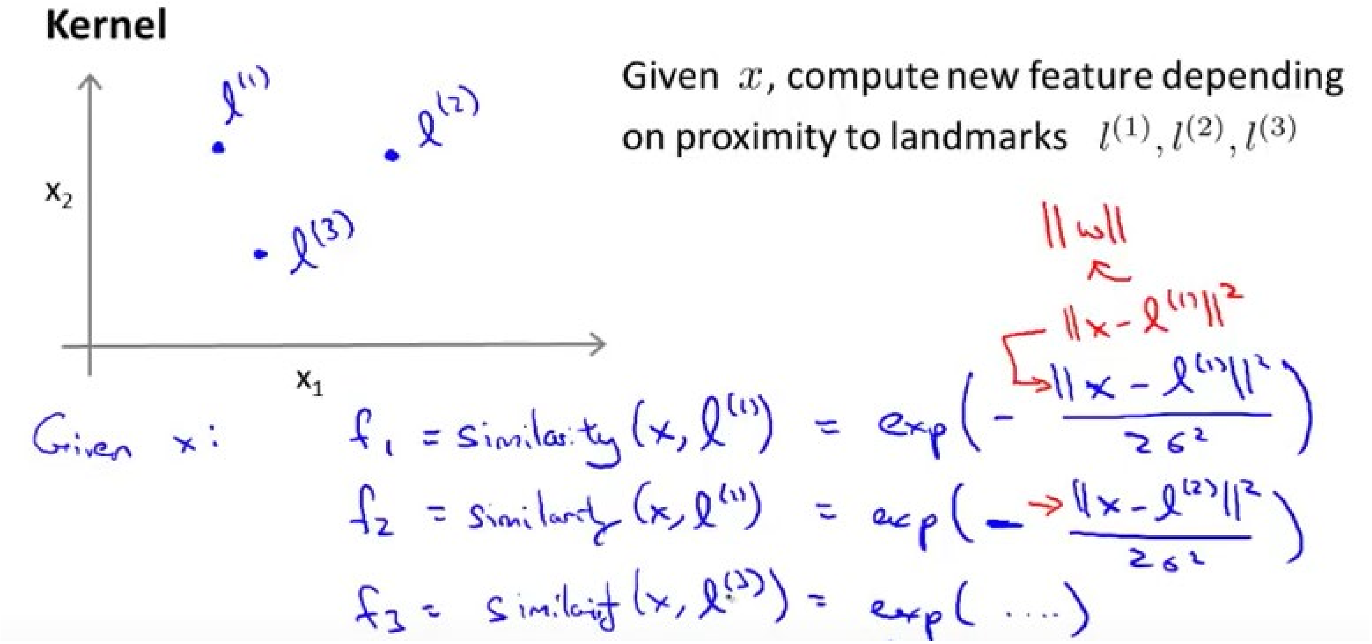
Predict $y = 1$ if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$$

Lecture 12.4 — Support Vector Machines | (Kernels-I) — [Machine Learning | Andrew Ng]

2. Support Vector Machine (SVM)

- **SVM kernels.**
  - Professor Andrew Ng
    - Similarity functions are kernel functions



Lecture 12.4 — Support Vector Machines | (Kernels-I) — [Machine Learning | Andrew Ng]

2. Support Vector Machine (SVM)

- **SVM kernels.**
  - Professor Andrew Ng

Kernels and Similarity

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\boxed{\|x-l^{(1)}\|^2}}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

If $x \approx l^{(1)}$:

$$f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

$l^{(1)} \to f_1$

$l^{(2)} \to f_2$

$l^{(3)} \to f_3$.

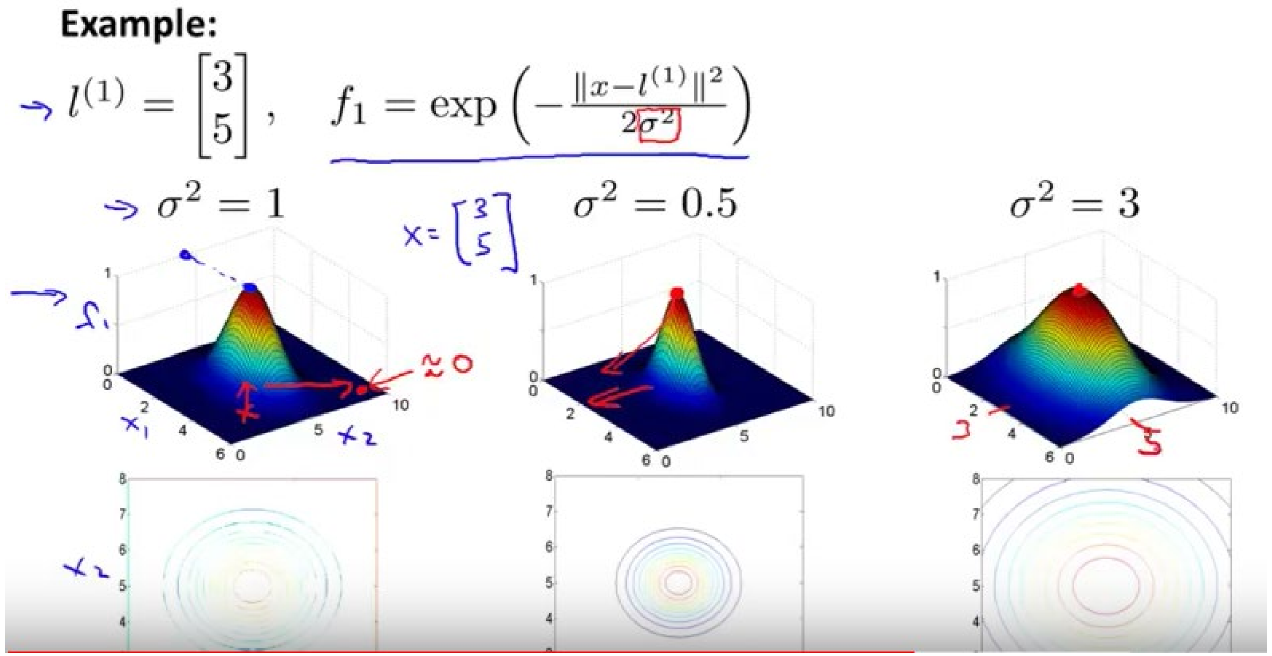
If x if far from $l^{(1)}$:

$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \approx 0.$$

Lecture 12.4 — Support Vector Machines | (Kernels-I) — [Machine Learning | Andrew Ng]

2. Support Vector Machine (SVM)

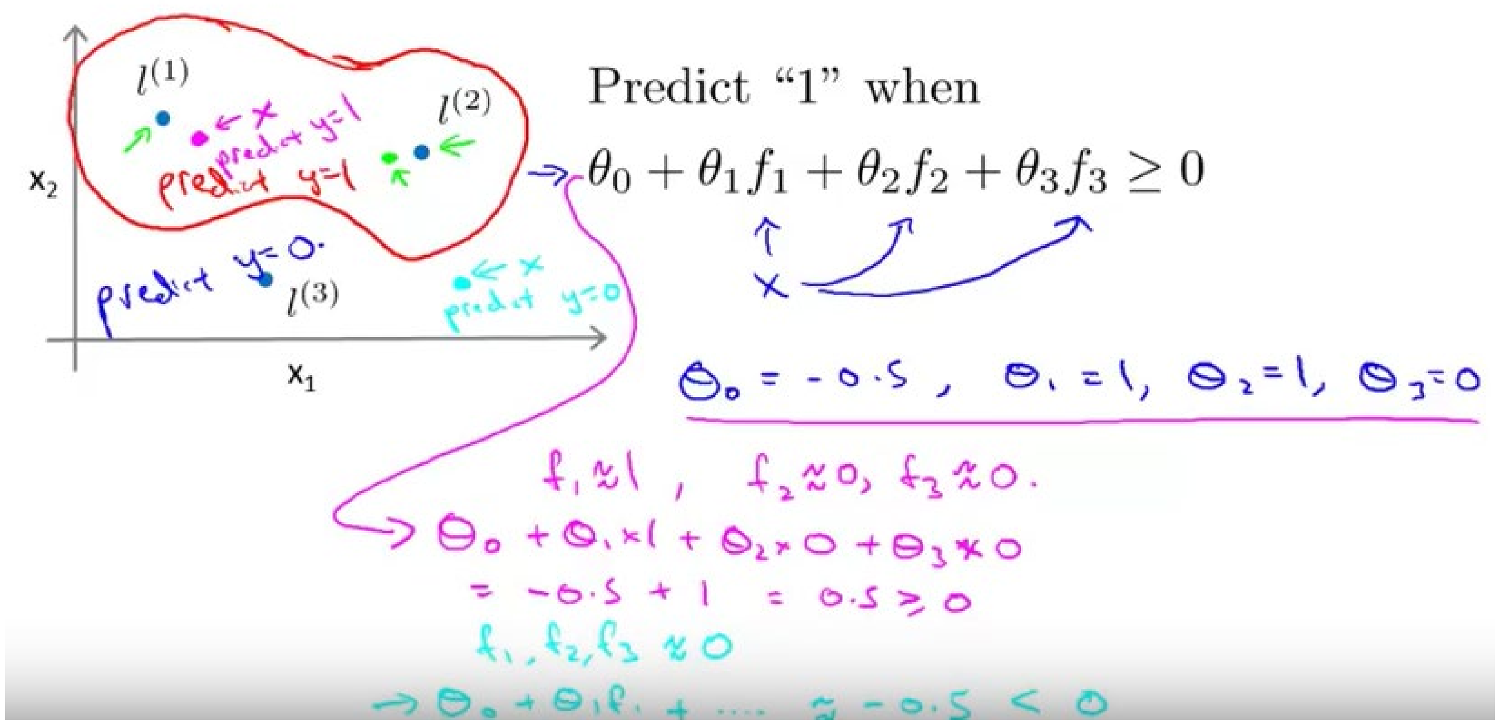
- **SVM kernels.**
  - Professor Andrew Ng
    - Sigma squared is the parameter of GK.



Lecture 12.4 — Support Vector Machines | (Kernels-I) — [Machine Learning | Andrew Ng]

2. Support Vector Machine (SVM)

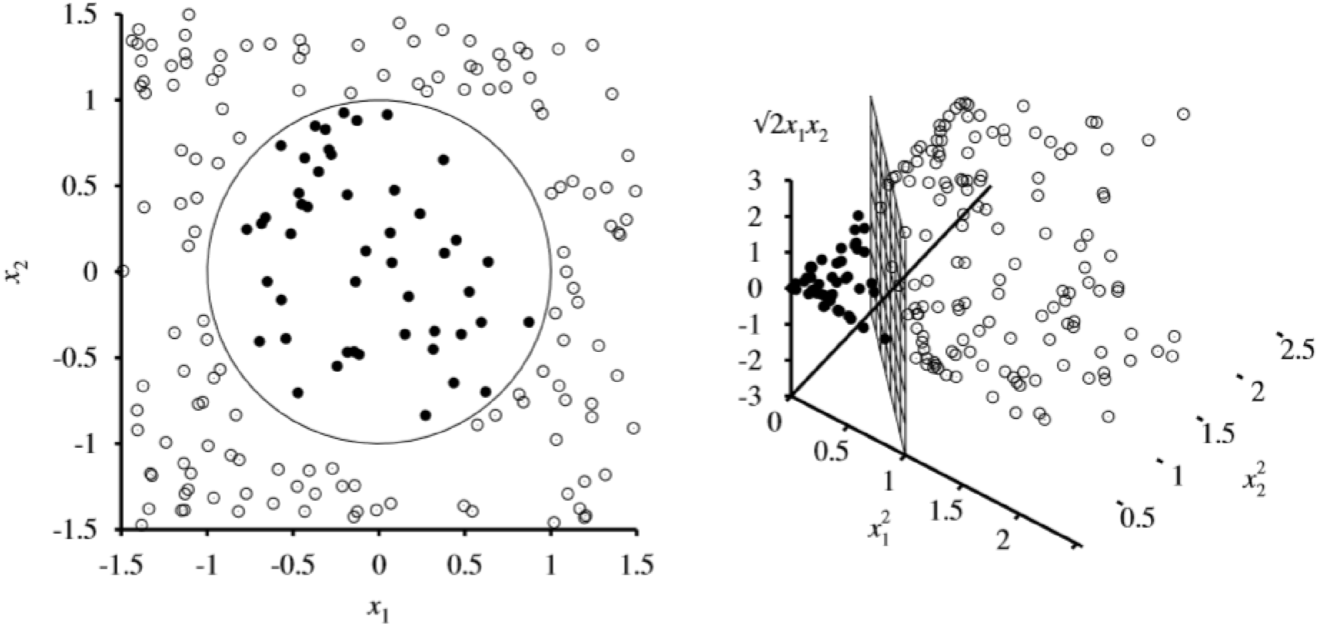
- **SVM kernels.**
  - Professor Andrew Ng



Lecture 12.4 — Support Vector Machines | (Kernels-I) — [Machine Learning | Andrew Ng]

2. Support Vector Machine (SVM)

- Support vector machines (SVM)
  - One weakness is high computational cost of training with large data.

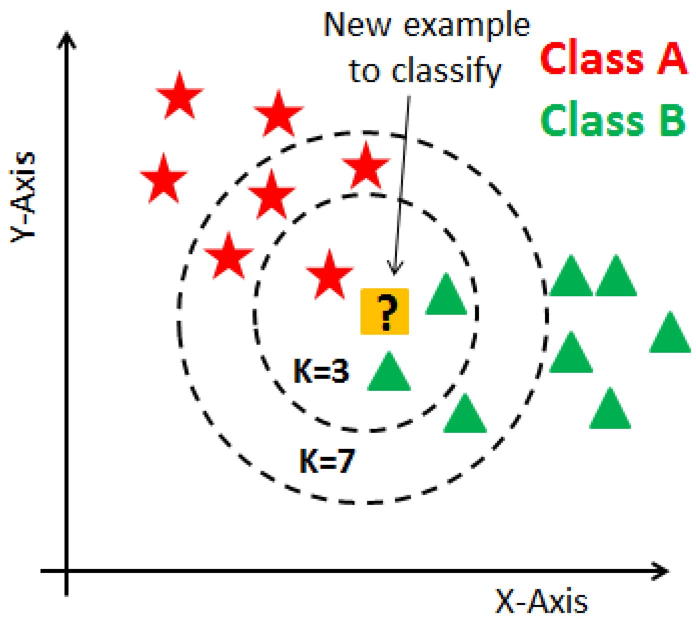


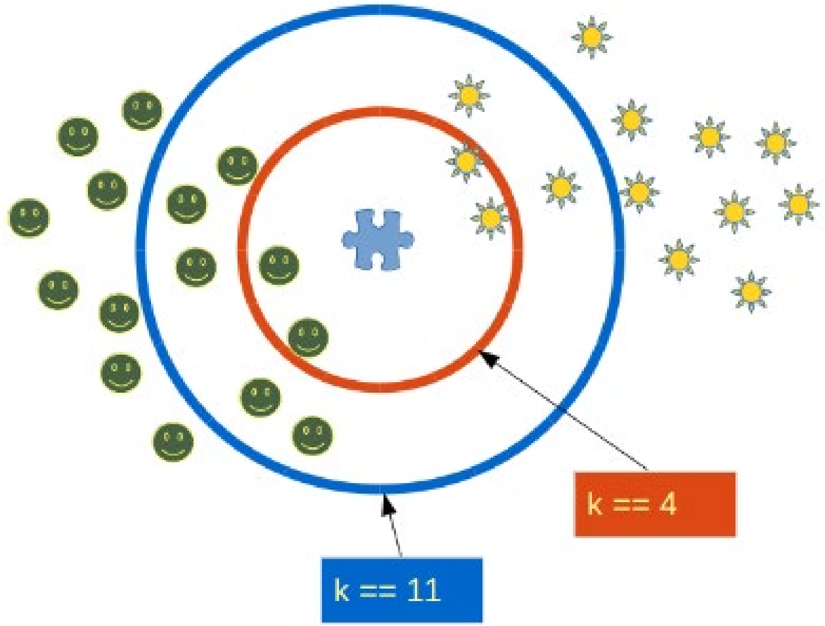
A kernel function with three features from Russell and Norvig.

3. K-nearest neighbors

- A non-parametric supervised learning algorithm
  - Find **the k closest point to a new instance** and then classify the point by majority vote of its k neighbors. Each object votes for their class and **the class with the most votes is taken as the prediction**.
- Algorithm properties
  - A family of techniques used for classification or regression.
  - Actually no training!
  - Simply return y value for x by looking up the k-nearest neighbors to x in the training data X.
  - Achieves very high capacity but it cannot learn that which feature is more discriminative than others.

3. K-nearest neighbors

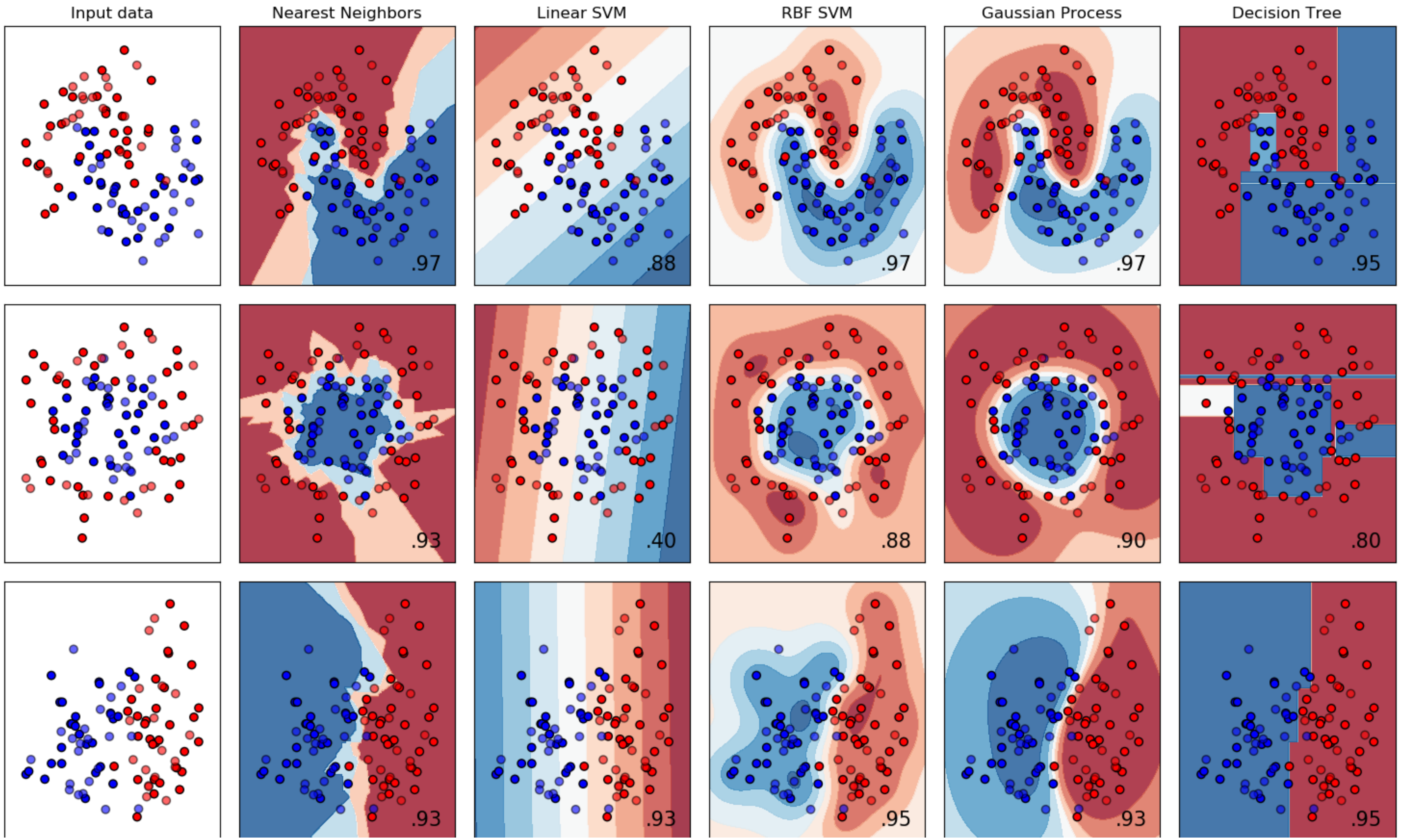




4. Scikit-learn library

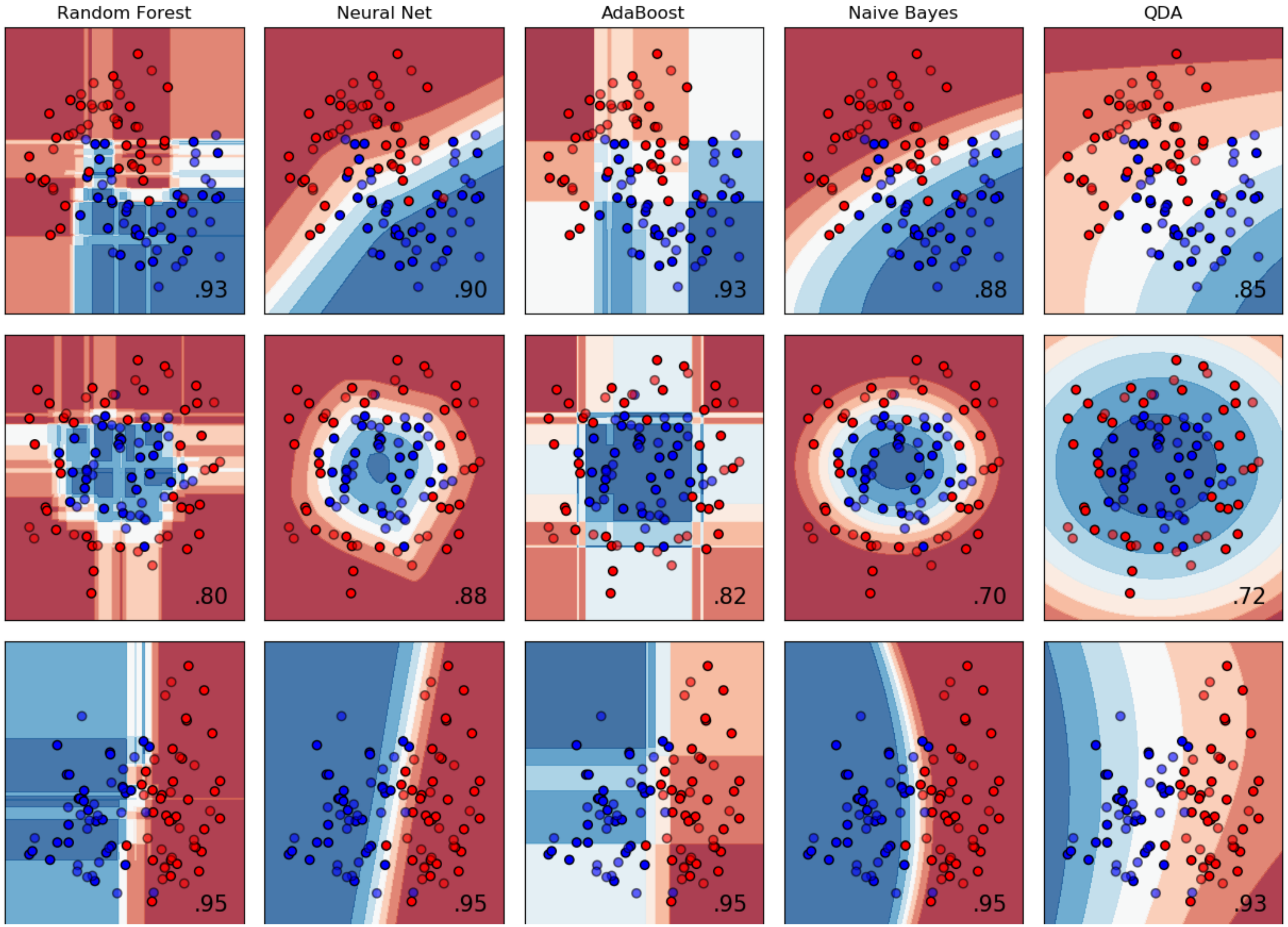
- Scikit-learn (sklearn) offers **machine learning algorithms** for classification, regression, clustering, dimension reduction, and so on.
  - Check if scikit-learn and scikit-image are installed in your working environment by using ‘conda list’ at your working environment.
  - If not, install also scikit-learn and scikit-image, which is for image processing.
  - Now you can use SVM module.

5. Classifier Comparison



https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

5. Classifier Comparison



https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html