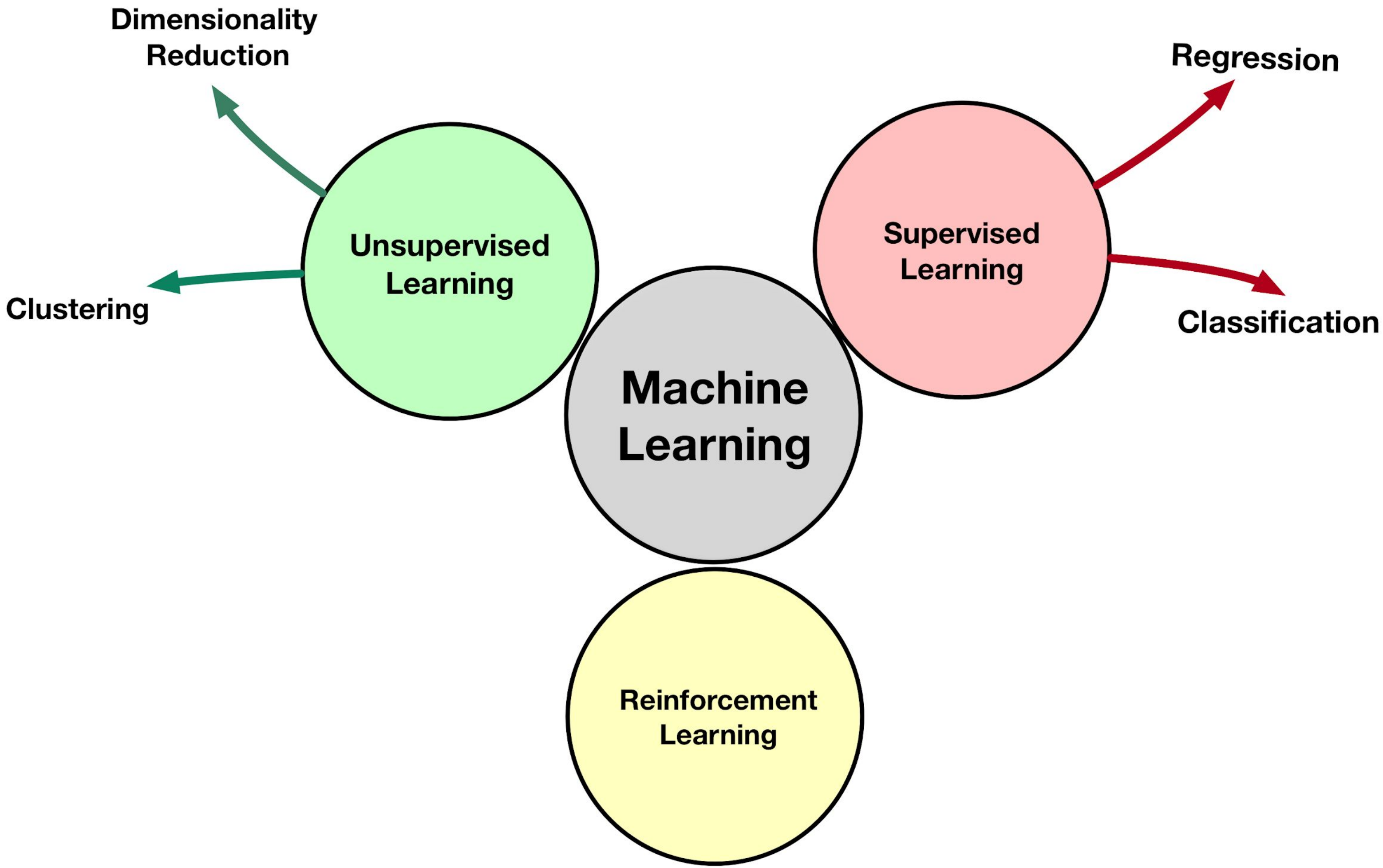
Supervised learning

Farnoosh Khodakarami

This material is made by

Farnoosh Khodakarami and Ali Madani


Dimensionality Reduction
Regression
Unsupervised Learning
Supervised Learning
Clustering
Classification
Machine Learning
Reinforcement Learning

Supervised vs Unsupervised Learning

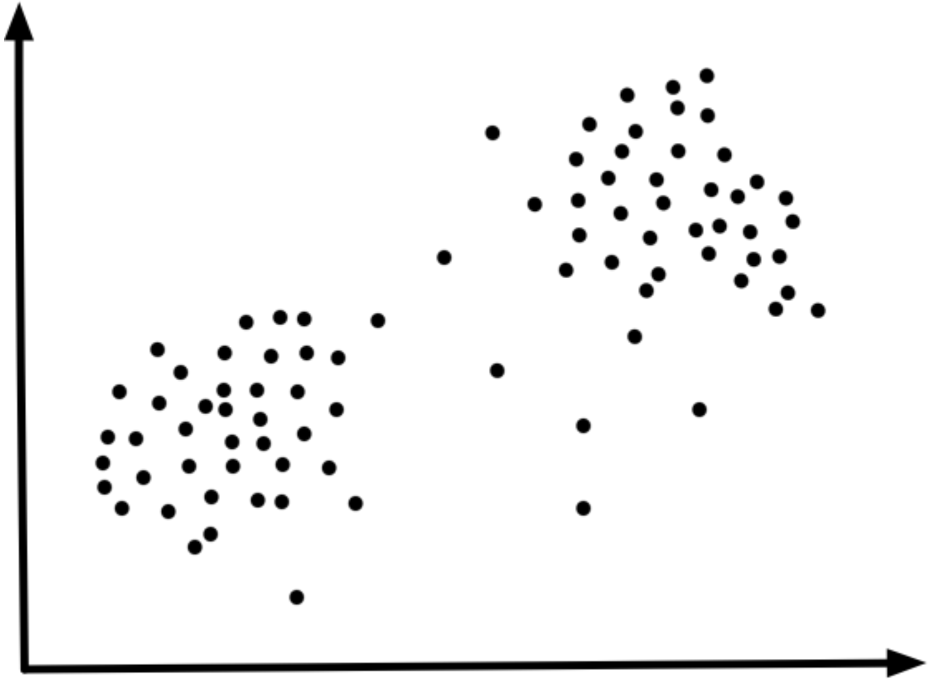
Unsupervised Learning

- **No Knowledge** of output
- data is **unlabeled**
- Self guided learning
- **Goal:** determine data patterns/grouping

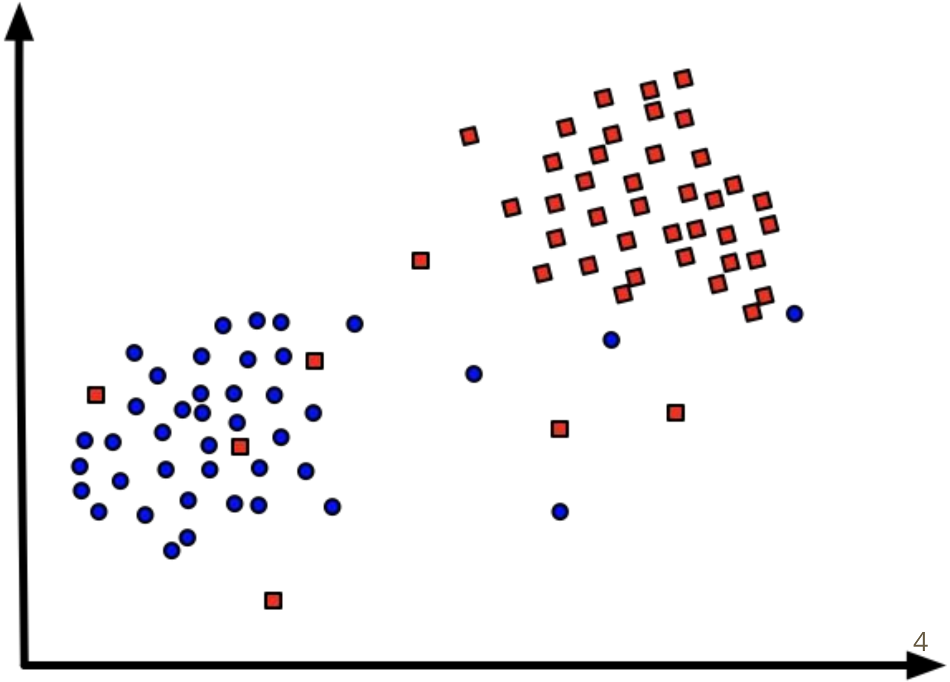
Supervised Learning

- **Knowledge** of output
- data is **labeled** with class or value
- **Goal:** predict value label or class label

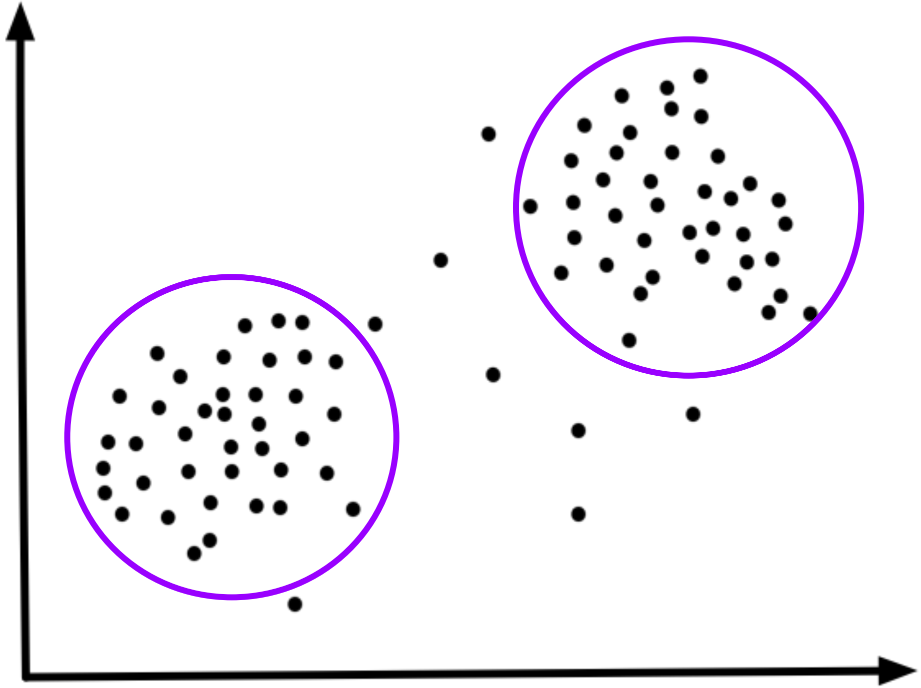
Unsupervised Learning



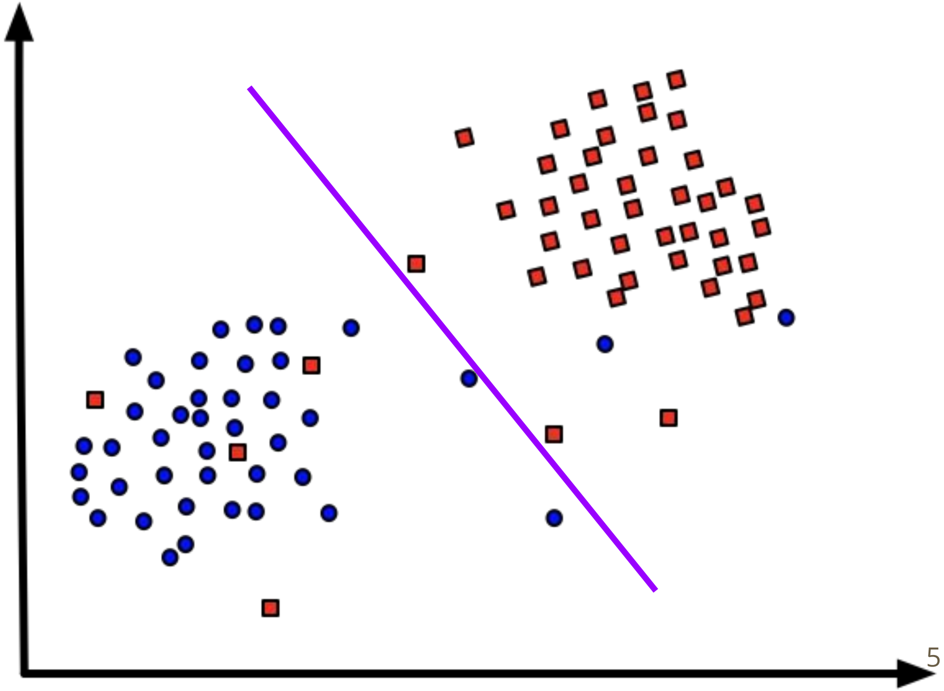
Supervised Learning

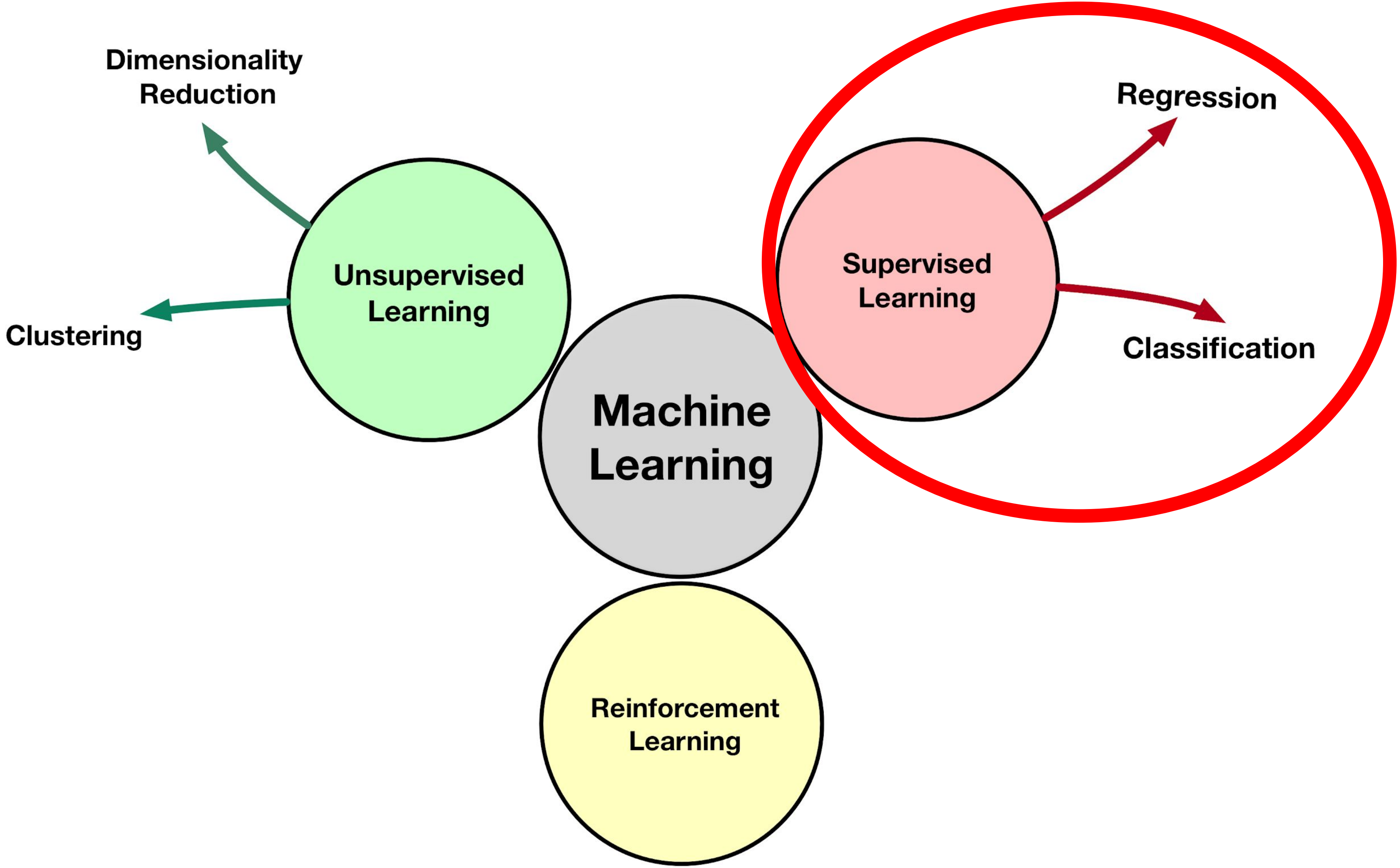


Unsupervised Learning



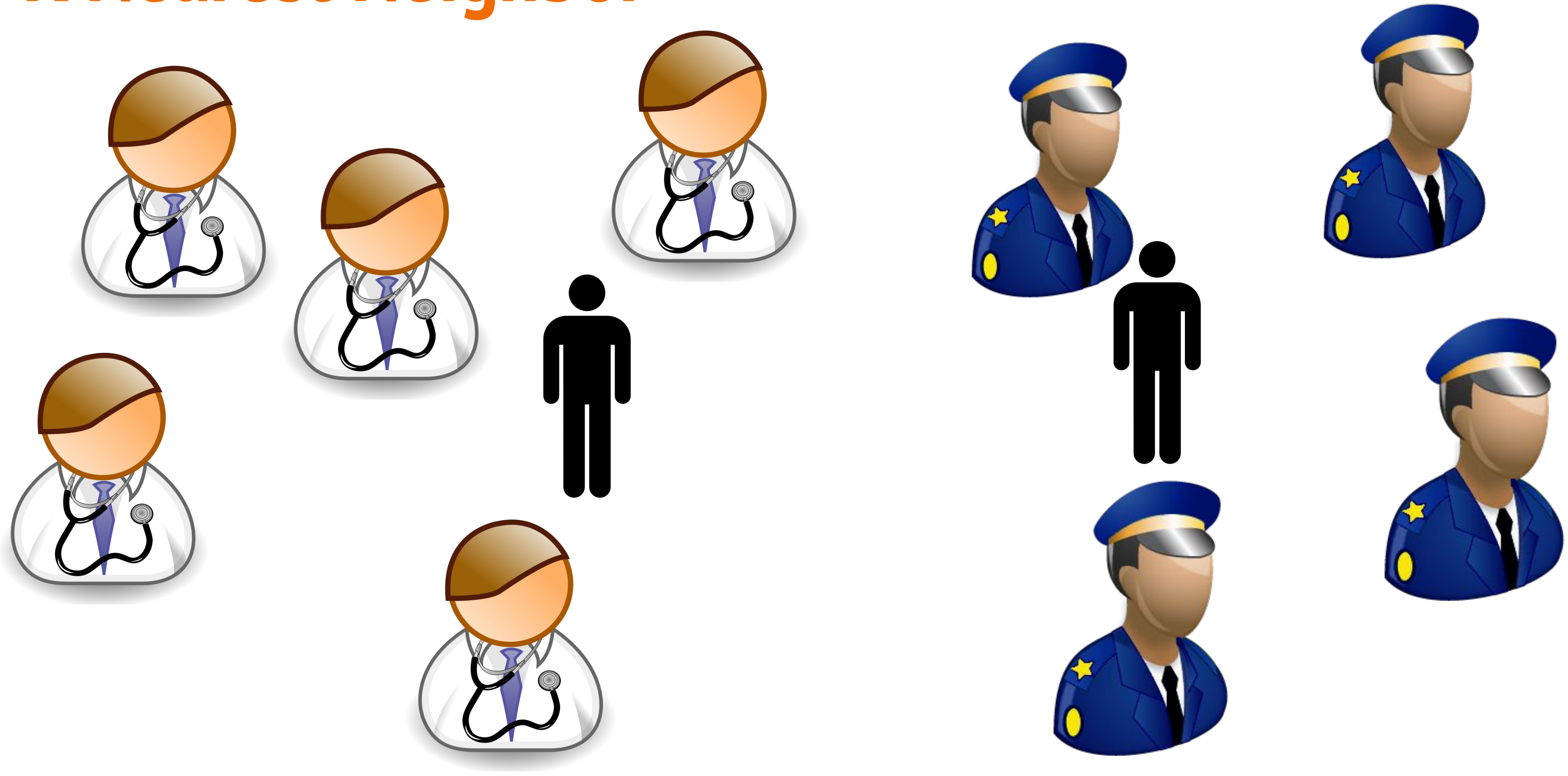
Supervised Learning




Dimensionality
Reduction
Unsupervised
Learning
Clustering
Machine
Learning
Supervised
Learning
Regression
Classification
Reinforcement
Learning

Machine Learning Algorithms

K Nearest Neighbor



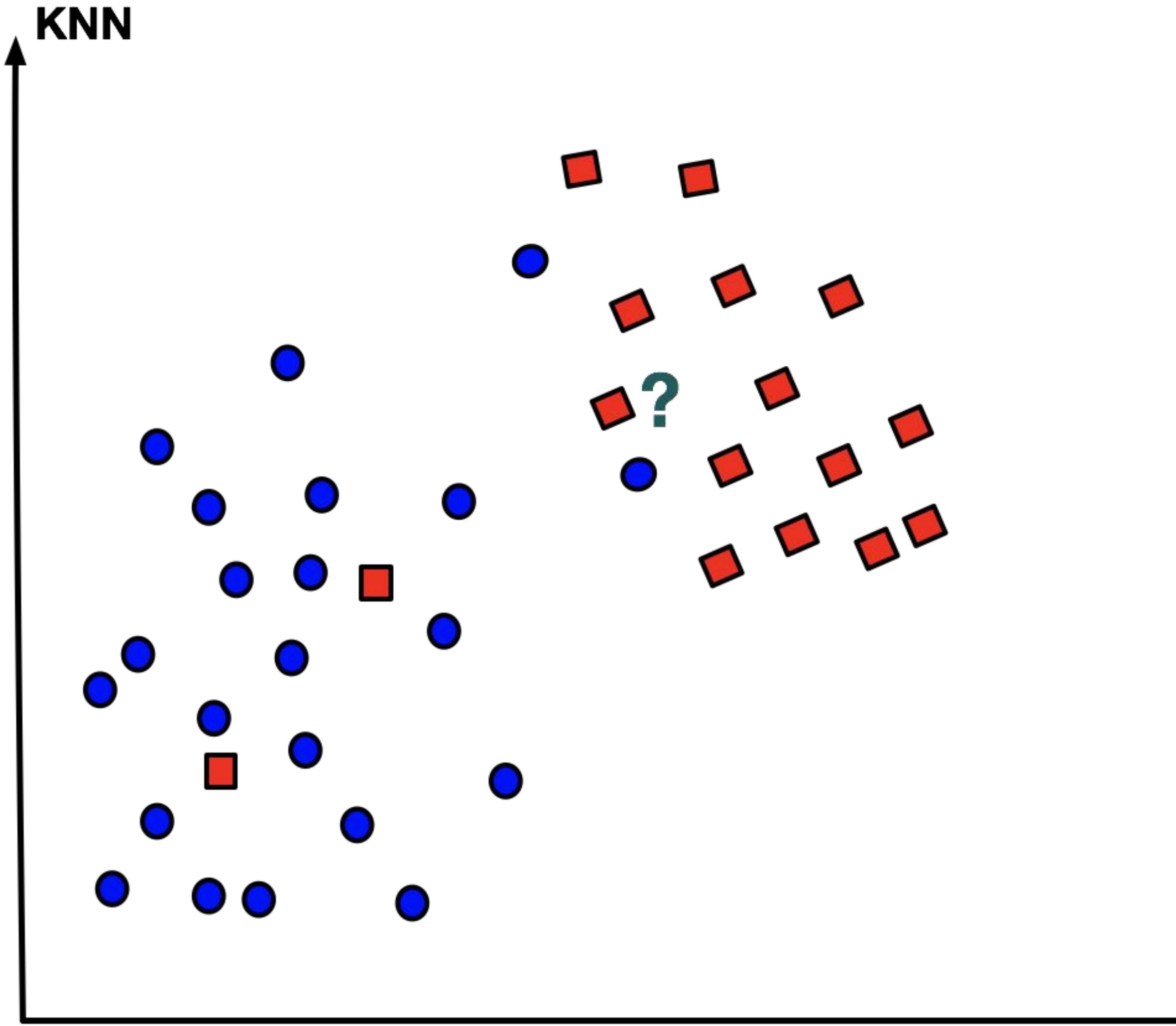
K Nearest Neighbor

- Can be used both for classification and regression.
- Uses **feature similarity** to predict values of any new data points.
- The output based on the majority vote (for classification)
- or mean (or median, for regression)

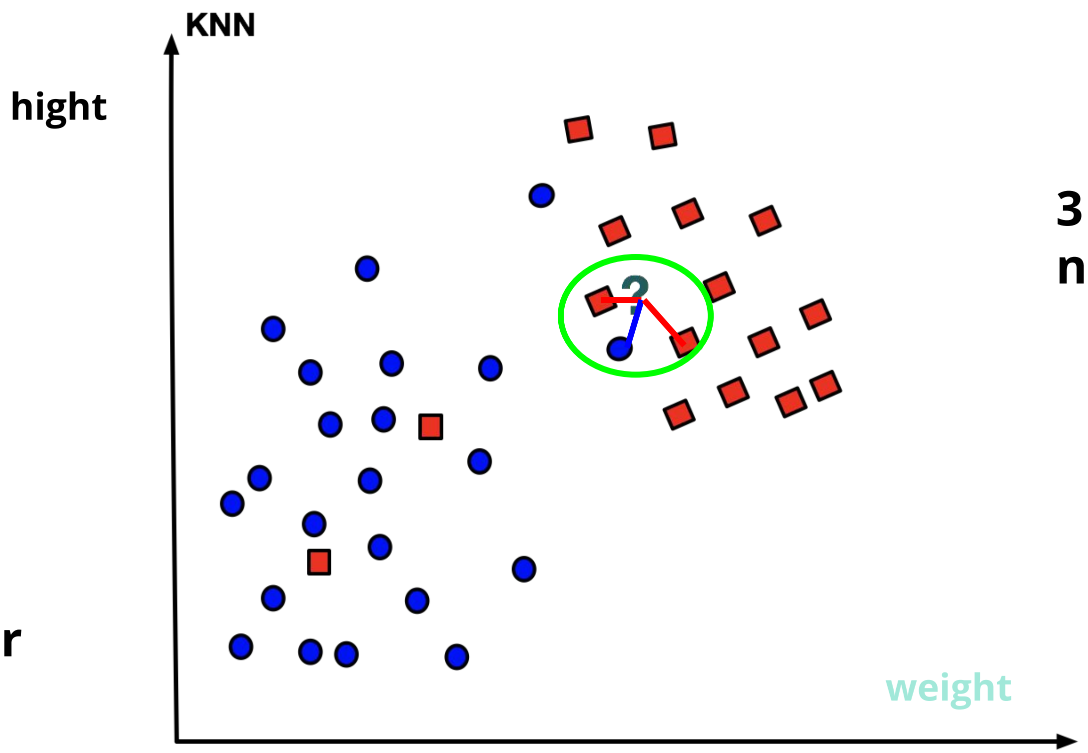
K Nearest Neighbor

Pick a value k

Use x's K-Nearest Neighbors to vote on what x's label should be.



K Nearest Neighbor

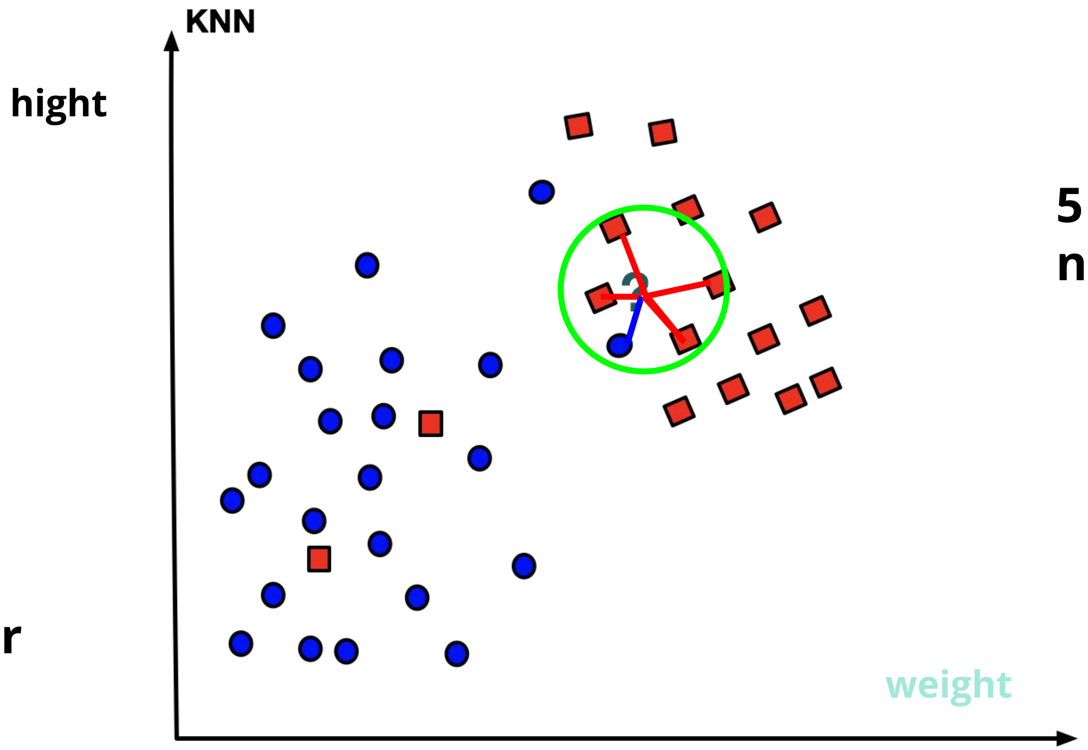


3 nearest neighbours

Gymnast

Basketball player

K Nearest Neighbor

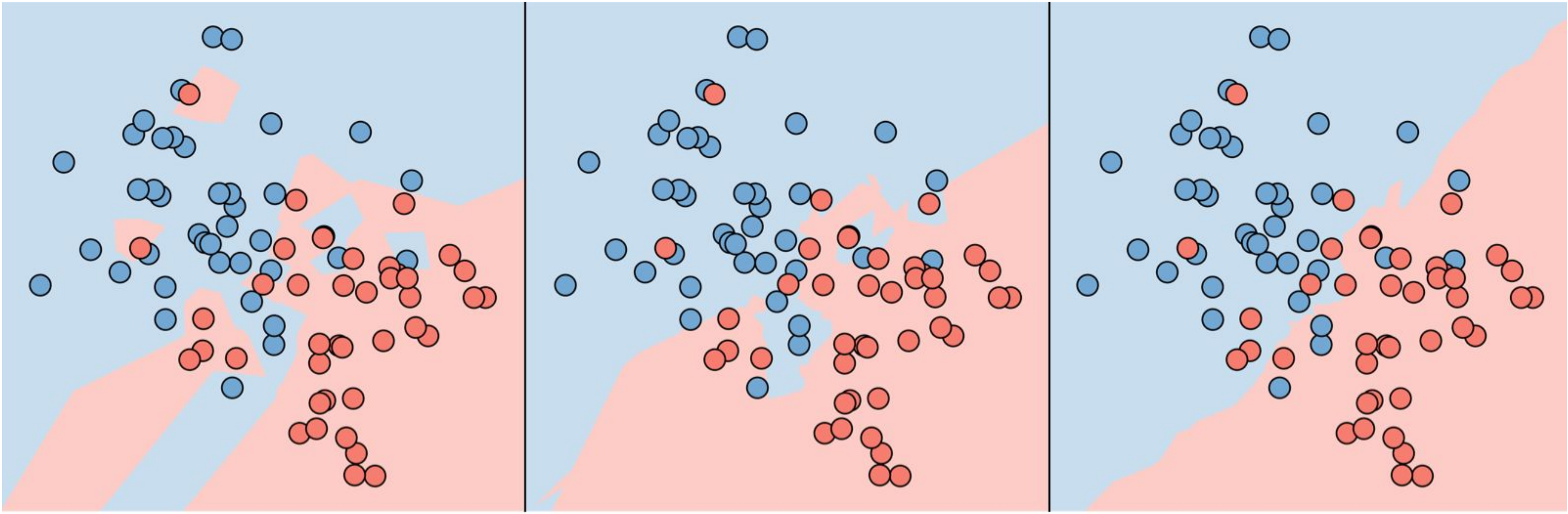


5 nearest neighbours

Gymnast

Basketball player

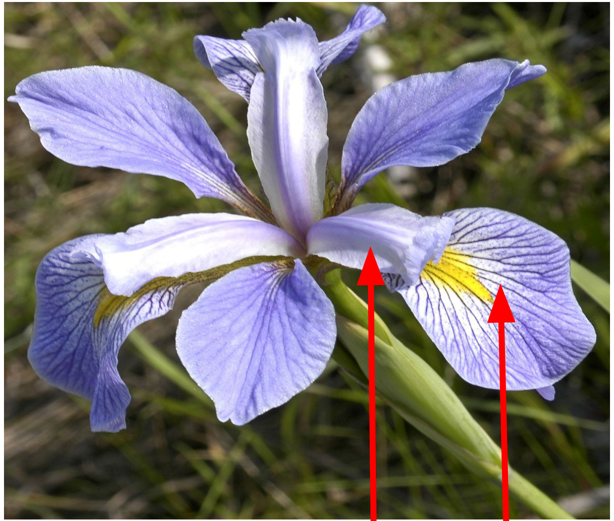
K Nearest Neighbor



$k = 1$ $k = 3$ $k = 11$

https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning

Iris DataSet



Iris virginica

Sepal

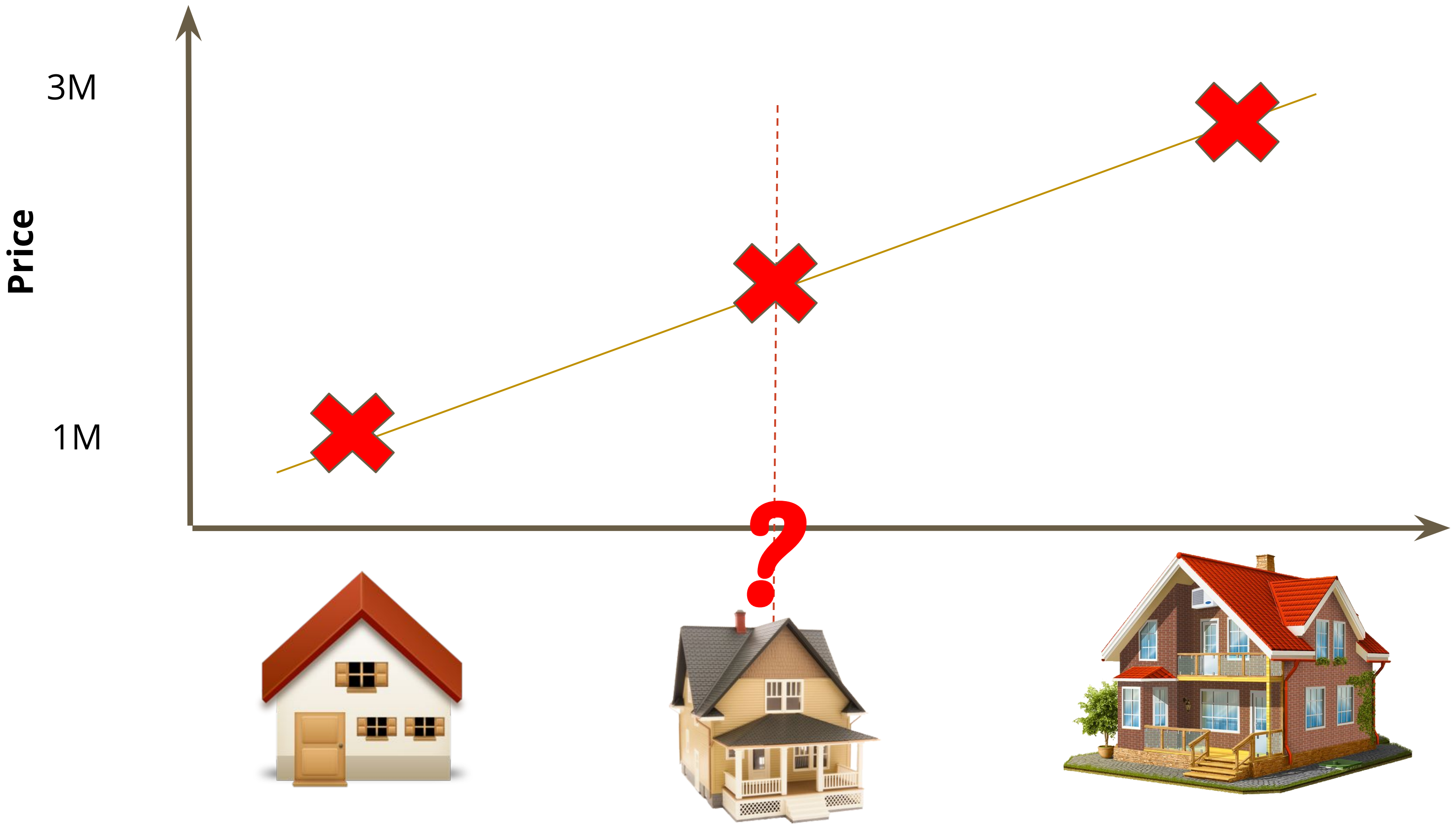
Petal

Iris setosa

Iris versicolor

https://ca.wikipedia.org/wiki/Conjunt_de_dades_de_flors_Iris

Linear Regression

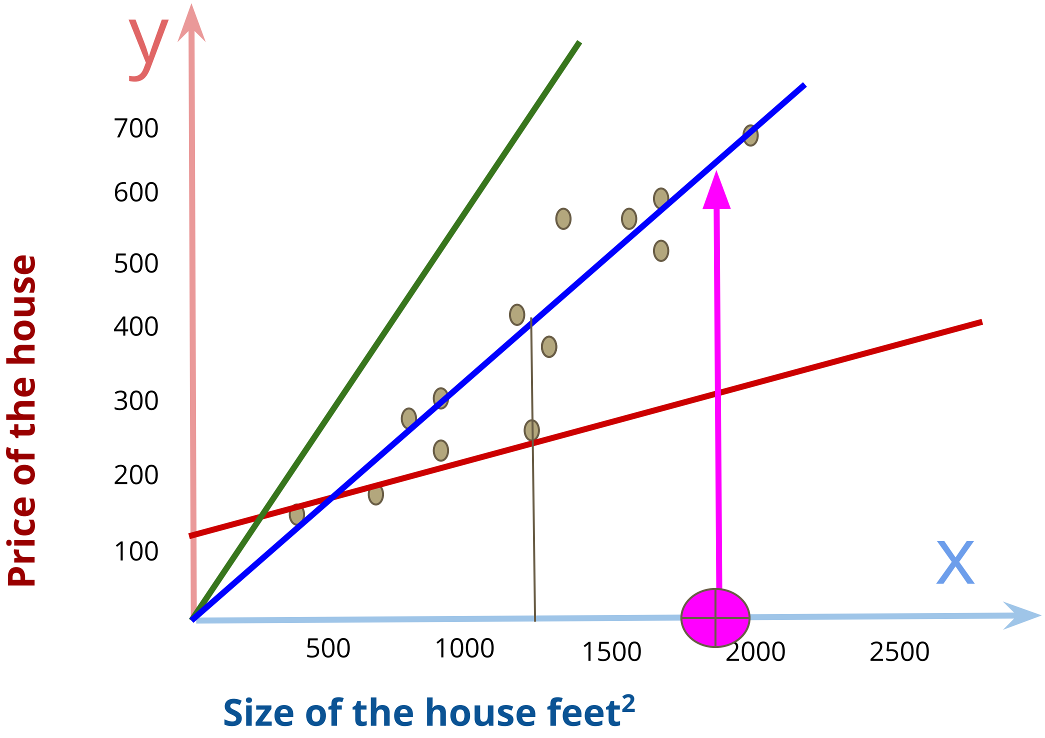

3M
Price
1M
?

Linear Regression

Linear regression is the simplest and most widely used statistical technique

A linear model expresses the target output value in terms of a sum of weighted input variables.

Linear Regression



$$f_i = w_0 x_i + b_0$$

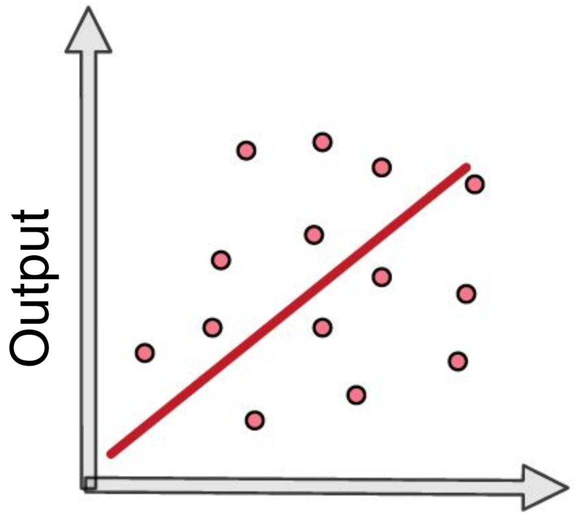
Mean squared error

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i.

Univariate versus Multivariate Modeling

Univariate

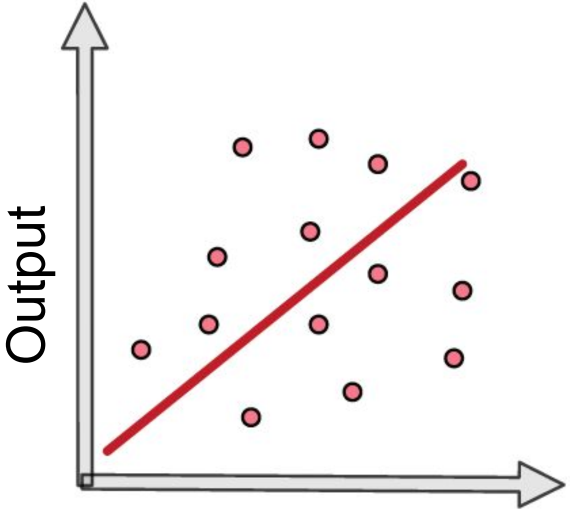


$f_i = w_0 x_{1i} + b_0$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

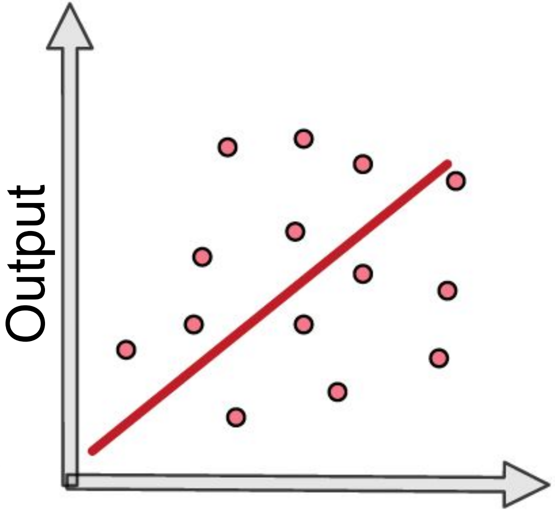
Univariate versus Multivariate Modeling

Univariate



$$f_i=w_0x_{1i}+b_0$$

Multivariate



$$f_i=w_0x_{1i}+b_0$$
$$+w_1x_{2i}+b_0$$
$$+.....$$
$$+w_1x_{mi}+b_0$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

Diabetes

Ten baseline variables:

age, sex, body mass index, average blood pressure, and six blood serum measurements

n = 442 diabetes patients

Target value:

A quantitative measure of disease progression one year after baseline.

Overfitting

Overfitting: Good performance on the training data, poor generalization to other data.

Underfitting: Poor performance on the training data and poor generalization to other data

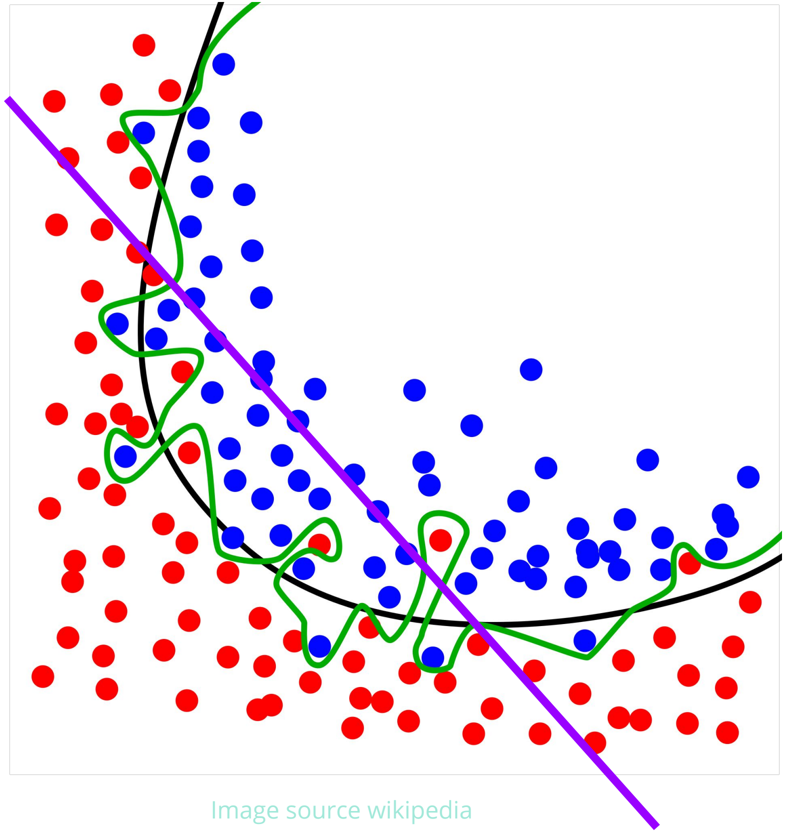
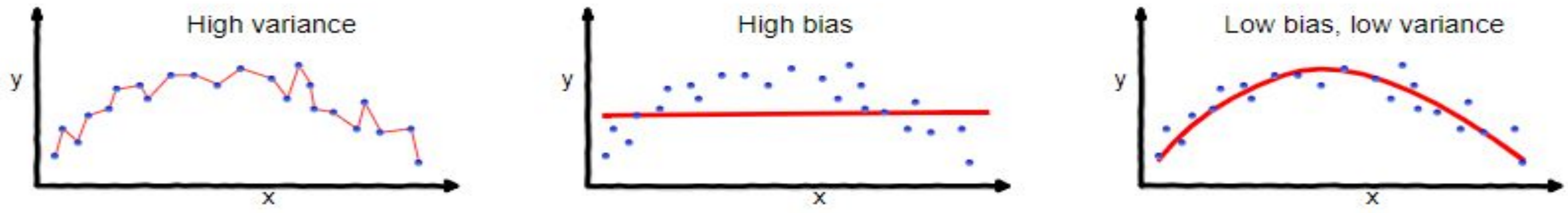


Image source wikipedia

Bias–Variance Tradeoff

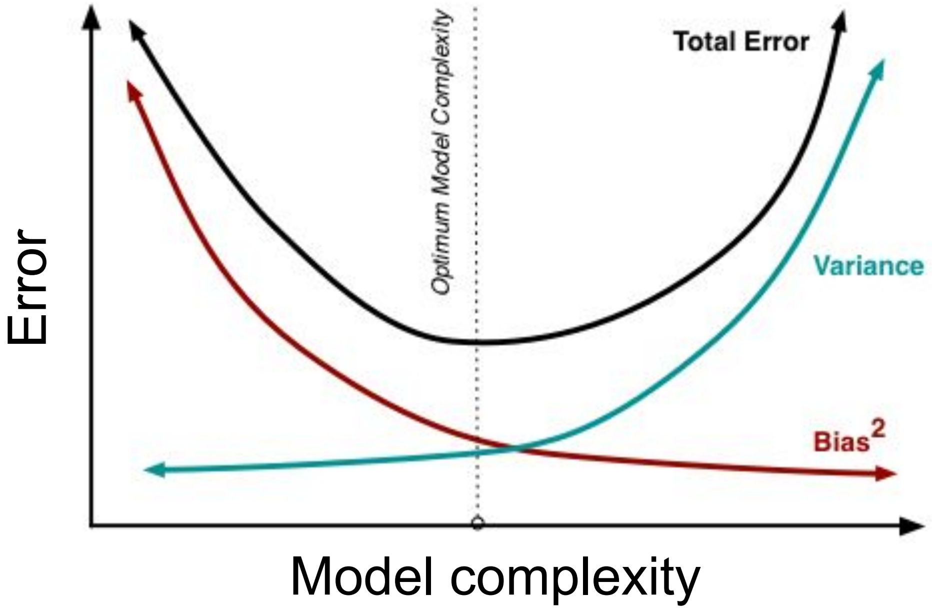


overfitting

underfitting

Good balance

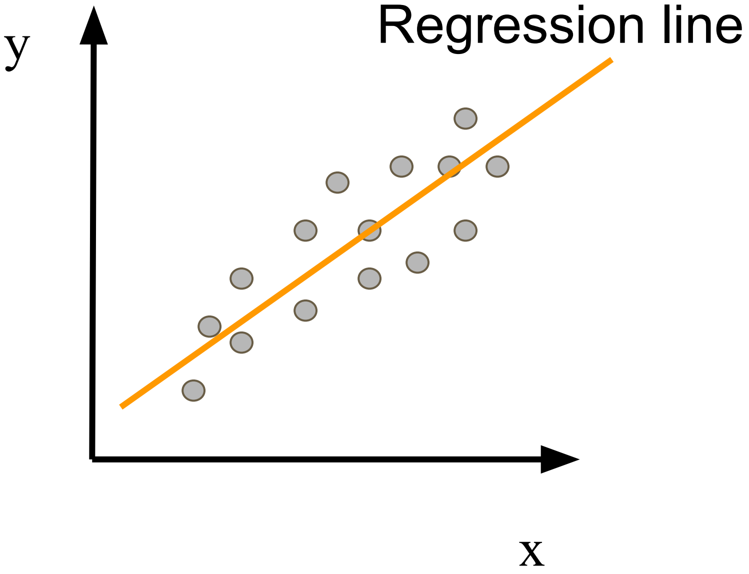
Bias–Variance Tradeoff



Logistic Regression

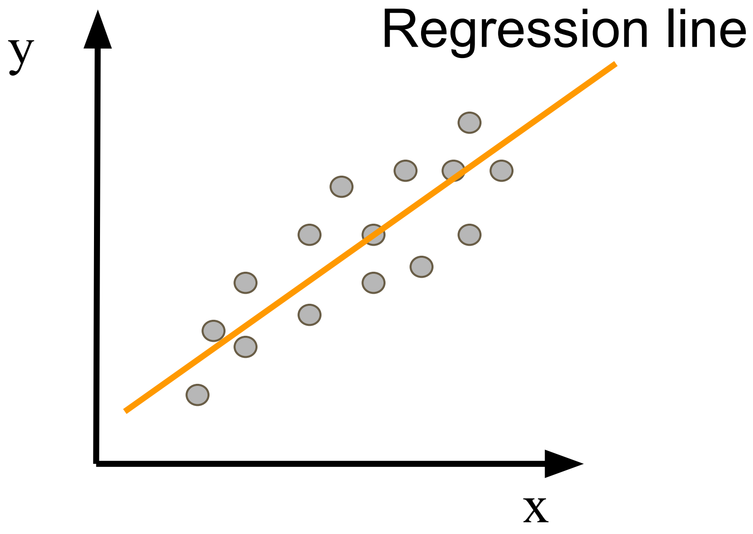
Linear versus Logistic Regression

Regression (linear regression)

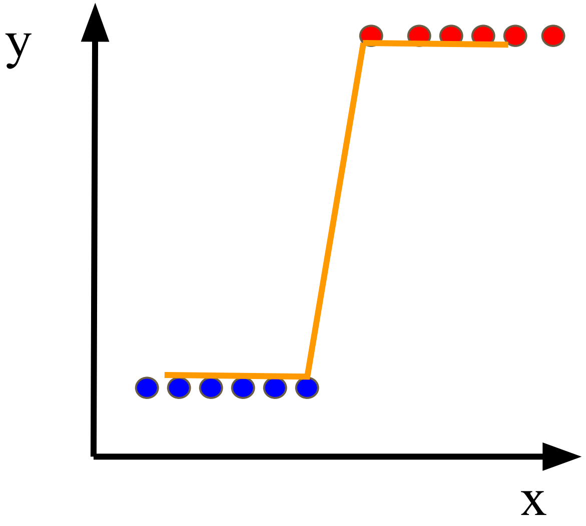


Linear versus Logistic Regression

Regression (linear regression)

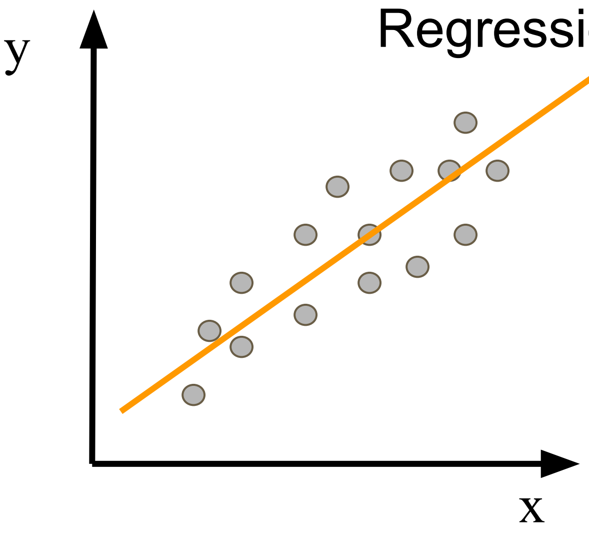


Classification (logistic regression)

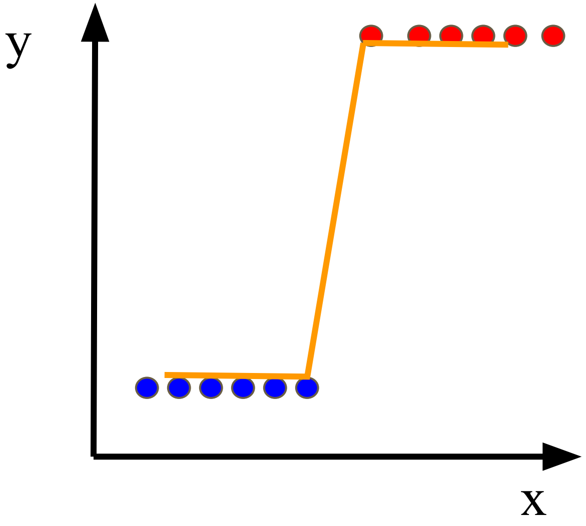


Linear versus Logistic Regression

Regression (linear regression)



Classification (logistic regression)



We need a smooth function that gives us this trend (Sigmoid Function)

Linear versus Logistic Regression

Linear regression

$$f_i = \Sigma_i w_i x_i + b_0$$

Logistic regression

$$f_i = \frac{1}{1+e^{\Sigma_i w_i x_i + b_0}}$$

W will be identified to minimize cost

$$\mathrm{Cost}(w)=\mathrm{function}(w,f_i,y_i)$$

Bayes rule and Naive Bayes classifier

Bayes' Theorem

provides a way that we can calculate the probability of a piece of data belonging to a given class

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

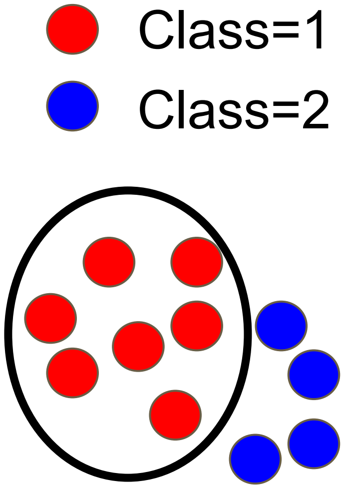
A, B	= events
$P(A \mid B)$	= probability of A given B is true
$P(B \mid A)$	= probability of B given A is true
$P(A), P(B)$	= the probabilities of A and B

Bayes’ theorem allows us to calculate conditional probabilities. It comes extremely handy because it enables us to use some knowledge that we already have

- 80% of the time, if she wins the race, she had a good breakfast. This is **P(breakfast|win)**.
- 60% of the time, she has a nice breakfast **P(breakfast)**.
- 20% of the time, she wins a race **P(win)**.

we can compute **P(win|breakfast)** to be 0.2 times 0.8, divided by 0.6 = 0.26.

What we know when training a model



$p(X_1 = x_1 \mid Class = 1)$

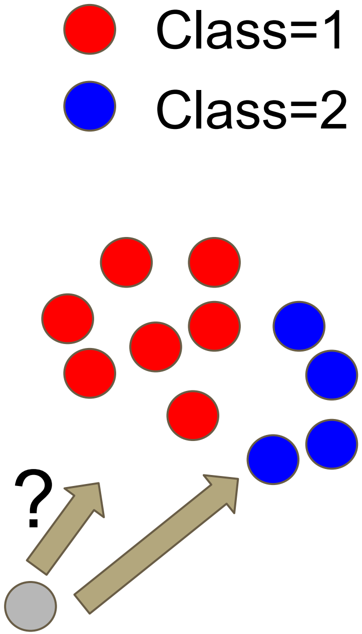
$p(X_2 = x_2 \mid Class = 1)$

$\vdots$

$p(X_m = x_m \mid Class = 1)$

What do we care about?

$$p(\text{Class}=1 \mid X_1=x_1, X_2=x_2, \ldots, X_m=x_m)=?$$



Bayes rule is useful to figure out the relationship

$$p(A|B)p(B)=p(B|A)p(A)$$

$$p(Class=1|X_1=x_1,X_2=x_2,\ldots,X_m=x_m)*$$

$$p(X_1=x_1,X_2=x_2,\ldots,X_m=x_m)=$$

$$p(X_1=x_1,X_2=x_2,\ldots,X_m=x_m|Class=1)p(Class=1)$$

The relationship looks complicated

$$\mathbf{WWW*p(X_1=x_1, X_2=x_2, \ldots, X_m=x_m) =}$$

$$\mathbf{p(X_1=x_1, X_2=x_2, \ldots, X_m=x_m | Class=1) p(Class=1)}$$

WWW: What We Want

p(Class=1) : easy to calculate

$$p(Class = i) = \frac{N_i}{\Sigma_i^C N_i}$$

Naive Bayes

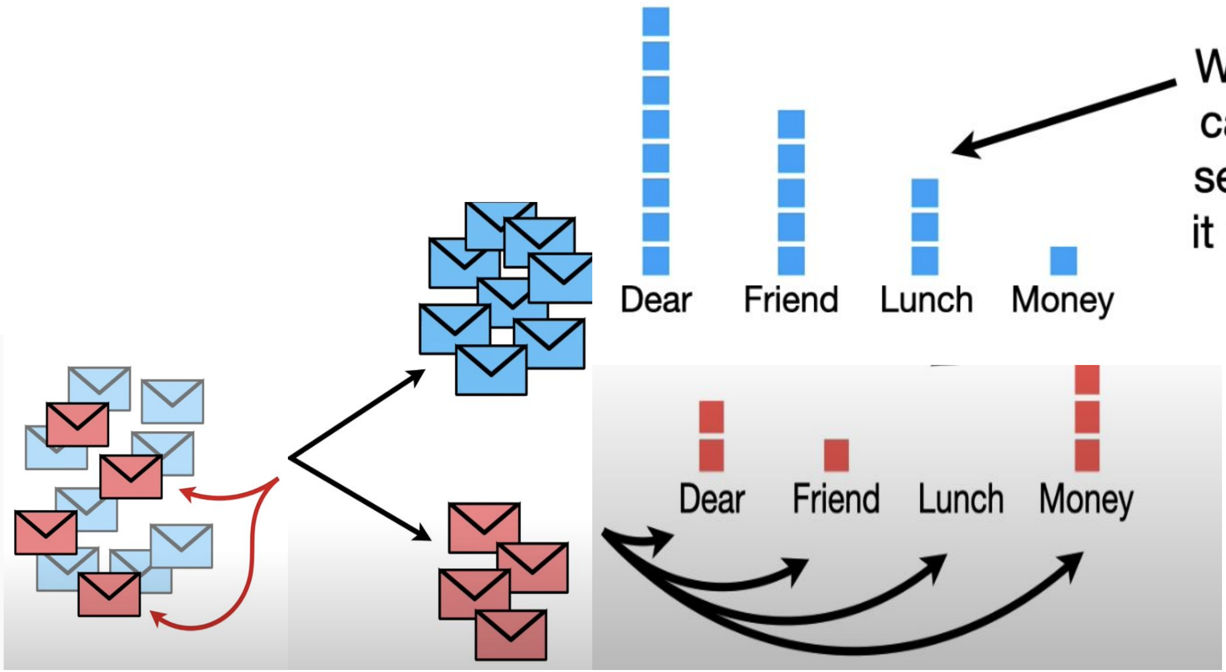
- simplify the calculation.
- Naive Bayes classifier is called ***Naive*** as it assumes each feature will independently contribute in prediction of a class for each data point

$$p(X1=x1, X2=x2, \ldots, Xm=xm) = p(X1=x1)\,p(X2=x2)\ldots p(Xm=xm)$$

$$p(X1=x1, X2=x2, \ldots, Xm=xm \mid Class=1) =$$

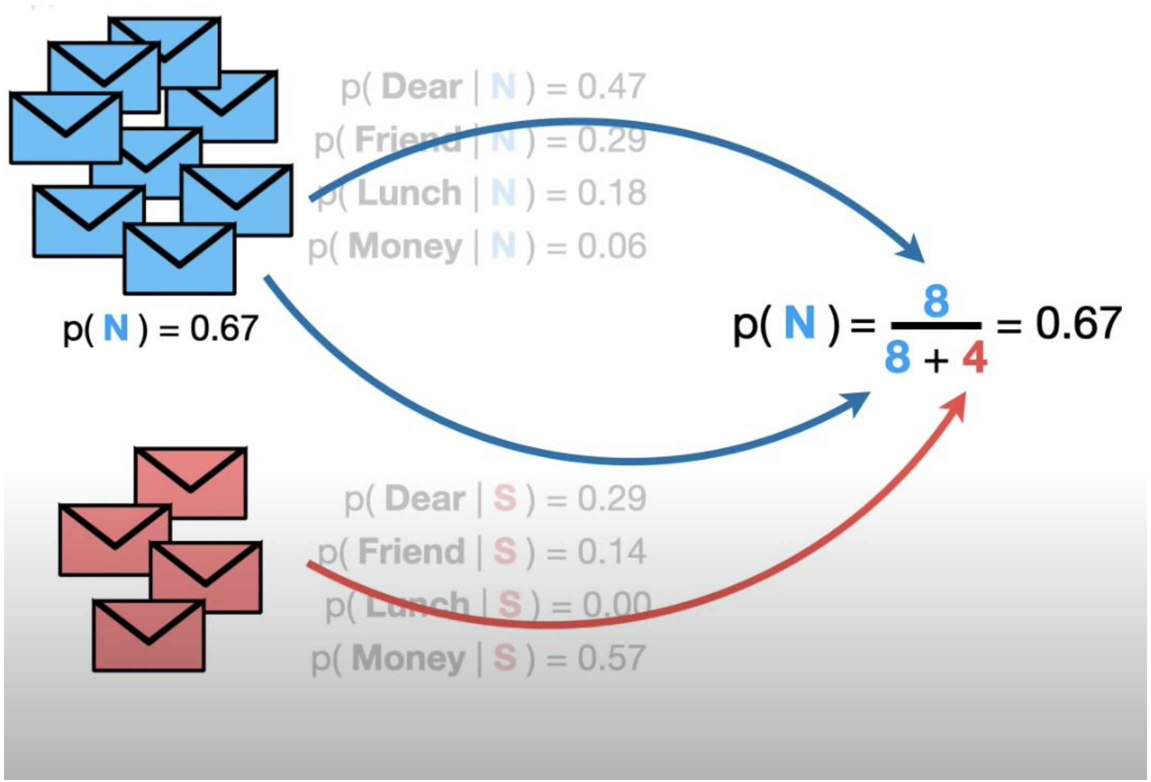
$$p(X1=x1 \mid Class=1)\,p(X2=x2 \mid Class=1)\ldots p(Xm=xm \mid Class=1)$$

Example Naive Bayes



We can use the histogram to calculate the probabilities of seeing each word, given that it was in a **normal message**.

Example Naive Bayes



Example Naive Bayes

p(**Dear** | **N**) = 0.47
p(**Friend** | **N**) = 0.29
p(**Lunch** | **N**) = 0.18
p(**Money** | **N**) = 0.06

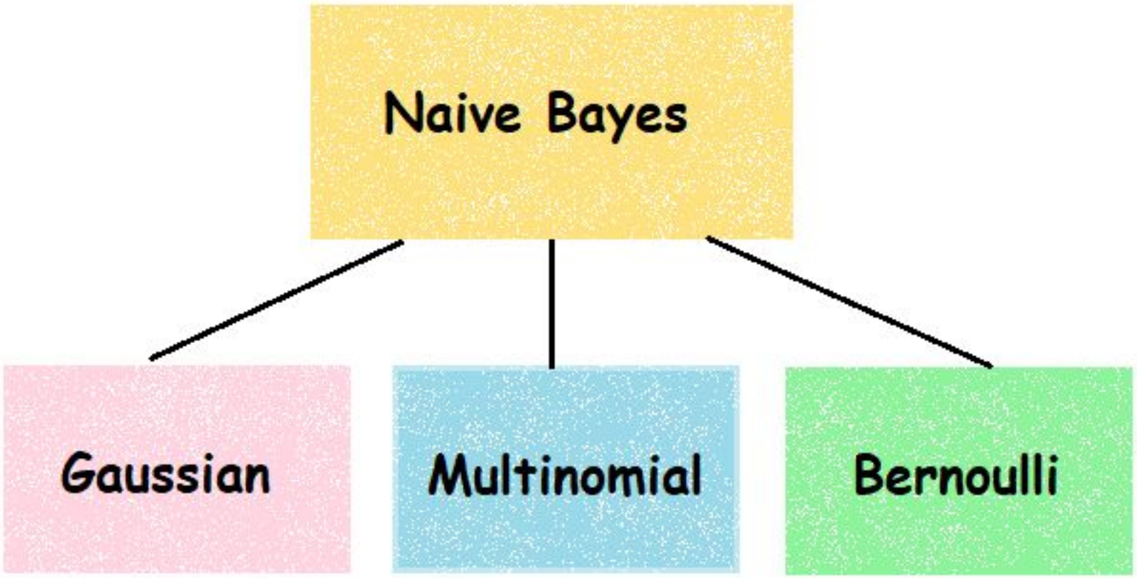
$$p(N) \times p(\textbf{Dear} \mid N) \times p(\textbf{Friend} \mid N) \boxed{= 0.09}$$

$$p(S) \times p(\textbf{Dear} \mid S) \times p(\textbf{Friend} \mid S) = 0.01$$

p(**Dear** | **S**) = 0.29
p(**Friend** | **S**) = 0.14
p(**Lunch** | **S**) = 0.00
p(**Money** | **S**) = 0.57

Dear Friend

Then we did the math and decided that **Dear Friend** was a **normal message** because **0.09** > **0.01**.


Naive Bayes
Gaussian
Multinomial
Bernoulli

Naive Bayes

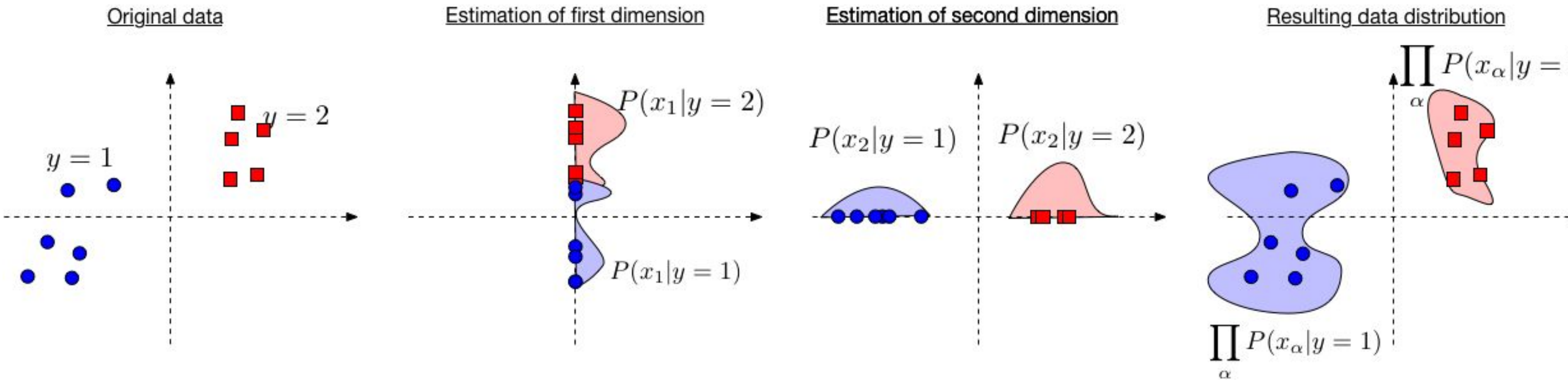
Bernoulli Naive Bayes : It assumes that all our features are binary:

- they take only two values.
- Means 0s can represent “word does not occur in the document” and 1s as "word occurs in the document" .

Multinomial Naive Bayes : Its is used when we have discrete data

Gaussian Naive Bayes : Because of the assumption of the normal distribution, Gaussian Naive Bayes is used in cases when all our features are continuous.

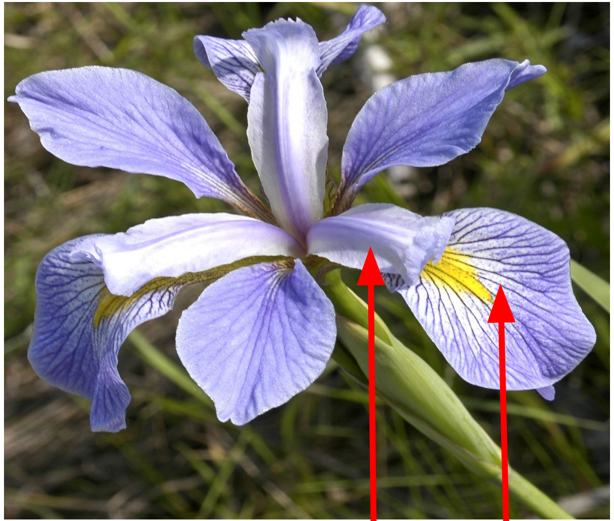
Gaussian Naive Bayes



https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html

Thanks

Iris DataSet



Iris virginica

Sepal

Petal

Iris setosa

Iris versicolor

https://ca.wikipedia.org/wiki/Conjunt_de_dades_de_flors_Iris

Diabetes

Ten baseline variables:

age, sex, body mass index, average blood pressure, and six blood serum measurements

n = 442 diabetes patients

Target value:

A quantitative measure of disease progression one year after baseline.

Breast cancer dataset

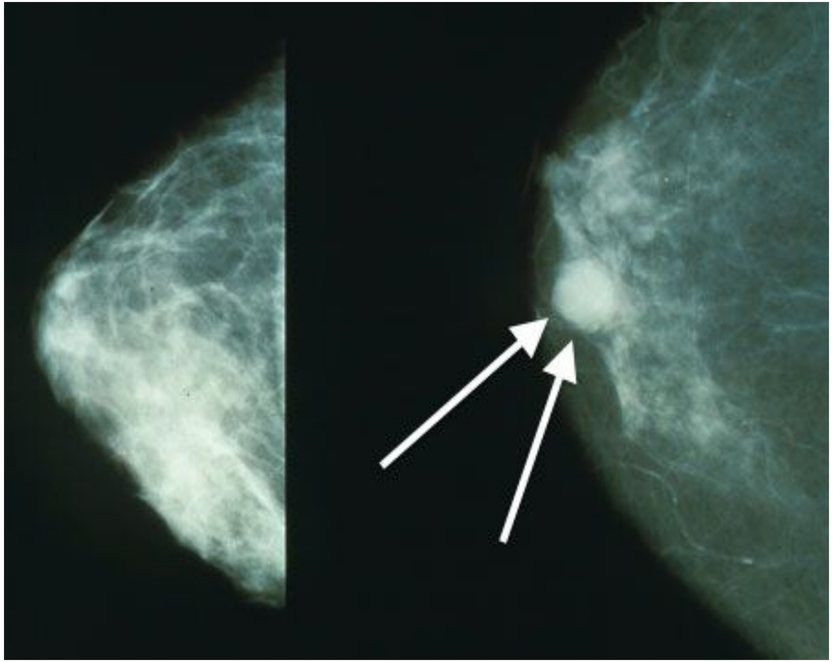
The breast cancer dataset is a classic and very easy binary classification dataset.

Features :

Computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

Target values:

Benign /Malignant



Extra useful information

Useful links

Installation instructions

- scikit-learn
- IPython

Data Sets

- scikit-learn DataSet

scikit-learn: machine learning in Python :

- https://scikit-learn.org/stable/

Useful cheat sheets:

- https://www.analyticsvidhya.com/blog/2017/02/top-28-cheat-sheets-for-machine-learning-data-science-probability-sql-big-data/