BACKGROUND OF DATA MINING

Week02

Essential Math for Data Mining

Linear Algebra

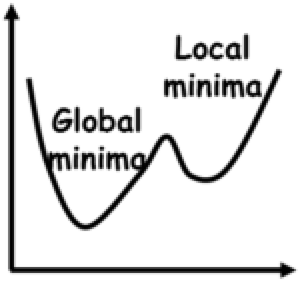
$\vec{a}$

$\vec{b}$

$$\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$$

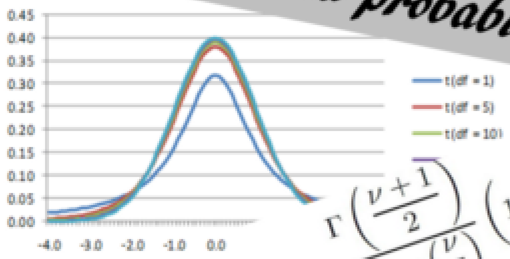
Multi-variable Calculus

$$\frac{dy}{dx} = 0$$



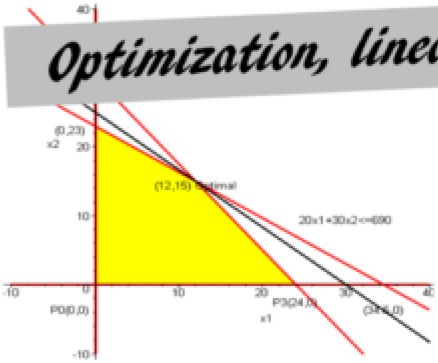
$$\frac{\partial}{\partial x}\left(f_{x,t}, g_{x,w}\right)$$

Statistics and probability



$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Optimization, linear programming

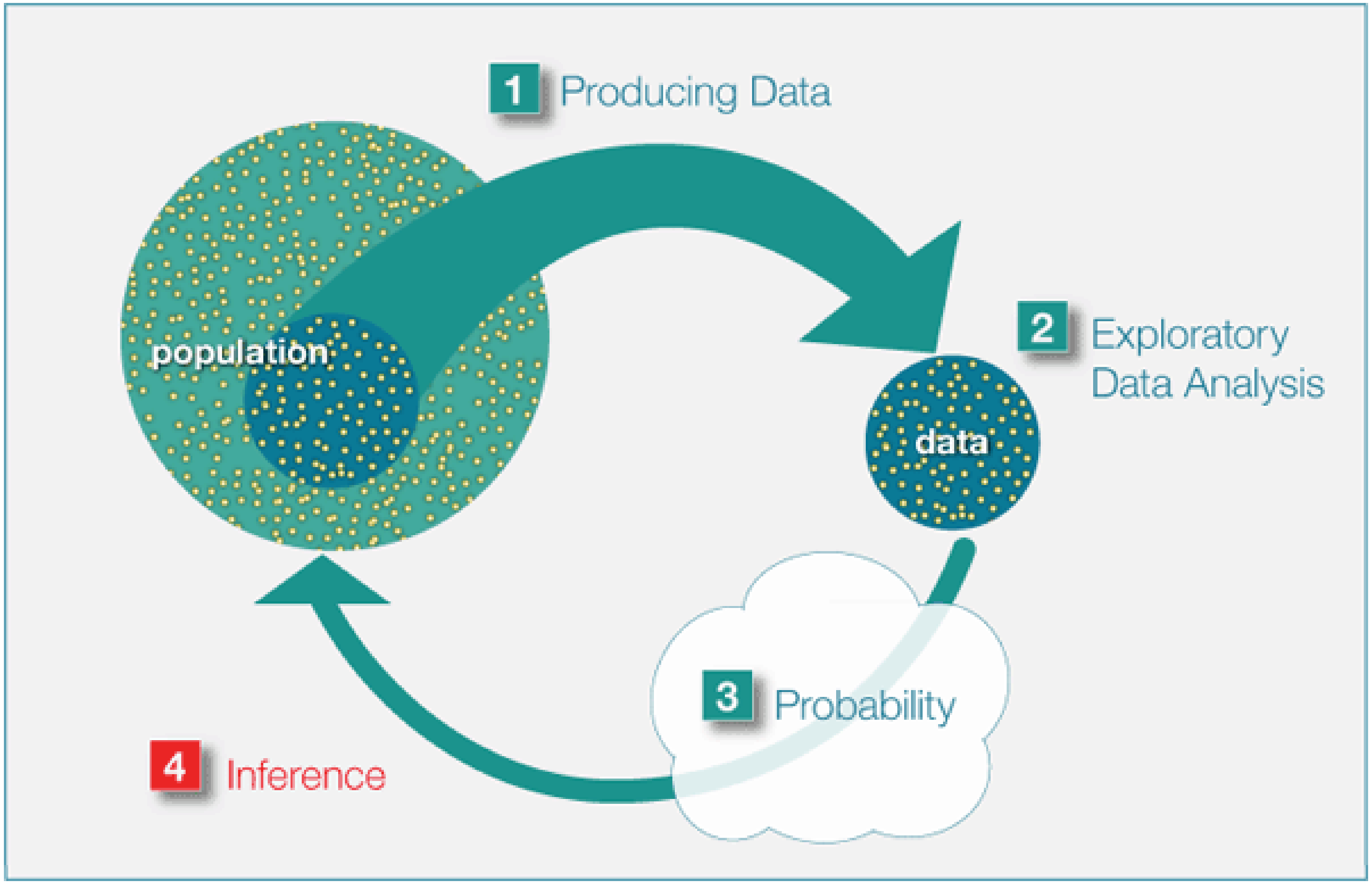


Essential Math for Data Mining: Statistics

- Two main branches of statistics
  - Descriptive statistics
    - Describe the basic features of data
    - Data summaries and descriptive statistics, central tendency, variance, covariance, correlation
  - Inferential statistics
    - Deduce properties of an underlying distribution of probability

- Probability
  - Sampling, measurement, error, random number generation
  - Basic probability: basic idea, expectation, probability calculus, Bayes theorem, conditional probability
  - Probability distribution functions — uniform, normal, binomial, chi-square, student's t-distribution, Central limit theorem

Essential Math for Data Mining: Statistics



Essential Math for Data Mining: Linear Algebra

- Linear algebra
  - The study of linear sets of equations and their transformation properties
  - Concern linear equations, linear functions and their representations in vector spaces and through matrices
  - Used in most areas of science and engineering, because it allows modeling many natural phenomena, and efficiently computing with such models

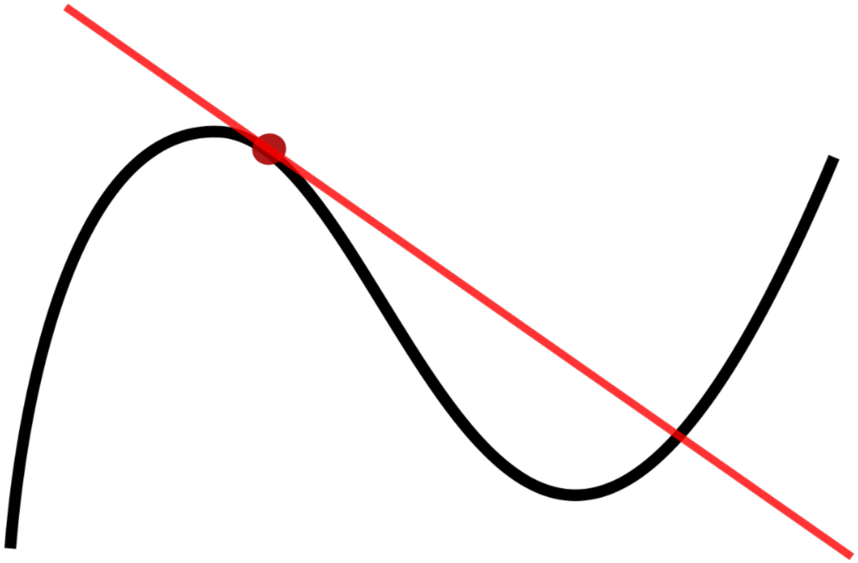
$$\begin{matrix} 3x + 5y = 7 \\ x - 2y = 6 \end{matrix} \quad \Rightarrow \quad \begin{bmatrix} 3 & 5 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$

$$\Downarrow$$

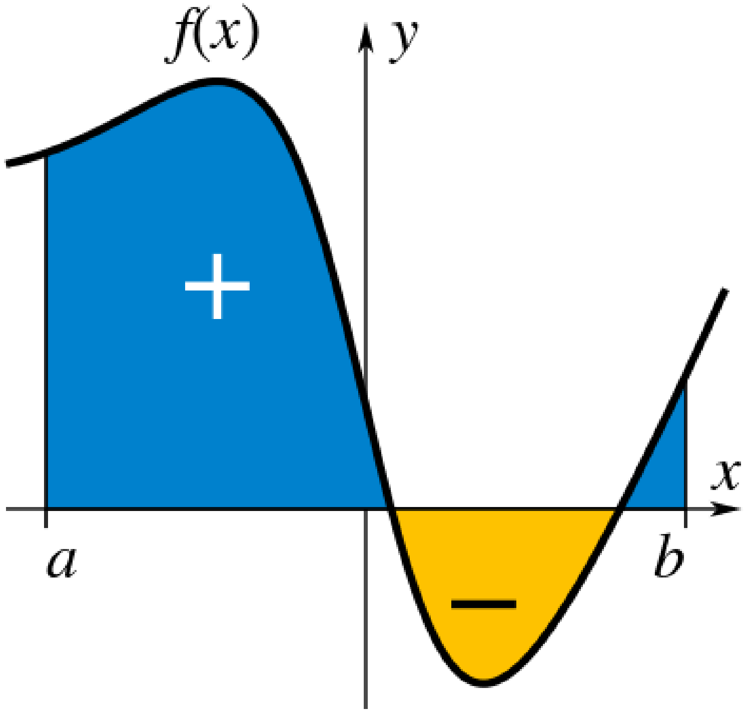
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 1 & -2 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$

Essential Math for Data Mining: Calculus

- Calculus
  - Branch of mathematics concerned with the calculation of instantaneous rates of change (differential calculus) and the summation of infinitely many small factors to determine some whole (integral calculus)



differential calculus



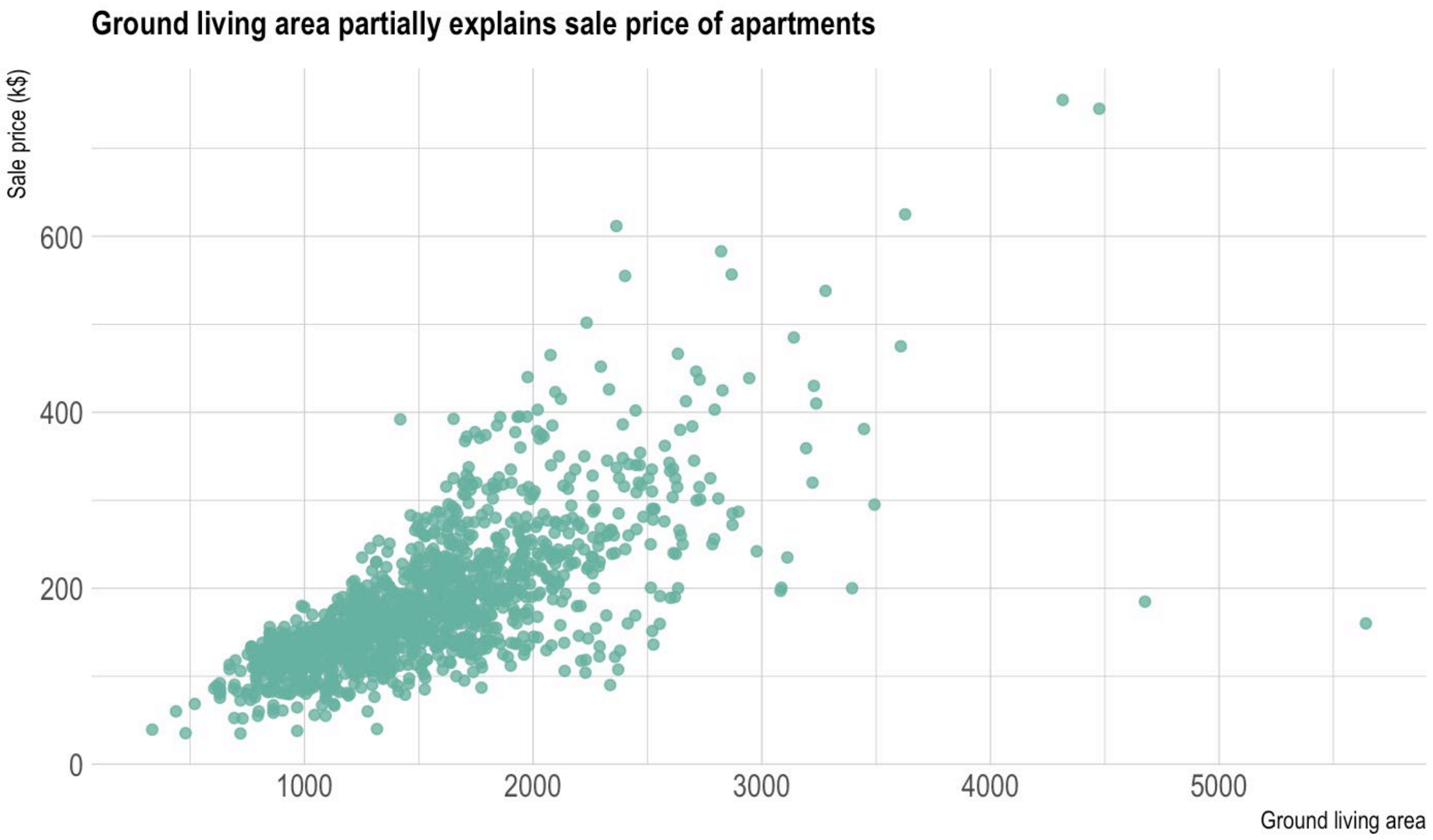
integral calculus

Essential Math for Data Mining: Optimization

- Optimization
  - Collection of mathematical principles and methods used for solving optimization problems
  - Optimization problem is the problem of fining the best solution from all feasible solutions
    - In the simplest case, an optimization problem consists of maximizing or minimizing a real function

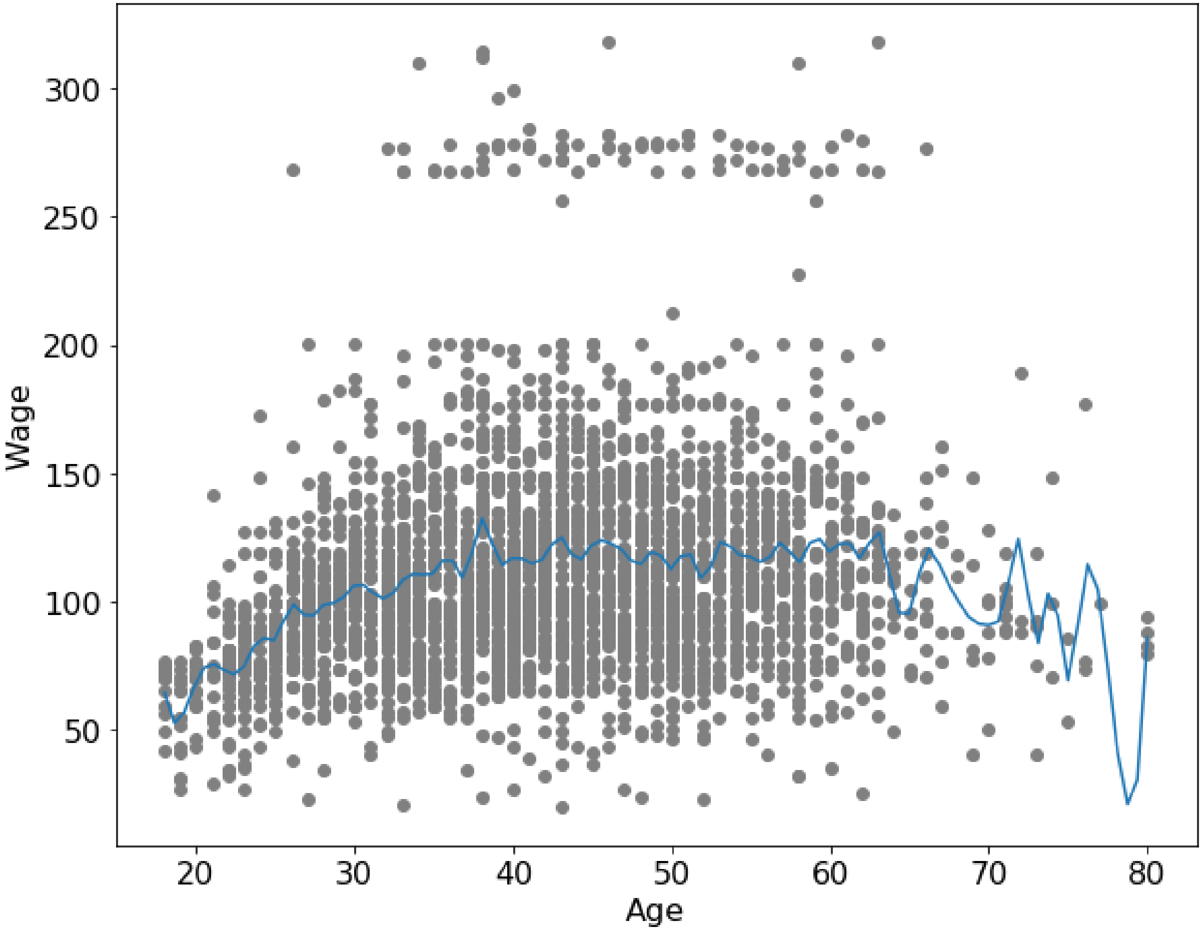
Statistics

- A vast set of tools for understanding data



Statistics

- A vast set of tools for understanding data

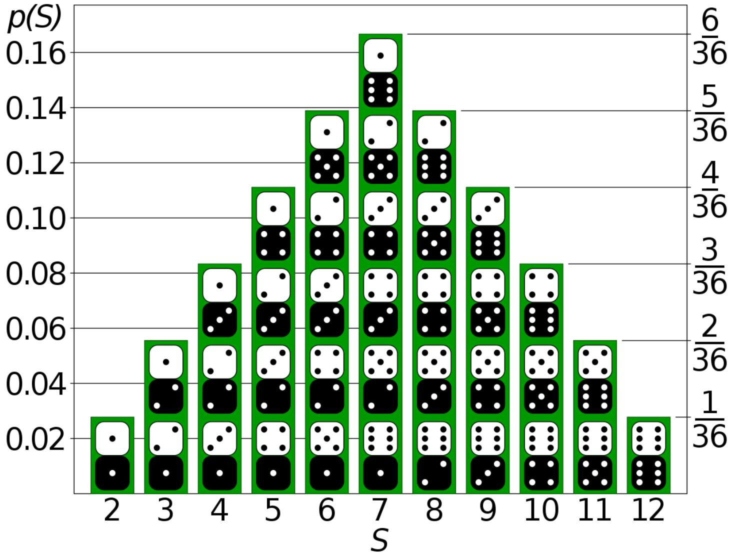


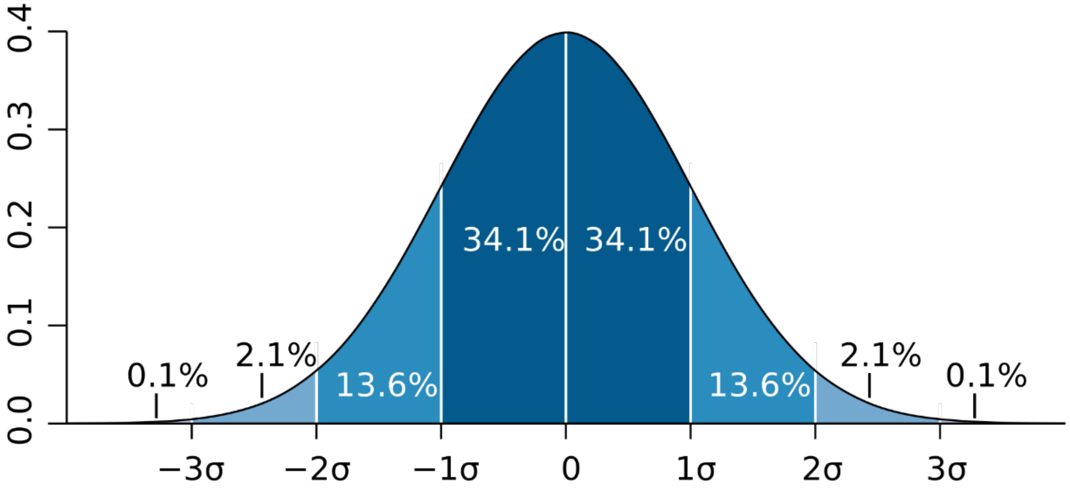
Statistics

- Descriptive statistics
  - A summary statistic that quantitatively describes or summarizes features of a collection of information
  - Univariate
    - Mean, Median, Mode
    - Variance, standard deviation, Percentile
    - Skewness, kurtosis
  - Bivariate or multivariate
    - Cross-tabulations and contingency tables
    - Graphical representation via scatterplots
    - Quantitative measures of dependence (covariance, correlation)

Statistics

- Probability distribution
  - A mathematical function that provides the probabilities of occurrence of different possible outcomes
    - The probabilities of occurrence of the specific observations





Discrete random variable → probability mass function

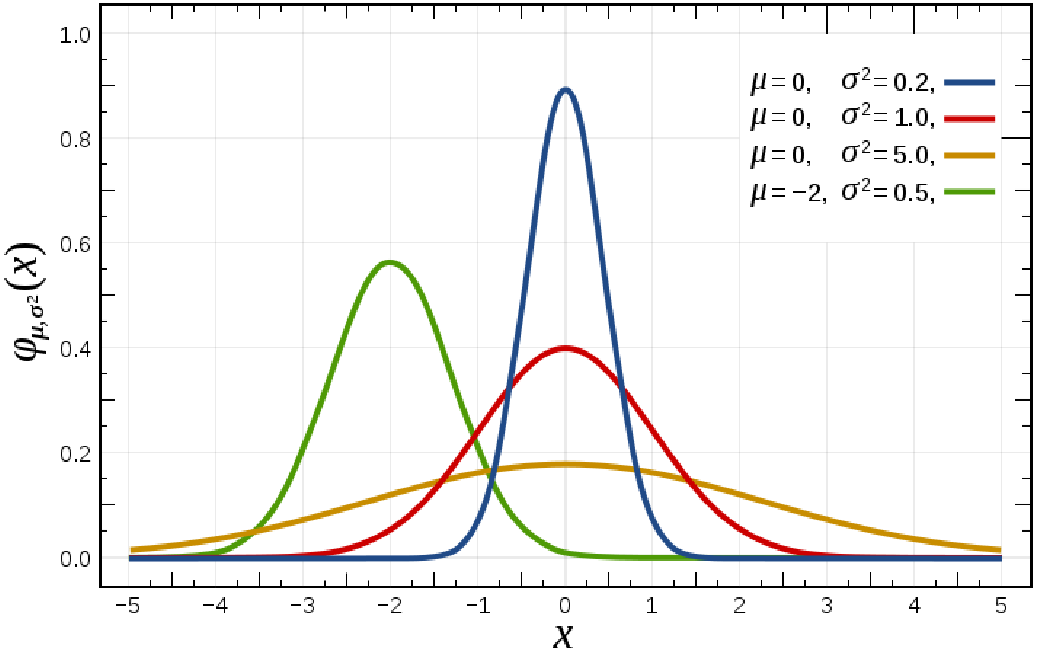
Continuous random variable → probability density function

Statistics: Normal distribution

- Normal (Gaussian) distribution
  - Very common continuous probability distribution
  - Bell-shaped

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

    - μ: mean
    - σ: standard deviation



Statistics: Student's t-distribution

- Student's t-distribution (t-distribution)
  - Continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the **sample size is small** and **population standard deviation is unknown**

- Let $X_1, \ldots, X_n$ be independent and identically distributed (iid) as $N(\mu, \sigma^2)$
  - Sample mean

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

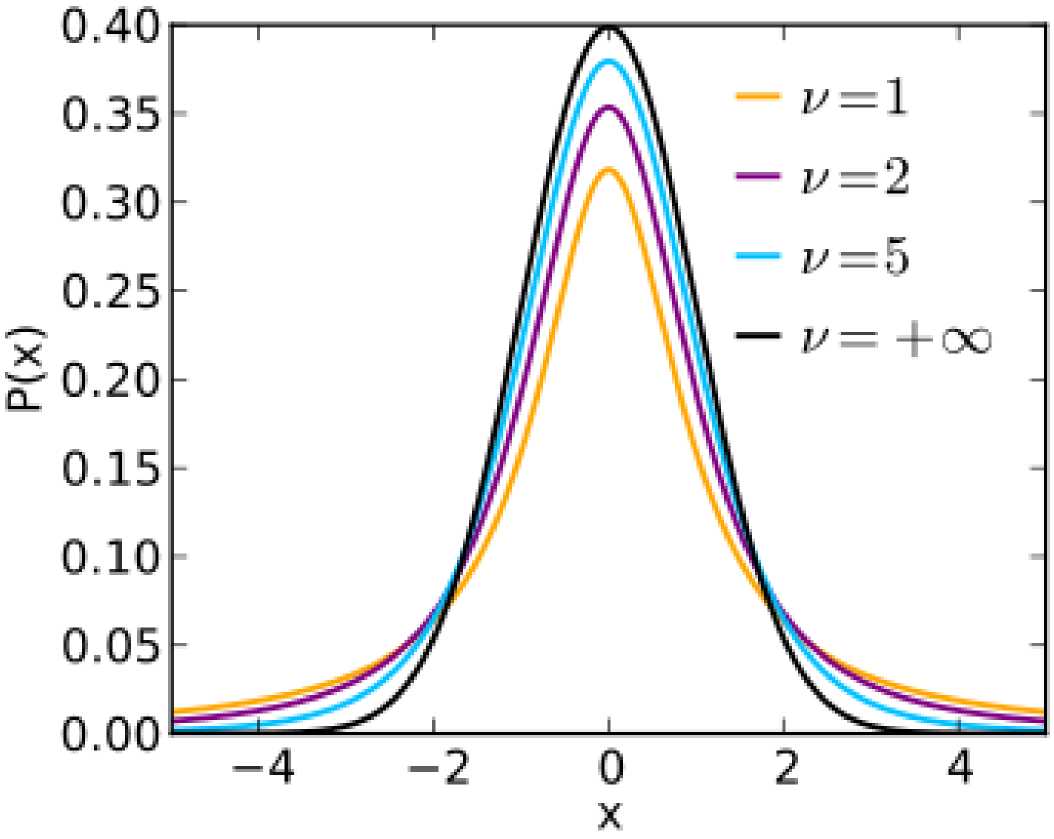
  - Sample variance

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

  - The random variable $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$ has a standard normal distribution
  - The random variable $\frac{\bar{X}-\mu}{S/\sqrt{n}}$ has a Student's t-distribution with $n-1$ degrees of freedom

Statistics: Student's t-distribution

- □ The probability density function of t-distribution with varying degree of freedom



Statistics: Student's t-distribution

- Probability density function

$$f(x;\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

  - ν: degree of freedom
  - Γ: gamma function

$$\Gamma(n) = (n-1)! \quad \text{if } n \text{ is positive integer}$$

$$\Gamma(z) = \int_0^{-\infty} x^{z-1}e^{-x}dx$$

Statistics: Chi-squared distribution

- Chi-squared distribution (χ^2)
  - The distribution of a sum of the squares of k independent standard normal random variables

- Let $X_1, \ldots, X_k$ be independent, standard normal random variables

$$Y = \sum_{i=1}^{k} X_i^2$$

is distributed according to the chi-squared distribution with k degrees of freedom

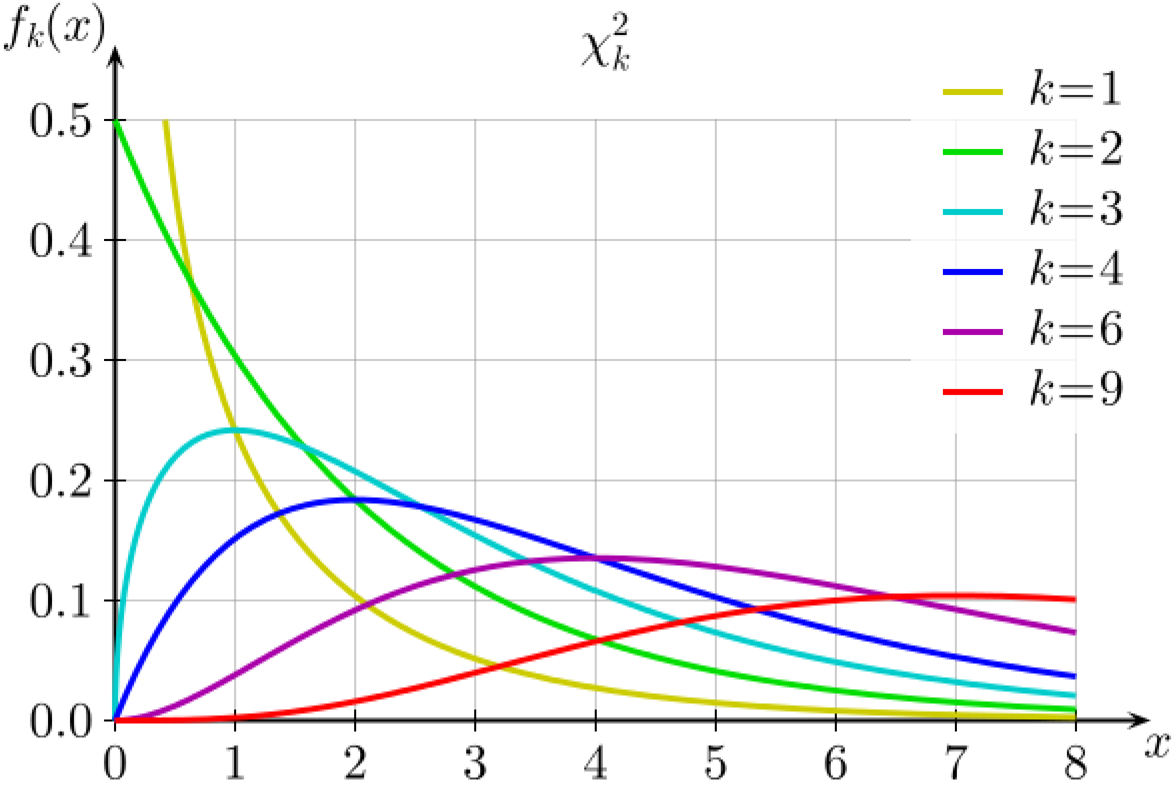
$$Y \sim \chi^2(k) \;\; or \; Y \sim \chi_k^2$$

- Probability density function

$$f(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$$

Statistics: Chi-squared distribution

- The probability density function of chi-squared distribution with varying degree of freedom



Statistics: F-distribution

- F-distribution
  - A random variate of the F-distribution with parameters d_1 and d_2 arises as the ratio of two appropriately scaled chi-squared variates
- Let X_1 and X_2 be two independent random variables and $X_1 \sim \chi^2(d_1)$ and $X_2 \sim \chi^2(d_2)$

$$Y = \frac{X_1/d_1}{X_2/d_2}$$

is distributed according to the F-distribution with d_1 and d_2 degrees of freedom

$$Y \sim F(d_1, d_2)$$

- Probability density function

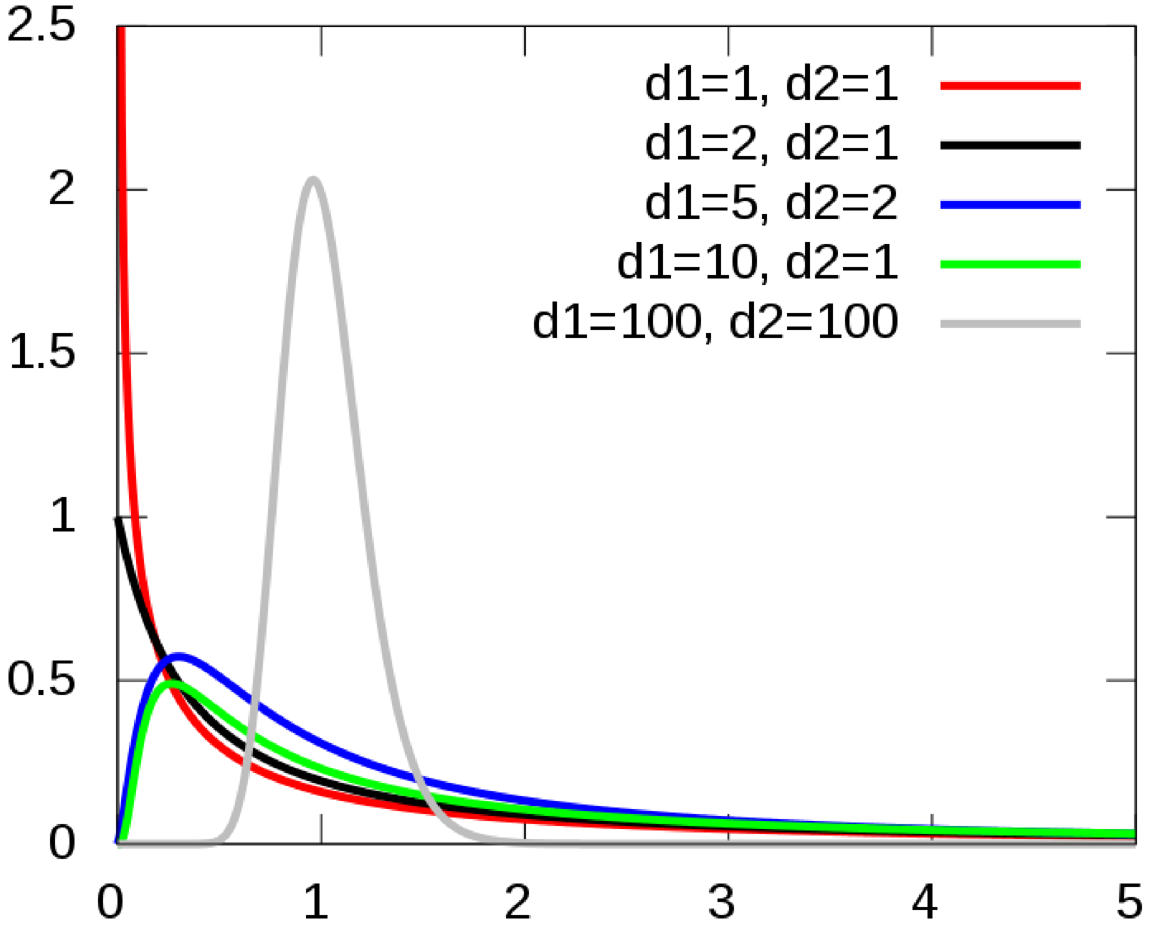
$$f(x; d_1, d_2) = \frac{\sqrt{\frac{(d_1x)^{d1}d_2^{d_2}}{(d_1x + d_2)^{d_1+d_2}}}}{x\mathrm{B}\left(\frac{d_1}{2}, \frac{d_2}{2}\right)}$$

  - B: beta function

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$$

Statistics: F-distribution

- □ The probability density function of F-distribution with varying degree of freedom



Linear Algebra

- Linear Algebra
  - Basic properties of matrix and vectors — scalar multiplication, linear transformation, transpose, conjugate, rank, determinant
  - Inner and outer products, matrix multiplication rule and various algorithms, matrix inverse
  - Special matrices — square matrix, identity matrix, triangular matrix, idea about sparse and dense matrix, unit vectors, symmetric matrix, Hermitian, skew-Hermitian and unitary matrices
  - Gaussian/Gauss-Jordan elimination, solving Ax=b linear system of equation
  - Matrix factorization and decomposition
  - Vector space, basis, span, orthogonality, orthonormality, linear least square
  - Eigenvalues, eigenvectors, and diagonalization, singular value decomposition (SVD)

Linear Algebra

- Linear algebra is the study of vectors and linear functions
  - Scalar
    - A scalar is a number
  - Vector
    - A vector is a list of numbers

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

  - Matrix
    - A matric is also a collection of numbers
    - The difference is that a matrix is a table of numbers rather than a list

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

  - Linear equation

$$a_1 x_1 + \cdots + a_n x_n = b$$

  - Linear function

$$(x_1, \ldots, x_n) \longmapsto a_1 x_1 + \cdots + a_n x_n$$

Linear Algebra

- Vectors
  - Addition

$$\mathbf{v} + \mathbf{w}$$

    - Example

$$\mathbf{v} + \mathbf{w} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

  - Linear combination

$$a\mathbf{v} + b\mathbf{w}$$

    - Example

$$3\mathbf{v} + 4\mathbf{w} = 3\begin{bmatrix} 1 \\ 2 \end{bmatrix} + 4\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 15 \\ 22 \end{bmatrix}$$

Linear Algebra

- Vectors
  - Transpose
    - column vector ↔ row vector
    - Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \to \mathbf{v}^T = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

  - Dot product, inner product

$$\mathbf{v} \cdot \mathbf{w}$$

    - Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\mathbf{v} \cdot \mathbf{w} = (1)(3) + (2)(4) = 9$$

  - Length

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

    - Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \|\mathbf{v}\| = \sqrt{(1)(1) + (2)(2)} = \sqrt{5}$$

Linear Algebra

- Matrix
  - Addition

$$A + B = C$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

  - Multiplication

$$AB = D$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

  - Linear equation

$$A\mathbf{x} = \mathbf{b}$$

    - Example

$$\begin{aligned} x_1 \qquad\qquad\qquad &= b_1 \\ -x_1 + x_2 \qquad\quad &= b_2 \\ -x_2 + x_3 &= b_3 \end{aligned}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Linear Algebra

- Matrix
  - Inverse matrix
    - An n-by-n square matrix, A is called invertible (or nonsingular) if there exists an n-by-n square matrix, B such that

      $$AB = BA = I_n$$

      where I_n denotes the n-by-n identity matrix which is a square matrix with ones on the main diagonal and zeros elsewhere
    - B is the inverse of A (A^{-1})
    - If A has no inverse, A is singular or non-invertible
    - Example

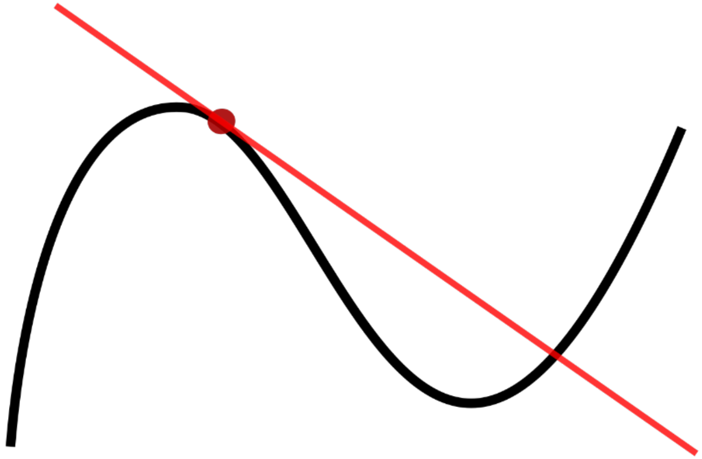
      $$A = \begin{bmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{bmatrix}, A^{-1} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}$$
  - Solution of a linear equation

    $$\mathbf{x} = A^{\mathbf{-1}}\mathbf{b}$$

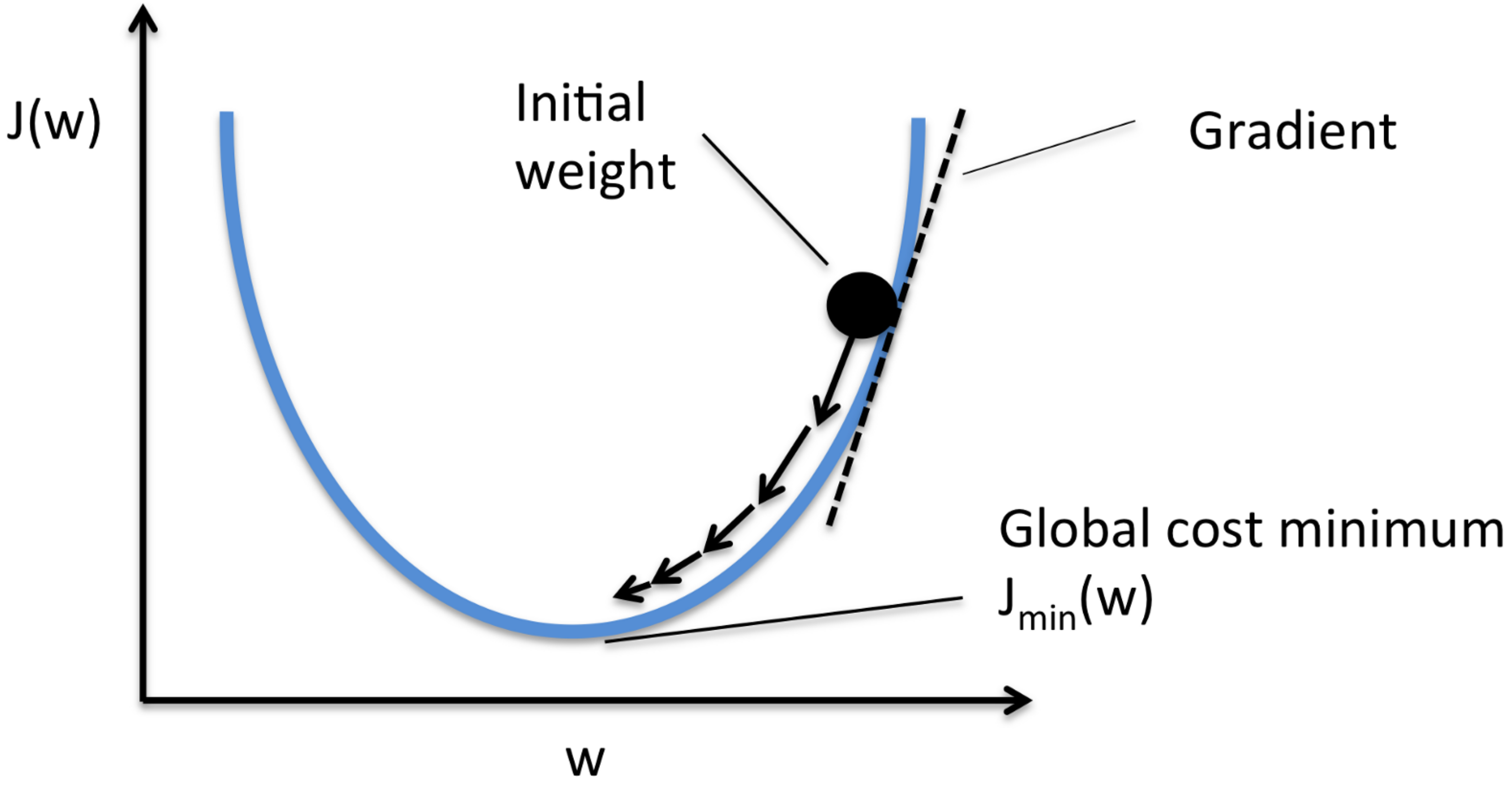
Calculus

- Derivative
  - A function of a real variable measures the sensitivity to change of the function value (output value) with respect to a change in its argument (input value)

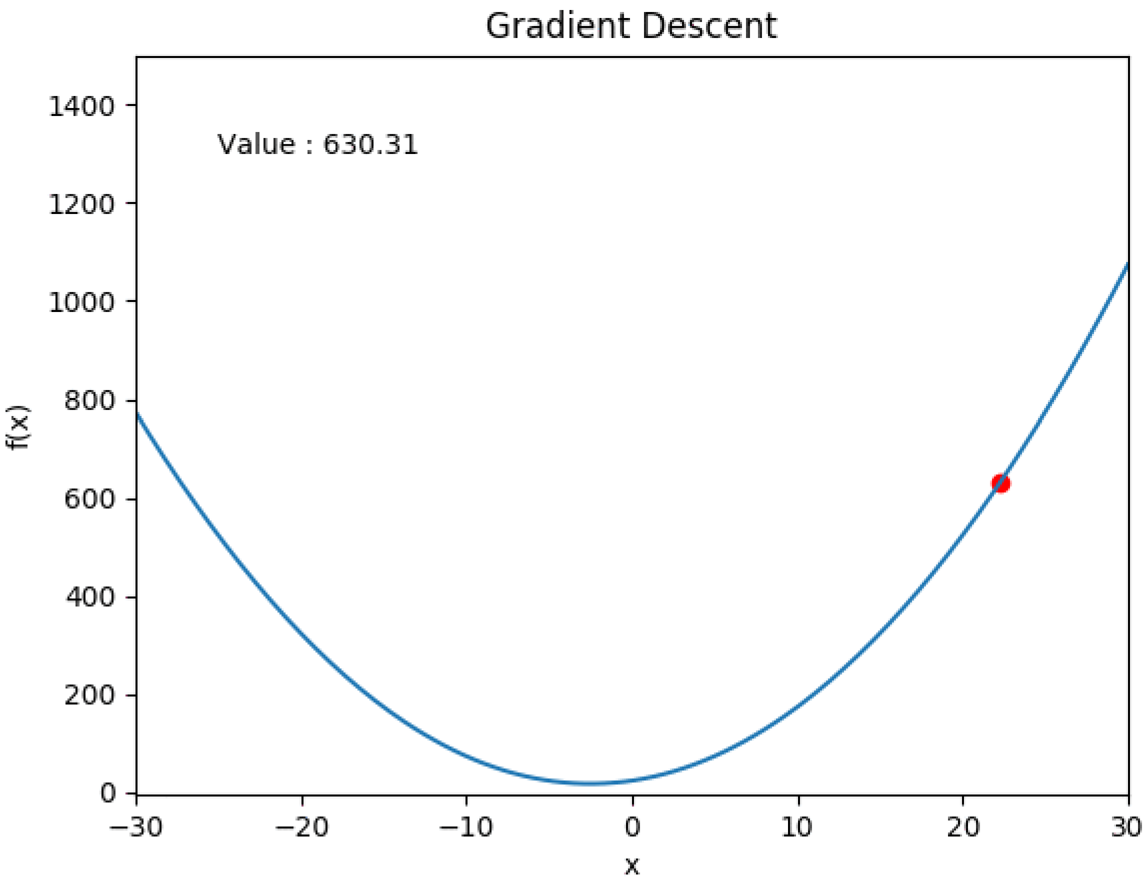
$$\frac{dy}{dx}$$



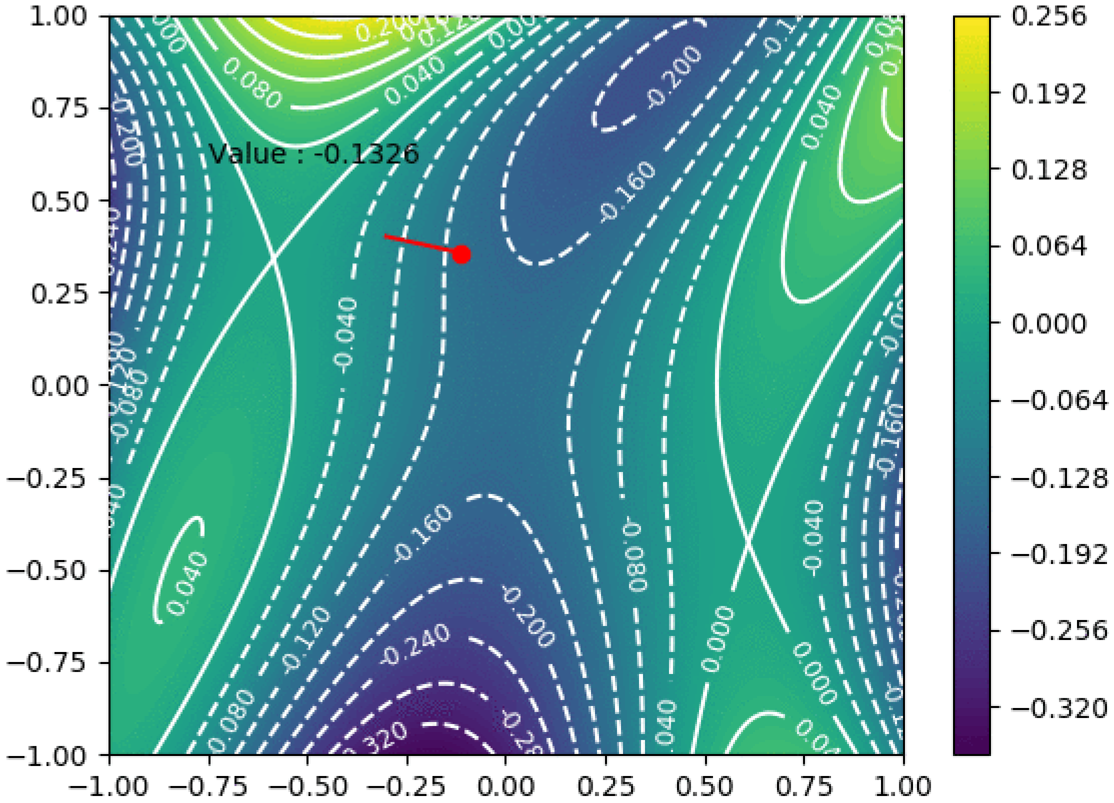
Calculus



Calculus


Gradient Descent
Value : 630.31
1400
1200
1000
800
600
400
200
0
f(x)
−30
−20
−10
0
10
20
30
x

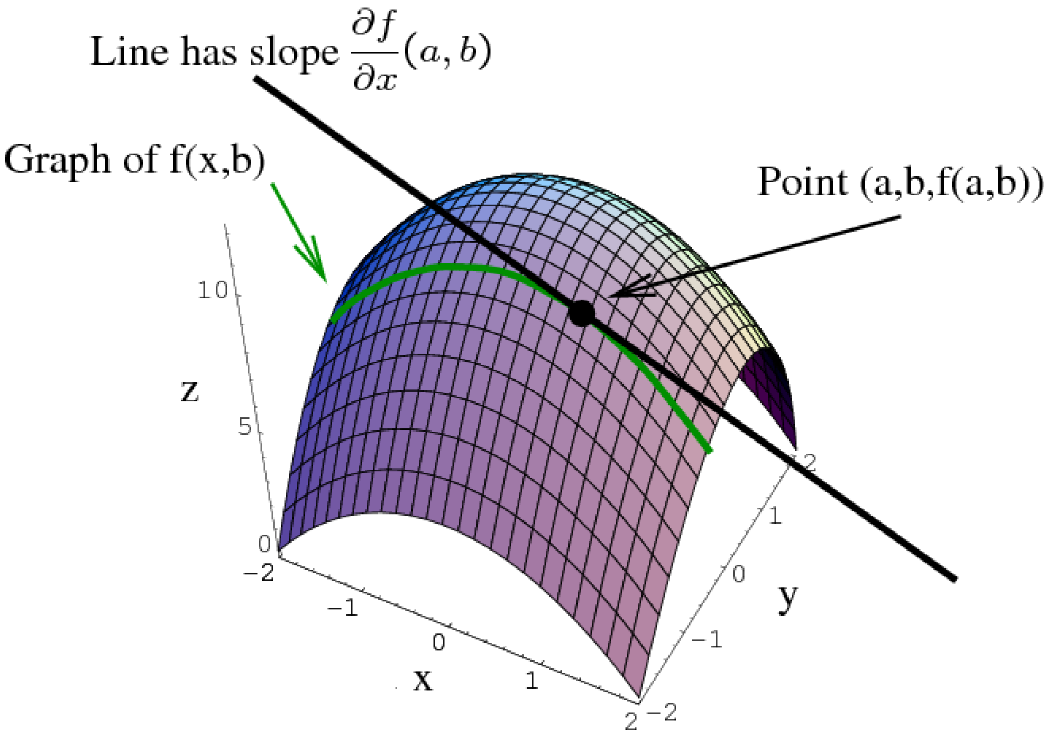
Calculus


Value : -0.1326
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
0.256
0.192
0.128
0.064
0.000
−0.064
−0.128
−0.192
−0.256
−0.320
0.080
0.040
-0.200
-0.160
-0.040
-0.080
-0.120
-0.240
0.000
-0.320

Calculus

- Partial derivative
  - A function of several variables is its derivative with respect to one of those variables, with the others held constant

$$\frac{\partial f}{\partial x}$$

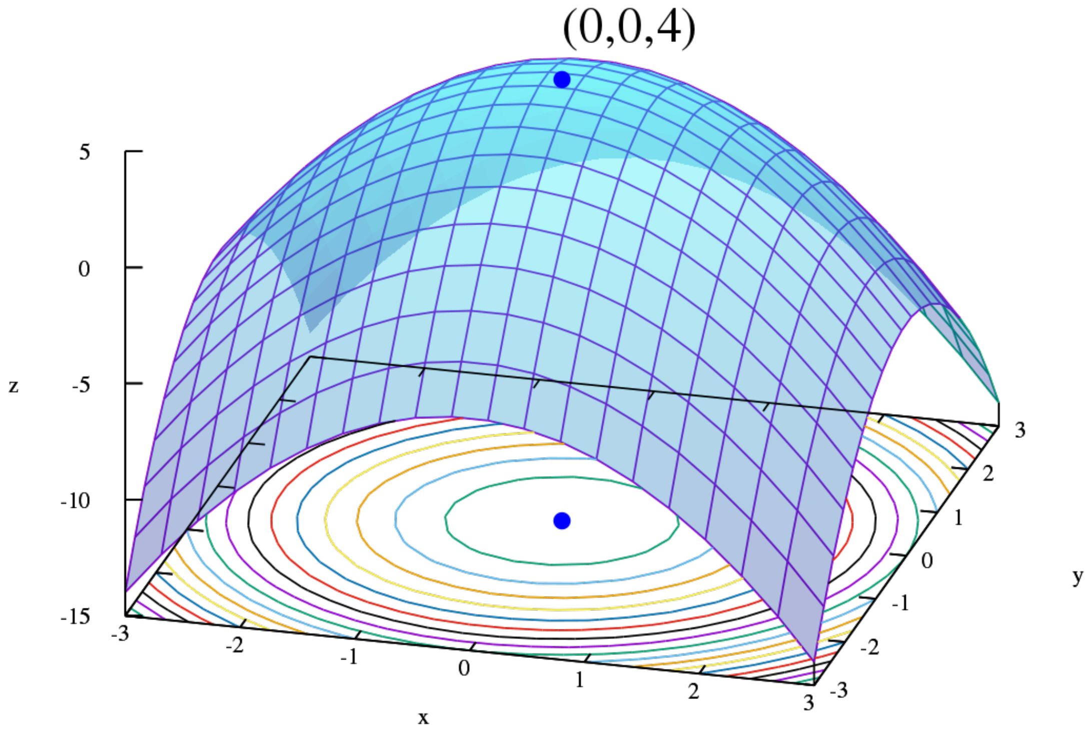


Optimization

- Optimization
  - Basics of optimization —how to formulate the problem
  - Linear programming, simplex algorithm
  - Integer programming
  - Constraint programming, knapsack problem
  - Randomized optimization techniques — hill climbing, simulated annealing, Genetic algorithms

Optimization

- Optimization problem
  - Maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function



Optimization

- Example
  - For materials, the manufacturer has 750 ㎡ of cotton textile and 1,000 ㎡ of polyester. Every pair of pants (1 unit) needs 1 ㎡ of cotton and 2 ㎡ of polyester. Every jacket needs 1.5 ㎡ of cotton and 1 ㎡ of polyester.
  - The price of the pants is fixed at $50 and the jacket, $40.
  - **What is the number of pants and jackets that the manufacturer must give to the stores so that these items obtain a maximum sale?**
    - Variables to be determined

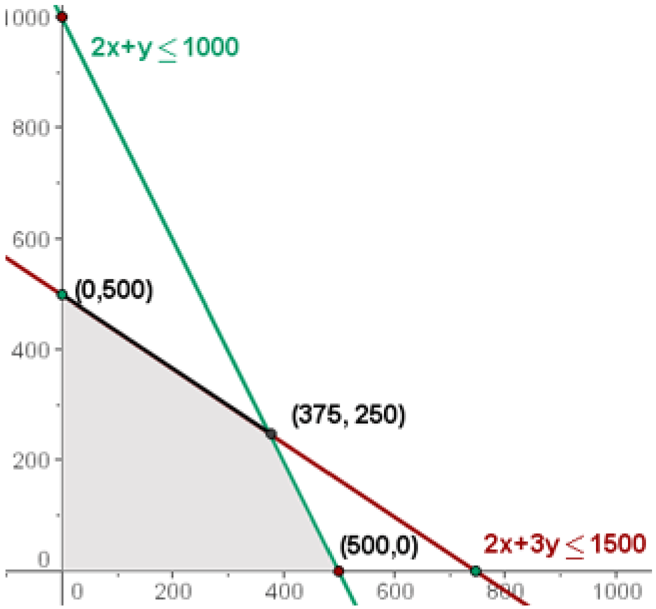
$$x = number\ of\ pants$$
$$y = number\ of\ jackets$$

    - Objective function

$$f(x, y) = 50x + 40y$$

    - Constraints

$$x + 1.5y \leq 750$$
$$2x + y \leq 1000$$



Optimization

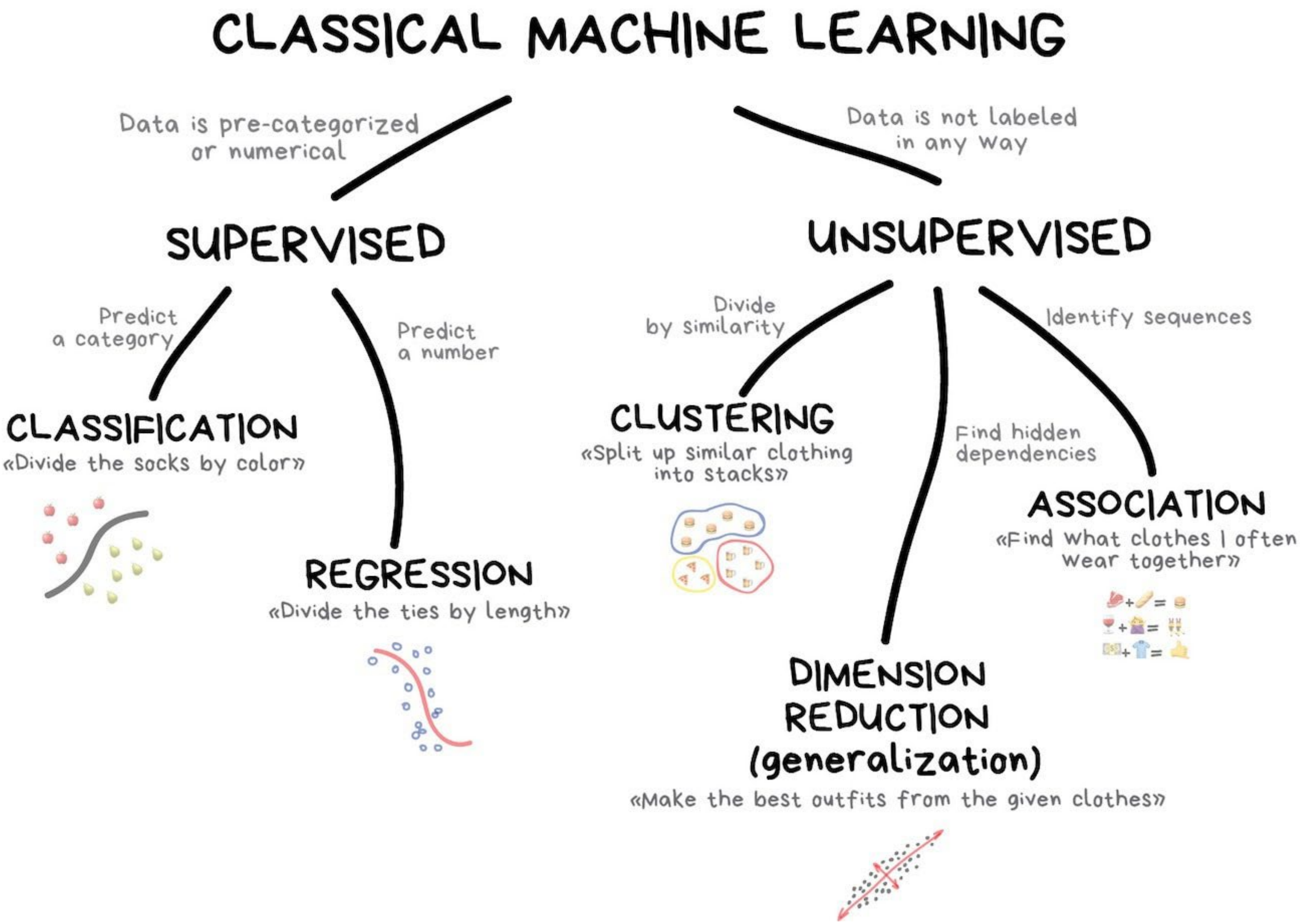
- Why are optimization algorithms important for data analysis?
  - One of fundamental data analysis tasks is to seek a function that approximately maps $\mathbf{x}_i$ to y_i for each observation, i

$$y = f(\mathbf{x})$$

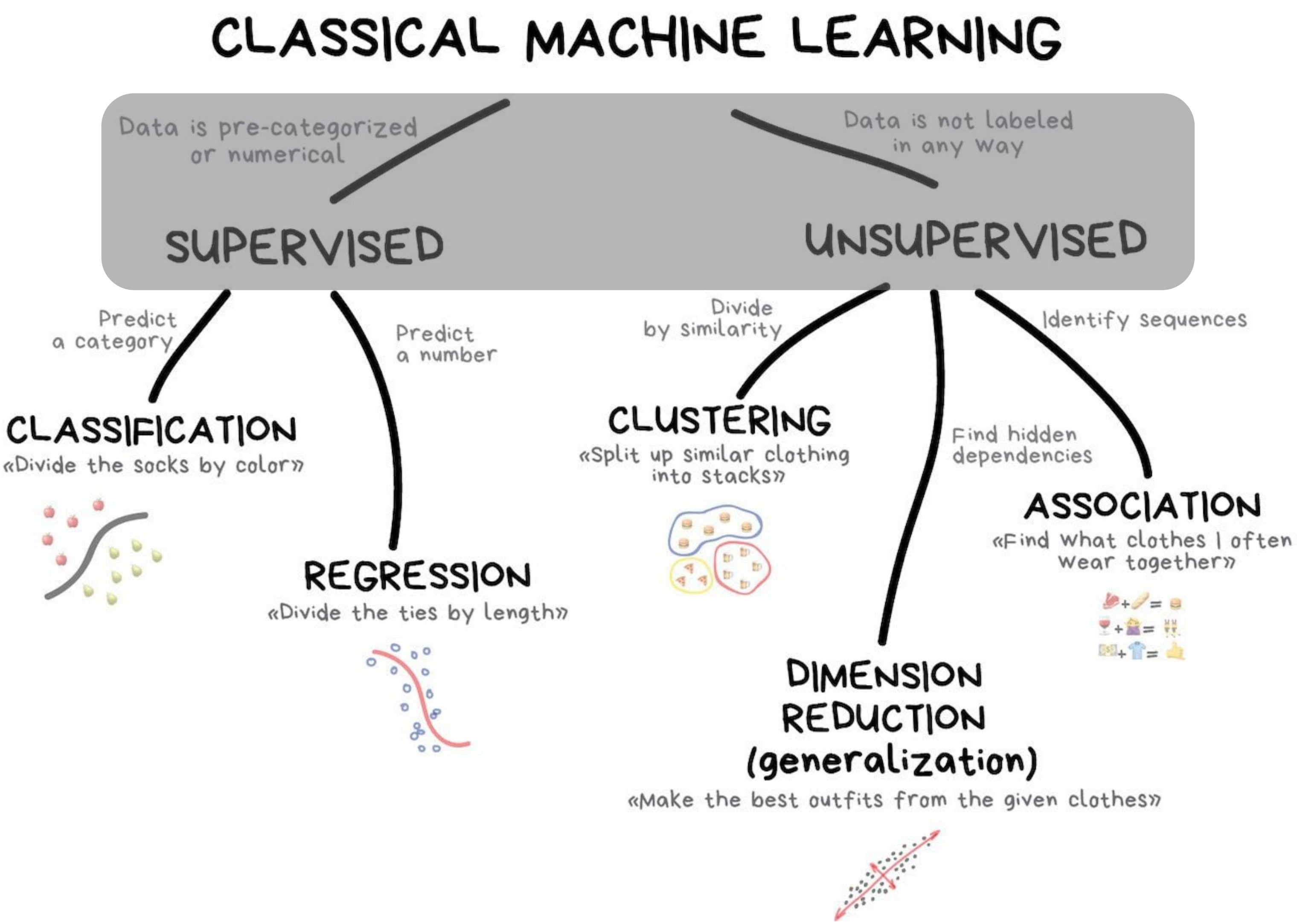
  - The process of finding f based on data is called learning or training
    - During learning, optimization algorithms provide a tool to find the most appropriate f

Basic Terminologies

Topics Covered in This Class


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
Divide by similarity
Identify sequences
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
CLUSTERING
«Split up similar clothing into stacks»
Find hidden dependencies
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

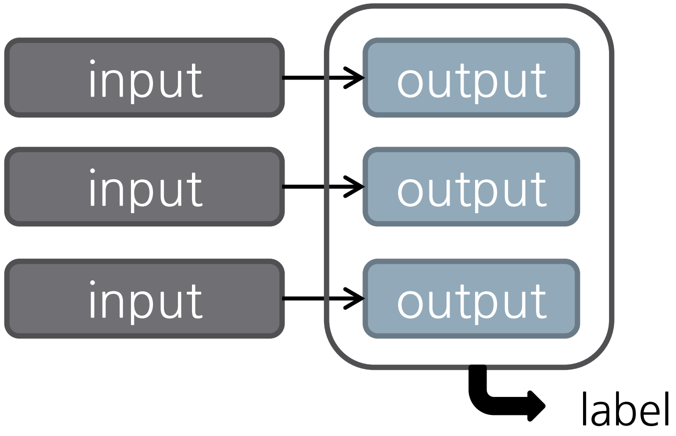
Topics Covered in This Class


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
Divide by similarity
Identify sequences
Find hidden dependencies
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
CLUSTERING
«Split up similar clothing into stacks»
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

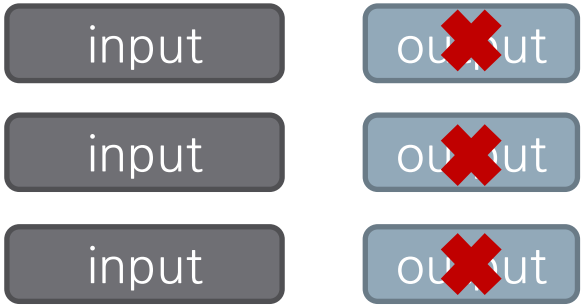
Types of Learning

- Supervised learning
  - We have knowledge of output
    - We call such data labeled

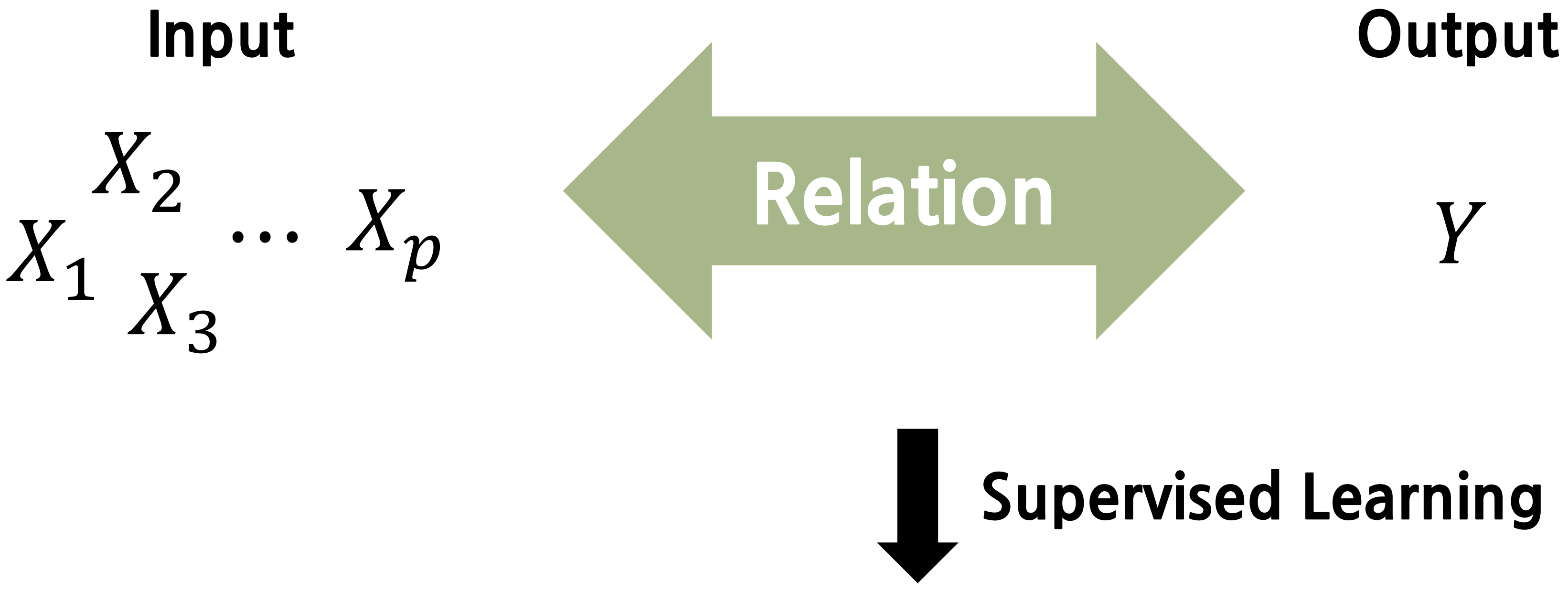
    → We know answer
  - Goal
    - Estimate output for unlabeled input



- Unsupervised learning
  - No output
    - We call such data unlabeled
  - Goal
    - Find patterns, groups, or relation



Supervised learning



Input

X_1 X_2 X_3 $\cdots$ X_p

Relation

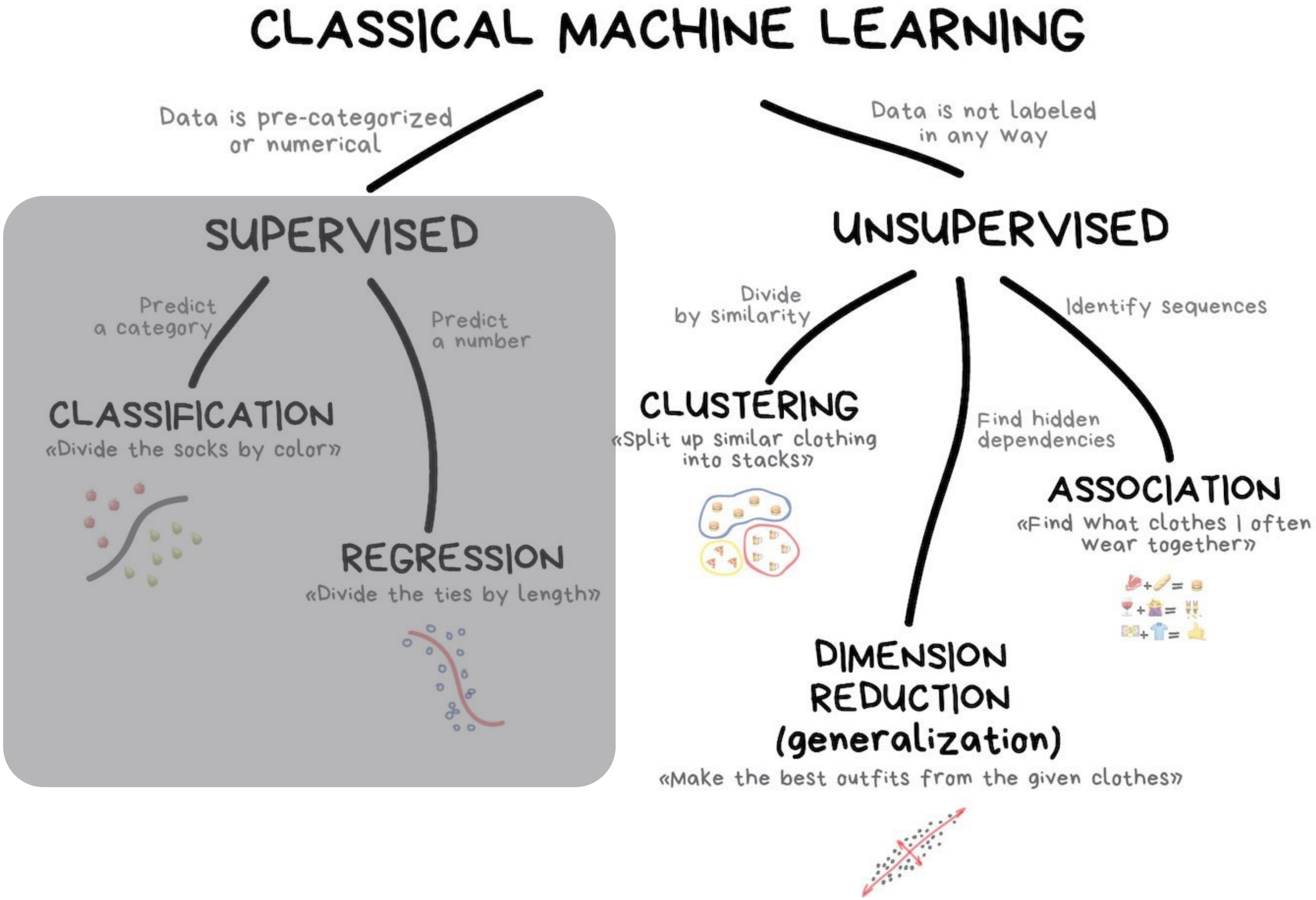
Output

Y

Supervised Learning

$$Y = f(X_1, X_2, \cdots, X_p)$$

Topics Covered in This Class


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
Divide by similarity
Identify sequences
CLUSTERING
«Split up similar clothing into stacks»
Find hidden dependencies
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

Data for Data Mining: Structured Data

- Example of data set
  - The input data set is usually expressed as a set of independent instances

instance,
sample,
example

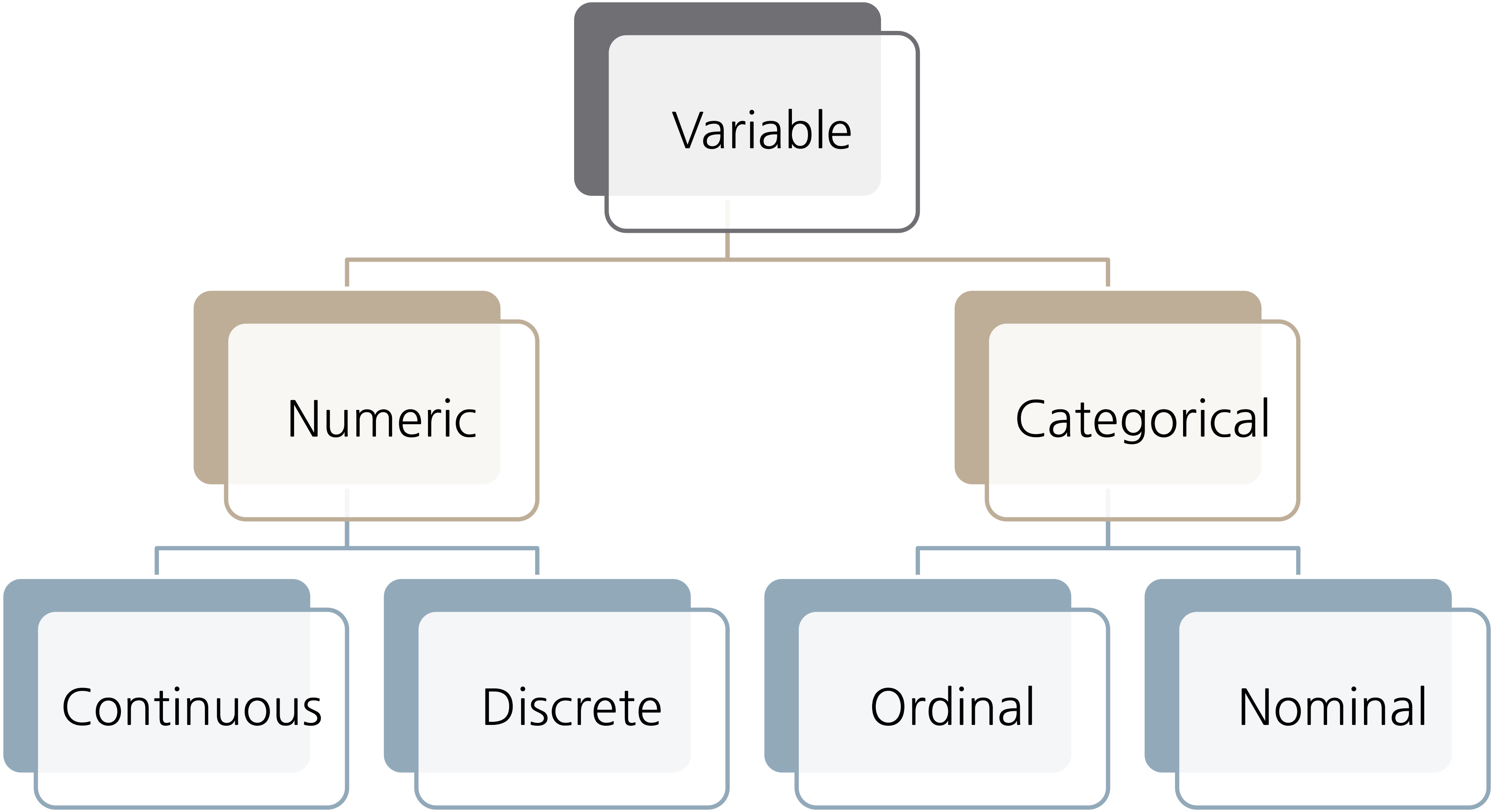
Outlook	Temperature(℉)	Humidity(%)	Windy	Play Time(min)
Sunny	85	85	false	5
Sunny	80	90	true	0
Rainy	70	96	false	40
Rainy	68	80	false	65
Sunny	72	95	false	0
Sunny	69	70	false	70
Rainy	75	80	true	45

variable,
attribute,
feature

Types of Data

- Structured
  - Values of variable reside in a fixed field
  - Examples
    - Numeric
    - Date
    - Restricted terms: (male, female), (Mr., Ms., Mrs.)
    - Address
- Unstructured
  - Values of variable do not reside in a fixed field
  - Examples
    - Documents
    - Webpages
    - Images
    - Videos

Structured Data: Types of Variables



Structured Data: Types of Variables

- Numeric (Quantitative)
  - A broad category that includes any variable that can be counted, or has a numerical
- Continuous
  - A variable with infinite number of values
  - Example
    - Many numeric variables: temperature, weight, height, pressure and etc.
- Discrete
  - A variable that can only take on a certain number of values or have a countable number of values between any two values
  - Example
    - The number of cars in a parking lot
    - the number of flaws or defects

Structured Data: Types of Variables

- Categorical
  - A variable that contains a finite number of categories or distinct groups
- Nominal
  - A Variable that has two or more categories, but there is no intrinsic ordering to the categories.
  - Example
    - (Male, Female), (Class 1, Class 2, Class 3), (Red, Yellow, Green)
- Ordinal
  - Similar to a nominal variable, but the difference between the two is that there is a clear ordering of the variables.
  - Example
    - Score: A+,A,A-,B+,B,B-,C+,C,C-,D,F
    - Size: S, M, L, XL, XXL

Example: The Input to a Data Mining

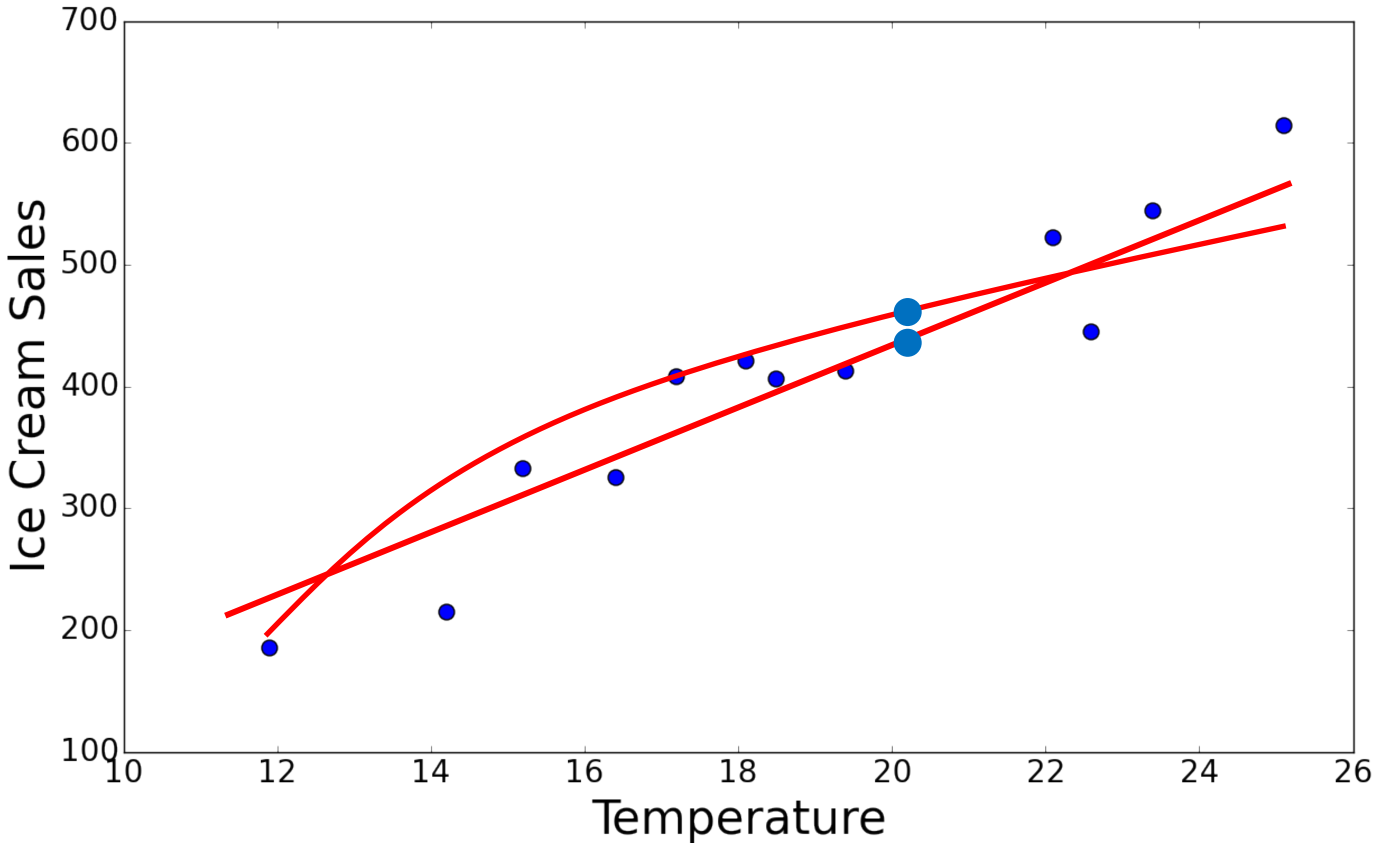
- Example of data set

num-of-doors	body-style	wheel-base	length	make
2	convertible	88.6	168.8	Audi
2	convertible	88.6	168.8	BMW
2	hatchback	94.5	171.2	Chevrolet
4	sedan	99.8	176.6	BMW
4	sedan	99.4	176.6	Audi
2	sedan	99.8	177.3	Audi
4	wagon	105.8	192.7	Chevrolet

Types:	**Discrete**	**Nominal**	**Continuous**	**Continuous**	**Nominal**

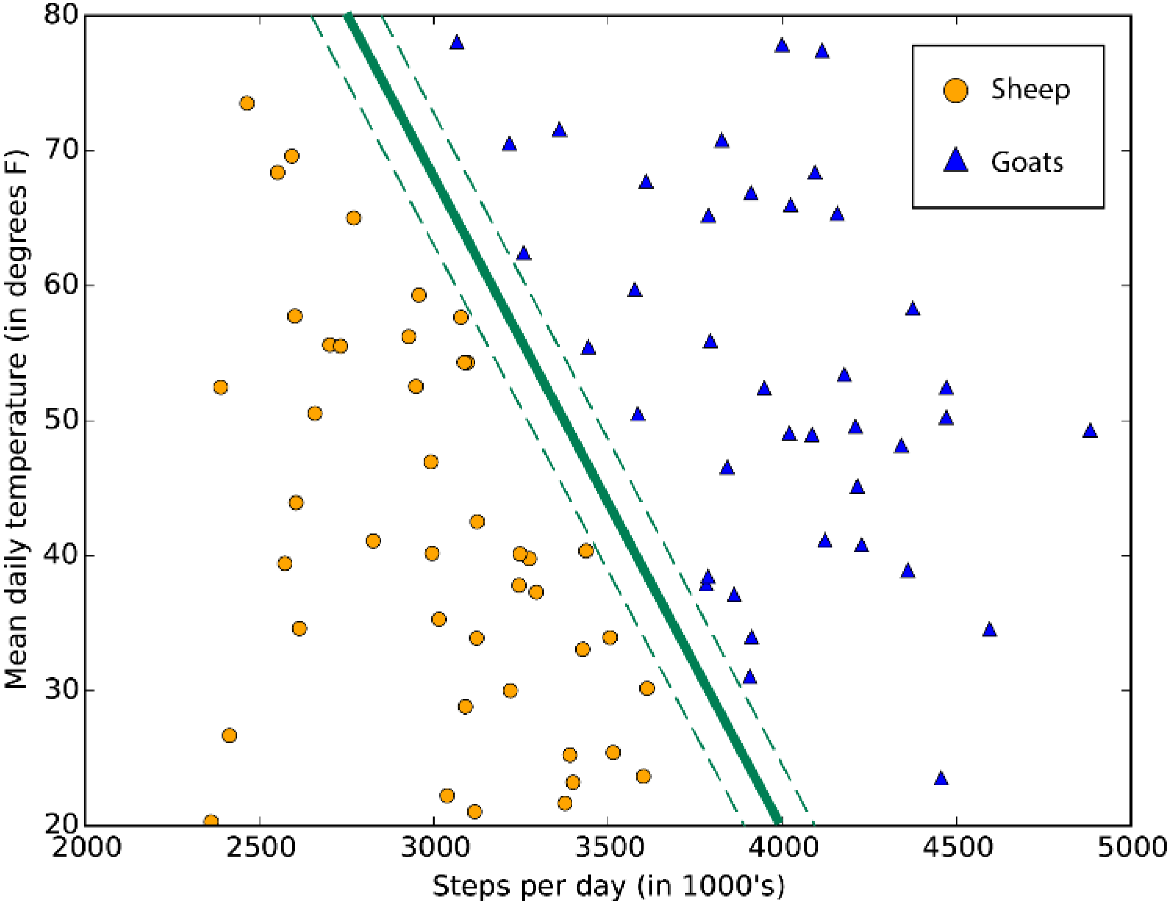
Supervised Learning: Regression

- Temperature vs. Ice Cream Sales
  - How about 21℃?



Supervised Learning: Classification

- Which one is a sheep?



https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice

Question

□ Suppose you are working on weather prediction, and you would like to predict whether or not it will be raining at 5pm tomorrow.
You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?

① Regression

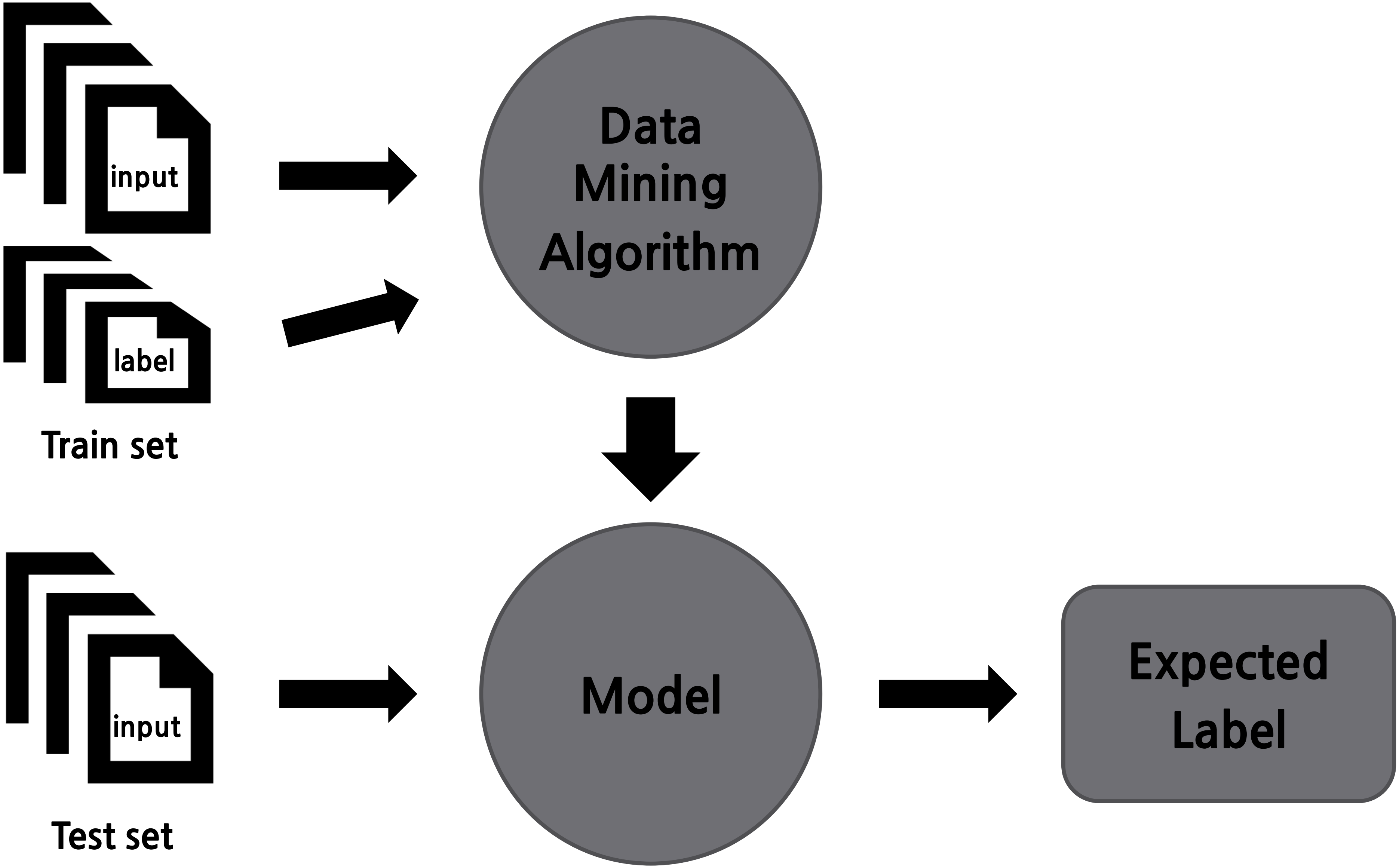
② Classification

□ Suppose you are working on stock market prediction, and you would like to predict the price of the specific stock tomorrow (measured in dollars).
You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?

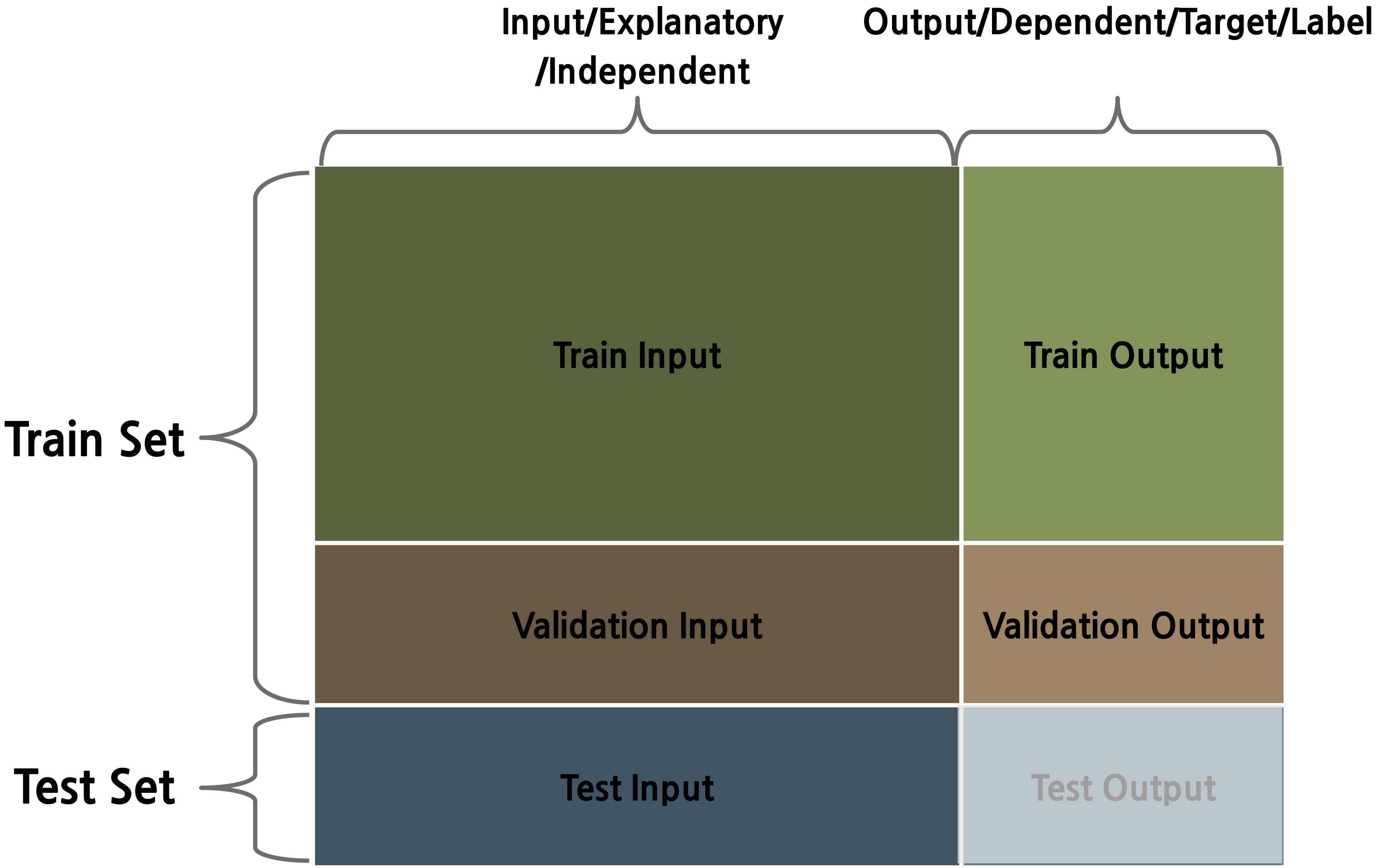
① Regression

② Classification

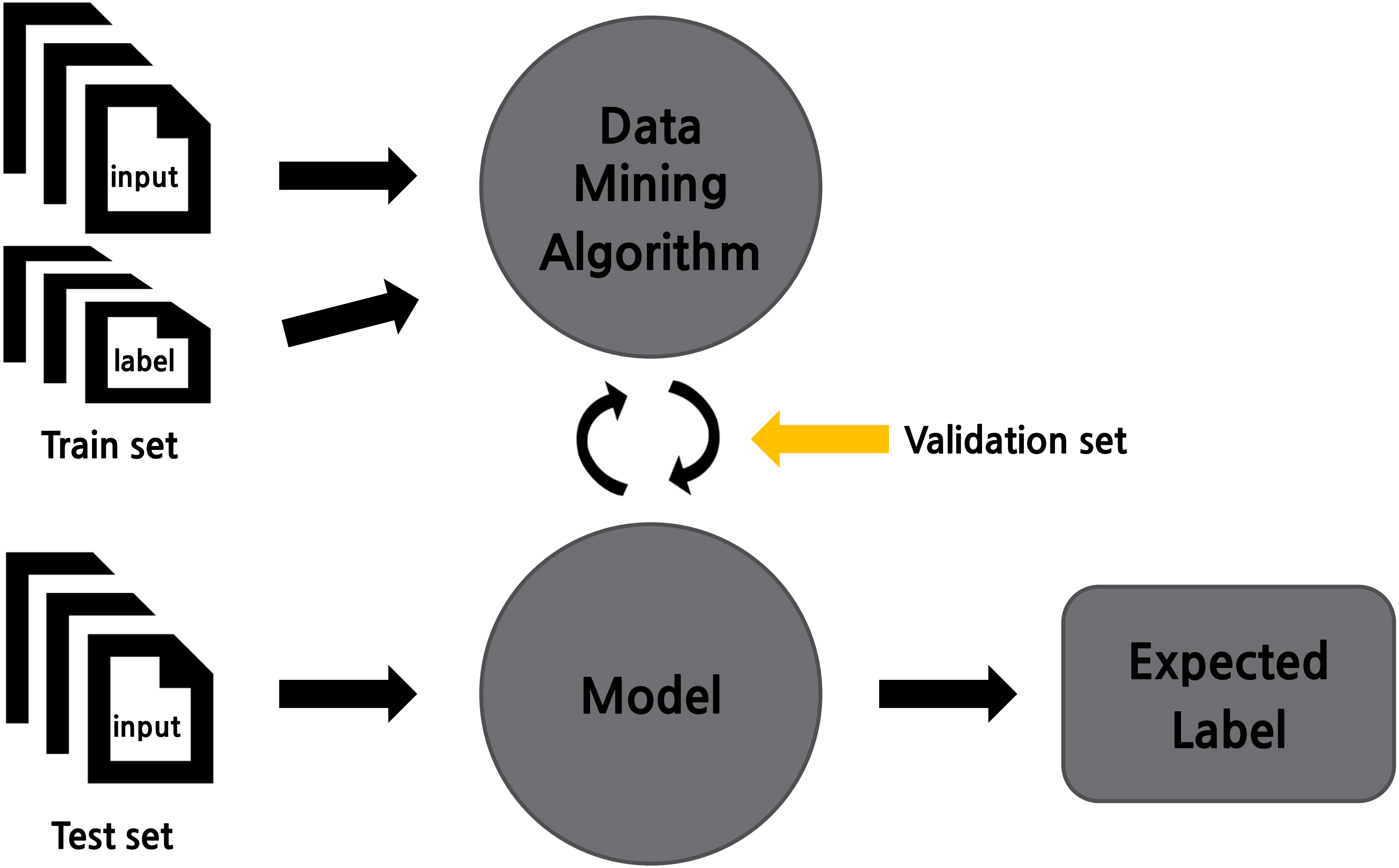
Process of Supervised Learning



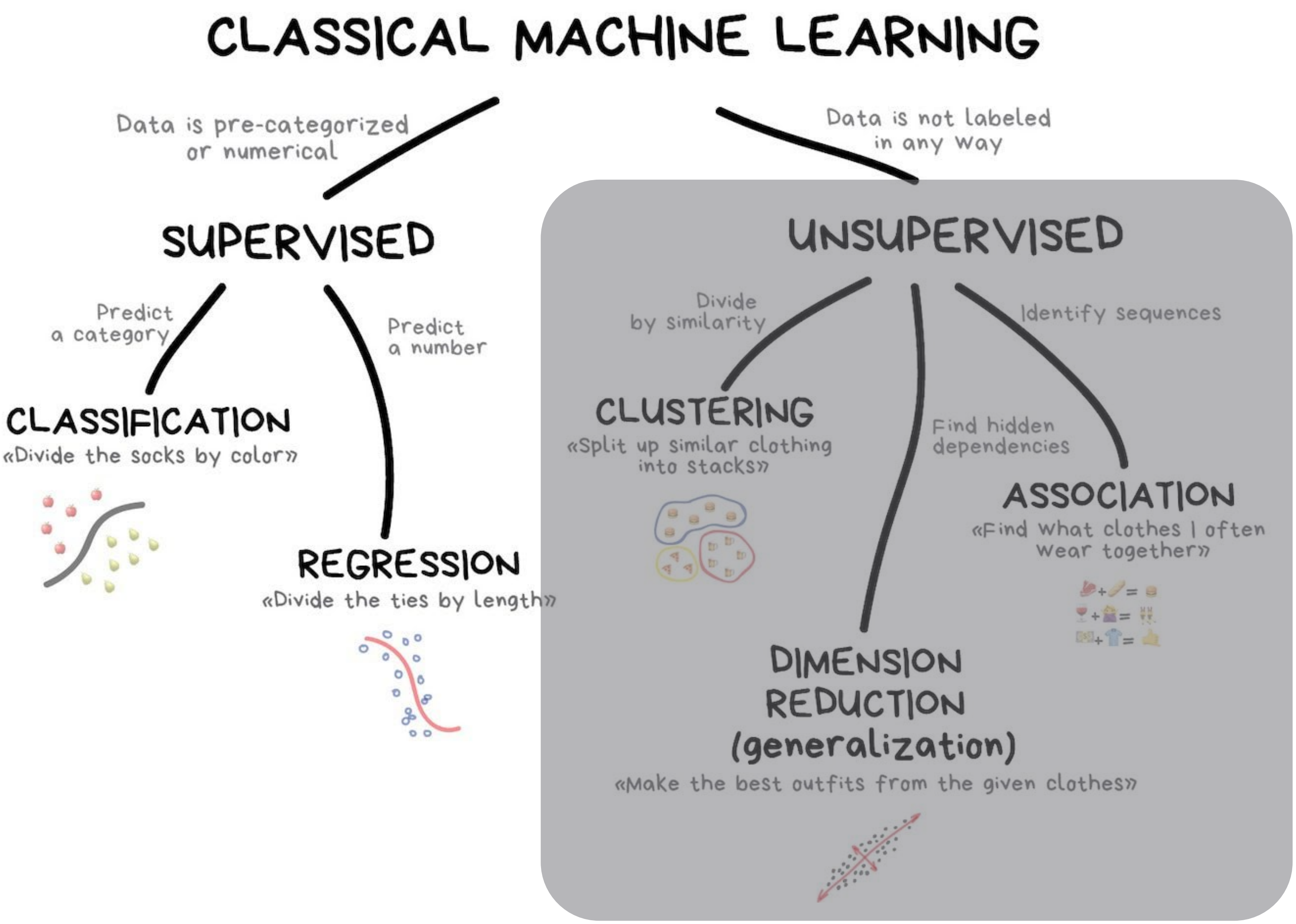
Data Partition



Process of Supervised Learning

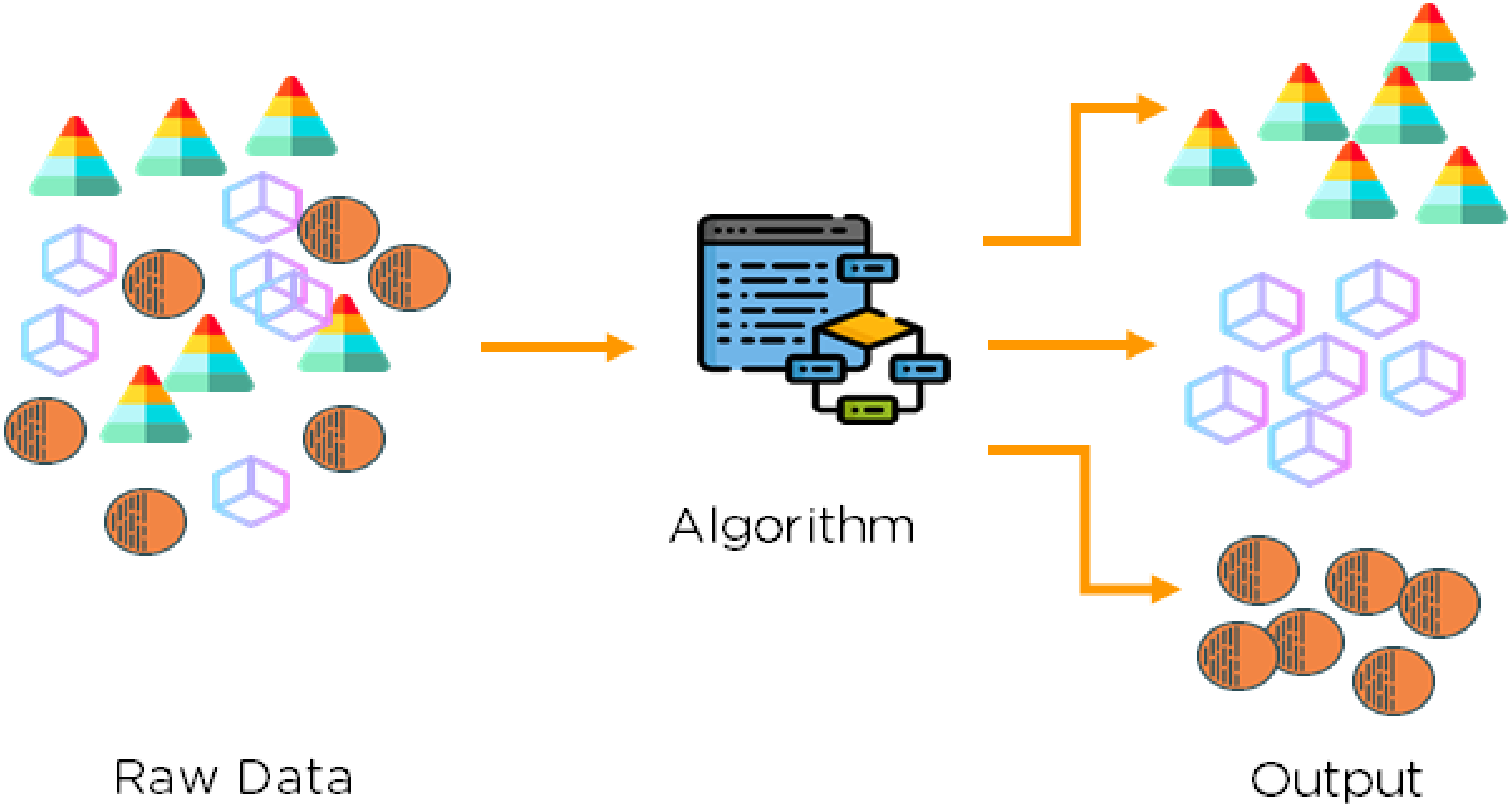


Topics Covered in This Class


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
Divide by similarity
Identify sequences
Find hidden dependencies
CLUSTERING
«Split up similar clothing into stacks»
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

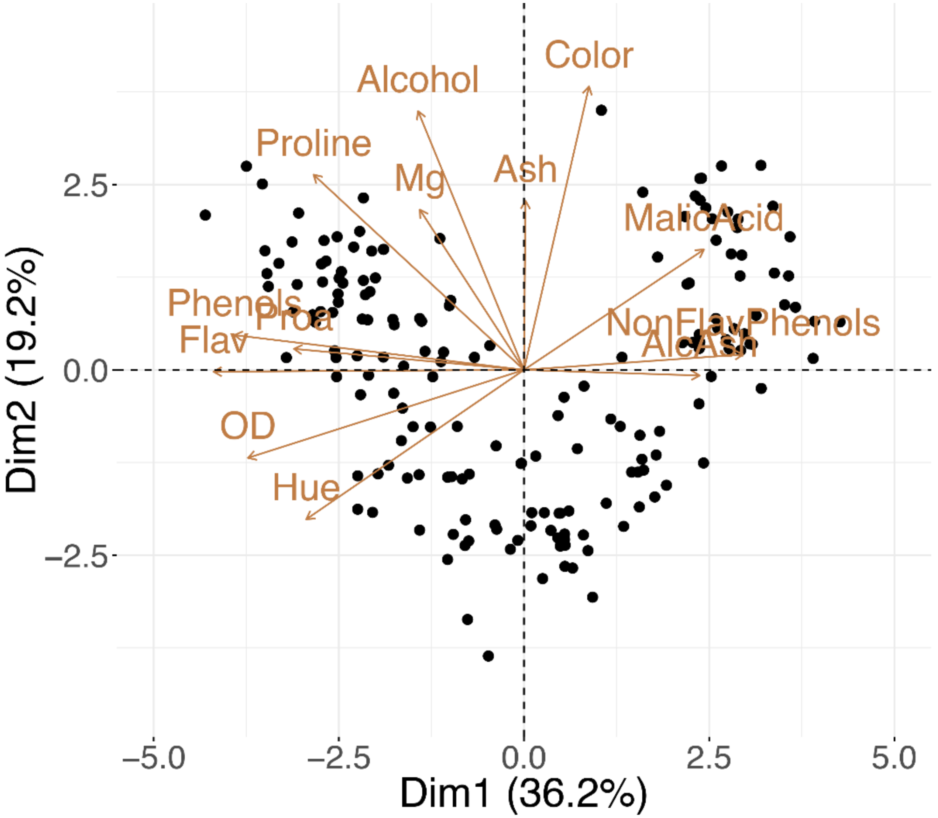
Unsupervised Learning: Clustering

- Grouping data points
  - How to determine which group does each data belongs to?



Unsupervised Learning: Dimensionality Reduction

- Dimensionality reduction
  - The process of reducing the number of random variables under consideration by obtaining a set of principal variables
  - High dimension → Low dimension



Unsupervised Learning: Association Rule Mining

- Find useful information from transactions

Datetime	Customer	Items
2015-07-15 14:03	1	orange juice, banana
2015-07-15 16:20	2	orange juice, milk
2015-07-16 10:14	3	detergent, banana, orange juice
2015-07-25 19:34	2	milk, bread, soda
2015-07-29 09:41	4	detergent, window cleaner
2015-08-01 20:55	1	bread, milk

- One of useful information is information like “If item A then item B”
  - This information is called association rule

- Find pair of items that are more likely to be purchased together based on transactions

Question

- □ Of the following examples, which would you address using an unsupervised learning algorithm? (Find all that apply.)

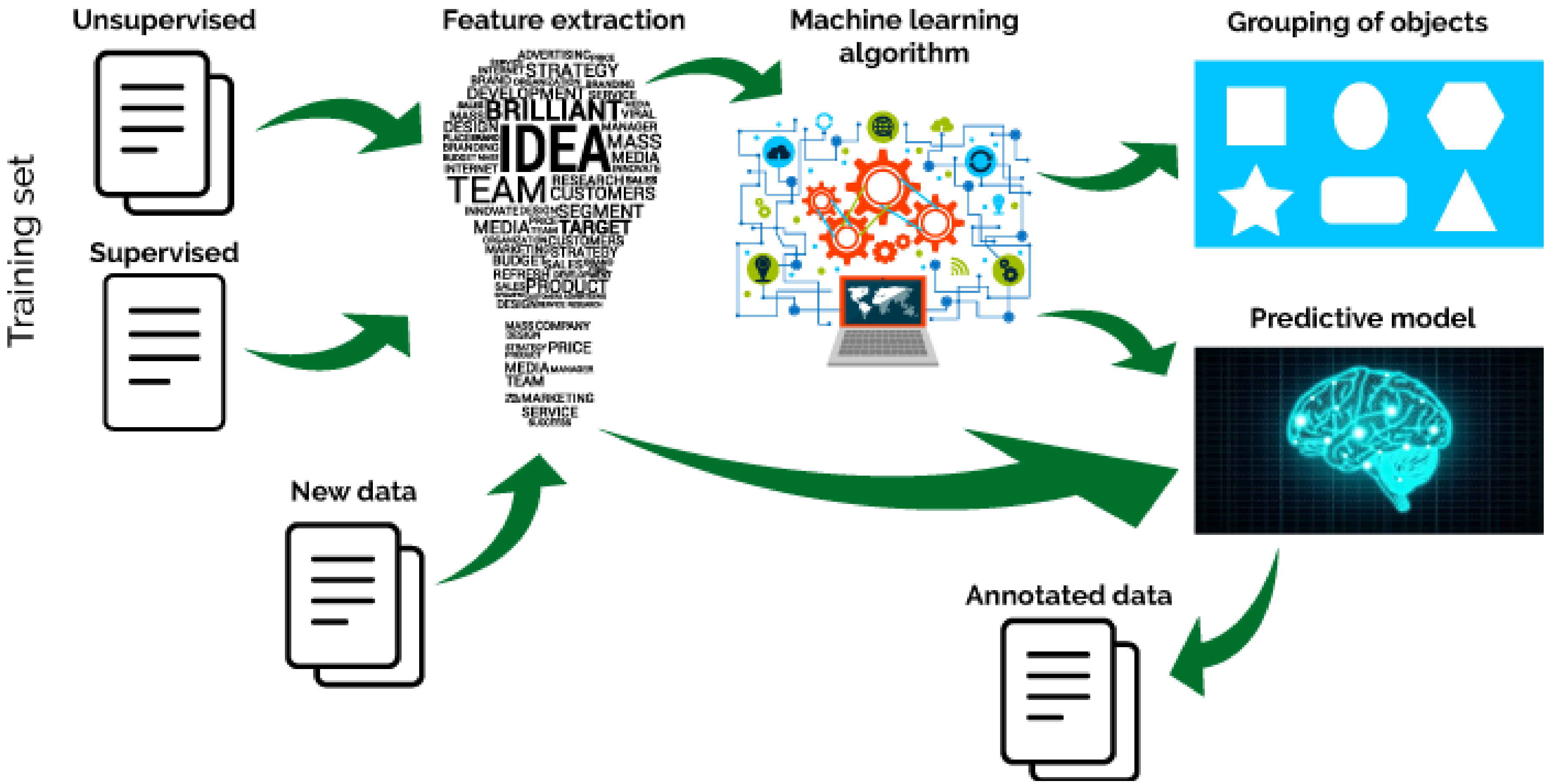
① Given email labeled as spam/not spam, learn a spam filter.

② Given a set of news articles found on the web, group them into set of articles about the same story.

③ Given a database of customer data, automatically discover market segments and group customers into different market segments.

④ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Overall Description



Training set

Unsupervised

Supervised

Feature extraction

Machine learning algorithm

Grouping of objects

Predictive model

New data

Annotated data

Python: Short Tutorial

Data Handling

pandas

- pandas consists of the following things
  - A set of labeled array data structures, the primary of which are Series and DataFrame
  - Index objects enabling both simple axis indexing and multi-level / hierarchical axis indexing
  - An integrated group by engine for aggregating and transforming data sets
  - Date range generation (date_range) and custom date offsets enabling the implementation of customized frequencies
  - Input/Output tools: loading tabular data from flat files (CSV, delimited, Excel 2003), and saving and loading pandas objects from the fast and efficient PyTables/HDF5 format.
  - Memory-efficient "sparse" versions of the standard data structures for storing data that is mostly missing or mostly constant (some fixed value)
  - Moving window statistics (rolling mean, rolling standard deviation, etc.)
  - Static and moving window linear and panel regression

Data structures

- Data structures provided by pandas

Dimensions	Name	Description
1	Series	1D labeled homogeneously-typed array
2	DataFrame	General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed columns
3	Panel	General 3D labeled, also size-mutable array

Object Creation

- Import

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

- Object creation

```
# Series
s = pd.Series([1,3,5,np.nan,6,8])

# DataFrame
dates = pd.date_range('20130101', periods=6)
df = pd.DataFrame(np.random.randn(6,4), index=dates,
columns=list('ABCD'))
```

Object Creation

- Creating DataFrame by passing a dict of objects that can be converted to series-like

```
df2 = pd.DataFrame({ 'A' : 1.,
                     'B' : pd.Timestamp('20130102'),
                     'C' : pd.Series(1,index=list(range(4)),dtype='float32'),
                     'D' : np.array([3] * 4,dtype='int32'),
                     'E' : pd.Categorical(["test","train","test","train"]),
                     'F' : 'foo' })
```

  - If you want to check types of each column

```
print(df2.dtypes)
```

Viewing Data

- See the top and bottom rows of the data

```
print(df.head())
print(df.tail(3))
```

- Display the index, columns, and the underlying numpy data

```
print(df.index)
print(df.columns)
print(df.values)
```

- Show a quick statistic summary of the data

```
print(df.describe())
```

- Sort by an axis or values

```
print(df.sort_index(axis=1,ascending=False))
print(df.sort_values(by='B'))
```

Selection

- Selecting a single column, which yields a Series, equivalent to df.A

```
a=df['A']
print(a)
```

- Selecting via [], which slices the rows

```
selRow=df[0:3]
print(selRow)
```

- For getting a cross section using a label

```
selRow=df.loc[dates[0]]
```

- Selecting on a multi-axis by label

```
selData=df.loc['20130102':'20130104',['A','B']]
```

Selection

- For getting a scalar value

```
print(df.iloc[3])
```

- By integer slices, acting similar to numpy/python

```
print(df.iloc[3:5,0:2])
```

- By lists of integer position locations, similar to the numpy/python style

```
print(df.iloc[[1,2,4],[0,2]])
```

- For slicing rows explicitly or columns explictly

```
print(df.iloc[1:3,:])
print(df.iloc[:,1:3])
```

Selection

- Using a single column’s values to select data

```
print(df[df.A > 0])
```

- A **where** operation for getting

```
print(df[df > 0])
```

Setting

- Setting a new column automatically aligns the data by the indexes

```
s1 = pd.Series([1,2,3,4,5,6], index=pd.date_range('20130102',
periods=6))
df['F'] = s1
```

- Setting values by label

```
df.at[dates[0],'A'] = 0
```

- Setting values by position

```
df.iat[0,1] = 0
```

- Setting by assigning with a numpy array

```
df.loc[:,'D'] = np.array([5] * len(df))
```

- A **where** operation with setting

```
df2 = df.copy()
df2[df2 > 0] = -df2
```

Missing Data

- pandas primarily uses the value np.nan to represent missing data
- To drop any rows that have missing data

```
df1 = df.reindex(index=dates[0:4], columns=list(df.columns) + ['E'])
df1.loc[dates[0]:dates[1],'E'] = 1

newdf1=df1.dropna(how='any')
```

- Filling missing data

```
newdf1=df1.fillna(value=5)
```

Operation

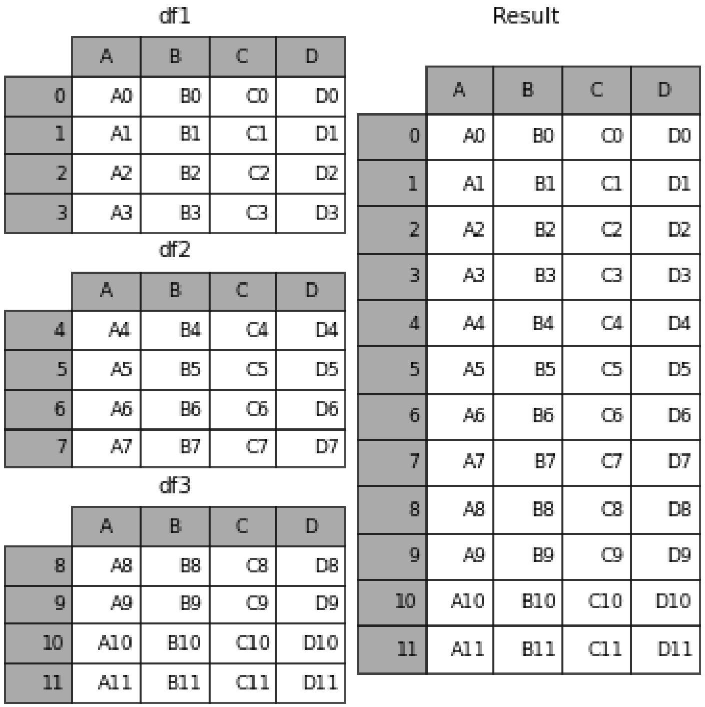
□ Performing a descriptive statistic

```
print(df.mean())
print(df.mean(1))

print(df.max())
print(df.max(1))

print(df.min())
print(df.min(1))
```

Merge

- pandas provides various facilities for easily combining together Series, DataFrame, and Panel objects
- Concatenating pandas objects together

```
df = pd.DataFrame(np.random.randn(10, 4))
pieces = [df[:3], df[3:7], df[7:]]
pd.concat(pieces)
```



df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Result

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

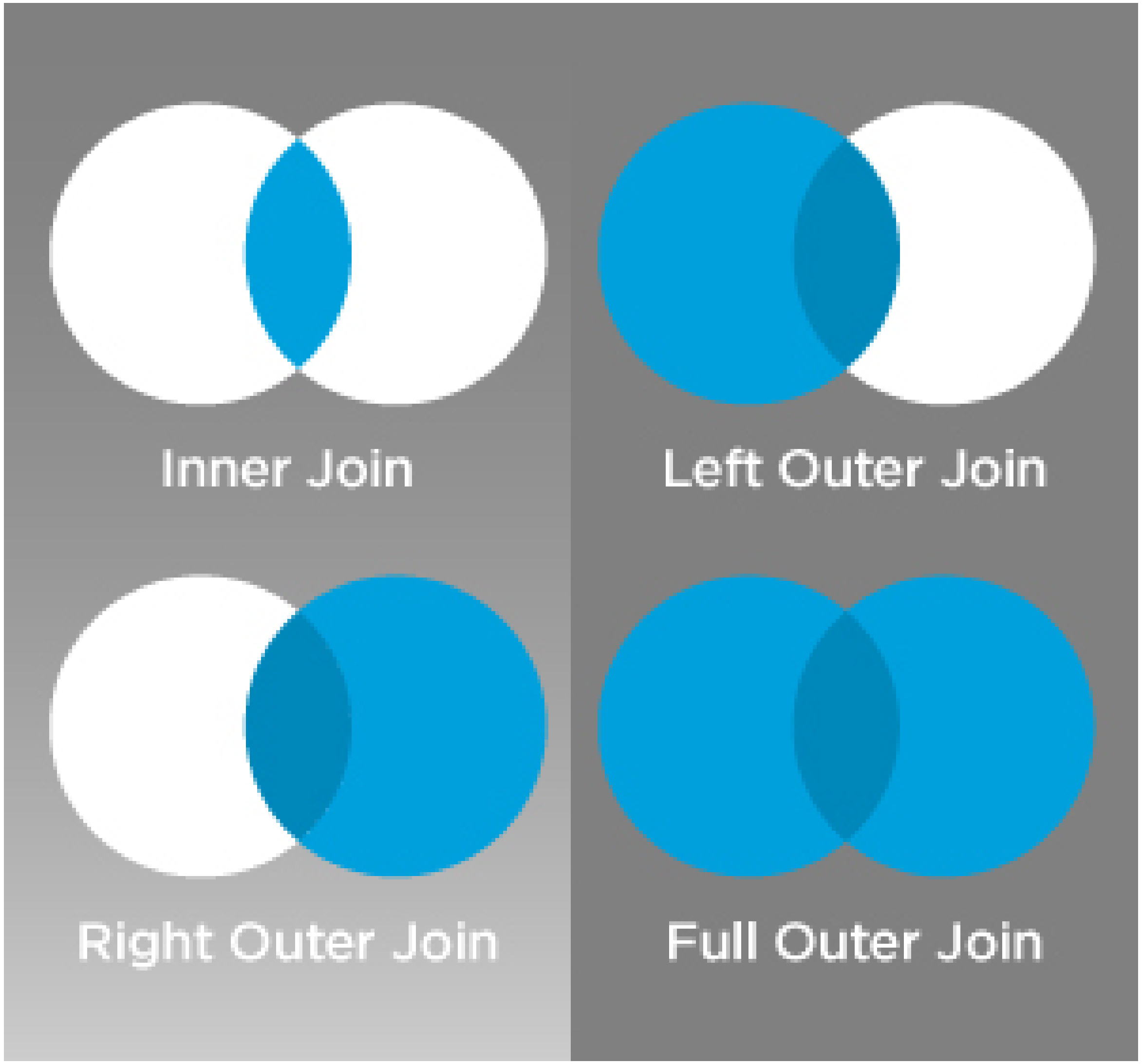
Merge

- SQL style merges
  - basic syntax

```
merge(left, right, how='inner', on=None, left_on=None,
right_on=None,left_index=False, right_index=False,
sort=True,suffixes=('_x', '_y'), copy=True, indicator=False)
```

- left_on: Columns from the left DataFrame to use as keys
- left_index: If True, use the index (row labels) from the left DataFrame as its join key(s)
- how: One of 'left', 'right', 'outer', 'inner'. Defaults to inner
- sort: Sort the result DataFrame by the join keys in lexicographical order
- suffixes: A tuple of string suffixes to apply to overlapping columns
- copy: Always copy data (default True) from the passed DataFrame objects, even when reindexing is not necessary
- indicator: Add a column to the output DataFrame called _merge with information on the source of each row

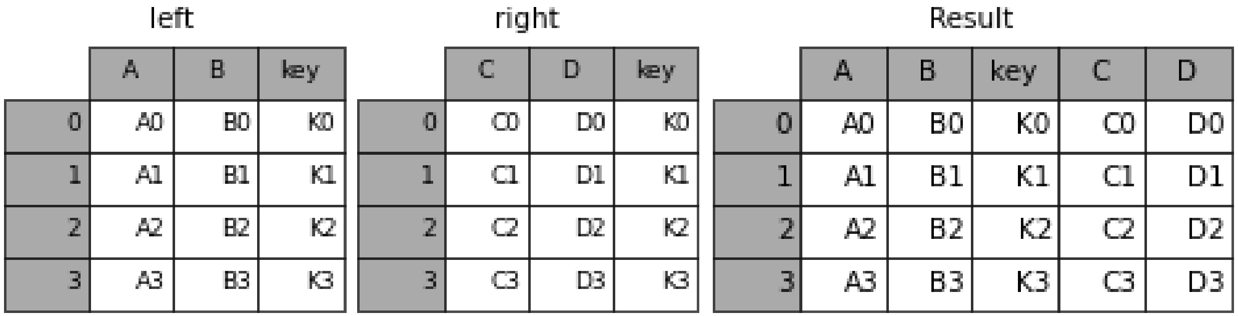
Merge



Merge

- □ One unique key combination

```
left = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],'A': ['A0', 'A1',
'A2', 'A3'],'B': ['B0', 'B1', 'B2', 'B3']})
right = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],                    'C':
['C0', 'C1', 'C2', 'C3'],'D': ['D0', 'D1', 'D2', 'D3']})
result = pd.merge(left, right, on='key')
```



left

	A	B	key
0	A0	B0	K0
1	A1	B1	K1
2	A2	B2	K2
3	A3	B3	K3

right

	C	D	key
0	C0	D0	K0
1	C1	D1	K1
2	C2	D2	K2
3	C3	D3	K3

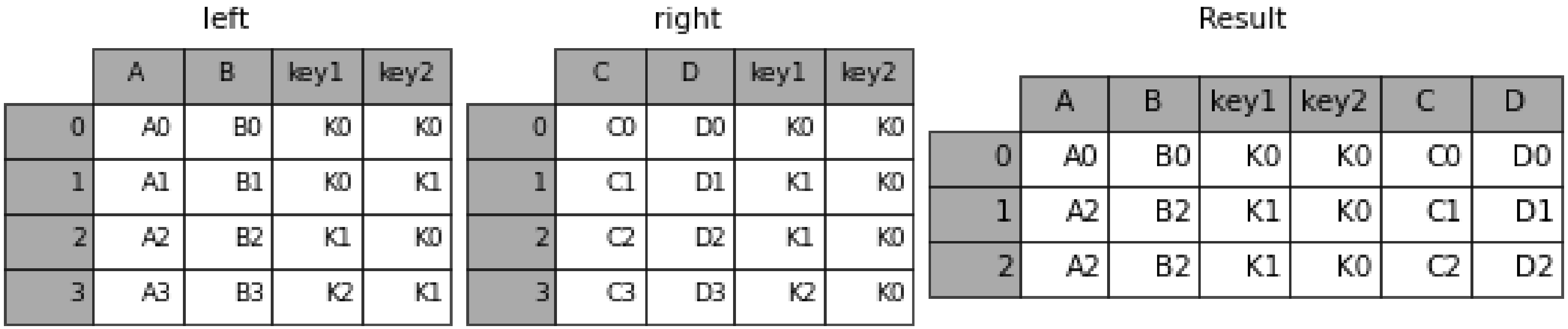
Result

	A	B	key	C	D
0	A0	B0	K0	C0	D0
1	A1	B1	K1	C1	D1
2	A2	B2	K2	C2	D2
3	A3	B3	K3	C3	D3

Merge

- Multiple join keys

```
left = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],'key2': ['K0', 'K1',
'K0', 'K1'],'A': ['A0', 'A1', 'A2', 'A3'],'B': ['B0', 'B1', 'B2', 'B3']})
right = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],'key2': ['K0',
'K0', 'K0', 'K0'],'C': ['C0', 'C1', 'C2', 'C3'],'D': ['D0', 'D1', 'D2',
'D3']})
result = pd.merge(left, right, on=['key1', 'key2'])
```



left

	A	B	key1	key2
0	A0	B0	K0	K0
1	A1	B1	K0	K1
2	A2	B2	K1	K0
3	A3	B3	K2	K1

right

	C	D	key1	key2
0	C0	D0	K0	K0
1	C1	D1	K1	K0
2	C2	D2	K1	K0
3	C3	D3	K2	K0

Result

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A2	B2	K1	K0	C1	D1
2	A2	B2	K1	K0	C2	D2

- For more information
  - http://pandas.pydata.org/pandas-docs/stable/merging.html#merging-join

Grouping

- By “group by” we are referring to a process involving one or more of the following steps
  - Splitting the data into groups based on some criteria
  - Applying a function to each group independently
  - Combining the results into a data structure
- Group by either the A or B columns or both

```
grouped = df.groupby('A')
grouped = df.groupby(['A', 'B'])
```

- Grouping and then applying a function sum to the resulting groups

```
 df = pd.DataFrame({'A' : ['foo', 'bar', 'foo', 'bar','foo', 'bar', 'foo',
'foo'],'B' : ['one', 'one', 'two', 'three','two', 'two', 'one', 'three'],'C' :
np.random.randn(8),'D' : np.random.randn(8)})
df.groupby('A').sum()
df.groupby(['A','B']).sum()
```

- For more information
  - http://pandas.pydata.org/pandas-docs/stable/groupby.html#groupby

Getting Data In/Out

- Reading from a csv file and writing to a csv file

```
df=pd.read_csv('foo.csv')
df.to_csv('foo.csv')
```

- Reading from a excel file and writing to a excel file

```
df=pd.read_excel('foo.xlsx', 'Sheet1', index_col=None,
na_values=['NA'])
df.to_excel('foo.xlsx', sheet_name='Sheet1')
```

Reading Material

- Getting started
  - URL: https://pandas.pydata.org/pandas-docs/stable/getting_started/index.html