

AI6102: Machine Learning Methodologies & Applications

L2: Data & Operations

Prof Dacheng Tao

dacheng.tao@ntu.edu.sg

Nanyang Technological University, Singapore

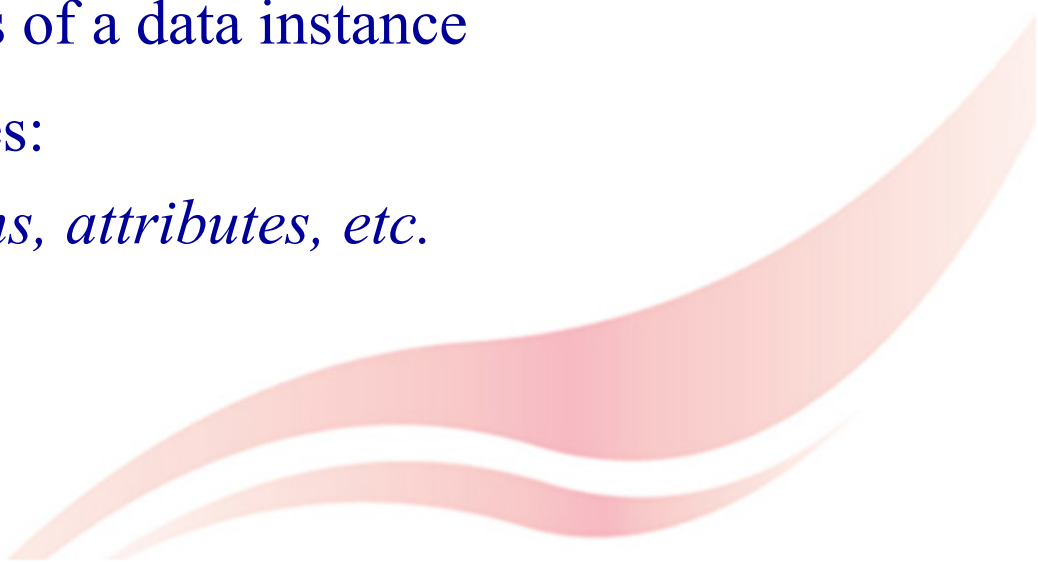
Outline

- Types of data
- Feature engineering
- Data operations



What is Data?

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
...
10	M	Student	8k	5k	No


- Data sets are made up of data instances
 - A data instance represents an “entity”
 - Alternative names of data instances:
examples, data objects, data points, etc.
 - Data instances are described/represented by features that capture the basic properties of a data instance
 - Alternative names of features:
variables, fields, dimensions, attributes, etc.
- 

Feature Values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
...
10	M	Student	8k	5k	No

- Feature values are numbers or symbols assigned to a feature
- Distinction between features and feature values
 - Same feature can be mapped to different feature values
 - Example: height can be measured in feet or meters
 - Different features can be mapped to the same set of values
 - Example: feature values for year and age are integers
 - But properties of feature values can be different
 - Year has no limit but age has a maximum and minimum value

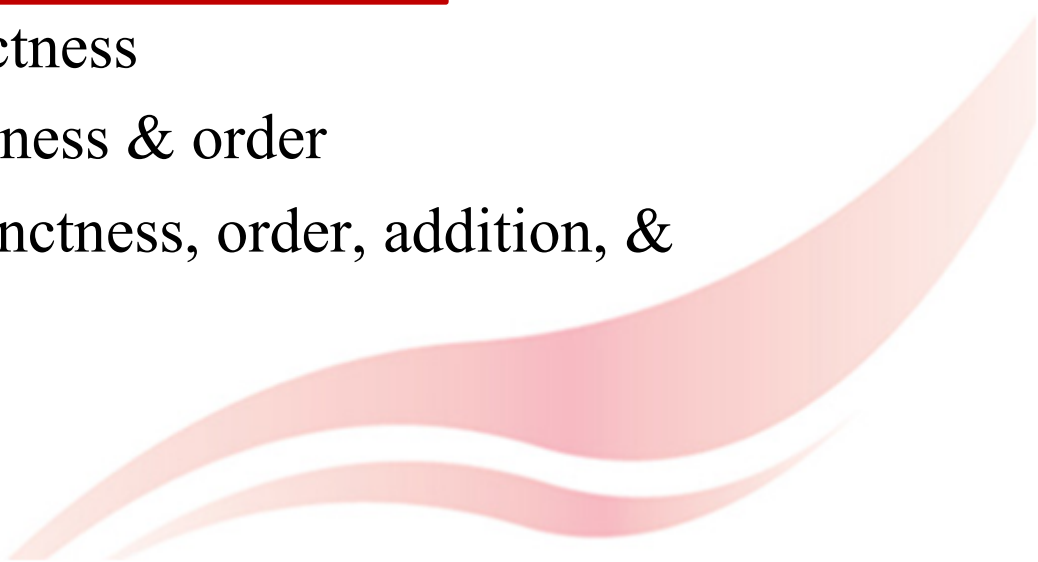
Types of Features

- Categorical
 - Nominal: has no intrinsic ordering to its categories
 - Examples: ID numbers, color, zip codes
 - Ordinal: has a clear ordering
 - Examples: grades in {A, B, C, F}, height in {tall, medium, short}
 - Numerical
 - The differences between values are interpretable
 - Examples: length, time, counts
- 

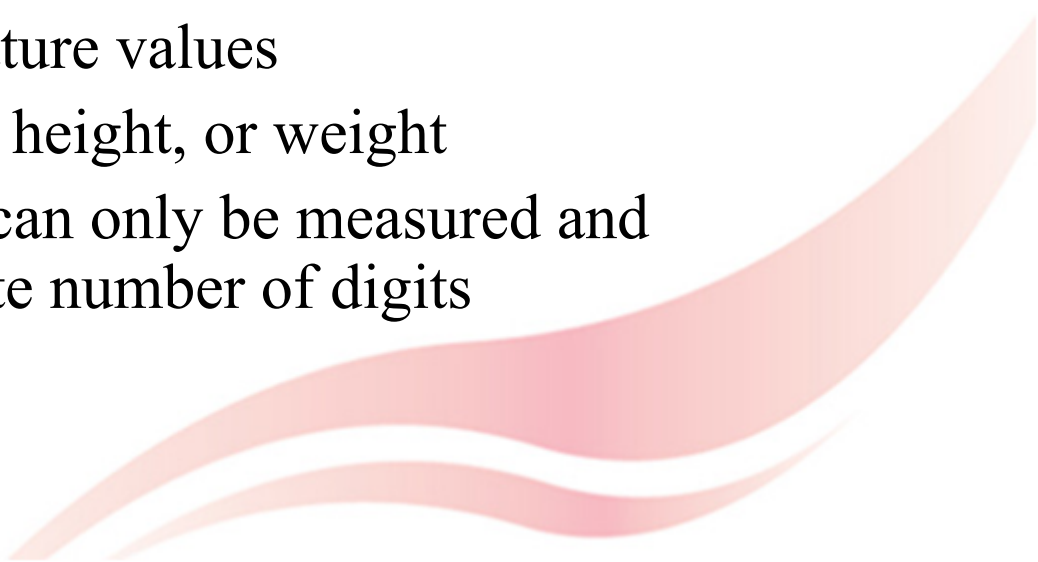
Properties of Feature Values

- The type of a feature depends on which of the following properties (operations) it possesses:


1) Distinctness:	$=$ and \neq
2) Order:	$<$, \leq , $>$ and \geq
3) Addition:	$+$ and $-$
4) Multiplication:	\times and $/$

- Nominal feature: distinctness
 - Ordinal feature: distinctness & order
 - Numerical feature: distinctness, order, addition, & multiplication
- 

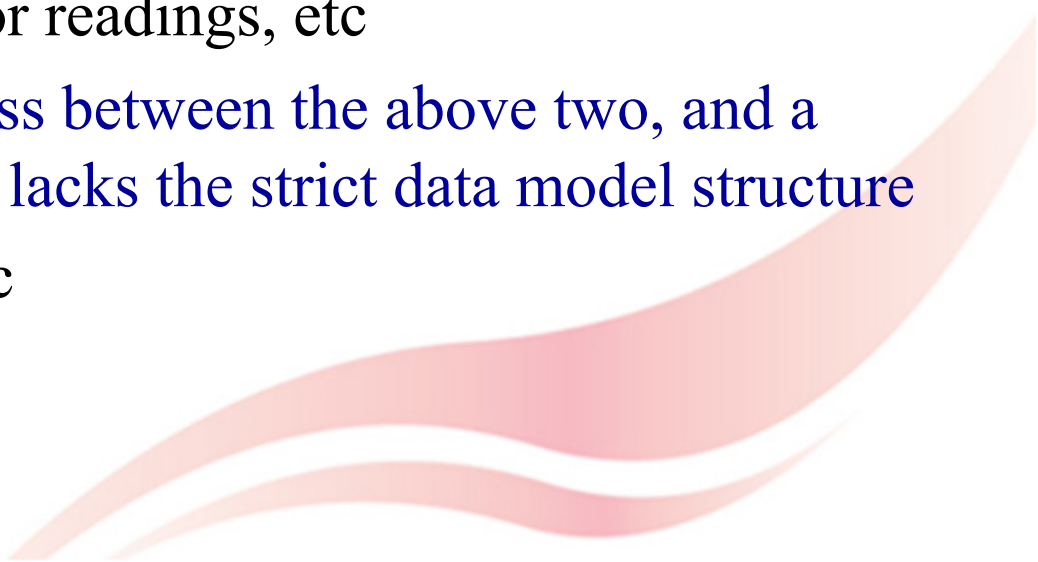
Alternative Categorization

- Distinguished by number of values
 - Discrete Feature
 - Has only a finite or countably infinite set of values
 - Examples: zip codes, counts, etc.
 - Often represented as integer variables
 - Continuous Feature
 - Has real numbers as feature values
 - Examples: temperature, height, or weight
 - Practically, real values can only be measured and represented using a finite number of digits
- 

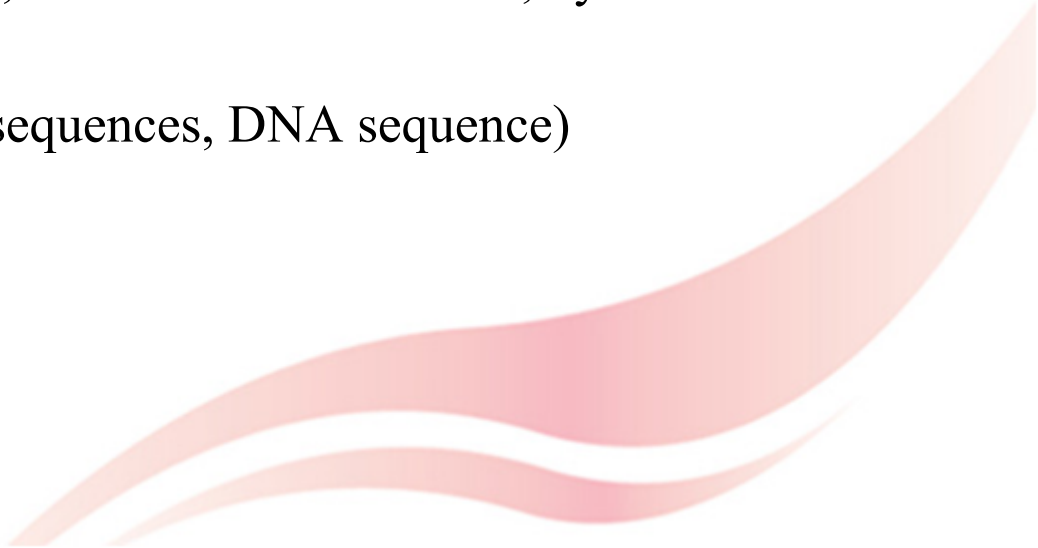
Binary Features

- A special case of discrete features
 - Nominal feature with only 2 states (e.g., 0 and 1)
 - Symmetric binary: both outcomes equally important
 - e.g., gender
 - Asymmetric binary: outcomes not equally important.
 - e.g., medical test (positive vs. negative)
 - Convention: assign 1 to most important outcome (e.g., COVID-19 positive)
- 

Types of Data

- Structured data: data that adheres to a pre-defined data model (structure of data)
 - E.g., spreadsheets, transaction records, etc
 - Unstructured data: information that neither has a pre-defined data model (structure of data) nor is organized in a pre-defined manner
 - E.g., text, images, sensor readings, etc
 - Semi-structured data: a cross between the above two, and a type of structured data, but lacks the strict data model structure
 - E.g., webpages, xml, etc
- 

Some Specific Types of Data

- Record
 - Relational records, Data matrix, Transaction data
 - Graph & Network
 - Webpages in WWW, Social networks, Molecular structures
 - Order
 - Time series data (video data, real-time financial data, dynamic sensor readings)
 - Sequence data (transaction sequences, DNA sequence)
 - Spatial
 - Maps, Sensor networks
- 

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of features

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

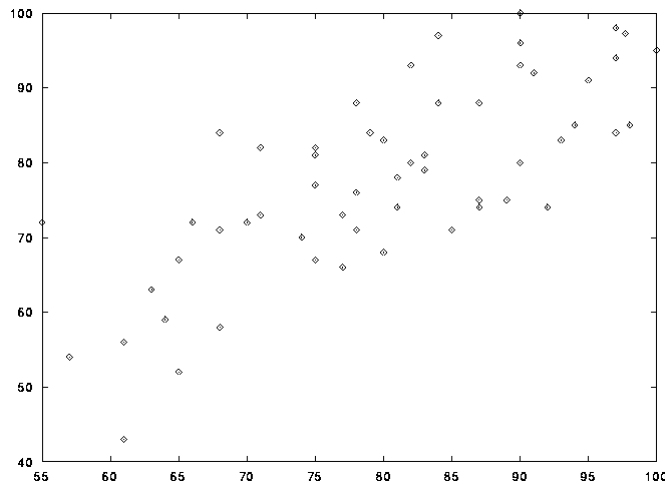
Transaction Data

- A special type of record data, where
 - Each record (transaction) involves a set of items
 - For example, consider a supermarket. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items

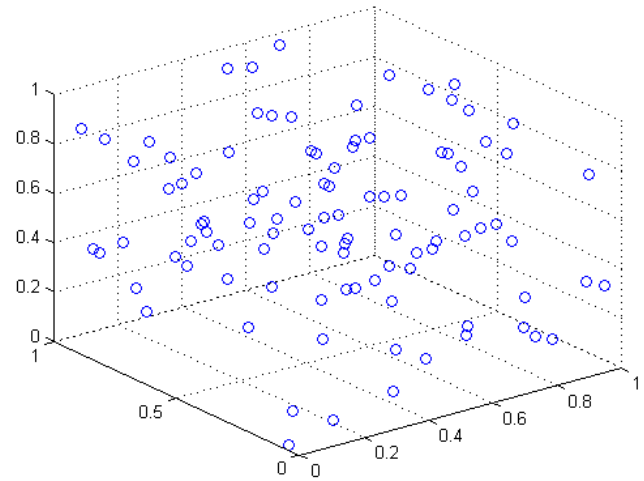
TID	Items
1	Egg, Coke, Milk, Rice, Oil
2	Coke, Bread
3	Rice
4	Milk, Coke, Egg
5	Bread, Egg

Data Matrix

- Data instances have the same fixed set of numerical features
- Each data instance can be thought of as a point in a multi-dimensional space, where each dimension represents a distinct feature



2D



3D

Data Matrix

- Such a dataset can be represented by a $N \times m$ matrix, where there are N rows, one for each data instance, and m columns, one for each feature
 - Or by a $m \times N$ matrix, where each column corresponds a data instance and each row corresponds a feature

ID	Age	Weight	Height
1	25	65	175
2	40	80	178

2×3 matrix

A grayscale image
of 28×20 pixels

20			
0	0	...	87
12	0	...	79
...
255	223	...	0

28

0 for black, 255 for white, values in between make up the different shades of gray

Sparse Data Matrix

- A special case of data matrix
- In a recommender system, users' ratings on products can be represented by a sparse matrix or a binary sparse matrix (only like or dislike information is stored)

	Item 1	Item 2	...	Item M
User 1	1	?	5	?
User 2	?	1	?	2
...
User N	?	?	4	?

Ratings: $5 > 4 > 3 > 2 > 1$ (Ordinal)

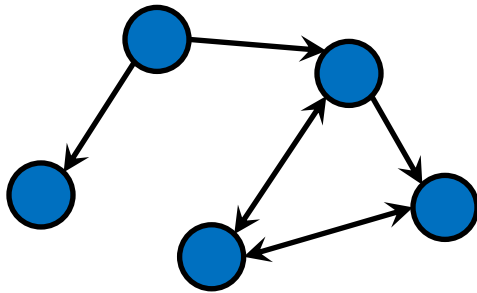
	Item 1	Item 2	...	Item M
User 1	1	?	1	?
User 2	?	0	?	0
...
User N	?	?	1	?

1: like, 0: dislike

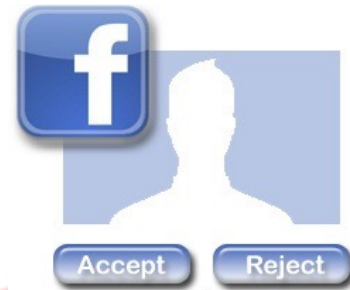
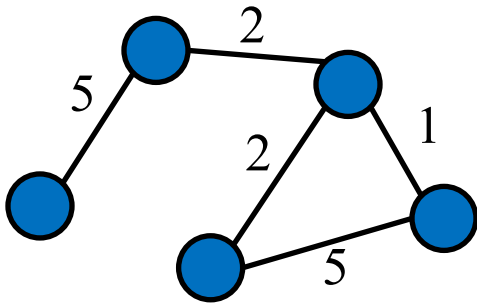
Graph Data

Each data instance is linked to some other data instance(s), and the whole dataset forms a graph

Directed Graph



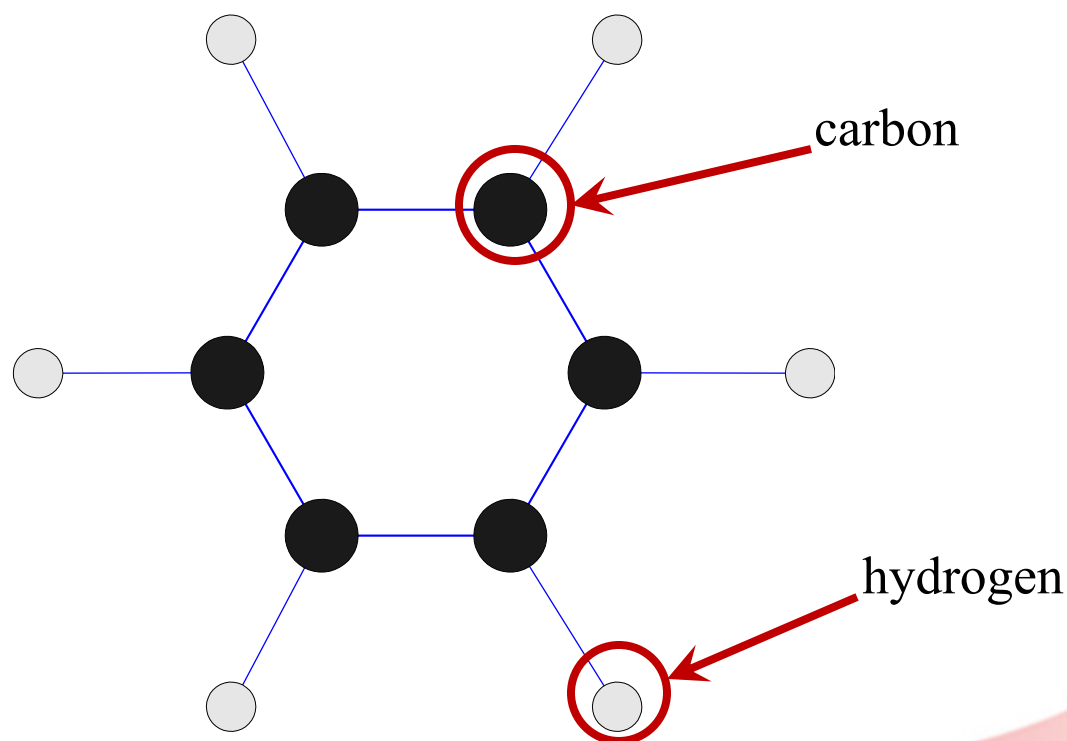
Undirected Graph



Graph Data (cont.)

Benzene Molecule: C_6H_6

Each data instance
itself is a graph

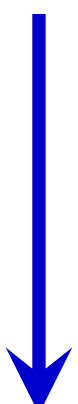


A ball-and-stick diagram of the chemical compound Benzene

Order Data – Sequence

- Sequence transactions

Timeline



Time	Customer	Item Purchased
T1	C1	A, B
T2	C3	A, C
T2	C1	C, D
T3	C2	A, D
T4	C2	E
T5	C1	A, E

Customer	Item Purchased
C1	(T1: A, B) (T2: C, D) (T5: A, E)
C2	(T3: A, D) (T4: E)
C3	(T2: A, C)

A sequence

Order Data – Sequence (cont.)

- Genomic sequence data

- Example: a section of the human genetic code expressed using the four nucleotides from which all DNA is constructed: **A**, **T**, **G**, and **C**

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```



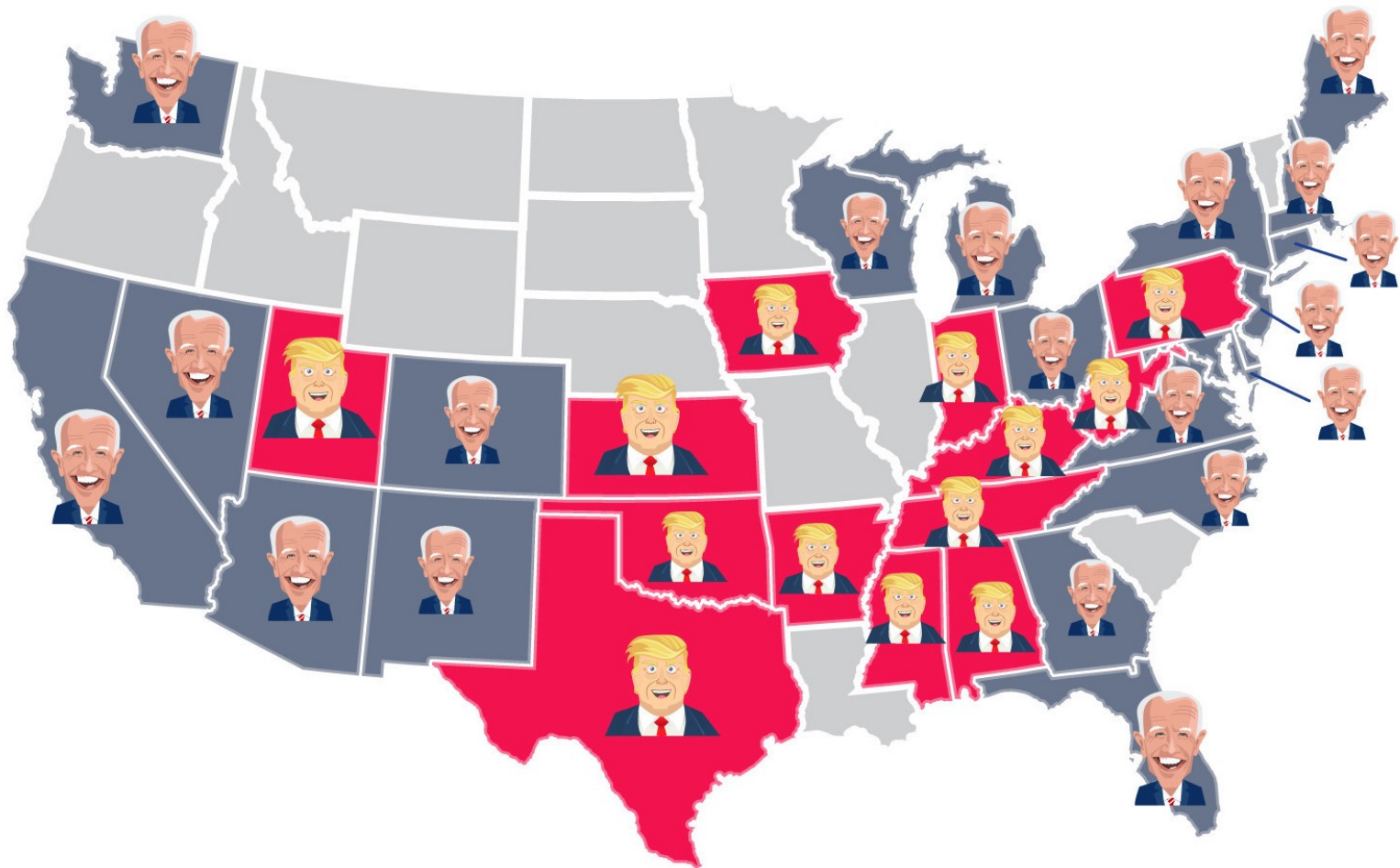
Ordered Data – Time Series

- A special type of sequence data in which each record is a time series, i.e., a series of measurements over (continuous) time.
 - Example: a time series of prices of a stock over days/months/years



Spatial Data

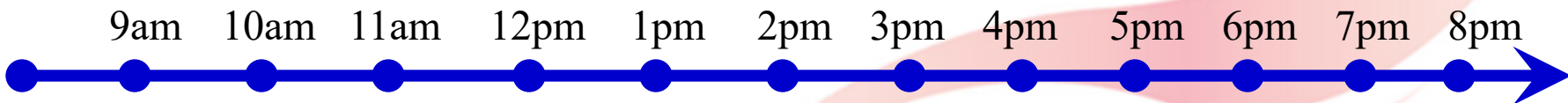
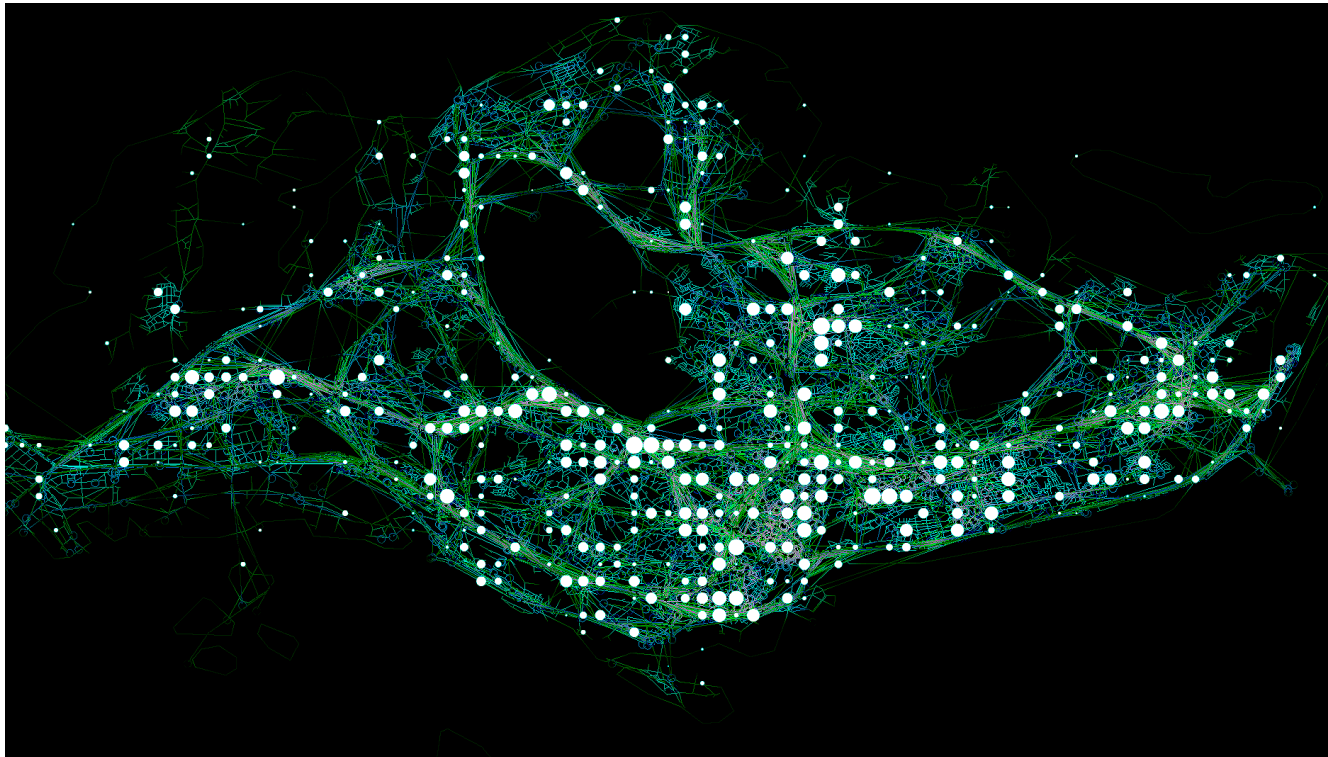
2020 US presidential elections



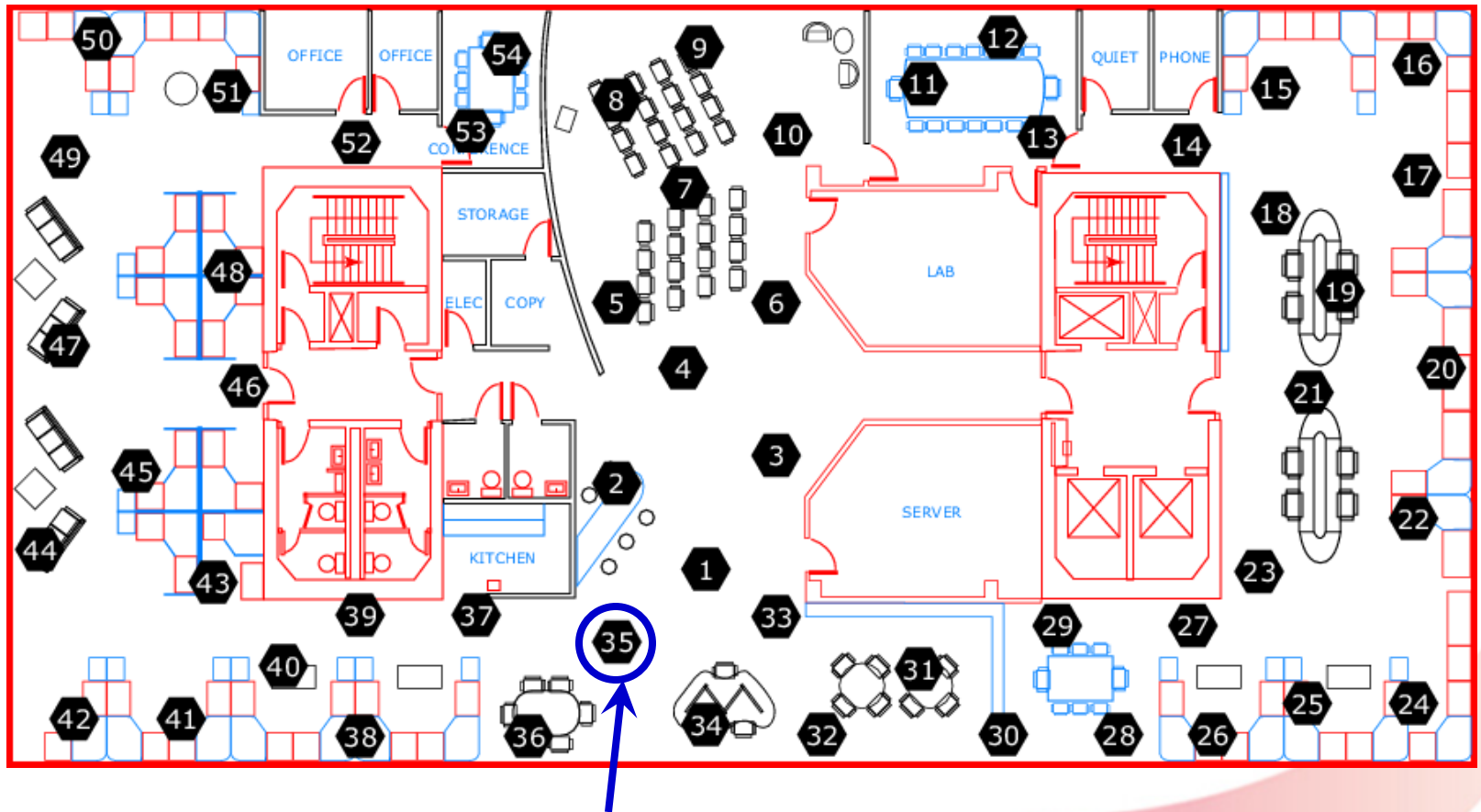
This image is downloaded from <https://www.prnewswire.com/>

Spatio-Temporal Data

Maps of taxi trajectories over time




Spatio-Temporal Data (cont.)



Sensors to monitor temperature, humidity, light, and voltage

Outline

- Types of data
 - Feature engineering
 - Data operations
- 
- A decorative graphic consisting of several overlapping, wavy, curved lines in shades of light pink and peach, located in the bottom right corner of the slide.

Feature Engineering

- The process of using domain knowledge and experience to construct features from raw data such that the performance of machine learning algorithms can be improved
- Note: feature engineering is an “engineering” process, and there is no “formula” telling you how to do it
 - Feature cleaning
 - Feature aggregation
 - Feature construction
 - Feature transformation
 - Feature normalization & discretization

Trial and error



Feature Cleaning

- Data in the real world is dirty: lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
 - Incomplete (missing): lacking features values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	N/A	20k	Yes
...

- Noisy: containing noise, errors, or outliers

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	-10k	30k	Yes
2	M	Student	10k	20k	Yes
...

Dealing with Missing Values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	N/A	20k	Yes
...

- Eliminate the whole data instances
- Not effective when the % of data instances containing missing values is large

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	N/A	200k	Yes
2	M	Student	N/A	20k	Yes
3	M	N/A	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

Dealing with Missing Values (cont.)

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	N/A	20k	Yes
...

- Eliminate the feature that consists missing values
- Not effective when the % of features containing missing values is large
- Not effective when the features containing missing values are important to the machine learning task

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	N/A	Yes
2	M	Student	N/A	20k	Yes
3	M	N/A	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

Dealing with Missing Values (cont.)


- Estimate missing values
 - Fill in the missing value manually based on prior knowledge
 - Fill in the missing value automatically
 - the feature mean/median
 - the value of other similar data objects
 - the mode

The diagram shows a table with 5 rows and 6 columns: ID, Gender, Profession, Income, Saving, and Repay. Row 1: ID 1, Gender F, Profession Engineer, Income 60k, Saving N/A, Repay Yes. Row 2: ID 2, Gender M, Profession Student, Income N/A, Saving 20k, Repay Yes. Row 3: ID 3, Gender M, Profession N/A, Income 56k, Saving 100k, Repay Yes. Row 4: ID 4, Gender F, Profession Student, Income 12k, Saving 15k, Repay Yes. Row 5: ID 5, Gender M, Profession Lawyer, Income 80k, Saving 60k, Repay No. Annotations: A blue box highlights rows 2 and 4, which are both 'Student' professionals. A blue arrow points from the '12k' value in row 4 to the 'N/A' value in row 2. A red arrow points from the 'N/A' value in row 1 to a calculation: $\frac{20 + 100 + 15 + 60}{4} = 48.75k$. The word 'similar' is written in blue next to the blue box. The value '12k' is also written in blue above the 'N/A' in row 2.

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	N/A	Yes
2	M	Student	N/A	20k	Yes
3	M	N/A	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

The mode: Student

Dealing with Noisy Values

- Define some rules, e.g., if the value is $>$ the reasonably maximal value, then set it to be the reasonably maximal value
 - Similar approaches as dealing with missing values
 - Eliminate the whole data instances
 - Eliminate the features that consists missing values
 - Estimate missing values
- 

Feature Aggregation

- Combining two or more features or feature values into a single feature or feature value
- Example 1: For a feature “Location”, the dataset originally stores “cities”
 - There are a huge amount of distinct values (cities), and a lot of them may only appear one or two time(s)
 - Rescale (aggregation) the values to states, provinces or countries

ID	Location
1	New York
2	Modesto
3	Los Angeles
4	Buffalo
5	Chicago
6	Anaheim
7	Los Angeles
8	New York
9	Chicago
10	Chicago

Aggregation

ID	Location
1	NY
2	CA
3	CA
4	NY
5	IL
6	CA
7	CA
8	NY
9	IL
10	IL

Feature Aggregation (cont.)

- Example 2: Stock price over time
 - To analyze more coarse-grained patterns, the “hour price” features can be aggregated to “day price”, “month price” or “year price”

Stock ID	Jul 1 10am	Jul 1 11am	...	Jul 1 4pm	Jul 2 10am	...	Aug 1 10am	...	Sept 1 10am	...	Oct 1 10am	...	Dec 31 4pm
1001	10.5	10.8	...	10.6	10.7	...	8.5	...	11.6	...	13.5	...	12.7
1050	46.3	50.2	...	49.3	48.5	...	55.6	...	54.6	...	54.1	...	59.6
...
2055	101.2	99.5	...	100.6	100.1	...	97.3	...	94.5	...	88.2	...	85.6

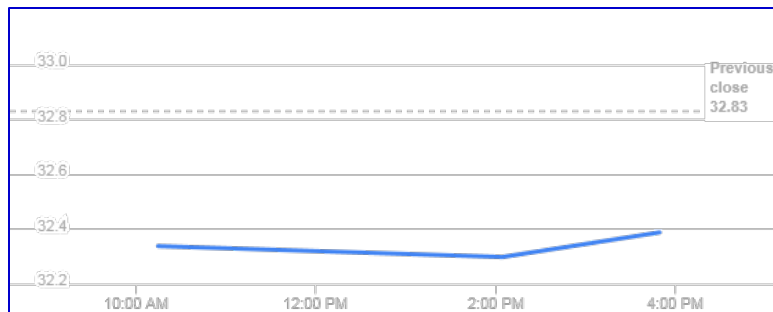
Aggregation



Stock ID	Jul	Aug	Sept	Oct	Nov	Dec
1001	10.6	9.4	11.4	13.4	13.1	12.6
1050	48.2	54.8	53.7	53.1	57.9	59.3
...
2055	100.1	98.5	94.9	87.6	89.7	84.9

Feature Aggregation (cont.)

- Example 2: Stock price over time



Over 1 day (unit: hour)



Over 1 month (unit: day)



Over 1 year (unit: month)



Over 20+ years (unit: year)

Features Construction

- To create new features to capture more important information of the data than the original features for a specific task

income-saving ratio

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
...
10	M	Student	8k	5k	No



ID	Gender	Profession	Income	Saving	I:S Ratio	Repay
1	F	Engineer	60k	200k	3/10	Yes
2	M	Student	10k	20k	1/2	Yes
...
10	M	Student	8k	5k	8/5	No

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height (m)}^2}$$


ID	Age	Weight	Height	...	Healthy
1	25	65	175	...	Yes
2	40	80	178	...	No
...




ID	Age	Weight	Height	BMI	...	Healthy
1	25	130	175	21.22	...	Yes
2	40	160	178	25.24	...	No
...

Features Construction (cont.)

ID	Expiry Date
1	13/08/2020
2	20/04/2018
...	...
10	04/07/2022




ID	Expiry Date Day	Expiry Date Month	Expiry Date Year
1	13	8	2020
2	20	4	2018
...
10	4	7	2022



ID	Expiry Date Day	Expiry Date Month	Expiry Date Year	Expired?	# Expired days
1	13	8	2020	Yes	10
2	13	8	2018	Yes	740
...
10	4	7	2022	No	0

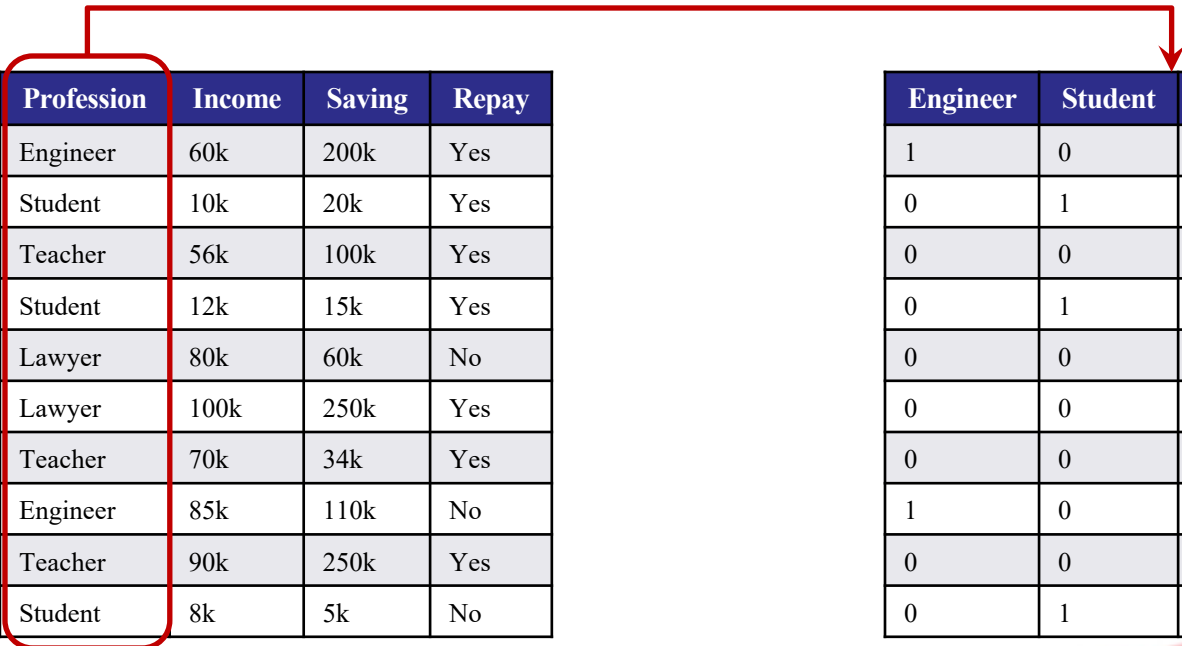
Using current
date information

Feature Transformation

- For most supervised learning algorithms, each input data instance needs to be represented by a numerical vector \mathbf{x}_i of a fixed dimension (e.g., m)
 - Categorical features \rightarrow one-hot encoding
 - Unstructured data \rightarrow feature vector
- 

One-hot Encoding

- Transform a feature of k distinct categorical values to k numerical features of binary values (0/1)



The diagram illustrates the process of one-hot encoding. A red box highlights the 'Profession' column in the first table. A red arrow points from this box to the second table, which shows the resulting one-hot encoded binary columns for each profession: Engineer, Student, Teacher, and Lawyer.

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Engineer	Student	Teacher	Lawyer
1	0	0	0
0	1	0	0
0	0	1	0
0	1	0	0
0	0	0	1
0	0	0	1
0	0	1	0
1	0	0	0
0	0	1	0
0	1	0	0

Why One-hot Encoding?

Engineer: 1
Student: 2
Teacher: 3
Lawyer: 4



Numerical values

ID	Profession
1	Engineer
2	Student
3	Teacher
5	Lawyer

ID	Profession
1	1
2	2
3	3
5	4

- Distance between IDs 1 & 2 (Engineer v.s. Student): 1
- Distance between IDs 1 & 5 (Engineer v.s. Lawyer): 3
- Each distinct values should be equally important
- The distance between them should be the same after transformation

Using one-hot encoding

ID	Engineer	Student	Teacher	Lawyer
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
5	0	0	0	1

Distances between IDs 1, 2, 3 and 5 are all $\sqrt{2}$

Binary Features

Unnecessary

Using one-hot encoding?



ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Female	Male
1	0
0	1
0	0
0	1
0	1
0	1
1	0
1	0
0	1
0	1

Female: 1
Male: 0

Gender
1
0
0
1
0
0
1
0
0
0

Distance between two same categories is 0
Distance between two distinct categories is 1

Extension of One-hot Encoding

Each distinct item over all the transactions
is used to construct a binary feature

TID	Items
1	Egg, Coke, Milk, Rice, Oil
2	Coke, Bread
3	Rice
4	Milk, Coke, Egg
5	Bread, Egg

ID	Bread	Coke	Egg	Milk	Oil	Rice
1	0	1	1	1	1	1
2	1	1	0	0	0	0
3	0	0	0	0	0	1
4	0	1	1	1	0	0
5	1	0	1	0	0	0

Unstructured Data – Text

Doc1	Compact; easy to operate; very good picture quality; looks sharp!
Doc2	It is also quite blurry in very dark settings. I will never_buy HP again.
...	...

Scan through the whole training dataset once to build a [dictionary](#)

F1	F2	F3	F4	F5	F6	...
compact	easy	quite	blurry	good	never_buy	...

	F1	F2	F3	F4	F5	F6	...
Doc1	1	1	0	0	1	0	...
Doc2	0	0	1	1	0	1	...
...

Bag-of-words representation

Unstructured Data – Images

Grayscale image



20×30



0	0	...	87
12	0	...	79
...
255	223	...	0


20×30



Concatenating rows to
construct a single vector
of $20 \times 30 = 600$
dimensions

- Use image processing algorithms to detect and isolate various desired portions or shapes (features) of an image
 - The scale-invariant feature transform (SIFT) is a feature detection algorithm to detect and describe local features in images
 - SIFT keypoints of objects are first extracted from the training dataset to construct a visual “words” dictionary
 - Bag-of-(visual)-words representation is used to represent each image

Feature Normalization & Discretization

- Normalization
 - A function that maps the entire set of values of a given feature to a pre-defined and specified-range new set of replacement values such that each old value can be identified with one of the new values
 - Min-max normalization
 - Standardization (z-score normalization)
 - Discretization
 - Divide the range of a continuous features into intervals
- 

Min-Max Normalization

- To rescale values to $[\min_{new}, \max_{new}]$
 - e.g. to normalize saving ranging from 5k to 250k to $[0.0, 1.0]$. What is the value for 100k after normalization?

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$$v_{new} = \frac{v_{old} - \min_{old}}{\max_{old} - \min_{old}} (\max_{new} - \min_{new}) + \min_{new}$$

$$100k \rightarrow \frac{100k - 5k}{250k - 5k} (1.0 - 0) + 0 = 0.388$$

Standardization

- Also known as z-score normalization, rescale values such that the mean of new values is 0, and the standard deviation is 1 (μ : mean, σ : standard deviation)
 - e.g. the mean of saving is $\mu = 104.4\text{k}$, and the standard deviation of saving $\sigma = 91.38\text{k}$. What is the value for 100k after standardization?

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$$v_{new} = \frac{v_{old} - \mu_{old}}{\sigma_{old}} \quad \longrightarrow \quad \mu_{new} = 0, \text{ and } \sigma_{new} = 1$$

$$\frac{100 - 104.4}{91.38} = -0.05$$

Discretization

- Some classification algorithm do not prefer continuous features (potentially a lot of distinct values)
- Solution: to discretize values of a continuous feature into intervals, interval “labels” are used to replace values
 - Binning
 - Binarization



Binning: Equal-frequency

- Divides the range into K intervals, each containing approximately same number of data

ID	F1
1	4
2	34
3	9
4	21
5	8
6	26
7	29
8	10
9	25
10	24
11	28
12	21

Divide into 3 intervals

ID	F1
1	1
2	3
3	1
4	2
5	1
6	3
7	3
8	1
9	2
10	2
11	3
12	2

or

ID	F1
1	7.75
2	29.25
3	7.75
4	22.75
5	7.75
6	29.25
7	29.25
8	7.75
9	22.75
10	22.75
11	29.25
12	22.75

Sorted: 4, 8, 9, 10, 21, 21, 24, 25, 26, 28, 29, 34

1 or 7.75 2 or 22.75 3 or 29.25

Binning: Equal-frequency (cont.)

- Advantage
 - Data sizes of each interval are balanced
- Disadvantage
 - Variance of values in some interval(s) could be very large

ID	F1
1	2
2	4
3	27
4	21
5	30
6	26
7	30
8	33
9	25
10	24

Divide into 3 intervals: 3 : 3 : 4

2, 4, 21, 24, 25, 26, 27, 29, 30, 33

1

2

3

or 20

or 3

or 7

ID	F1
1	1
2	1
3	3
4	1
5	3
6	2
7	3
8	3
9	2
10	2

or

ID	F1
1	9
2	9
3	30
4	9
5	30
6	25
7	30
8	30
9	25
10	25

Binning: Equal-distance

- Divides the range into K intervals of equal size: uniform grid
- Denote by Max and Min the lowest and highest values of the feature, the width of intervals will be $\Delta = \frac{\text{Max} - \text{Min}}{K}$

ID	F1
1	4
2	34
3	9
4	21
5	8
6	26
7	29
8	10
9	25
10	24
11	28
12	21

Divide into 3 intervals

$$\Delta = \frac{34 - 4}{3} = 10$$

$[4, 14), [14, 24), [24, 34]$

1

2

3

4, 8, 9, 10, 21, 21, 24, 25, 26, 28, 29, 34

ID	F1
1	1
2	3
3	1
4	2
5	1
6	3
7	3
8	1
9	2
10	2
11	3
12	2

Binning: Equal-distance

- Advantage
 - The most straightforward, but outliers may dominate
- Disadvantage
 - The instance sizes of each interval would be highly imbalanced on skewed dataset

Divide into 3 intervals

$$\Delta = \frac{29 - 2}{3} = 9$$

[2, 11), [11, 20), [20, 29]

2,	12,	21, 24, 25, 25, 27, 28, 28, 29, 29
1	2	3

Binarization

- A special case of discretization
- To transform each numerical value of a feature to one of the binary values
- Set a threshold value T , if the feature value $\geq T$, then it is mapped to 1, otherwise, 0

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$T = 91k$



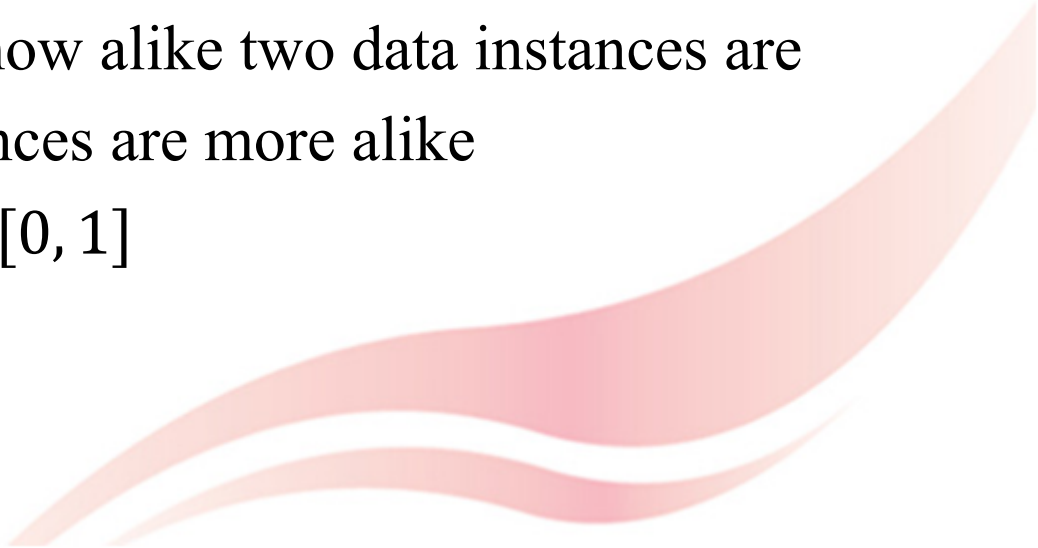
ID	Saving $\geq 91k$?
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	1
9	1
10	0

Outline

- Types of data
- Feature engineering
- Data operations
 - Proximity
 - Correlation

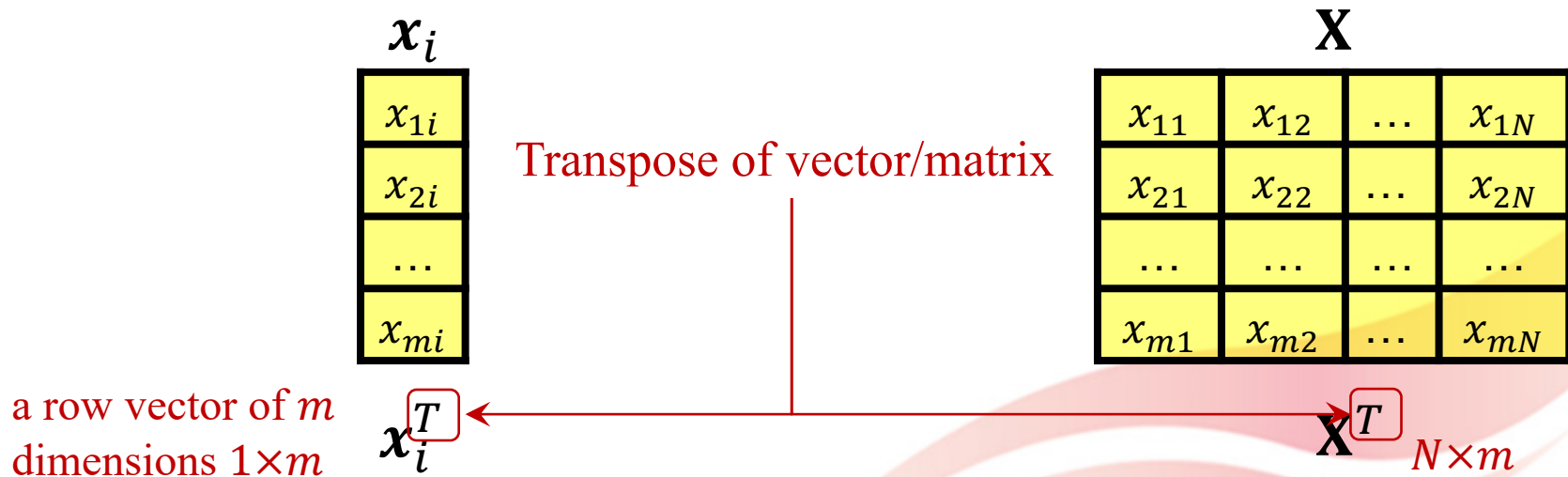


Proximity

- Distance (Dissimilarity)
 - Numerical measure of how different two data instances are
 - Lower when data instances are more alike
 - Minimum distance is 0
 - Upper limit varies
 - Similarity
 - Numerical measure of how alike two data instances are
 - Higher when data instances are more alike
 - Often falls in the range $[0, 1]$
- 

Notations

- For each m -dimensional data instance \mathbf{x}_i , we represent it by a column vector, i.e, $m \times 1$, where $x_{ki}, k = 1, \dots, m$ is the value of the k -th feature or dimension of the data instance \mathbf{x}_i
- Given a dataset of N data instances, each of which is m -dimensional, we represent it by a $m \times N$ matrix \mathbf{X} , where x_{ki} indicates the value of the k -th feature of the i -th instance



Euclidean Distance

- Given two m -dimensional data instances \mathbf{x}_i and \mathbf{x}_j , the Euclidean distance between them is defined as

$$\mathbf{x}_i$$

x_{1i}
x_{2i}
\dots
x_{mi}

$$\mathbf{x}_j$$

x_{1j}
x_{2j}
\dots
x_{mj}

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^m (x_{ki} - x_{kj})^2}$$

- A more compact form of the Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}$$

Inner product

Inner Product

- Given two m -dimensional data instances \mathbf{x}_i and \mathbf{x}_j , the inner product between them is defined as

$$\mathbf{x}_i \cdot \mathbf{x}_j = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{k=1}^m (x_{ki} \times x_{kj}) = \mathbf{x}_i^T \mathbf{x}_j$$

- The Euclidean distance can be rewritten as

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^m (x_{ki} - x_{kj})^2} = \sqrt{\sum_{k=1}^m ((x_{ki} - x_{kj}) \times (x_{ki} - x_{kj}))}$$

$$= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}$$

$$= \sqrt{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)} \quad \text{OR} \quad \sqrt{\langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle}$$

$\mathbf{x}_i - \mathbf{x}_j$

$x_{1i} - x_{1j}$
$x_{2i} - x_{2j}$
...
$x_{mi} - x_{mj}$

L2 Norm

- The Euclidean distance between \mathbf{x}_i and \mathbf{x}_j can be written as

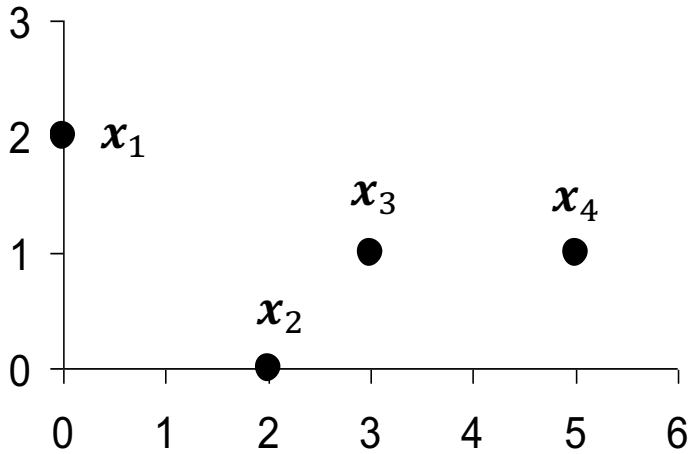
$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

$\|\mathbf{x}\|_2$ is known as the L2 norm of a m -dimensional vector \mathbf{x} , defined as $\|\mathbf{x}\|_2 = \sqrt{\sum_{k=1}^m x_k^2}$

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{k=1}^m (x_{ki} - x_{kj})^2}$$

Note: $\|\mathbf{x}\|_2$ can be viewed as the measure of Euclidean distance between \mathbf{x} and the origin $\mathbf{0}$

An Example



	X_1	X_2
x_1	0	2
x_2	2	0
x_3	3	1
x_4	5	1

	x_1	x_2	x_3	x_4
x_1	0	2.828	3.162	5.099
x_2	2.828	0	1.414	3.162
x_3	3.162	1.414	0	2
x_4	5.099	3.162	2	0

Distance matrix

Manhattan Distance

- Given two m -dimensional data instances \mathbf{x}_i and \mathbf{x}_j , the Manhattan distance between them is defined as

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^m |x_{ki} - x_{kj}|$$

- The Manhattan distance is also known as the L1-norm distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_1$$

$\|\mathbf{x}\|_1$ is known as the L1 norm of a m -dimensional vector \mathbf{x} , defined as $\|\mathbf{x}\|_1 = \sum_{k=1}^m |x_k|$

An Example

A hash table

Binary bits

	X_1	X_2	X_3
x_1	0	1	0
x_2	1	0	0
x_3	1	1	1
x_4	1	1	0

Distance matrix

	x_1	x_2	x_3	x_4
x_1	0	2	2	1
x_2	2	0	2	1
x_3	2	2	0	1
x_4	1	1	1	0

A Survey on Learning to Hash, Wang et al., TPAMI 2017

Common Properties of Distances

- Distances have some well known properties:
 - Positive definiteness:
 - $d(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ for any \mathbf{x}_i and \mathbf{x}_j and $d(\mathbf{x}_i, \mathbf{x}_j) = 0$ only if $\mathbf{x}_i = \mathbf{x}_j$
 - Symmetry:
 - $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$ for any \mathbf{x}_i and \mathbf{x}_j
 - Triangle inequality:
 - $d(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_k) + d(\mathbf{x}_k, \mathbf{x}_j)$ for any $\mathbf{x}_i, \mathbf{x}_j$ and \mathbf{x}_k
- A distance that satisfies these properties is a metric

Similarity

- Recall that distance also known as dissimilarity is to measure how different two data instances are, while similarity is to measure how alike two data instances are
- Distance can be simply revised to measure similarity, e.g,

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

where $s(\mathbf{x}_i, \mathbf{x}_j) \triangleq 1$ when $d(\mathbf{x}_i, \mathbf{x}_j) = 0$

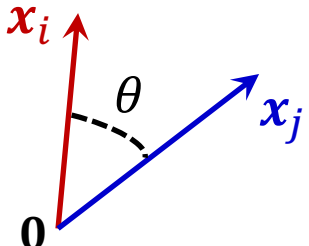
- In this way, for any \mathbf{x}_i and \mathbf{x}_j , normalize $s(\mathbf{x}_i, \mathbf{x}_j) \in (0, 1]$
- Set a threshold T : if $d(\mathbf{x}_i, \mathbf{x}_j) \geq T$, then $s(\mathbf{x}_i, \mathbf{x}_j) = 0$

Cosine Similarity

- Given two m -dimensional non-zero data instances \mathbf{x}_i and \mathbf{x}_j , the Cosine similarity between them is defined as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2} = \cos(\theta)$$

$\mathbf{x}_i \cdot \mathbf{x}_j = \sum_{k=1}^m (x_{ki} \times x_{kj}) = \|\mathbf{x}_i\|_2 \times \|\mathbf{x}_j\|_2 \times \cos(\theta)$



Angle between \mathbf{x}_i and \mathbf{x}_j

- The outcome of Cosine similarity is in $[-1, 1]$
- Cosine similarity is particularly used in positive space, i.e., \mathbf{x}_i and \mathbf{x}_j are of non-negative numerical values \rightarrow outcome of Cosine similarity is in $[0, 1]$

Similarity Properties

- Maximum: $s(\mathbf{x}_i, \mathbf{x}_j) = 1$ if $\mathbf{x}_i = \mathbf{x}_j$ (for normalized \mathbf{x} , iff)
- Symmetry: $s(\mathbf{x}_i, \mathbf{x}_j) = s(\mathbf{x}_j, \mathbf{x}_i)$ for any \mathbf{x}_i and \mathbf{x}_j
- A general way to change distance to similarity is to define a strictly monotone decreasing function $f(x)$:

$$\text{similarity} = f(\text{distance})$$

- Some commonly used forms of the function $f(x)$ include

$$f(x) = \frac{1}{x + b}, \text{ where } b \geq 0 \text{ is a parameter}$$

$$f(x) = e^{-x^b}, \text{ where } b > 0 \text{ is a parameter}$$


Feature Correlation

\mathbf{X}

	x_1	x_2		x_N
X_1	x_{11}	x_{12}	...	x_{1N}
X_2	x_{21}	x_{22}	...	x_{2N}

X_m	x_{m1}	x_{m2}	...	x_{mN}

Similarity or distance is to measure the relationship between data instances, i.e., the columns of the data matrix \mathbf{X}

Feature correlation is to measure the relationship between features e.g, what is the relationship between height and weight?

Given a data matrix \mathbf{X} , each feature X_i can be represented by the corresponding column of the matrix

Pearson Correlation Coefficient

- Pearson correlation coefficient (PCC) is a statistical measure that calculates the linear relationship between two features (or continuous variables)
- Its outcome is in $[-1, +1]$
 - $+1$ means the two features have a perfectly positive linear correlation
 - 0 means that there is no linear correlation between them
 - -1 means they have a perfectly negative linear correlation

$$\text{Pearson}(X_i, X_j) = \frac{\mathbb{E}[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]}{\sigma_{X_i} \times \sigma_{X_j}}$$

where σ_{X_i} and σ_{X_j} are the standard deviations of X_i and X_j , respectively

PCC (cont.)

		x_1	x_2	x_N
\mathbf{X}	X_i
		x_{i1}	x_{i2}	...
	
	X_j	x_{j1}	x_{j2}	...
	

- In practice, PCC between X_i and X_j can be calculated as

$$\text{Person}(X_i, X_j) = \frac{\sum_{k=1}^N \left((x_{ik} - \hat{\mu}_{X_i}) \times (x_{jk} - \hat{\mu}_{X_j}) \right)}{\sqrt{\sum_{k=1}^N (x_{ik} - \hat{\mu}_{X_i})^2} \sqrt{\sum_{k=1}^N (x_{jk} - \hat{\mu}_{X_j})^2}}$$

where $\hat{\mu}_{X_i}$ and $\hat{\mu}_{X_j}$ are the (unbiased) sample means of the features X_i and X_j , respectively.

$$\hat{\mu}_{X_i} = \frac{1}{N} \sum_{k=1}^N x_{ik}$$

PCC (cont.)

	x_1	x_2	x_N	
\mathbf{X}
	x_{i1}	x_{i2}	...	x_{iN}

	x_{j1}	x_{j2}	...	x_{jN}

$$\begin{aligned}
 \text{Person}(X_i, X_j) &= \frac{\sum_{k=1}^N \left((x_{ik} - \hat{\mu}_{X_i}) \times (x_{jk} - \hat{\mu}_{X_j}) \right)}{\sqrt{\sum_{k=1}^N (x_{ik} - \hat{\mu}_{X_i})^2} \sqrt{\sum_{k=1}^N (x_{jk} - \hat{\mu}_{X_j})^2}} \\
 &= \frac{\sum_{k=1}^N \left((x_{ik} - \hat{\mu}_{X_i}) \times (x_{jk} - \hat{\mu}_{X_j}) \right)}{(N-1) \sqrt{\frac{\sum_{k=1}^N (x_{ik} - \hat{\mu}_{X_i})^2}{N-1}} \sqrt{\frac{\sum_{k=1}^N (x_{jk} - \hat{\mu}_{X_j})^2}{N-1}}} \\
 &= \frac{\sum_{k=1}^N \left((x_{ik} - \hat{\mu}_{X_i}) \times (x_{jk} - \hat{\mu}_{X_j}) \right)}{(N-1) \times \hat{\sigma}_{X_i} \times \hat{\sigma}_{X_j}}
 \end{aligned}$$

where $\hat{\sigma}_{X_i}$ and $\hat{\sigma}_{X_j}$ are the (unbiased) sample standard deviations of the features X_i and X_j , respectively

$$\hat{\sigma}_{X_i} = \sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_{ik} - \hat{\mu}_{X_i})^2}$$

PCC (cont.)

	x_1	x_2	x_N	
\mathbf{X}
	x_{i1}	x_{i2}	...	x_{iN}

	x_{j1}	x_{j2}	...	x_{jN}

$$\text{Person}(X_i, X_j) = \frac{\sum_{k=1}^N \left((x_{ik} - \hat{\mu}_{X_i}) \times (x_{jk} - \hat{\mu}_{X_j}) \right)}{(N-1) \times \hat{\sigma}_{X_i} \times \hat{\sigma}_{X_j}}$$

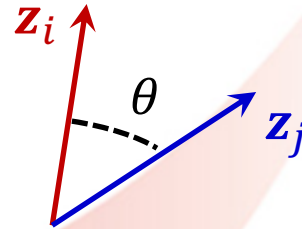
$$= \frac{1}{N-1} \sum_{k=1}^N \left(\left(\frac{x_{ik} - \hat{\mu}_{X_i}}{\hat{\sigma}_{X_i}} \right) \times \left(\frac{x_{jk} - \hat{\mu}_{X_j}}{\hat{\sigma}_{X_j}} \right) \right)$$

Standardization on feature X_i

Standardization on feature X_j

$$\mathbf{z}_i = \begin{pmatrix} x'_{i1} \\ x'_{i2} \\ \dots \\ x'_{iN} \end{pmatrix}$$

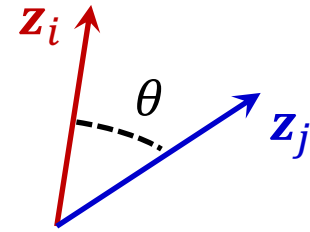
$$\mathbf{z}_j = \begin{pmatrix} x'_{j1} \\ x'_{j2} \\ \dots \\ x'_{jN} \end{pmatrix}$$



$$\text{Person}(X_i, X_j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{N-1} = \frac{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2}{N-1} \times \cos(\theta) = \frac{N-1}{N-1} \cos(\theta) = \cos(\theta)$$

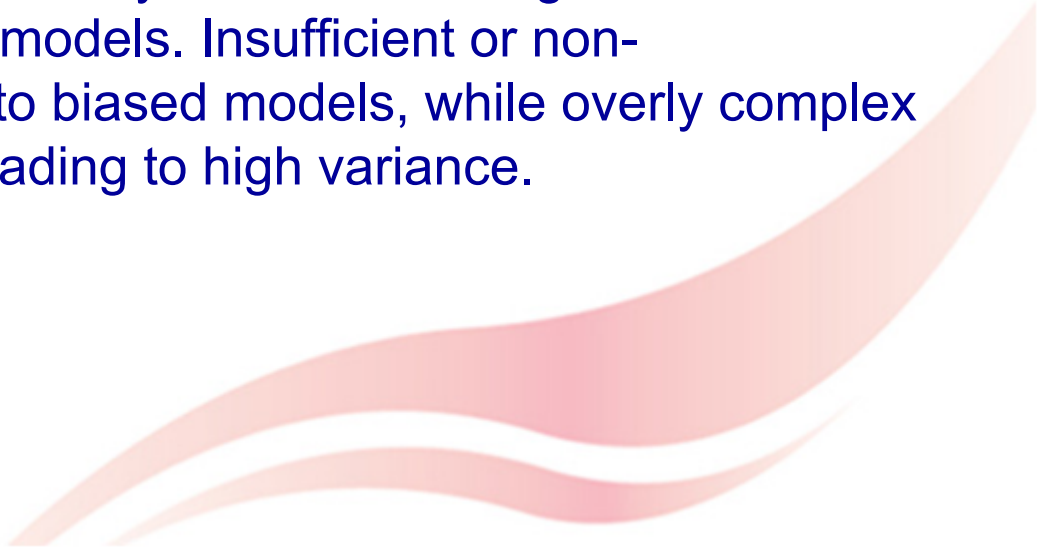
$$\text{As } \sqrt{\frac{1}{N-1} \sum_{k=1}^N (x'_{ik} - 0)^2} = 1, \text{ thus } \|\mathbf{z}_i\|_2 = \sqrt{\sum_{k=1}^N x'^2_{ik}} = \sqrt{N-1}$$

PCC (cont.)



- If $\text{Person}(X_i, X_j) > 0$, X_i and X_j are positively correlated: X_i 's values increase (or decrease) as X_j 's values increase (or decrease) and vice versa
 - The higher the value, the stronger the positive correlation
 - Maximum value: $+1$ when $\theta = 0^\circ$,
- If $\text{Person}(X_i, X_j) = 0$, there is no correlation between values of X_i and X_j ($\theta = 90^\circ$,)
- If $\text{Person}(X_i, X_j) < 0$, X_i and X_j are negatively correlated: X_i 's values increase (or decrease) as X_j 's values decrease (or increase) and vice versa
 - The lower the value, the stronger the negative correlation
 - Minimum value: -1 when $\theta = 180^\circ$

Key Roles of Data

- **Training:** Data is used to train machine learning models.
 - **Evaluation:** After training, the model's performance is evaluated on separate data (e.g., validation or test data) to assess its ability to generalize.
 - **Feature Engineering:** The data directly influences feature engineering, where raw data is transformed into features that can improve model performance.
 - **Bias and Variance:** Data plays a key role in balancing bias and variance in machine learning models. Insufficient or non-representative data can lead to biased models, while overly complex data may cause overfitting, leading to high variance.
- 

Thank you!

