


Data Preprocessing

- Data Preprocessing: An Overview 
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
 - Accuracy: correct or wrong, accurate or not
 - Completeness: not recorded, unavailable, ...
 - Consistency: some modified but some not, dangling, ...
 - Timeliness: timely update?
 - Believability: how trustable the data are correct?
 - Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

- **Data cleaning**

- Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

- **Data integration**

- Integration of multiple databases, data cubes, or files


- **Data reduction**

- Dimensionality reduction
- Numerosity reduction
- Data compression

- **Data transformation and data discretization**

- Normalization
- Concept hierarchy generation

Data Preprocessing

- Data Preprocessing: An Overview
 - Why ? Ans: Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning 
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
 - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - e.g., *Occupation*=" " (missing data)
 - noisy: containing noise, errors, or outliers
 - e.g., *Salary*="−10" (an error)
 - inconsistent: containing discrepancies in codes or names, e.g.,
 - *Age*="42", *Birthday*="03/07/2010"
 - Was rating "1, 2, 3", now rating "A, B, C"
 - discrepancy between duplicate records
 - Intentional (e.g., *disguised missing* data)
 - Jan. 1 as everyone's birthday?

Incomplete (Missing) Data

- Data is not always available
 - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
 - a global constant : e.g., “unknown”, a new class?!
 - the attribute mean
 - the attribute mean for all samples belonging to the same class: smarter

Noisy Data

- **Noise**: random error or variance in a measured variable
- **Incorrect attribute values** may be due to
 - faulty data collection instruments
 - data entry problems
 - data transmission problems
 - technology limitation
 - inconsistency in naming convention
- **Other data problems** which require data cleaning
 - duplicate records
 - incomplete data
 - inconsistent data

How to Handle Noisy Data?

- Binning

- first sort data and partition into (equal-frequency) bins
- then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.

- Regression

- smooth by fitting the data into regression functions

- Clustering

- detect and remove outliers

- Combined computer and human inspection

- detect suspicious values and check by human (e.g., deal with possible outliers)

Data Cleaning as a Process


■ Data discrepancy detection

- Use metadata (e.g., domain, range, dependency, distribution)
- Check field overloading
- Check uniqueness rule, consecutive rule and null rule
 - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)

■ Data migration and integration

- Data migration tools: allow transformations to be specified
- ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface

Data Preprocessing

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration 
- Data Reduction
- Data Transformation and Data Discretization
- Summary


Data Integration

- **Data integration:**
 - Combines data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust-id \equiv B.cust-\#$
 - Integrate metadata from different sources
- **Entity identification problem:**
 - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
 - For the same real world entity, attribute values from different sources are different
 - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
 - *Object identification*: The same attribute or object may have different names in different databases
 - *Derivable data*: One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Preprocessing

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction 
- Data Transformation and Data Discretization
- Summary

Data Reduction Strategies

- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
 - **Dimensionality reduction**, e.g., remove unimportant attributes
 - Feature subset selection, feature creation
 - **Numerosity reduction** (some simply call it: Data Reduction)
 - Regression and Log-Linear Models
 - Histograms, clustering, sampling
 - Data cube aggregation
 - **Data compression**

Data Reduction 1: Dimensionality Reduction

- **Curse of dimensionality**

- When dimensionality increases, data becomes increasingly sparse
- Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
- The possible combinations of subspaces will grow exponentially

- **Dimensionality reduction**

- Avoid the curse of dimensionality
- Help eliminate irrelevant features and reduce noise
- Reduce time and space required in data mining
- Allow easier visualization

- **Dimensionality reduction techniques**

- Wavelet transforms
- Principal Component Analysis
- Supervised and nonlinear techniques (e.g., feature selection)

Attribute Creation (Feature Generation)

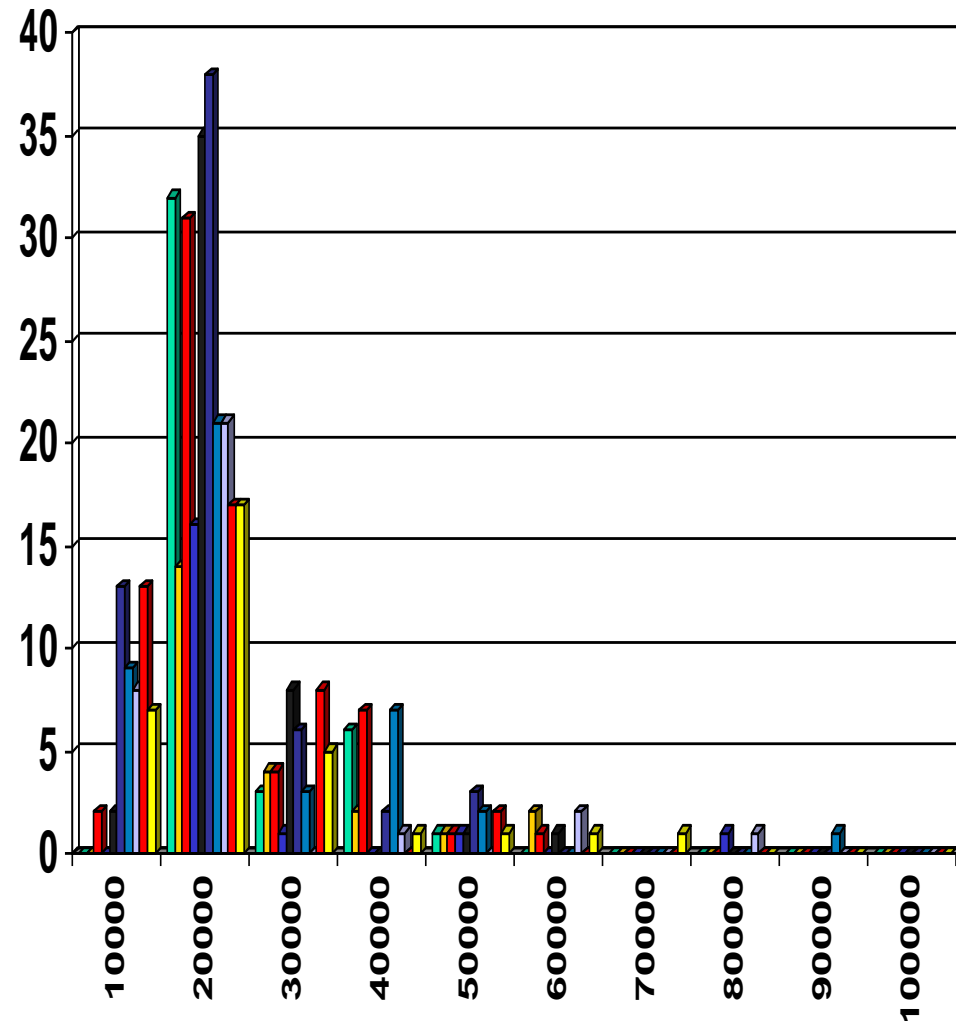
- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
 - Attribute extraction
 - Domain-specific
 - Mapping data to new space (see: data reduction)
 - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
 - Attribute construction
 - Combining features
 - Data discretization

Data Reduction 2: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
- **Non-parametric** methods
 - Do not assume models
 - Major families: histograms, clustering, sampling, ...

Histogram Analysis

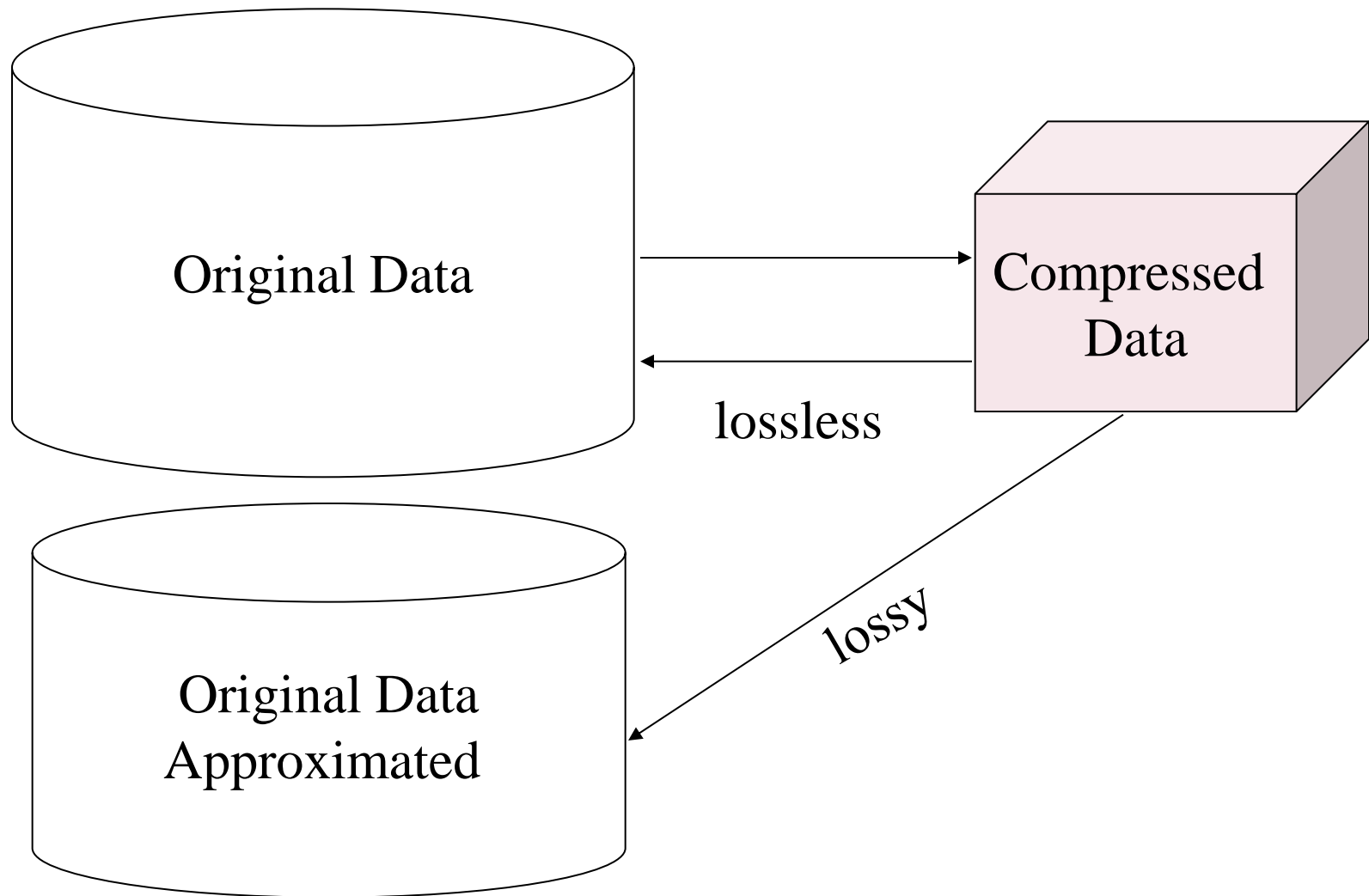
- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth)



Data Reduction 3: Data Compression

- String compression
 - There are extensive theories and well-tuned algorithms
 - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
 - Typically lossy compression, with progressive refinement
 - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
 - Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression

Data Compression



Data Preprocessing

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary



Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods
 - Smoothing: Remove noise from data
 - Attribute/feature construction
 - New attributes constructed from the given ones
 - Aggregation: Summarization, data cube construction
 - Normalization: Scaled to fall within a smaller, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
 - Discretization: Concept hierarchy climbing

Normalization

- **Min-max normalization:**

$$v' = \frac{v - \min_A}{\max_A - \min_A}$$

- **Z-score normalization** (μ : mean, σ : standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

| S.NO. | Normalization | Standardization |
|-------|---|---|
| 1. | Minimum and maximum value of features are used for scaling | Mean and standard deviation is used for scaling. |
| 2. | It is used when features are of different scales. | It is used when we want to ensure zero mean and unit standard deviation. |
| 3. | Scales values between [0, 1] or [-1, 1]. | It is not bounded to a certain range. |
| 4. | It is really affected by outliers. | It is much less affected by outliers. |
| 5. | Scikit-Learn provides a transformer called <code>MinMaxScaler</code> for Normalization. | Scikit-Learn provides a transformer called <code>StandardScaler</code> for standardization. |
| 6. | This transformation squishes the n-dimensional data into an n-dimensional unit hypercube. | It translates the data to the mean vector of original data to the origin and squishes or expands. |
| 7. | It is useful when we don't know about the distribution | It is useful when the feature distribution is Normal or Gaussian. |
| 8. | It is a often called as Scaling Normalization | It is a often called as Z-Score Normalization. |

Discretization

- Discretization: Divide the range of a continuous attribute into intervals
 - Interval labels can then be used to replace actual data values
 - Reduce data size by discretization
- Typical methods: All the methods can be applied recursively
 - Binning
 - Histogram analysis
 - Clustering analysis
 - Decision-tree analysis

Simple Discretization: Binning

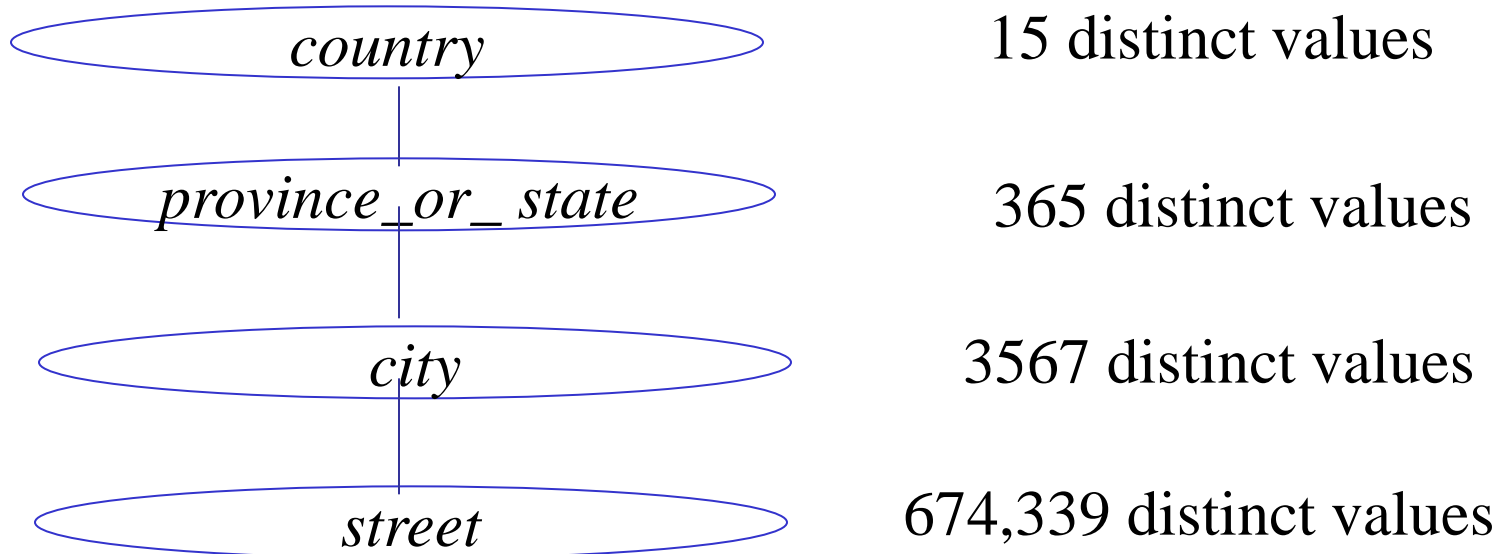
- **Equal-width** (distance) partitioning
 - Divides the range into N intervals of equal size: uniform grid
 - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A) / N$.
 - The most straightforward, but outliers may dominate presentation
 - Skewed data is not handled well
- **Equal-depth** (frequency) partitioning
 - Divides the range into N intervals, each containing approximately same number of samples
 - Good data scaling
 - Managing categorical attributes can be tricky

Concept Hierarchy Generation for Nominal Data


- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
 - *street* < *city* < *state* < *country*
- Specification of a hierarchy for a set of values by explicit data grouping
 - {Urbana, Champaign, Chicago} < Illinois
- Specification of only a partial set of attributes
 - E.g., only *street* < *city*, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
 - E.g., for a set of attributes: {*street*, *city*, *state*, *country*}

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
 - The attribute with the most distinct values is placed at the lowest level of the hierarchy
 - Exceptions, e.g., weekday, month, quarter, year



Data Preprocessing

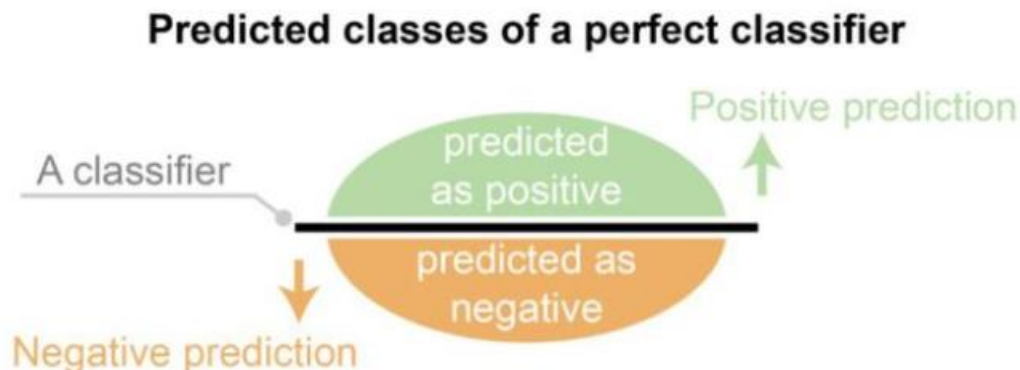
- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary 

Summary

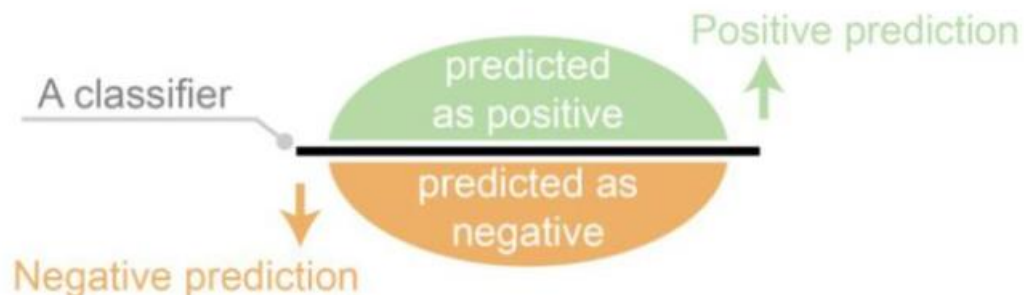
- **Data quality:** accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning:** e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
 - Entity identification problem
 - Remove redundancies
 - Detect inconsistencies
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data transformation and data discretization**
 - Normalization
 - Concept hierarchy generation

Binary Classifier

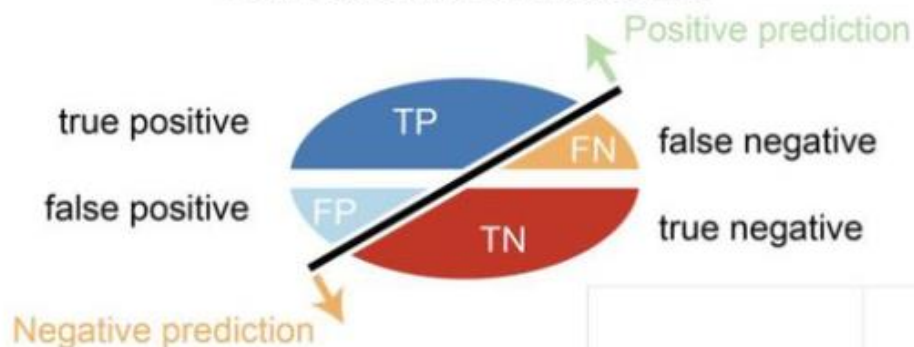
- A binary classifier produces output with two class values or labels, such as Yes/No, 1/0, Positive/Negative for given input data
- A dataset used for performance evaluation is called a test dataset. Observed labels are used to compare with the predicted labels for performance evaluation after classification
- The predicted labels will be exactly the same if the performance of a classifier is perfect, But it is uncommon to be able to develop a perfect classifier



Predicted classes of a perfect classifier



Four outcomes of a classifier



| | | Predicted | |
|----------|----------|---------------|---------------|
| | | Positive | Negative |
| Observed | Positive | TP (# of TPs) | FN (# of FNs) |
| | Negative | FP (# of FPs) | TN (# of TNs) |

Confusion Matrix

- A confusion matrix is formed from the four outcomes produced as a result of binary classification
- True positive (TP): correct positive prediction
- False positive (FP): incorrect positive prediction
- True negative (TN): correct negative prediction
- False negative (FN): incorrect negative prediction

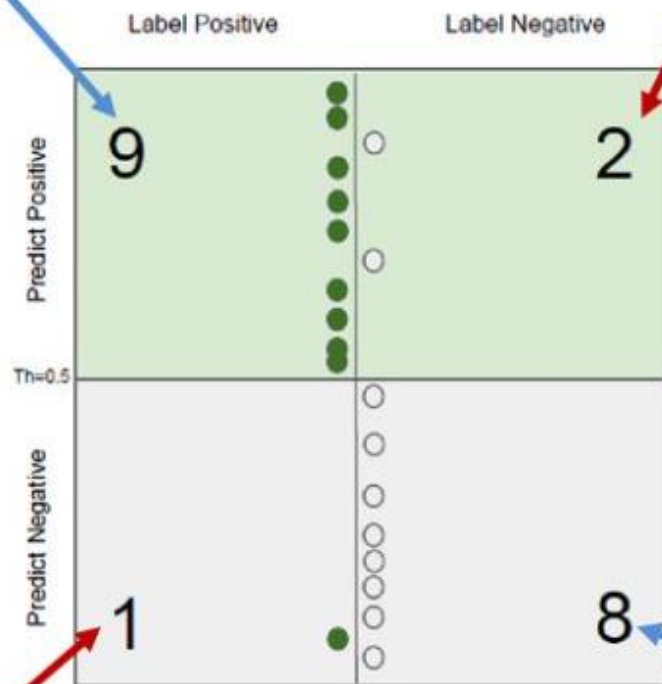
True Positives

Green examples **correctly** identified as green

False Positives

Gray examples **falsely** identified as green

Confusion Matrix



False Negatives

Green examples **falsely** identified as gray

True Negatives

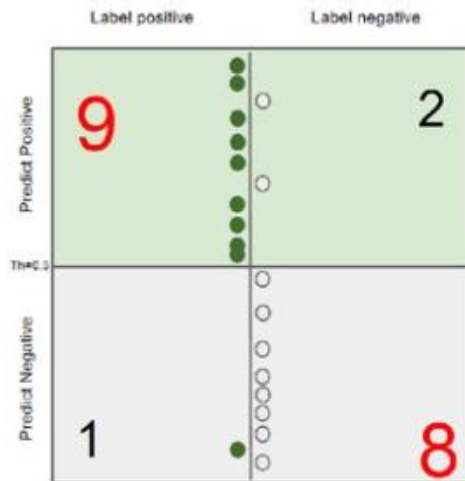
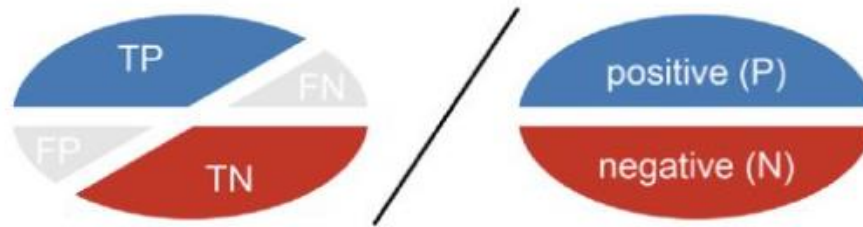
Gray examples **correctly** identified as grey

Accuracy

Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset

The best ACC is 1.0, whereas the worst is 0.0

$$\text{Accuracy: } (TP + TN) / (P + N)$$



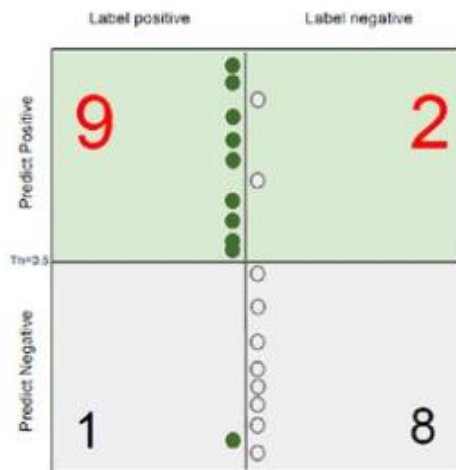
- Accuracy = $(9+8) / (9+2+1+8) = 0.85$

Precision

Precision is calculated as the number of correct positive predictions divided by the total number of positive predictions

The best precision is 1.0, whereas the worst is 0.0

$$\text{Precision: } TP / (TP + FP)$$



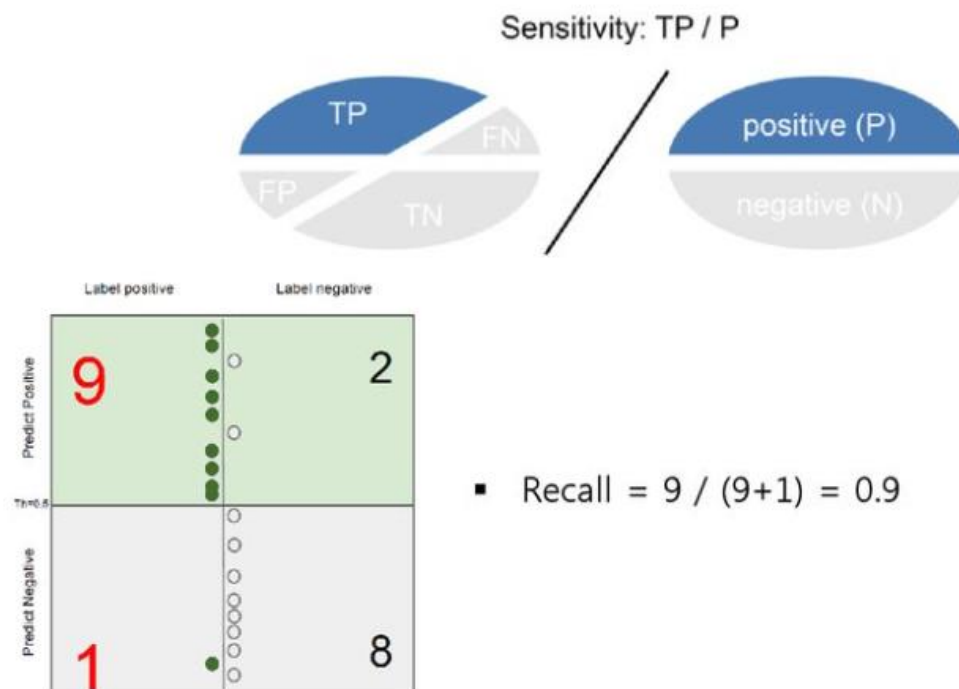
- $\text{Precision} = 9 / (9+2) = 0.81$

Recall (Probability of detection)

Sensitivity = Recall = True Positive Rate

Recall is calculated as the number of correct positive predictions divided by the total number of positives

The best recall is 1.0, whereas the worst is 0.0



TPR – True positive Rate

Example: The example to classify whether images contain either a dog or a cat

- The training data contains 25000 images of dogs and cats;
- The training data 75% of 25000 images; ($25000 \times 0.75 = 18750$)
- Validation data 25% of training data; ($25000 \times 0.25 = 6250$)

Confusion matrix

| | | True class | |
|-----------------|-----|------------|-----|
| | | cat | dog |
| Predicted class | cat | 2 | 0 |
| | dog | 3 | 5 |

Test Data, 5 cats, 5 dogs



- **Precision** = $2 / (2 + 0) * 100\% = 100\%$
- **Recall** = $2 / (2 + 3) * 100\% = 40\%$
- **Accuracy** = $(2 + 5) / (2 + 0 + 3 + 5) * 100\% = 70\%$