Data Preprocessing and Feature Engineering Techniques in Healthcare

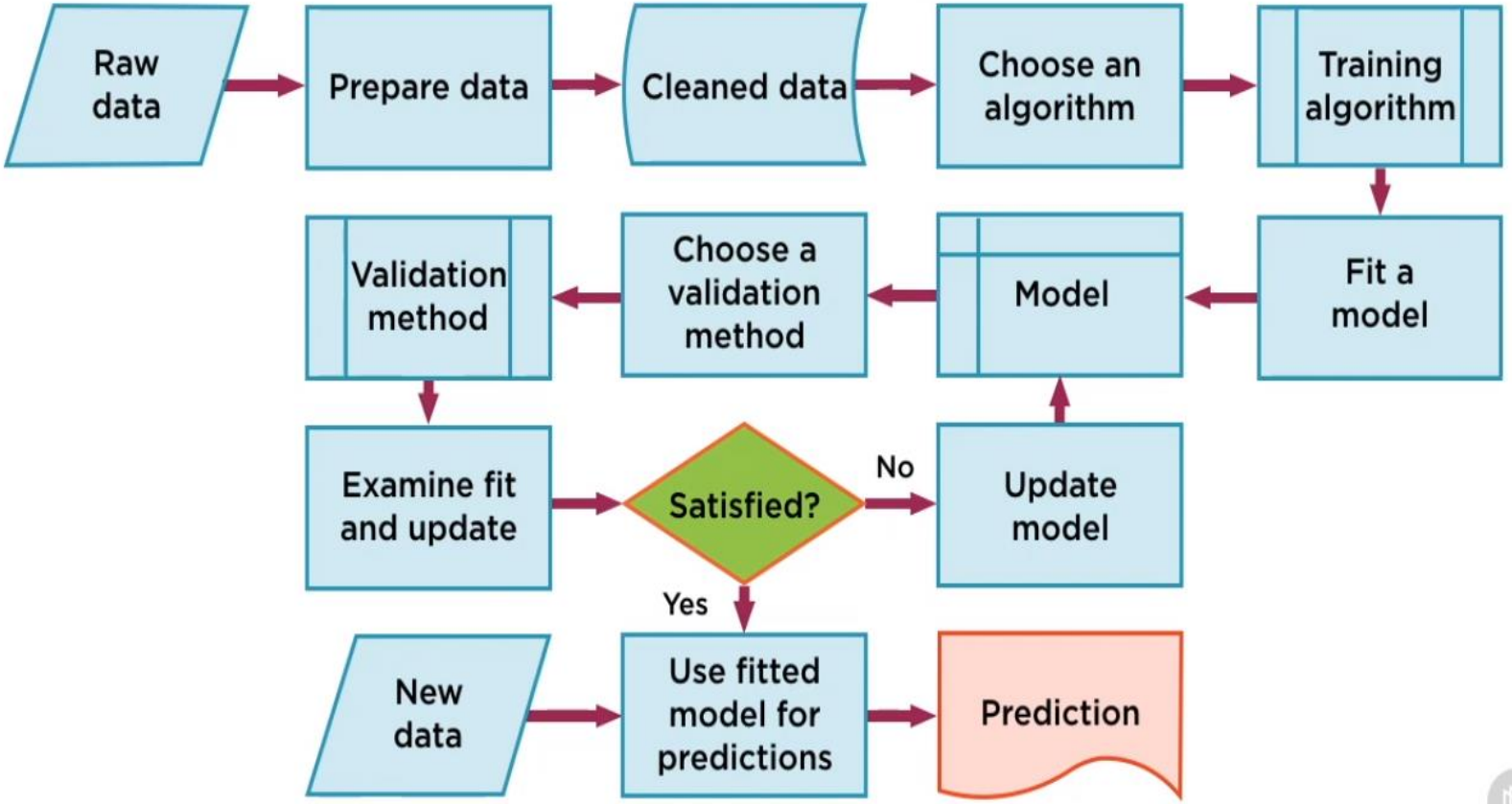
By

Mrs. Hemavati
Research Assistant

Under the Guidance of
Prof. V Susheela Devi

Department of CSA, IISc Bangalore

Basic Machine Learning Workflow



Machine Learning

Work with a huge maze of data

Find patterns

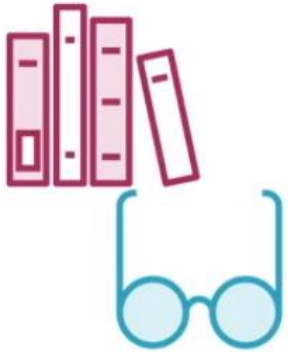
Make intelligent decisions

Types of Machine Learning Problems

Classification

Regression

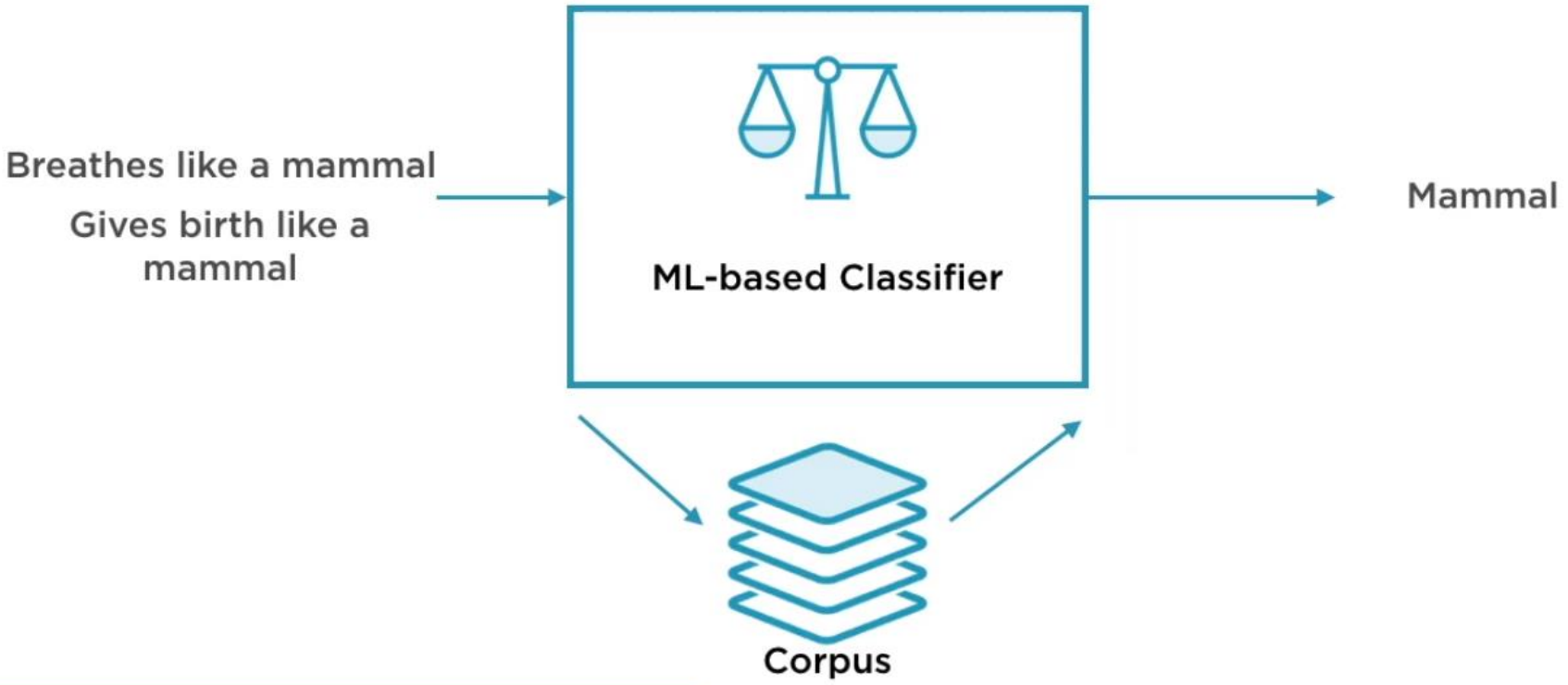
Clustering



Dimensionality Reduction

Scope of Feature Engineering

ML-based Binary Classifier



x Variables

The attributes that the ML algorithm focuses on are called features

Each data point is a list - or vector - of such features

Thus, the input into an ML algorithm is a feature vector

Feature vectors are usually called the x variables

y Variables

The attributes that the ML algorithm tries to predict are called labels

Labels are usually called the y variables

Types of labels

- categorical (classification)
- continuous (regression)

Garbage In, Garbage Out

If data fed into an ML model is of poor quality, the model will be of poor quality

Choosing Feature Selection

Use Case	Possible Solution
Many X-variables	
Most of which contain little information	Feature selection
Some of which are very meaningful	
Meaningful variables are independent of each other	

Feature Selection Techniques

Filter methods

Embedded methods

Wrapper methods

Filter Methods

Applying statistical techniques to select the most relevant features

Embedded Methods

Relevant features selected by training a machine learning model i.e. Lasso regression, decision trees

Wrapper Methods

Build candidate models by selecting feature subsets - choose the subset which gives the best model

Feature Learning

Rely on ML algorithms rather than human experts to "learn" the best representations of complex data such as images, videos.

(Also known as Representation Learning)

Supervised Feature Learning

Features are learnt using labeled data

Neural networks are classic example

Greatly reduce need for expert judgment

"Traditional" ML-based systems rely on experts to decide what features to pay attention to

"Representation" ML-based systems figure out by themselves what features to pay attention to

Unsupervised Feature Learning

Features need to be learned in absence of labeled corpus

- Clustering
- Dictionary learning
- Autoencoders

Feature Extraction

Image descriptors for images

Principal components for matrices

Tf-Idf for documents

Feature Extraction

Feature extraction usually also leads to dimensionality reduction

However explicit objective is to re-express feature in a "better" form

Not to reduce number of X columns

Feature Combination

Some features naturally work better when considered together

Original feature might be raw or too granular

Improve the predictive power of features

Feature Combination

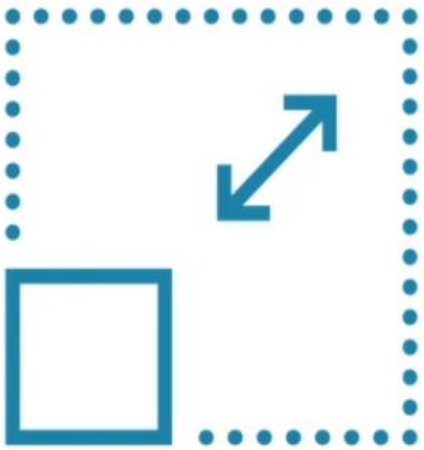
Feature cross in predicting traffic

- Day-of-week
- Time-of-day

Feature cross in predicting temperature

- Season
- Time-of-day

Dimensionality Reduction



Apply pre-processing algorithms to reduce complexity of raw features

Specifically aim to reduce number of input features

Excessive number of features leads to severe problems

- Curse of Dimensionality

Why preprocess the data?

- Data in the real world is *Dirty*…
  - Incomplete Data: Lacking attribute values, Lacking certain attributes of interest, or containing only aggregate data

    e.g. Occupation=“ ”, year_salary = “13.000”, …
  - Inconsistent Data: Containing discrepancies in codes or names

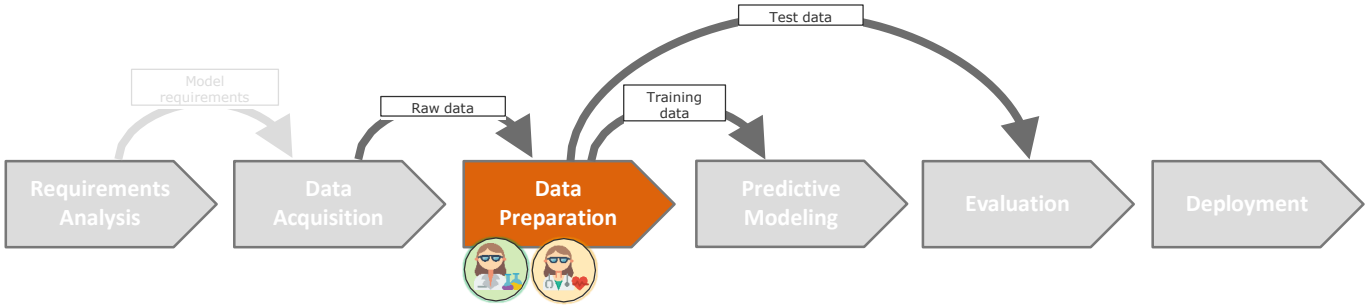
    e.g. Age=“42” Birthday=“03/07/1997”

    Previous rating “1,2,3”, Present rating “A, B, C”

    Discrepancy between duplicate records
  - Noisy Data: Containing errors or outliers

    e.g. Salary=“-10”, Family=“Unknown”, …

Data Preparation



Data Preparation

- Exploration
- Quality assessment
- Cleansing
- Labeling
- Imputation
- Feature engineering

Roles — Data Scientist — Domain Expert — (Data) Engineer

Icons made by Smashicons from www.flaticon.com

What is Data Preprocessing?

- Data
  - Text
  - Image
  - Video
  - Audio

- Data Preprocessing is a process to convert raw data into meaningful data using different techniques.

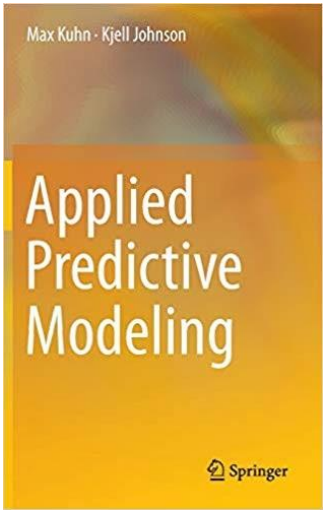
What Is Data Preparation

Data preparation can make or break the predictive ability of your model

According to Kuhn and Johnson data preparation is the process of addition, deletion or transformation of training set data

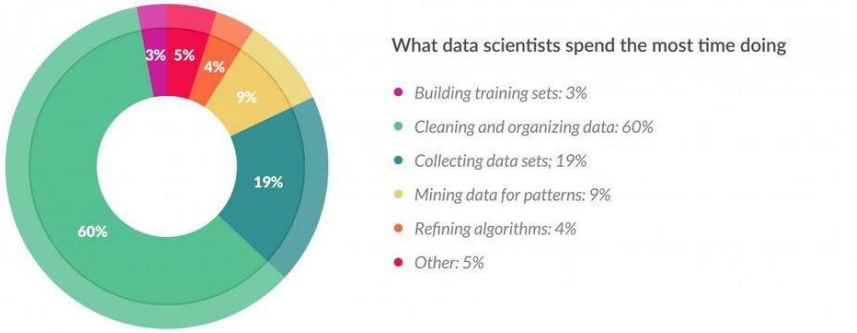
Sometimes, preprocessing of data can lead to unexpected improvements in model accuracy

Data preparation is an important step and you should experiment with data pre-processing steps that are appropriate for your data to see if you can get that desirable boost in model accuracy

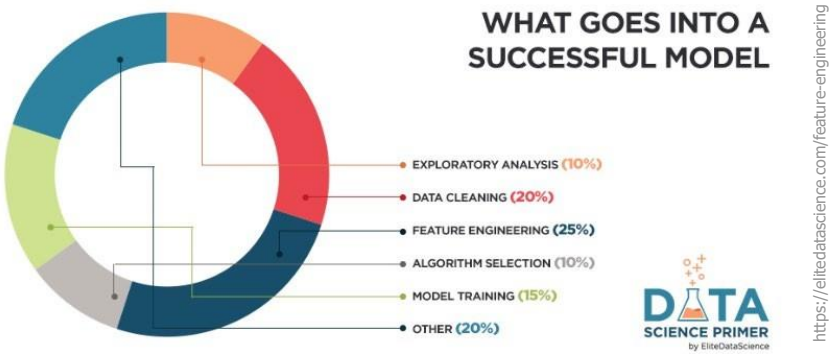


Data Preparation Importance Motivation

- Data in Healthcare → sparse and incomplete
- Preparing the proper input dataset, compatible with the machine learning algorithm requirements
- Integral step in Machine Learning
- Directly affects the ability of our model to learn
- Make sure that it is in a useful scale, format and even that meaningful features are included
- Improving the performance of machine learning models



Source: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/



Why Data Preparation Is so Important in Digital Health

algorithms exist but selecting the best classifier surely improves the accuracy of the predictions. The preprocessing methods selected in this study are Multiple Imputation, k-means for missing values treatment, Discretization to change in discrete values, Standard scaler, Min-Max scalar for feature scaling and, Random Forest (RF) for feature selection. For the classification Logistic Regression (LR), Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest (RF) are used. To evaluate the performance of model accuracy, sensitivity, specificity are used. This study compares the model performance with and without preprocessed data and has proved that the selected preprocessing methods significantly improves the model performance.

have used two different datasets of healthcare sector for predicting the type 2 diabetes onset. Firstly, we started

Table 5: Outcome of classifiers before and after applying pre-processing techniques on PIDDs datasets

Preprocessing Techniques	Classifier	Accuracy	Sensitivity	Specificity
No Preprocessing	LR	0.75	0.57	0.86
With Preprocessing	**LR**	**0.77**	**0.56**	**0.88**
No Preprocessing	ANN	0.67	0.25	0.89
With Preprocessing	**ANN**	**0.80**	**0.65**	**0.88**
No Preprocessing	SVM	0.65	0	1
With Preprocessing	**SVM**	**0.79**	**0.82**	**0.78**
No Preprocessing	Random Forest	0.74	0.48	0.89
With Preprocessing	**Random Forest**	**0.78**	**0.67**	**0.84**

Table 6: Outcome of classifiers before and after applying pre-processing techniques on LUDB2 datasets

Preprocessing Techniques	Classifier	Accuracy	Sensitivity	Specificity
No Preprocessing	LR	0.98	0.96	1.0
With Preprocessing	**LR**	**0.98**	**1.0**	**0.96**
No Preprocessing	ANN	0.94	0.88	1.0
With Preprocessing	**ANN**	**0.96**	**0.92**	**1.0**
No Preprocessing	SVM	0.74	1.0	0.48
With Preprocessing	**SVM**	**1.0**	**1.0**	**1.0**
No Preprocessing	Random Forest	1.0	1.0	1.0
With Preprocessing	**Random Forest**	**1.0**	**1.0**	**1.0**

Medical Use Cases

- Predicting and treating disease
- Providing medical imaging and diagnostics
- Discovering and developing new drugs
- Organizing medical records
- Clinical outcomes and results
- Claims and billing
- Population health

Data Preparation Steps

How do I clean up the data? → Data Cleaning

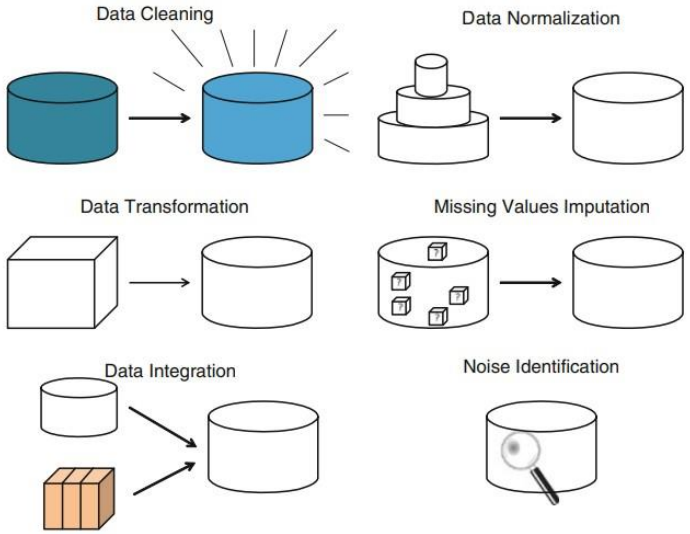
How do I provide accurate data? → Data Transformation

How do I incorporate and adjust data? → Data Integration

How do I unify and scale data? → Data Normalization

How do I handle missing data? → Missing Data Imputation

How do I detect and manage noise? → Noise Identification

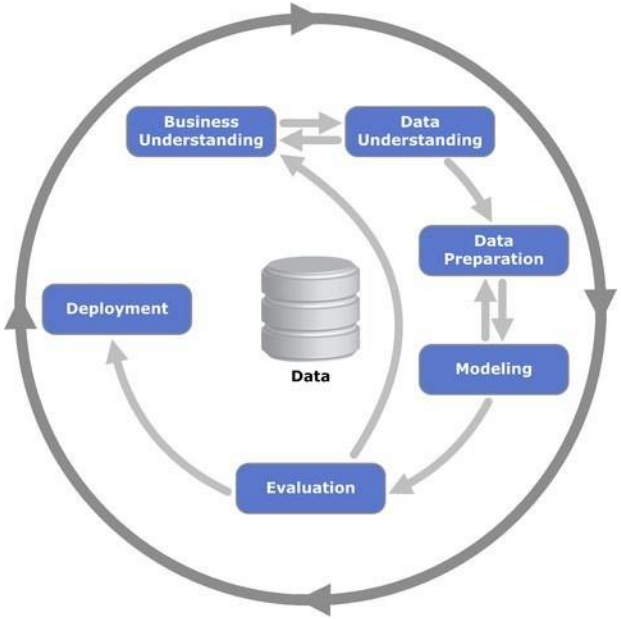


Data Preparation Process

Process for getting data ready for a machine learning algorithm can be summarized

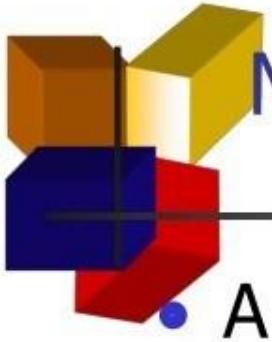
- Step 1: Select Data
- Step 2: Preprocess Data
- Step 3: Transform Data

Follow this process in a linear manner



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 transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results



Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data

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 (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Fill in the missing value manually: tedious + infeasible?
- Use a global constant to fill in the missing value: e.g., "unknown", a new class?!
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all samples belonging to the same class to fill in the missing value: smarter
- Use the most probable value to fill in the missing value: inference-based such as Bayesian formula or decision tree

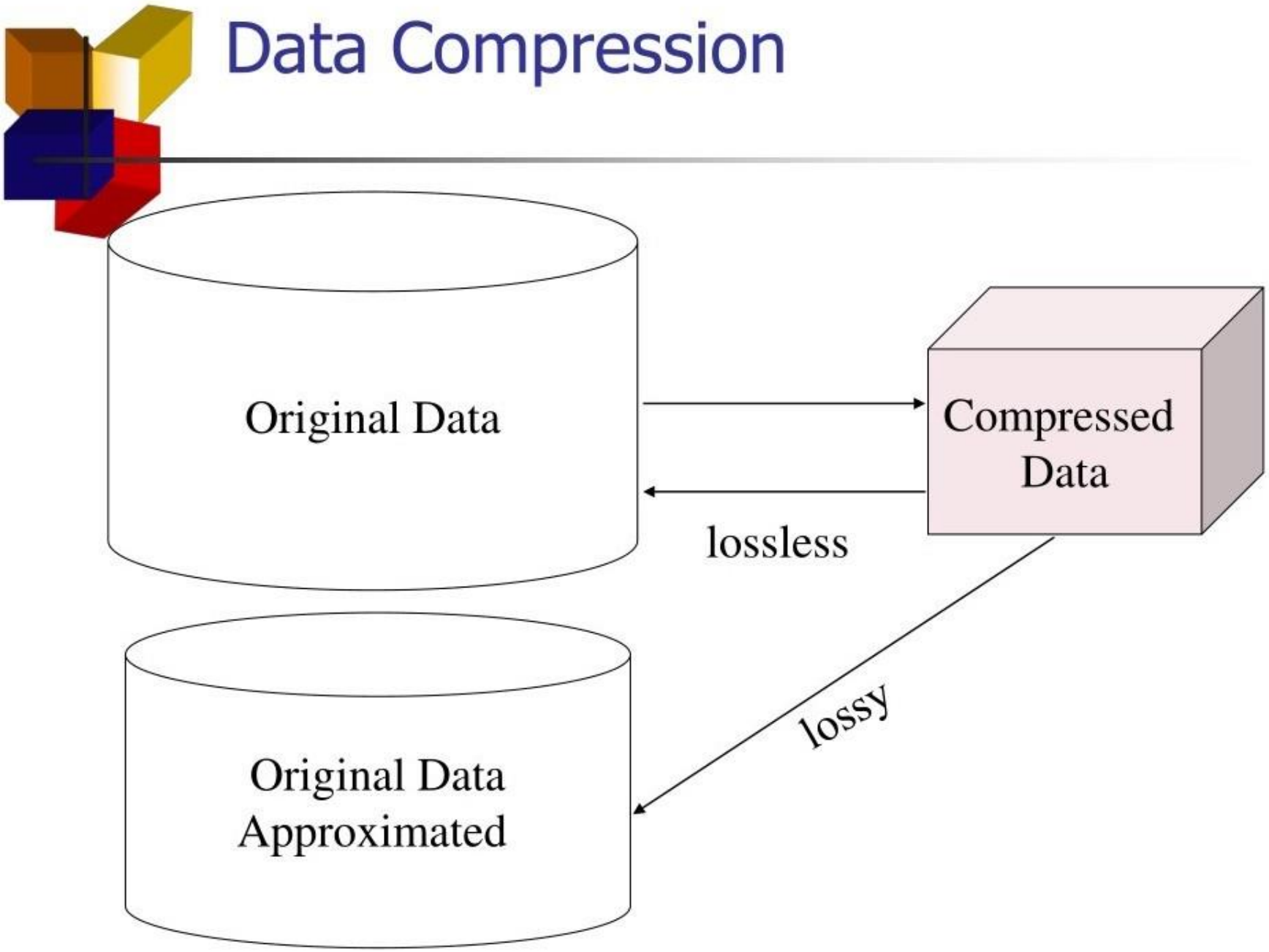
Data Integration

- Data integration:
  - combines data from multiple sources.
  - Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources, e.g., A.cust-id ≈ B.cust-#
- 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 Redundant Data in Data Integration

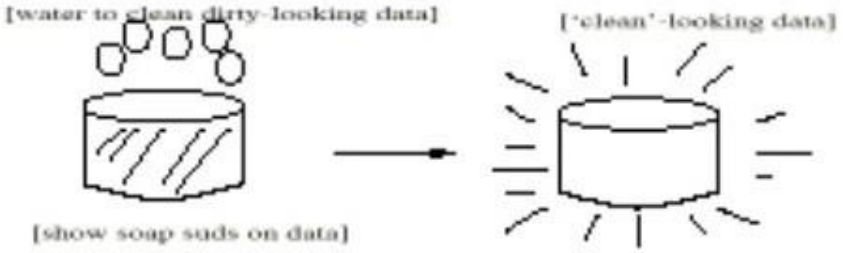
- Redundant data occur often when integration of multiple databases
  - The same attribute may have different names in different databases
  - One attribute may be a "derived" attribute in another table.
  - Redundant data may be able to be detected by correlational analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Compression

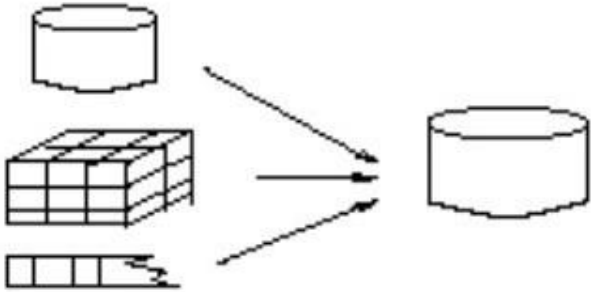


Forms of data preprocessing

Data Cleaning



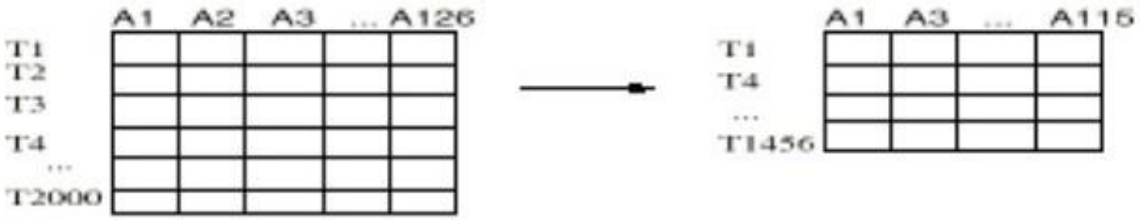
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Data imputation techniques

- **Data imputation techniques** are used to fill in missing data values in a dataset.
- Some of the most common techniques include :Mean or median

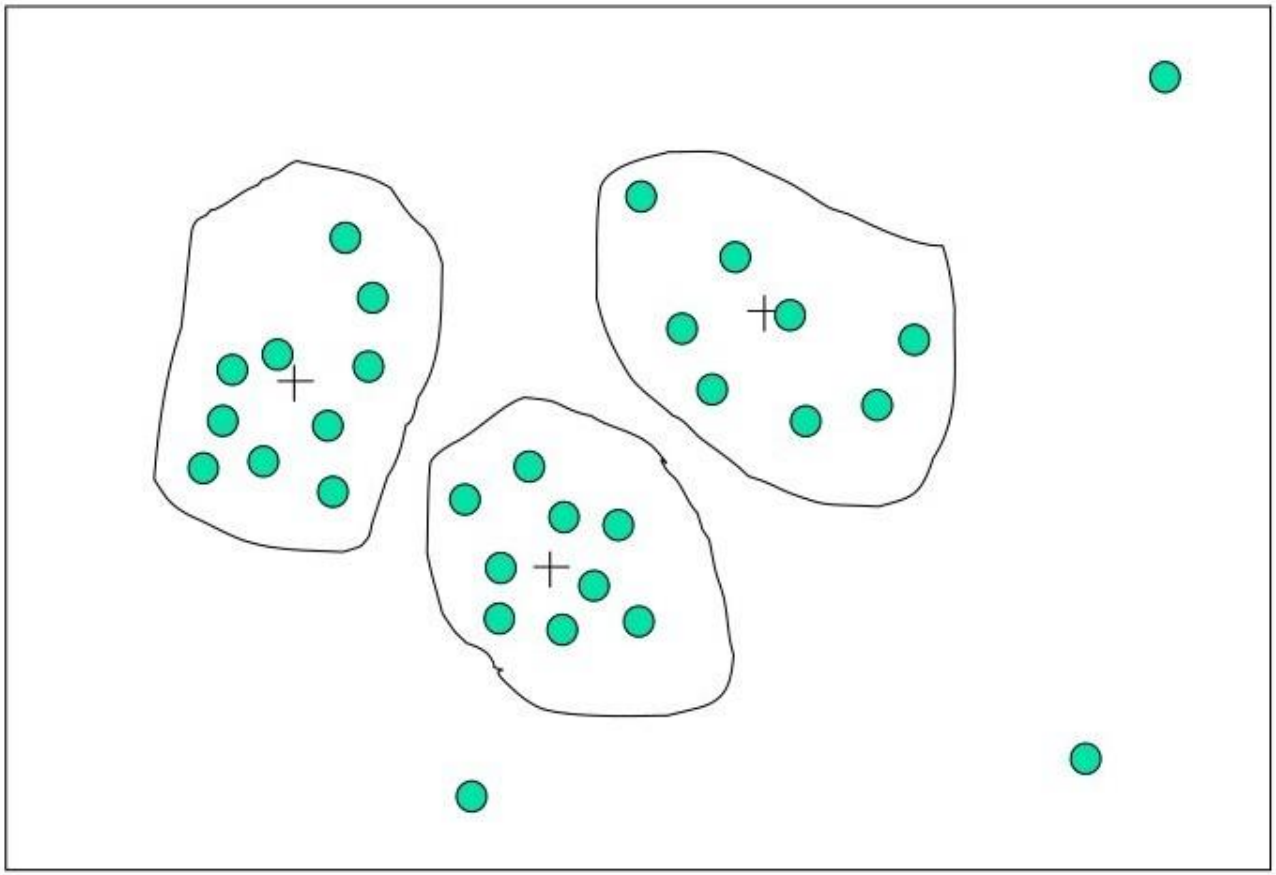
- imputation: replacing missing values with the mean or median value for that variable.
- Hot-deck imputation: replacing each missing value with an existing value from a similar case or participant within your dataset.
- Next or previous value imputation: for time-series data or ordered data, there are specific imputation techniques.
- K-nearest neighbors imputation: finding the k nearest examples in the data where the value in the relevant variable is not missing.
- Maximum or minimum value imputation: replacing missing values with the maximum or minimum value for that variable.

Noisy Data

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

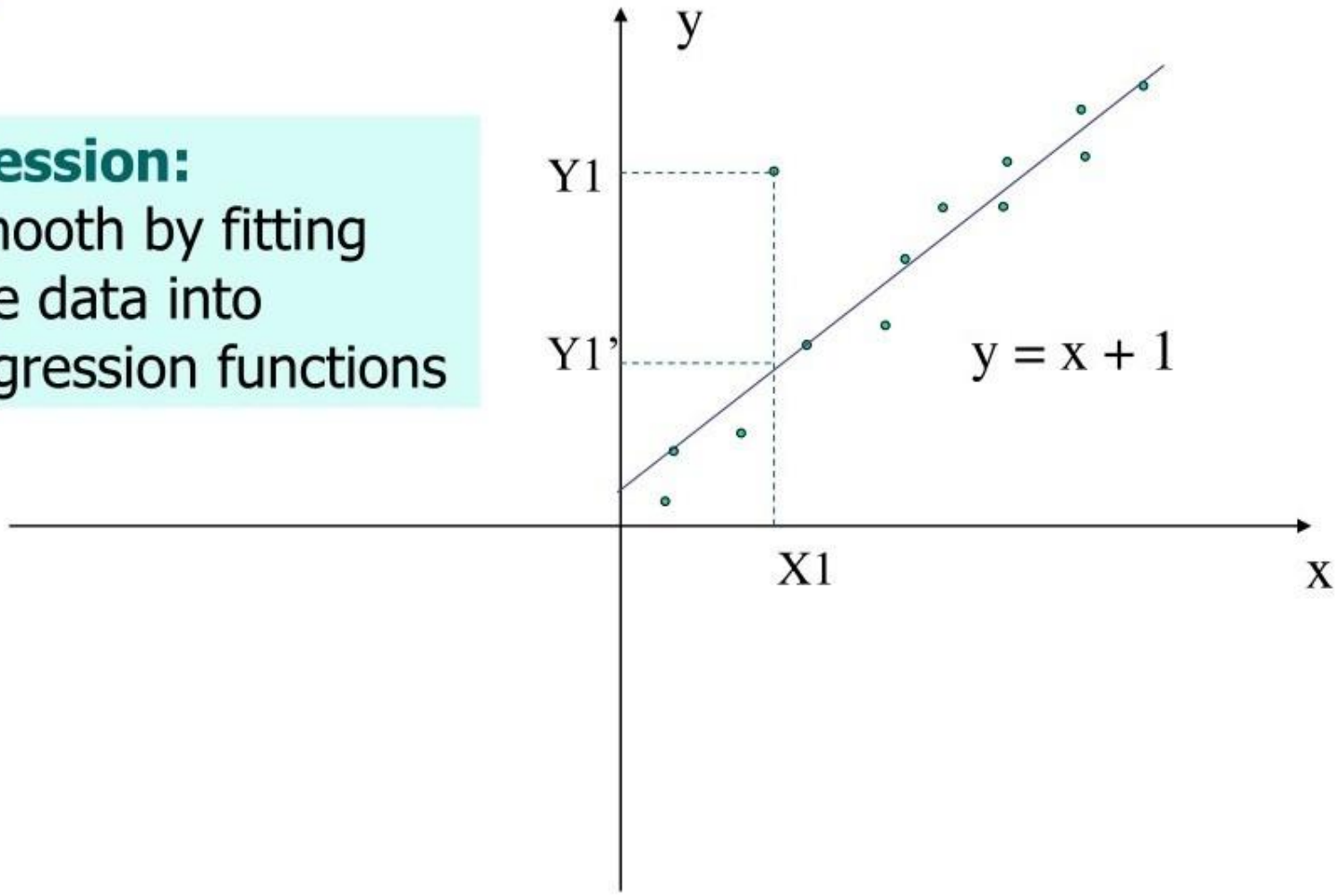
Cluster Analysis

Clustering: detect and remove outliers



Regression

Regression:
smooth by fitting the data into regression functions



How to Handle Noisy Data?

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median
  - Equal-width (distance) partitioning:
    - It 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: $W = (B-A)/N$.
- Equal-depth (frequency) partitioning:
  - It divides the range into *N* intervals, each containing approximately same number of samples
  - Managing categorical attributes can be tricky.
- Combined computer and human inspection
  - detect suspicious values and check by human

Data Transformation: Normalization

- **Normalization by Decimal Scaling**
  - Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A.
  - The number of decimal points moved depends on the maximum absolute value of A.
  - a value v of A is normalized to v′ by computing: $v' = (v / 10j)$. Where j is the smallest integer such that $Max(|v'|)<1$.

Data Transformation: Normalization

- **Normalization by Decimal Scaling**
  - Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A.
  - The number of decimal points moved depends on the maximum absolute value of A.
  - a value v of A is normalized to v′ by computing: $v' = (v / 10j)$. Where j is the smallest integer such that $Max(|v'|)<1$.

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones

Data Reduction Strategies

- Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set

- Data reduction
  - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results

- Data reduction strategies
  - Data cube aggregation
  - Dimensionality reduction
  - Data Compression

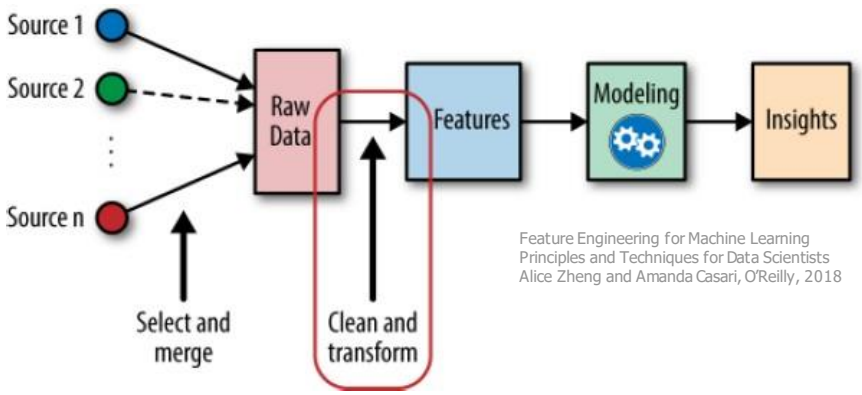
Feature Engineering

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering. ~Andrew Ng

The features you use influence more than everything else the result. No algorithm alone can supplement the information gain given by correct feature engineering. ~Luca Massaron

Good data preparation and feature engineering is integral to better prediction ~Marios Michailidis (KazAnova), Kaggle GrandMaster, Kaggle #3, former #1



Feature Engineering for Machine Learning
Principles and Techniques for Data Scientists
Alice Zheng and Amanda Casari, O'Reilly, 2018

Feature Engineering

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

Data may be hard to understand and process

Conduct feature engineering to make reading of the data easier for our machine learning models

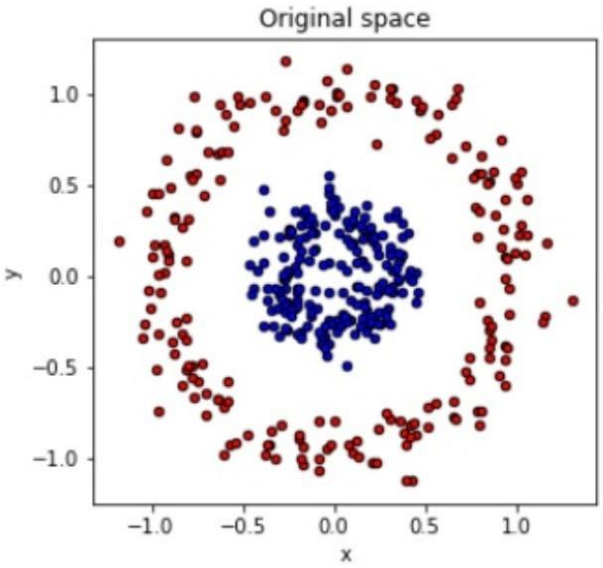
Feature Engineering is a process of transforming the given data into a form which is easier to interpret

In general: Features can be generated so that the data visualization prepared for people without a data-related background can be more digestible

Different models often require different approaches for the different kinds of data

Feature Engineering
Example: Coordinate Transformation

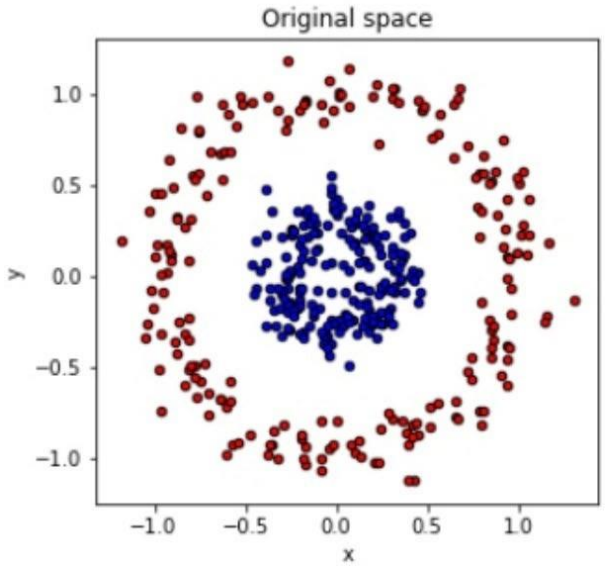
Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data



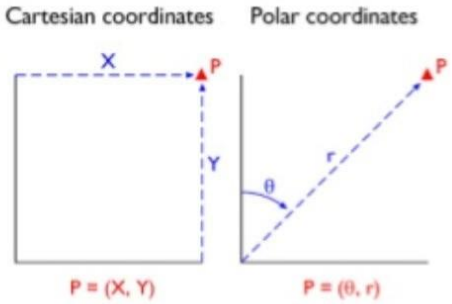
Not possible to seperate using linear classifier

Feature Engineering
Example: Coordinate Transformation

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

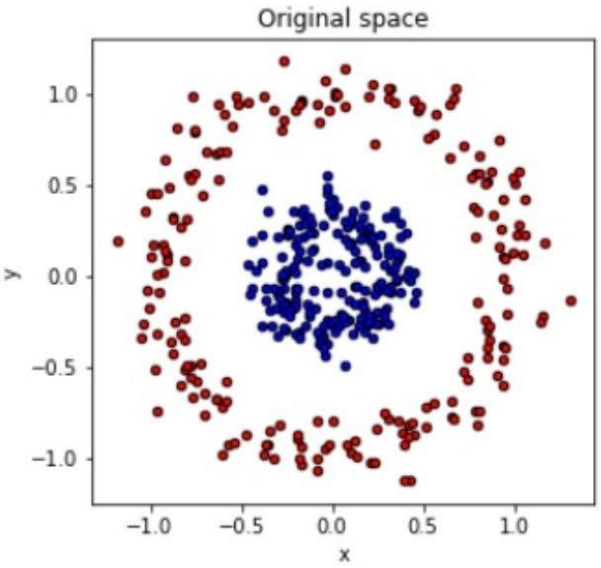


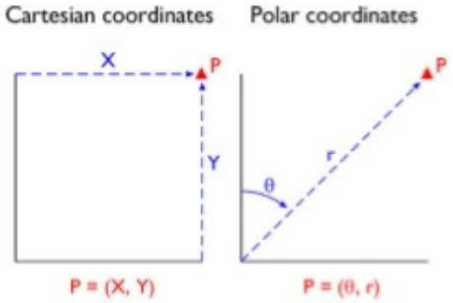
What if you use polar Coordinates instead?

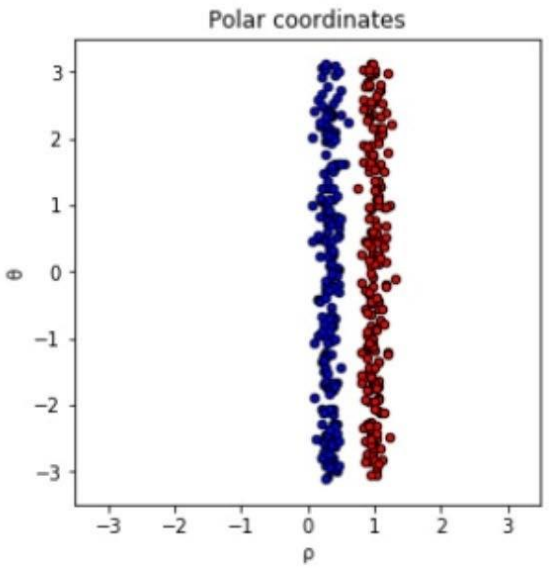


Feature Engineering
Example: Coordinate Transformation

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data







Iterative Process of Feature Engineering

Brainstorm features: Really get into the problem, look at a lot of data, study feature engineering on other problems and see what you can steal

Devise features: Depends on your problem, but you may use automatic feature extraction, manual feature construction and mixtures of the two

Select features: Use different feature importance scorings and feature selection methods to prepare one or more "views" for your models to operate upon

Evaluate models: Estimate model accuracy on unseen data using the chosen features

Aspects of Feature Engineering

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

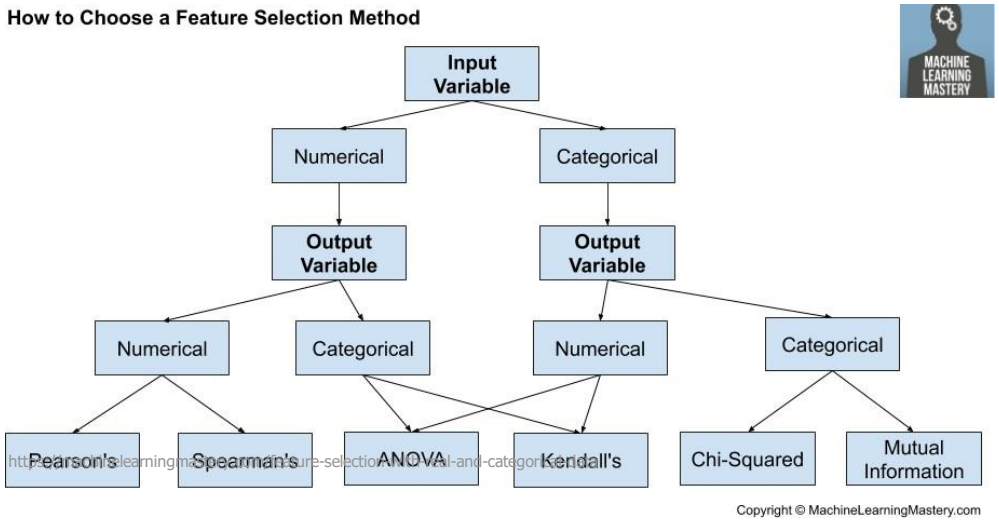
Feature Engineering	
Feature Selection	Most useful and relevant features are selected from the available data
Feature Extraction	Existing features are combined to develop more useful ones
Feature Addition	New features are created by gathering new data
Feature Filtering	Filter out irrelevant features to make the modeling step easy

Feature Selection

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

Process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested

Having irrelevant features in your data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression

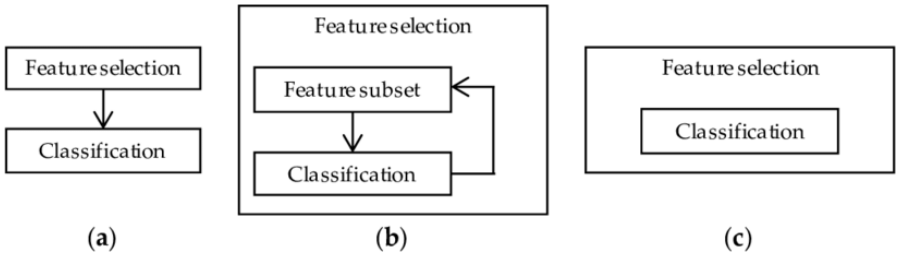


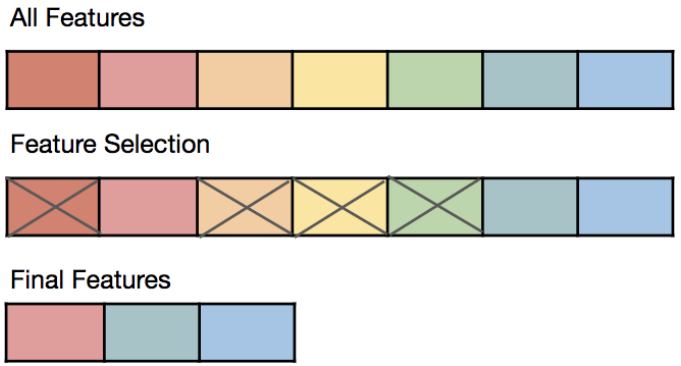
Feature Selection

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

Three benefits of performing feature selection before modeling your data are:

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise
- Improves Accuracy: Less misleading data means modeling accuracy improves
- Reduces Training Time: Less data means that algorithms train faster





https://quantdare.com/what-is-the-difference-between-feature-extraction-and-feature-selection/

https://towardsdatascience.com/feature-selection-techniques-1bfab5fe0784

Feature Extraction

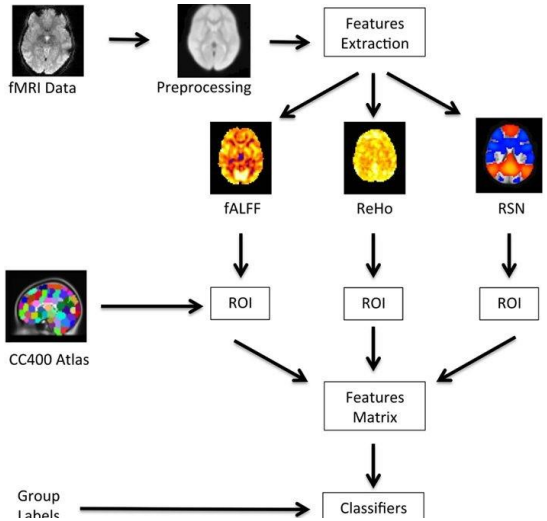
Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

Aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features)

New reduced set of features should then be able to summarize most of the information contained in the original set

Create some interaction (e.g., multiply or divide) between each pair of variables → lengthy process

Deep feature synthesis (DFS) is an algorithm which enables you to quickly create new variables with varying depth



https://matlab1.com/feature-extraction-image-processing/

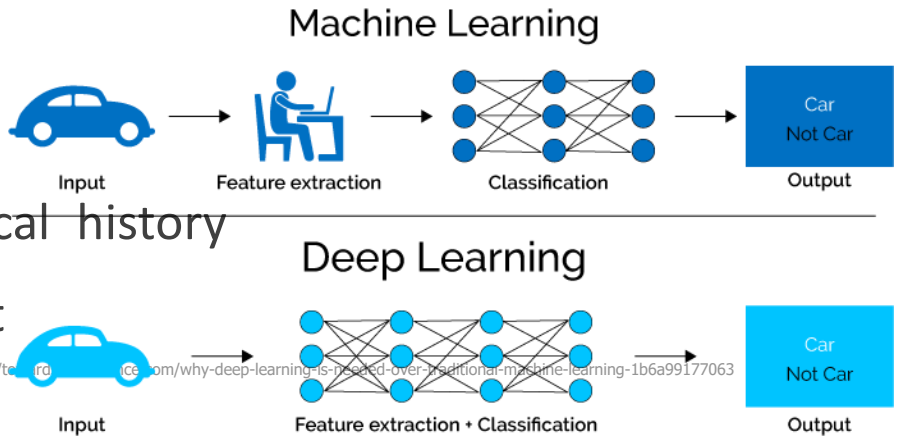
Preprocessing and

Automated Feature Engineering Why Do It?

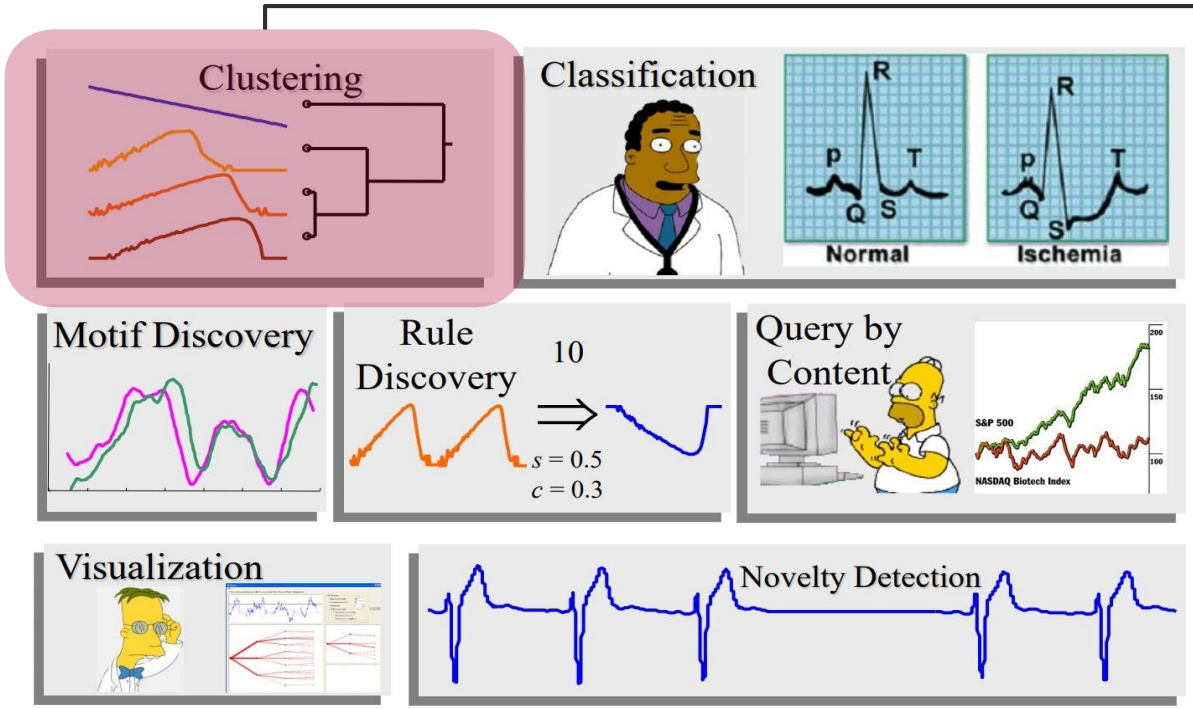
Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

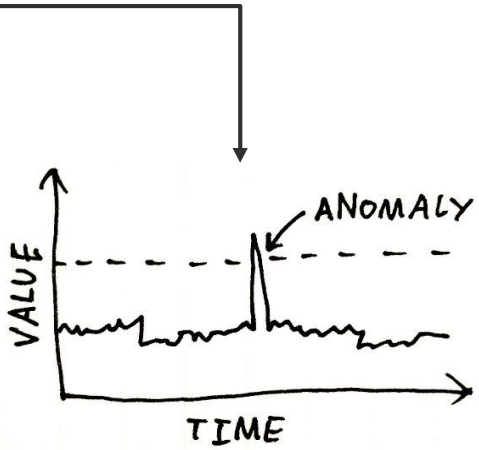
We're interested in *features*—we want to know which are relevant. If we fit a model, it should be interpretable

- What causes lung cancer?
  - Features are aspects of a patient's medical history
  - Binary response variable: did the patient develop lung cancer?
  - Which features best predict whether lung cancer will develop? Might want to legislate against these features.



What Next?





http://didawiki.cli.di.unipi.it/lib/exe/fetch.php/dm/time_series_2017.pdf
http://amid.fish/anomaly-detection-with-k-means-clustering

What to Take Home?

Data preparation allows simplification of data to make it ready for Machine Learning and involves data selection, preprocessing, and transformation

Step 1: Data Selection Consider what data is available, what data is missing and what data can be removed

Step 2: Data Preprocessing Organize your selected data by formatting, cleaning and sampling from it

Step 3: Data Transformation Transform preprocessed data ready for machine learning by engineering features using scaling, attribute decomposition and attribute aggregation