Machine Learning

Overview

Outline

- Introduction
  - Myself
  - TAs
  - Course

- Machine Learning
  - Industry Standard Process for Data Mining
  - Supervised and Unsupervised Learning
  - Data Terminology
  - Inputs and Outputs
  - Parametric and Nonparametric Methods
  - Generative vs Discriminative Models
  - Representation and Deep Learning
  - Machine Learning workflow

Myself

- Volkan Vural (vvural@ucsd.edu)
- Ph.D. in Machine Learning
- Research in Classification/ Computer Aided Diagnosis
- 6+ years of experience in Investment Technologies
- Boston College / MBA classes
  - Forecasting in Business and Economics
  - Machine Learning for Business Intelligence

TAs

- Mengjie Wang (mew006@ucsd.edu)
- Yanki Kalfa (skalfa@ucsd.edu)
- Sri Pamidi (spamidi@ucsd.edu)

Course

- Machine Learning
  - Important concepts & methods
  - Both theory and applications
- Requires introductory level statistics background
- Practical and application oriented
- Python will be used for implementations

Class Outline:

- CRoss Industry Standard Process for Data Mining
- Supervised and Unsupervised Learning
- Data Terminology
- Inputs and Outputs
- Parametric and Nonparametric Methods
- Generative vs Discriminative Models
- Representation and Deep Learning
- Machine Learning workflow

Machine Learning

Forbes / Tech

FEB 16, 2012 @ 11:02 AM 2,920,119 VIEWS

How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

Kashmir Hill, FORBES STAFF

Welcome to The Not-So Private Parts where technology & privacy collide

FOLLOW ON FORBES (2081)

Opinions expressed by Forbes Contributors are their own.

FULL BIO

TARGET

Target has got you in its aim

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target, for example, has figured out how to data-mine its way into your

How Financial Industry is applying ML?

Maybe you have heard of these:

- Risk Modeling: modeling techniques such as: logistic regression, discriminant analysis, classification trees, etc.
- Portfolio Management: real-time asset allocation

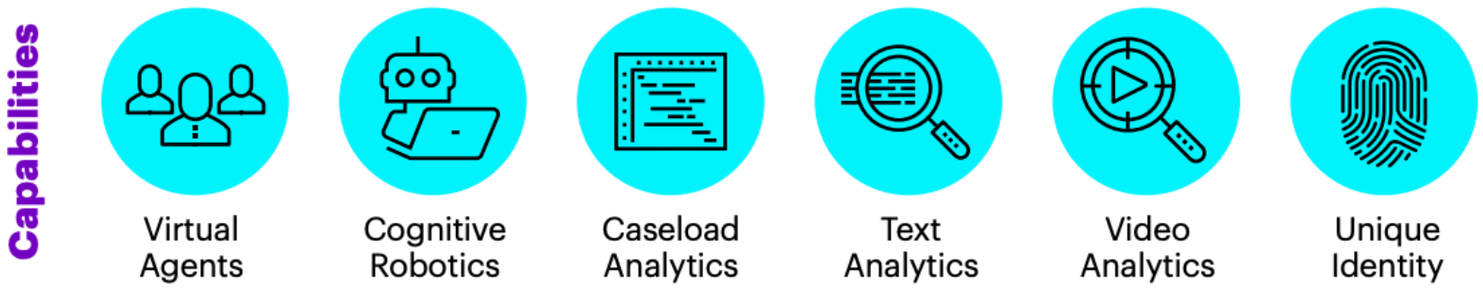
age, income, current financial assets --> investment decision

- Fraud and Misconduct Detection: How to reduce False Positive using ML tools?
- Loan/Insurance Underwriting: Will this client default?

How Financial Industry is applying ML?

Here is something interesting… Accenture (2017)

https://www.accenture.com/_acnmedia/accenture/conversion-assets/mainpages/documents/global/accenture-emerging-trends-in-the-validation-of-ml-and-ai-models.pdf

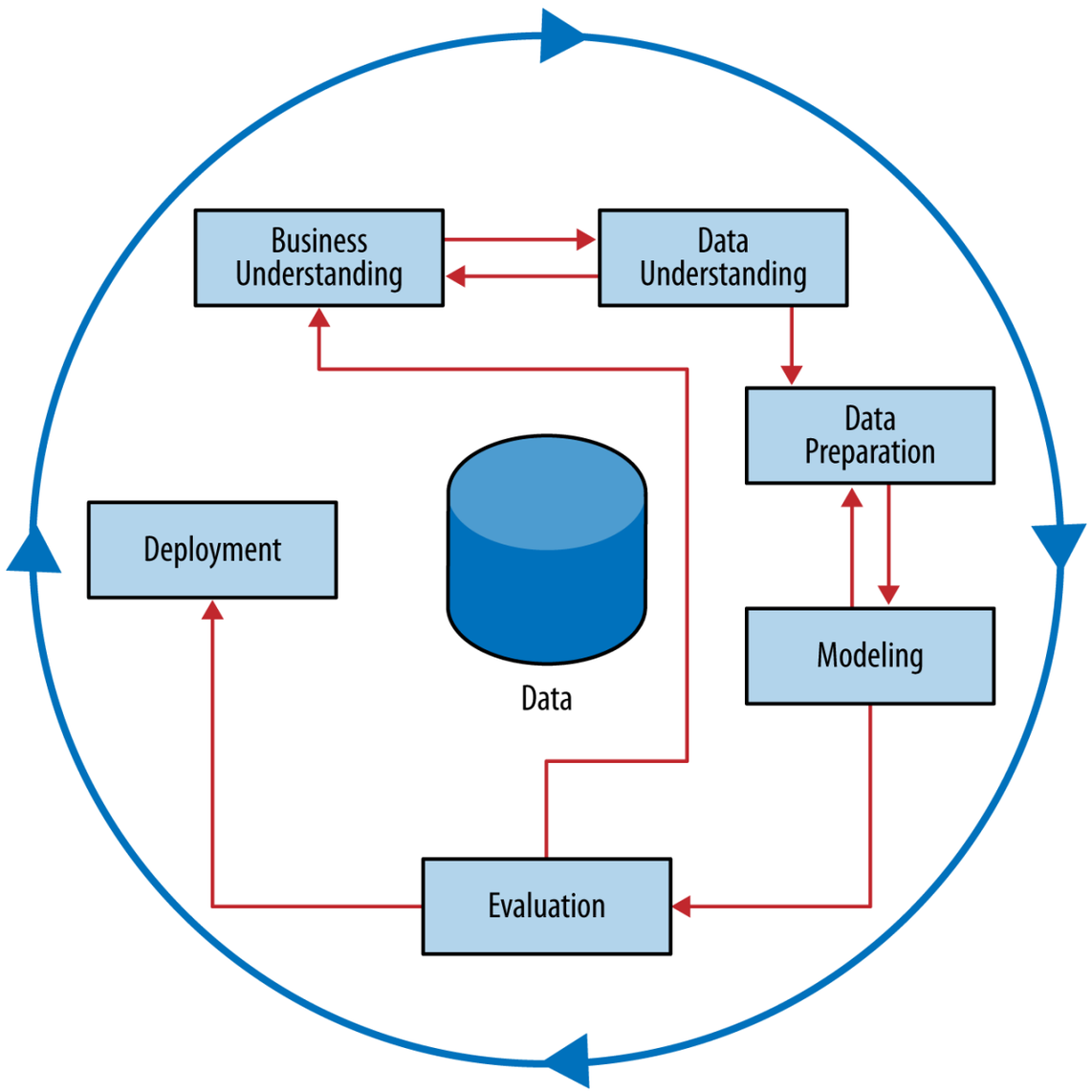


- **Virtual Agents:** Virtual compliance agents deployed to answer queries from enterprise personnel.
- **Text Analytics:** Financial contracts (legal documentation) --> embedded contractual risks
- **Unique Identity:** Fraud detection, cyber-crime
- **Cognitive Robotics:** empower robotics software to learn and evolve in the sophistication and accuracy of some inconsistent or unrepeatable processes to support decision making
- **Video Analytics:** Compliance, audit , automation of report writing.

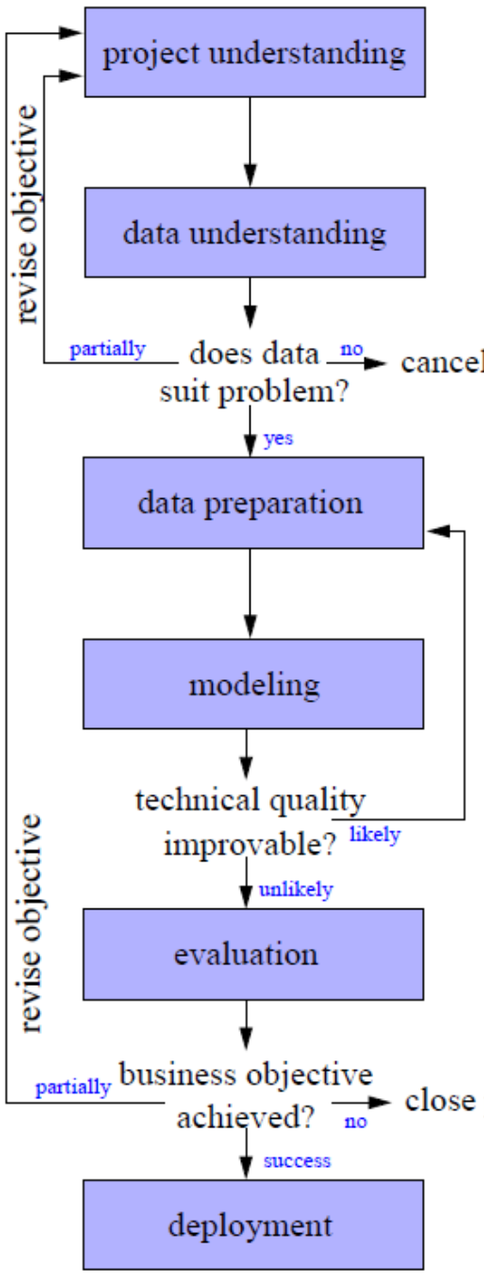
Class Outline:

- **CRoss Industry Standard Process for Data Mining**
- Supervised and Unsupervised Learning
- Data Terminology
- Inputs and Outputs
- Parametric and Nonparametric Methods
- Generative vs Discriminative Models
- Representation and Deep Learning
- Machine Learning workflow

CRoss-Industry Standard Process for Data Mining (CRISP-DM)



CRISP-DM



What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?

How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

Cross
Industry
Standard
Process for
Data **M**ining

Iteration as a rule

Process of data exploration

Intro to Machine Learning

- Goals
  - Learn objectives (patterns, decision functions, mapping) from information at hand (training data)
  - Make accurate predictions on test subjects (test data)

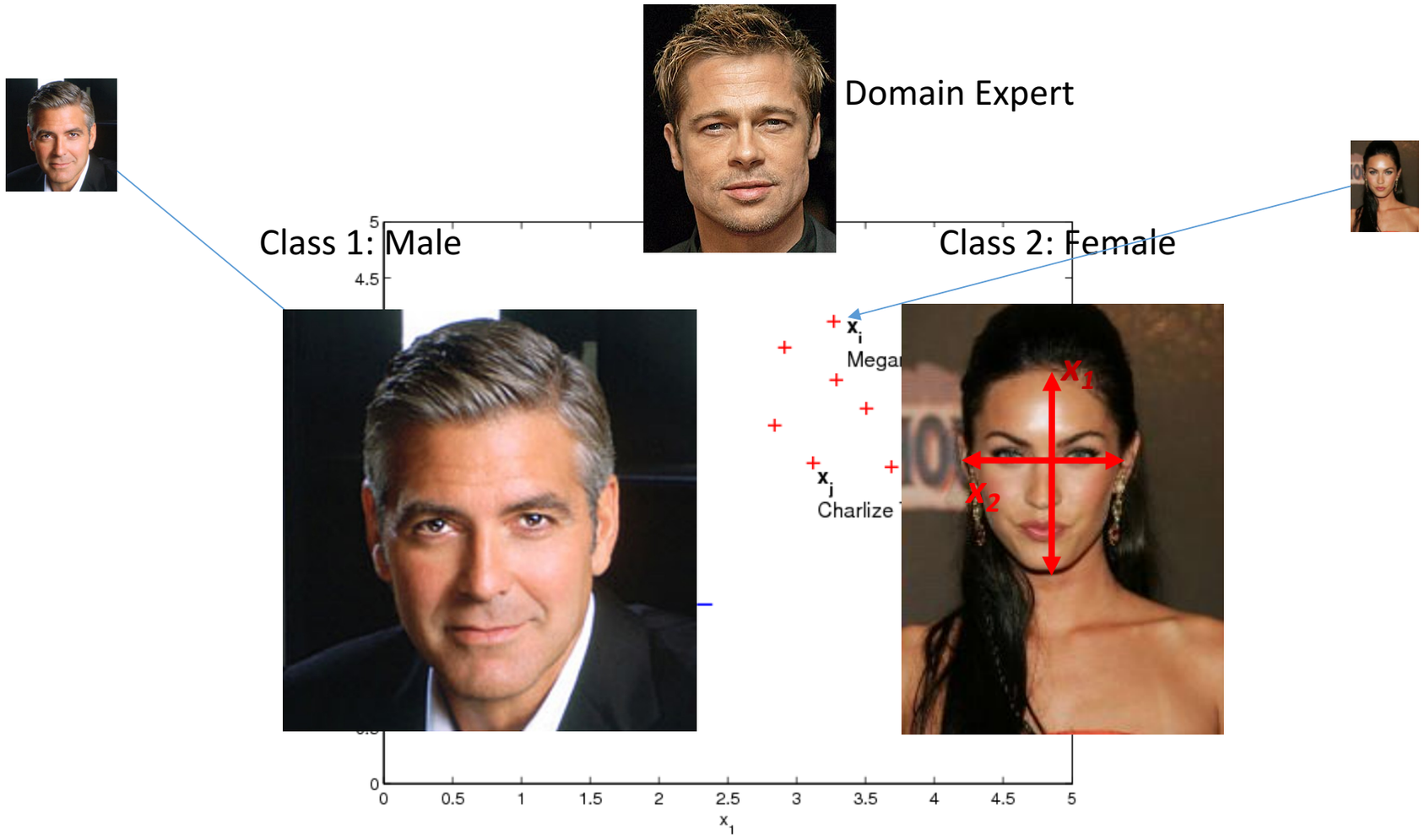
Intro to Machine Learning

- Machine Learning Tasks
  - Supervised Learning
    - Each sample in training data is labeled by a field expert
  - Unsupervised Learning
    - Learning without the supervision of a field expert
  - Semi-supervised Learning
    - A portion of the training data is labeled

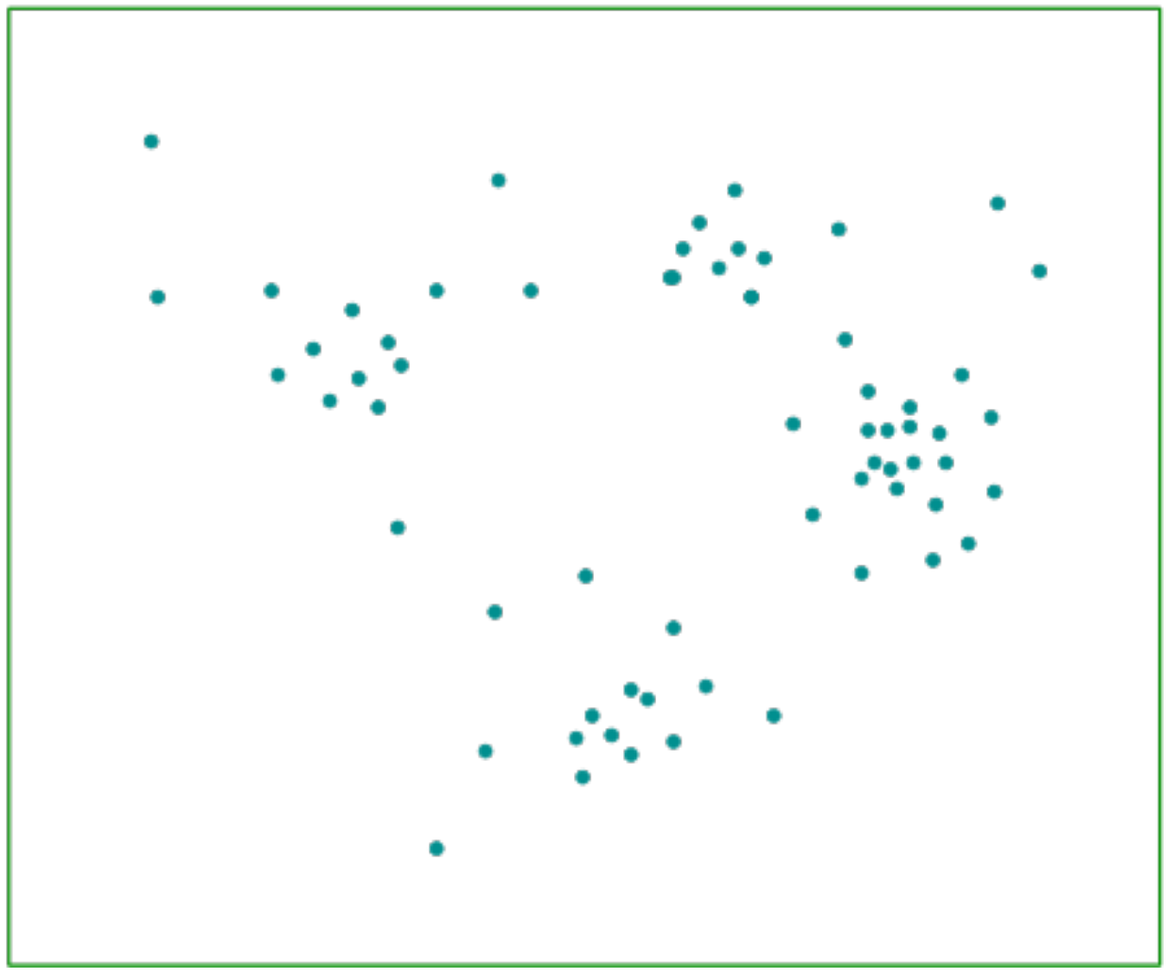
Class Outline:

- CRoss Industry Standard Process for Data Mining
- **Supervised and Unsupervised Learning**
- Data Terminology
- Inputs and Outputs
- Parametric and Nonparametric Methods
- Generative vs Discriminative Models
- Representation and Deep Learning
- Machine Learning workflow

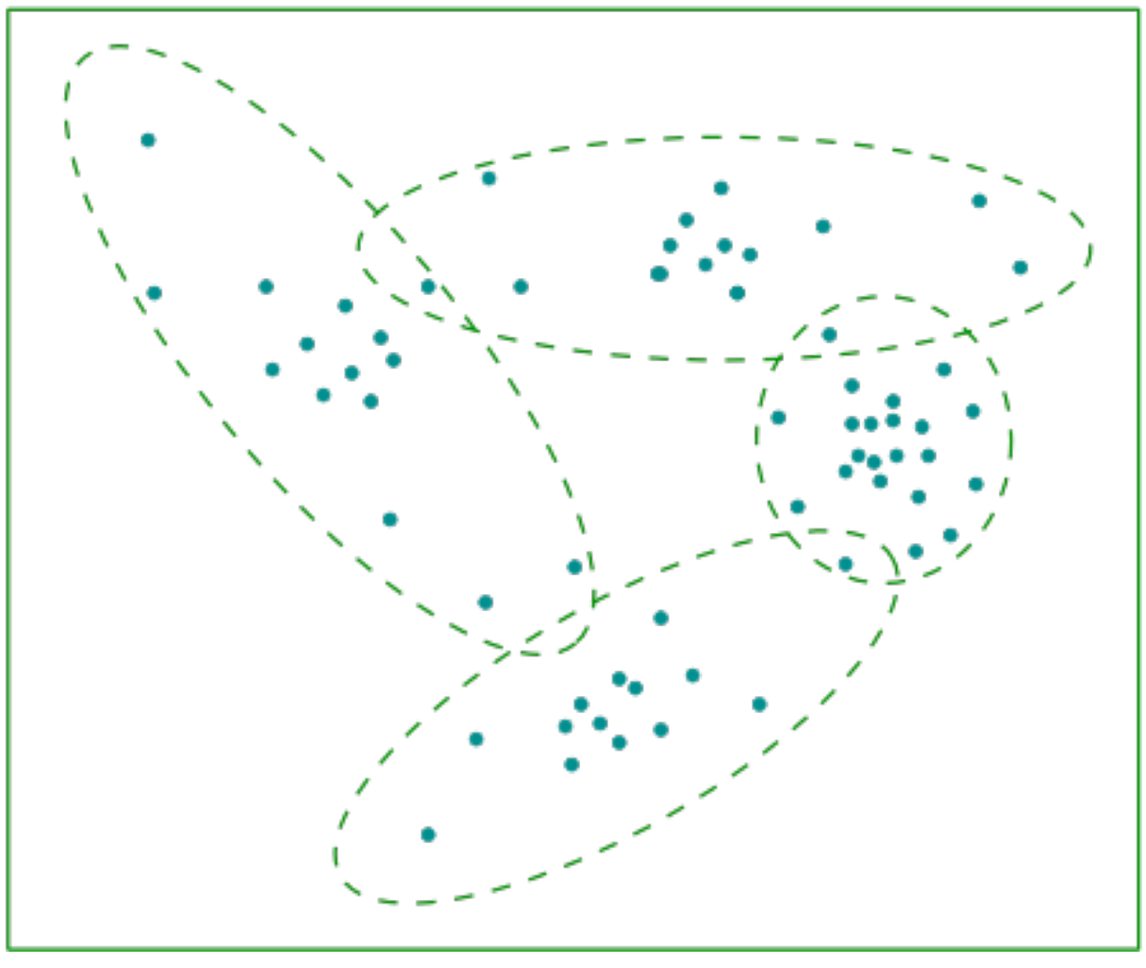
Supervised Learning



Unsupervised Learning



Unsupervised Learning



Supervised or Unsupervised?

Task: Given Google's Previous Stock Prices, Predict the stock prices for the next year

Supervised or Unsupervised?

Task: Given Google's Previous Stock Prices, Predict the stock prices for the next year

Answer: Supervised

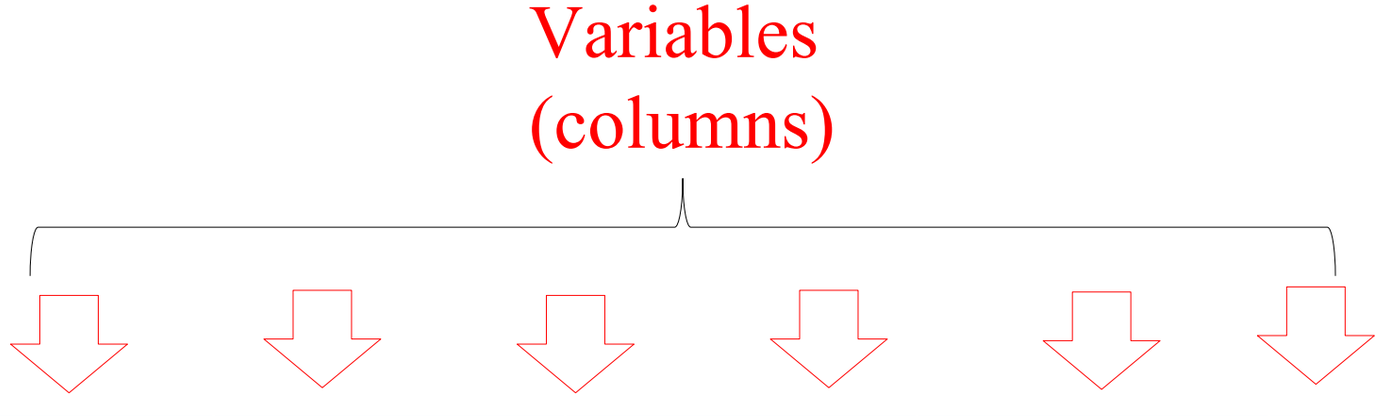
Class Outline:

- Industry Standard Process for Data Mining
- Supervised and Unsupervised Learning
- **Data Terminology**
  - Inputs and Outputs
- Parametric and Nonparametric Methods
- Generative vs Discriminative Models
- Representation and Deep Learning
- Machine Learning workflow

Data Terminology

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology



Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

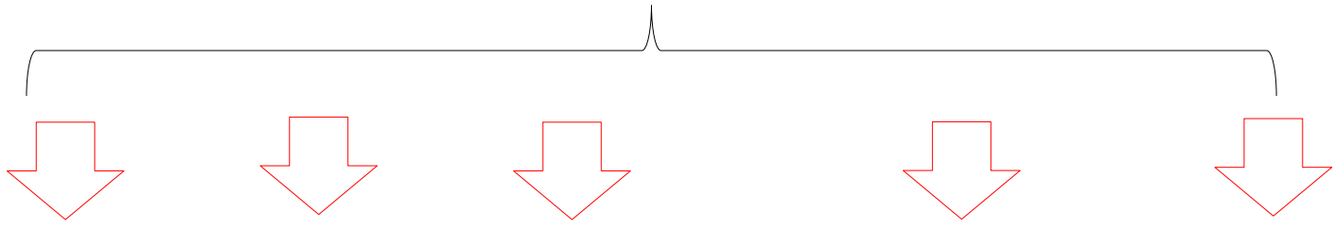
Data Terminology

Target variable
Label
Dependent variable
Output space

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology

Attributes
Features
Explanatory or independent variables



Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology

Records (Data) Instances

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

(17824, 49, M, 12000, -3000) is a feature vector

Data Terminology

Target variable = Income

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology

NO Target variable

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Machine learning versus Algorithms

In both fields, the goal is to develop

procedures that exhibit a desired input-output behavior.

- **Algorithms**: the input-output mapping can be precisely defined.

  Input: Graph G .

  Output: Minimum Spanning Tree of G .

- **Machine learning**: the mapping cannot easily be made precise.

  Input: Picture of an animal.

  Output: Name of the animal.

Instead, we simply provide examples of (input,output) pairs and ask the machine to *learn* a suitable mapping itself.

Class Outline:

- Industry Standard Process for Data Mining
- Supervised and Unsupervised Learning
- Data Terminology
- **Inputs and Outputs**
- Parametric and Nonparametric Methods
- Generative vs Discriminative Models
- Representation and Deep Learning
- Machine Learning workflow

Inputs and outputs

Basic terminology:

- The input space, X .

  E.g. 32 × 32 RGB images of animals

- The output space, Y.

  E.g Names of 100 animals.

  .

x :

y : "bear"

After seeing a bunch of examples (x , y), pick a mapping

$$f : X \rightarrow Y$$

that accurately replicates the input-output pattern of the examples.

Learning problems are often categorized according to the type of *output space:* (1) discrete, (2) continuous, (3) probability values, or (4) more general structures

Discrete output space: classification

Binary classification:

- Spam detection

  X = {email messages}

  Y = {spam, not spam}

- Credit card fraud detection

  X = {descriptions of credit card transactions}

  Y = {fraudulent, legitimate}

Multiclass classification:

- Animal recognition

  X = {animal pictures}

  Y = {dog, cat, giraffe, . . .}

- News article classification

  X = {news articles} Y = {politics, business, sports, . . .}

Continuous output space: regression

- A parent's concerns

  How cold will it be tomorrow morning?

  $Y = [-273, \infty)$

- For the asthmatic

  Predict tomorrow's air quality (max over the whole day)

  $Y = [0, \infty)$ (< 100: okay, > 200: dangerous)

- Insurance company calculations

  In how many years will this person die?

  $Y = [0, 200]$

What are suitable predictor variables (X) in each case?

Conditional probability functions

Here $Y = [0, 1]$ represents probabilities.

- Dating service

  What is the probability these two people will go on a date if introduced to each other?

  If we modeled this as a classification problem, the binary answer would basically always be “no”. The goal is to find matches that are slightly less unlikely than others.

- Credit card transactions

  What is the probability that this transaction is fraudulent?

  The probability is important, because – in combination with the amount of the transaction – it determines the overall risk and thus the right course of action.

Structured output spaces

The output space consists of structured objects, like sequences or trees.

Dating service

Input: description of a person
Output: rank-ordered list of all possible matches

Y = space of all permutations

Example:
x = Tom
y = (Nancy , Mary , Chloe, . . .)

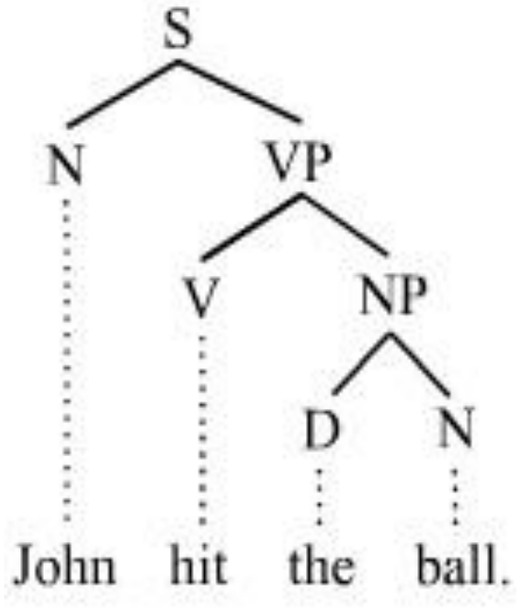
Language processing

Input: English sentence
Output: parse tree showing grammatical structure

Y = space of all trees

Example:
x = "John hit the ball"

y =

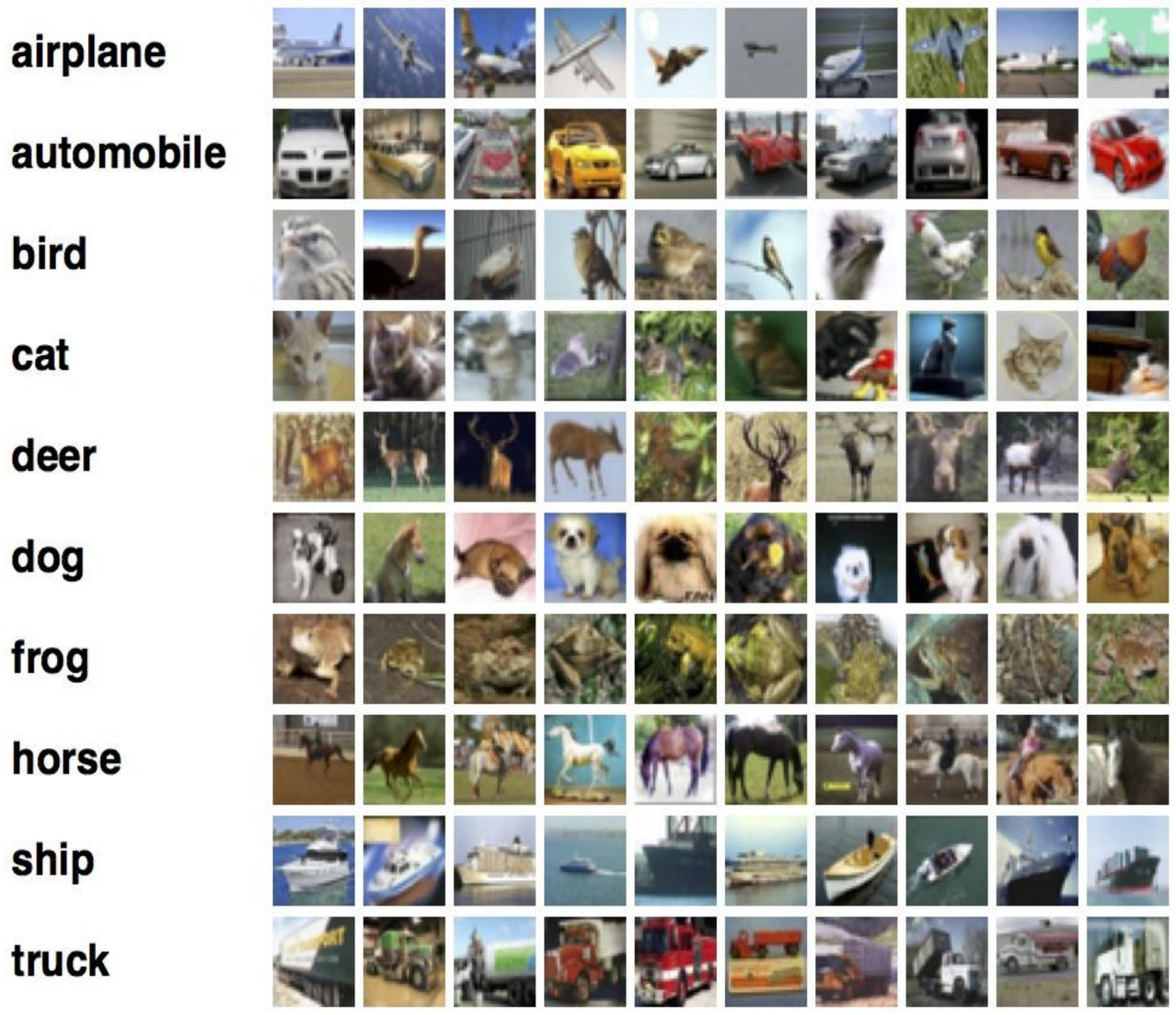


Class Outline:

- ➢ Industry Standard Process for Data Mining
- ➢ Supervised and Unsupervised Learning
- ➢ Data Terminology
- Ø Inputs and Outputs
- ➢ **Parametric and Nonparametric Methods**
- ➢ Generative vs Discriminative Models
- ➢ Representation and Deep Learning
- ➢ Machine Learning workflow

Nonparametric methods: nearest neighbor

Training set: a collection of (x , y) pairs:

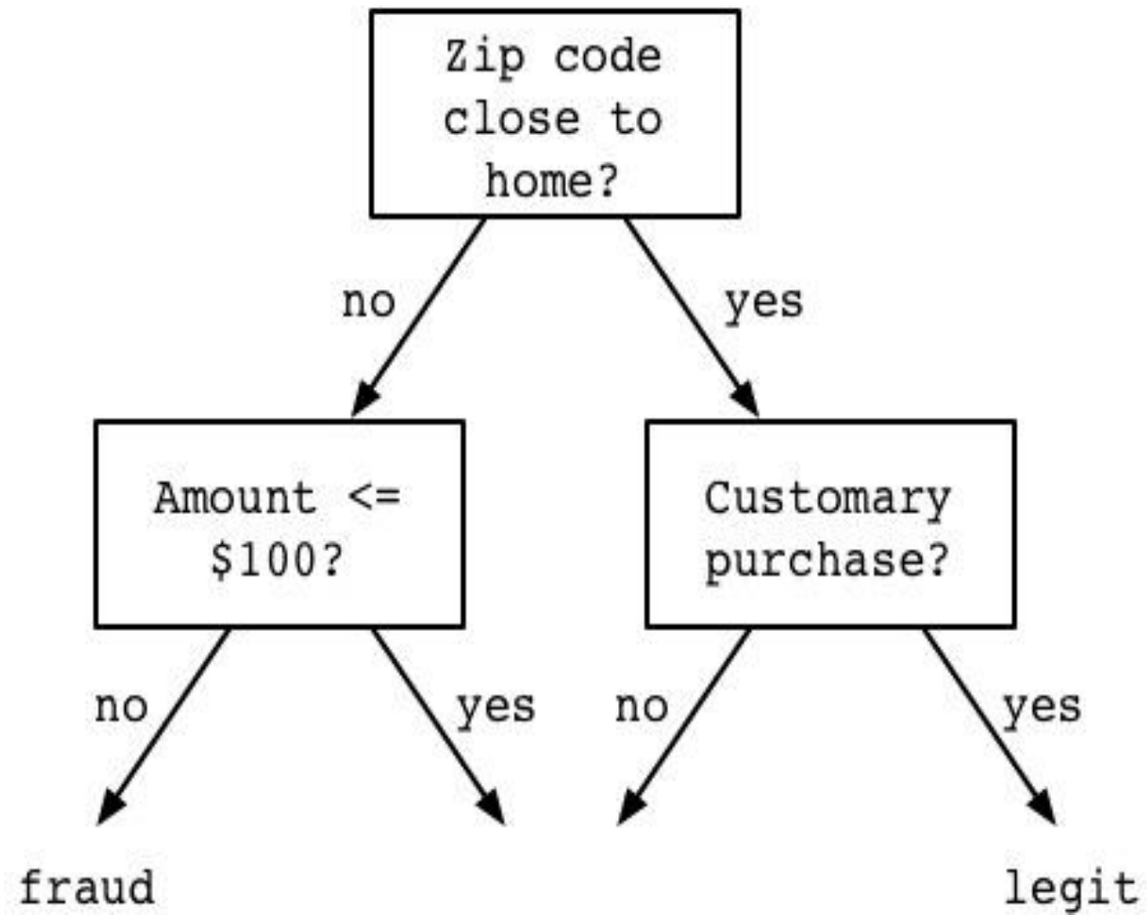


Given any x , find its nearest neighbor in the training set and predict that neighbor's y value.

Issues: (1) What distance function? (2) How to speed up search?

Nonparametric methods: decision tree

Credit card fraud detection: use training data to build a tree classifier



What do nearest neighbor and decision trees have in common?

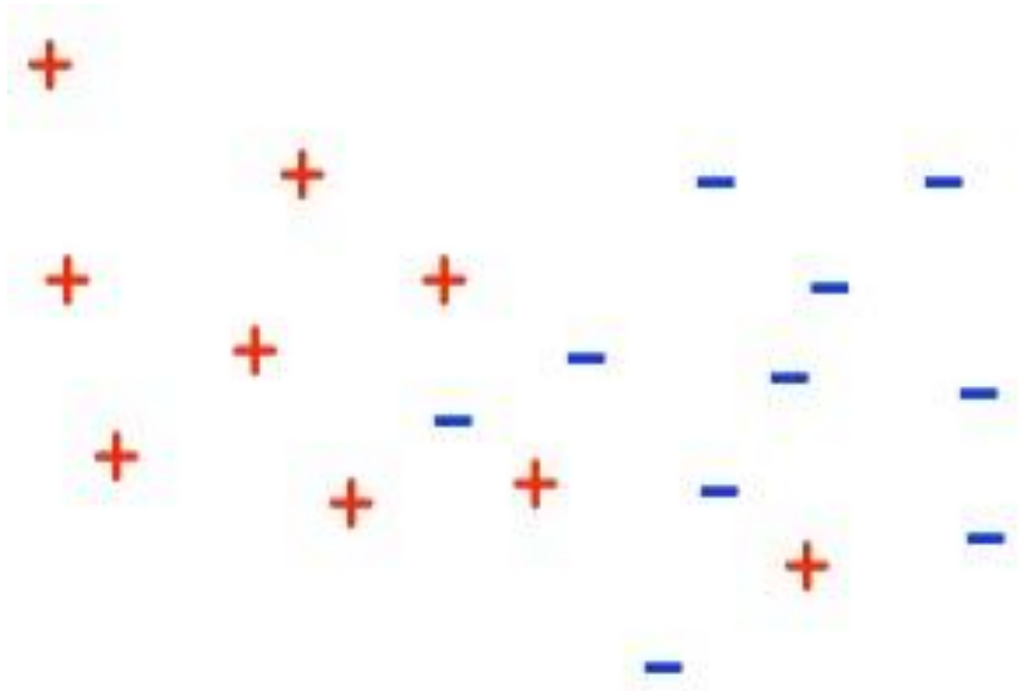
- Unbounded in size
- Can model arbitrarily complex functions

They are *nonparametric methods*.

Classification with parametrized models

Classifiers with a fixed number of parameters can represent a limited set of functions. Learning a model is about picking a good approximation.

Typically the *x* 's are points in *d* -dimensional Euclidean space, R^d



Two ways to classify:

- *Generative: model the individual classes.*
- *Discriminative: model the decision boundary between the classes.*

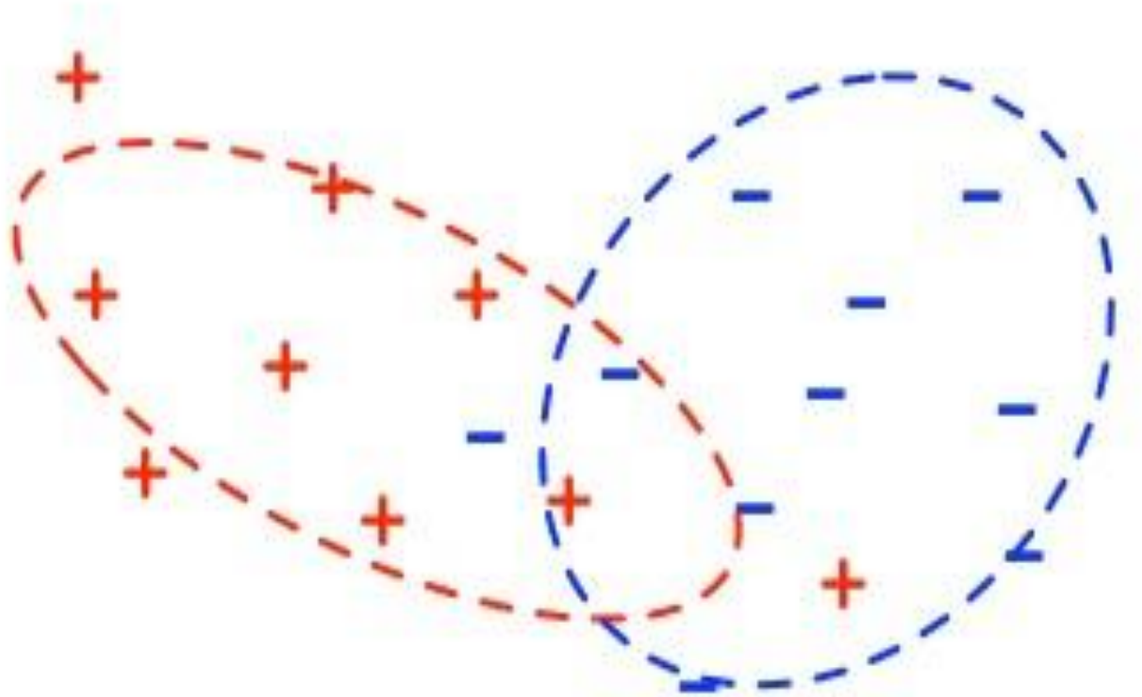
Class Outline:

- ➢ Industry Standard Process for Data Mining
- ➢ Supervised and Unsupervised Learning
- ➢ Data Terminology
- Ø Inputs and Outputs
- ➢ Parametric and Nonparametric Methods
- ➢ **Generative vs Discriminative Models**
- ➢ Representation and Deep Learning
- ➢ Machine Learning workflow

Generative models

Fit a probability distribution – like a multivariate Gaussian – to each class.
Thereafter use this *summary* rather than the data points themselves.

To classify a new point: find the most probable class.



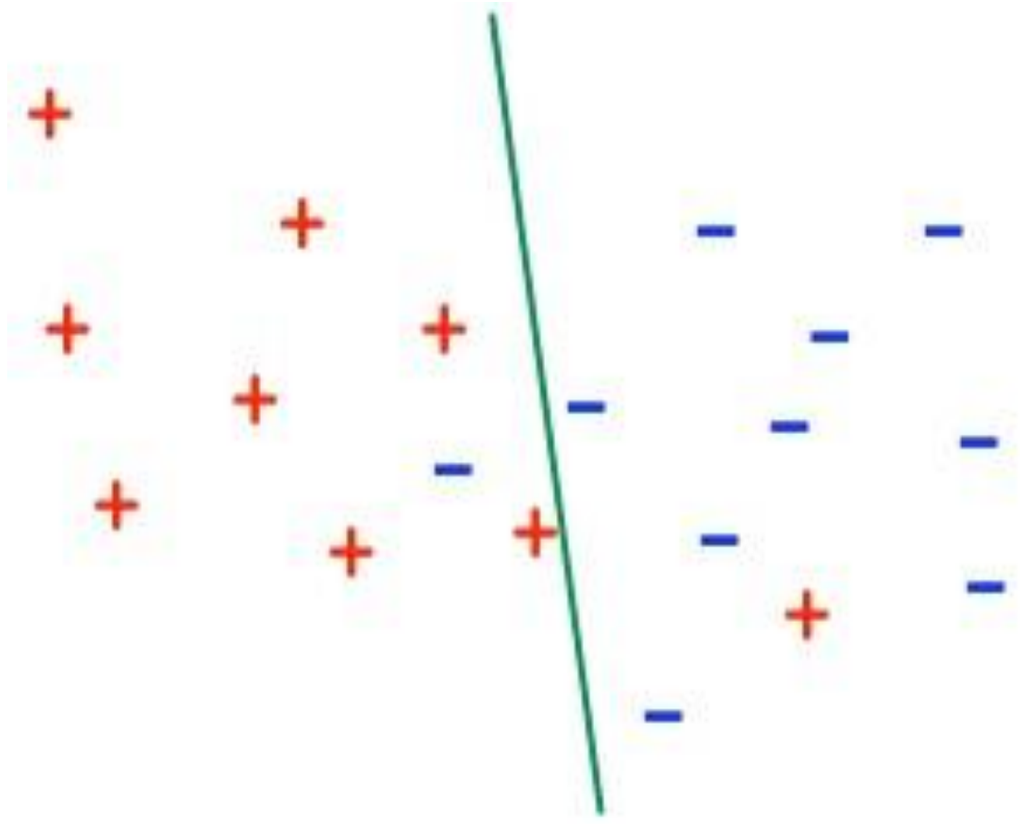
Examples: Naive Bayes, Fisher discriminant.

Under the hood: Bayes' rule, linear algebra (eigenvalues, eigenvectors).

Discriminative models

Approximate the boundaries between classes by simple – e.g. linear functions.

To classify a new point: figure out which side of the boundary it lies on.



Examples: support vector machine, logistic regression.

Under the hood: convex duality, optimization.

Generalization theory

- Complex, e.g. nonparametric, classifiers require a lot of training data to learn accurately.
- Simple, e.g. linear, classifiers require less.

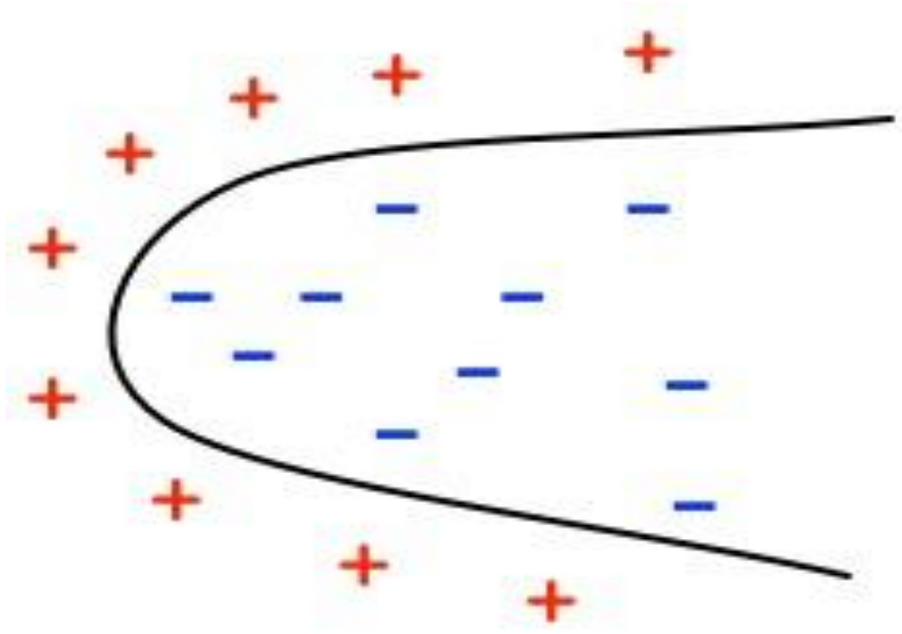
What is the right notion of complexity? Are there formulas for how much data is enough? The answers are based on *large deviation theory*.

Richer classifiers via the kernel trick

We are good at finding linear classifiers in Euclidean space. But what if:

- The boundary between classes is far from linear?

  Example: quadratic, or higher-order polynomial, or even stranger.



- The data aren't even vectors of numbers?

  Example: documents, DNA sequences, parse trees.

The *kernel trick* handles these scenarios seamlessly, by mapping the data to a suitable Euclidean space in which linear classification is possible!

Richer output spaces

Many classification methods were developed for the binary (two-label) case. Usually the output space is larger than this.

- *Y = several classes.*

  Examples:

  x = image, y = name of object in image

  x = news article, y = category (sports, politics, business, . . .)

- *Y = structured objects.*

  Examples:

  x = sentence in Swahili, y = transcription into English

  x = sentence in English, y = parse tree

Extend binary classification to handle such cases!

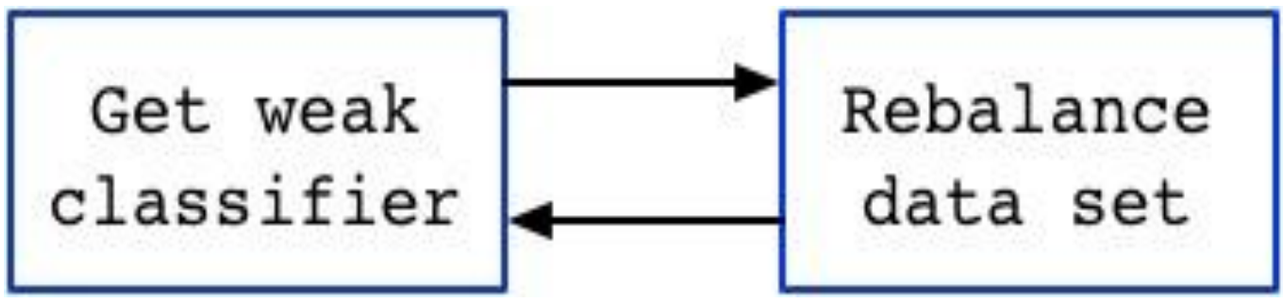
Under the hood: error-correcting codes, dynamic programming.

Composing simple classifiers

A common situation in classifier learning:

*Easy to find **weak classifiers** – not very accurate, but better than random to increase accuracy, compose weak classifiers.*

Example: *boosting*



Final classifier is a linear combination of all these weak classifiers.

Generically improve the performance of *any* kind of classifier!

Class Outline:

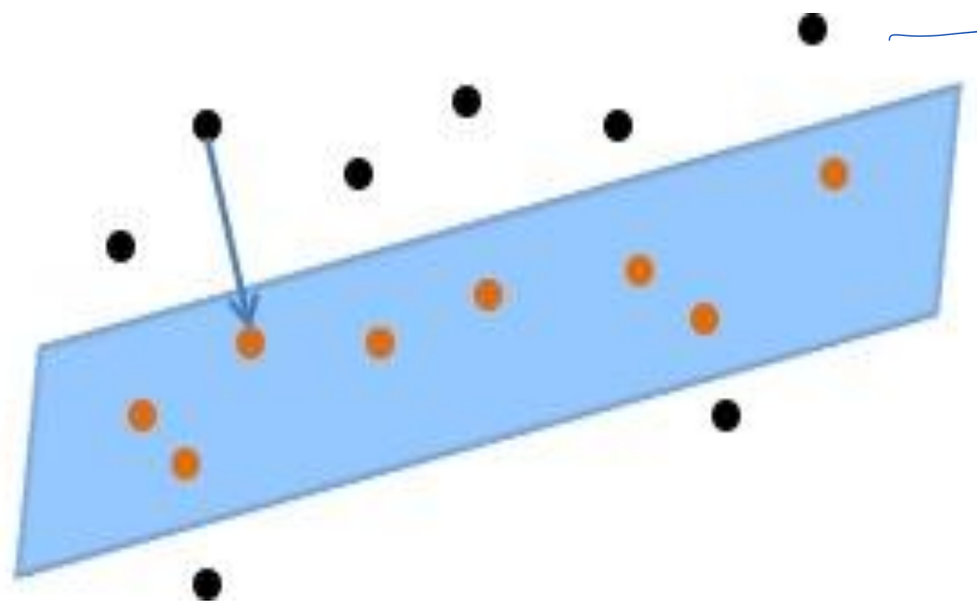
- ➢ Industry Standard Process for Data Mining
- ➢ Supervised and Unsupervised Learning
- ➢ Data Terminology
- Ø Inputs and Outputs
- ➢ Parametric and Nonparametric Methods
- ➢ Generative vs Discriminative Models
- ➢ **Representation and Deep Learning**
- ➢ Machine Learning workflow

Representation learning

A handful of key primitives:

1. Dimensionality reduction and denoising.

   Given data in high-dimensional Euclidean space, project to a low-dimensional linear subspace while retaining as much of the signal as possible.



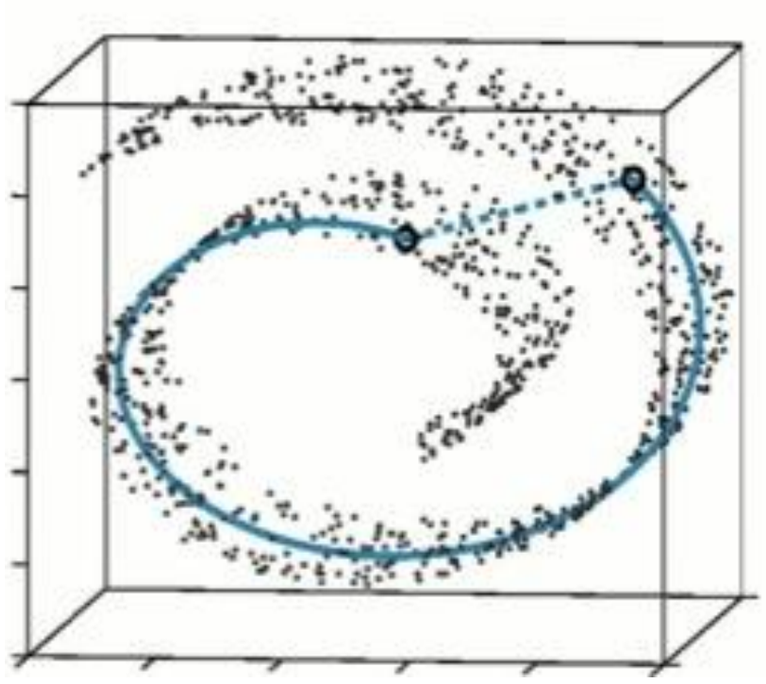
2. Embedding and manifold learning.
3. Metric learning.

Representation learning

A handful of key primitives:

1. Dimensionality reduction and denoising.
2. Embedding and manifold learning.

   Given data that lie in a non-Euclidean space, find an embedding into Euclidean space that preserves as much of the geometry as possible.



3. Metric learning.

Representation learning

A handful of key primitives:

1. Dimensionality reduction and denoising.
2. Embedding and manifold learning.
3. Metric learning.

   Given data with only vague positional information, impose an Euclidean geometry that is suitable for classification.

   Example: X = {a collection of m books}.

   A user supplies $\binom{m}{2}$ similarity ratings, such as:

   ("Pride and Prejudice", "Great Expectations"): similar

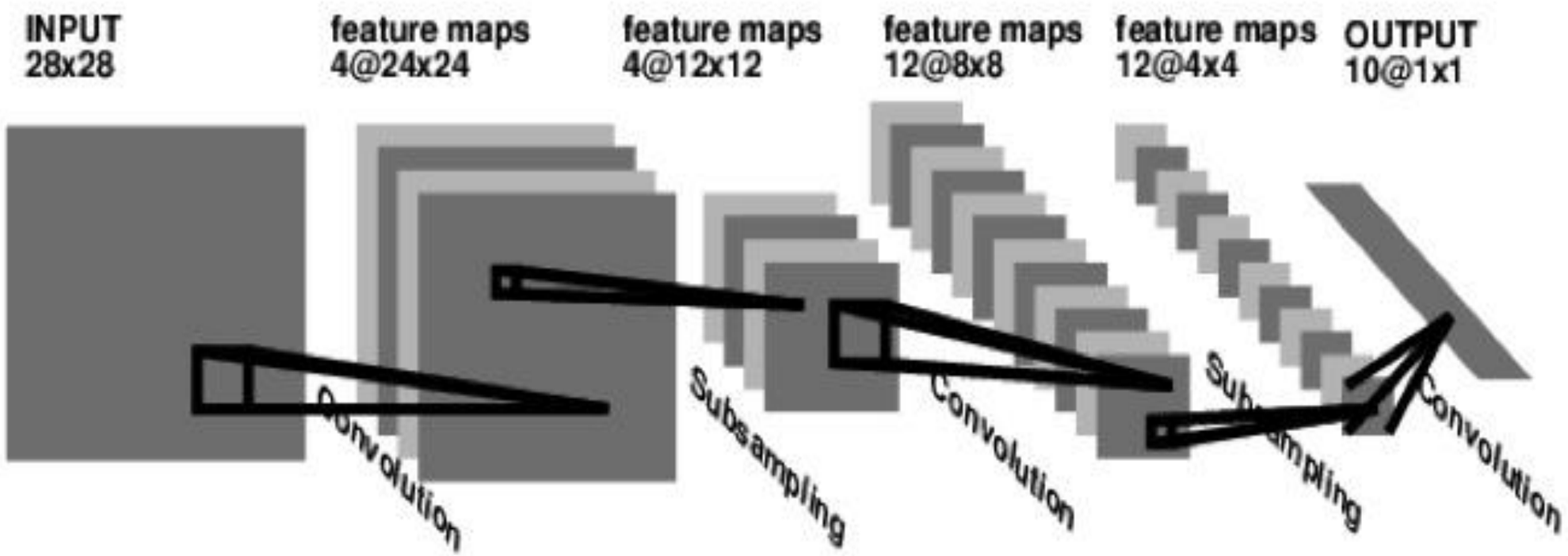
   ("Hamlet", "Great Expectations"): dissimilar

   Represent each book by a vector, respecting these ratings.

Under the hood: linear algebra, semidefinite programming.

Deep learning

Multi-layer neural nets achieve state-of-the-art performance across a range of benchmark problems in natural language processing, speech, and vision.



Under the hood: stochastic gradient descent, dictionary learning, autoencoders.

Identifying Machine Learning Tasks

- Will this customer purchase service S1 if given incentive I1 ?
  - Supervised Learning
    - Classification Problem
    - Binary Target(the customer either purchases or not)

- Which Service package (S1,S2,or none) will a customer likely purchase if given incentive I1 ?
  - Supervised Learning
    - Classification Problem
    - Three valued target

Identifying Machine Learning Tasks

"I want to know which of my customers are the most profitable?"

Database query

"I have a budget to target 10,000 existing customers with a special offer. I would like to identify those customers most likely to respond to the special offer"

Supervised Learning

Probability ranking - Classification problem

Identifying Machine Learning Tasks

How can we categorize our customers?

- Unsupervised Learning
  - Clustering

“How much will this customer use the service?”

- Supervised Learning
  - Regression problem
    - Numeric target
    - Target variable: amount of usage per customer

Identifying Machine Learning Tasks

"I would like to segment my customers into groups based on their demographics and prior purchase activity. I am not focusing on improving a particular task, but would like to generate ideas."

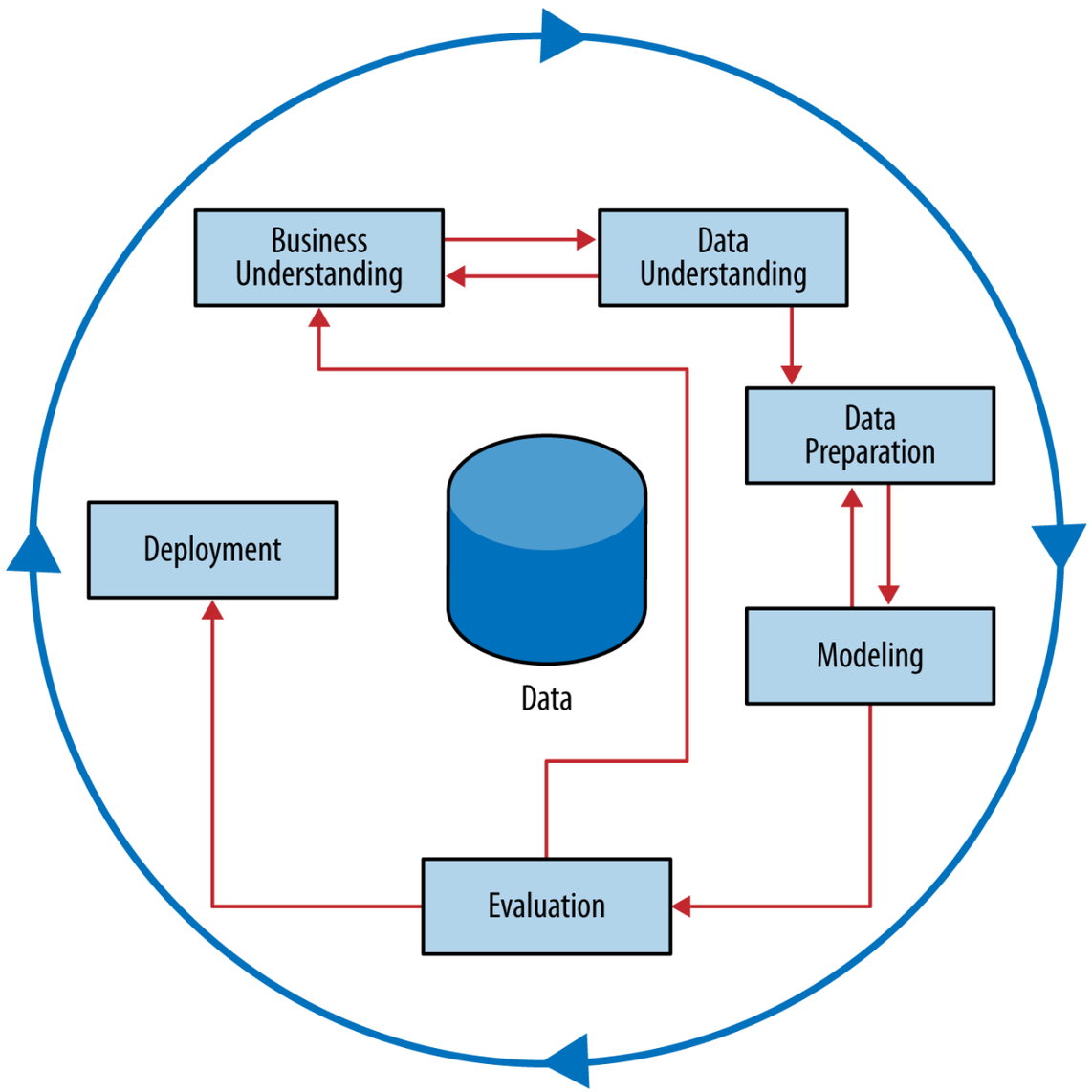
Unsupervised Learning

Clustering

Class Outline:

- ➢ Industry Standard Process for Data Mining
- ➢ Supervised and Unsupervised Learning
- ➢ Data Terminology
- Ø Inputs and Outputs
- ➢ Parametric and Nonparametric Methods
- ➢ Generative vs Discriminative Models
- ➢ Representation and Deep Learning
- ➢ **Machine Learning workflow**

Workflow

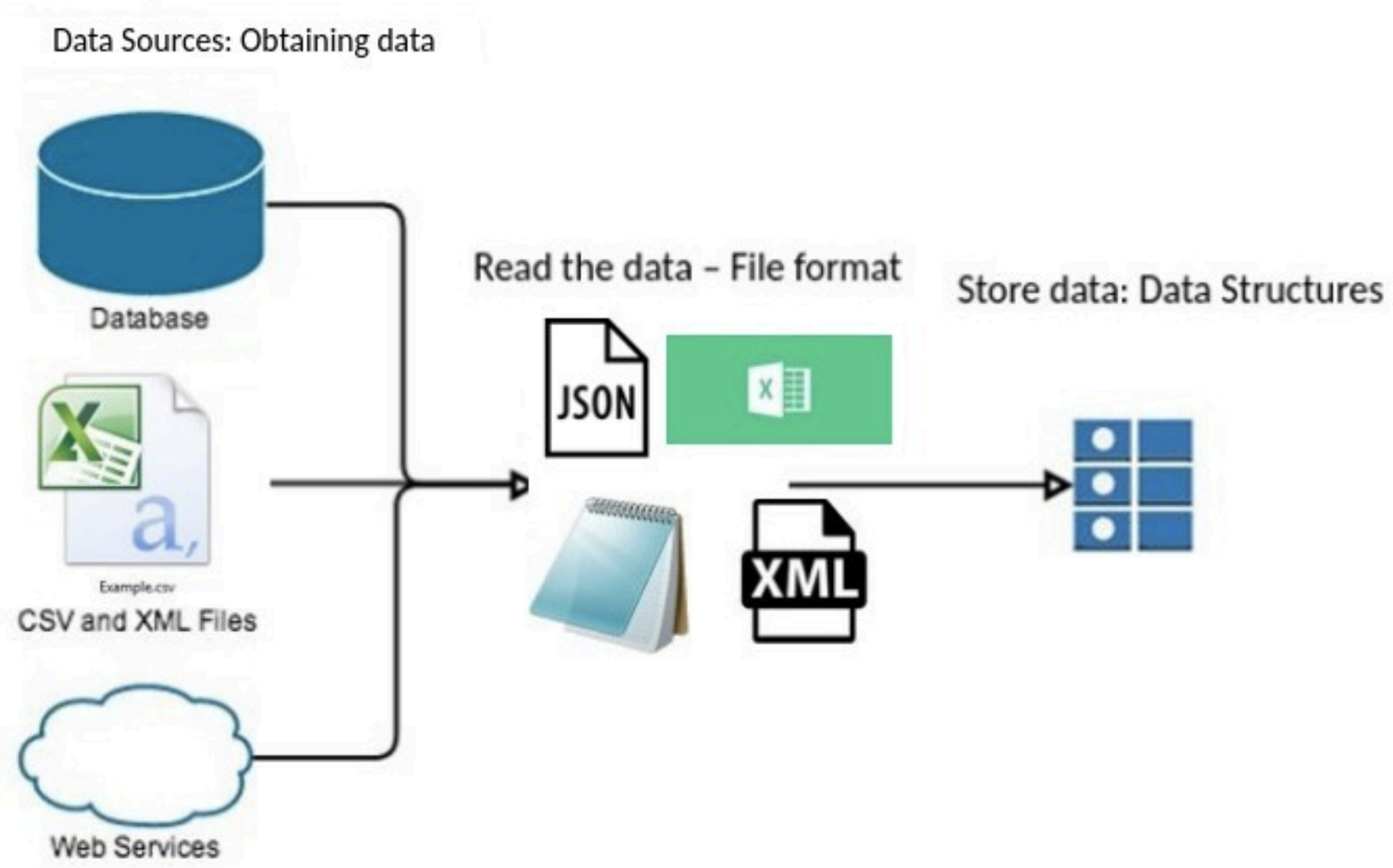

Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data

Gathering Data

Data can be gathered from many sources like :

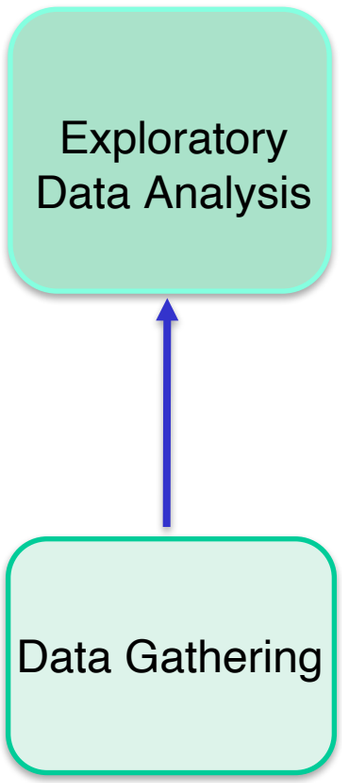
- Databases
- Web Services
- Internet
- Locally stored files (.csv or .xml files)

Data Gathering

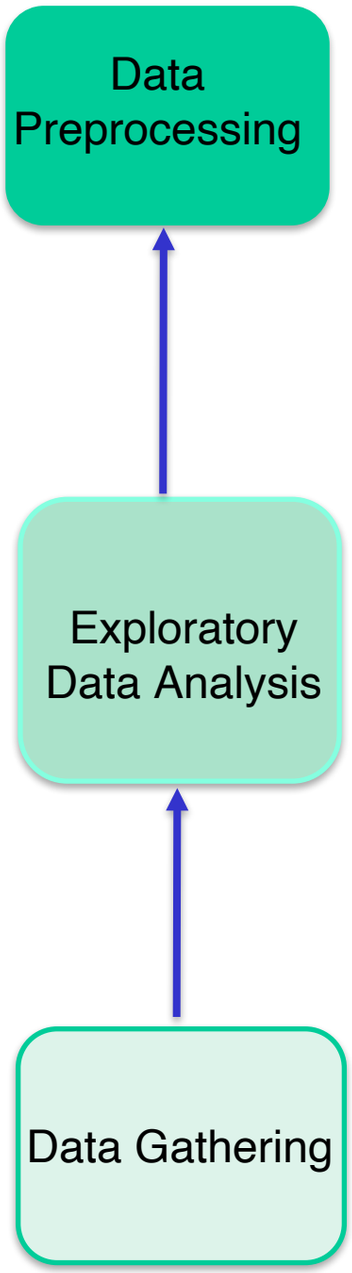


Data Gathering

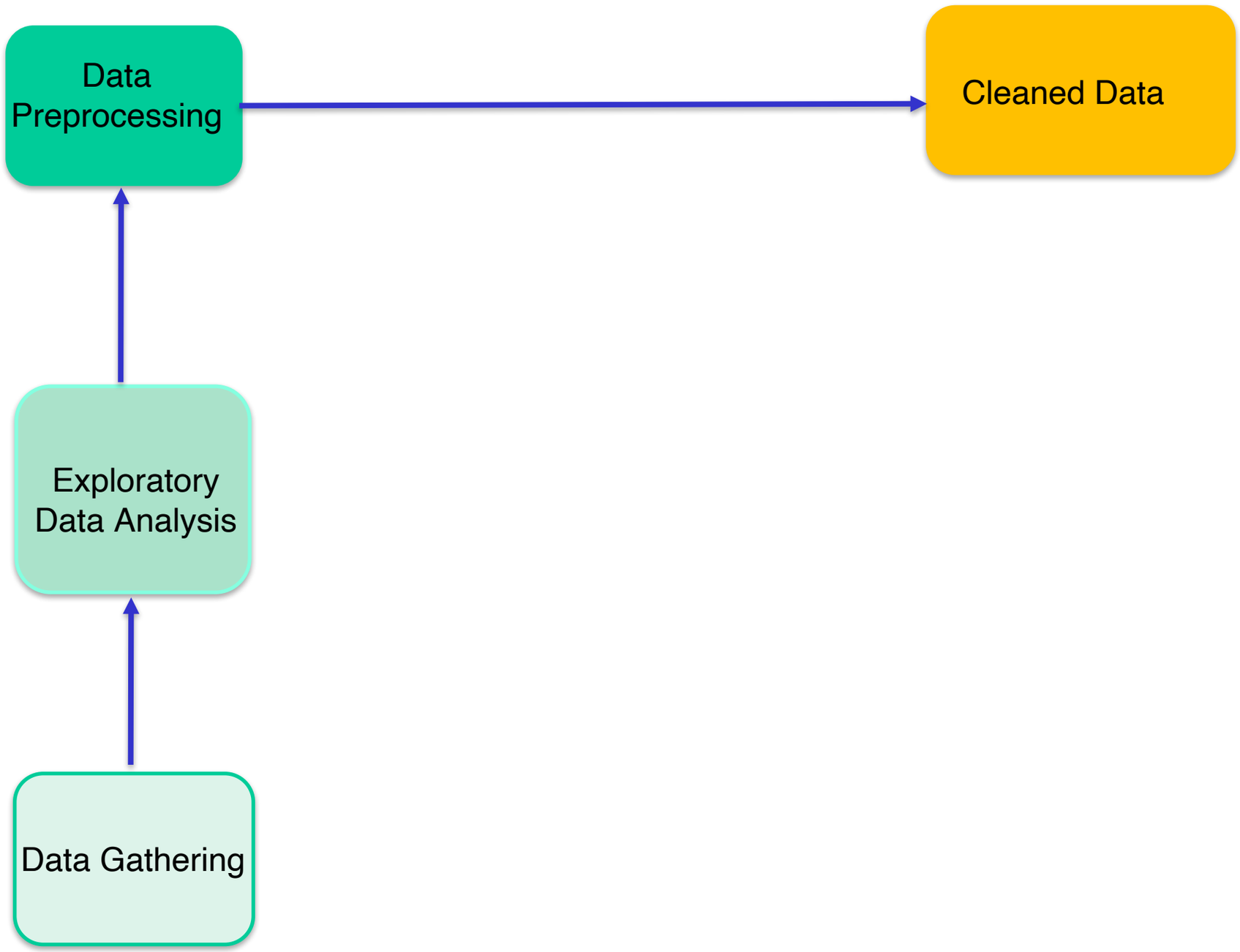
Exploratory Data Analysis



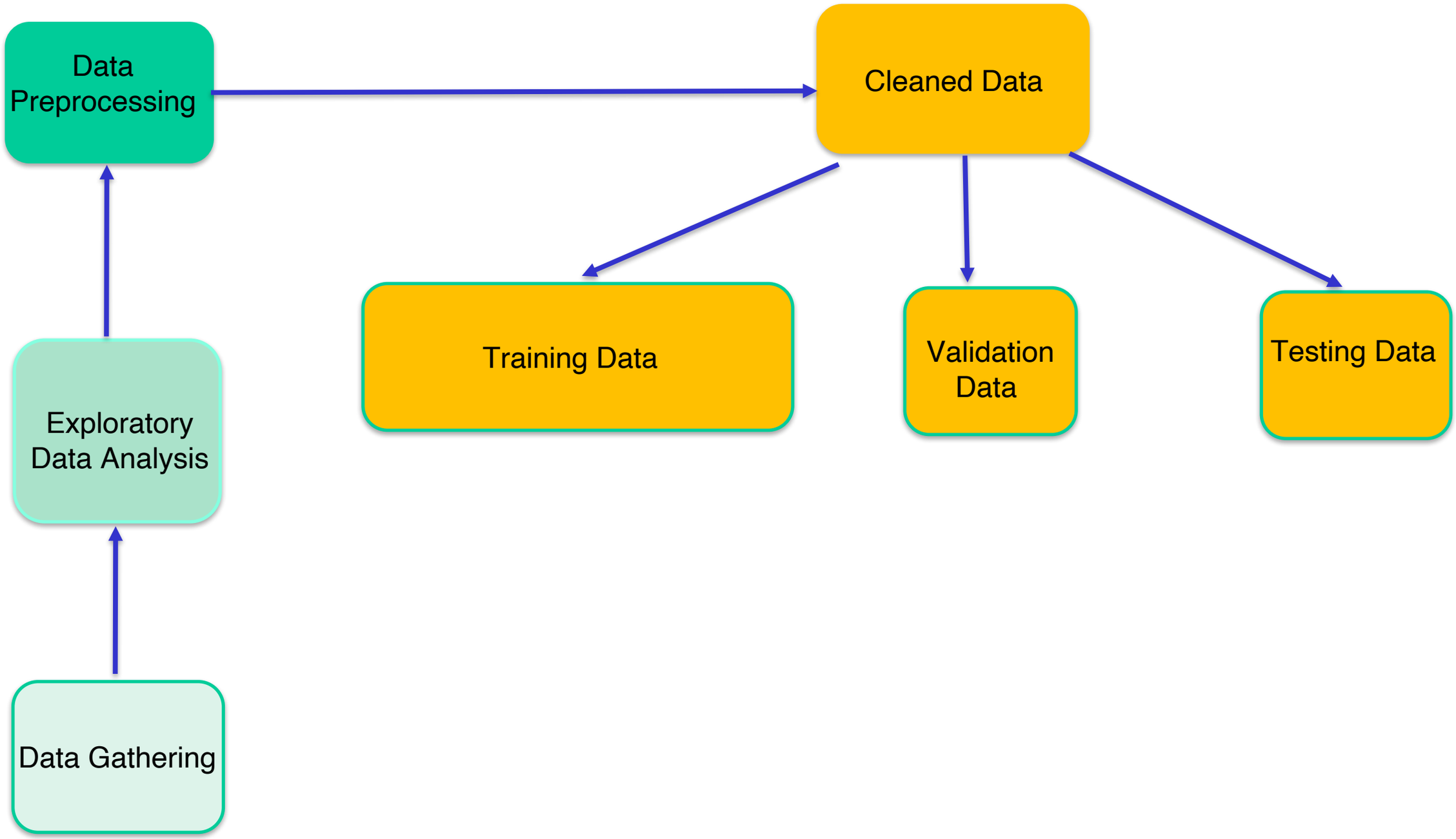
Data Wrangling/Data Pre-processing



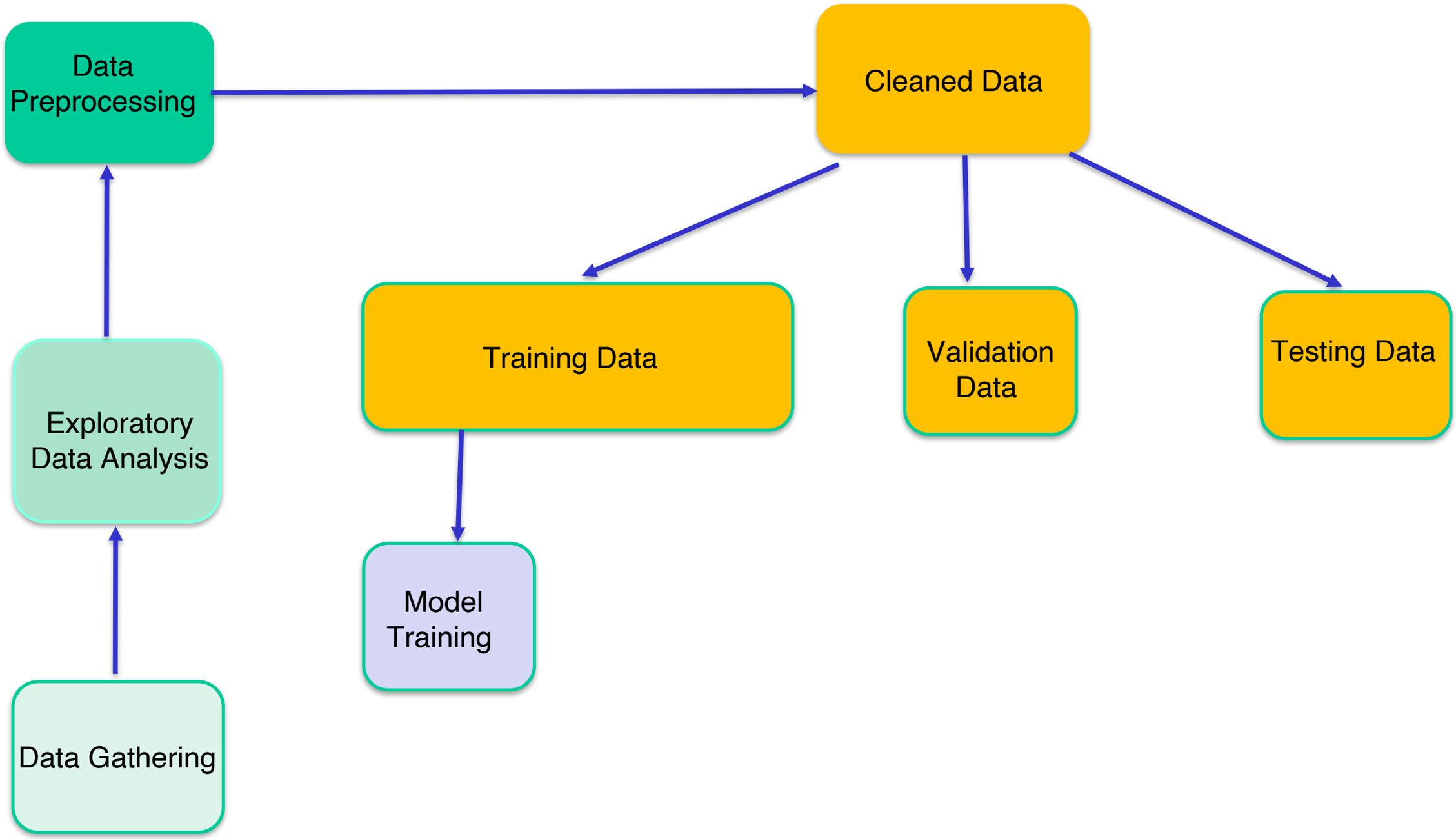
Data Formatting



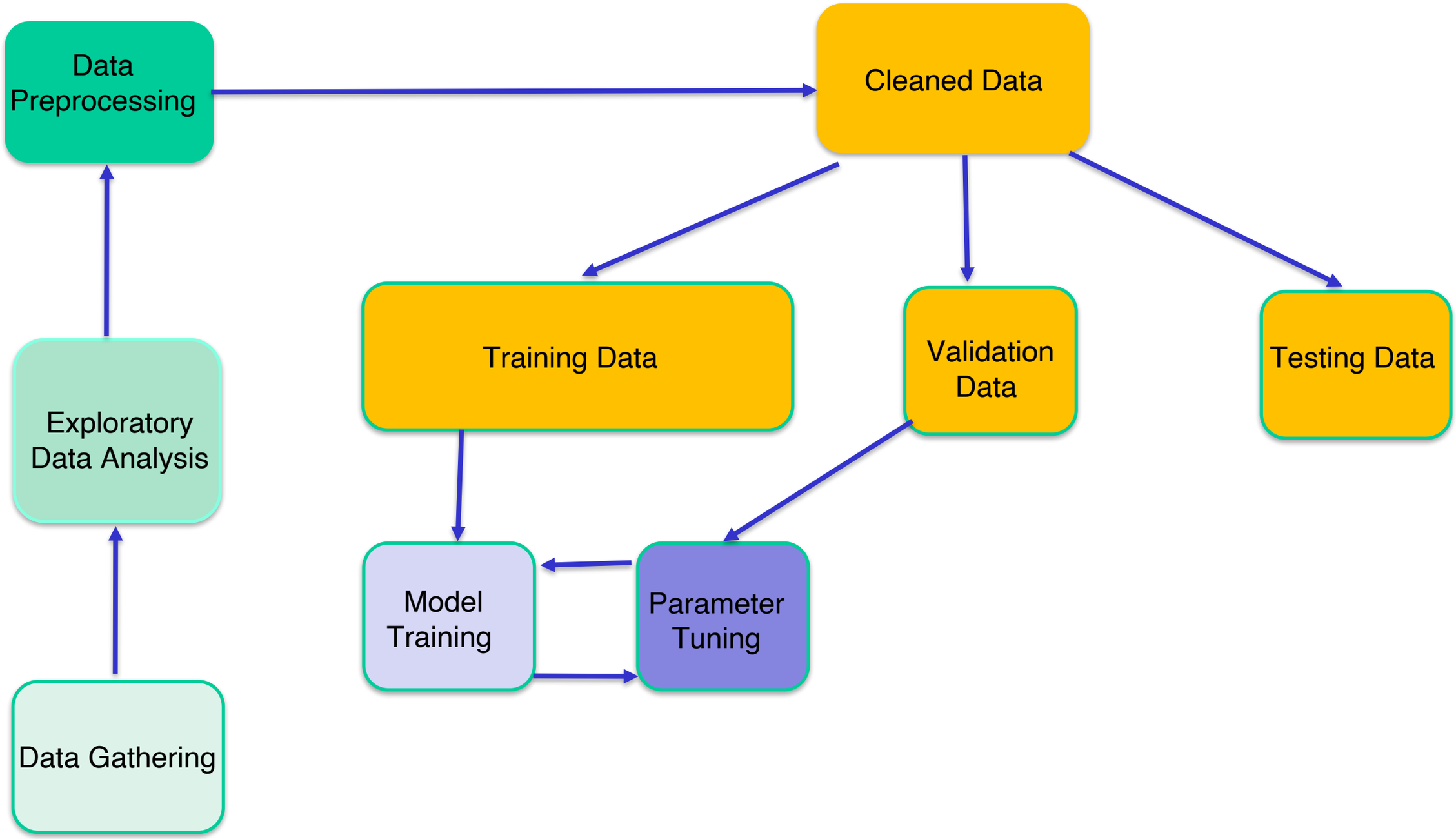
Splitting the data



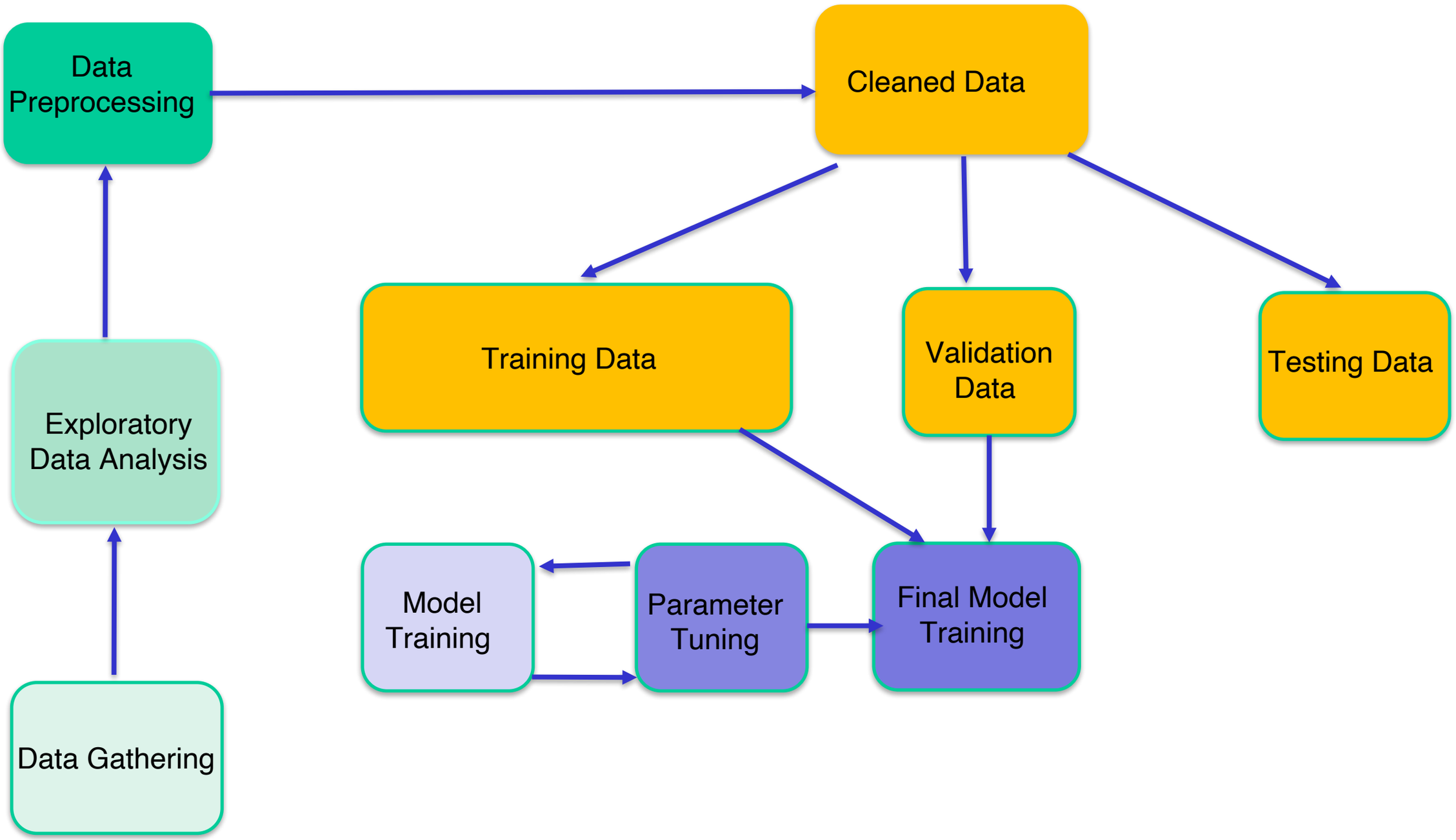
Model Training



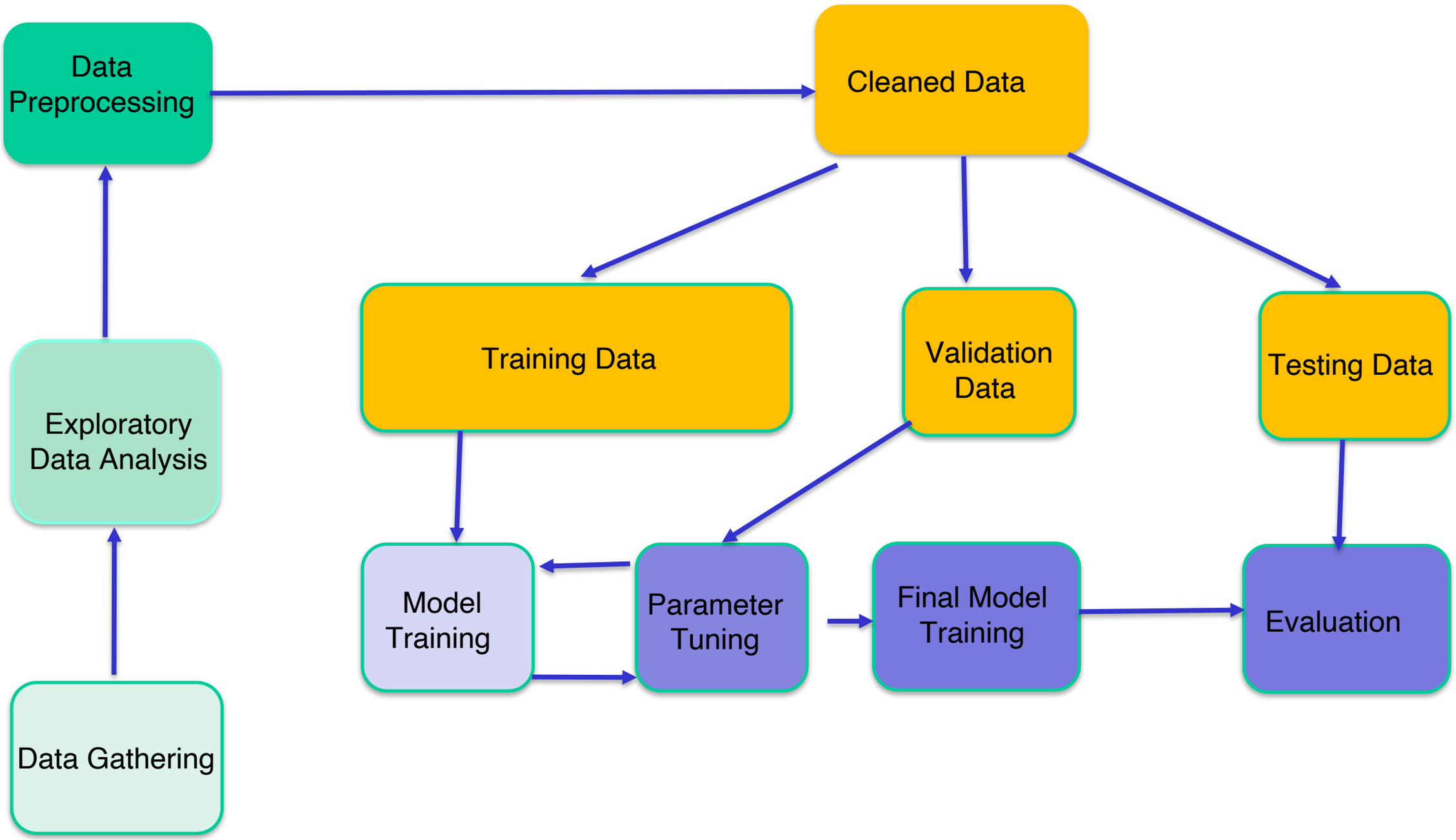
Hyperparameter Tuning



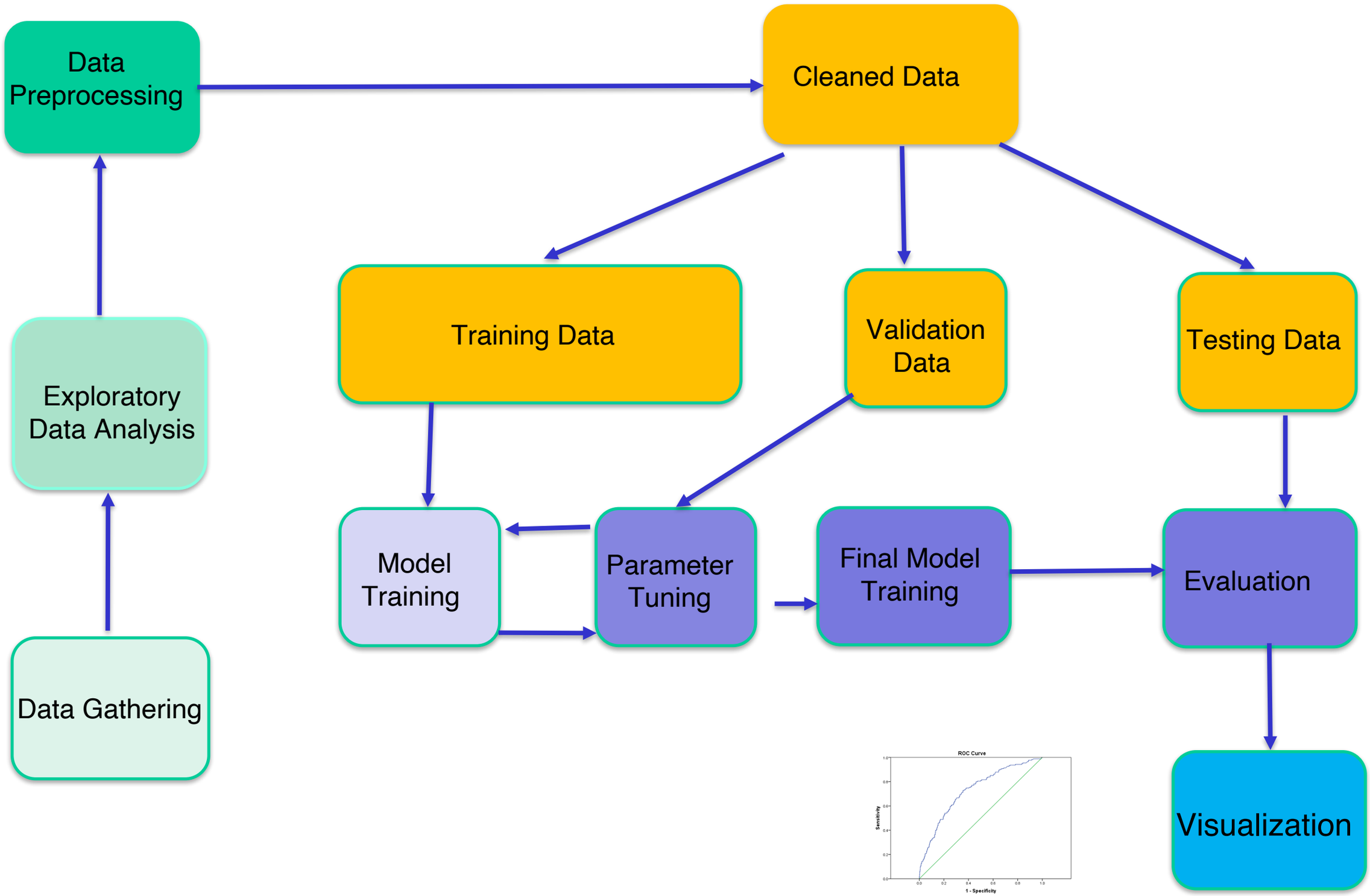
Retraining a final model



Model Evaluation



Model Evaluation



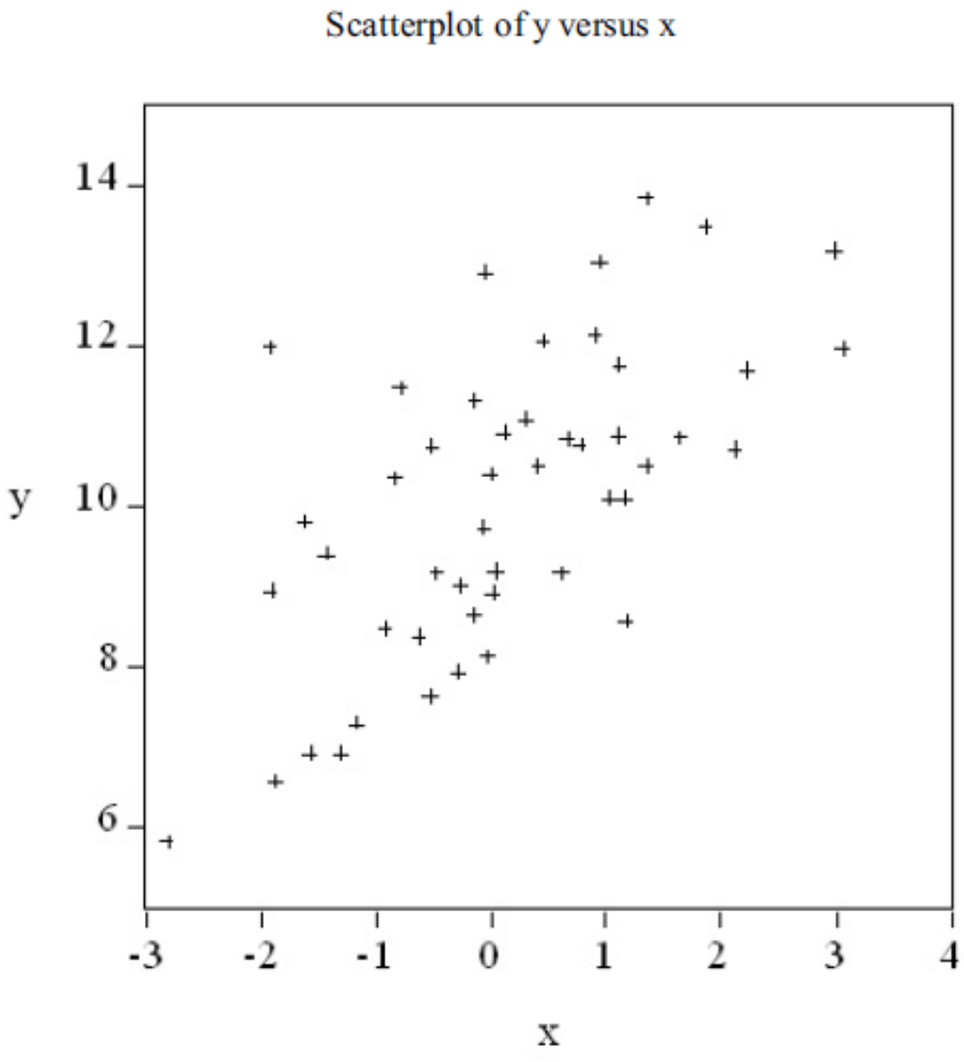
Regression

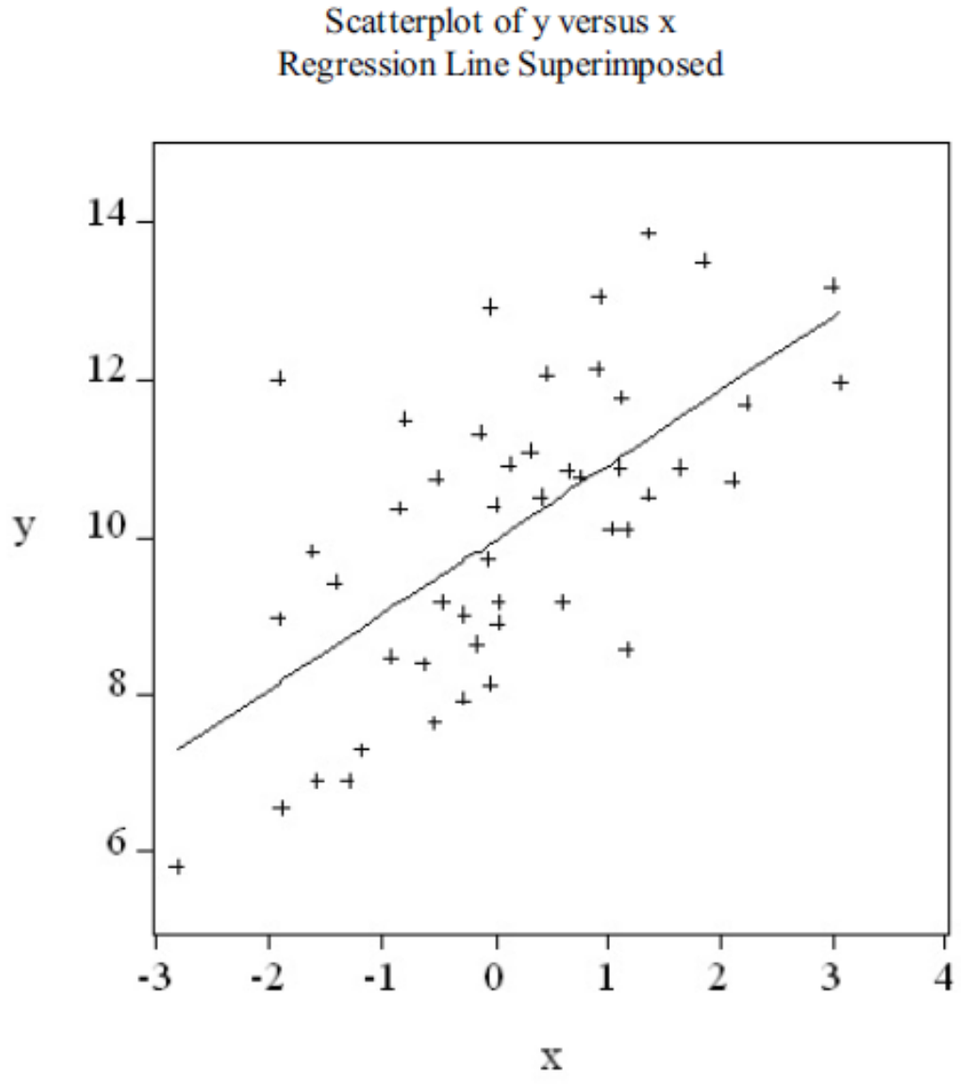
Overview

Regression Analysis

- One of the basic tools for forecasting
- A statistical technique to describe relationships among variables
- Consider two variables y and x
  - Describe y using x
  - y: dependent variable
  - x: independent variable (explanatory, exogenous)

Regression Analysis





Regression Analysis

- How to find the line that fits best?
  - Line: $y = \beta_0 + \beta_1 X$
  - How to find β_0 and β_1?
- Example

Regression Analysis

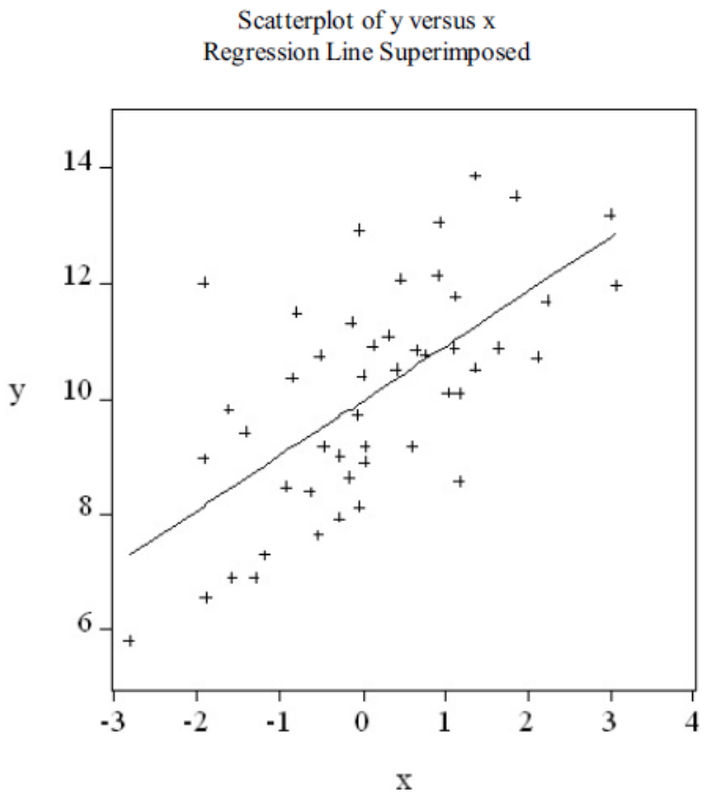
- Probabilistic Model
  - $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$
  - $\varepsilon_t \overset{iid}{\rightarrow} N(0, \sigma^2)$
  - Model parameters: $\beta_0, \beta_1, \sigma^2$
- If this model is correct:
  - Expected value of y conditional on $x = x^*$
    - $E(y|x^*) = \beta_0 + \beta_1 x^*$

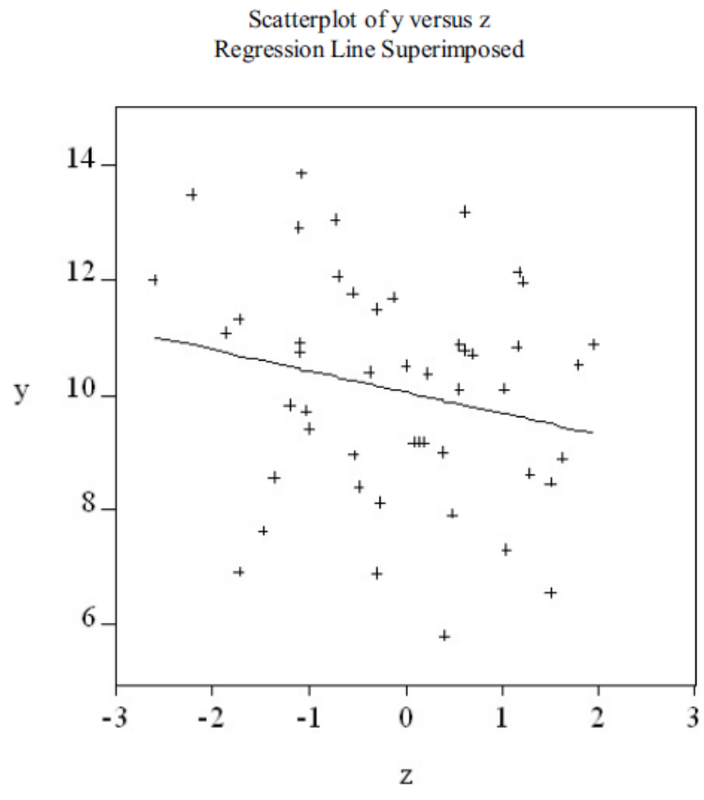
Regression Analysis

- Fitted values
  - $\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_t$
- Minimize residuals
  - Residuals = in-sample forecast errors
  - $e_t = y_t - \hat{y}_t$
- Least-squares estimation
  - $\sum_{t=1}^{T} (y_t - \hat{y}_t)^2$

Regression Analysis

- Multiple linear regression
  - y is described by more than one explanatory variable





Regression Analysis

- Multiple linear regression model
  - $y_t = \beta_0 + \beta_1 x_t + \beta_2 z_t + \varepsilon_t$
  - $\varepsilon_t \overset{iid}{\rightarrow} N(0, \sigma^2)$
- Fitted values
  - $\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_t + \hat{\beta}_2 z_t$
- Residuals
  - $e_t = y_t - \hat{y}_t$
- Least-squares estimation
  - $\sum_{t=1}^{T} (y_t - \hat{y}_t)^2$

Goodness-of-fit Statistics

- Sum squared resid. $\mathrm{SSR} = \sum_{t=1}^{T} e_t^2$

- Objective of the least-squares estimation
- Sum of squared residuals
- Not of much value in isolation
- Input to other diagnostics
- Useful for comparing models and testing hypotheses

Goodness-of-fit Statistics

- R-squared (R^2)
  - Indicates how much of var(y) can be explained by the variables included in the regression
  - Intuition: $\frac{var(y|x)}{var(y)}$
  - Measurement of in-sample success of the regression equation
  - If intercept is included, $0 < R^2 < 1$

$$R^2 = 1 - \frac{\frac{1}{T}\sum_{t=1}^{T} e_t^2}{\frac{1}{T}\sum_{t=1}^{T} (y_t - \bar{y}_t)^2}$$

Goodness-of-fit Statistics

- Adjusted R-squared ($\bar{R}^2$)
  - Same interpretation as R^2 but formula is slightly different
  - Adjusted for degrees of freedom used in fitting the model
  - Adjustment: penalize the amount of right-hand-side variables

$$\bar{R}^2 = 1 - \frac{\frac{1}{T-k}\sum_{t=1}^{T} e_t^2}{\frac{1}{T-1}\sum_{t=1}^{T} (y_t - \bar{y}_t)^2}$$

Goodness-of-fit Statistics

- Akaike info criterion (AIC)
  - Estimate of the out-of-sample forecast error variance
  - Similar to s^2 but penalizes degrees of freedom more harshly
  - Used to compare forecasting models

$$AIC = e^{\left(\frac{2k}{T}\right)} \frac{\sum_{t=1}^{T} e_t^2}{T}$$

Goodness-of-fit Statistics

- Schwarz criterion (SIC)
  - Alternative to AIC
  - Harsher penalty for degrees-of-freedom
  - Used to compare forecasting models

$$SIC = T^{\left(\frac{k}{T}\right)} \frac{\sum_{t=1}^{T} e_t^2}{T}$$

Goodness-of-fit Statistics

- Durbin-Watson stat. (DW)
  - Errors from a good forecasting model should be unforecastable
    - Forcastable error -> room for improvement in the model
    - Correlation among errors -> forecastable information
    - DW tests if the regression disturbances over time are serially correlated.

- $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$, where $\varepsilon_t = \varphi \varepsilon_{t-1} + v_t$

  $v_t \overset{iid}{\rightarrow} N(0, \sigma^2)$

  - ε_t is serially correlated when $\varphi \neq 0$.
  - Ideal case $\varphi = 0$

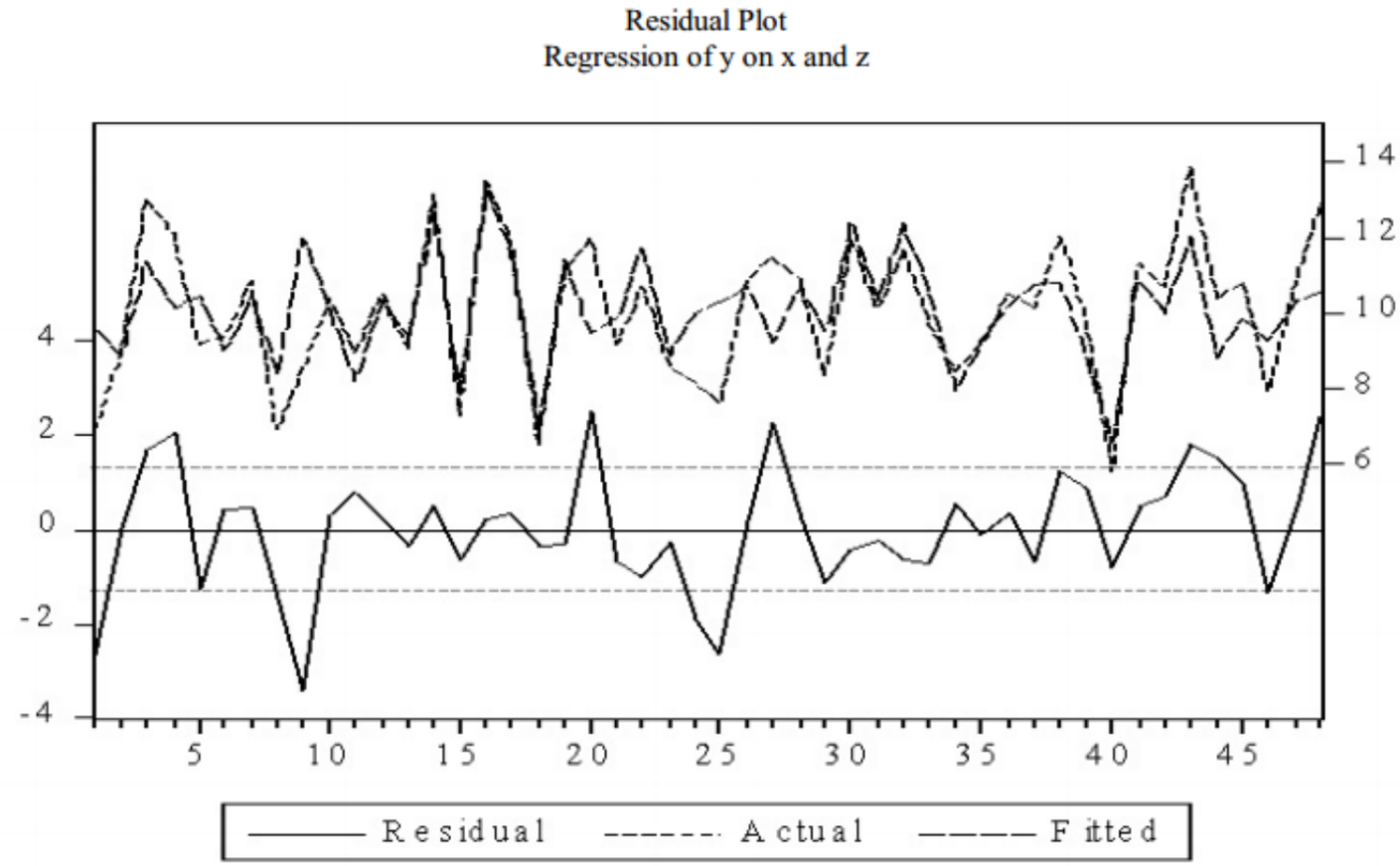
Goodness-of-fit Statistics

- Durbin-Watson stat. (DW)
  - $H_0: \varphi = 0$
  - $0 \leq DW \leq 4$
  - **OK:** $DW \sim 2$
  - **Alarm:** $DW < 1.5$
    - Consult DW tables for significance level for rejecting the null hypothesis

$$\mathbf{DW} = \frac{\sum_{t=2}^{T} (e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2}$$

Goodness-of-fit Statistics

- Residual plot
  - Examine :
    - Actual data (y_t)
    - Fitted values ($\hat{y}_t$)
    - Residuals (e_t)



Evaluation Metrics

Evaluation Metrics

- Up to now: measure a model‘s performance by some simple metric
  - classifier error rate, accuracy, …
- Simple example: accuracy

$$accuracy = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$

- Classification accuracy is popular, but usually **too simplistic** for applications of data mining to real business problems
- **Decompose** and count the different types of correct and incorrect decisions made by a classifier

Unequal costs and benefits

- How much do we care about the different **errors** and correct decisions?
  - Classification accuracy makes no distinction between **false positive** and **false negative** errors
  - In real-world applications, different kinds of errors lead to different consequences!
- Examples for medical diagnosis:
  - a patient has cancer (although he does not)
    → **false positive error**, expensive, but not life threatening
  - a patient has cancer, but she is told that she has not
    → **false negative error**, more serious
- Errors should be counted separately
  - Estimate cost or benefit of each decision

Confusion Matrix

- A confusion matrix for a problem involving n classes
  - is an $n \times n$ matrix with the columns labeled with actual classes and the rows labels with predicted classes

		Predicted Classes	
		p	**n**
True Values	**1**	True Positives (TP)	False Negative (FN)
	0	False Positives (FP)	True Negatives (TN)

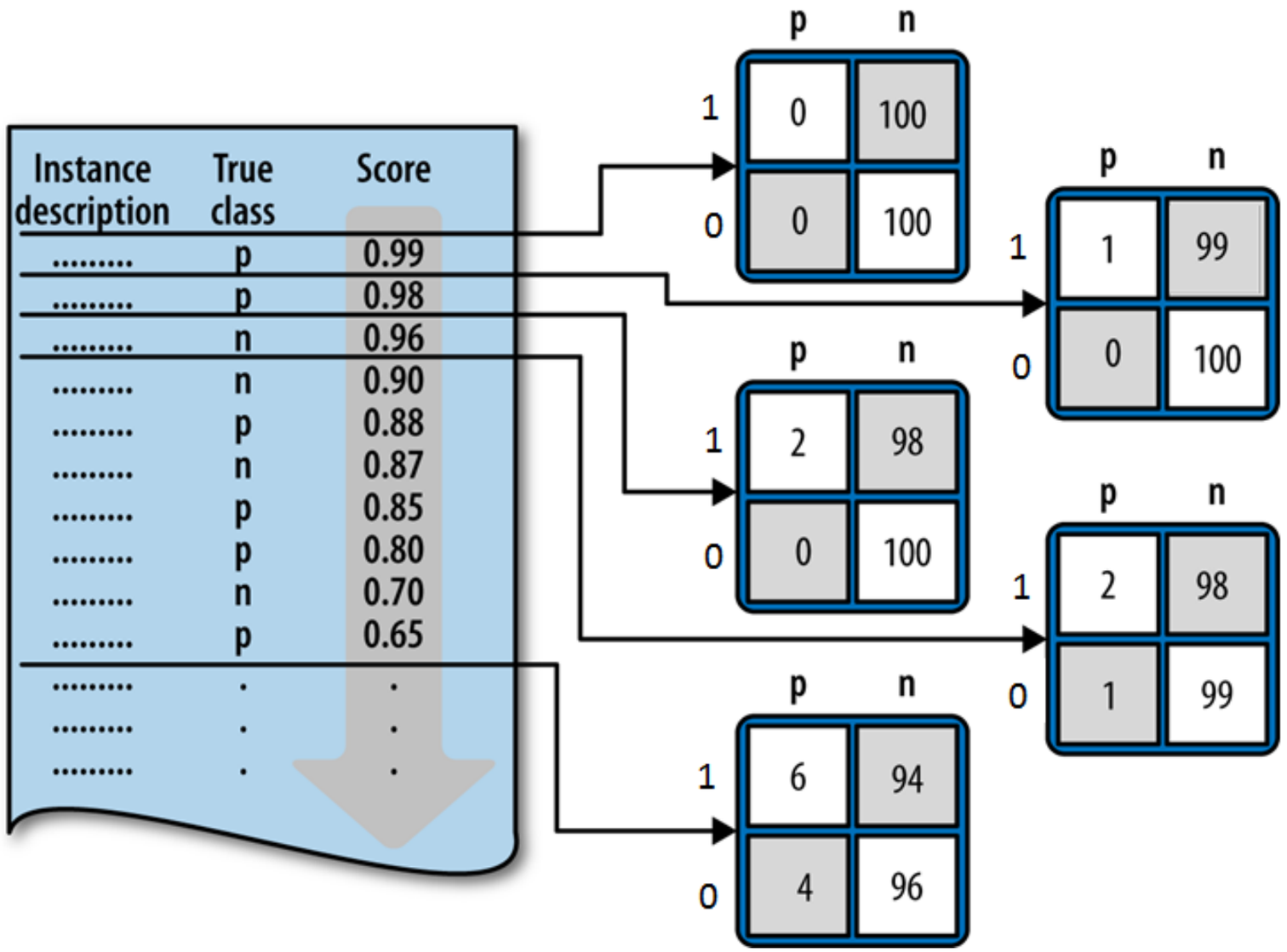
- Each example in a test set has an **actual class label** and the **class predicted** by the classifier
- The confusion matrix separates out the decisions made by the classifier
  - actual/true classes: **1** (Positive Label), **0** (Negative Label)
  - predicted classes: **p**(ositive), **n**(egative)
  - The main diagonal contains the count of correct decisions

Other Evaluation Metrics

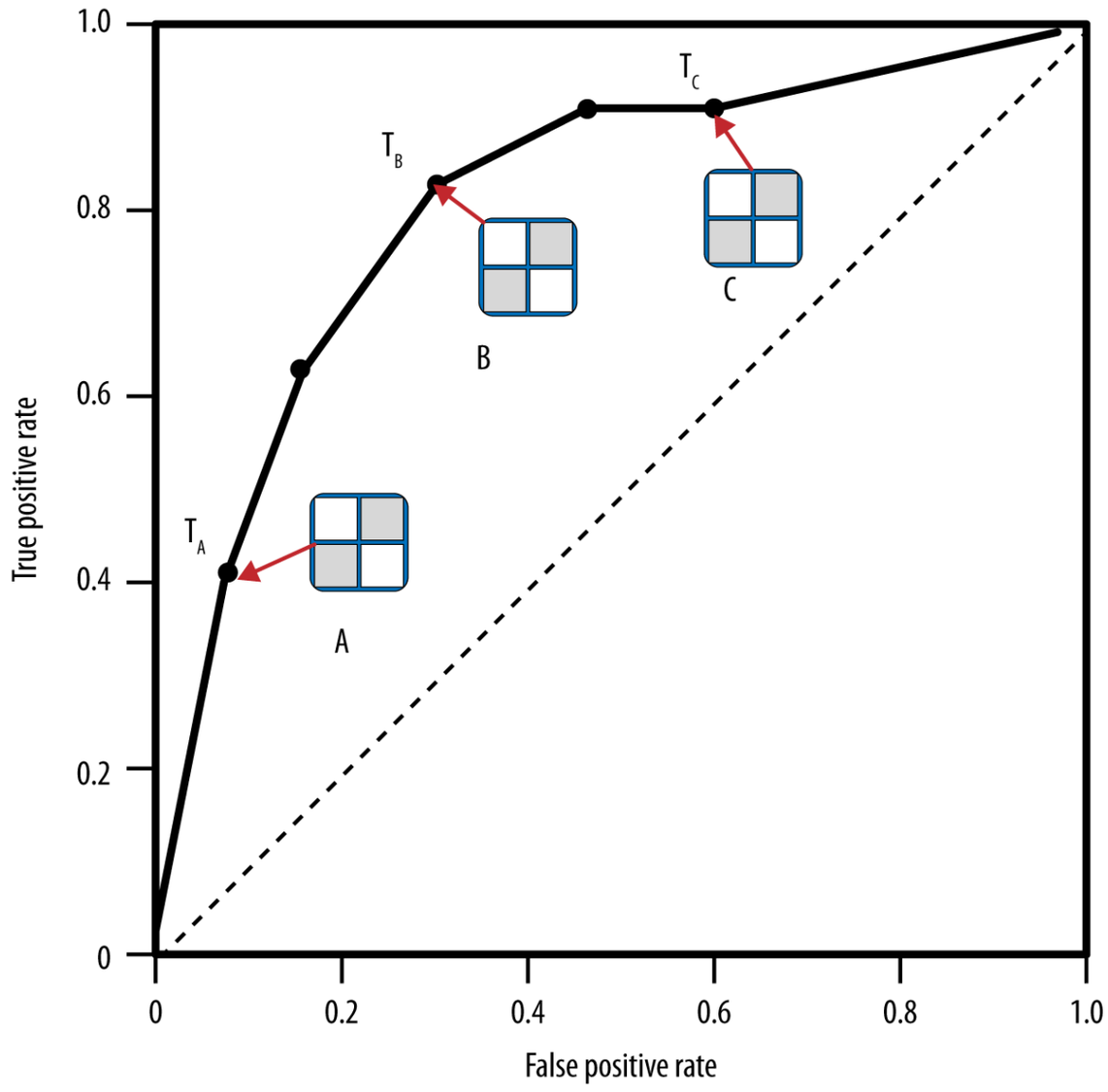
- Based on the entries of the confusion matrix, we can describe various evaluation metrics
  - True positive rate (Recall): $\frac{TP}{TP+FN}$
  - False negative rate: $\frac{FN}{TP+FN}$
  - Precision (accuracy over the cases predicted to be positive): $\frac{TP}{TP+FP}$
  - F-measure (harmonic mean): $2 \cdot \frac{precision \cdot recall}{precision+recall}$
  - Specificity: $\frac{TN}{TN+FP}$
  - Sensitivity: $\frac{TP}{TP+FN}$
  - Accuracy (count of correct decisions): $\frac{TP+TN}{P+N}$
  - False Positive Rate = 1 – Sensitivity

ROC Curve

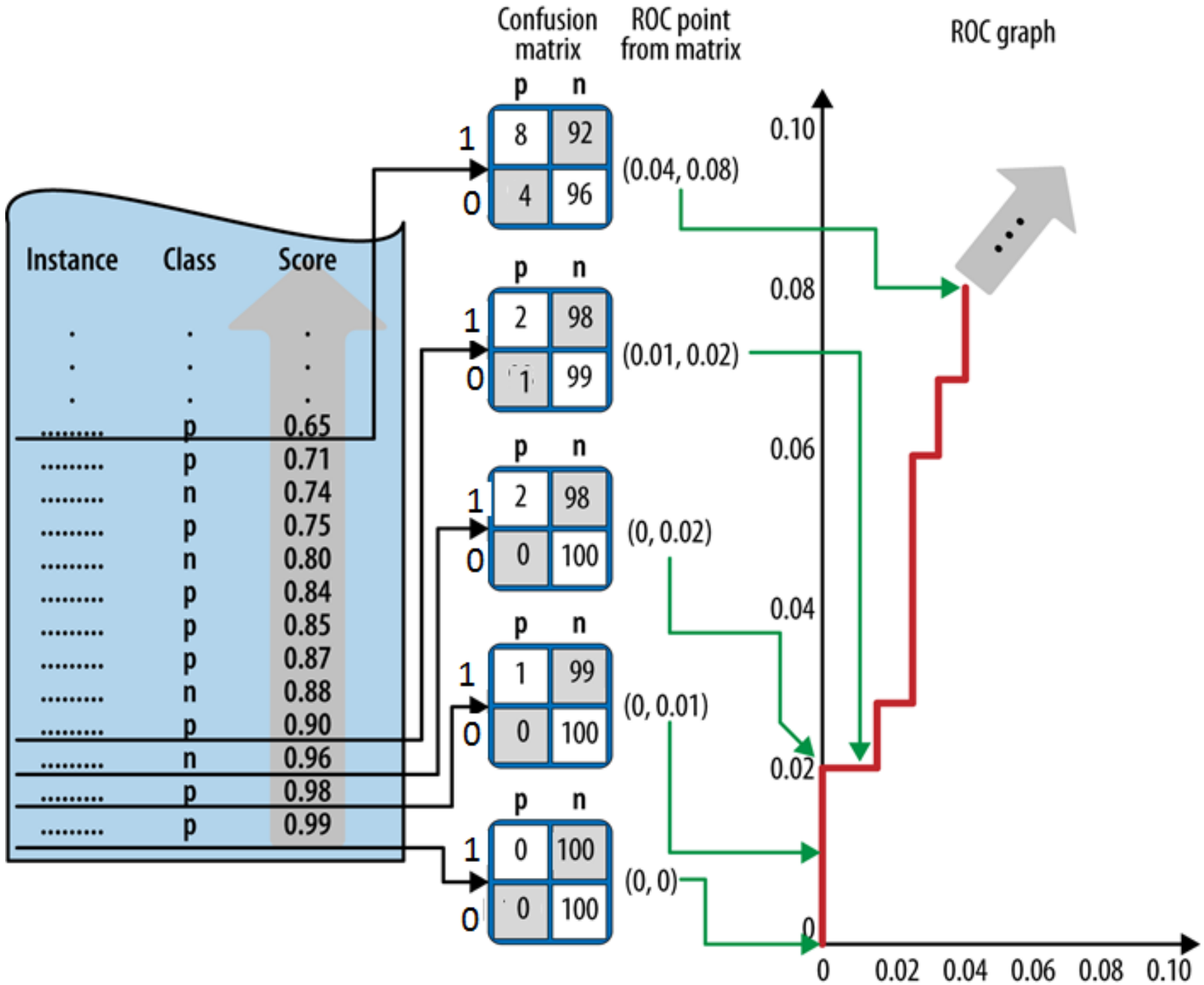
Ranking instead of classifying



ROC graphs and curves



ROC graphs and curves



Getting ROC curve: Algorithm

- Sort the test set by the model predictions
- Start with cutoff = max (prediction)
- Decrease cutoff, after each step count the number of true positives TP (positives with prediction above the cutoff) and false positives FP (negatives above the cutoff)
- Calculate TP rate (TP/P) and FP (FP/N) rate
- Plot current number of TP/P as a function of current FP/N

Area Under the ROC Curve (AUC)

- The area under a classifier's curve expressed as a fraction of the unit square
    - Its value ranges from zero to one
- The AUC is useful when a single number is needed to summarize performance, or when nothing is known about the operating conditions
    - A ROC curve provides more information than its area
- Equivalent to the Mann-Whitney-Wilcoxon measure
    - Also equivalent to the Gini Coefficient (with a minor algebraic transformation)
    - Both are equivalent to the probability that a randomly chosen positive instance will be ranked ahead of a randomly chosen negative instance

Performance evaluation

- Training Set:

Model	Accuracy
Classification Tree	95%
Logistic Regression	93%
k-Nearest Neighbors	100%
Naive Bayes	76%

- Test Set:

Model	Accuracy	AUC
Classification Tree	91.8% ± 0.0	0.614 ± 0.014
Logistic Regression	93.0% ± 0.1	0.574 ± 0.023
k-Nearest Neighbors	93.0% ± 0.0	0.537 ± 0.015
Naive Bayes	76.5% ± 0.6	0.632 ± 0.019

Performance evaluation

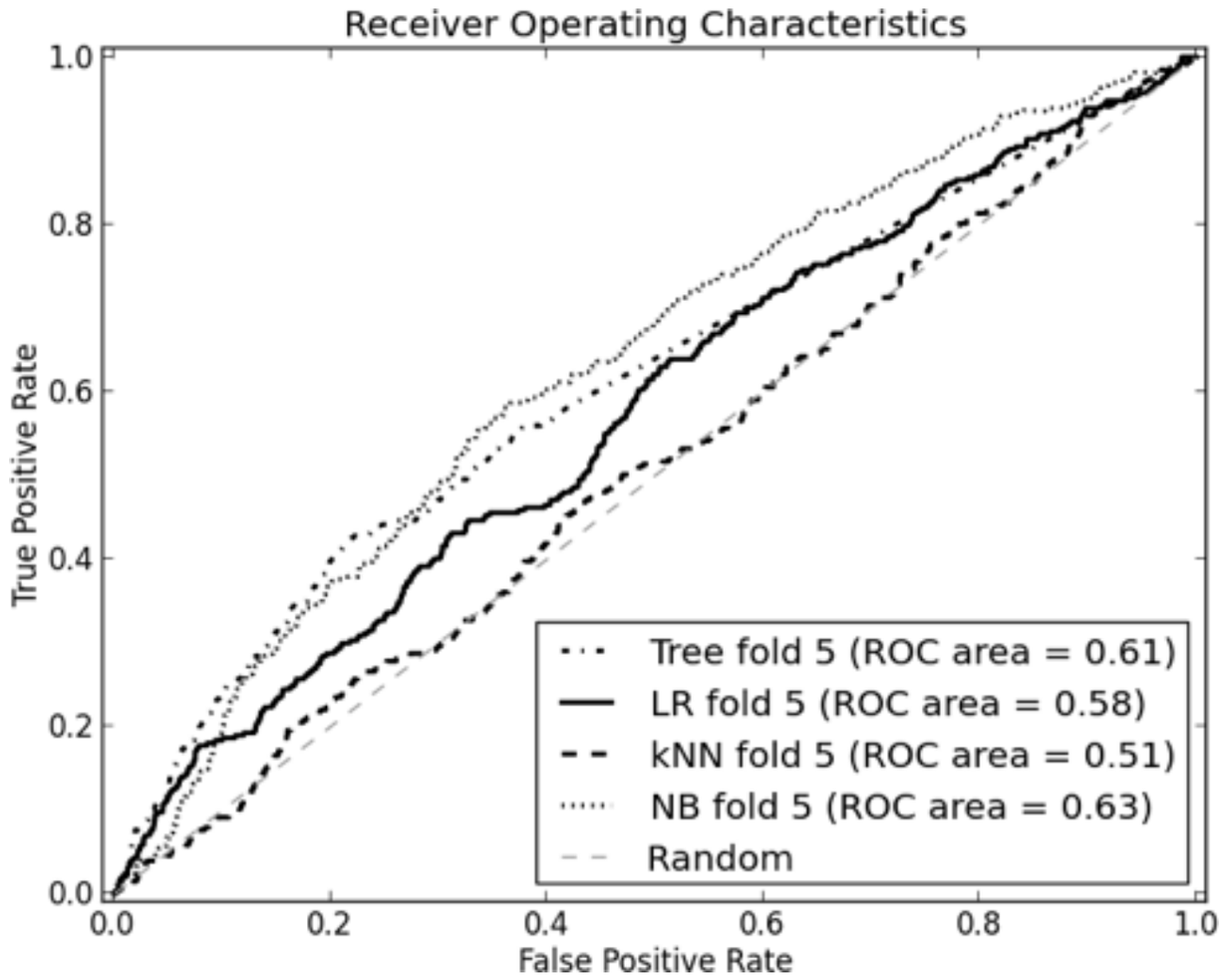
- Naive Bayes confusion matrix:

	p	n
1	127 (3%)	200 (4%)
0	848 (18%)	3518 (75%)

- k-Nearest Neighbors confusion matrix:

	p	n
1	3 (0%)	324 (7%)
0	15 (0%)	4351 (93%)

ROC Curve


Receiver Operating Characteristics
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Tree fold 5 (ROC area = 0.61)
LR fold 5 (ROC area = 0.58)
kNN fold 5 (ROC area = 0.51)
NB fold 5 (ROC area = 0.63)
Random