Methodology of data science

- **Start:** Real-world task or problem (business needs, societal or scientific problem)
  - E.g, investigate sales performance, transit access and equity, pollution impacts on society
- Develop research questions (RQs) and hypotheses
  - Can we predict seasonal trends in demands of perishable food to reduce waste & sales loss?
  - Is public transit access equitably allocated among high and low income areas of a city?
  - Does increased pollution lead to increased mortality rates?
- Collect relevant data to test the hypotheses and answer the RQs
  - Need to collect data or extract it from existing sources (Data Engineering, ETL)
  - Need data for RQ that is representative of downstream use (performance, fairness & bias)
- Clean the data!
- Perform exploratory analysis on the data (feature analysis and visualization)
  - Consider revisiting the data collection and cleaning process based on this data
- Evaluate the original hypotheses and RQs w.r.t. the data, iterate as needed
- **Finish:** Report back to stakeholders (decision-making)

Multiple comparisons problem:
The more comparisons made, the more correlations found. Counter w/ Bonferroni correction.

Case studies problem formulation
1. Dataset
2. Task
3. Problem
4. Cause

Feature analysis

Correlation: changes in one var correspond to changes in another.

Linear: $y = \alpha + \beta x$, $\beta \neq 0$. $y = \alpha + \beta x + \varepsilon$, ε is error term for relationship.
To test for lin. correl, use Pearson coeff $r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \in [-1,1]$

Marginalization: $P(A) = \sum_{b \in B} P(A, B=b)$, $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{\sum_{a \in A} P(B|a)P(a)}$

Independence: $P(x|y=y) = P(x)$. $P(x, y) = P(x) \cdot P(y)$

Information Theory: $P(x)$ encodes uncertainty about X.

Entropy: avg # of bits required to encode X.

$$H_p(x) = E\left[\log \frac{1}{P(x)}\right] = \sum_x P(x) \log \frac{1}{P(x)} = -\sum_x P(x) \log P(x)$$

↳ low bits + low entropy → low uncertainty.

Conditional entropy (CE): $H_p(x|y) = E\left[\log \frac{1}{P(x|y)}\right] = H_p(x, y) - H_p(y)$
Test how well $y=b$ predicts x.
$CE=0 \Rightarrow$ perfect pred.
$CE=1 \Rightarrow$ noise
$\uparrow MI + \uparrow PMI \Rightarrow$ independent

MI: $I(x, y) = \sum_{y \in y} \sum_{x \in x} P(x,y) \log\left(\frac{P(x,y)}{P(x)P(y)}\right)$

PMI: $PMI(x,y) = \log\left(\frac{P(x,y)}{P(x)P(y)}\right) = \log \frac{P(x|y)}{P(x)} = \log \frac{P(y|x)}{P(y)}$

discrete RVs useful for ranking features.

Power law: Discrete (Zipf) $P(f) = \alpha f^{-1-s}$
Continuous $P(X > x) \sim L(x) x^{-\alpha+1}$

→ Evaluate causes of multimodality.
→ Evaluate sufficient stats of data over time.

Data Cleaning

Issues without cleaning:
1. Missing data.
2. Entity resolution / Unnormalized cleaning.
3. Unit mismatch.

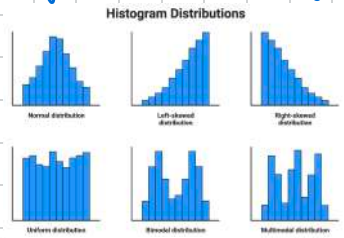
Perspectives on dirty data
1. Statistics view.
→ There is a process that produces data.
→ Any dataset is a sample of that process.
Good model > No model > Bad model
2. Database view.
→ Results are absolute (relational model).
→ Improve quality of results.
3. Domain expert view.
→ Data/answer doesn't look right
→ Figure out what happened.
4. Data scientist view: all above!

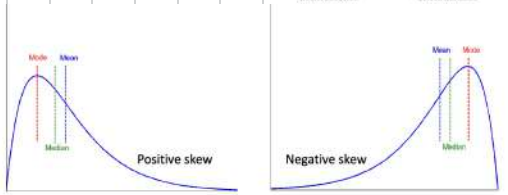
Data issues
1) Parsing text into fields (separator issues)
2) Naming conventions and entity resolution: NYC vs New York
3) Missing required field (birthdate)
4) Gaps in time series
5) Different representations ("2" vs 2), Unicode lookalikes
6) Fields too long (get truncated)
7) Mixed data types (feet, meters)
8) Redundant Records (exact match or other)
9) Formatting issues – especially dates
10) Licensing issues/privacy/cost prevent access to all data

Can be categorized as:
- Incomplete (lacking)
  - ↳ N/A when filled out
  - ↳ Changes in data collected
  - ↳ External issues
- Noisy (errors/outliers)
  - ↳ Faulty data collection
  - ↳ Conversion / typing errors
- Inconsistent (e.g. formats)
  - ↳ Changes in data collection ways
- Redundant (duplicated)
  - ↳ Human error / data integration

Data cleaning

1. Data summary
- Look at data types of columns.
  - ↳ String / categorical: frequency & # of unique vals.
  - ↳ Numeric: summary stats, plots.
    Histogram: empirical density.





2. Missing values.
- Missing completely at random (MCAR): no dependencies on other vars.
- Missing at random (MAR): data missing depends on another var but not latent var.
- Missing Not at Random (MNAR): data missing depends on latent var.

Imputation: makes assumptions & create missing values. Options:
- String / Categorical: mode.
- Integer: median, maybe mode.
- Continuous (float): mean/median.
- MNAR data: Appropriate defaults
- MCAR data: How do other cols impact prediction?
- MAR data: consider if missingness condition should influence prediction.

3. Noise, inconsistency, redundancy.
- Do reality checks!
- Identify outliers & potential errors
- Entity matching:
  Merging / Deduplication
  ↓
  Record / Entity linkage

Data transformation
- Smoothing: remove noise from data
- Aggregation: summarization
- Generalization: back off labels to a concept hierarchy
- Attribute/feature construction
  - New attributes derived from the given ones

Normalization:
- Min-max range: if natural range known.
- z-score: $v' = \frac{v-\mu}{\sigma}$
- Log normalization: positive skewness handle outliers. More general from Box-Cox power transforms.

Natural Language Processing

nltk: POS tagging, phrase chunking.

MIE223: Data Science.

Document → Sections & Paragraphs → Sentence (sentence segmentation/extraction) → Tokens
- Lemmas
- Morphological variants/stems
- POS tags → Phrase chunks (Keyphrases, Named entities)
- POS tags → Parse trees

Word tokenization

Text normalization – necessary.
a) Segment / Tokenize words in running text.
b) Normalize word format.
c) Segment sentences in running text.

Lemma: same step, similar meaning of word.
cat & cats = same lemma.

Wordform: full inflected surface form.
specific instance of a word.

Type: element of vocabulary. Vocab = {set of types}

Token: instance of that type in running text.

Morphology

Morphemes: meaningful units that make up words.
Stems: core meaning-bearing units.
Affixes: bits and pieces that adhere to stems.
] Morpheme segmentation

Word normalization & Stemming

Information retrieval: indexed text & query terms must have same form.
e.g. U.S.A. = USA
↓
Asymmetric expansion: less efficient but searches for lemmas.

Lemmatization: extract word meaning (rather than just stem).

Stemming: reduce terms to their stems. Crude chopping of affixes. e.g. Porter's algorithm.

Sentence segmentation

Options to segment sentences:
- Binary classifier. - end of sentence or not?
- Decision tree

Regular expressions

- Formal lang. for specifying text strings.
- Detecting word pattern variations.

Disjunctions: [wW]ood → wood, Wood
wood | Wood → wood, Wood
Ranges: [A-z] → upper case letter
[a-z] → lower case letter
[0-9] → single digit
Negations: [^Ss] → neither S nor s.
needs to come first in []

Good for generalizations, often the first model for any text processing task.

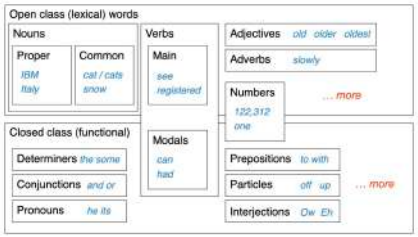
More Regex patterns

Pattern	Matches	
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 "Hello"
\.$	The end.
.$	The end? The end!

Find me all instances of the word "the" in a text.
the
Misses capitalized examples
[tT]he
Incorrectly returns other or theology
[^a-zA-Z][tT]he[^a-zA-Z]

Tokens: POS tagging



Classes

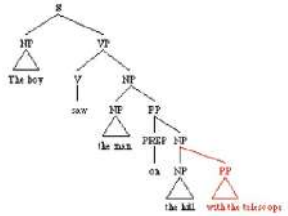
- Closed:
  - determiners: a, an, the, ...
  - pronouns
  - prepositions
- Open: nouns, verbs, adjectives, adverbs, ...

Phrase chunking and special noun phrases:

Noun phrases (NP) → verb phrases (VP)
Noun phrases (NP) → Proper noun phrases
Keyphrases e.g. machine learning, support vector machines.
Proper noun phrases → Named entity recognition (NER) e.g. people, places, organizations.

Parse trees: language structuring. Syntactic language analysis.

- Illustrate how words cluster into grammatical units.
- Good for interpreting, manipulating & generating language.



Semantic language analysis: coreference & entailment.

- Coreference: discourse (many sentences) use coreferring phrases.

Referent	Phrases
John	{John, He1, He2}
Integra	{a beautiful Acura Integra, it1, it2}
Bob	{Bob}
dealership	{the dealership}

← coreference resolution

- Entailment: highly contextual. →
  - **Question:** When did the Berlin wall open?
  - **Text contains:** The Berlin wall fell on November 9, 1989.
  - Simple entailment? Does "fall" → "open"?
    - A wall falling is a wall opening
    - A person falling is not a person opening

Sentiment analysis: detection of attitudes.

Scherer typology of affective states
- **Emotion**: brief organically synchronized ... evaluation of a major event
  - *angry, sad, joyful, fearful, ashamed, proud, elated*
- **Mood**: diffuse non-caused low-intensity long-duration change in subjective feeling
  - *cheerful, gloomy, irritable, listless, depressed, buoyant*
- **Interpersonal stances**: affective stance toward another person in a specific interaction
  - *friendly, flirtatious, distant, cold, warm, supportive, contemptuous*
- **Attitudes**: enduring, affectively colored beliefs, dispositions towards objects or persons
  - *liking, loving, hating, valuing, desiring* ← what most review sentiment focuses on.
- **Personality traits**: stable personality dispositions and typical behavior tendencies
  - *nervous, anxious, reckless, morose, hostile, jealous*

1. Holder (source) of attitude.
2. Target (aspect) of attitude.
3. Type of attitude:
   - from set of types: like, love, ...
   - or polarity → simpler.
4. Text containing the attitude.

Sentiment prediction
- Sentiment-aware tokenization.
- Prediction w/ classifiers/regressors.
  - Vader
  - don't capture everything.
  - Solution: noun + adj.

Sentiment lexicons: list of words associated w/ sentiment value.

1. General Inquirer: diff. categories. 2. LIWC: affective + cognitive processes. 3. MPQA Subjectivity Cues Lexicon: mood + intensity.

Automatic Polarity Analysis with PMI
→ How likely is each word to appear in each sentiment class? $P(w|c) = \frac{f(w,c)}{\sum_{w \in c} f(w,c)}$

Make them comparable between words:
Scaled likelihood: $\frac{P(w,c)}{P(w)}$

Potts experiment: logical negations.

Did we match it? / Should string be matched?

	Yes	No
Yes	TP	FP (Type I) ↑ accuracy
No	FN (Type II) ↑ recall	TN

↓ increase to reduce FN rate.

Time Series Analysis

Time series data: observations/measurements indexed according to time. S_t.

Goals of time series analysis

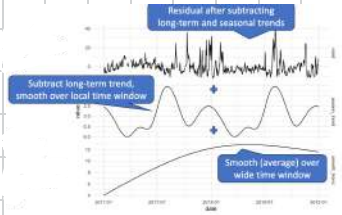
- Time scale analysis: at what scale is data useful to analyze?
- Smoothing: infer true state of past given a noisy signal. Use nearby values to smooth noise.
- Filtering: find true present state given a noisy signal. → Kalman filtering, true trajectory.

(Smoothing & Filtering: not covered)

- Forecasting: predict future values.
- Regression: given 2 time series, predict association between them. Often done w/ auto- & cross-correlation.

1. Time-scale analysis.

Tool: decompose time series into
1) Smooth long-term trend.
2) Seasonal variation.
3) Residual variation.



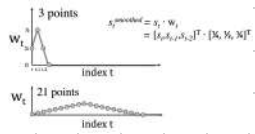
2. Smoothing.

- Given a noisily measured signal, can we reconstruct the true state of nature in past?

Smoothing methods/filters:

- Weighted: e.g. $\frac{1}{4} S_{t-1} + \frac{1}{2} S_t + \frac{1}{4} S_{t+1}$ (coeffs add up to 1) – Box smoothing.
- Triangle-weighted / Gaussian smoothing.
- Exponential smoothing

$$S_0^{smoothed} = S_0$$
$$S_t^{smoothed} = \alpha S_t + (1-\alpha) S_{t-1}, \quad t>0, \alpha \in (0,1].$$



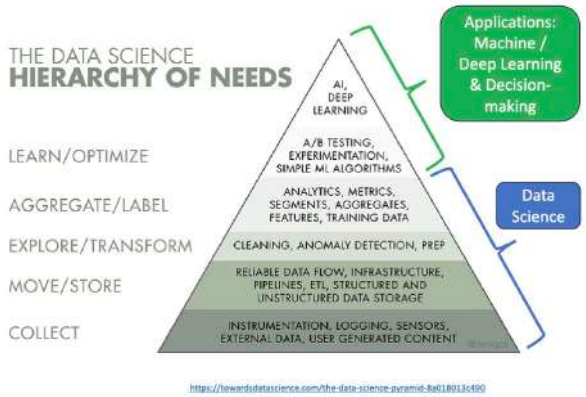
Non-causal: takes future values into consideration.
Causal: smoothed value doesn't depend on future.

Autocorrelation: compute Pearson coeff of a time series w/ itself moved by t positions.
Measure of correlation in time series at different lags.

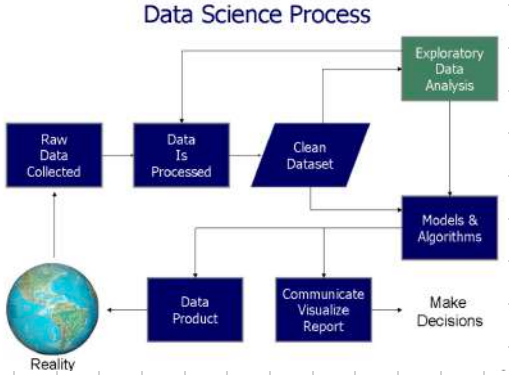
Cross-correlation: considers how 1 time series predicts another.

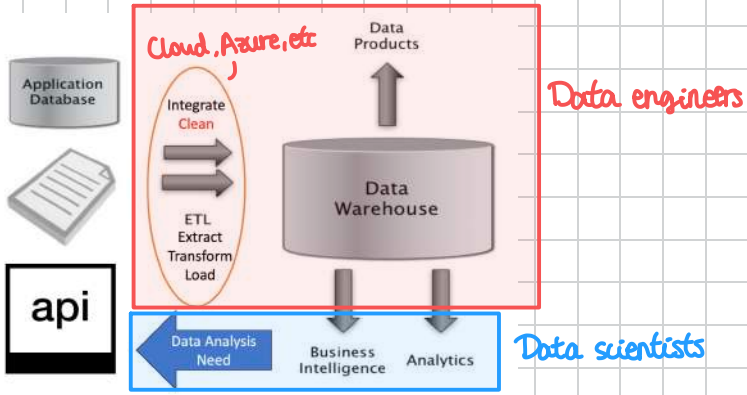
5. Time series regression.

- Covariance: $Cov_{x,y} = E[(x-E(x))(y-E(y))]$, 0 if x & y independent.
- Pearson correl. coeff.



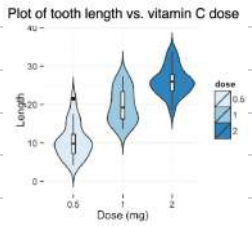
https://towardsdatascience.com/the-data-science-pyramid-8a018013c490





Visualization

1. Violin plot: full distributions side-by-side. Can show multimodality



2. Discrete heatmaps.
3. Scatterplots: don't need lin. relationships.
3. Heatmap: good when points in scatterplot dense & don't reflect density.
4. Bubble plot: 3rd data dimension – show w/ colour.
5. Choropleth: country data.

- Explore and visualize whenever possible
  - Tabular summaries: frequency, summary statistics
    - Much (human-generated) data is not Gaussian
    - Be cautious with summary statistics like mean, standard deviation
    - Median, quartiles, quantiles, full distribution much better (boxplot, histogram)
  - Dependency analysis: (pointwise) mutual information, CE, Chi2
  - Histogram, Box plot, Violin plot for univariate / bivariate data
  - Scatter plot, Heatmap / Density plot for bivariate data
  - Bubble plot for 3 or 4 dimensional data
  - Choropleth for geographical data
- Beware of bimodalities
  - Indicate important latent variables

Networks

Networks: defines interactions between system components.

Application domains in network analysis:
- Social
- Political/Organization
- Biology
- Information
- Computer
- Transportation

Social network problems

1. **Small World Phenomena**

Email experiment: 18 targets, 13 countries, 60000+ participants. Avg path length = 4.

Used for **link prediction**: "Given a snapshot of a social network, can we infer which new interactions among its members are likely to occur in near-future?"

→ Application domains: friend/product/page recommendation; predicting academic collab; predicting acquisitions.

→ Measure performance w/ held-out data, conversion rate.

2. **Tracking what goes viral.**

3. **Community detection** (interactions, profile, dynamics, etc)

Networks: **equivalent definitions** across domains.

Domain	Points	Lines
Math	Vertices	Edges, arcs
CS	Nodes	Links
Physics	Sites	Bonds
Sociology	Actors	Ties, relations

Graphs: Directed / Undirected

Edge attributes

- Weight
- Ranking
- Type
- Properties depending on rest of graph's structure, e.g. betweenness.

e.g. Twitter's elements:
- → User
- → Hashtag
- → Mention
- → Hyperlink

3 ways to represent networks:

1) Adjacency matrices.

from $\begin{bmatrix} a_{11} & \cdots & a_{1j} \\ \vdots & & \\ a_{i1} & & \end{bmatrix}$ to

$\sum$ across row for out-deg

$\sum$ across col for in-deg.

$a_{ij} = 1$ if $i \to j$ (row → col)

2) Edge list.

e.g. 2,3 / 2,4 / 3,2 / 4,5

3) Adjacency lists.

Good for large &/or sparse networks.

Quickly retrieve all neighbours.

1:
2: 3 4
3: 2 4
4: 5
5: 1 2

sparse matrix in Scipy (implementation exists)

Graph metrics

1. Path and distance.
2. In/Out degree.
3. Centrality.

Distances in networks

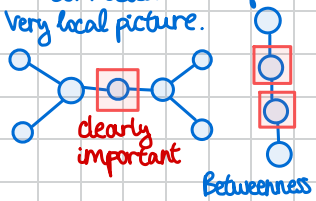
- Path: walk $(i_1, i_2, \dots i_k)$ where each node i_j is distinct.
- Cycle: walk where $i_1 = i_k$.
- Geodesic: shortest path between 2 nodes.

In/Out degree

- In-degree: # edges incident on node.
- Out-degree: # edges originating on node.
- Degree: # edges incident on node.

Why not degrees for centrality?

Degree only captures how connected you are, not how well-positioned. Not all connections are equal. Very local picture.



Center of the network

1. **Betweenness**: # short paths going through node.

$$C_B(i) = \sum_{j<k} g_{jk}(i) / g_{jk}$$

→ Finding key individuals who can control info flow.

g_{jk}: # shortest paths connecting jk.

$g_{jk}(i)$: # of those shortest paths that pass through i.

Usually normalized: $C_B'(i) = \frac{C_B(i)}{(n-1)(n-2)/2}$ (# pairs of vertices excl. the vertex itself)

Betweenness clustering: edges that connect communities.

a) Compute betweenness of all edges: all pairs is $O(n^3)$.

b) While (betweenness of any edge > threshold):
- → remove edge w/ highest betweenness.
- → recalculate betweenness.

2. **PageRank**: influential nodes.

Each page i is given a rank x_i.

Goal: assign x_i s.t. rank of each page governed by ranks of pages linking to it.

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} \cdot x_j$$

x_i: rank of page i.

B_i: every page j that links to i.

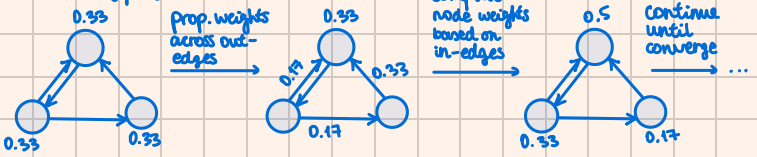
N_j: # links out of page j.

j: rank of page j.

Iterative PageRank:

1) Initialize all ranks to be equal: $x_i^{(0)} = \frac{1}{n}$

2) Iterate until convergence: $x_i^{(k+1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(k)}$



$$\begin{bmatrix} \text{PageRank}(p_1') \\ \vdots \\ \text{PageRank}(p_n') \end{bmatrix} = M \begin{bmatrix} \text{PageRank}(p_1) \\ \vdots \\ \text{PageRank}(p_n) \end{bmatrix}, \quad M(i,j) = \begin{cases} \frac{1}{n_j}, & \text{if page } j \to i \\ 0, & \text{otherwise} \end{cases}$$

Computes principal eigenvector via power iteration.

Shortest path: (V, E, s, t, c) — Directed graph; s: source, t: sink, $\in V$; cost of path: sum of arc costs in path.

Connected components

- Strongly connected: kind of like recurrent class in MCs.
- Weakly connected: every node can be reached from somewhere.
- Giant component: large fraction of graph.

Link prediction

Given $G[t_0, t_0']$ (a graph on edges up to t_0'), output a ranked list L of links (not in $G[t_0, t_0']$) predicted to appear in $G[t_1, t_1']$.

Evaluation

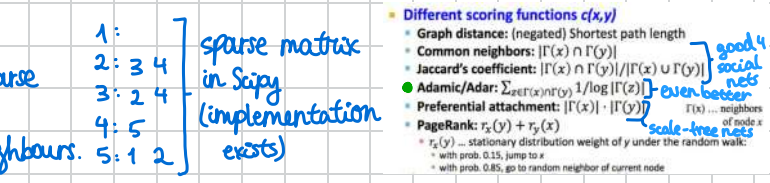
- $n = |E_{new}|$: # edges that appear during test period $[t_1, t_1']$.
- Take top n elements of L, count correct edges.

Procedure for link prediction via proximity

→ For sparse networks, consider only nodes w/ in/out degrees above certain threshold. (e.g. 3)

1. For each pair of nodes (x, y), compute c(x, y). e.g. # common neighbours c(x, y) of x & y.
2. Sort pairs by decreasing score c(x, y).
3. Predict top n pairs as new links.
4. See which links actually appear in $G[t_1, t_1']$.

Performance score: frac. of new edges guessed correctly.



Knowledge graphs and RDF

Knowledge graphs: "triple" data on the web.
- distributed (across the web), dynamic (changes often).

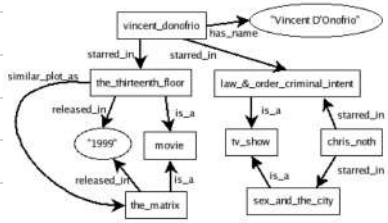
Resource Description Framework (RDF): standard for data interchange on the semantic web. Making statements (triples). Enables inference, querying, and integration across domains.

triple: (resource, predicate, value). Some fact / assertion.

Paris – is capital of – France.
Entity – Relationship (edge) – Object (another node)

Resource: anything that can be identified.
Predicate: property name that has a URI.
- **URI**: uniform resource identifier.
- **URL**: uniform resource locator.

Value: another resource / literal.



Entity resolution

- Matching words / cooccurrence don't always indicate similarity.
- Knowledge graphs resolve entities by linking related concepts through structured knowledge.

Noun phrases help w/ context to figure out the real meaning.
e.g. treating "apple" and "juice" in "apple juice" as an NP helps w/ meaning. → basically semantic anchors.

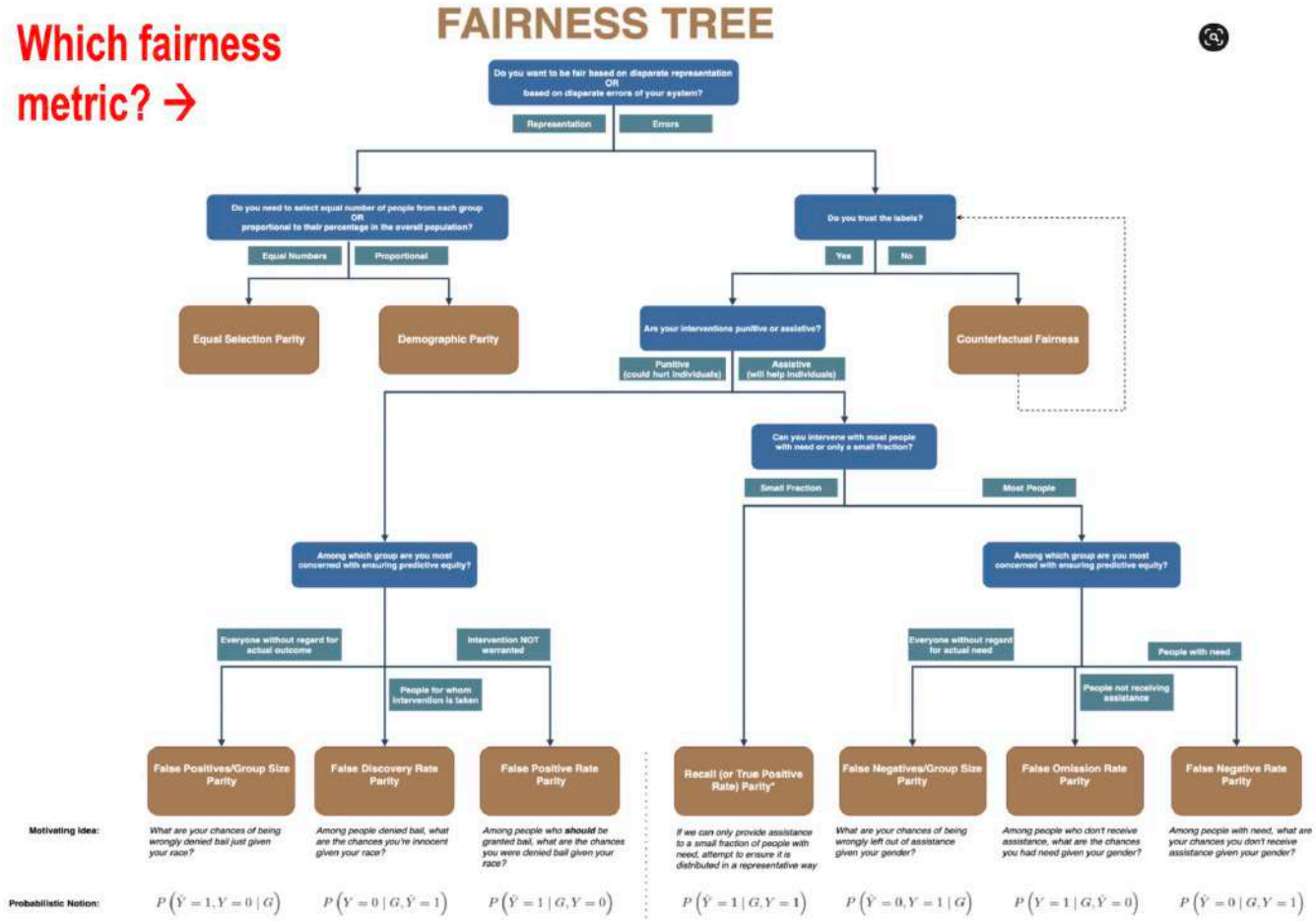
Named Entity Recognition (NER) tags named entities.

Why express world knowledge in structured format?

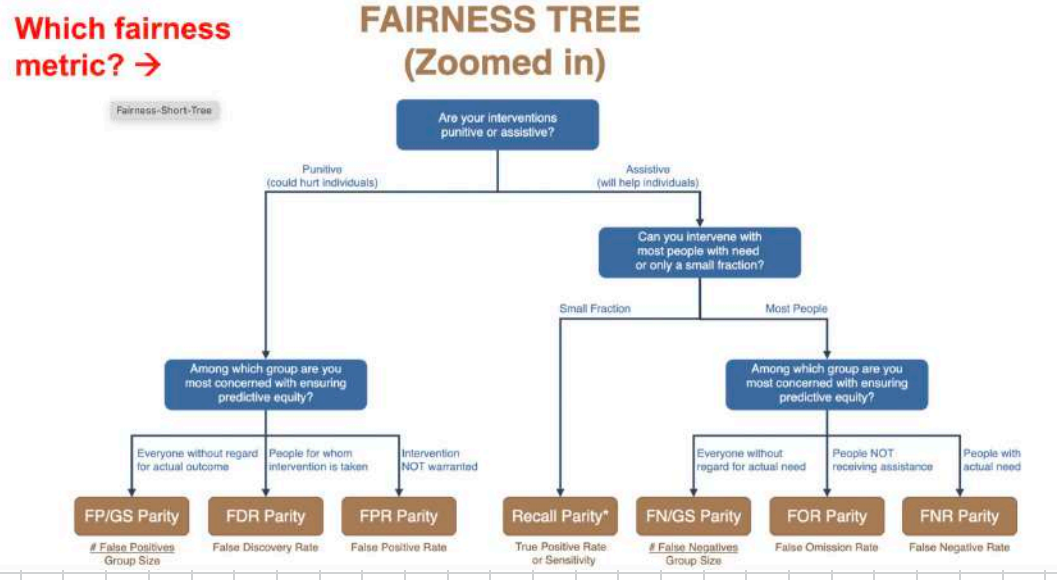
- Dynamic source of structured data.
  - Query with SPARQL. Inherently linked.
- Answering questions.
- Entity resolution (e.g. which Michael Jordan are you asking about?)

Some important figures

Which fairness metric? →



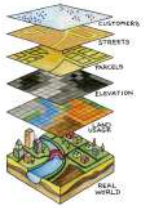
Which fairness metric? →



Geographic Information Systems (GIS)

Data model: objects in a spatial database. Provides a formal means of representing & manipulating spatially-referenced information.

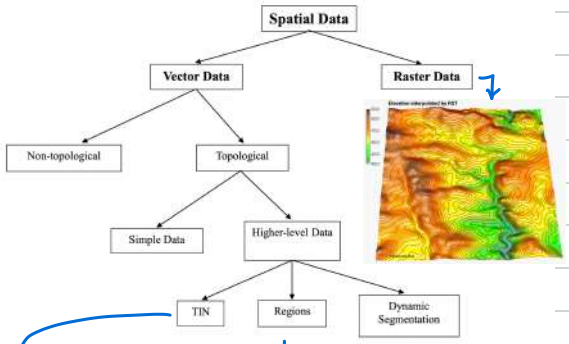
Coordinates: define locations & extent of spatial objects. Objects described by attribute data.



Thematic layers: logical separation of data according to theme. each layer is a characteristic.

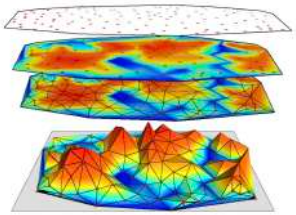
2 common spatial data models:
- Vector
- Raster

Common GIS Data Models



TIN: Triangulated Irregular Network.

↳ Shapefiles.



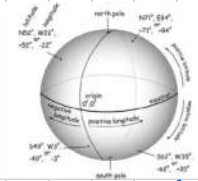
- good for terrain models.

Vector vs. Raster data

Vector: discrete features.
Raster: continuous features.

Spatial operations can only be performed on 1 type of layer.

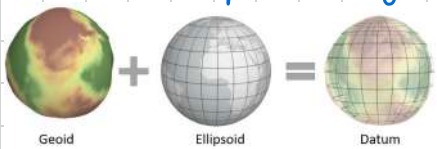
Coordinate data



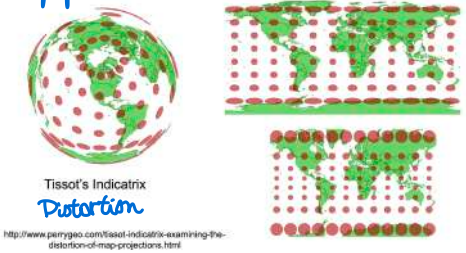
Origin: intersection of Equator and Greenwich meridian.

Spherical coords:
- Deg, min, sec (DMS).
- Decimal degrees (DD).

→ Earth is an oblate spheroid. Density anomalies, etc.



Tissot indicatrix: Examines distortion of map projections.



Great Circle: any line resulting from a plane passing through the center of the globe.

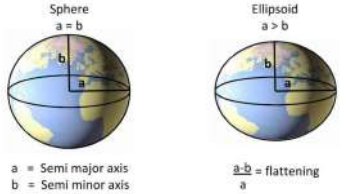
Converting arc to surface distances

- At equator 1° of longitude ≈ 111 km.

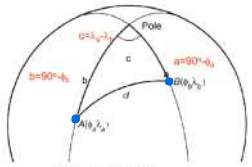
$d = r\theta$: ground distance

r: radius of sphere.

θ: radians.



Distance between 2 points on sphere



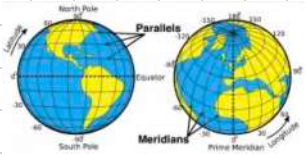
$$d = r\cos^{-1}[\sin\phi_a \sin\phi_b + \cos\phi_a \cos\phi_b \cos(\lambda_a - \lambda_b)]$$

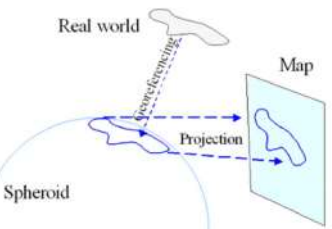
arc length — φ: latitude. λ: longitude.

Geographic vs. Projected Coordinate System

- **Geographic Coordinate System (GCS)**: spheroidal approx. of Earth's surface.
- **Projected Coordinate System (PCS)**: proj of GCS onto rectangular (Euclidean) map coordinates for viewing.

GCS nomenclature

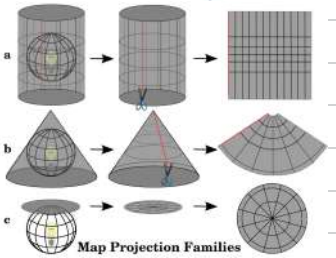




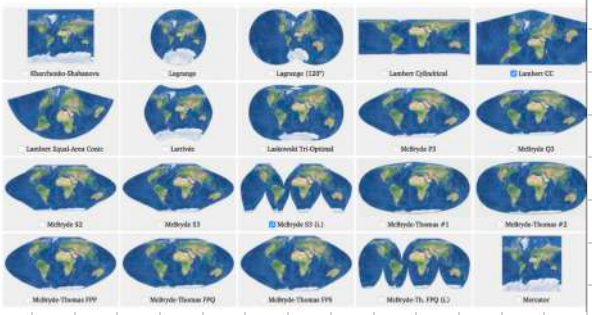
3 types of projected coordinate systems:

1. Cylindrical.
2. Conical.
3. Planar.

→ maintain different spatial properties.



Projected Coordinate Systems



Spatial properties maintained:
- NESW cardinal directions
- Distances
- Angles
- Surface area of diff. regions.

Geographic to Cartesian coords

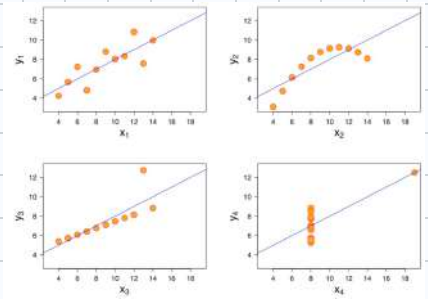
$$(x, y, z) = (R\cos\varphi\cos\lambda, R\cos\varphi\sin\lambda, R\sin\varphi)$$

Advanced Data Science

Visualization is very important

Ex.: Anscombe Quartet: 4 datasets that have the same descriptive stats.

Property	Value	Accuracy
Mean of x	9	exact
Sample variance of x	11	exact
Mean of y	7.50	to 2 decimal places
Sample variance of y	4.125	plus/minus 0.003
Correlation between x and y	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively



Confounding variables

- Latent variables: not present in data. Averaging over them can hide trends.

To find confounders, search.

Ways to deal w/ latent confounders:

- Mixture models (EM: Expectation maximization algorithm).
- Deep learning.

Normalization introduces spurious correlations

Ex.: $x_i, y_i \sim N(10,1)$; $z_i \sim N(30,9)$.

Correl coeff $\rho = 0.53$ (high) even though x, y, z all independent.

→ Ratios induce correlations. (Closure problem)

→ e.g. if x, y, z must all sum to 1, we induce a negative correlation on them.

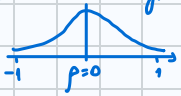
Compositional Data Analysis (CoDA) solves the issue: it normalizes by common divisors. Works w/ log ratios between components.

Check for data compositionality:

- Are vars part of a constrained total?
- Does increasing 1 val lead to decrease in another?
- Would it make sense to look at ratios?

Reason: correlation coefficient ρ when $\rho = 0$ is actually drawn from a symmetric distribution.

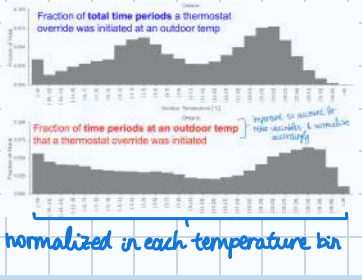
Esp. wider when small sample size.



The more vars, the more spurious correlations.

Correct Normalization

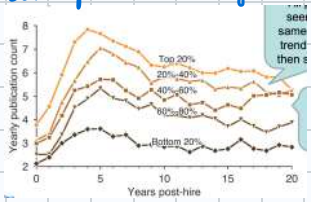
- You need to choose the right denom.



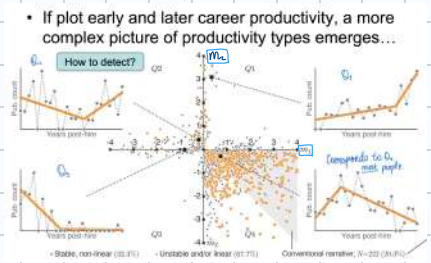
normalized in each temperature bin

Myth of the Average

Ex.: Productivity of publications. Same picture emerges:



More nuanced look reveals this:



Scientists used this procedure:

1) Brute force to find year split.
2) Fit lin. regs to find m_1 and m_2.

Synthetic Data for Debugging

- Your code will have bugs!
- How to debug data analysis / machine learning?
  - **Make a synthetic dataset**
    - E.g., (Features **x**: Age, Gender; Label **y**: {Snapchat, Facebook})
      - Everyone under 20 uses Snapchat
      - Everyone 20 and over uses Facebook
      - Random selection for gender

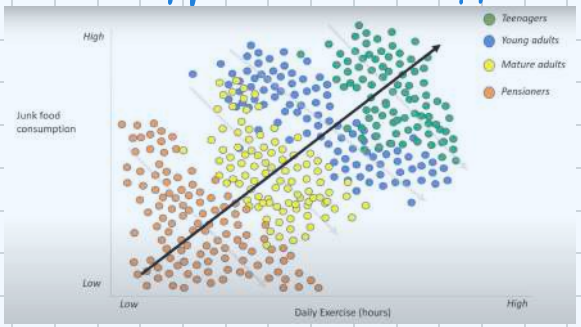
Or use label as a feature!

  - **Does your analysis uncover the expected trends?**
  - **What happens as you add noise?**
    - 90%, 80%, 70% of people under 20 use Snapchat?

Inability to uncover expected trends indicates a bug in your code or the need to revisit your analysis approach or hypothesis space.

Simpson's paradox

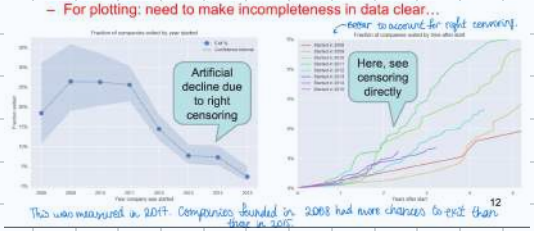
- Appears if data not randomly sampled.
- Stat. phenomenon where an association between 2 vars in a population emerges, disappears, or reverses when population divided into subpopulations.



More in dedicated notes.

Plotting "right-censored" time series data

- "Right censored": event may not have happened in all samples. Prominent in "survival/failure analysis".



↑ Not a bias in data but creates a bias in interpretation when subgroup dynamics are ignored.

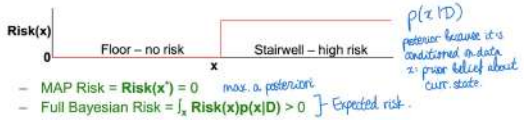
Bayesian Data Analysis

2 approaches to probability

1. Frequentist: single point parameter estimates.
2. Bayesian: belief distribution over parameters.

- **Robot has belief P(x|D) over position**
  - D consists of noisy range finder readings

- **Associate Risk(x) w\ position x (e.g., stairs!)**



  - MAP Risk = **Risk(x*) = 0** max. a posteriori
  - Full Bayesian Risk = $\int_x$ **Risk(x)p(x|D) > 0** } Expected risk.

Bayesian generative view: there's a hidden (latent) process that generates data.

Bayesian perspective: places weights on hypotheses.

→ The more data we see, the more peaked our belief is in correct value of p.

Latent data generating process

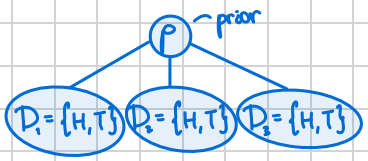
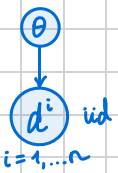


Plate notation



D: data. Parameters determined by model.

Bayes Theorem for updating beliefs:

$$P(W|D) = \frac{P(W) \cdot P(D|W)}{P(D)}$$

P(W|D): posterior probability of weight vector W given training data D.

P(W): prior probability of weight vector W.

P(D|W): likelihood probability of observed data given W.

P(D): normalizing constant

$P(D) = \int P(W)P(D|W)\,dW$

→ use mcmc and variational inference to find P(D).

Joint probability:

$P(W,D) = P(W) \cdot P(D|W) = P(D) \cdot P(W|D)$

All NNs are just maximizing likelihood.

→ As # of trials $n \to \infty$, we converge to the Dirac-Delta function. It's a "consistent" estimator.

↳ Asymptotic behaviour of posterior.

Bayesian data analysis good when data sparse/small.

Bayesian Graphical Model Notation

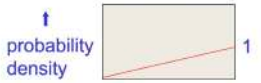
- Recursive Bayesian update
- $P(\vec{\theta}|D^n) \propto P(d^n|\vec{\theta})P(\vec{\theta}|D^{n-1})$,
  where $D^n = \{d^1, \ldots d^n\}$
  → Update posterior after every data point.
- $P(\vec{\theta}|D^n) \propto P(\vec{\theta}, D^n) = P(\theta)\prod_{i=1}^{n} P(d^i|\vec{\theta})$

Using a distribution over parameter values

- Start with a **prior** distribution over p. In this case we used a uniform distribution. Basically, if you know nothing

$P(p) = \begin{cases} 1 & 0 \le p \le 1 \\ 0 & \text{else} \end{cases}$ $P(D_i = \text{head}|p) = p$

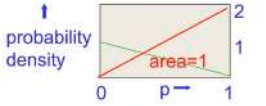
- Multiply the prior probability of each parameter value by the probability of observing a head given that value (**likelihood**).



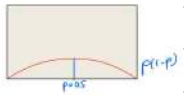
- Then scale up all of the probability densities so that their integral comes to 1. This gives the **posterior** distribution.

Lets do it again: Suppose we get a tail

- Start with a prior distribution over p.



- Multiply the prior probability of each parameter value by the probability of observing a tail given that value.

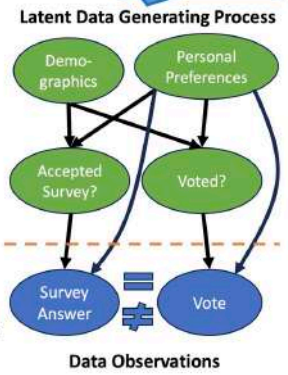


- Then renormalize to get the posterior distribution. Look how sensible it is!

A More Complex Generative Process

Inference is difficult, need MCMC inference tools like PyMC!

- Data are just the observables of a generative process
  - We observe election **votes** of a population
  - We observe Gallup poll **survey** of a population's voting disposition
  - But what process generated both?
- Example: 2016 US Presidential Election
  - Gallup Poll Surveys predicted a landslide for Hillary Clinton
  - We know how the election turned out
  - Why was there a discrepancy?
    - Sample bias in surveys
    - Human behavioural biases in revealing their true preferences
- We care about the (latent) variables that generate the data
  - We can use understanding of this generative process for prediction
  - Aside: critically connected to modern perspective of Generative AI



The Bayesian framework

- The Bayesian framework assumes that we always have a prior distribution for everything.
  - The prior may be very vague.
  - When we see some data, we combine our prior distribution with a likelihood term to get a posterior distribution.
  - The likelihood term takes into account how probable the observed data is given the parameters of the model.
    - It favors parameter settings that make the data likely.
    - It fights the prior
    - With enough data the likelihood terms always wins.

Privacy and Anonymity

Consent: GDPR (EU), US Privacy Act of 1974.

Major providers of user web data: phone providers, weather apps.

Deidentification: only release attributes that couldn't identify you.

Re-identification: multiple attributes can reidentify you.

Privacy

1) Anonymize the data. Remove personally identifying information (PII). (Dictated by HIPAA)
   → Quasi-identifiers can still be used for linking anonymized data w/ other datasets. Birth date, gender, ZIP, etc.

Data attributes

Key Attribute (often PII)	Quasi-identifier (Could be combined to identify someone)			Sensitive attribute
Name	DOB	Gender	Zipcode	Disease
Andre	1/21/76	Male	53715	Heart Disease
Beth	4/13/86	Female	53715	Hepatitis
Carol	2/28/76	Male	53703	Brochitis
Dan	1/21/76	Male	53703	Broken Arm
Ellen	4/13/86	Female	53706	Flu
Eric	2/28/76	Female	53706	Hang Nail

→ More in Extra Notes.

AOL privacy debacle: anonymized search query logs released, someone gets doxxed.

k-Anonymity

Info for each person contained in the released table can't be distinguished from at least k-1 individuals whose information also appears in the release.

→ You could replace quasi-identifiers with less specific but semantically consistent value.

→ Why not drop rows? Creates missingness-not-at-random, changes to distribution.

Achieving k-anonymity

- **Generalization**: replace specific quasi-identifiers w/ less specific values until k identical values.
  - ↳ Can cause suppression: too much info loss. Common w/ outliers.
  - ↳ Techniques: range, remove last few zipcode digits.
  - ↳ Relies on assumption of locality of quasi-identifiers. Real-world is sparse. Curse of Dimensionality.

Fairness and Bias

→ Fair treatment = Parity.

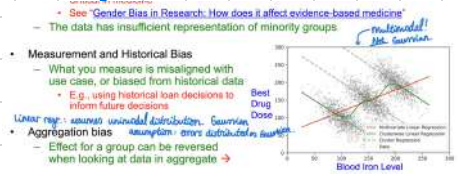
- Statistical biases make ML possible but not all biases are good.

Multimodality $\overset{\text{often}}{\Longrightarrow}$ confounding / latent vars.

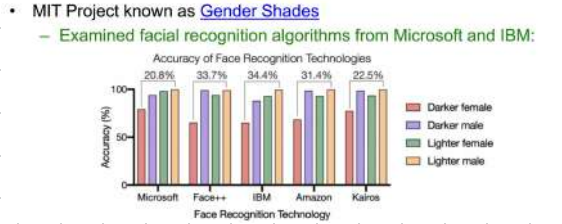
Biases:

- **Representation & Collection Bias**: Data not representative of population distribution it's meant to serve. Usually minorities. Critical in medicine.
- **Measurement & Historical Bias**: Data measured misaligned w/ use case or biased from historical data.
- **Aggregation bias**: effect for a group can be reversed when looking at data in aggregate.

  Lin. reg.: assumes that errors are Gaussian distributed.



MIT study on racial/gender discrimination in image classifiers.



What types of parity in *binary classification* can we aim for?

Contingency table (for binary classification) ← include this in final note.

		True condition: Condition positive	True condition: Condition negative		
Predicted condition	Predicted condition positive	True positive, Power	False positive, Type I error	Positive predictive value (PPV), Precision = Σ True positive / Σ Predicted condition positive	False discovery rate (FDR) = Σ False positive / Σ Predicted condition positive
	Predicted condition negative	False negative, Type II error	True negative	False omission rate (FOR) = Σ False negative / Σ Predicted condition negative	Negative predictive value (NPV) = Σ True negative / Σ Predicted condition negative
		True positive rate (TPR), Recall, Sensitivity, probability of detection = Σ True positive / Σ Condition positive	False positive rate (FPR), Fall-out, probability of false alarm = Σ False positive / Σ Condition negative	Positive likelihood ratio (LR+) = TPR / FPR	Diagnostic odds ratio (DOR) = LR+ / LR−
		False negative rate (FNR), Miss rate = Σ False negative / Σ Condition positive	Specificity (SPC), Selectivity, True negative rate (TNR) = Σ True negative / Σ Condition negative	Negative likelihood ratio (LR−) = FNR / TNR	F_1 score = $\frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$

Precision = $\frac{TP}{TP+FP}$ FDR = $\frac{FP}{TP+FP}$ FPR = $\frac{FP}{FP+TN}$ Confusion matrix (for multiclass classification)

False Discovery Rate is a good metric for parole decisions.

→ Can't balance all metrics if prevalence between diff. groups is actually different.

2 types of parity

usually determined by demographic attribute.

1) **Group fairness**: achieve parity across protected & advantaged groups. → Intersectional bias.
2) **Individual fairness**: similar individuals should have sim. predictions / outcomes. → Enforce this basically
   ↳ Requires measure of similarity between demographic feature vectors.

→ Complete equality (all parities) can't be achieved. We have to prioritize some metrics.

Fairness in treatment: don't consider sensitive attributes.

Fairness in impact: classification outcome should be balanced across group. e.g. same FP rate.

Parities to aim for:

- **Demographic / statistical parity**: Predicted outcome only.
  ↳ e.g. same pos. classification rates.
- **Parity in errors**: diff. groups have diff. classification rates but we constrain error rates.

Use cases of data science

- **Forecasting**
  - Inventory planning
  - Financial budgeting
  - Workforce management
  - Waste management
- **Personalization**
  - ML-powered recommendation systems
    - ↳ product-listing
    - ↳ item cross-selling
- **Product Attribute Extraction**
  - ↳ richer product descriptions
  - ↳ optimized management of item category.

Week in the life of a data scientist – Lifecycle

1) Define problem statement: meet w/ business stakeholders, create rough timeline w/ a simple deliverable & follow-up meeting.
   - → Business requirement: workforce management.
   - → Alignment: time series model that forecasts for orders/day at store level.
2) Data pipeline
   - → Breaking down query into smaller parts.
3) EDA
   - → Explore feasibility of problem statement. Understand the data.

   } 70/80%
4) Modelling & Experimentation
   - → Experiments to compare models
   - → A business wants an MVP.

   } 20-30% of time
5) Running pilot programs (AB testing) – make sure it works in practice
   - → Real-time impact of your work
   - → Successful pilot program leads to rollout.
6) Reporting.
   - → Monitor model performance & provide summarized results to stakeholders.
   - → Emphasis on model explainability & quick results.
7) Model enhancements.