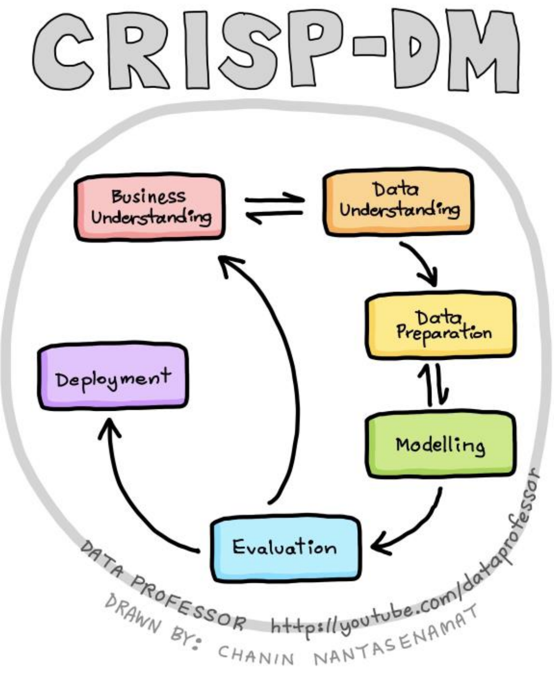
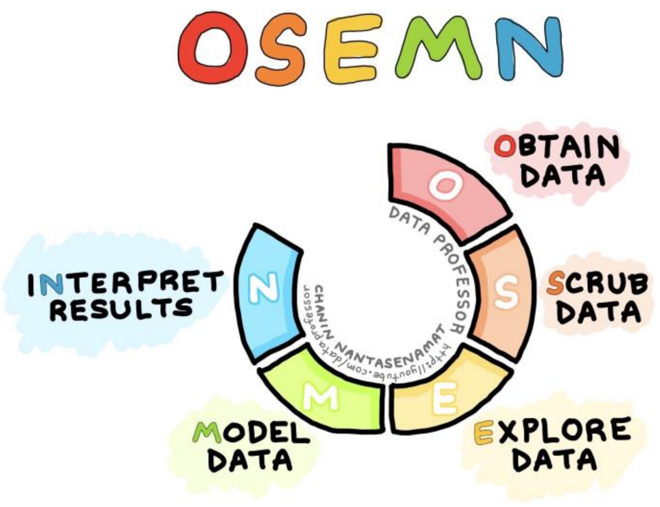
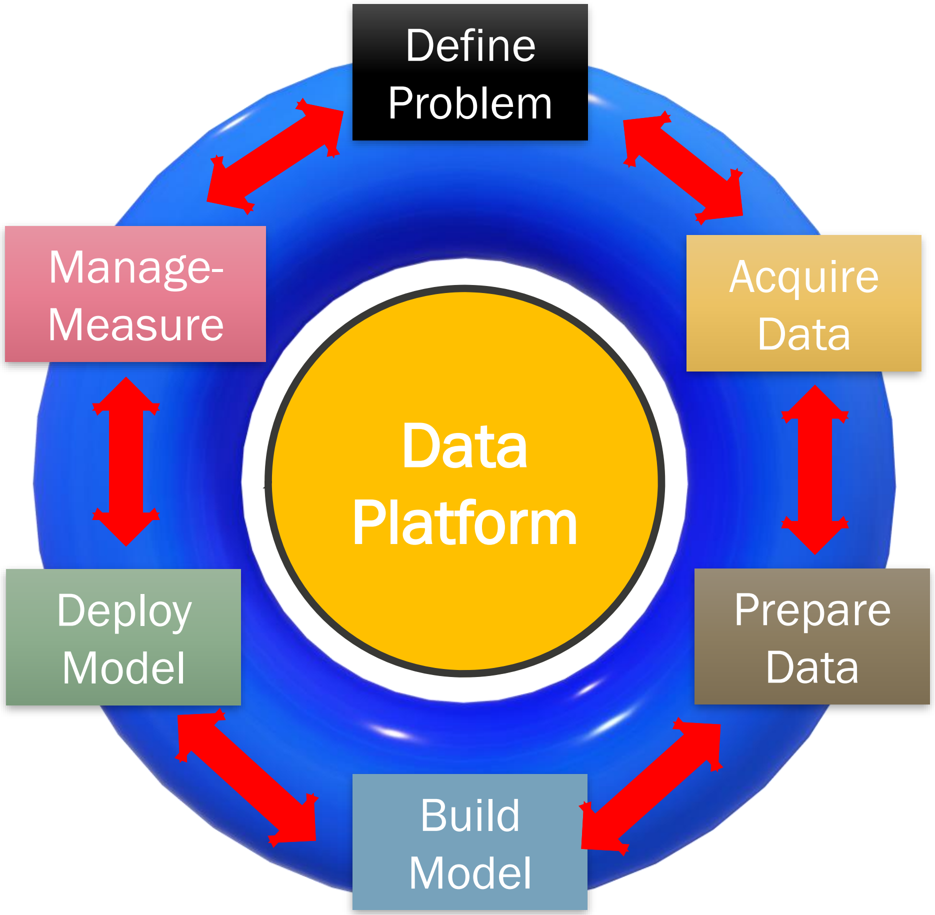
DATA DATA DATA

Thinking about Data







Define Problem

Acquire Data

Prepare Data

Build Model

Deploy Model

Manage-Measure

Data Platform

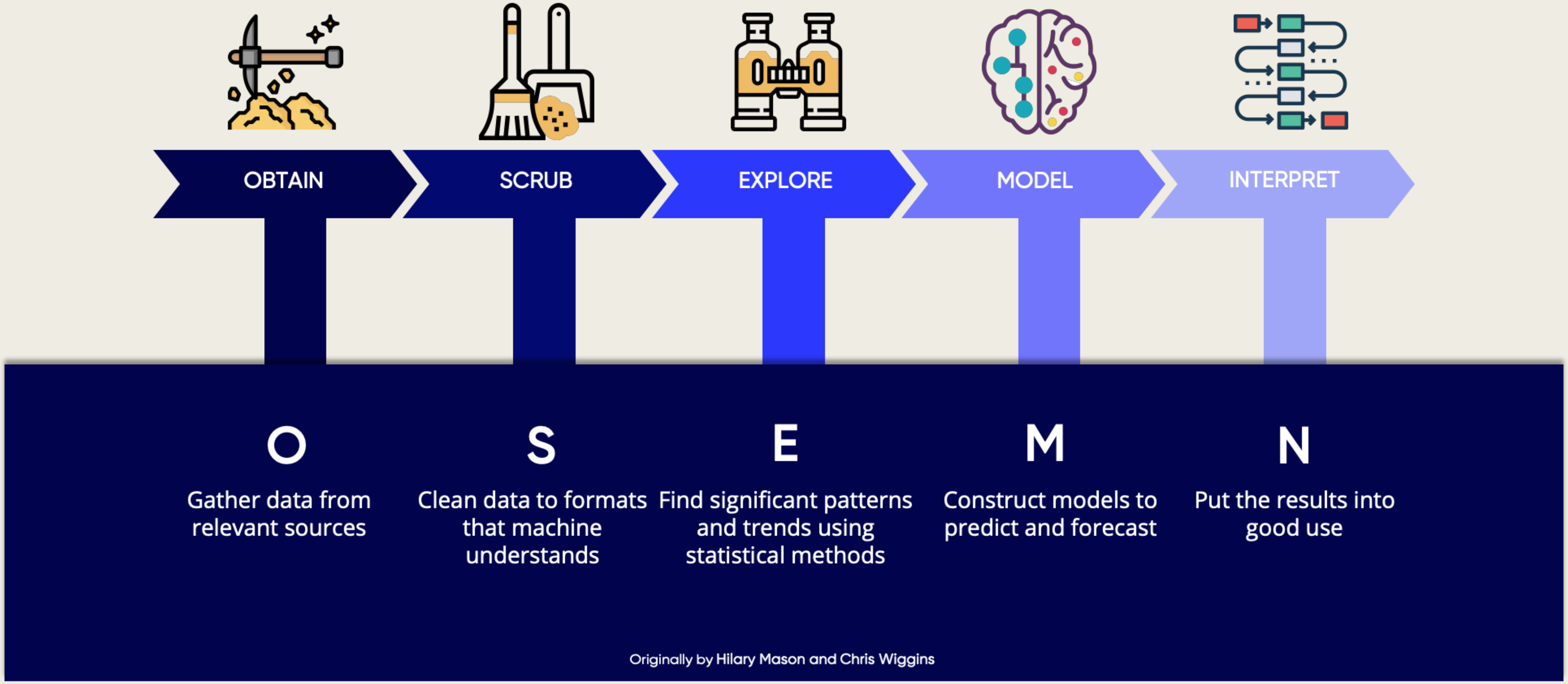
Many different ways to express the **data science process**

- a systematic approach for tackling a data problem.

Notice the similarities

Aquire, Prepare, Explore, Model, Interpret, Evaluate

Data Science Process



OBTAIN

SCRUB

EXPLORE

MODEL

INTERPRET

O

Gather data from relevant sources

S

Clean data to formats that machine understands

E

Find significant patterns and trends using statistical methods

M

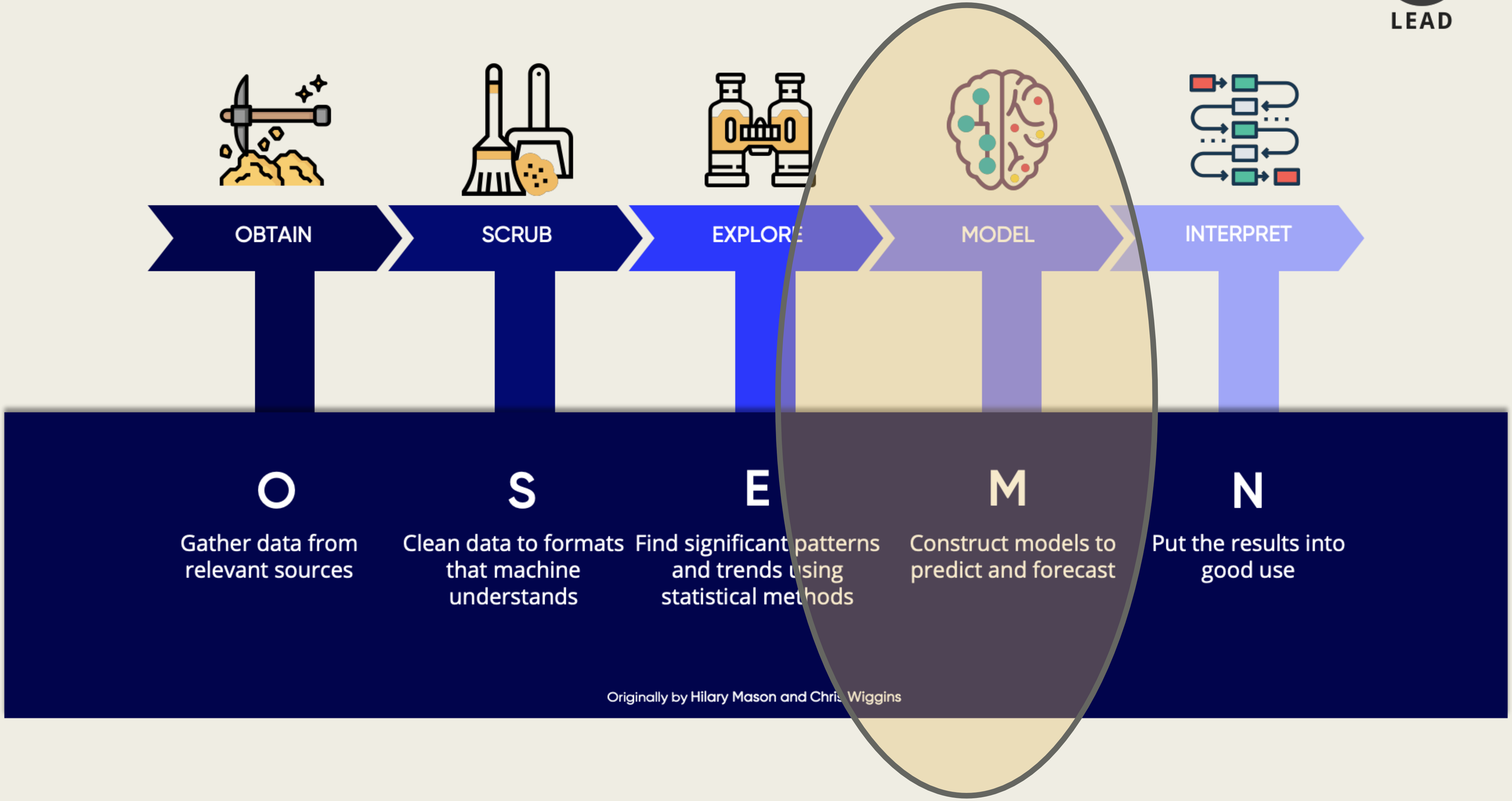
Construct models to predict and forecast

N

Put the results into good use

Originally by Hilary Mason and Chris Wiggins

Data Science Process



OBTAIN

SCRUB

EXPLORE

MODEL

INTERPRET

O
Gather data from relevant sources

S
Clean data to formats that machine understands

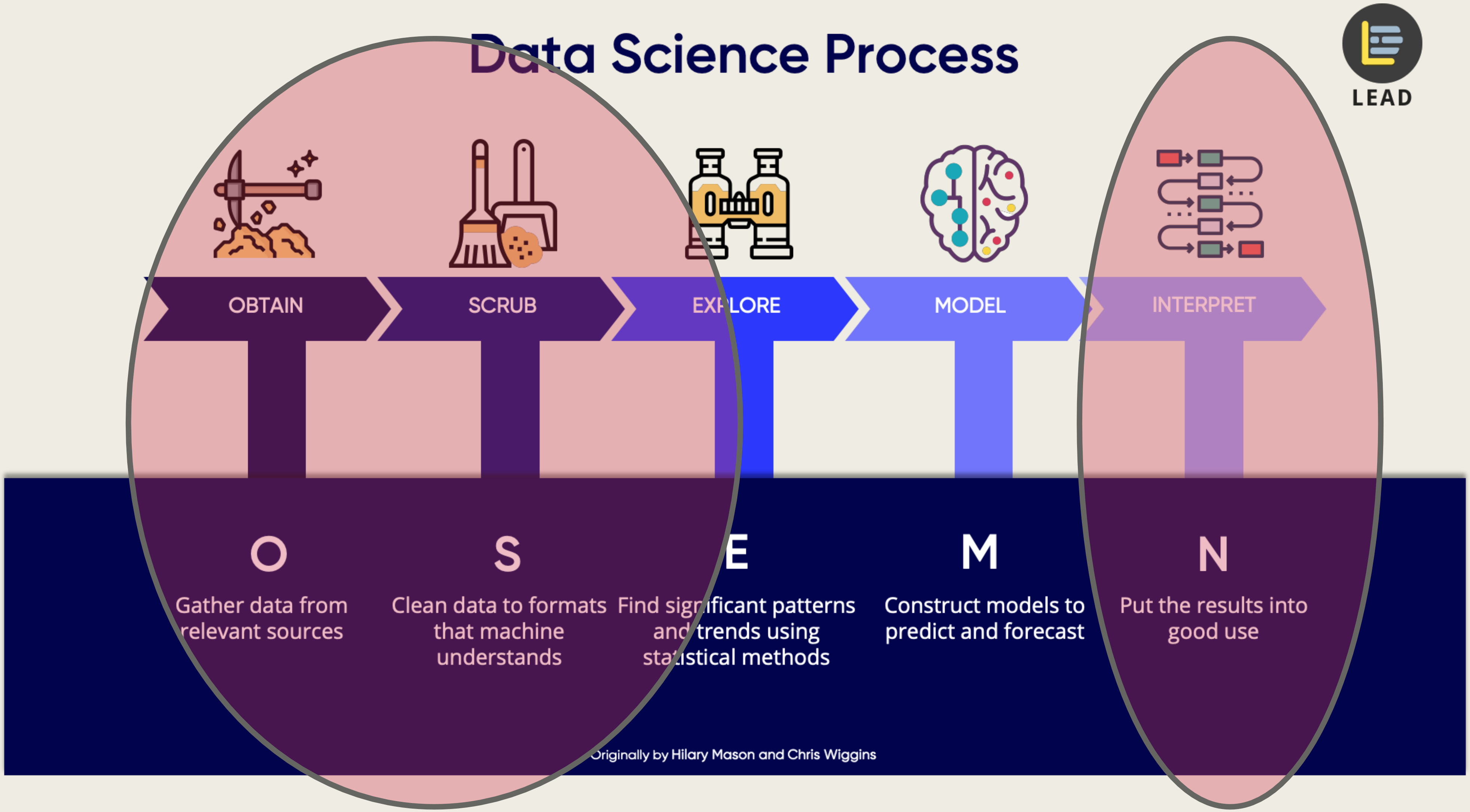
E
Find significant patterns and trends using statistical methods

M
Construct models to predict and forecast

N
Put the results into good use

Originally by Hilary Mason and Chris Wiggins

Machine Learning, Deep Learning, Artificial Intelligence: This step gets the most hype



Data Science Process

LEAD

OBTAIN

SCRUB

EXPLORE

MODEL

INTERPRET

O

Gather data from relevant sources

S

Clean data to formats that machine understands

E

Find significant patterns and trends using statistical methods

M

Construct models to predict and forecast

N

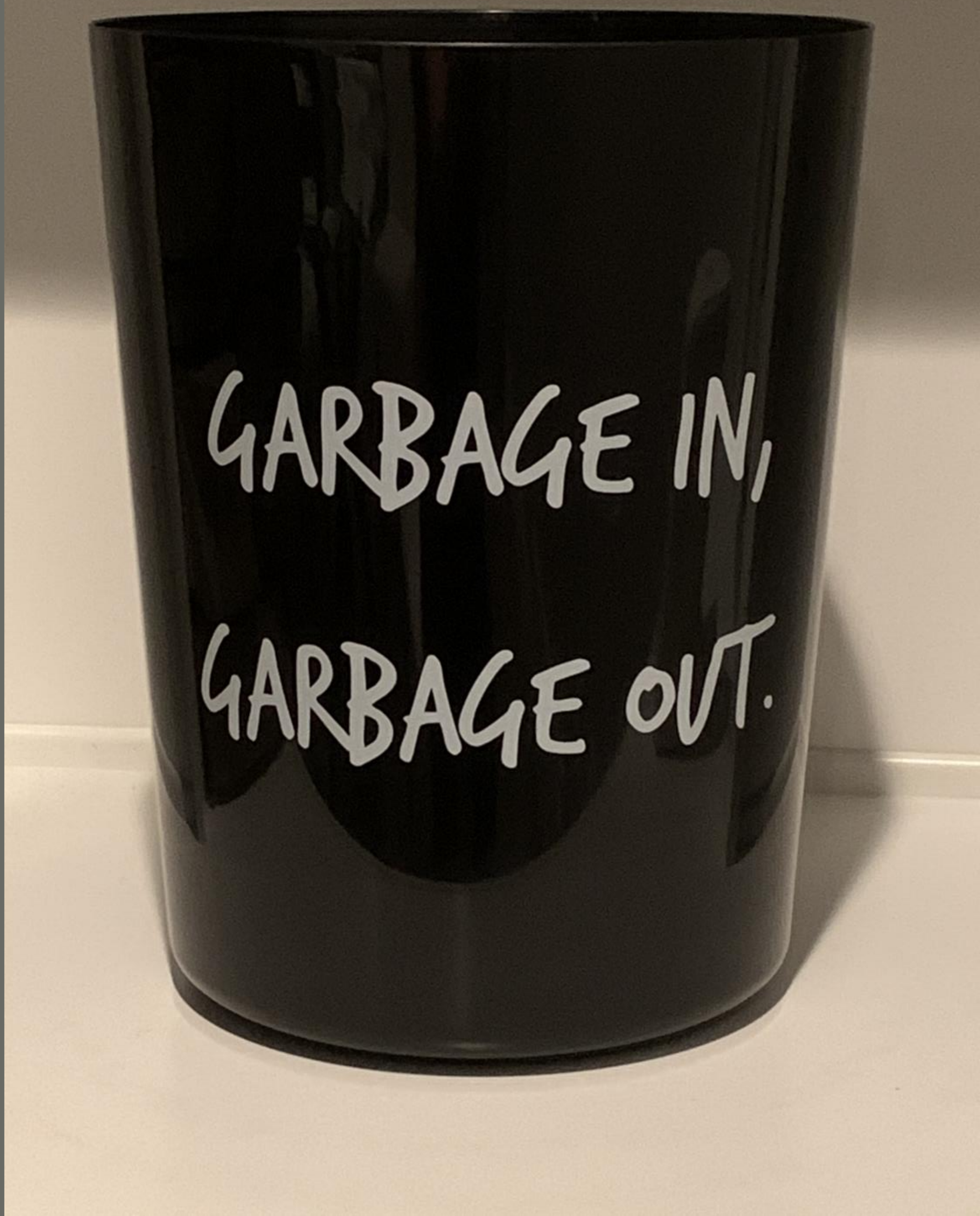
Put the results into good use

Originally by Hilary Mason and Chris Wiggins

These steps will really set you apart as a data scientist and help you add value to your company / research

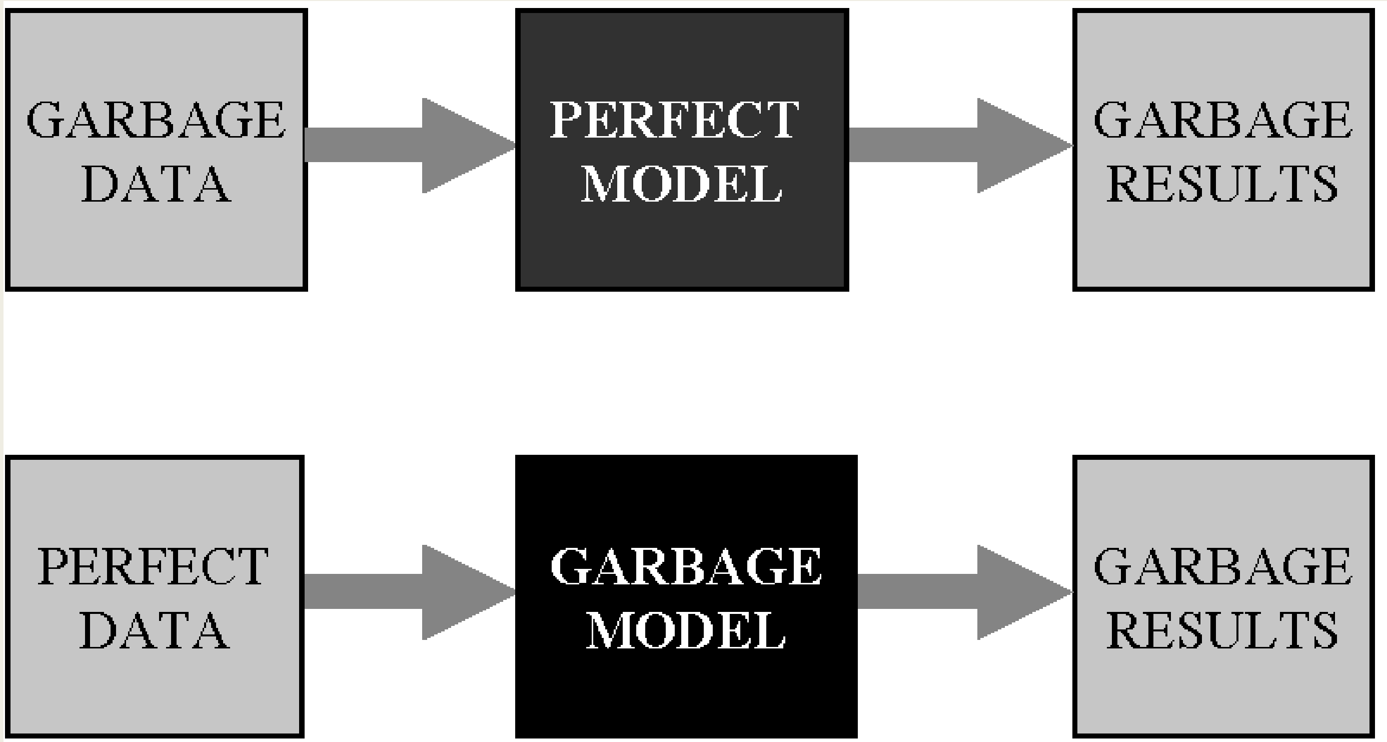
OBTAINING DATA

Data is the FUEL that powers the machine (learning)



Accuracy depends on the Data

Guiding Principle: Better to have an ok model and good data than to have THE BEST model and poor data



GARBAGE DATA → PERFECT MODEL → GARBAGE RESULTS

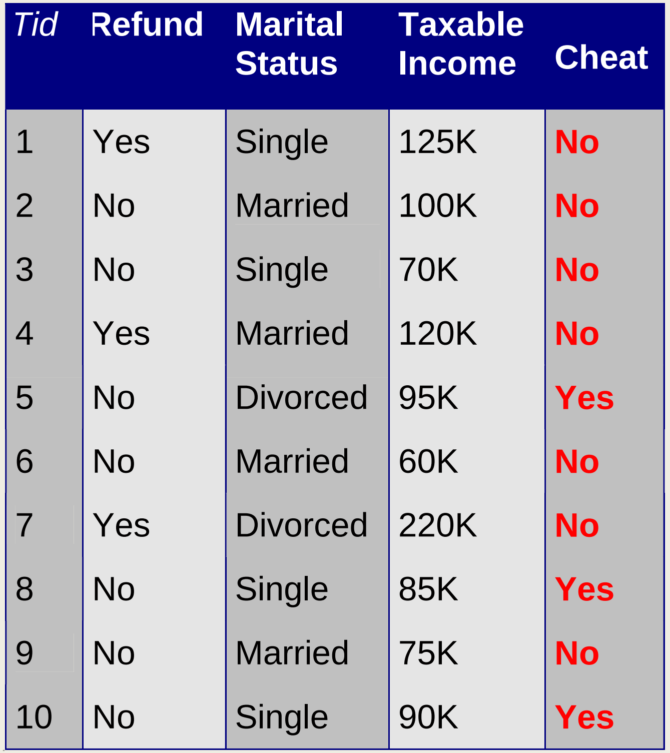
PERFECT DATA → GARBAGE MODEL → GARBAGE RESULTS

Types of data sets

- Record
  - *Data Matrix*
  - *Document Data*
  - *Transaction Data*
- Graph
  - *World Wide Web*
  - *Molecular Structures*
- Ordered
  - *Spatial Data*
  - *Temporal Data*
  - *Sequential Data*
  - *Genetic Sequence Data*
- Unstructured Data

Record Data

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



Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Data Matrix

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute

- Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Document Data

- Each document becomes a `term' vector, (bag of words)
  - *each term is a component (attribute) of the vector,*
  - *the value of each component is the number of times the corresponding term occurs in the document.*

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

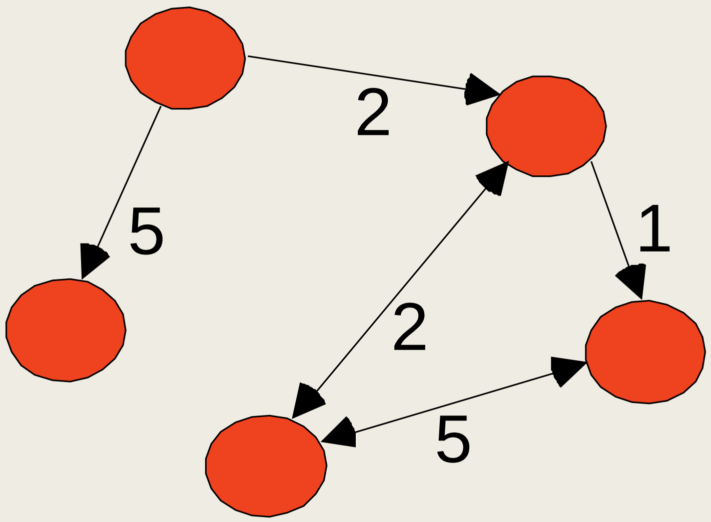
Transaction Data

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

TID	***Items***
1	**Bread, Coke, Milk**
2	**Beer, Bread**
3	**Beer, Coke, Diaper, Milk**
4	**Beer, Bread, Diaper, Milk**
5	**Coke, Diaper, Milk**

Graph Data

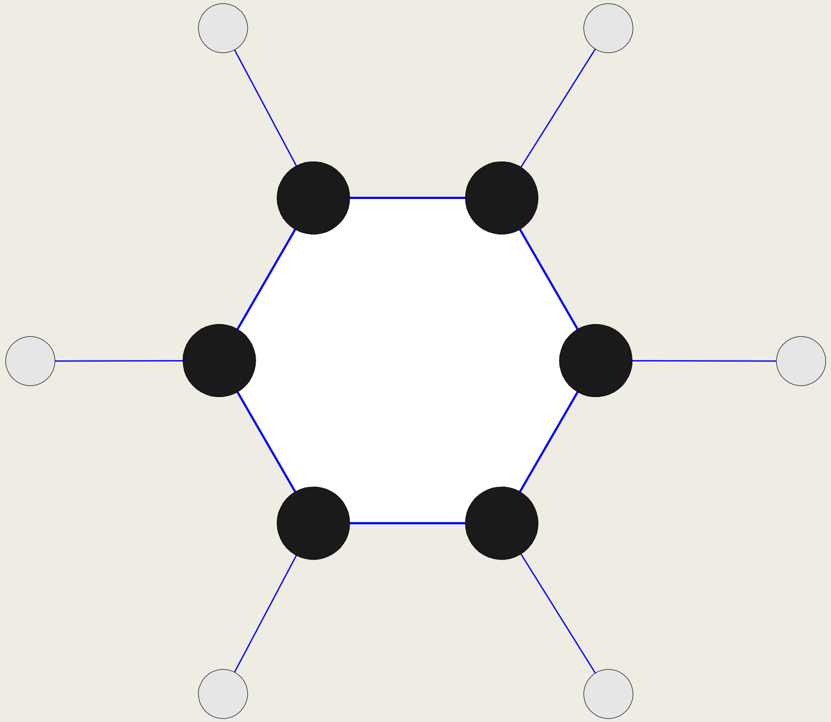
- Examples: Generic graph and HTML Links



```
<a href="papers/papers.html#bbbb">
Data Mining </a>
<li>
<a href="papers/papers.html#aaaa">
Graph Partitioning </a>
<li>
<a href="papers/papers.html#aaaa">
Parallel Solution of Sparse Linear System of Equations </a>
<li>
<a href="papers/papers.html#ffff">
N-Body Computation and Dense Linear System Solvers
```

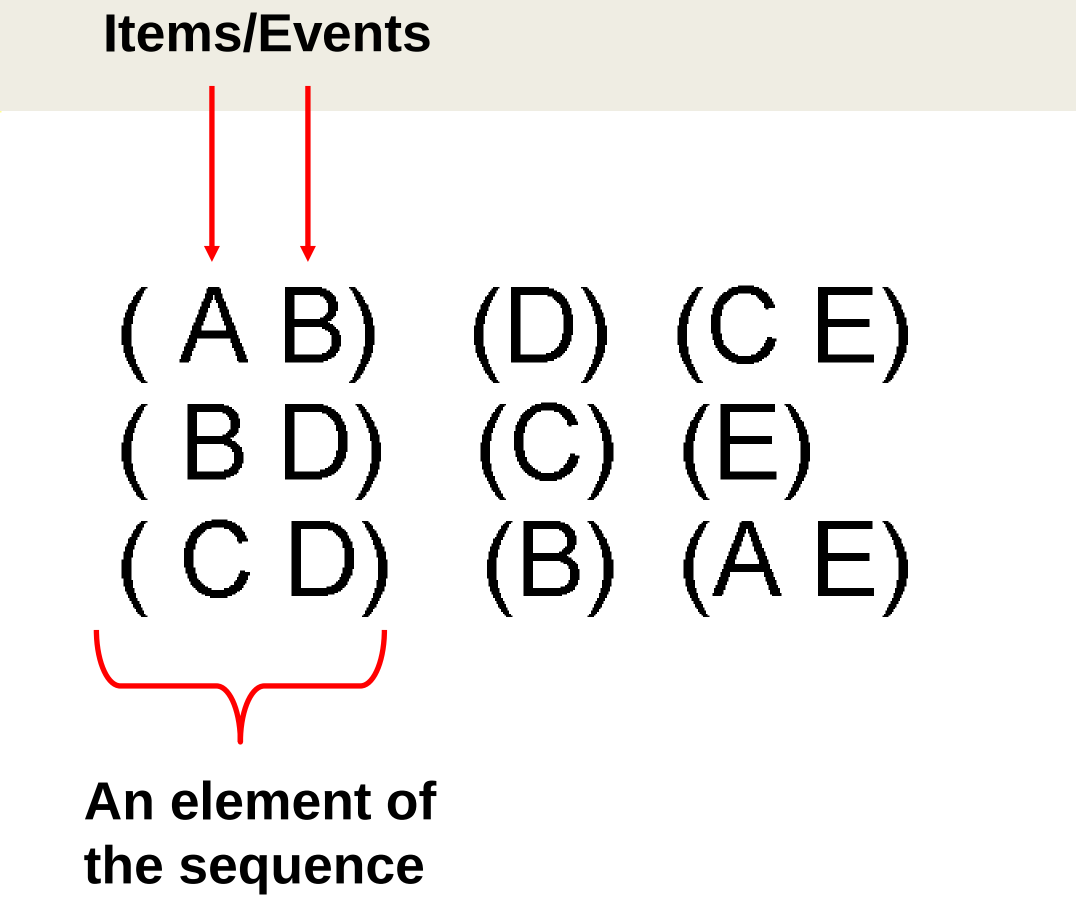
Chemical Data

- Benzene Molecule: C_6H_6



Ordered Data

- Sequences of transactions



Items/Events

(A B) (D) (C E)
(B D) (C) (E)
(C D) (B) (A E)

An element of the sequence

Ordered Data

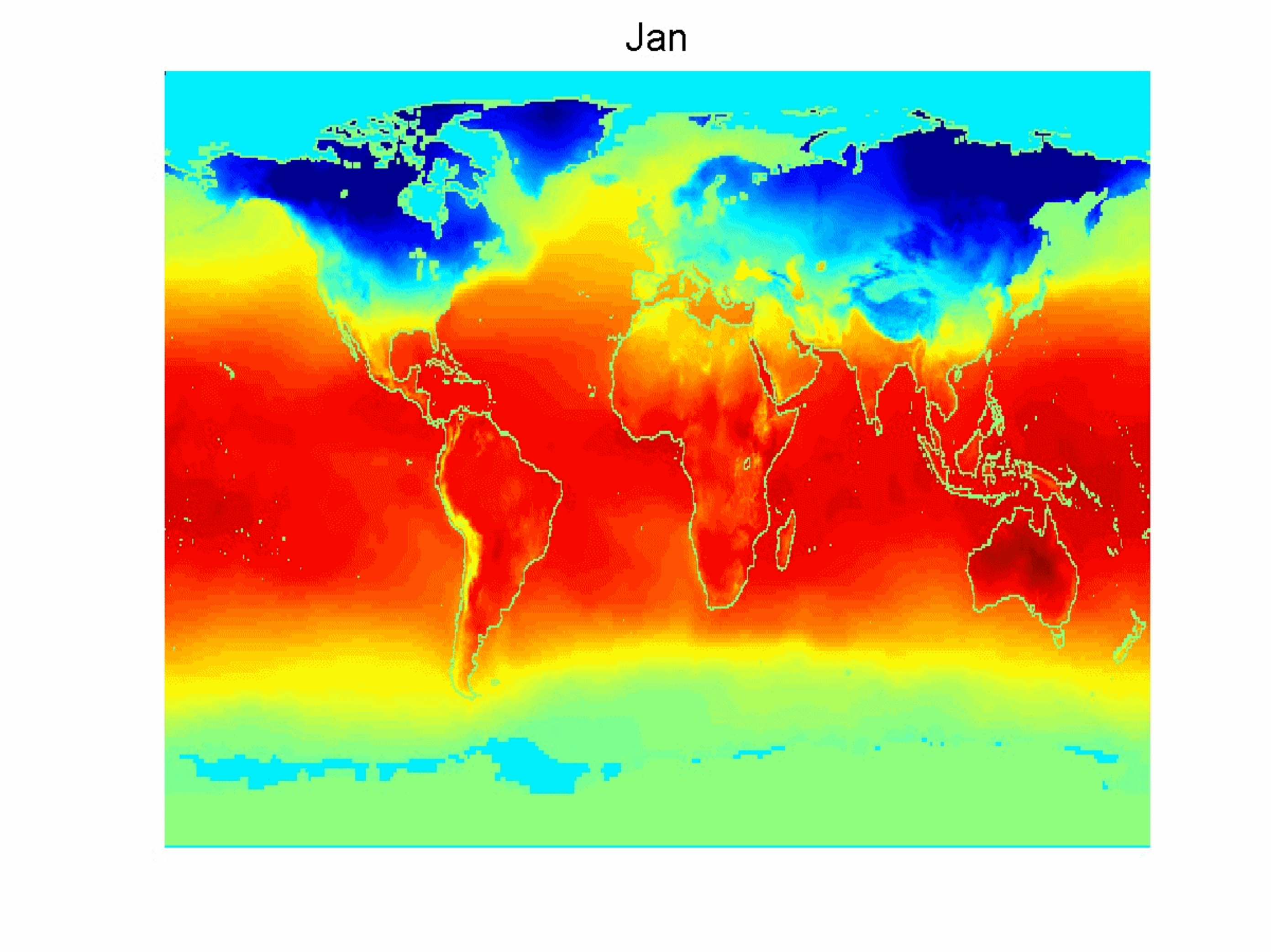
- Genomic sequence data

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

Ordered Data

- Spatio-Temporal Data

Average Monthly Temperature of land and ocean

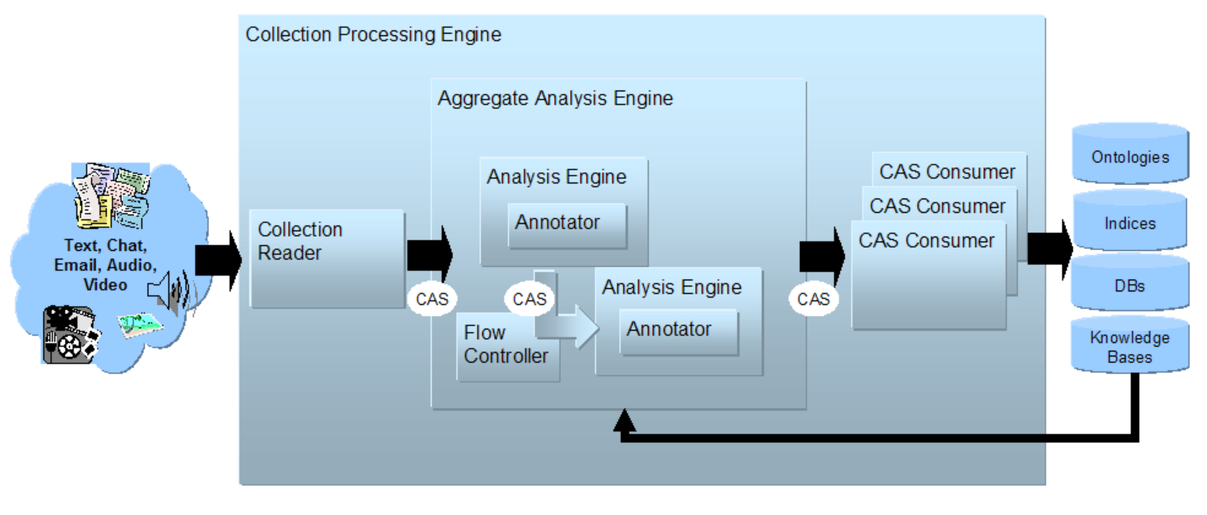


Unstructured Data

- No pre-defined data model or not organized in a pre-defined way
- Typically text heavy

- In 1998 Merrill Lynch cited a rule of thumb that somewhere between 80-90% of all potential usable business information may originate in unstructured form
- Computer World states that unstructured information might account for more than 70%–80% of all data in organizations.

Unstructured Data



- Unstructured Information Management Architecture (UIMA) is a component software architecture for analysis of unstructured information
  - *Developed by IBM*
  - *Potential use: convert unstructured data into relational tables for traditional data analysis*
- From the uima.apache.org website:
  - Unstructured Information Management applications are software systems that analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at.
- Watson (Jeopardy Challenge) uses UIMA for real-time content analytics

Things to consider

What data is available?

- *Does the available data answer the question of interest?*
- *Is the data still relevant?*
- *Is additional relevant data available?*

How much data is available?

Where did the data come from?

What is the quality of the data?

Potential Biases in Data

- Selection Bias
  - *Data is selected for convenience or perhaps self-selected – doesn't represent the population*
- Confirmation Bias
  - *We are less critical of models & data that confirm our prior beliefs*
- Societal Bias – Demographic Bias
  - *Data reflects cultural stereotypes*
- Measurement Bias
  - *Device used for data collection is faulty*
- Proxy Measurement Bias
  - *Proxy used for measurement doesn’t actually measure what we think it does*
- Exclusion Bias
  - *An important variable is left out maybe because we think it isn’t important*
- Sample Bias
  - *“Training” data does not reflect the realties of where the ML model will run*

Problems due to Bias

- Missed Opportunities
  - *Bad results lead to missed conversions, retention, etc and you operate on flawed insights*
- Skewed Customer Journey Insights
  - *Skewing the model with bad data prevents you from accurately addressing your users' needs*
- Ineffective Campaigns
  - *Biased data yields flawed insights and assumptions*
- Compliance Violations
- Bad Press - Lawsuits

Potential *Remedies for* Biases in Data

- Leverage reliable data
- Ask questions about your data
  - *Look at the forms of bias and ask questions for each case*
- Document how data is **selected** and **cleansed**
  - *Transparency allows for root cause analysis*
- Audit your data and models for bias
  - *Examine your data distribution. Is it representative?*
  - *What features are most important to your model?*
    - Use a feature importance package – SHAP
    - Remove a feature and see how the model performs
  - *Perform regular audits.*
  - *Add in measures of bias that run during deployment*
- Know the compliance regulations

Twyman's Law

Any statistic or result that appears interesting is probably a mistake

https://www.linkedin.com/pulse/twymans-law-ronny-kohavi/

https://www.exp-platform.com/Documents/TwymansLaw.pdf

For Example:

New York Times article from 1989 "On Landing Like a Cat: It is a Fact”

"Every year, scores of cats fall from open windows in NYC. From June 4 through Nov 4, 1984, for example, 132 such victims were admitted to [an] Animal Medical Center. … Most of the cats landed on concrete. Most survived. Experts believe they were able to do so because of the laws of physics, superior balance and what might be called the flying-squirrel tactic. … Even more surprising, the longer the fall, the greater the chance of survival."

For Example:

New York Times article from 1989 "On Landing Like a Cat: It is a Fact”

"Every year, scores of cats fall from open windows in NYC. From June 4 through Nov 4, 1984, for example, 132 such victims were admitted to [an] Animal Medical Center. … Most of the cats landed on concrete. Most survived. Experts believe they were able to do so because of the laws of physics, superior balance and what might be called the flying-squirrel tactic. … Even more surprising, the longer the fall, the greater the chance of survival."

For Example:

New York Tin is a Fact”

"Every year, s rom June 4
through Nov dmitted to [an]
Animal Medi ete. Most
survived. Ex f the laws of
physics, sup g-squirrel
tactic. ... Eve er the chance
of survival."

No one brings their dead cat to the hospital. Cats that are clearly alive after a fall are more likely to be taken to the hospital, and more likely to survive. Ambiguity of a cat's condition likely decreases with the height of the fall (cats are either obviously dead or obviously alive)

Other Examples

5% of your customers were born on Jan 1

You see a sales decline on March 12, 2017 on all US e-commerce sites

Customers willing to receive emails also tend to buy your product

Other Examples

5% of your customers were born on Jan 1

- *If birthday is a mandatory field, many choose Jan 1 for ease*

You see a sales decline on March 12, 2017 on all US e-commerce sites

- *Daylight savings loses an hour*

Customers willing to receive emails also tend to buy your product

- *Default value at checkout is "accept emails"*

Take Home Message

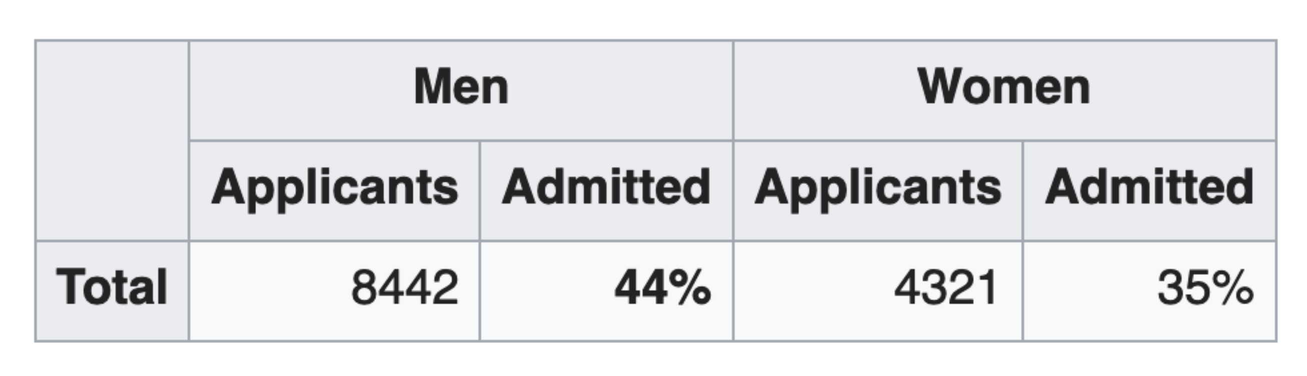
Have	Have cautious optimism and healthy skepticism
Validate	Validate all discoveries
Learn	If there is bias in the data, the model will learn the bias
Ensure	Ensure data is sufficient, relevant, and of high quality
Uncover	Uncover existing data biases and remove them if possible
Be	Be transparent in your assumptions
Know	Know how your data was collected

SIMPSON’S PARADOX

When a data trend reverses based on the addition or exclusion of another variable

Famous Example: Gender Bias at Berkley (1973)

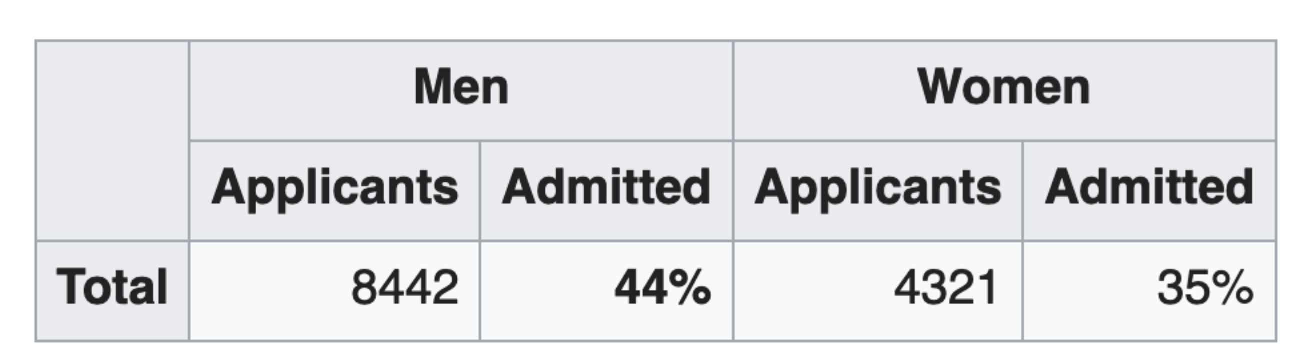
Are men applying to Berkeley more likely to get in than women?



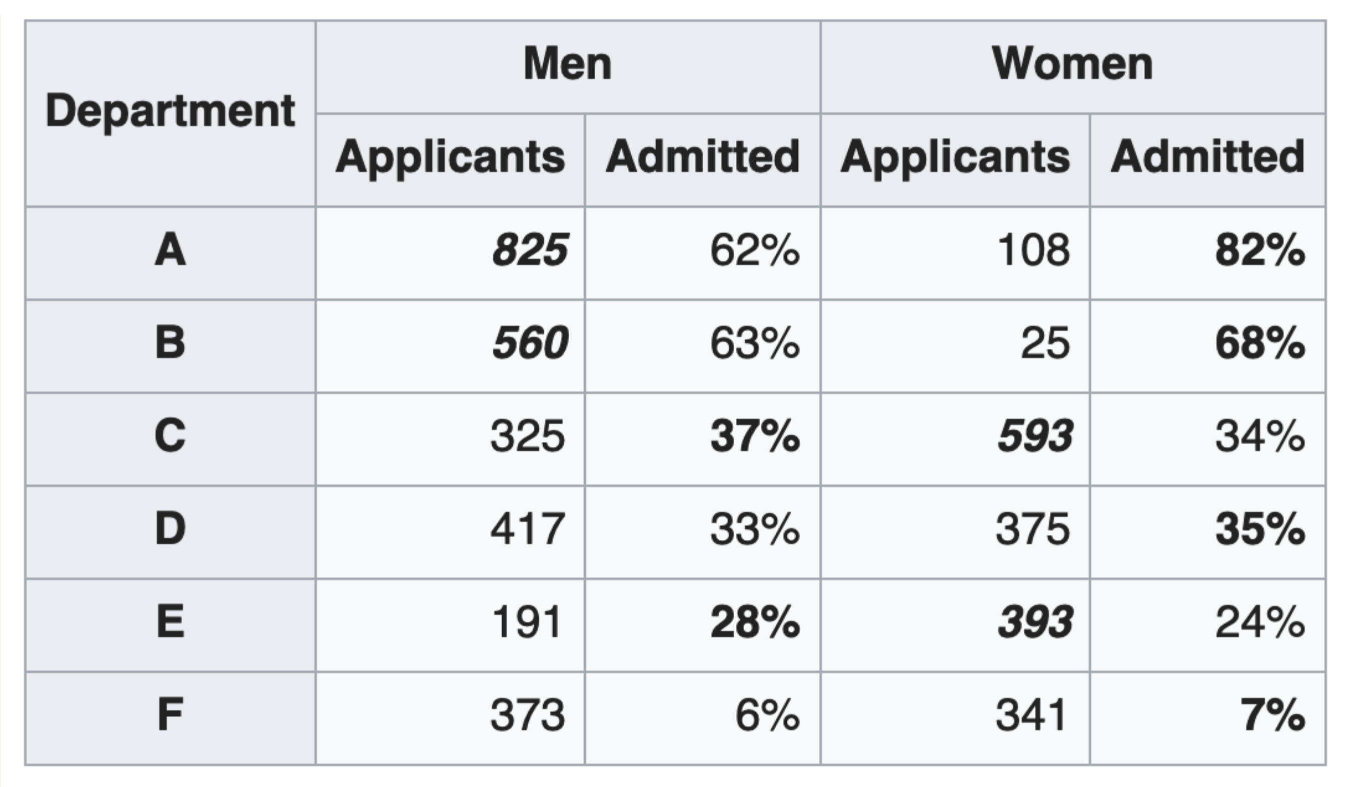
	Men		Women	
	Applicants	**Admitted**	**Applicants**	**Admitted**
Total	8442	**44%**	4321	35%

Famous Example: Gender Bias at Berkley (1973)

Are men applying to Berkeley more likely to get in than women?



	Men		Women	
	Applicants	**Admitted**	**Applicants**	**Admitted**
Total	8442	**44%**	4321	35%

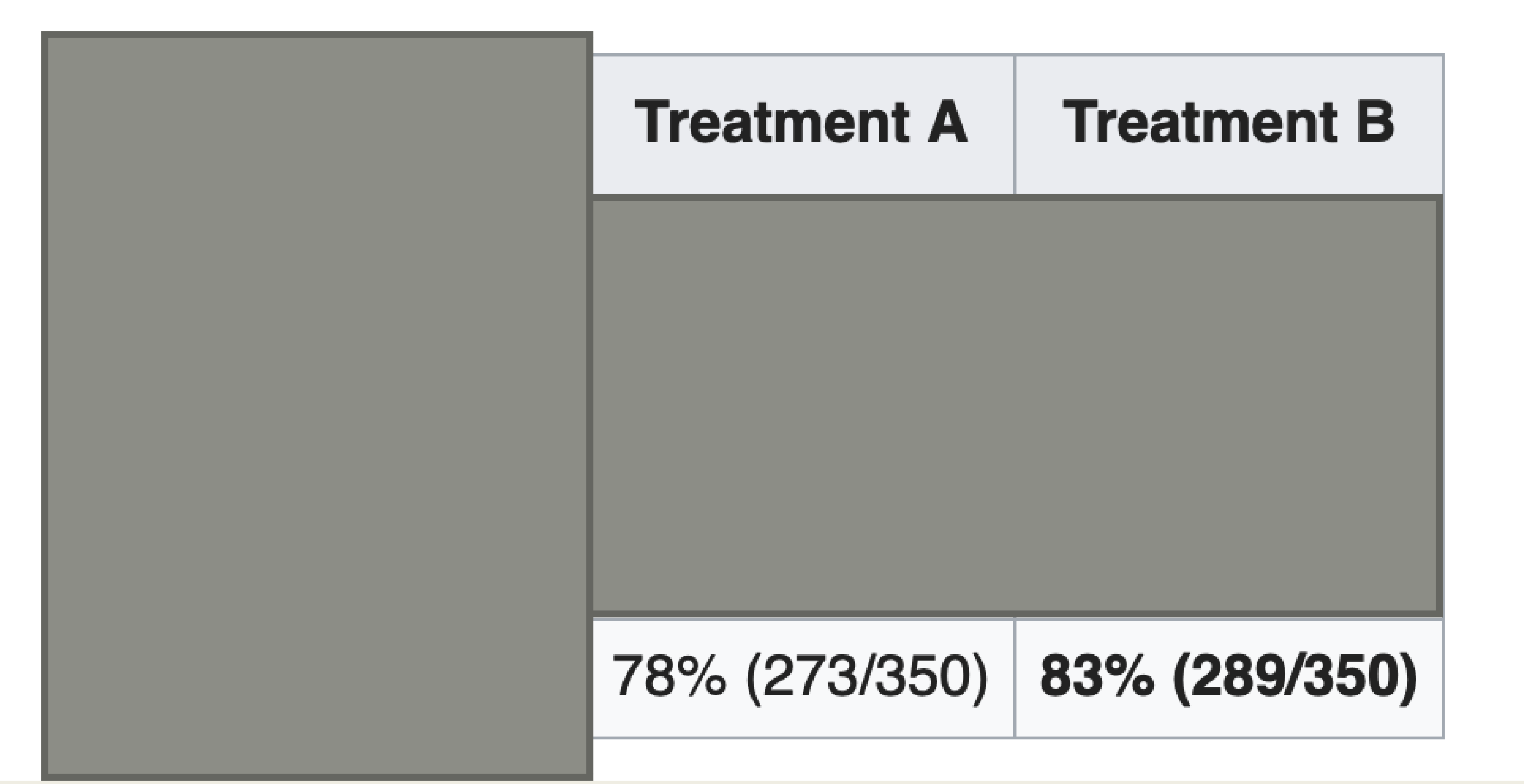


Department	**Men**		**Women**	
	Applicants	**Admitted**	**Applicants**	**Admitted**
A	***825***	62%	108	**82%**
B	***560***	63%	25	**68%**
C	325	**37%**	***593***	34%
D	417	33%	375	**35%**
E	191	**28%**	***393***	24%
F	373	6%	341	**7%**

Departments have different acceptance rates and more women applied to departments with lower acceptance rates

Famous Famous Example: Kidney Stone Treatment

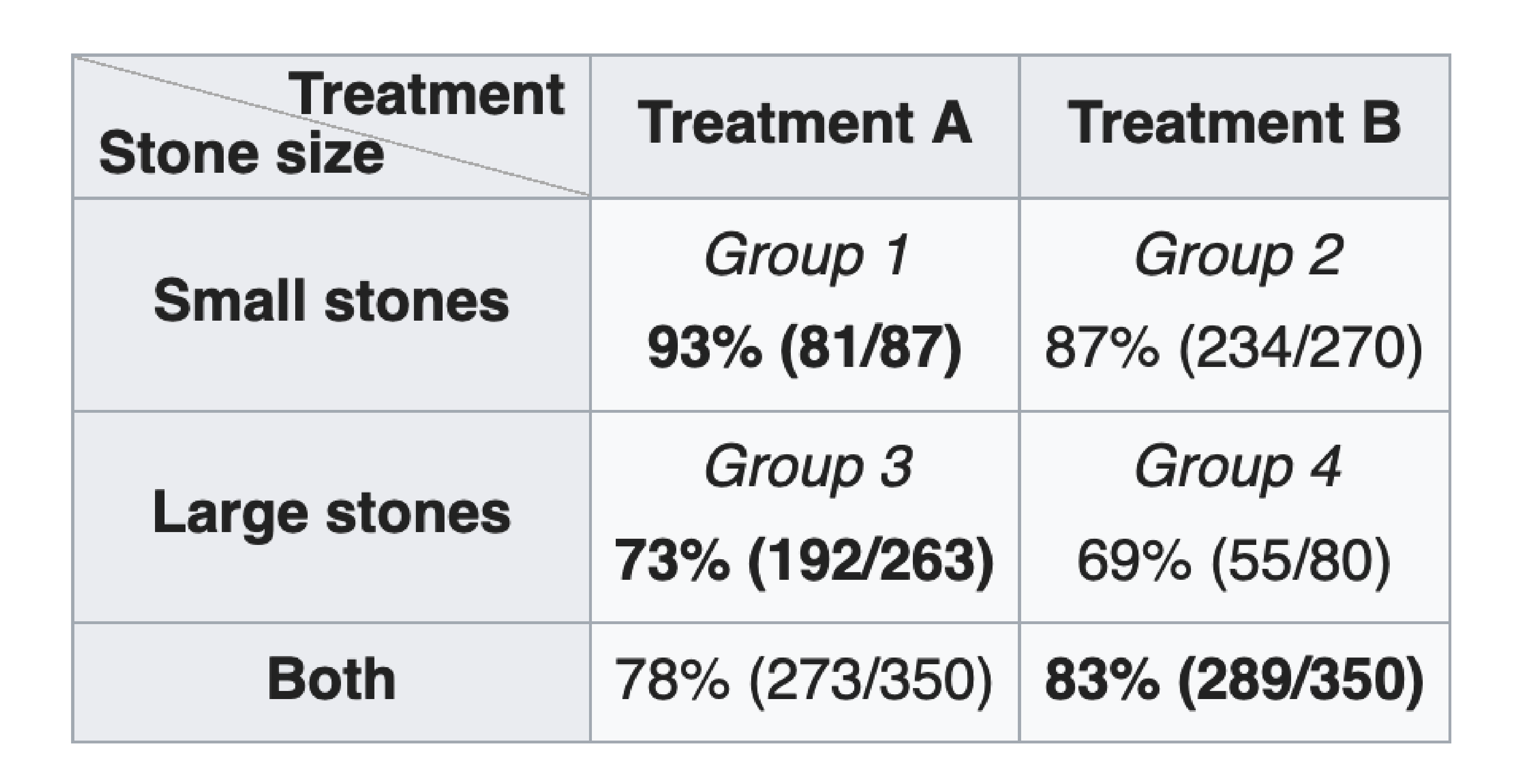
Based on the success rate, which treatment is better?



[redacted]	Treatment A	Treatment B
	[redacted]	
	78% (273/350)	**83% (289/350)**

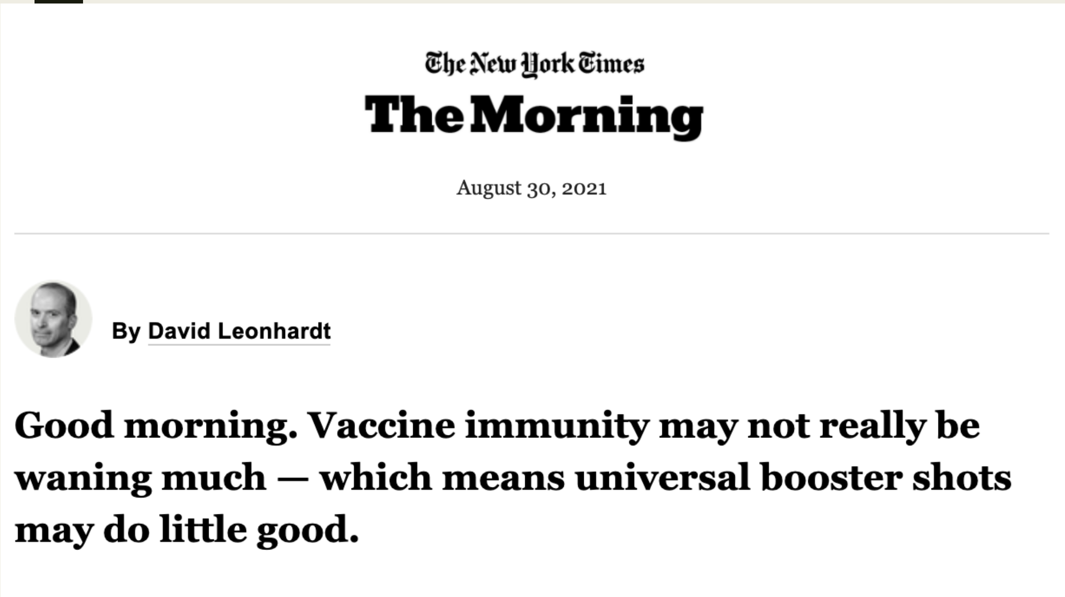
Famous Famous Example: Kidney Stone Treatment

Based on the success rate, which treatment is better?



Treatment Stone size	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

The less effective treatment (B) is applied more frequently to less severe cases and the more effective treatment (A) is applied more frequently to more severe cases.



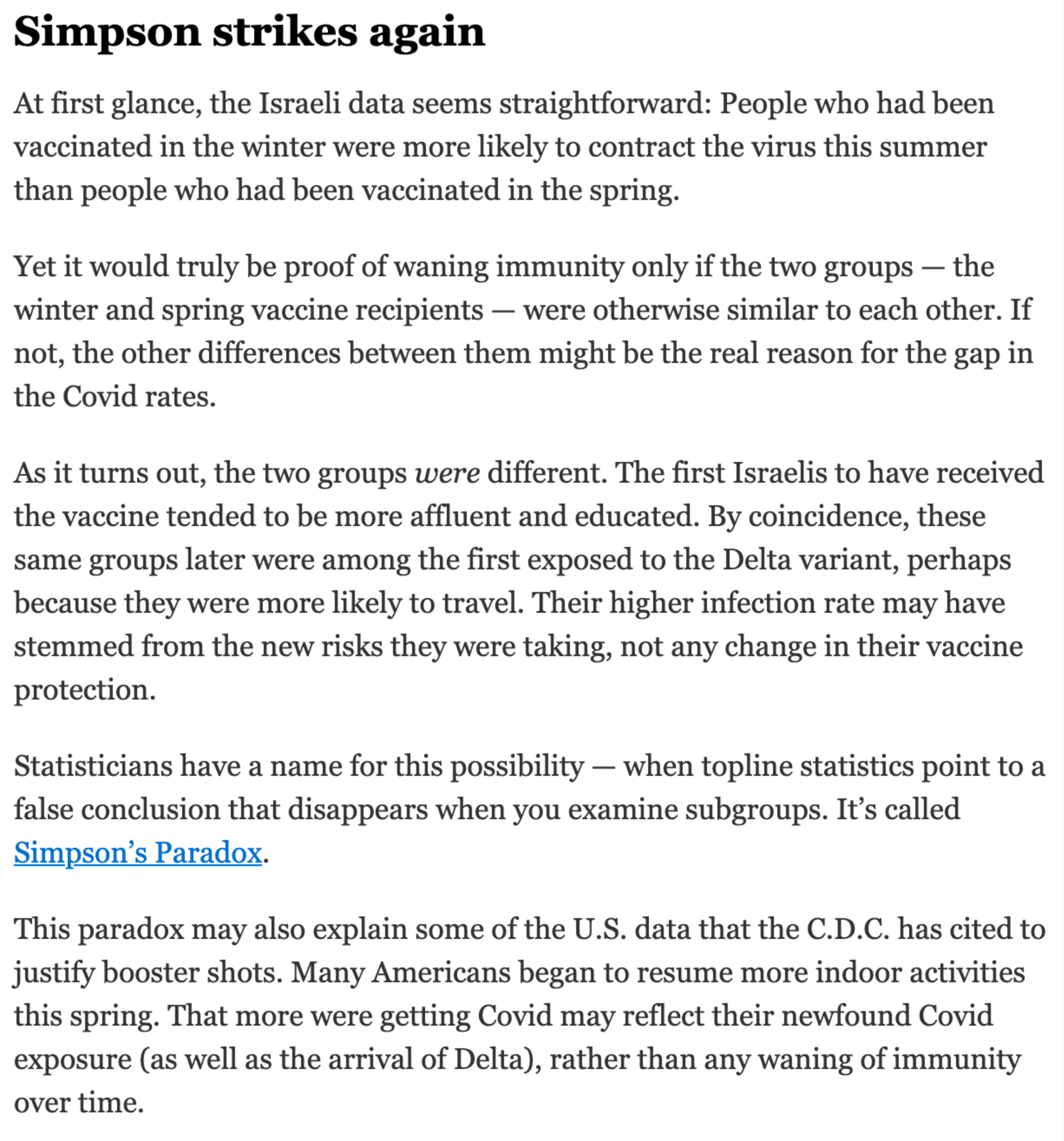
The New York Times

The Morning

August 30, 2021

By David Leonhardt

Good morning. Vaccine immunity may not really be waning much — which means universal booster shots may do little good.



Simpson strikes again

At first glance, the Israeli data seems straightforward: People who had been vaccinated in the winter were more likely to contract the virus this summer than people who had been vaccinated in the spring.

Yet it would truly be proof of waning immunity only if the two groups — the winter and spring vaccine recipients — were otherwise similar to each other. If not, the other differences between them might be the real reason for the gap in the Covid rates.

As it turns out, the two groups *were* different. The first Israelis to have received the vaccine tended to be more affluent and educated. By coincidence, these same groups later were among the first exposed to the Delta variant, perhaps because they were more likely to travel. Their higher infection rate may have stemmed from the new risks they were taking, not any change in their vaccine protection.

Statisticians have a name for this possibility — when topline statistics point to a false conclusion that disappears when you examine subgroups. It's called Simpson's Paradox.

This paradox may also explain some of the U.S. data that the C.D.C. has cited to justify booster shots. Many Americans began to resume more indoor activities this spring. That more were getting Covid may reflect their newfound Covid exposure (as well as the arrival of Delta), rather than any waning of immunity over time.

1

Be careful of lurking / confounding variables

2

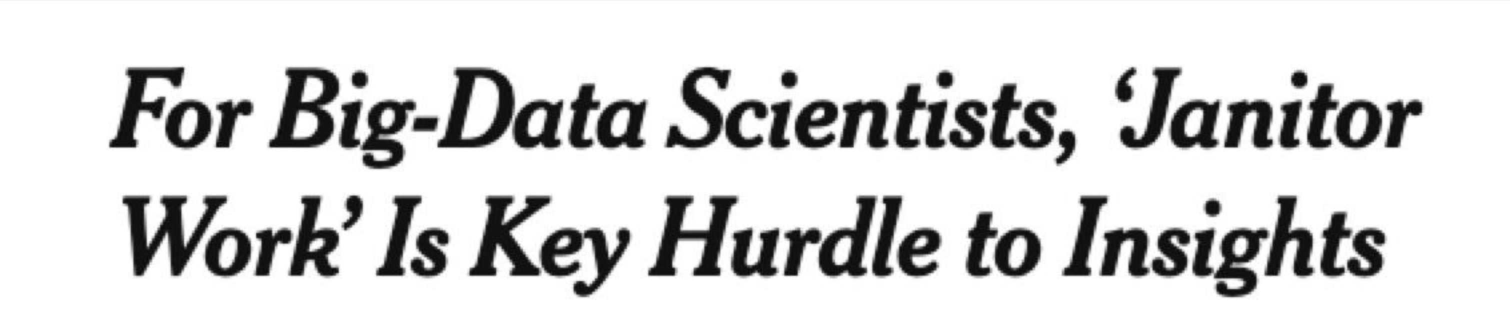
Lurking variable might not even be among the data collected

3

Be very careful not to infer causality from what are only correlations!

Take Home Message

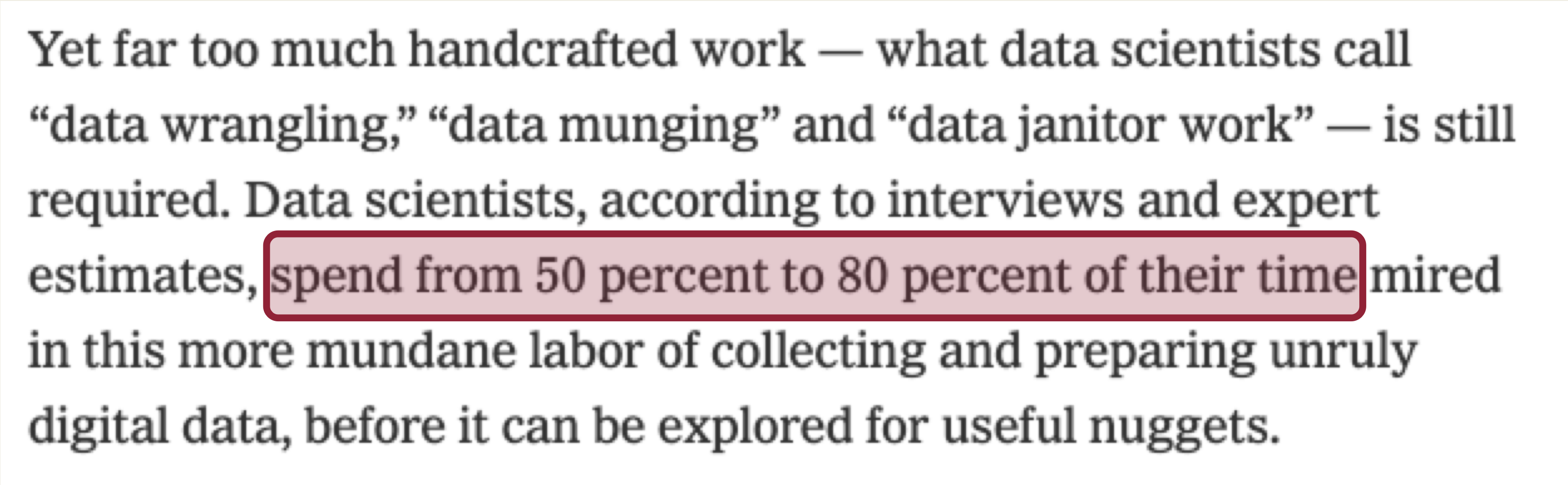
SCRUBBING DATA



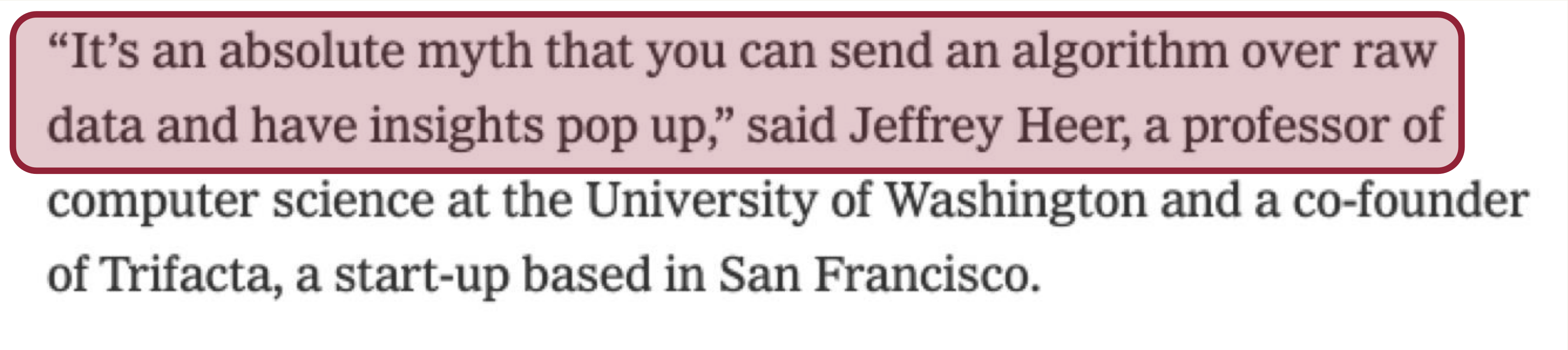
For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

The New York Times

Aug. 17, 2014



Yet far too much handcrafted work — what data scientists call "data wrangling," "data munging" and "data janitor work" — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.



"It's an absolute myth that you can send an algorithm over raw data and have insights pop up," said Jeffrey Heer, a professor of computer science at the University of Washington and a co-founder of Trifacta, a start-up based in San Francisco.

https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

Definitions: same thing, different names

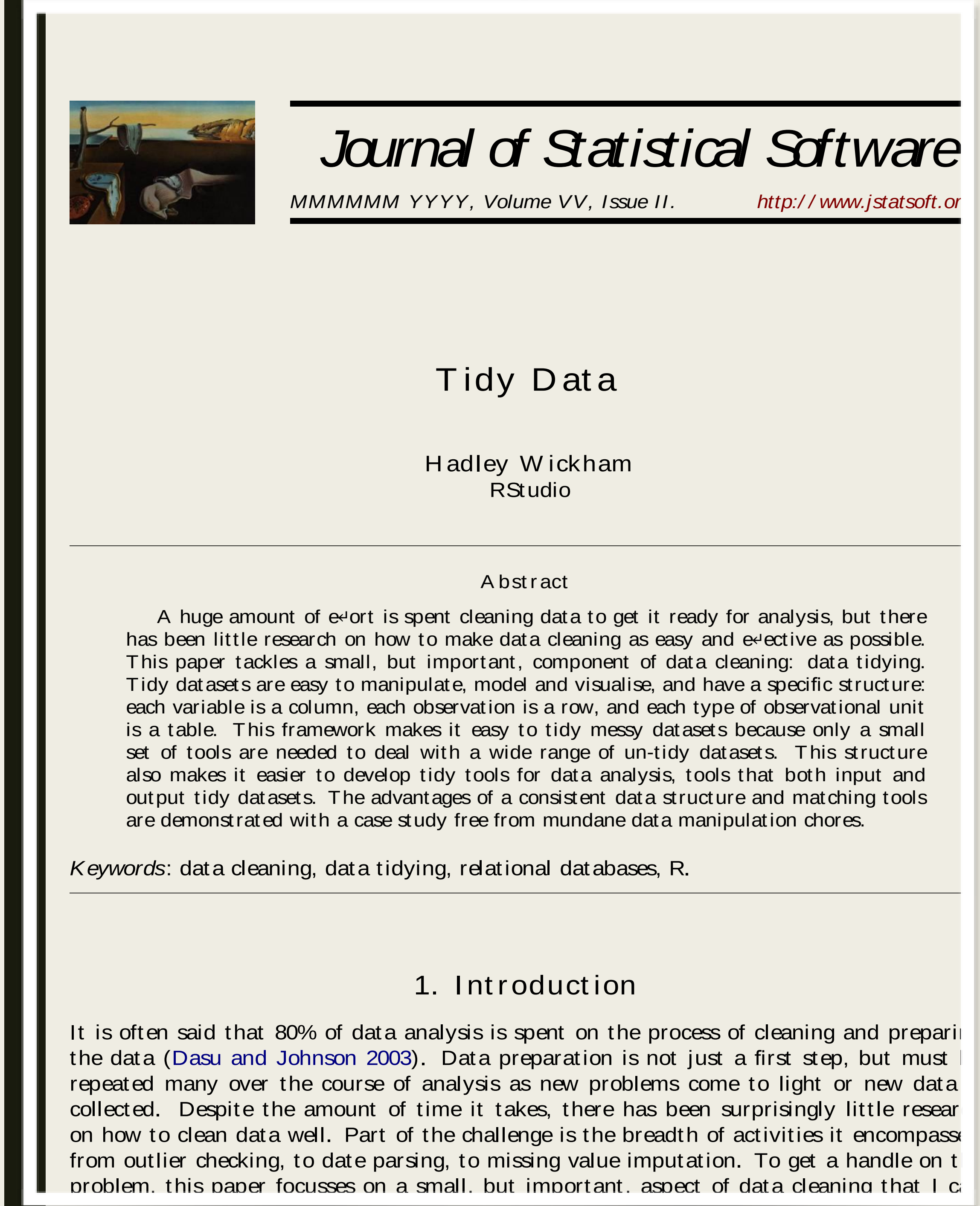
The thing you want to predict

- *Response*
- *Response variable*
- *Dependent variable*
- *Target*

The things you use to predict

- *Explanatory variables*
- *Independent variables*
- *Factors*
- *Features*

Tidy Data



Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. http://www.jstatsoft.or

Tidy Data

Hadley Wickham
RStudio

Abstract

A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible. This paper tackles a small, but important, component of data cleaning: data tidying. Tidy datasets are easy to manipulate, model and visualise, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table. This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.

Keywords: data cleaning, data tidying, relational databases, R.

1. Introduction

It is often said that 80% of data analysis is spent on the process of cleaning and prepari the data (Dasu and Johnson 2003). Data preparation is not just a first step, but must repeated many over the course of analysis as new problems come to light or new data collected. Despite the amount of time it takes, there has been surprisingly little resear on how to clean data well. Part of the challenge is the breadth of activities it encompass from outlier checking, to date parsing, to missing value imputation. To get a handle on t problem, this paper focusses on a small, but important, aspect of data cleaning that I ca

- The principles of tidy data provide a standard way to organize a dataset

- To tidy a dataset means to structure it in a way that facilitates analysis

- "Tidy datasets are all alike but every messy dataset is messy in its own way"

Principles of Tidy Data

Each variable (feature) forms a column

Each observation forms a row

Each type of observational unit forms a table

Basically, this is Third Normal Form

See Tidy Data paper for examples of common problems and how to fix them https://vita.had.co.nz/papers/tidy-data.pdf

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

Data Cleaning – Best Practices

- Establish a pipeline that is repeatable
  - *Use it on the training data, test data, **and production data***
- Use domain expertise
- Understand the data gathering process and behaviors of instruments and people
  - *How does an instrument act when it is failing or confused*
  - *What do subjects do when they are filling out forms*
  - *Peculiarities of the process*
- Document every transformation
- Keep a copy of the uncleaned data

- Fixing missing data
- Fixing outliers and noise
- Changing data types
- Data transformations
- Problem Features
- Feature engineering
- Feature Selection / Data Reduction

Common Data Cleaning / Preparation Tasks

FIXING MISSING DATA

Missing Data

1. Structurally missing
   - *Missing because it does not exist (# pregnant men)*
2. Missing completely at random
   - *Whether or not a value is missing is unrelated to its value or the value of any other variable*
   - *Missing values can be excluded or imputed with mean or median*
3. Missing at random
   - *Whether or not a value is missing does not depend on its value after controlling for another variable*
   - *Missing values can be predicted from other variables*
4. Missing not at random
   - *Missing values that are not MCAR or MAR*
   - *Hardest type of missingness*

For more information: https://www.uvm.edu/~statdhtx/StatPages/Missing_Data/Missing.html

Missing Data

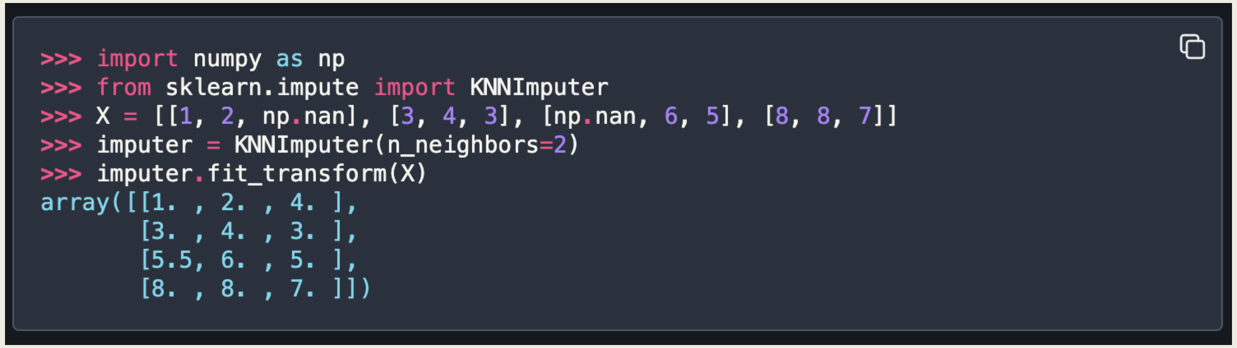
- Missing Completely at Random (MCAR)
  - A survey participant accidentally skips a question
  - A batch of lab samples is processed improperly
  - A weighing scale runs out of batteries
  - A computer glitch deletes some entries
  - *Missing values can be excluded or imputed with mean or median*

- Missing at Random (MAR)
  - Imagine a survey where people are asked about their income, but some individuals choose not to answer. If the likelihood of not answering the income question is only related to their age (e.g., younger people are less likely to respond), and not to their actual income level itself (once age is considered), then the data would be considered MAR.
  - *Missing values can be predicted from other variables*

Missing Data

BP	Pain	Out
.5	Low	0
.8	Hi	1
?	Low	1
1	?	0

- Need to consider approach for learning and execution (could differ)
- Throw out data with missing attributes
  - *Do only if rare, else could lose a significant amount of training set*
  - *Doesn't work during execution*
- Set (impute/imputation) attribute to its mode/mean (based on rest of data set)
  - *Too big of an assumption?*
- Use a learning scheme (NN, DT, etc) to impute missing values
  - *Train imputing models with a training set which has the missing attribute as the target and the rest of the attributes as input features. Better accuracy, though more time consuming - multiple missing values?*
- Impute based on the most similar instance(s) in the data set - SK KNNImputer



```
>>> import numpy as np
>>> from sklearn.impute import KNNImputer
>>> X = [[1, 2, np.nan], [3, 4, 3], [np.nan, 6, 5], [8, 8, 7]]
>>> imputer = KNNImputer(n_neighbors=2)
>>> imputer.fit_transform(X)
array([[1. , 2. , 4. ],
       [3. , 4. , 3. ],
       [5.5, 6. , 5. ],
       [8. , 8. , 7. ]])
```

- *Let unknown be just another attribute value – Can work well in many cases
  - *Missing attribute may contain important information, (didn't vote can mean something about congressperson, extreme measurements aren't captured, etc.).*
  - *Natural for nominal data*
  - *With continuous data, can use an indicator node, or a value which does not occur in the normal data (-1, outside range, etc.), however, in the latter case, the model will treat this as an extreme ordered feature value and may cause difficulties*

Challenge Question - Missing Data

- Given the dataset to the right, what approach would your group use to deal with the two unknown values and what would those values be set to?
  - *Assume you are learning with an MLP and Backpropagation*
  - *Throwing out those instances is not an option in this case*
  - *I left out targets on purpose since they won't be available with novel data*
- If your approach requires any modifications to the MLP inputs, be ready to explain exactly what those are

x	*y*	*z*	*Out*
.1	.6	B	
.6	.6	G	
.3	.35	R	
.6	.45	G	
.6	.4	?	
.8	?	B	

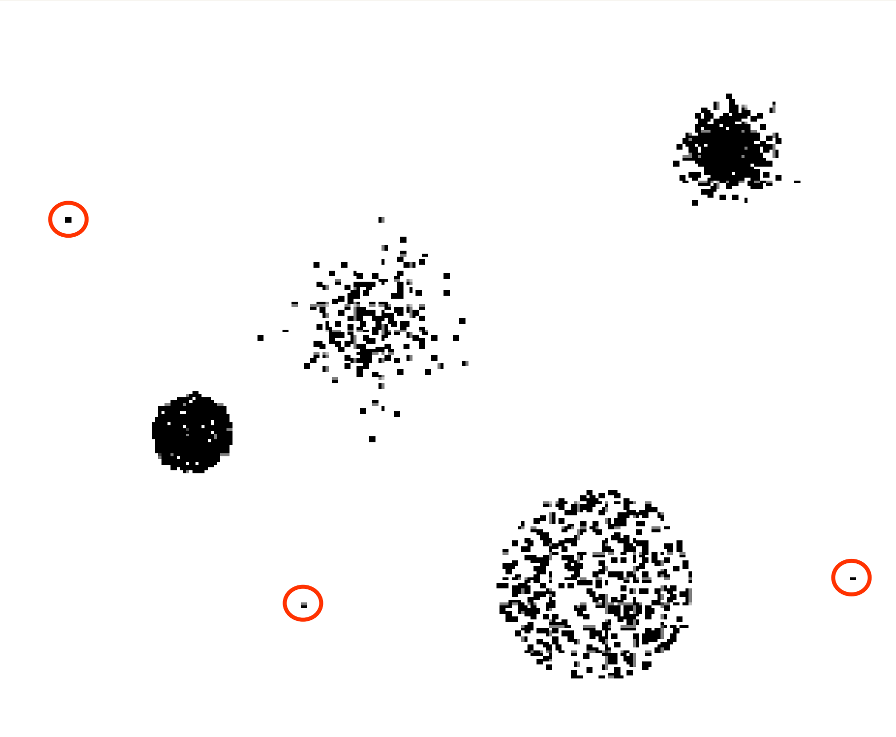
Pandas functions for handling missing data

- **`dropna`**: filter axis labels based on whether values for each label have missing data https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html
- **`fillna`**: Fill in missing data with a value or an interpolation method such as ffill or bfill https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html
- **`isnull`**: Return boolean values indicating which values are missing/NA
- **`notnull`**: Negation of isnull

OUTLIERS AND NOISE

Outliers

- Outliers are values that are thought to be out of range
- Outliers can influence the results of an analysis
- Not all outliers are mistakes or wrong
- Approaches for dealing with outliers:
  - *Do nothing*
  - *Remove outliers*
  - *Run analysis with and without outliers*
  - *Enforce upper and/or lower bounds*
  - *Use discretization*

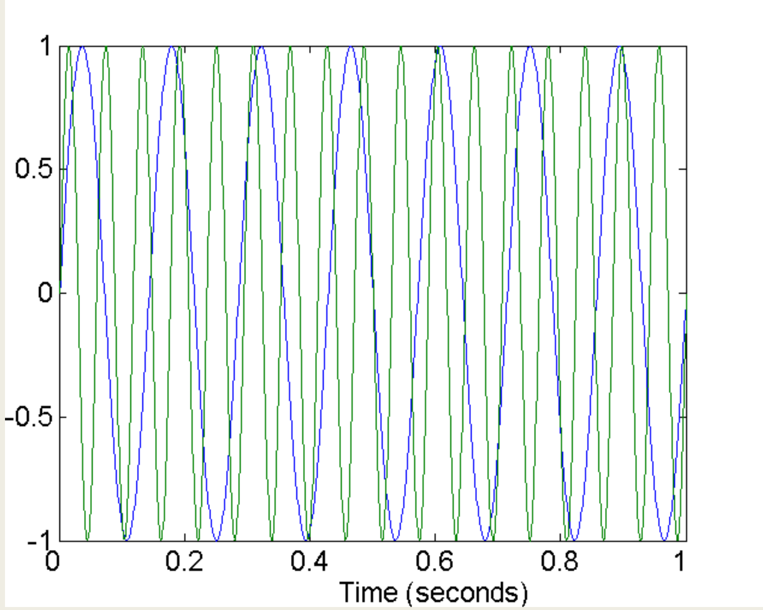


Outliers

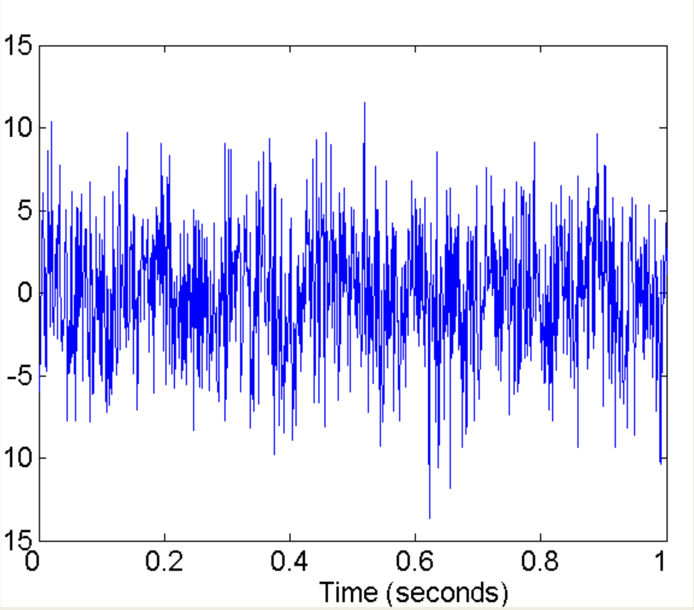
- You should remove outliers from a dataset when you have a sound reason to do so, such as when they represent:
  - Measurement errors
  - Data entry or processing errors
  - Poor sampling
  - Values that are outside of a known range
  - A questionable data point that you can recollect or verify
  - A clear error, irrelevance, or inconsistency with the rest of the data
- You should consider the following when deciding whether to remove an outlier:
  - If it appropriately reflects your target population, subject-area, research question, and research methodology
  - If there was anything unusual happen while measuring these observations
- You should document the excluded data points and explain your reasoning.
- Some outliers represent natural variations in the population and should be left as is in your dataset.
  - These are called true outliers.
- Try Bootstrapping techniques to see how the outliers affect your results
- Make sure you understand the underlying cause of the outliers!

Noise

- Noise refers to modification of original values
  - *Examples: distortion of a person's voice when talking on a poor phone connection and "snow" on television screen*
- Removal of noise is often problem domain dependent → Understand the domain



Two Sine Waves



Two Sine Waves + Noise

DATA TYPES

Changing Data Types:

Types of variables

Categorical

- Nominal (no natural ordering, e.g., hair color)
- Ordinal (ordered, e.g. grade assigned)

Numeric

- Continuous (e.g., blood pressure, height)
- Discrete (e.g., # of children, # of defects)

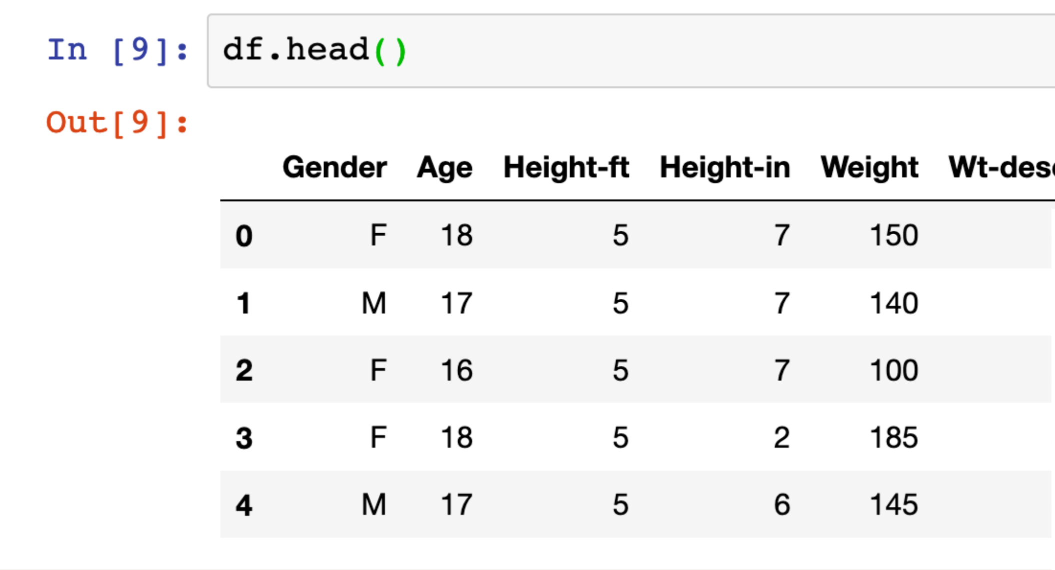
Other

- Dates & Times (technically also continuous)

Why does variable type matter?

- Most ML models expect numeric input
- It is important to consider the consequences when a model mistakes the variable type
- We especially need to consider how to handle categorical variables

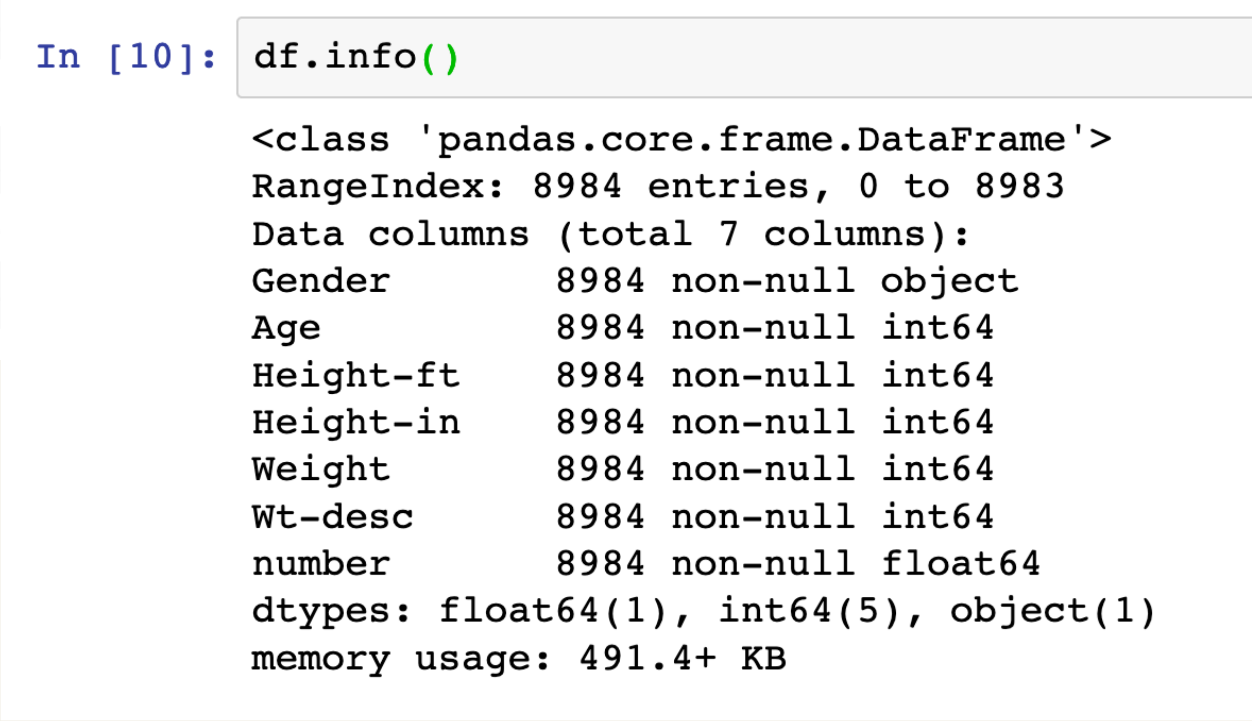
Checking Variable Type in Pandas



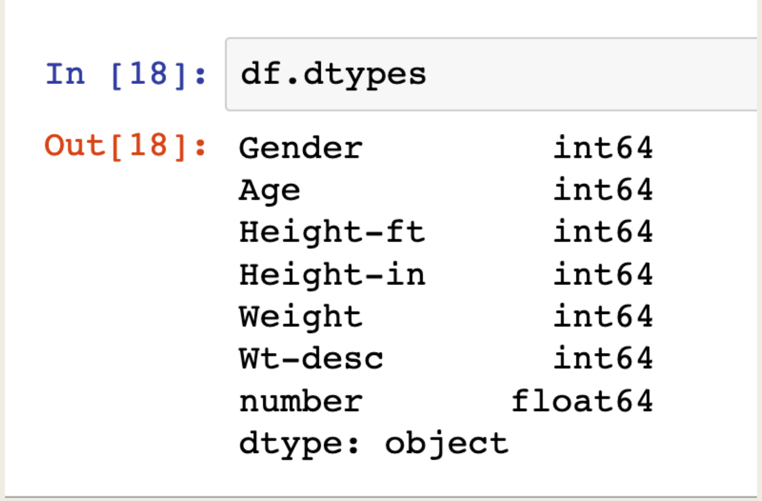
```
In [9]: df.head()
```

Out[9]:

	Gender	Age	Height-ft	Height-in	Weight	Wt-desc	number
0	F	18	5	7	150		
1	M	17	5	7	140		
2	F	16	5	7	100		
3	F	18	5	2	185		
4	M	17	5	6	145		



```
In [10]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8984 entries, 0 to 8983
Data columns (total 7 columns):
Gender       8984 non-null object
Age          8984 non-null int64
Height-ft    8984 non-null int64
Height-in    8984 non-null int64
Weight       8984 non-null int64
Wt-desc      8984 non-null int64
number       8984 non-null float64
dtypes: float64(1), int64(5), object(1)
memory usage: 491.4+ KB
```



```
In [18]: df.dtypes

Out[18]: Gender         int64
         Age            int64
         Height-ft      int64
         Height-in      int64
         Weight         int64
         Wt-desc        int64
         number       float64
         dtype: object
```

Changing Data Type in Pandas

- Specify `dtype` option when reading in data
- Use functions to cast objects to another `dtype`
  - `df.astype()`
  - `pd.to_datetime()`
  - `pd.to_numeric()`

```
import pandas as pd

df = pd.DataFrame({'col1': [1, 2, 3], 'col2': ['a', 'b', 'c']})

# Change the datatype of 'col1' to float
df['col1'] = df['col1'].astype(float)

print(df.dtypes)
```

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.astype.html

TRANSFORMING VARIABLES

Encoding Categorical Data

- Replace Values with a Number
  - {M, F} --> {1, 2}
  - {red, yellow, blue} --> {1, 2, 3}
  - {cold, medium, hot} --> {-1, 0, 1}
    (ordinal variables should keep ordering)

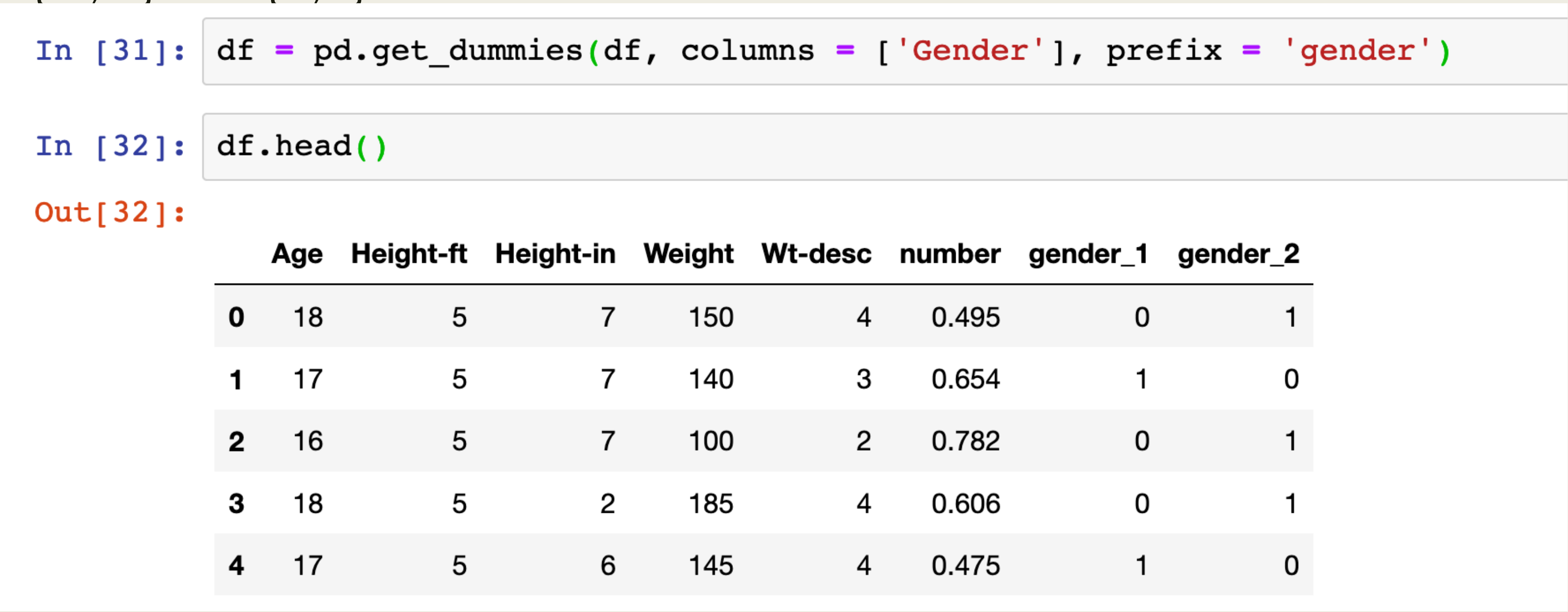
```
#Possible way to do this in Python

replace_map = {'var_name': {'cat1':1, 'cat2':2, 'cat3':3}}
df.replace(replace_map, inplace=True)
```

Why might this method of encoding be problematic?

Encoding Categorical Data

- One-hot encoding (dummy variables)
  - {M, F} --> {0,1}

```
#Possible
# df is a

pd.get_du
```

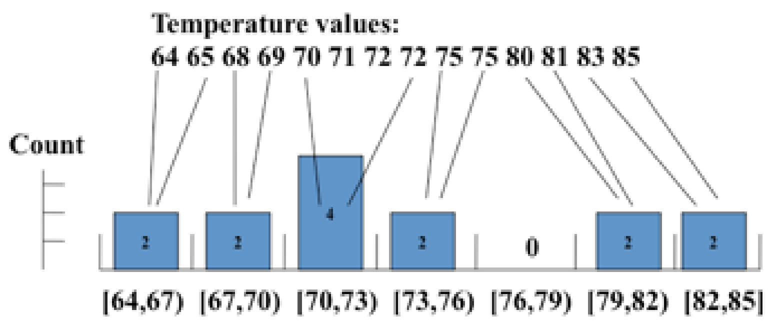


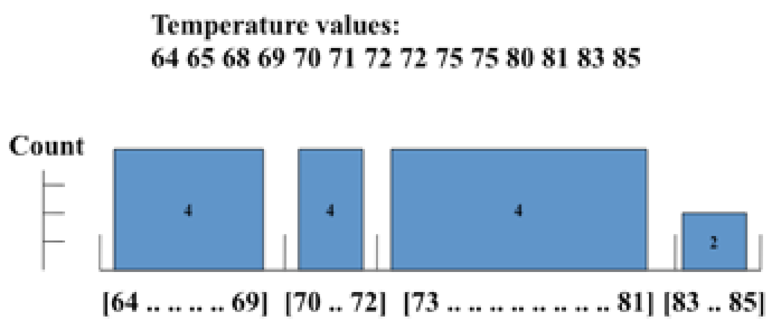
```
In [31]: df = pd.get_dummies(df, columns = ['Gender'], prefix = 'gender')

In [32]: df.head()

Out[32]:
```

	Age	Height-ft	Height-in	Weight	Wt-desc	number	gender_1	gender_2
0	18	5	7	150	4	0.495	0	1
1	17	5	7	140	3	0.654	1	0
2	16	5	7	100	2	0.782	0	1
3	18	5	2	185	4	0.606	0	1
4	17	5	6	145	4	0.475	1	0

Other common transformation tasks

- Combining categories
  - *Helpful when some categories have small sample sizes*
- Discretizing / binning numeric values





- Standardizing or normalizing numeric values
  - *Some ML models need features to be on the same scale*
    - Standardization: new_value = (current_value – mean) / stdev
    - Normalization: new_value = (current_value – min) / (max – min)

PROBLEM FEATURES

Useless Features

- Features with little or no variability
    - All (or almost all) cases have the same value
    - Ok to remove this feature

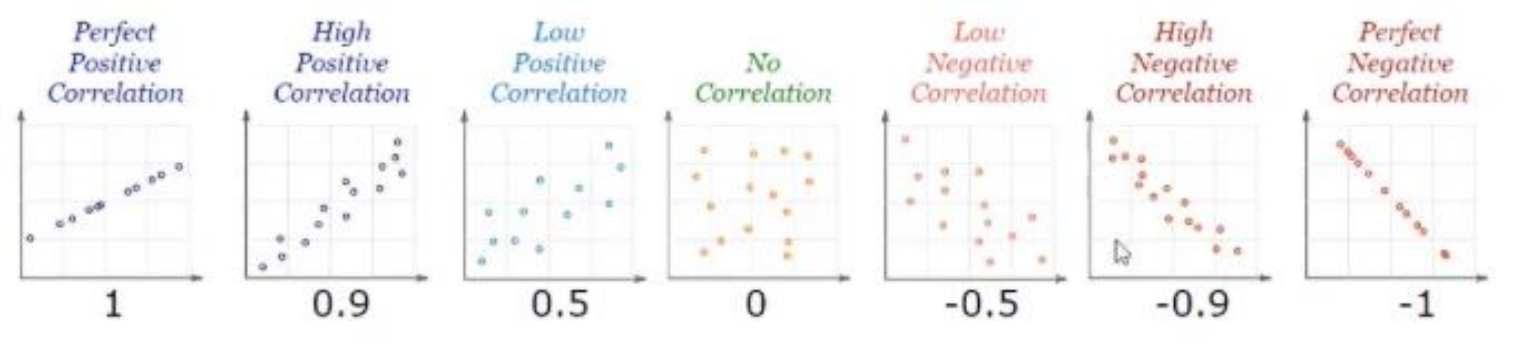
- Features with too much variability
    - All (or almost all) cases have a different *non-numeric* value (such as id/key)
    - Ok to remove or exclude from analysis
    - In longitudinal analysis, id identifier will be necessary

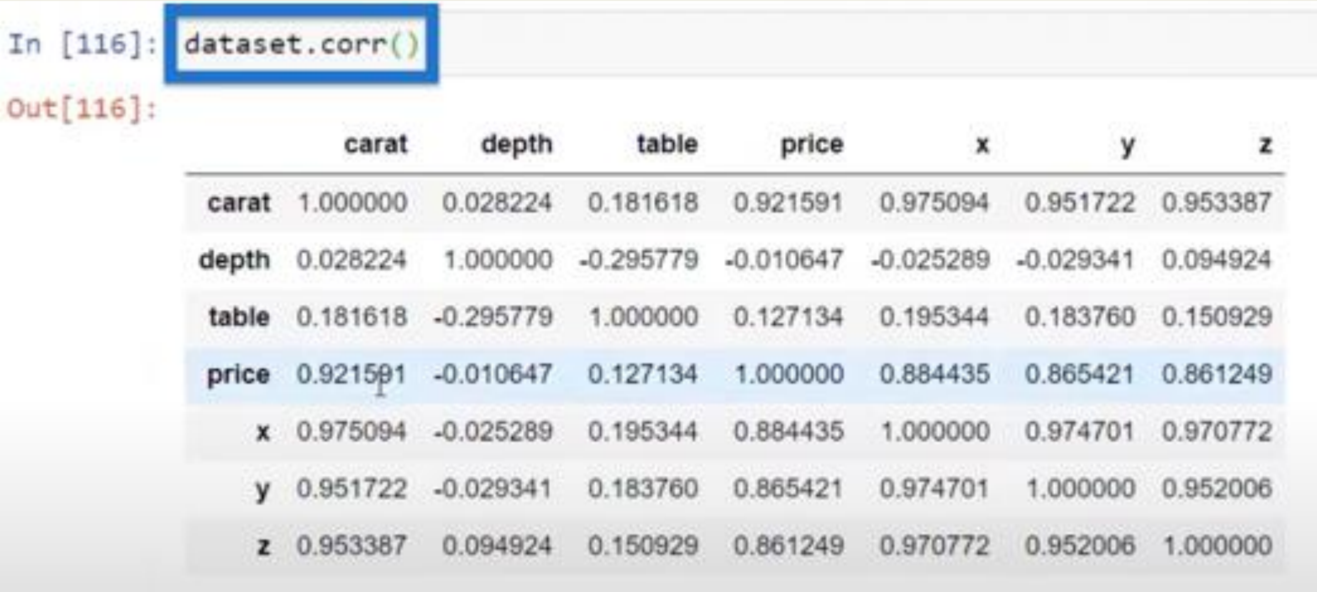
Dangerous Features

- Features that are highly correlated with another feature
    - Features might be redundant and only one is needed
    - This is known as collinearity and can be problematic for interpretation
    - Penalization methods (e.g., ridge regression) can also be used

- Features that are highly correlated with the target
    - Feature might be a synonym with target (data leak) and will lead to over fitting

Correlation



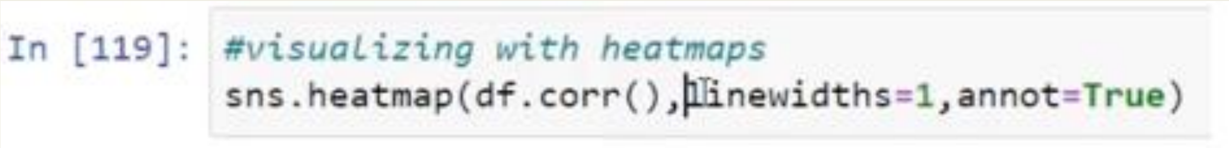


```
In [116]: dataset.corr()
Out[116]:
```

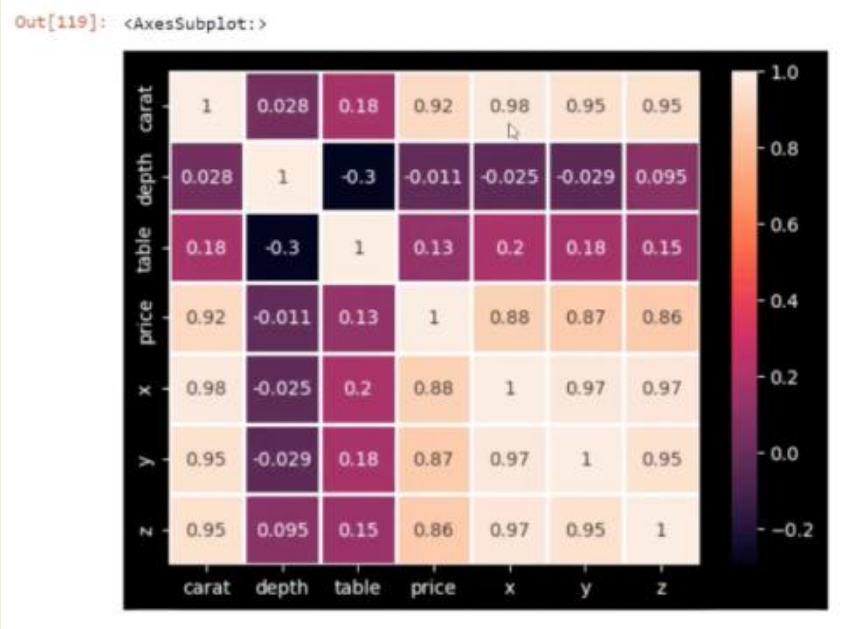
	carat	depth	table	price	x	y	z
carat	1.000000	0.028224	0.181618	0.921591	0.975094	0.951722	0.953387
depth	0.028224	1.000000	-0.295779	-0.010647	-0.025289	-0.029341	0.094924
table	0.181618	-0.295779	1.000000	0.127134	0.195344	0.183760	0.150929
price	0.921591	-0.010647	0.127134	1.000000	0.884435	0.865421	0.861249
x	0.975094	-0.025289	0.195344	0.884435	1.000000	0.974701	0.970772
y	0.951722	-0.029341	0.183760	0.865421	0.974701	1.000000	0.952006
z	0.953387	0.094924	0.150929	0.861249	0.970772	0.952006	1.000000

Visualizing Correlation

- Use seaborn's heatmap function



```
In [119]: #visualizing with heatmaps
          sns.heatmap(df.corr(),linewidths=1,annot=True)
```



```
Out[119]: <AxesSubplot:>
```

	carat	depth	table	price	x	y	z
carat	1	0.028	0.18	0.92	0.98	0.95	0.95
depth	0.028	1	-0.3	-0.011	-0.025	-0.029	0.095
table	0.18	-0.3	1	0.13	0.2	0.18	0.15
price	0.92	-0.011	0.13	1	0.88	0.87	0.86
x	0.98	-0.025	0.2	0.88	1	0.97	0.97
y	0.95	-0.029	0.18	0.87	0.97	1	0.95
z	0.95	0.095	0.15	0.86	0.97	0.95	1

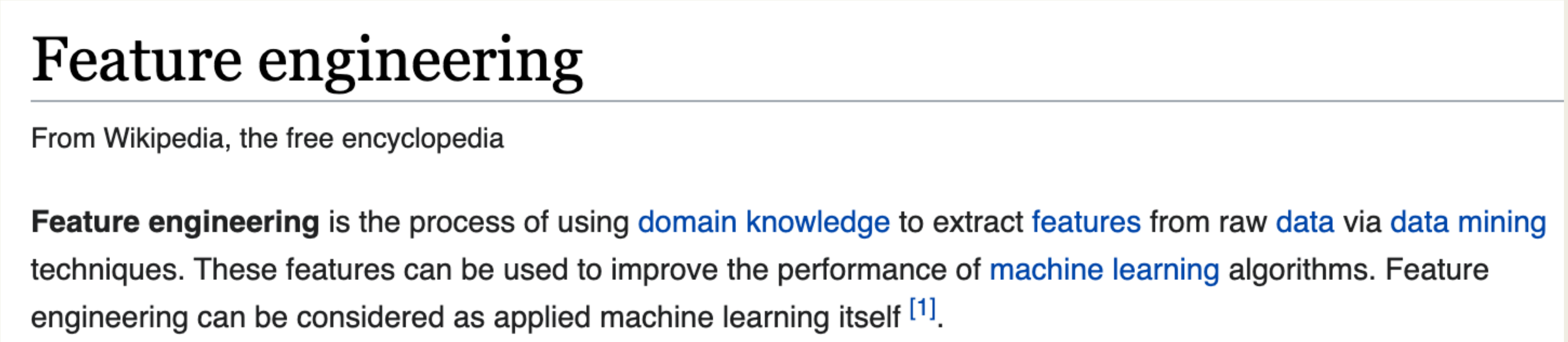
Output Class Skew

- Important output class may be rare
  - *Consider nuclear reactor data – Meltdowns vs non-meltdown prediction*
- Undersampling or Oversampling
  - *Undersampling – if you have 100,000 instances and only 1,000 in minority class, use a high percentage of the minority class and sample from the majority class until you get your desired distribution for training (50/50?)*
  - *Oversampling – make duplicates of the minority class and add it to the training data to get a better distribution (might cause memorization)*
    - Could add copies with some jitter (be careful!)
  - *SMOTE (Synthetic Minority Oversampling Technique)*
    - pip install imbalanced-learn

```
over = SMOTE(sampling_strategy=0.1)
under = RandomUnderSampler(sampling_strategy=0.5)
...
steps = [('o', over), ('u', under)]
pipeline = Pipeline(steps=steps)
...
# transform the dataset
X, y = pipeline.fit_resample(X, y)
```

- Change your loss function to weight the minority class more heavily
- Use Precision/Recall/F1 or ROC curve rather than just accuracy

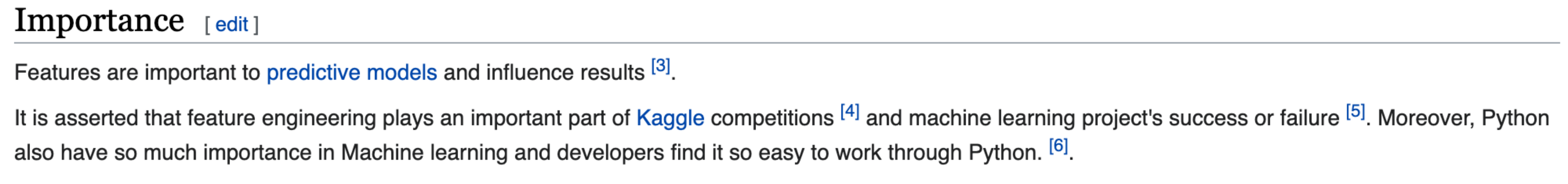
FEATURE ENGINEERING



Feature engineering

From Wikipedia, the free encyclopedia

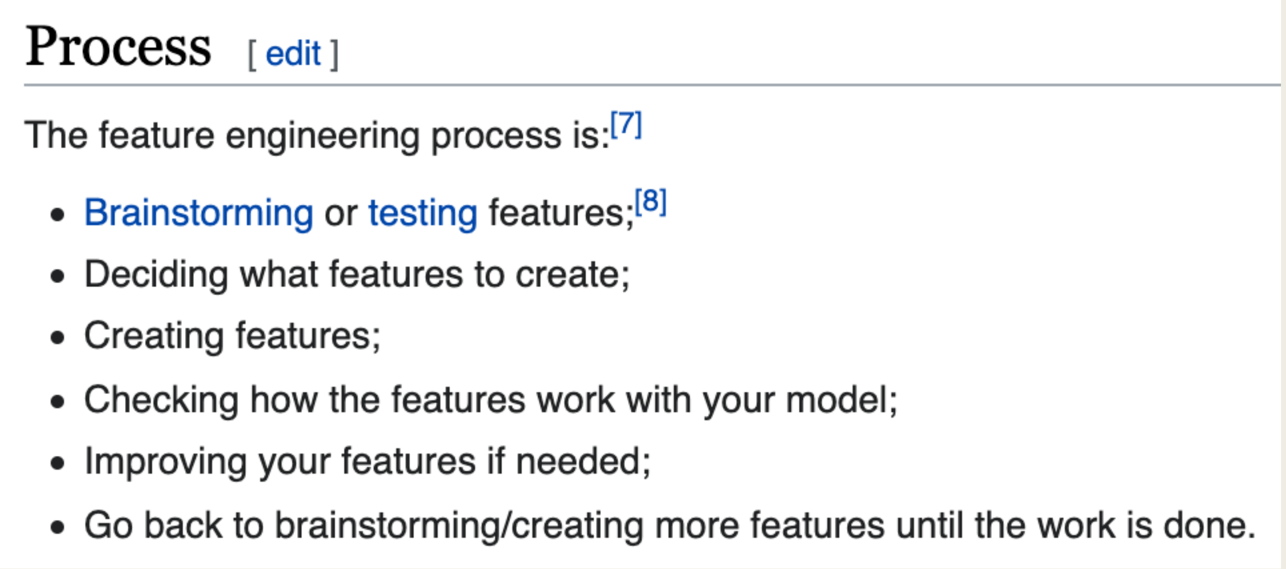
Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself [1].



Importance [edit]

Features are important to predictive models and influence results [3].

It is asserted that feature engineering plays an important part of Kaggle competitions [4] and machine learning project's success or failure [5]. Moreover, Python also have so much importance in Machine learning and developers find it so easy to work through Python. [6].

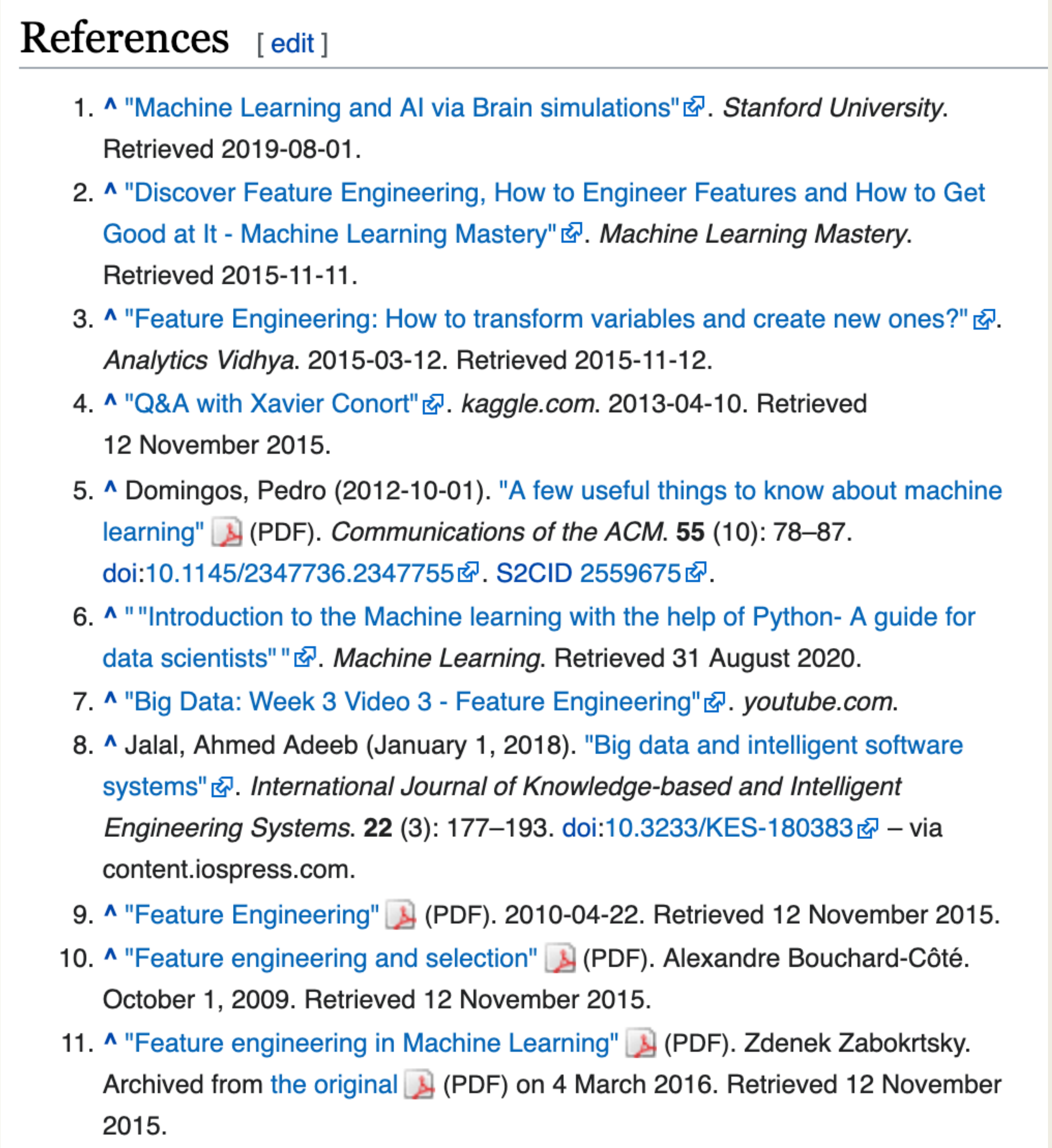


Process [edit]

The feature engineering process is:[7]

- Brainstorming or testing features;[8]
- Deciding what features to create;
- Creating features;
- Checking how the features work with your model;
- Improving your features if needed;
- Go back to brainstorming/creating more features until the work is done.

https://en.wikipedia.org/wiki/Feature_engineering

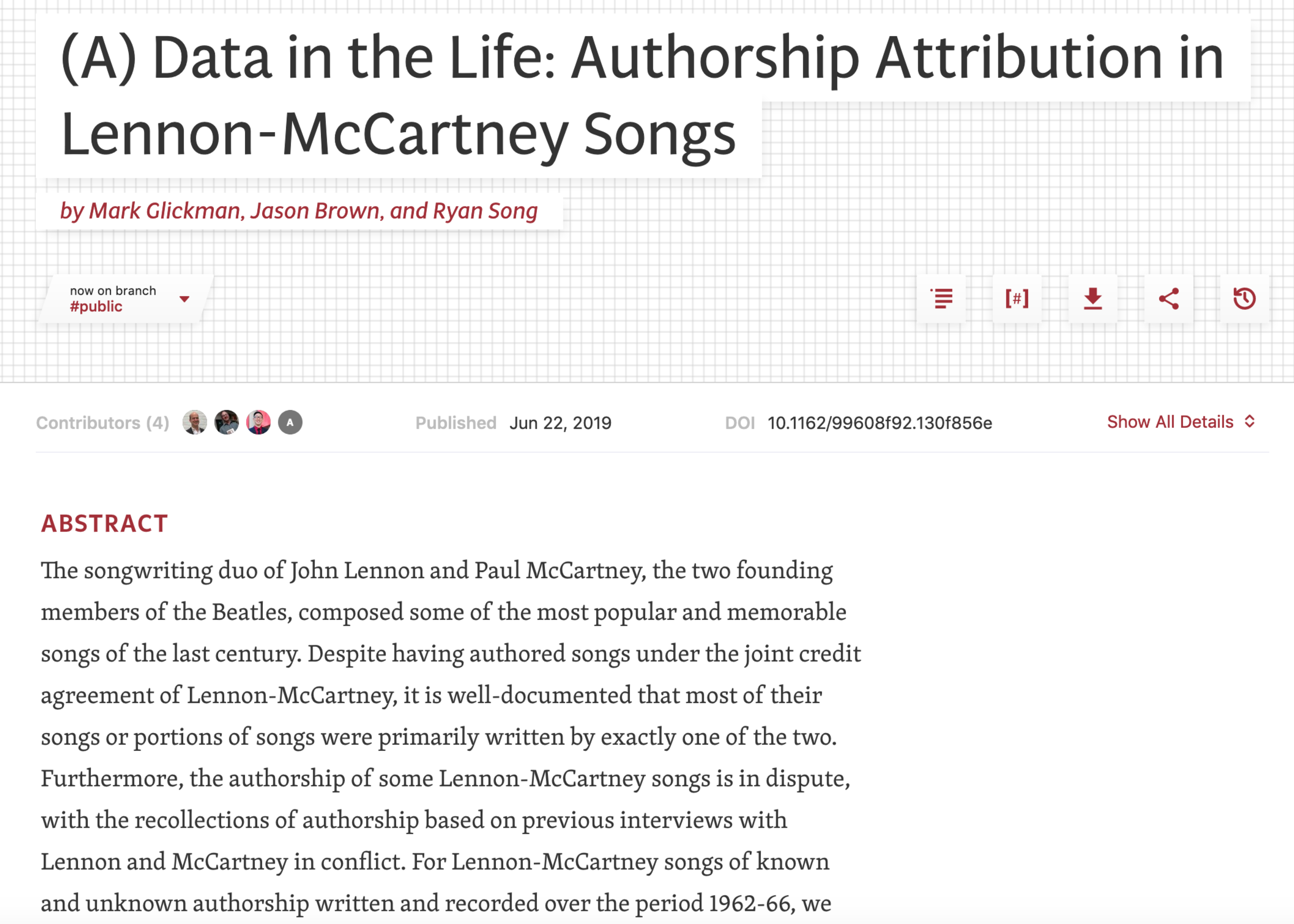


References [edit]

1. ^ "Machine Learning and AI via Brain simulations". *Stanford University*. Retrieved 2019-08-01.
2. ^ "Discover Feature Engineering, How to Engineer Features and How to Get Good at It - Machine Learning Mastery". *Machine Learning Mastery*. Retrieved 2015-11-11.
3. ^ "Feature Engineering: How to transform variables and create new ones?". *Analytics Vidhya*. 2015-03-12. Retrieved 2015-11-12.
4. ^ "Q&A with Xavier Conort". *kaggle.com*. 2013-04-10. Retrieved 12 November 2015.
5. ^ Domingos, Pedro (2012-10-01). "A few useful things to know about machine learning" (PDF). *Communications of the ACM*. **55** (10): 78–87. doi:10.1145/2347736.2347755. S2CID 2559675.
6. ^ " "Introduction to the Machine learning with the help of Python- A guide for data scientists" ". *Machine Learning*. Retrieved 31 August 2020.
7. ^ "Big Data: Week 3 Video 3 - Feature Engineering". *youtube.com*.
8. ^ Jalal, Ahmed Adeeb (January 1, 2018). "Big data and intelligent software systems". *International Journal of Knowledge-based and Intelligent Engineering Systems*. **22** (3): 177–193. doi:10.3233/KES-180383 – via content.iospress.com.
9. ^ "Feature Engineering" (PDF). 2010-04-22. Retrieved 12 November 2015.
10. ^ "Feature engineering and selection" (PDF). Alexandre Bouchard-Côté. October 1, 2009. Retrieved 12 November 2015.
11. ^ "Feature engineering in Machine Learning" (PDF). Zdenek Zabokrtsky. Archived from the original (PDF) on 4 March 2016. Retrieved 12 November 2015.

Feature Engineering

- Transform existing features (e.g., use area code to approximate location of home)
- Create new features (e.g., compute time to event from dates in dataset)
- Extract numeric data from non-numeric sources
  - *Images*
  - *Video*
  - *Text*
  - *Audio*



(A) Data in the Life: Authorship Attribution in Lennon-McCartney Songs

by Mark Glickman, Jason Brown, and Ryan Song

Contributors (4) A

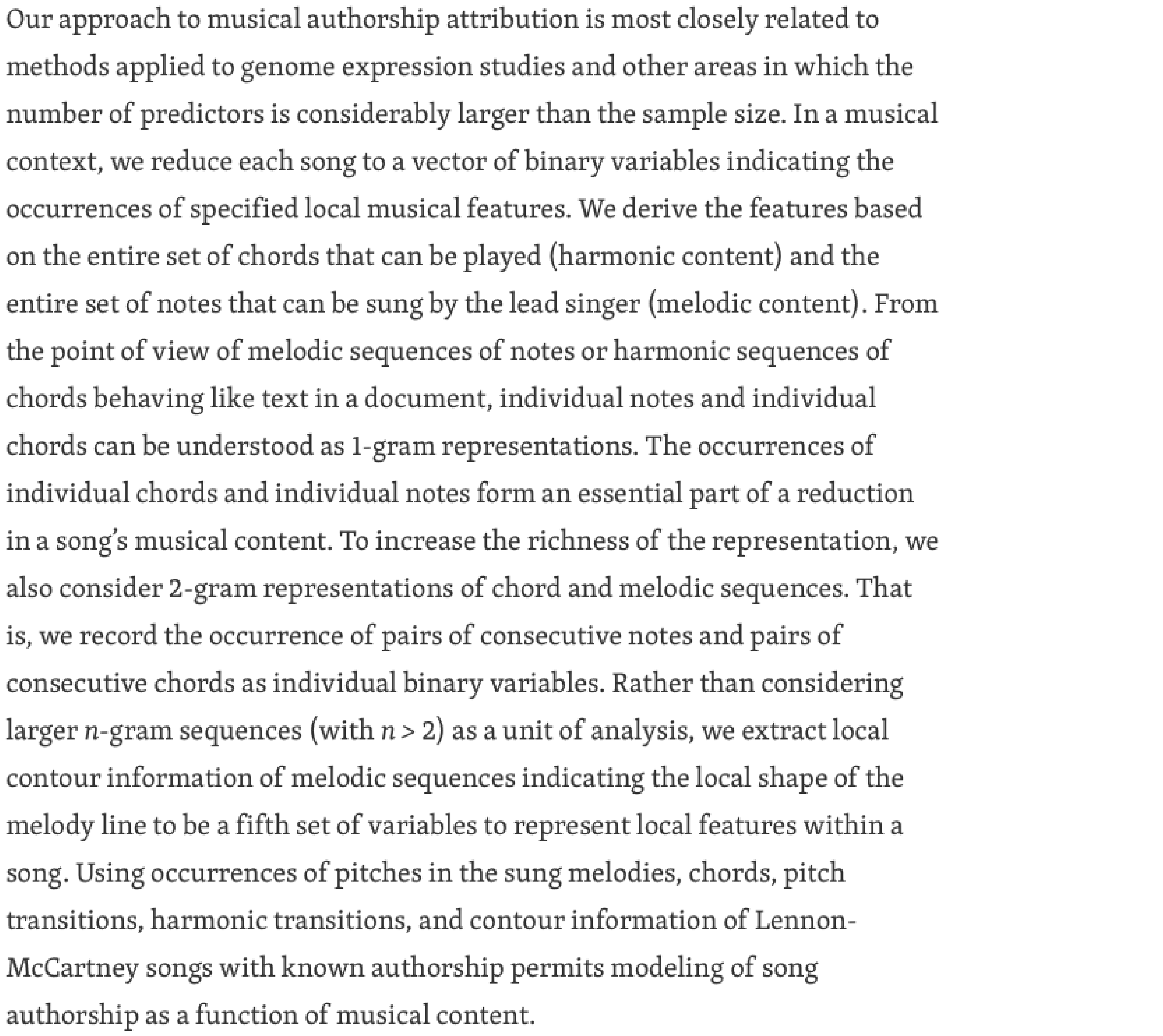
Published Jun 22, 2019

DOI 10.1162/99608f92.130f856e

ABSTRACT

The songwriting duo of John Lennon and Paul McCartney, the two founding members of the Beatles, composed some of the most popular and memorable songs of the last century. Despite having authored songs under the joint credit agreement of Lennon-McCartney, it is well-documented that most of their songs or portions of songs were primarily written by exactly one of the two. Furthermore, the authorship of some Lennon-McCartney songs is in dispute, with the recollections of authorship based on previous interviews with Lennon and McCartney in conflict. For Lennon-McCartney songs of known and unknown authorship written and recorded over the period 1962-66, we

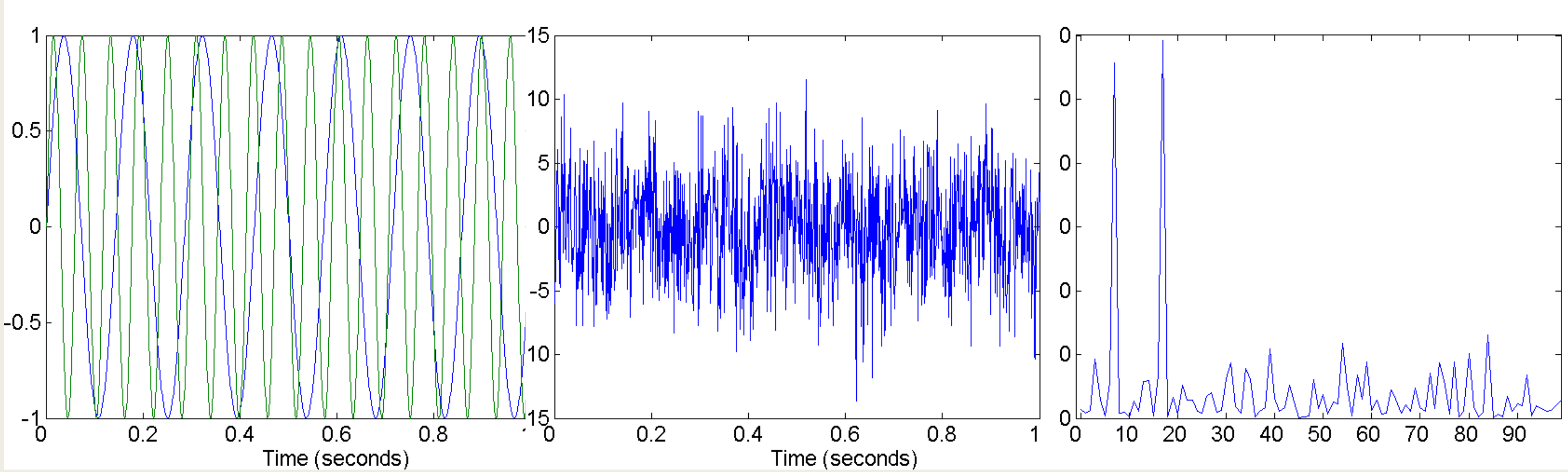
https://hdsr.mitpress.mit.edu/pub/xcq8a1v1



Our approach to musical authorship attribution is most closely related to methods applied to genome expression studies and other areas in which the number of predictors is considerably larger than the sample size. In a musical context, we reduce each song to a vector of binary variables indicating the occurrences of specified local musical features. We derive the features based on the entire set of chords that can be played (harmonic content) and the entire set of notes that can be sung by the lead singer (melodic content). From the point of view of melodic sequences of notes or harmonic sequences of chords behaving like text in a document, individual notes and individual chords can be understood as 1-gram representations. The occurrences of individual chords and individual notes form an essential part of a reduction in a song's musical content. To increase the richness of the representation, we also consider 2-gram representations of chord and melodic sequences. That is, we record the occurrence of pairs of consecutive notes and pairs of consecutive chords as individual binary variables. Rather than considering larger *n*-gram sequences (with $n > 2$) as a unit of analysis, we extract local contour information of melodic sequences indicating the local shape of the melody line to be a fifth set of variables to represent local features within a song. Using occurrences of pitches in the sung melodies, chords, pitch transitions, harmonic transitions, and contour information of Lennon-McCartney songs with known authorship permits modeling of song authorship as a function of musical content.

Mapping Data to a New Space

- Fourier Transform
- Wavelet Transform
- Problem domain dependent → Understand the domain



Two Sine Waves

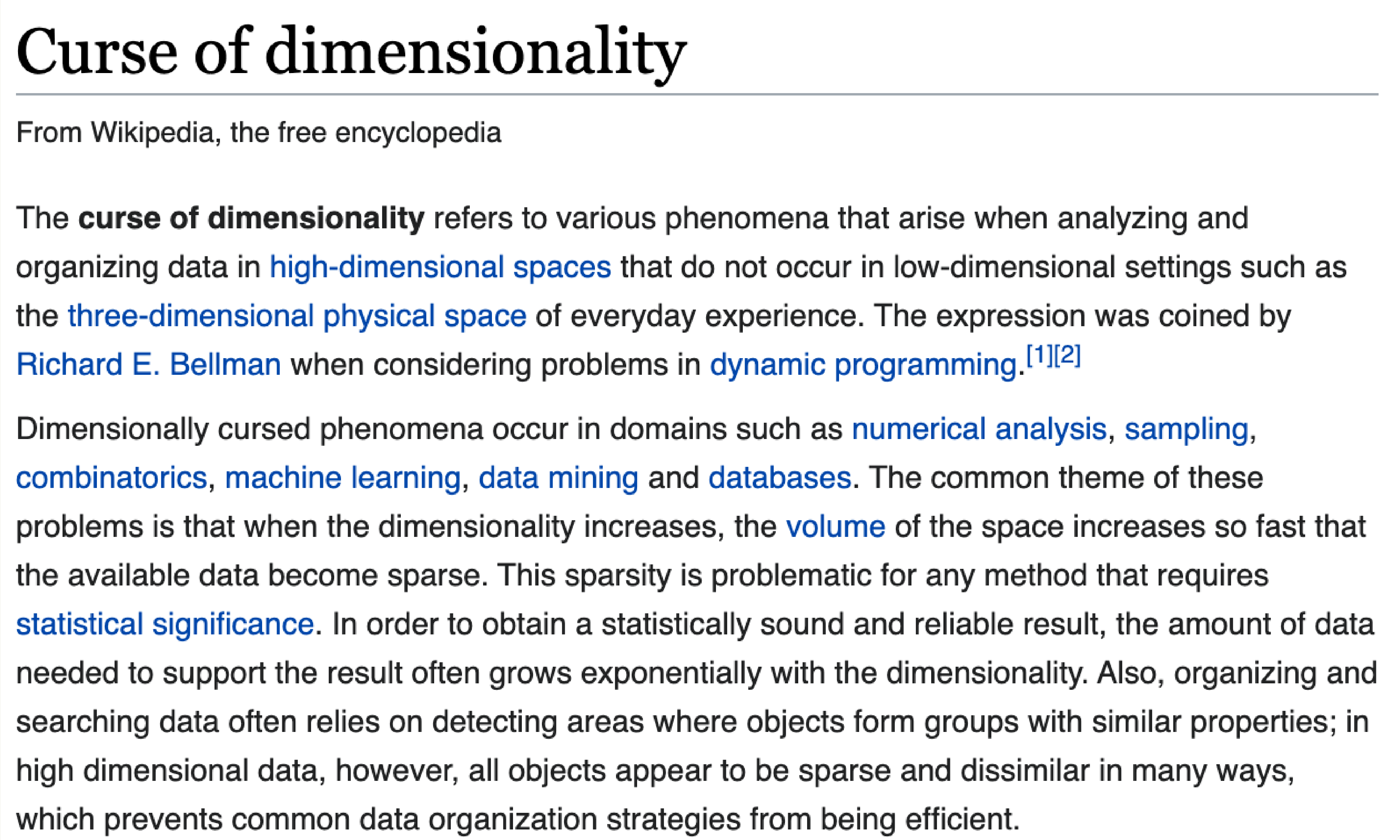
Two Sine Waves + Noise

Frequency

FEATURE SELECTION

The curse of Dimensionality

- Some machine learning problems will have A LOT of features (think thousands or even millions)
- This can make model training very slow (even using some of the big data solutions we've learned)
- It can also make it hard to find a good model (high dimensional spaces are sparse)
- Furthermore, it is extremely hard to visualize high dimensional problems
- The problem of having many many features is sometimes called the curse of dimensionality



Curse of dimensionality

From Wikipedia, the free encyclopedia

The **curse of dimensionality** refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience. The expression was coined by Richard E. Bellman when considering problems in dynamic programming.[1][2]

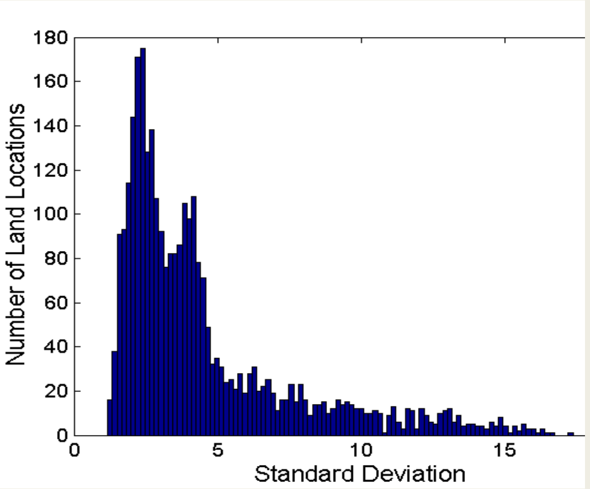
Dimensionally cursed phenomena occur in domains such as numerical analysis, sampling, combinatorics, machine learning, data mining and databases. The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality. Also, organizing and searching data often relies on detecting areas where objects form groups with similar properties; in high dimensional data, however, all objects appear to be sparse and dissimilar in many ways, which prevents common data organization strategies from being efficient.

Possible Remedies

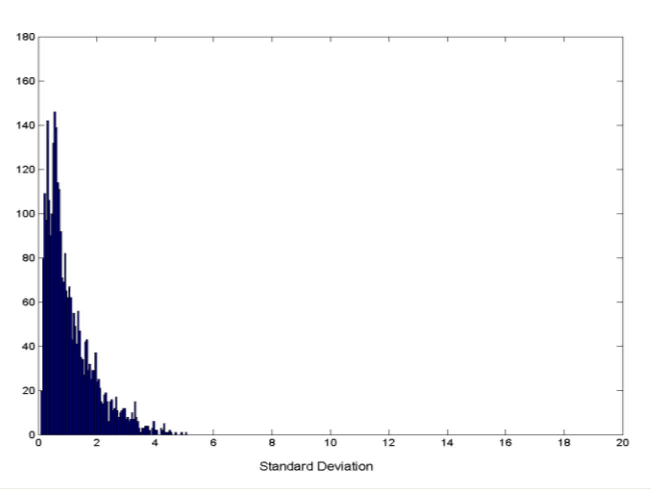
- Feature selection
  - *Filter-based methods (choose features based on some metric)*
  - *Wrapper-based methods*
    - Forward, backward, or stepwise selection
- Aggregation
- Dimension reduction
  - *Principal Component Analysis (PCA)*
  - *t-Distributed Stochastic Neighbor Embedding (t-SNE)*
  - *Multidimensional Scaling (MDS)*
  - *Uniform Manifold Approximation and Projection (UMAP)*

Aggregation

- Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - *Data reduction*
  - *Change of scale*
  - *More stability*



Standard Deviation of Average Monthly Precipitation



Standard Deviation of Average Yearly Precipitation

PCA

- Find a lower-dimensional hyperplane that preserves most of the variance in the data
- Project the data onto that hyperplane
- Consider the following example where we start with 2 dimensions and will project down to 1 dimension

PCA

Which hyperplane (or in this case line) would you choose the best represent the original data (i.e., has the most variance from the data)?

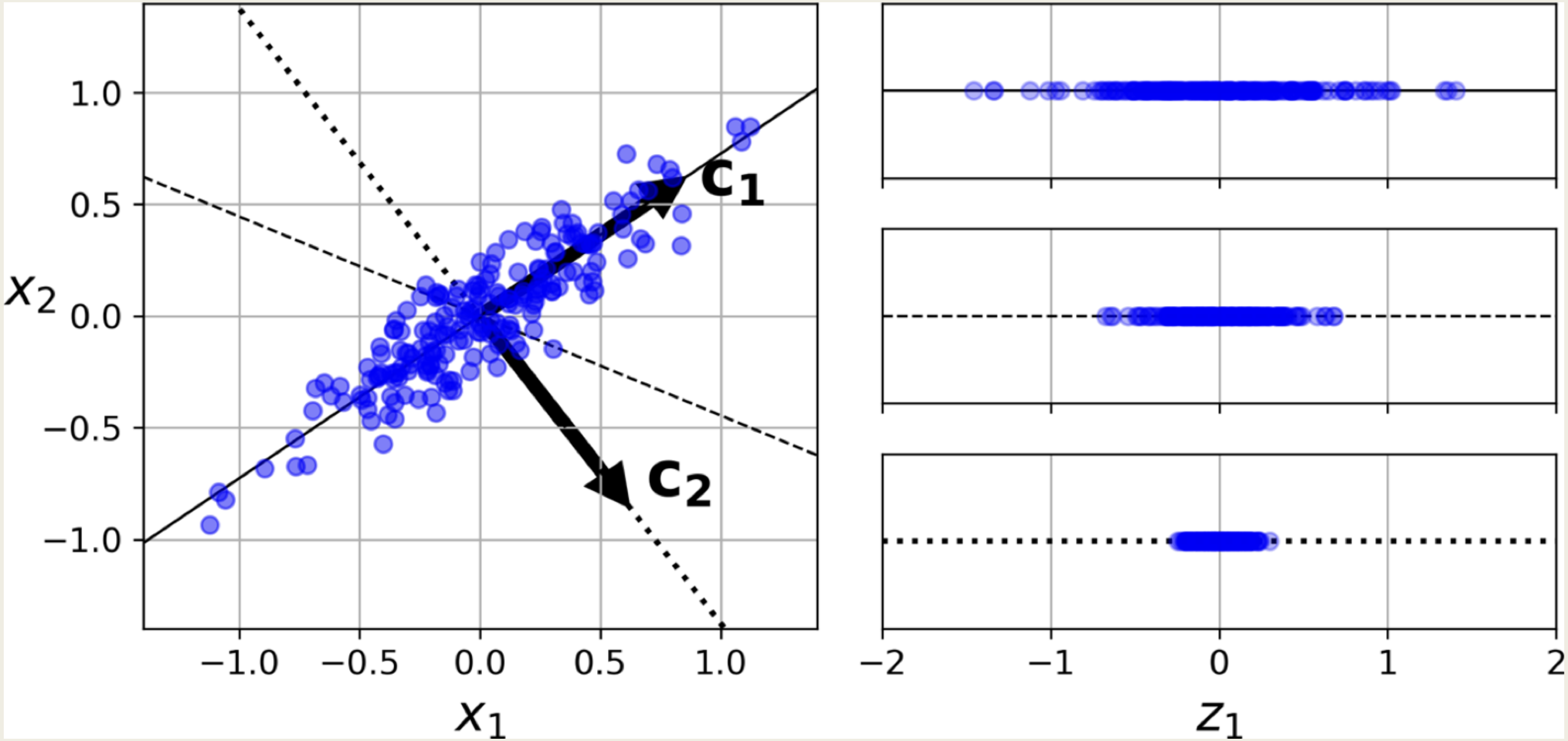


Image from "Hands-On Machine Learning with Scikit-Learn & TensorFlow" by Aurelien Geron

Finding Principal Components

- Principal components can be found by computing eigenvalues and eigenvectors of the covariance matrix of centered data
- They can also be found by computing the singular value decomposition of centered data
- When finding PCs, data should always be centered, and it should usually be scaled too

A few more PCA details (using eigenvalues and eigenvectors)

- Find the covariance matrix of the centered data, *S (d* x *d* matrix*)*
- Compute the eigenvalues and eigenvectors of *S*
- Order the eigenvalues from largest to smallest
- The percent of variation explained by the i^{th} PC is: $$\frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j}$$
- Select r dimensions based on desired variance explained
- Project onto r-dimensional hyperplane: $X_{proj} = XQ_r$

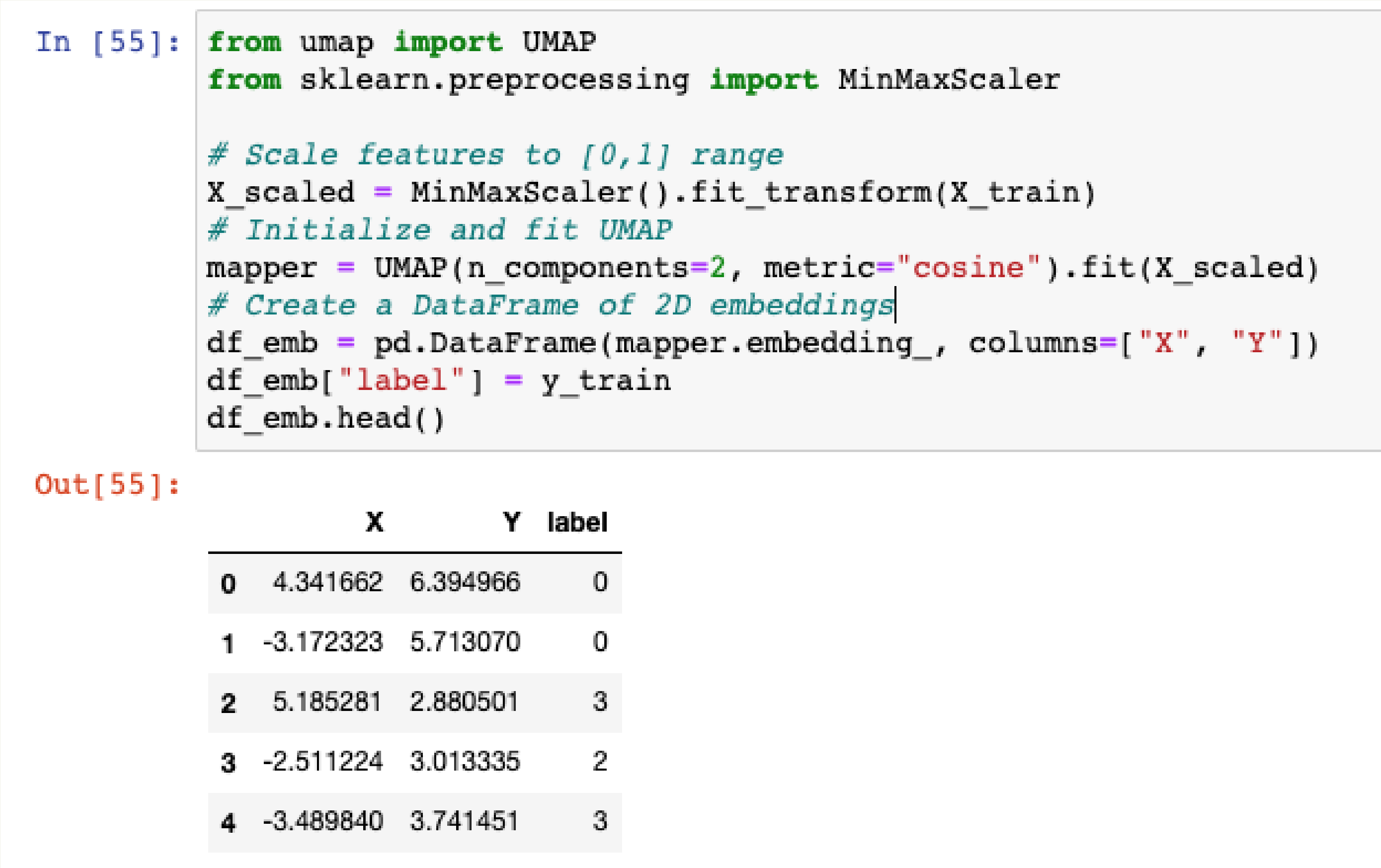
  (where Q_r is the matrix with the first *r* eigenvectors)

Or use sklearn

```
from sklearn.decomposition import PCA

pca = PCA(n_components = 2)
#OR
pca = PCA(n_components = .9)

# if n_components is between 0 and 1, it will be interpreted
# as the desired percent variation explained

pca.fit(X)
X_transform = pca.transform(X)

X_reduced = pca.inverse_transform(X)
```

You should normalize the data first.

UMAP

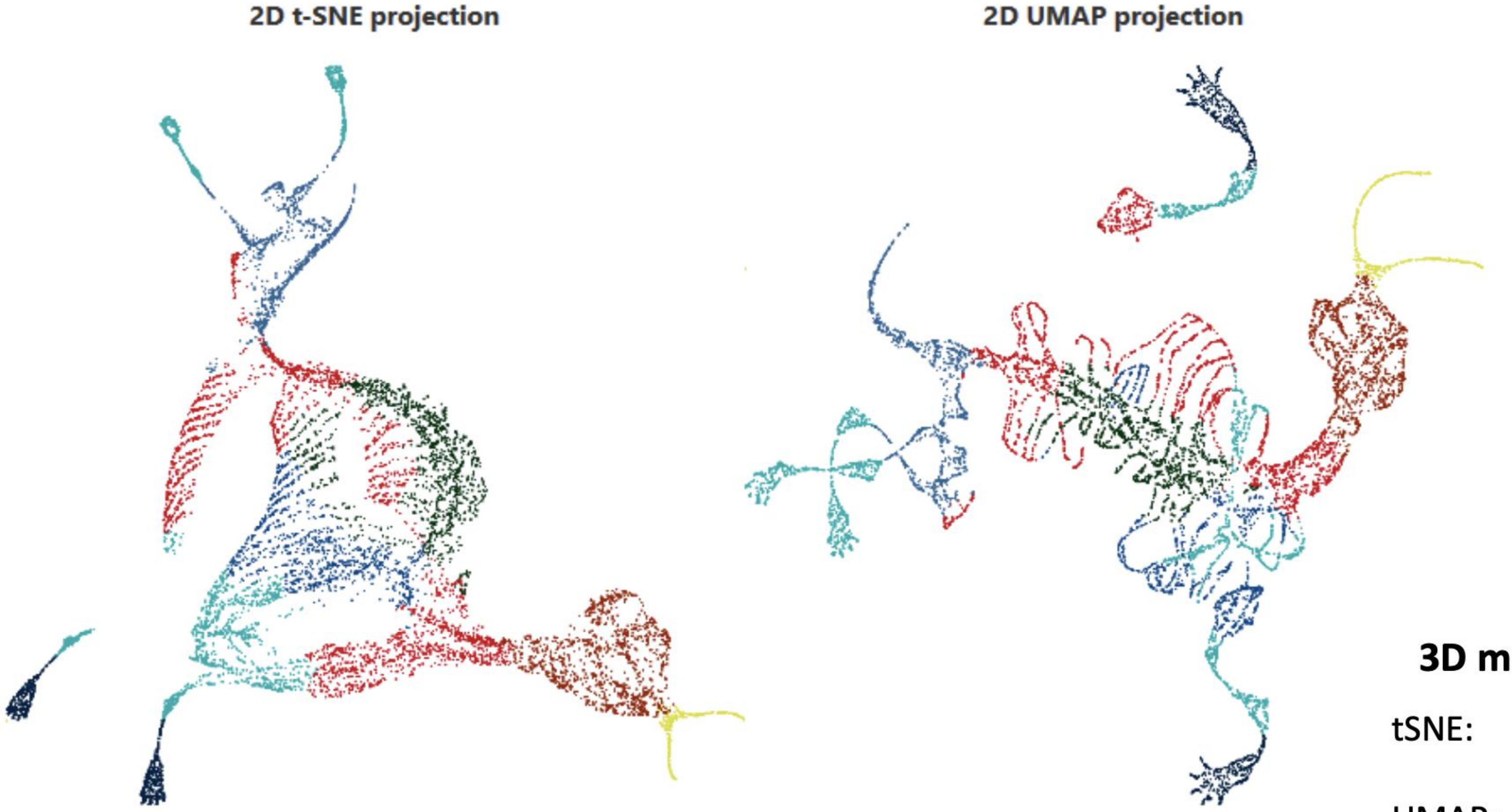


In [55]:

```
from umap import UMAP
from sklearn.preprocessing import MinMaxScaler

# Scale features to [0,1] range
X_scaled = MinMaxScaler().fit_transform(X_train)
# Initialize and fit UMAP
mapper = UMAP(n_components=2, metric="cosine").fit(X_scaled)
# Create a DataFrame of 2D embeddings
df_emb = pd.DataFrame(mapper.embedding_, columns=["X", "Y"])
df_emb["label"] = y_train
df_emb.head()
```

Out[55]:

	X	Y	label
0	4.341662	6.394966	0
1	-3.172323	5.713070	0
2	5.185281	2.880501	3
3	-2.511224	3.013335	2
4	-3.489840	3.741451	3



2D t-SNE projection

2D UMAP projection

https://pair-code.github.io/understanding-umap/

3D mammoth skeleton projected into 2D

tSNE:	Perplexity 2000	2h 5min
UMAP:	Nneigh 200, mindist 0.25,	3min

1

Know how your data was collected

2

Be thoughtful and transparent when cleaning and transforming

3

Consider how you can add value through the data collection and cleaning process

4

This part can be just as fun as modeling!

Take Home Message