

Data preparation for ML

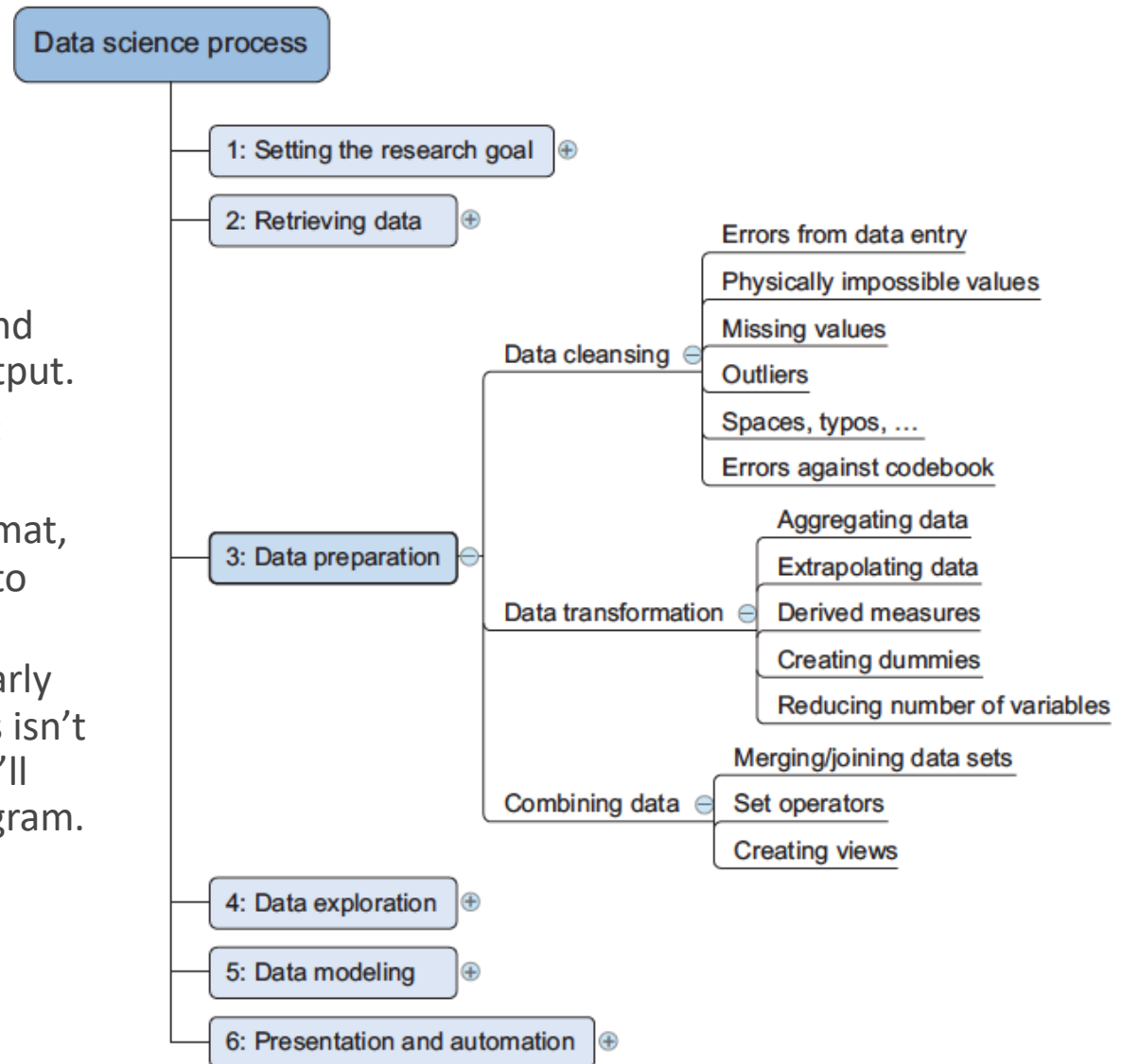
Parts of this lecture are based on slides from the Harvard Course CS109A Introduction to Data Science by Pavlos Protopapas, Kevin Rader and Chris Tanner, available at <https://github.com/Harvard-IACS/2019-CS109A>

Agenda

- Simple ML prediction task using K-nearest neighbors (KNN) algorithm
- Data cleaning
- Missing values
- Outliers
- Vectorization
- Data Transformation

Step 3: Cleansing, integrating, and transforming data

- Data cleaning is tremendously important because your models will perform better and you'll lose less time trying to fix strange output.
- It can't be mentioned nearly enough times: garbage-in equals garbage-out.
- Your model needs the data in a specific format, so data transformation will always come into play.
- It's a good habit to correct data errors as early on in the process as possible. However, this isn't always possible in a realistic setting, so you'll need to take corrective actions in your program.



Simple ML prediction task: Predict movie type

Movie title	# of kicks	# of kisses	Type of movie
<i>California Man</i>	3	104	Romance
<i>He's Not Really into Dudes</i>	2	100	Romance
<i>Beautiful Woman</i>	1	81	Romance
<i>Kevin Longblade</i>	101	10	Action
<i>Robo Slayer 3000</i>	99	5	Action
<i>Amped II</i>	98	2	Action
?	18	90	Unknown

K-nearest neighbors (KNN) algorithm

k-NN a simplistic and logical prediction method, that produces very competitive results.

- Step 1 – Choose the value of K i.e. the nearest data points. K can be any integer.
- Step 2 – For each point in the test data do the following –
 - 2.1 – Calculate the **similarity** between test data and each row of training
 - 2.2 – Now, based on the distance value, sort them in ascending order.
 - 2.3 – Next, it will choose the top K rows from the sorted array.
 - 2.4 – Now, it will assign a class to the test point based on most frequent class of these rows.

k -NN Similarity Measure: The Distance Metric

- Distance as a similarity metrics

Minkowski distance

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

If $q = 1$, then d is called Manhattan distance

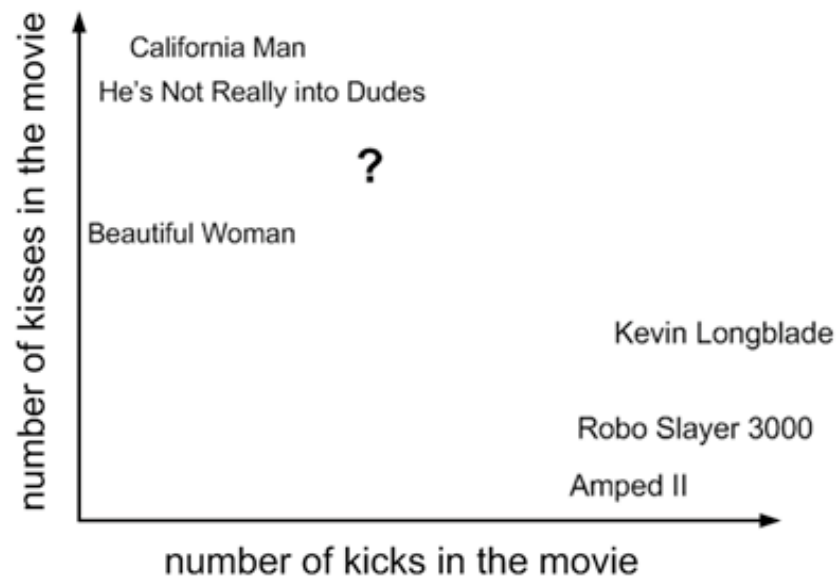
$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

If $q = 2$, then d is called Euclidean distance

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

Simple classification task: Predict movie type

- Distances between each movie and the unknown movie



Movie title	# of kicks	# of kisses	Type of movie
California Man	3	104	Romance
He's Not Really into Dudes	2	100	Romance
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
?	18	90	Unknown

Movie title	Distance to movie "?"
California Man	20.5
He's Not Really into Dudes	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9

- Let's assume $k=3$,
 - then closest movies are "He's Not Really into Dudes", "Beautiful Woman", and "California Man".
 - Because all three movies are romances, we forecast that the mystery movie is a romance movie.

Simple classification task: Predict movie type

- Input data need to be numerical
 - Most of the machine learning methods as input require numerical values
- All columns need to have a value
 - Most of the ML algorithms can't work with missing values

Movie title	# of kicks	# of kisses	Type of movie
<i>California Man</i>	3	104	Romance
<i>He's Not Really into Dudes</i>	2	100	Romance
<i>Beautiful Woman</i>	1	81	Romance
<i>Kevin Longblade</i>	101	10	Action
<i>Robo Slayer 3000</i>	99	5	Action
<i>Amped II</i>	98	2	Action
?	18	90	Unknown

Cleansing data

- Interpretation error
 - When you take the value in your data for granted
 - An example: data that represent a person's age is 150 years
- Inconsistencies error
 - Inconsistency between data sources or against your company's standardized values.
 - An example: putting "Female" in one table and "F" in another when they represent the same thing

Error description	Possible solution
<i>Errors pointing to false values within one data set</i>	
Mistakes during data entry	Manual overrules
Redundant white space	Use string functions
Impossible values	Manual overrules
Missing values	Remove observation or value
Outliers	Validate and, if erroneous, treat as missing value (remove or insert)
<i>Errors pointing to inconsistencies between data sets</i>	
Deviations from a code book	Match on keys or else use manual overrules
Different units of measurement	Recalculate
Different levels of aggregation	Bring to same level of measurement by aggregation or extrapolation

Data Cleaning Tools: OpenRefine

- Spreadsheet-like tool allowing data quality checking: reformatting, substitution, constraint checking etc.

11285 rows Extensions: Zemanta Freebase RDF CKW

Show as: rows records Show: 5 10 25 50 rows « first < previous 1 - 50 next »

		Capital or Rever	Directorate	Transaction Nur	Date	Service Area	Expenses Type	Amount	Supp
1.	Revenue		Community Wellbeing & Social Care	5105695746	05.04.2013	Youth & Community	Operational Equipment	120	REDACTE PERSON/
2.	Revenue		Community Wellbeing & Social Care	5105695746	05.04.2013	Youth & Community	Operational Equipment	80	REDACTE PERSON/
3.	Revenue		Community Wellbeing & Social Care	5105698650	24.04.2013	Leaseholds by LA	Accommodation Costs - Leaseholder Payments edit	695.89	REDACTE PERSON/
4.	Revenue		Community Wellbeing & Social Care	5105698650	24.04.2013	Leaseholds by LA	Accommodation Costs - Leaseholder Payments	695.89	REDACTE PERSON/
5.	Revenue		Community Wellbeing & Social Care	5105698650	24.04.2013	Leaseholds by LA	Accommodation Costs - Leaseholder Payments	695.89	REDACTE PERSON/
6.	Revenue		Community Wellbeing & Social Care	5105698650	24.04.2013	Leaseholds by LA	Accommodation Costs - Leaseholder Payments	695.89	REDACTE PERSON/
7.	Revenue		Community Wellbeing & Social Care	5105698650	24.04.2013	Leaseholds by LA	Accommodation Costs - Leaseholder Payments	695.89	REDACTE PERSON/
8.	Revenue		Chief Executive, Schools & Learning	5105698316	19.04.2013	L&A Commissioned Activity	Bought in Prof Services - Curriculum (Schools)	250	REDACTE PERSON/
9.	Revenue		Chief Executive, Schools & Learning	5105698318	19.04.2013	L&A Commissioned Activity	Bought in Prof Services - Curriculum (Schools)	710	REDACTE PERSON/
10.	Revenue		Economy & Environment	5105695879	05.04.2013	IW Biological Record	General Materials	220.2	REDACTE PERSON/
11.	Revenue		Chief Executive, Schools & Learning	5105698514	12.04.2013	Adult Services Training	Training and Conferences	150	REDACTE PERSON/
12.	Revenue		Community Wellbeing & Social Care	5105695832	10.04.2013	Short Breaks	Payments to Voluntary and Other Associations	1,260.00	REDACTE PERSON/
13.	Capital		Resources	5105696504	12.04.2013	Capital Receipts	External Design and Supervision Fees	400	REDACTE PERSON/
14.	Capital		Resources	5105696505	12.04.2013	Capital Receipts	External Design and Supervision Fees	1,350.00	REDACTE PERSON/
15.	Revenue		Economy & Environment	5105696707	12.04.2013	Schools Reorganisation	Security of Buildings	300	REDACTE PERSON/
16.	Revenue		Economy & Environment	5105696707	12.04.2013	Schools Reorganisation	Security of Buildings	300	REDACTE PERSON/

Dealing with missing values

- Missing values aren't necessarily wrong, but you still need to handle them separately
- **A lot of Machine Learning techniques can't handle missing values!!!**
 - So we need to deal with them!

Sources of Missingness

- Missing data can arise from various places in data:
 - A survey was conducted and values were just randomly missed when being entered in the computer.
 - A respondent chooses not to respond to a question like 'Have you ever done cocaine?'.
 - You decide to start collecting a new variable (like Mac vs. PC) partway through the data collection of a study.
 - You want to measure the speed of meteors, and some observations are just 'too quick' to be measured properly.
- The source of missing values in data can lead to the major types of missingness: Missing completely at random (MCAR), missing at random, missing not at random, and structurally missing.

Why use a Missingness Indicator Variable?

- Because the group of individuals with a missing entry may be systematically different than those with that variable measured.
- Treating them equivalently could lead to bias in quantifying relationships and underperform in prediction.
- For example:
 - Imagine a survey questions asks whether or not someone has ever recreationally used opioids, and some people chose not to respond.
 - Does the fact that they did not respond provide extra information?
 - Should we treat them equivalently as never-users?
 - This approach essentially creates a third group for this predictor: the “did not respond” group.

Naively handling missingness

- What are the simplest ways to handle missing data?
- Drop the observations that have any missing values.
 - Use `pd.DataFrame.dropna(axis=0)`
- Impute the mean/median (if quantitative) or most common class (if categorical) for all missing values.
 - Use `pd.DataFrame.fillna(value=x.mean())`

Types of Missingness

There are 3 major types of missingness to be concerned about:

1. **Missing Completely at Random (MCAR)** – the probability of missingness in a variable is the same for all units. Like randomly poking holes in a data set.
2. **Missing at Random (MAR)** - the probability of missingness in a variable depends only on available information (in other predictors).
3. **Missing Not at Random (MNAR)** - the probability of missingness depends on information that has not been recorded and this information also predicts the missing values.

Missing completely at random (MCAR)

- Missing Completely at Random is the best case scenario, and the easiest to handle:
- Examples:
 - A coin is flipped to determine whether an entry is removed. Or when values were just randomly missed when being entered in the computer.
- Effect if you ignore:
 - There is no effect on inferences.
- How to handle:
 - Lots of options, but best to impute (more on next slides).

Missing at random (MAR)

- Missing at random is still a case that can be handled.
- Example(s):
 - Men and women respond to the question "have you ever felt harassed at work?" at different rates (and may be harassed at different rates).
- Effect if you ignore:
 - Inferences are biased and predictions are usually worsened.
- How to handle:
 - Use the information in the other predictors to build a model and **impute** a value for the missing entry.
- Key:
 - We can fix any biases by modeling and imputing the missing values based on what is observed!

Missing Not at Random (MNAR)

- Missing Not at Random is the worst case scenario, and impossible to handle properly:
- Example(s):
 - Patients drop out of a study because they experience some really bad side effect that was not measured.
 - Or cheaters are less likely to respond when asked if you've ever cheated.
- Effect if you ignore:
 - There is no effect on inferences or predictions.
- How to handle:
 - You can 'improve' things by dealing with it like it is MAR, but you [likely] may never completely fix the bias.
 - And incorporating a **missingness indicator variable** may actually be the best approach (if it is in a predictor).






Imputation Methods

There are several different approaches to imputing missing values:

1. **Impute the mean or median** (quantitative) or most common class (categorical) for all missing values in a variable.
2. Create a new variable that is an **indicator of missingness**, and include it in any model to predict the response (also plug in zero or the mean in the actual variable).
3. **Hot deck imputation**: for each missing entry, randomly select an observed entry in the variable and plug it in.
4. **Model the imputation**: plug in predicted values (\hat{y}) from a model based on the other observed predictors.
5. **Model the imputation with uncertainty**: plug in predicted values plus randomness ($\hat{y} + \epsilon$) from a model based on the other observed predictors.








Schematic: imputation through modeling

- How do we use models to fill in missing data?

X	Y
	1
	?
	0.5
	0.1
	?
	10
	0.03








Schematic: imputation through modeling

- How do we use models to fill in missing data?

<u>X_train</u>	<u>Y_train</u>	<u>X_test</u>	<u>Y_pred</u>
	1		?
	0.5		
	0.1		
			?
	10		
	0.03		








Schematic: imputation through modeling

- How do we use models to fill in missing data? Using k -NN for $k = 2$?

X	Y
	1
	$? = (1 + 0.5) / 2$
	0.5
	0.1
	?
	10
	0.03










Schematic: imputation through modeling

- How do we use models to fill in missing data? Using k -NN for $k = 2$?

X	Y
	1
	? = $(1 + 0.5) / 2$
	0.5
	0.1
	? = $(0.1 + 10) / 2$
	10
	0.03

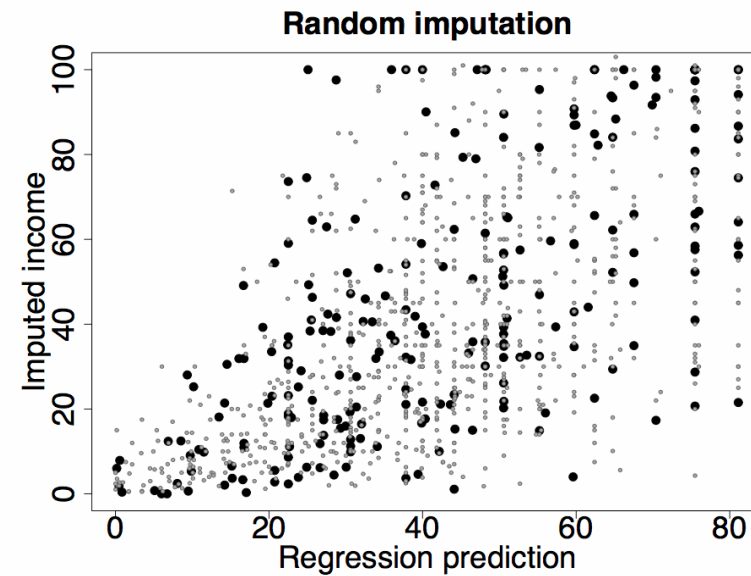
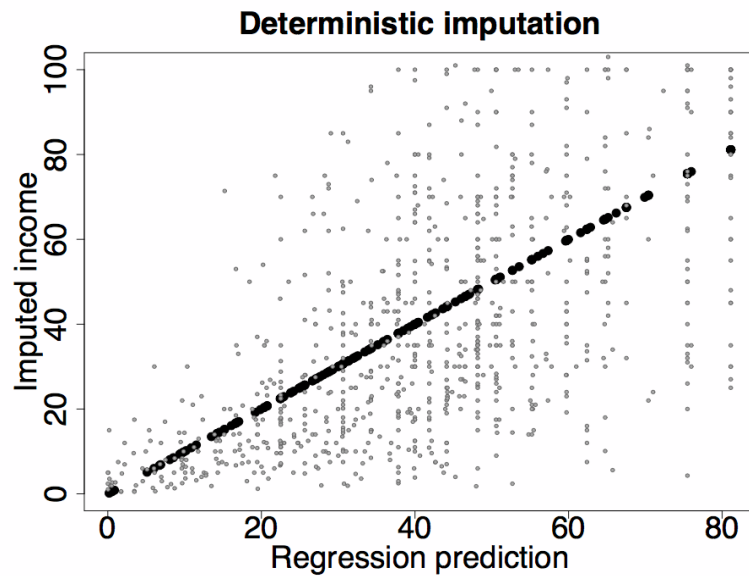
Schematic: imputation through modeling

- How do we use models to fill in missing data? Using linear regression?

X	Y
	1
	? = m  + b
	0.5
	0.1
	? = m  + b
	10
	0.03

- Where m and b are computed from the observations (rows) that do not have missingness (we should call them $b = \beta_0$ and $m = \beta_1$).

Imputation through modeling with uncertainty: an illustration



Imputation across multiple variables

- If only one variable has missing entries, life is easy. But what if all the predictor variables have a little bit of missingness (with some observations having multiple entries missing)? How can we handle that?
- It's an iterative process. Impute X_1 based on X_2, \dots, X_p . Then impute X_2 based on X_1 and X_3, \dots, X_p . And continue down the line.

Missingness Indicator Variable

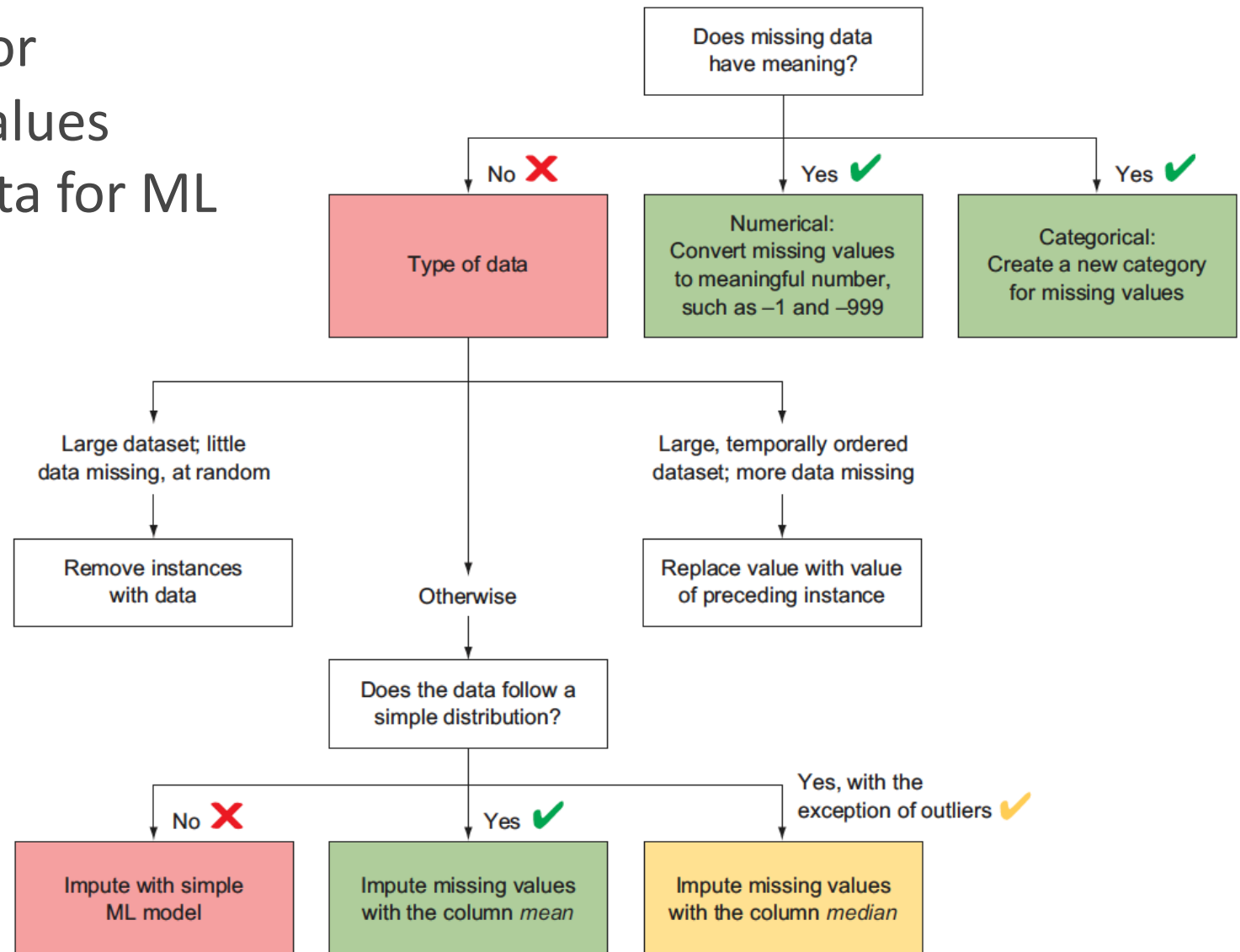
- One simple way to handle missingness in a variable, X_j , is to impute a value (like 0 or \bar{X}_j), then create a new variable, $X_{j,miss}$, that indicates this observation had a missing value. If X_j is categorical then just impute 0.
- Then include both $X_{j,miss}$ and X_j as predictors in any model.

X_1	X_2	X_1^*	X_2^*	$X_{1,miss}$	$X_{2,miss}$
10	.	10	0	0	1
5	1	5	1	0	0
21	0	21	0	0	0
15	0	15	0	0	0
16	.	16	0	0	1
.	.	0	0	1	1
21	1	21	1	0	0
12	0	12	0	0	0
.	1	0	1	1	0

Dealing With Missing Values

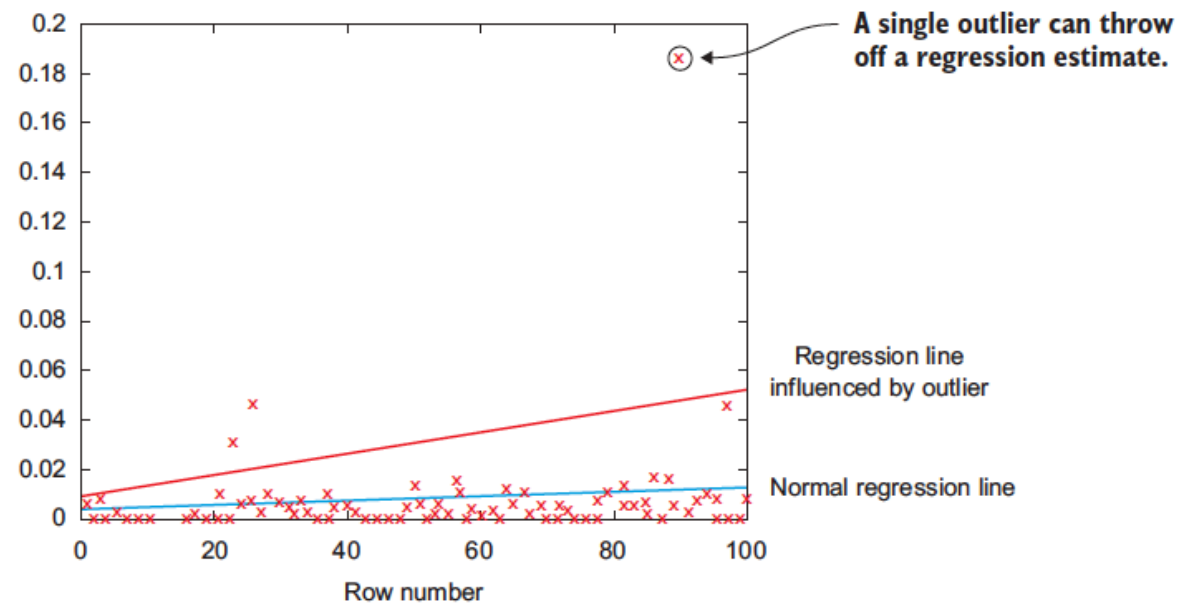
Technique	Advantage	Disadvantage
Omit the values	Easy to perform	You lose the information from an observation
Set value to null	Easy to perform	Not every modeling technique and/or implementation can handle null values
Impute a static value such as 0 or the mean	Easy to perform You don't lose information from the other variables in the observation	Can lead to false estimations from a model
Impute a value from an estimated or theoretical distribution	Does not disturb the model as much	Harder to execute You make data assumptions
Modeling the value (nondependent)	Does not disturb the model too much	Can lead to too much confidence in the model Can artificially raise dependence among the variables Harder to execute You make data assumptions

Decision diagram for handling missing values when preparing data for ML modeling



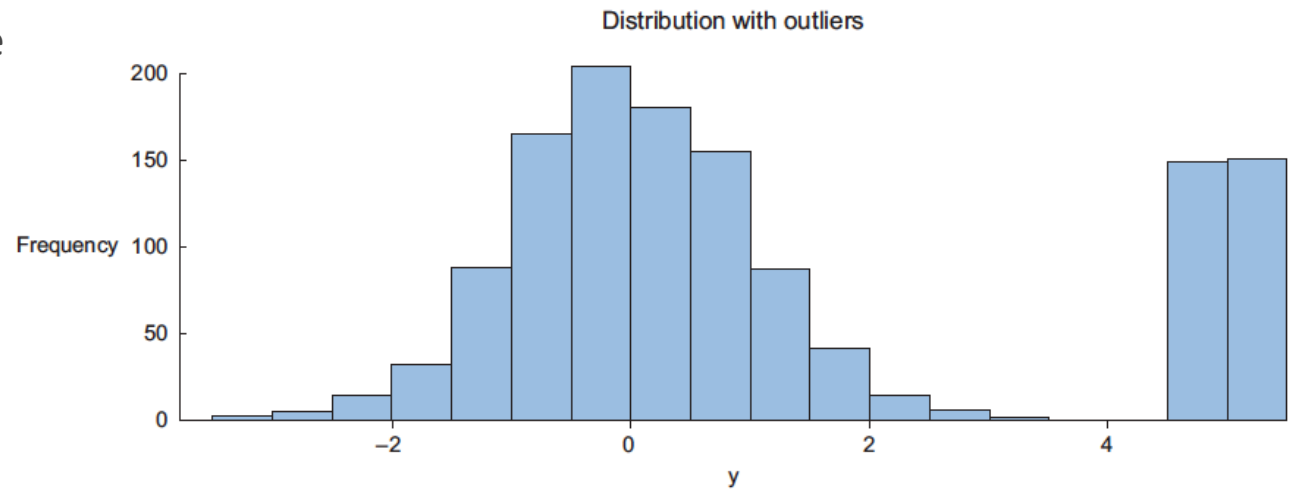
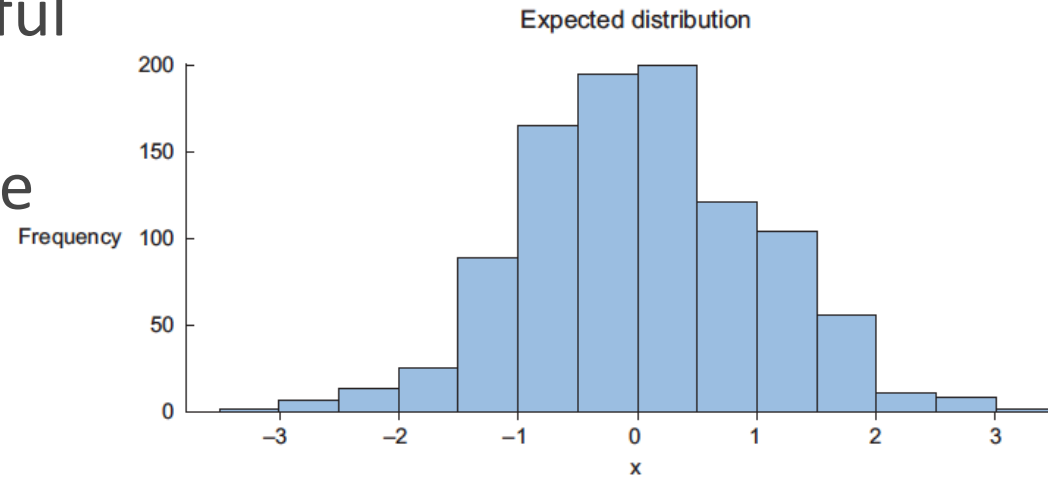
Outliers

- An outlier is an observation that seems to be distant from other observations
- Or observation that follows a different logic or generative process than the other observations
- The easiest way to find outliers is to use a plot or a table with the minimum and maximum values
- There are ML based methods for outliers detection



Distribution plots are helpful in detecting outliers and helping you understand the variable

- Simple rules of thumb to detect:
 - Data points three or more standard deviations from the mean



Dealing with Outliers

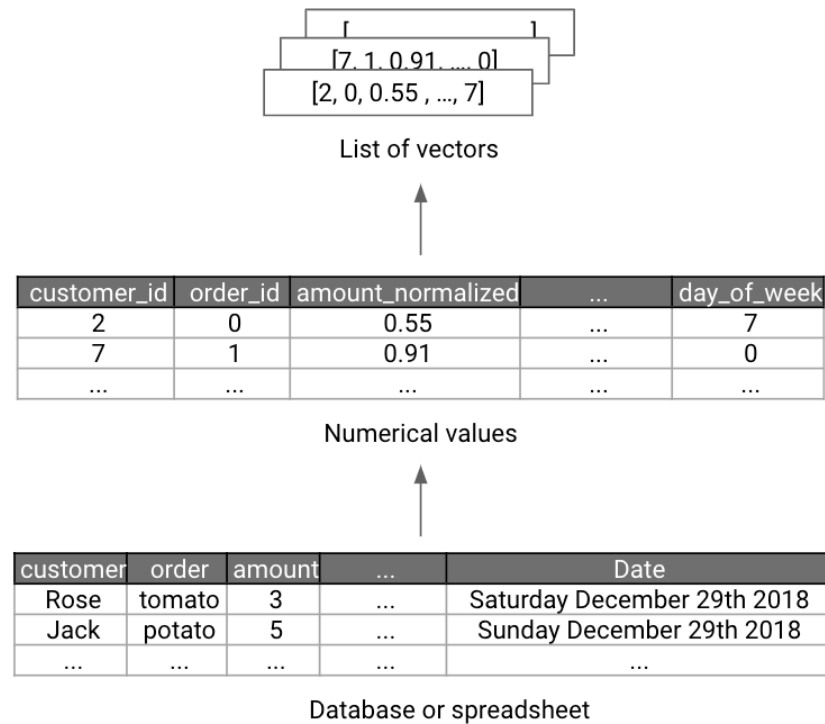
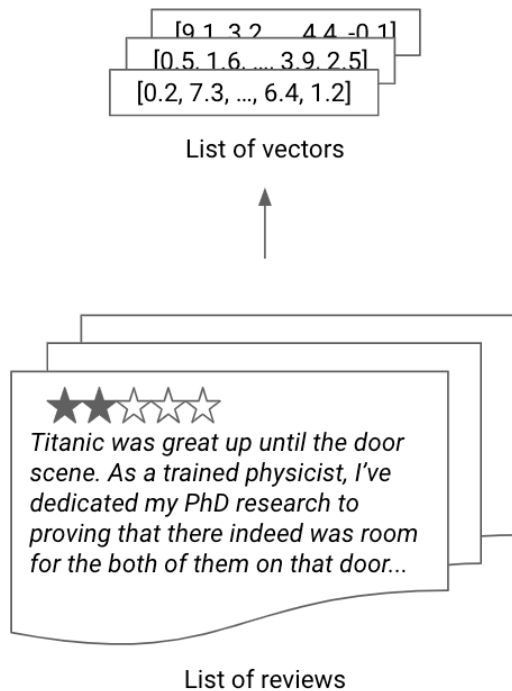
- Removal
 - 2,3,4,5,6,10,100 -> 2,3,4,5,6,10,~~100~~
- Transformations (e.g.: taking the log)
 - 2,3,4,5,6,10,100 -> 0.3, 0.47, 0.6, 0.69, 1, 2
- Truncation
 - 2,3,4,5,6,10,100 -> 2,3,4,5,6,10,10

Why Outliers matter

- Present false information (data errors, etc.)
- Ruin predictions (increase error-proneness)
- Create new questions (new clusters, etc.)
- Offer insights (anomalies, examples, etc.)
- Announce issues (fraud, etc.)

Vectorizing

- Vectorizing a dataset is the process of going from the raw data to a vector that represents it.



Handling of Categorical Data

- Categorical features can only take on a limited, and usually fixed, number of possible values.
 - For example, if a dataset is about information related to users, then you will typically find features like country, gender, age group, etc.
 - These features are typically stored as text values which represent various traits of the observations. For example, gender is described as Male (M) or Female (F)
- Many machine learning models are algebraic.
 - This means that their input must be numerical. To use these models, categories must be transformed into numbers first, before you can apply the learning algorithm on them.
 - While some ML packages or libraries might transform categorical data to numeric automatically based on some default embedding method, many other ML packages don't support such inputs.
- Nominal features: where the categories are only labeled without any order of precedence
 - For example, Male (M) or Female (F)
- Ordinal features: which have some order associated with them.
 - For example, economic status, with three categories: low, medium and high, which have an order associated with them.

Encoding Categorical Data

- We are faced with the challenge of figuring out how to turn these text values into numerical values for further processing and unmask lots of interesting information which these features might hide.
- Typically, any standard work-flow in feature engineering involves some form of transformation of these categorical values into numeric labels and then applying some encoding scheme on these values.
- Some of available techniques are the following:
 - Encoding labels
 - One-Hot encoding
 - Binary encoding
 - Proxy encoding

Encoding labels

	carrier	tailnum	origin	dest	carrier_code
0	AS	N508AS	PDX	ANC	1
1	US	N195UW	SEA	CLT	8
2	UA	N37422	PDX	IAH	7
3	US	N547UW	PDX	CLT	8
4	AS	N762AS	SEA	ANC	1

	carrier	tailnum	origin	dest
0	1	N508AS	PDX	ANC
1	8	N195UW	SEA	CLT
2	7	N37422	PDX	IAH
3	8	N547UW	PDX	CLT
4	1	N762AS	SEA	ANC

- Basic method, which is just replacing the categories with the desired numbers using a dictionary which contains mapping numbers for each category in the carrier column
- Another approach is to encode categorical values with a technique called "label encoding", which allows you to convert each value in a column to a number. Numerical labels are always between 0 and n_categories-1.

Binary Encoding

	carrier	tailnum	origin	dest
0	1	N508AS	PDX	ANC
1	8	N195UW	SEA	CLT
2	7	N37422	PDX	IAH
3	8	N547UW	PDX	CLT
4	1	N762AS	SEA	ANC

	carrier_0	carrier_1	carrier_2	carrier_3	tailnum	origin	dest
0	0	0	0	0	N508AS	PDX	ANC
1	0	0	0	1	N195UW	SEA	CLT
2	0	0	1	0	N37422	PDX	IAH
3	0	0	0	1	N547UW	PDX	CLT
4	0	0	0	0	N762AS	SEA	ANC

- This technique is not as intuitive as the previous ones. In this technique, first the categories are encoded as ordinal, then those integers are converted into binary code, then the digits from that binary string are split into separate columns. This encodes the data in fewer dimensions than one-hot.

Proxy encoding

- Find a value that can be associated with the category and have some additional meaning
- For example:
 - If you are making economical prediction you can code the country with its GDP value
 - If you have cities then you can code them with their geo locations
 - If you have a special interest for example for US flights then you can Binarize the feature, and code US with 1 and all other countries with 0
 - Sometimes coding the category with its size can be an easy and very effective method
- There are sophisticated methods for categorical variables encoding

Date and Time

- Convert to numerical features
- Date:
 - Year, Month, Day
 - Number of days after/before given start date
 - Day of week, Working day, Weekend, Holliday
- Time
 - Hour, Seconds, Milliseconds
 - Number of Seconds/Milliseconds after/before given start date
 - Working hour, Non-working hour
- Date and Time
 - Use combination of both methods

Text data vectorization

- The simplest way to vectorize text is to use a count vector, which is the word equivalent of one-hot encoding.
- Start by constructing a vocabulary consisting of the list of unique words in your dataset.
- Associate each word in our vocabulary to an index (from 0 to the size of our vocabulary).
- You can then represent each sentence or paragraph by a list as long as our vocabulary.
- For each sentence, the number at each index represents the count of occurrences of the associated word in the given sentence.
- This method ignores the order of the words in a sentence and so is referred to as a bag of words.

Text data vectorization

- Example: Getting bag-of-words vectors from sentences

	Input text
Sentence 1	"Mary is hungry for apples."
...	...
Sentence 345	"John is happy he is not hungry for apples."



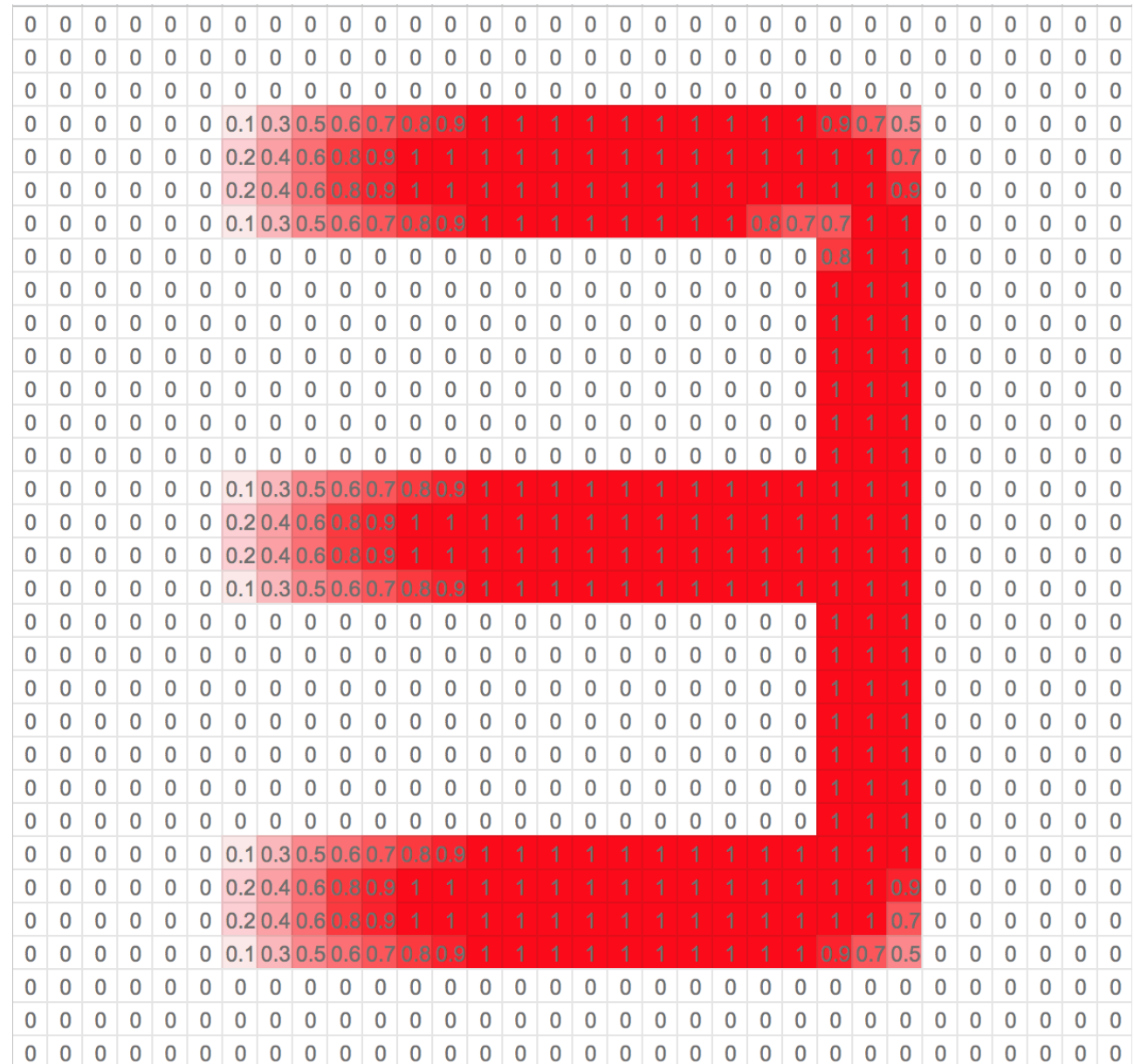
Word index	MARY	IS	HUNGRY	HAPPY	FOR	...	APPLES	NOT	JOHN	HE	SAND
Sentence 1	1	1	1	0	1	...	1	0	0	0	0
...
Sentence 345	0	2	1	1	1	...	1	1	1	1	0

Image data

- Image data is already vectorized, as an image is nothing more but a multidimensional array of numbers
- Most standard three-channel RGB images, for example, are simply stored as a list of numbers of length equal to the height of the image in pixels, multiplied by its width, multiplied by three (for the red, green, and blue channels).

Image data

- Example: Representing a 3 as a matrix of values from 0 to 1 (only showing the red channel)



Vectorization and Data Leakage

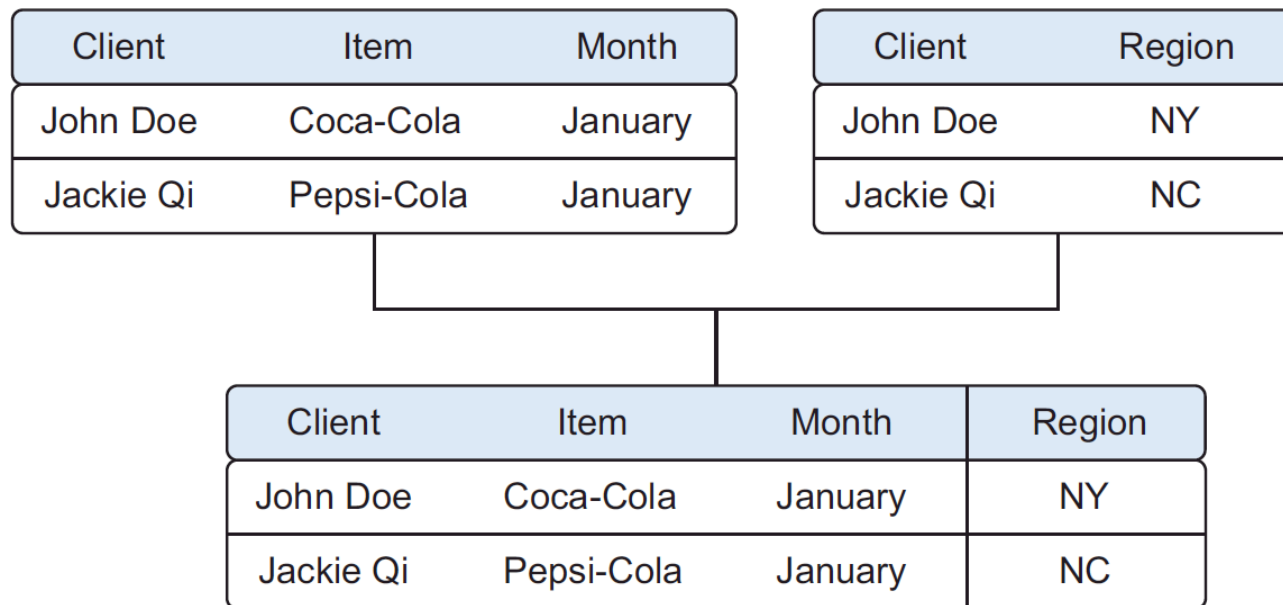
- You would usually use the same techniques to vectorize data to visualize it and to feed it to a model.
 - When you vectorize data to feed it to a model, you should vectorize your training data and save the parameters you used to obtain the training vectors.
 - You should then use the same parameters for your validation and test sets.
 - When normalizing data, for example, you should compute summary statistics such as mean and standard deviation only on your training set (using the same values to normalize your validation data), and during inference in production.
- Using both your validation and training data for normalization, or to decide which categories to keep in your one-hot encoding, would cause data leakage, as you would be leveraging information from outside your training set to create training features.
 - This would artificially inflate your model's performance but make it perform worse in production.

Combining data from different data sources

- Your data comes from several different places
- Data varies in size, type, and structure, ranging from databases and Excel files to text documents.
- We focus on data in table structures, but keep in mind that other types of data sources exist, such as key-value stores, document stores, ...
- You can perform two operations to combine information from different data sets.
 - The first operation is joining: enriching an observation from one table with information from another table.
 - The second operation is appending or stacking: adding the observations of one table to those of another table

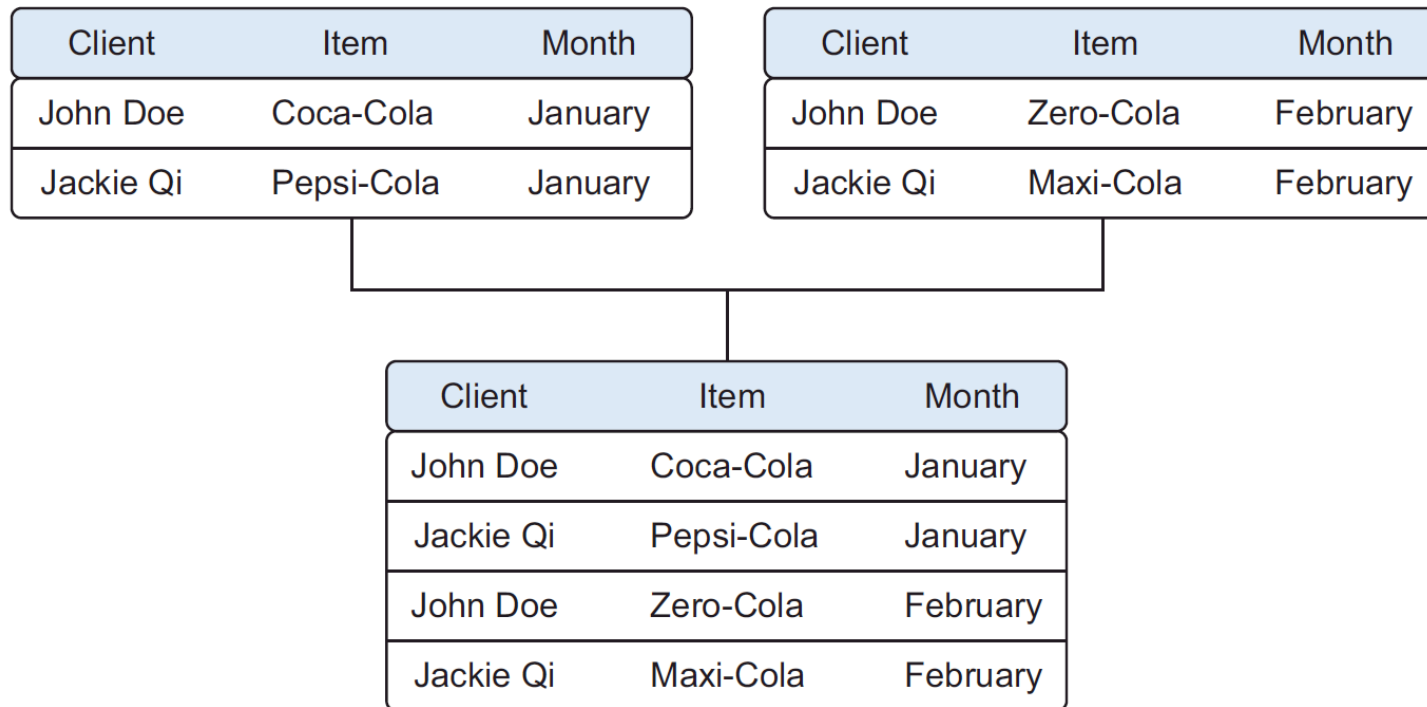
Joining tables

- To join tables, you use variables that represent the same object in both tables, such as a date, a country name, or a Social Security number.
- These common fields are known as keys.



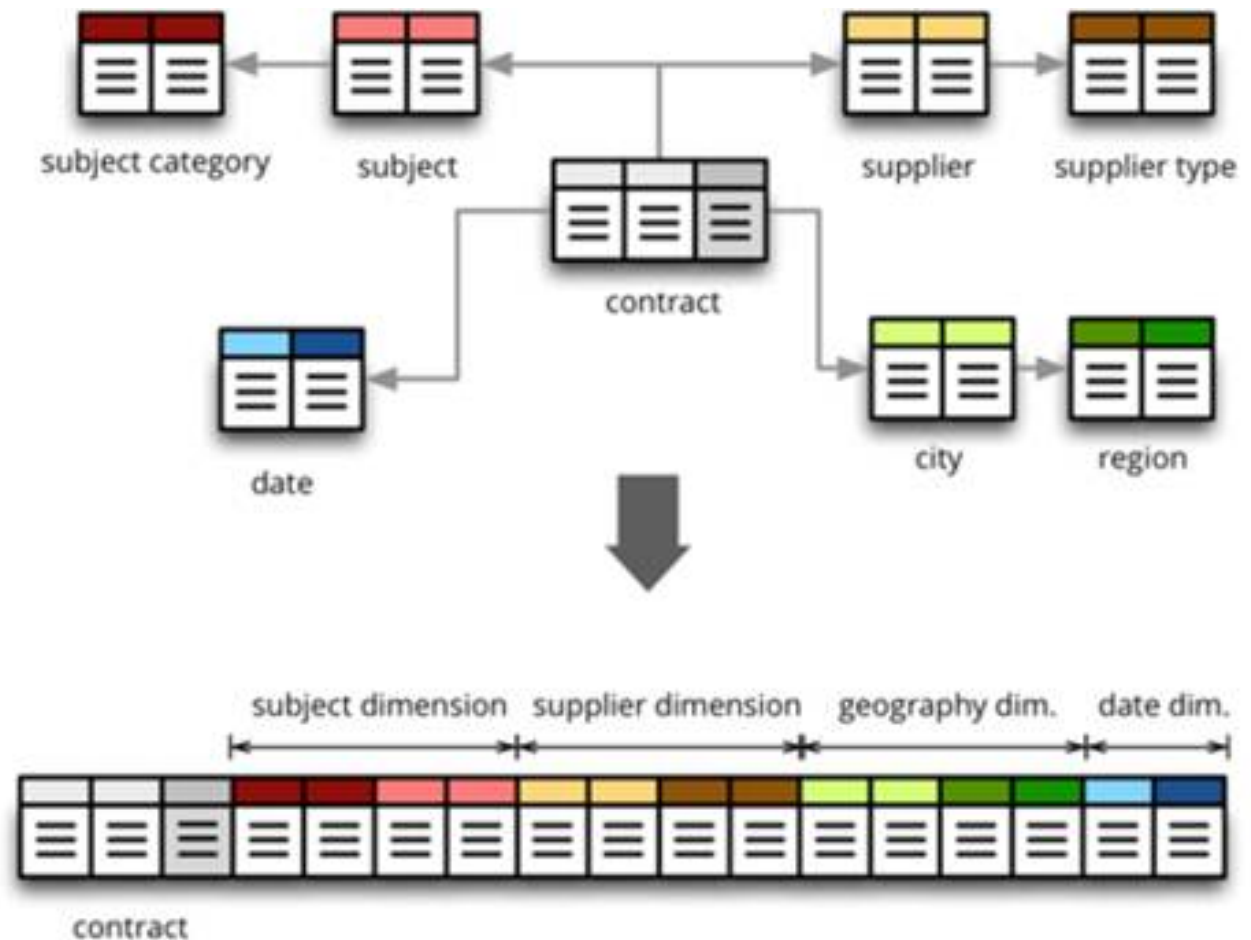
Appending tables

- Appending data from tables is a common operation but requires an equal structure in the tables being appended.



Data SCHEMA denormalization

- Combining data from different data sources into Single table
- Each row is a single instance with all attributes
- This is most suitable format for a lot of Machine Learning algorithms



Enriching aggregated measures

- Data enrichment can also be done by adding calculated information to the table, such as the total number of sales or what percentage of total stock has been sold in a certain region
- Extra measures such as these can add perspective.
- As always this depends on the exact case, but from experience models with “relative measures” such as % sales (quantity of product sold/total quantity sold) tend to outperform models that use the raw numbers (quantity sold) as input.

Product class	Product	Sales in \$	Sales t-1 in \$	Growth	Sales by product class	Rank sales
A	B	X	Y	$(X-Y) / Y$	AX	NX
Sport	Sport 1	95	98	-3.06%	215	2
Sport	Sport 2	120	132	-9.09%	215	1
Shoes	Shoes 1	10	6	66.67%	10	3

Data values Standardization and normalization

- Some ML algorithms require data to be normalized, meaning that each individual feature has been manipulated to reside on the same numeric scale.
- This step is very important when dealing with parameters of different units and scales.
 - For example, some data mining techniques use the Euclidean distance. Therefore, all parameters should have the same scale for a fair comparison between them.

- *Normalization*, which scales all numeric variables in the range [0,1]

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

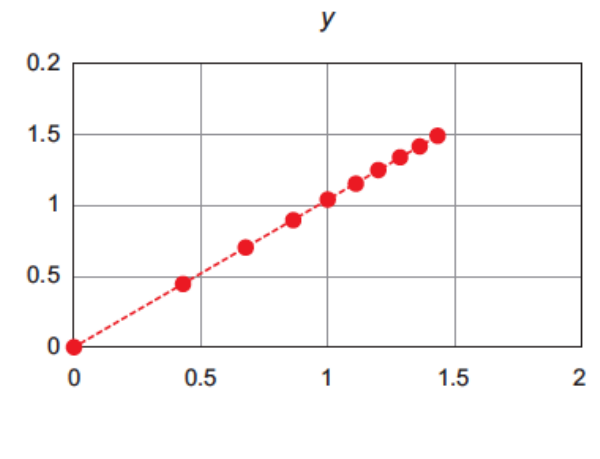
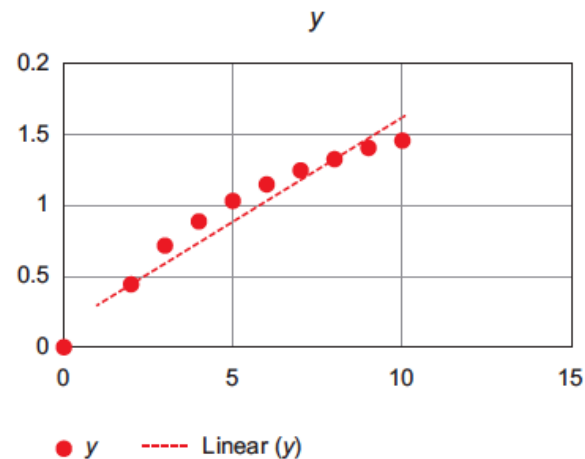
- *Standardization* will transform data to have zero mean and unit variance

$$x_{new} = \frac{x - \mu}{\sigma}$$

Transforming data

- Certain models require their data to be in a certain shape
- For example, a relationship of the form $y = ae^{bx}$.
 - Taking the log of the independent variables simplifies the estimation problem dramatically.

x	1	2	3	4	5	6	7	8	9	10
log(x)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46

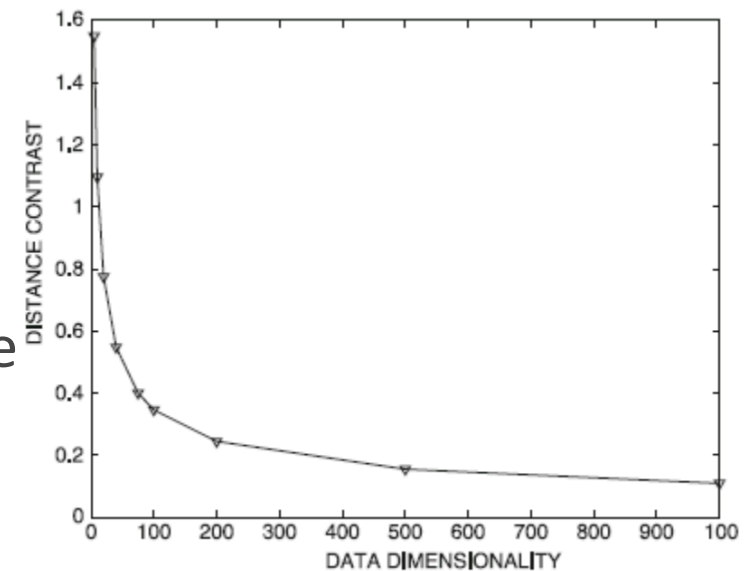


Reducing the number of variables (Dimensionality Reduction)

- Sometimes you have too many variables and need to reduce the number because they don't add new information to the model.
- Having too many variables in your model makes the model difficult to handle, and certain techniques don't perform well when you overload them with too many input variables.
- For instance, all the techniques based on a Euclidean distance perform well only up to 10 variables.
- Special ML methods exist to reduce the number of variables but retain the maximum amount of data

Curse of dimensionality

- The 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.
- The squared distance of random points to a selected point is, with high probability, close to the average (or median) squared distance.
 - For instance, all the techniques based on a Euclidean distance perform well only up to 10 variables
- Other similarity measures like cosine similarity can be used for higher dimensional space



Cosine similarity

- Cosine similarity is **the cosine of the angle between two n-dimensional vectors in an n-dimensional space**. It is the dot product of the two vectors divided by the product of the two vectors' lengths (or magnitudes).

$$\mathbf{A} = (A_1, A_2, \dots, A_n)$$

$$\mathbf{B} = (B_1, B_2, \dots, B_n)$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Correct errors as early as possible

- Not everyone spots the data anomalies.
- Decision-makers may make costly mistakes on information based on incorrect data from applications that fail to correct for the faulty data.
- If errors are not corrected early on in the process, the cleansing will have to be done for every project that uses that data.
- Data errors may point to
 - a business process that isn't working as designed.
 - defective equipment, such as broken transmission lines and defective sensors.
 - bugs in software or in the integration of software that may be critical to the company.
- Remark: always keep a copy of your original data (if possible)!!

Conclusion

- Check data for errors
 - You need as much as possible data
 - Data need to be vectorized
-
- The data preparation can have very high impact on the final ML model performance.