

Lecture Notes for **Machine Learning in Python**

Professor Eric Larson
Numpy, Pandas, Document Features

Data Quality Problems

- Missing
 - Easy to find, NaNs
- Duplicated
 - Easy to find, hard to verify
- Noise or Outlier
 - Hard to define
 - Hard to catch

Information is not collected
(e.g., people decline to give their age and weight)

Features **not applicable**
(e.g., annual income for children)

UCI ML Repository: 90% of repositories have missing data

<i>TID</i>	<i>Hair Color</i>	<i>Height</i>	<i>Age</i>	<i>Arrested</i>
1	Brown	5'2"	23	no
2	Hazel	1.5m	12	no
3	Bl	5	999	no
4	Brown	5'2"	23	no

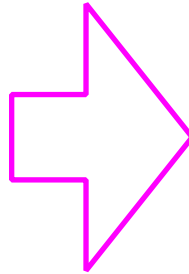
Handling Issues with Data Quality

- **Eliminate** Instance or Feature
- **Ignore** the Missing Value During Analysis Replace with all possible values (talk about later)
- **Impute** Missing Values

How?

Stats?
mean
median
mode

Split-Impute-Combine



<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Age</i>	<i>Diabetes</i>
1	Y	33.6	41-50	positive
2	N	26.6	31-40	negative
3	Y	23.3	?	positive
4	N	28.1	21-30	negative
5	N	43.1	31-40	positive
6	Y	25.6	21-30	negative
7	Y	31.0	21-30	positive
8	Y	35.3	?	negative
9	N	30.5	51-60	positive
10	Y	37.6	51-60	positive

split: pregnant
split: BMI > 32

<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Age</i>	<i>Diabetes</i>
1	Y	>32	41-50	positive
8	Y	>32	?	negative
10	Y	>32	51-60	positive

Mode: none, can't impute

<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Age</i>	<i>Diabetes</i>
3	Y	<32	?	positive
6	Y	<32	21-30	negative
7	Y	<32	21-30	positive

Mode: 21-30

K-Nearest Neighbors Imputation

For K=3, find 3 closest neighbors

TID	Pregnant	BMI	Age	Diabetes
1	Y	33.6	41-50	positive
2	N	26.6	31-40	negative
3	Y	23.3	?	positive
4	?	28.1	21-30	negative
5	N	43.1	31-40	positive
6	Y	25.6	21-30	negative
7	Y	31.0	21-30	positive
8	Y	35.3	?	negative
9	N	30.5	51-60	positive
10	Y	37.6	51-60	positive

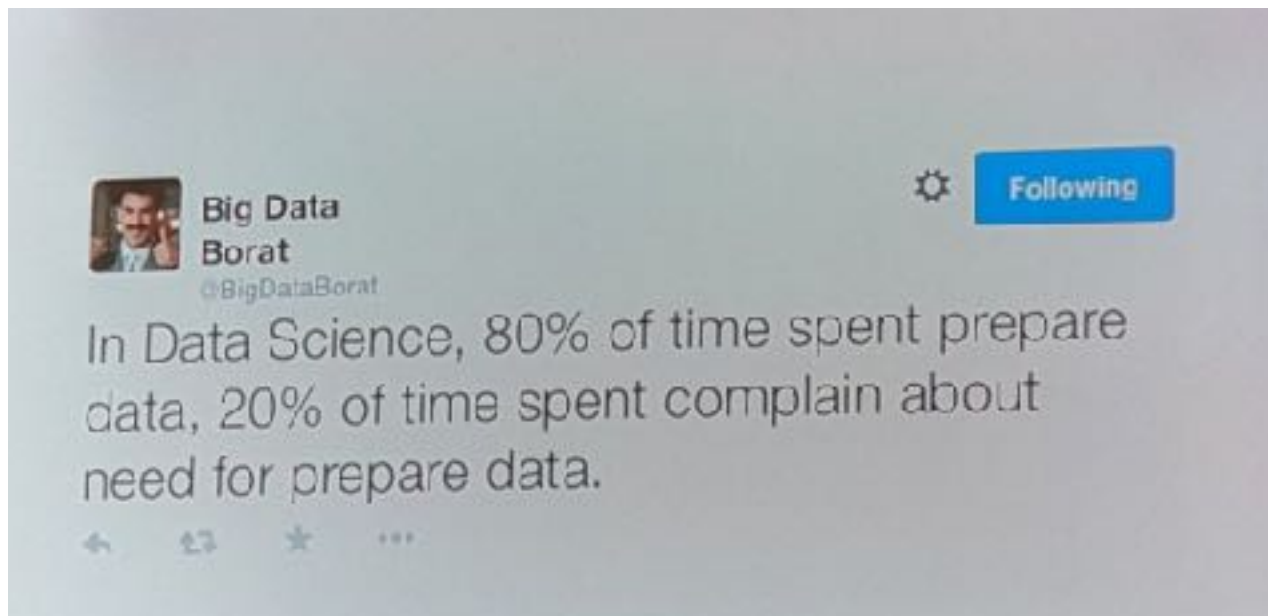
TID	Preg nant	BMI	Age	Diabetes	Distance
3	Y	23.3	?	positive	0
6	Y	25.6	21-30	negative	$(0 + 2.3 + 1)/3$
2	N	26.6	31-40	negative	$(1 + 3.3 + 1)/3$
4	?	28.1	21-30	negative	$(4.8 + 1)/2$

Imputed Age: 21-30

How to calculate distance?

- Difference for valid features only
- May need to normalize ranges
- Or weight neighbors differently
- Or have min # of valid features
- Euclidean, city-block, etc.

Data Representation and Documents



Feature Type Representation Review

	Attribute	Representation Transformation	Comments
Discrete	Nominal	Any permutation of values one hot encoding or hash function	If all employee ID numbers were reassigned, would it make any difference?
	Ordinal	An order preserving change of values, i.e., $\text{new_value} = f(\text{old_value})$ where f is a monotonic function. integer	An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by {0.5, 1, 10}.
Continuous	Interval	$\text{new_value} = a * \text{old_value} + b$ where a and b are constants float	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
	Ratio	$\text{new_value} = a * \text{old_value}$ float	Length can be measured in meters or feet.

Data Tables as Variable Representations

Table

<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Age</i>	<i>Eye Color</i>	<i>Diabetes</i>
1	Y	33.6	41-50	brown	positive
2	N	26.6	31-40	hazel	negative
3	Y	23.3	31-40	blue	positive
4	N	28.1	21-30	brown	inconclusive
5	N	43.1	31-40	blue	positive
6	Y	25.6	21-30	hazel	negative

Internal Rep.

<i>TID</i>
1
2
3
4
5
6

Data Tables as Variable Representations

Table

<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Age</i>	<i>Eye Color</i>	<i>Diabetes</i>
1	Y	33.6	41-50	brown	positive
2	N	26.6	31-40	hazel	negative
3	Y	23.3	31-40	blue	positive
4	N	28.1	21-30	brown	inconclusive
5	N	43.1	31-40	blue	positive
6	Y	25.6	21-30	hazel	negative

Internal Rep.

<i>TID</i>	<i>Binary</i>	<i>Float</i>	<i>Ordinal</i>	<i>Object</i>	<i>Diabetes</i>
1	1	33.6	2	hash(0)	1
2	0	26.6	1	hash(1)	0
3	1	23.3	1	hash(2)	1
4	0	28.1	0	hash(0)	2
5	0	43.1	1	hash(2)	1
6	1	25.6	0	hash(1)	0

Bag of words model

<i>TID</i>	<i>Pregnant</i>	<i>BMI</i>	<i>Chart Notes</i>	<i>Diabetes</i>
1	Y	33.6	Complaints of fatigue wh...	positive
2	N	26.6	Sleeplessness and some...	negative
3	Y	23.3	First saw signs of rash o...	positive
4	N	28.1	Came in to see Dr. Steve...	inconclusive
5	N	43.1	First diagnosis for hospit...	positive
6	Y	25.6	N/A	negative

Bag of Words

Vocabulary						
TID	Sleep	Fatigue	Weight	Rash	First	Sight
1	0	1	0	0	2	0
2	1	1	0	0	1	1
3	1	1	0	2	1	1

number of occurrences

Feature Hashing

what happens when we get more words?

TID	Slee	Fati	Wei	Ras	First	Sigh	Why	Fox	Bro	Lazy	Dog	Etc	Stev
1	0	1	0	0	2	0	0	0	0	1	0	2	0
2	1	1	0	0	1	1	0	0	4	0	1	3	0
3	1	1	0	2	1	1	1	0	1	0	0	1	0

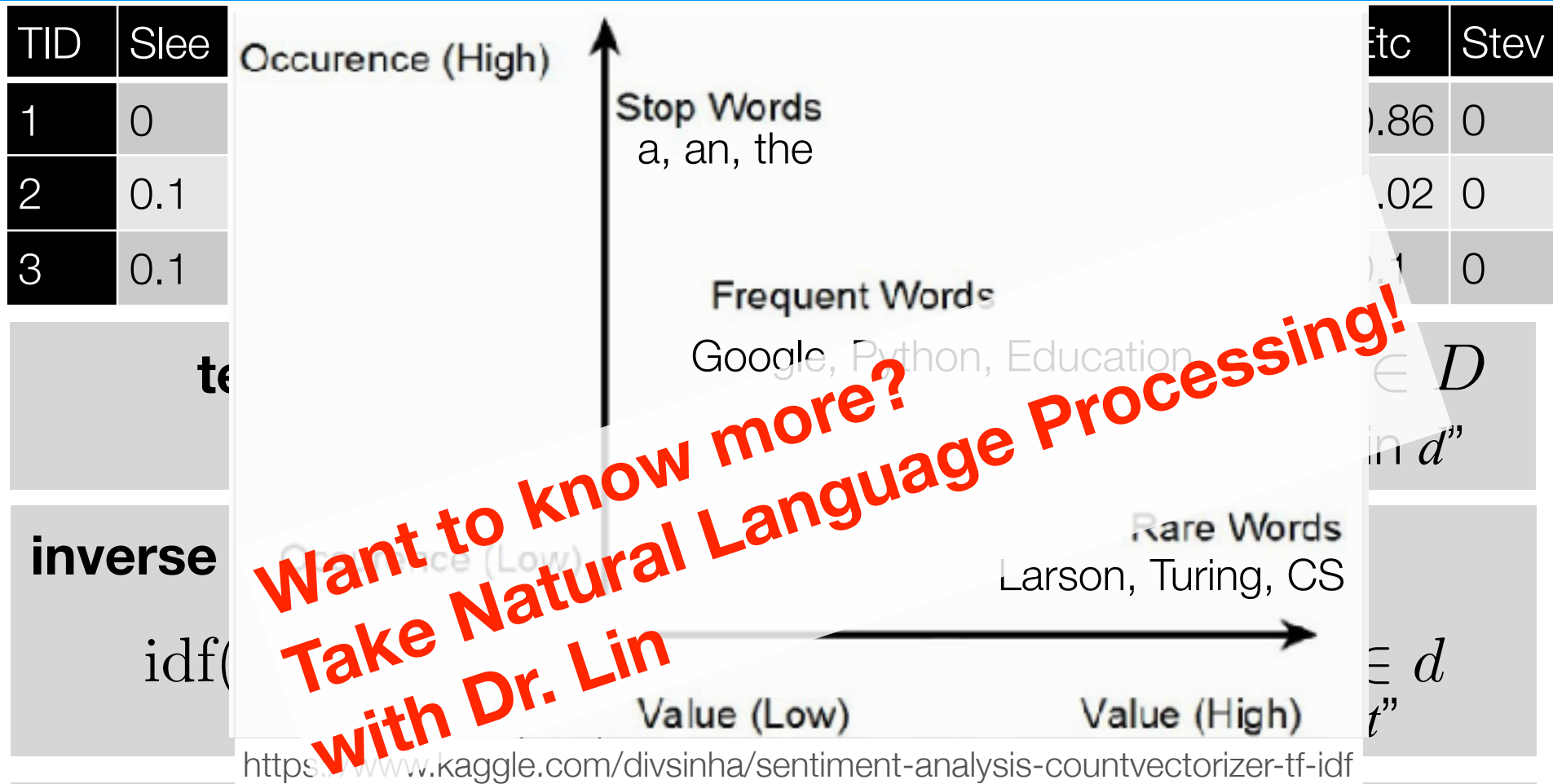
or we could have a hashing function, $h(x) = y$

	$h(x)=1$	$h(x)=2$	$h(x)=3$	$h(x)=4$	$h(x)=5$	$h(x)=6$
1	0	1	0	1	2	0
2	1	1	4	0	2	1
3	2	1	1	2	1	1

multiple words mapped to one hash:

(**want to** (1) minimize collisions **or** (2) make collisions meaningful)

Term-Frequency, Inverse-Document-Frequency



$$\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot \text{idf}(t, d)$$

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot (1 + \text{idf}(t, d)) \quad \text{smoothed}$$

Pandas and Imputation
Scikit-Learn



Start the following:
`03. Data Visualization.ipynb`

Other Tutorials:

<http://vimeo.com/59324550>

<http://pandas.pydata.org/pandas-docs/version/0.15.2/tutorials.html>

For Next Lecture

- Before next class:
 - verify installation of seaborn, plotly, (and/or bokeh if you want)
 - look at pandas table data and additional tutorials
- Next Week: Data Visualization