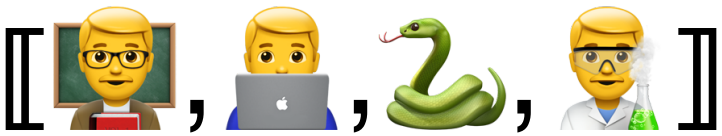
Lecture Notes for

Machine Learning in Python



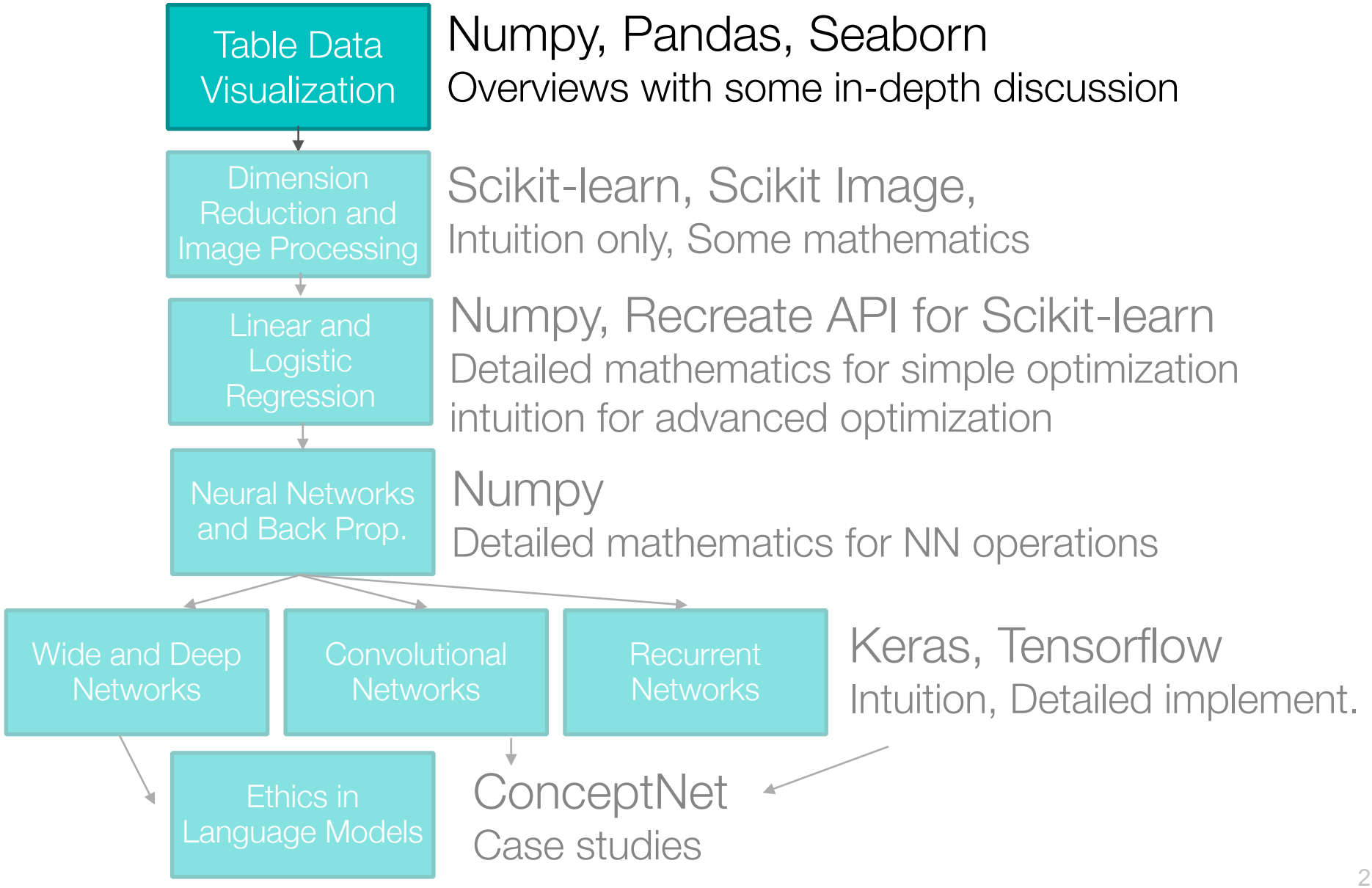
Professor Eric Larson

Table Data using Numpy, Pandas

Class Logistics and Agenda

- Canvas. Participation. Quizzes. Attendance.
- Agenda:
  - Data Encodings
  - Demo: Table Data, Numpy
  - Data Quality
  - Attributes Representation
    - documents
  - The Pandas eco-system
    - loading and manipulating attributes

Class Overview, by topic



Types of Data and Categorization

Table Data

- **Table Data**: Collection of data **instances** and their **features**
  - **Python:** Pandas Dataframe
  - **R:** Data.frame
  - **Matlab:** Table Class
  - **C++**: Make your own,

```
std::vector<Record>
```

Self Test: Table data storage is not memory efficient.

a) True
b) False
c) It depends on the backend
d) It depends on the programming language of the interface

Attributes, columns, variables, fields, characteristics, **Features**

Target, Class, Label

Objects, records, rows, **points**, **samples**, cases, entities, **instances**

	Pregnant	*BMI*	*Age*	*Diabetes*
1	Y	33.6	41-50	positive
2	N	26.6	31-40	negative
3	Y	23.3	31-40	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	21-30	negative
9	N	30.5	51-60	positive
10	Y	37.6	51-60	positive

Feature Vector Representation

$f_{mono}(\cdot)$ monotonic function

Discrete	**Nominal or Categorical**	Variable could be one value in a set of categories. No ordering of values. *Example: Employee ID*	**Allowed Transforms**: permuting values **boolean, one hot encoding, or hash function**
Discrete	**Ordinal**	Variable could be one value in a set of categories. Ordering matters. *Example: Start Ratings, 1-5*	**Allowed Transforms**: $V_{new} = f_{mono}(V_{old}) + b$ **integer (or boolean)**
Continuous	**Interval or Numeric**	Value is continuous numeric value. Could be in specified range. *Example: BMI, Temperature, etc.*	**Allowed Transforms**: $V_{new} = f_{mono}(V_{old}) + b$ **float**
Continuous	**Ratio or Numeric**	Value is continuous numeric value. Zero is meaningful. Often not treated differently than interval. *Example: Length, Elevation*	**Allowed Transforms**: $V_{new} = f_{mono}(V_{old})$ **float**

Data Tables as Variable Representations

Table

TID	Pregnant	BMI	Age	Eye Color	Diabetes
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.

TID
1
2
3
4
5
6

Demo

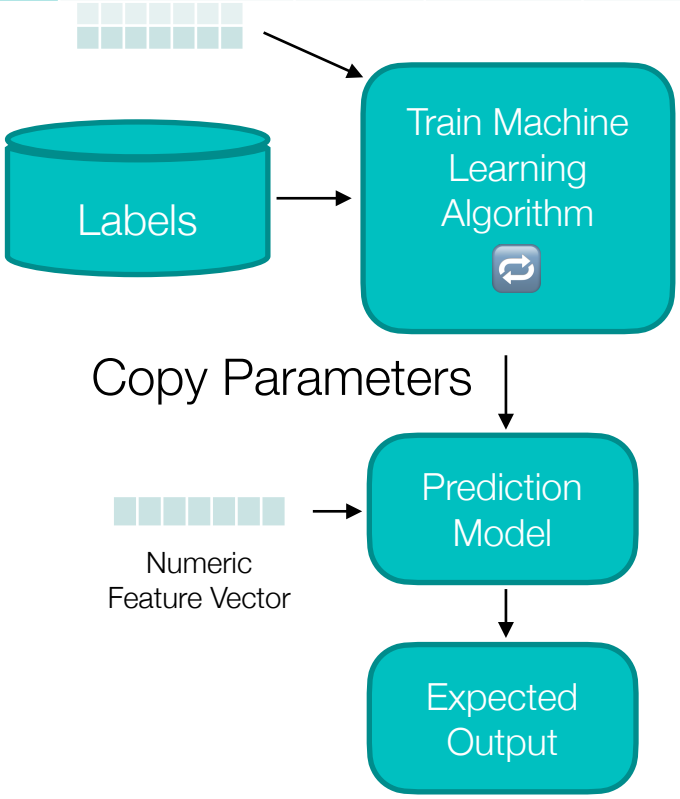
"Finish"
Jupyter Notebooks

`01_Numpy and Pandas Intro.ipynb`

Data Quality

Data Quality Problems

TID	Hair Color	Hgt.	Age	Arrested
1	Brown	5'2"	23	no
2	Hazel	1.5m	12	
3	NaN	5	999	no
4	Brown	5'2"	23	no



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

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?

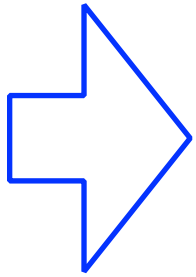
Stats?
mean
median
mode

Imputation

- When is it probably fine to impute missing data:
  - (A) When there is not much missing data
  - (B) When the missing feature is mostly predictable from another feature
  - (C) When there is not much missing data for each subgroup of the data
  - (D) When it is the class you want to predict

Split-Impute-Combine

TID	Pregnant	BMI	Age	Diabetes
1	Y	33.6	31-40	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	21-30	positive



split: pregnant
split: BMI > 32

TID	Pregnant	BMI	Age	Diabetes
1	Y	>32	31-40	positive
8	Y	>32	?	negative
10	Y	>32	21-30	positive

Mode: none, can't impute

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

Mode: 21-30

For Next Lecture

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

- Next time: Data Imputation Demo

Lecture Notes for

Machine Learning in Python

Professor Eric Larson

Table Data using Numpy, Pandas