Lecture Notes for

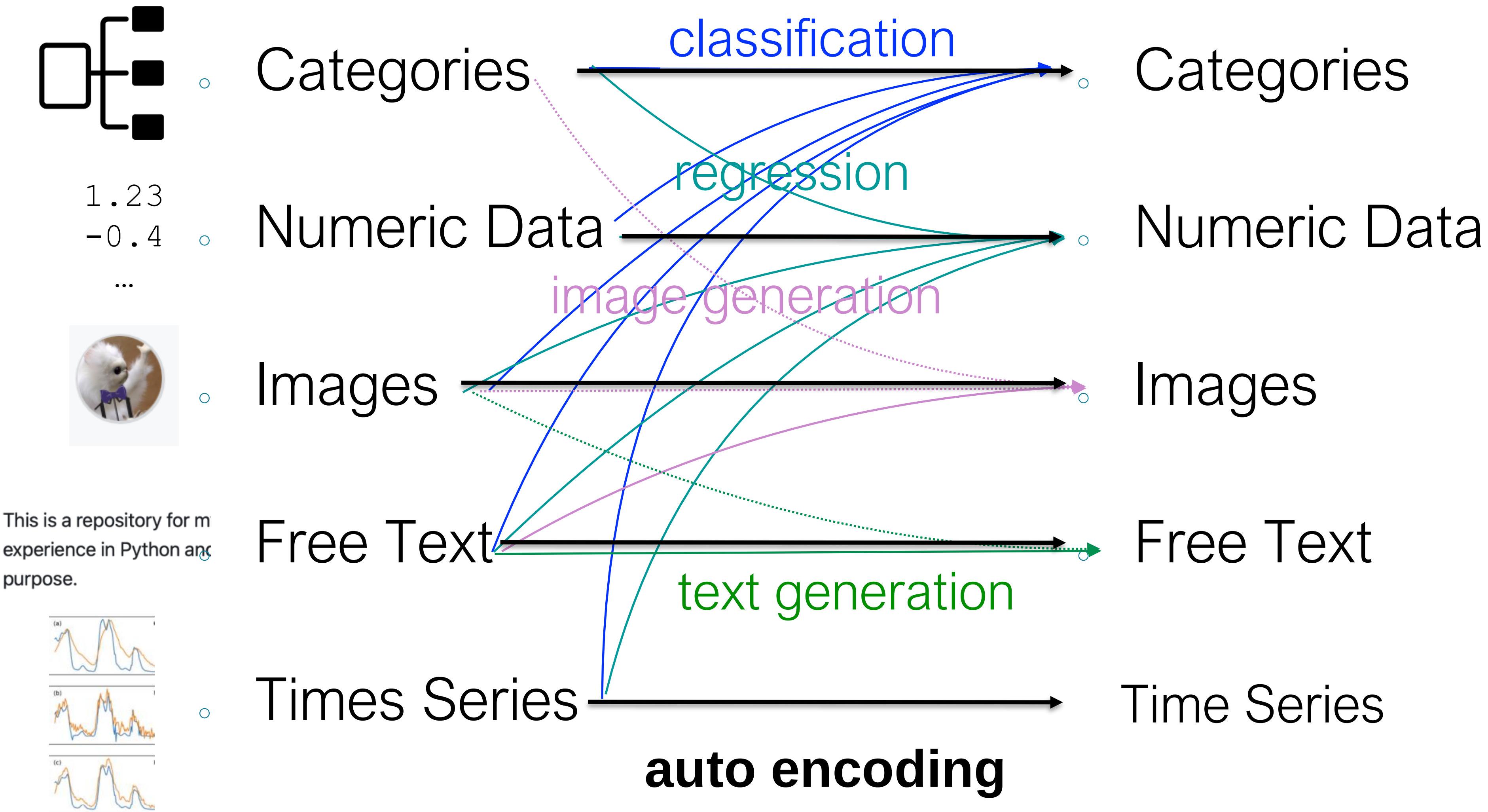
Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

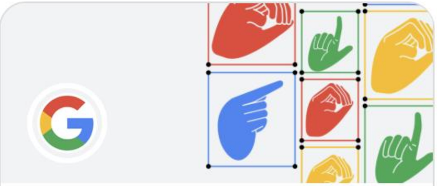
Table Data using Numpy, Pandas

Problem Types in Machine Learning

- Inputs
- Outputs



Problem Types in Machine Learning



Google - American Sign Language Fingerspelling...

Train fast and accurate American Sign...

Research · Code Competition

1269 Teams

$200,000 3 days to go

CommonLit - Evaluate Student Summaries

Automatically assess summaries writt...

Featured · Code Competition

925 Teams

$60,000 2 months to go

Bengali.AI Speech Recognition

Recognize Bengali speech from out-of...

Research · Code Competition

317 Teams

$53,000 2 months to go

CAFA 5 Protein Function Prediction

Predict the biological function of a pro...

Research · Code Competition

1655 Teams

$50,000 10 hours to go

Kaggle - LLM Science Exam

Use LLMs to answer difficult science ...

Featured · Code Competition

1471 Teams

$50,000 2 months to go

RSNA 2023 Abdominal Trauma Detection

Detect and classify traumatic abdomi...

Featured · Code Competition

333 Teams

$50,000 2 months to go

Predict CO2 Emissions in Rwanda

Playground Series - Season 3, Episod...

Playground

1401 Teams

Swag 10 hours to go

Titanic - Machine Learning from Disaster

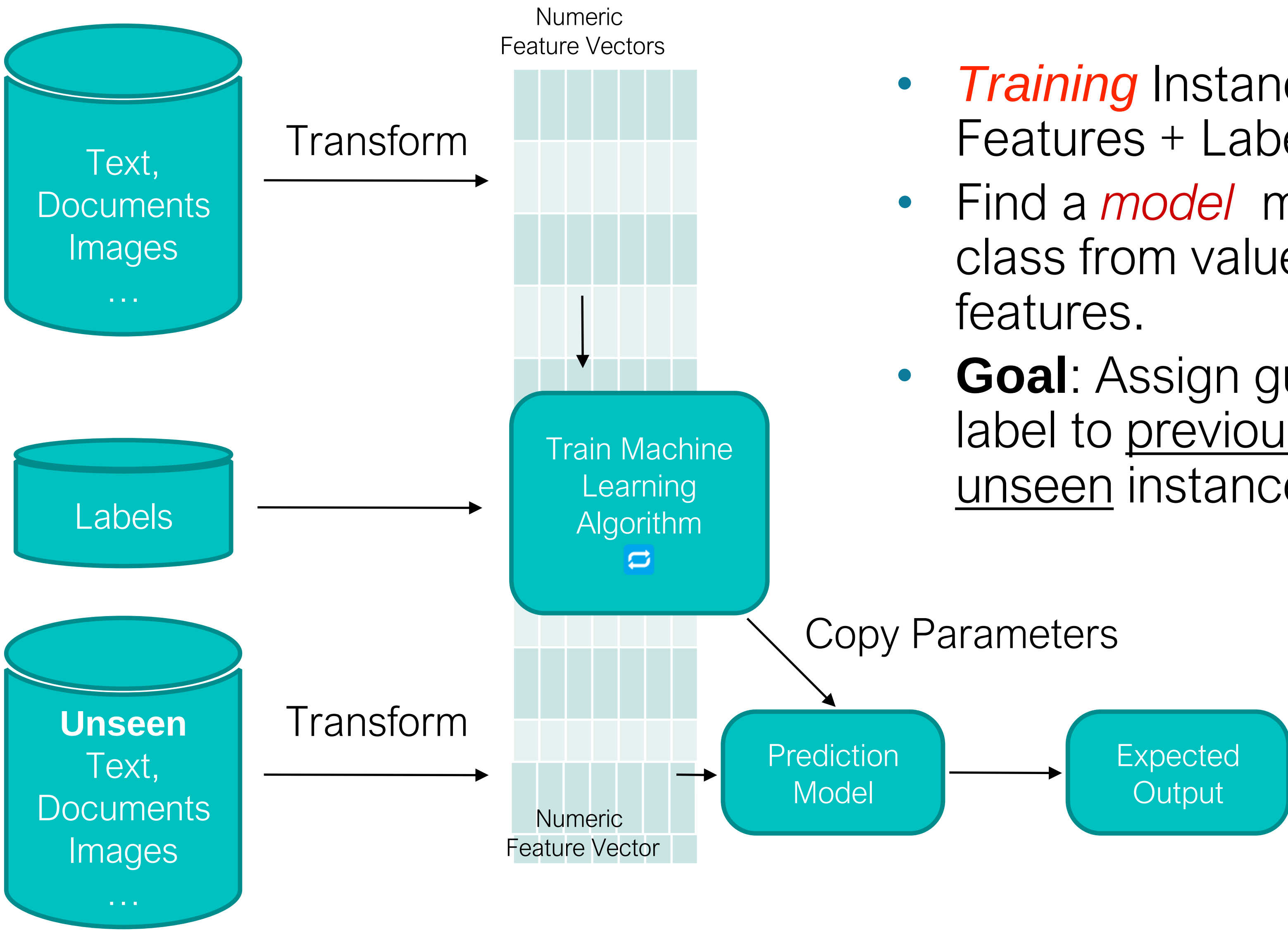
Start here! Predict survival on the Tita...

Getting Started

14897 Teams

Knowledge Ongoing

Classification and Regression, Supervised



- *Training* Instances: Features + Labels
- Find a *model* mapping class from values of features.
- **Goal**: Assign guessed label to previously unseen instances

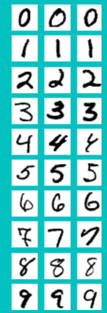
Some Popular Datasets

ImageNet

1M+

224 x 224 Color Image
⬇
1000 Classes
(prominent object)

MNIST



60k

24 x 24 Grey Image
⬇
10 Classes (digits)

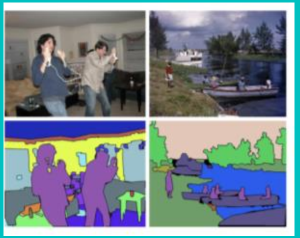
Adult

# feature	original feature
1	age
2	workclass
3	final weight
4	education
5	ed_num
6	marital_status
7	occupation
8	relationship
9	race
10	sex
11	capital_gain
12	capital_loss
13	hours × week
14	country

5k

Census Demographics
⬇
Binary (salary > 50k?)

CoCo



200k Images

Large, Multi-sized Images
⬇
Location, Size, 80 Objects

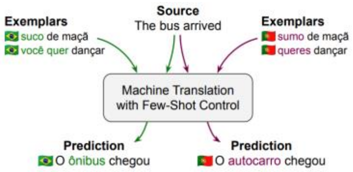
Boston Housing

1. CRIM
2. ZN
3. INDUS
4. CHAS
5. NOX
6. RM
7. AGE
8. DIS
9. RAD
10. TAX
11. PTRATIO
12. B
13. LSTAT
14. MEDV

House/Neighborhood Descriptions
⬇
House Price
$$

500 Examples

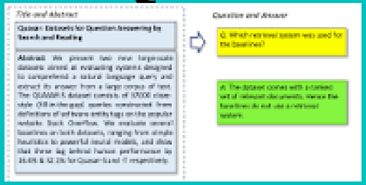
Translation



Language A
⬇
Language B

Many datasets

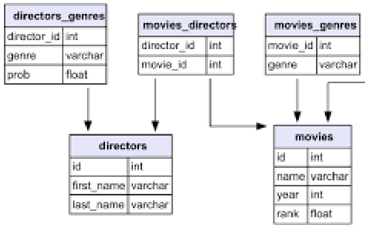
SQuAD



Question
⬇
Answer

100k+

Imdb



Movie/Actors/Director/+
⬇
Critic/Audience rating

50k reviews

Self Test

- **A. Classification**
  B. Regression
  C. Not Machine Learning
- **D. Machine Learning Generation**

- Dividing up customers by potential profitability?

- Extracting frequency of sound?

Before Next Lecture

- Before next class:
  - install python on your laptop
  - install anaconda distribution of python
  - use environments (`conda env`)

- Look at Python primer if you need review
  - Dr. Larson made ~4 hours of YouTube content…
  - https://www.youtube.com/playlist?list=PL7IPdRN5E0YKCnVl-fvx8jOOCWVeGTsrV

Class Logistics and Agenda

- Canvas? Anaconda Installs?
- In-person versus Zoom and other classes
- Agenda:
  - Data Encodings
  - Demo: Table Data, Numpy
  - Data Quality
  - Attributes Representation
    - documents
  - The Pandas eco-system
    - loading and manipulating attributes

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++**: Trick Question

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

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

TID	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

	Attribute	Representation Transformation	Comments
Discrete	**Nominal**	Permutation of values only. one hot encoding or hash function	If all **employee ID** numbers were reassigned, would it make any difference?
	Ordinal	Order must be preserved `new_value = f(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**	`new_value =f(old_value) + b` f is monotonic through origin 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	`new_value = f(old_value)` f is monotonic through origin float	**Length** can be measured in meters or feet, but **zero is zero**

from Tan et al. Introduction to Data Mining

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

Internal Rep.

TID							
1							
2							
3							
4							
5							
6	1	25.6	0	0	1	0	0

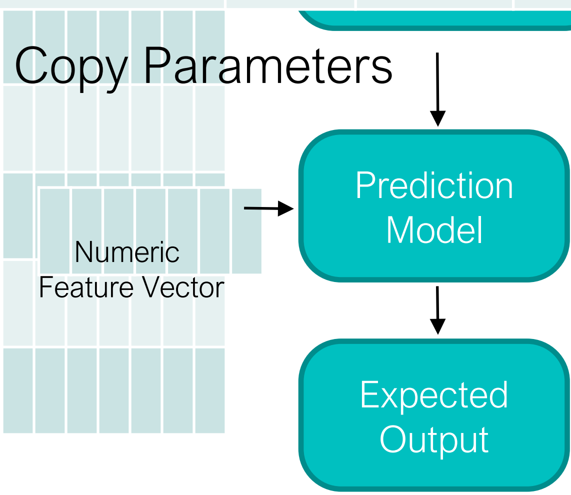
Opening Demo:
Jupyter Notebooks

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	no
3	Bl	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	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

TID	Pregnant	BMI	Age	Diabetes
1	Y	>32	41-50	positive
8	Y	>32	?	negative
		>32		

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

K-Nearest Neighbors Imputation

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

$$d_{i,j} = \frac{1}{|F_{valid}|} \sum_{f \in F_{valid}} \| f_i - f_j \|$$

For K=3, find 3 closest neighbors

TID	Preg.	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.

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 time: Documents, Data Imputation Demo