Computational Methods for Linguists

Ling 471

Olga Zamaraeva (Instructor)
Yuanhe Tian (TA)

04/27/21

Reminders

- Assignment 2 due today
- Assignment 3 published (due May 6)
- Please fill out midterm evaluations
  - time at the end of class today.

Plan for today

May 11	Basic Machine Learning and Naive Bayes	Regression and classification; Naive Bayes; Logistic regression (focus on the main idea)
May 13	Basic Deep Learning concepts and using pre-trained models	TBA

- More python tools
  - To start Assignment 3
- Evaluation metrics
  - Precision and recall
- Midterm evaluations
- **Next time:** Data science and probability
- **Next time:** Probability theory basics
  - Syllabus (topics and readings) updated
  - We'll see if we will push more things down to second part of May, too

May 18	Working with linguistic corpora	TBA
May 20	Working with linguistic corpora	TBA
May 25	Visualization and Communication	TBA
May 27	Visualization and Communication	TBA

From the class preliminary schedule.
We may push some of the statistics/ML/DL topics here.

More programming

Floats
and division

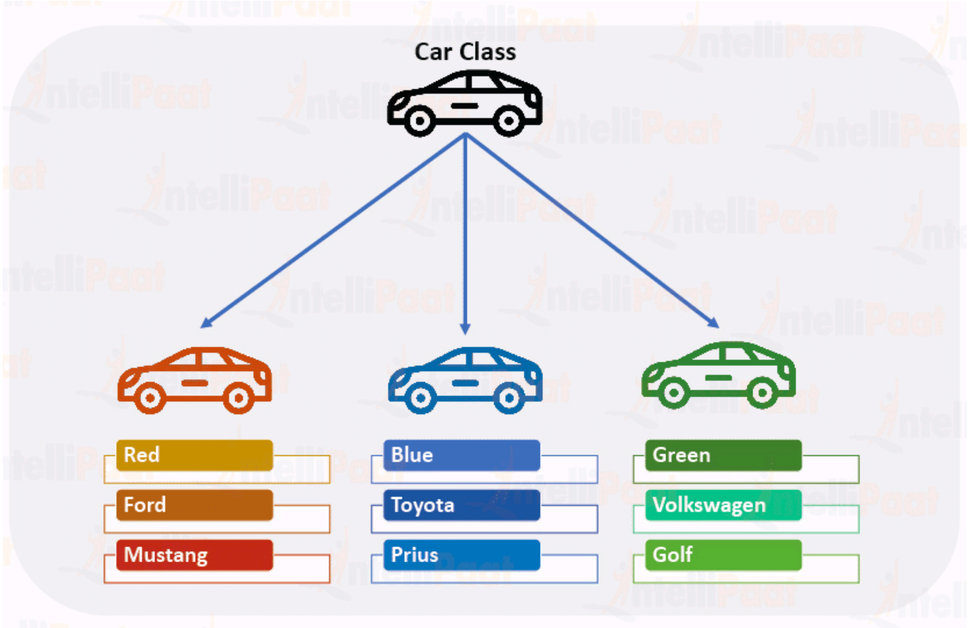
+	addition operator
—	subtraction operator
*	multiplication operator
/	division operator
//	integer divide operator
%	modulus (remainder) operator

http://www2.hawaii.edu/~takebaya/cent110/selection/operators.html

- Different types for numbers:
  - integer, float
    - **counts** are integers
    - **probabilities** are floats (ranging between 0 and 1)
  - floats require more storage…
    - it doesn't matter much anymore!
  - Integer division is different from float division
    - python has different operators for these
    - 5/2 will return 2.5
    - 5//2 will return 2
- How do you know what type of number you are dealing with?

Classes
and their instances

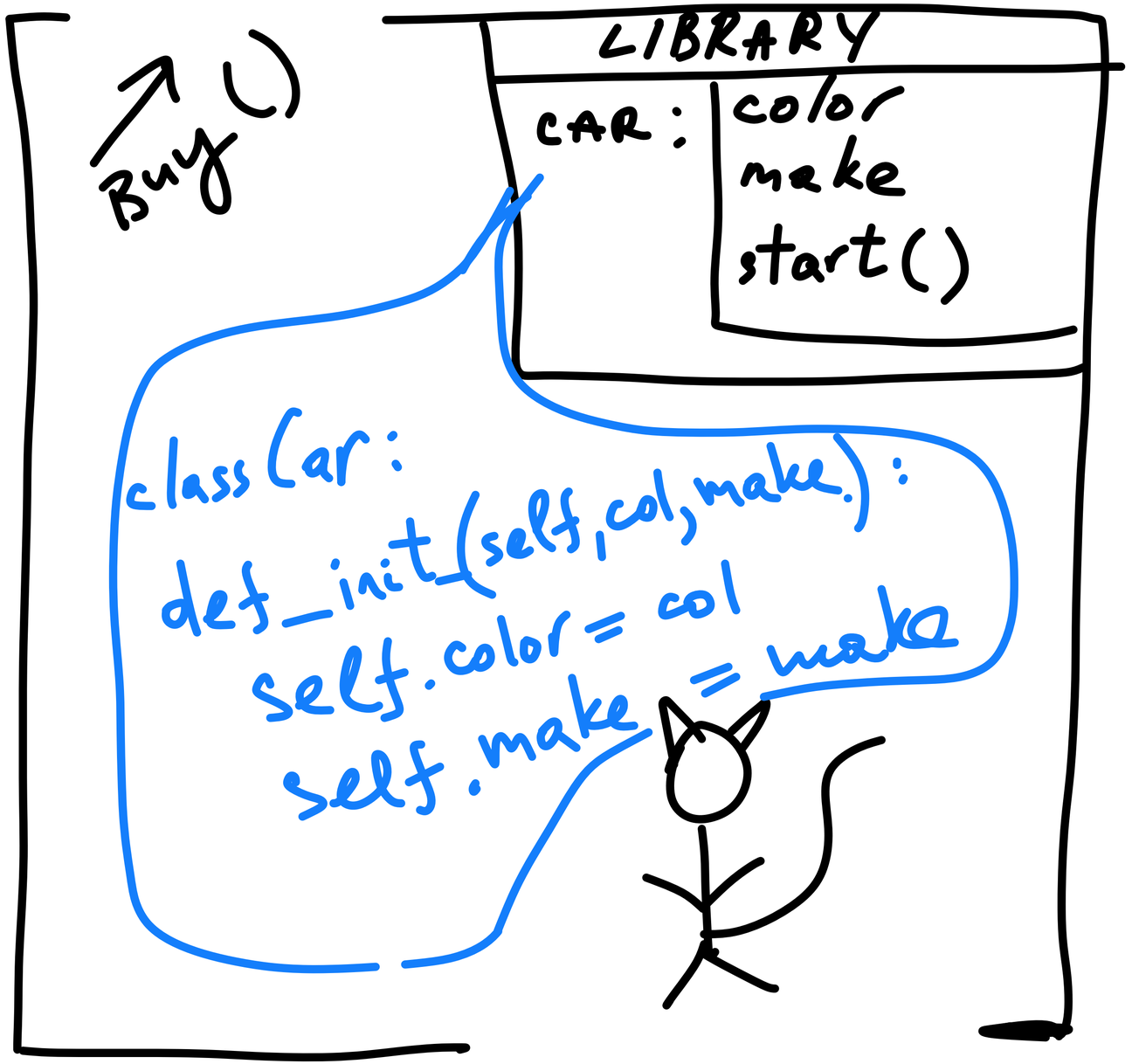
- **Classes** are like **blueprints** for objects
  - A class is an **abstract** collection of variables and functions
- **Instances** are **concrete** objects in memory
  - which have variable names
  - and have all the properties of the class
    - according to the blueprint
      - they are **specific** collections of variables and functions
    - NB: python is “dynamic” and allows you to add properties on the fly



https://intellipaat.com/blog/tutorial/python-tutorial/python-classes-and-objects/

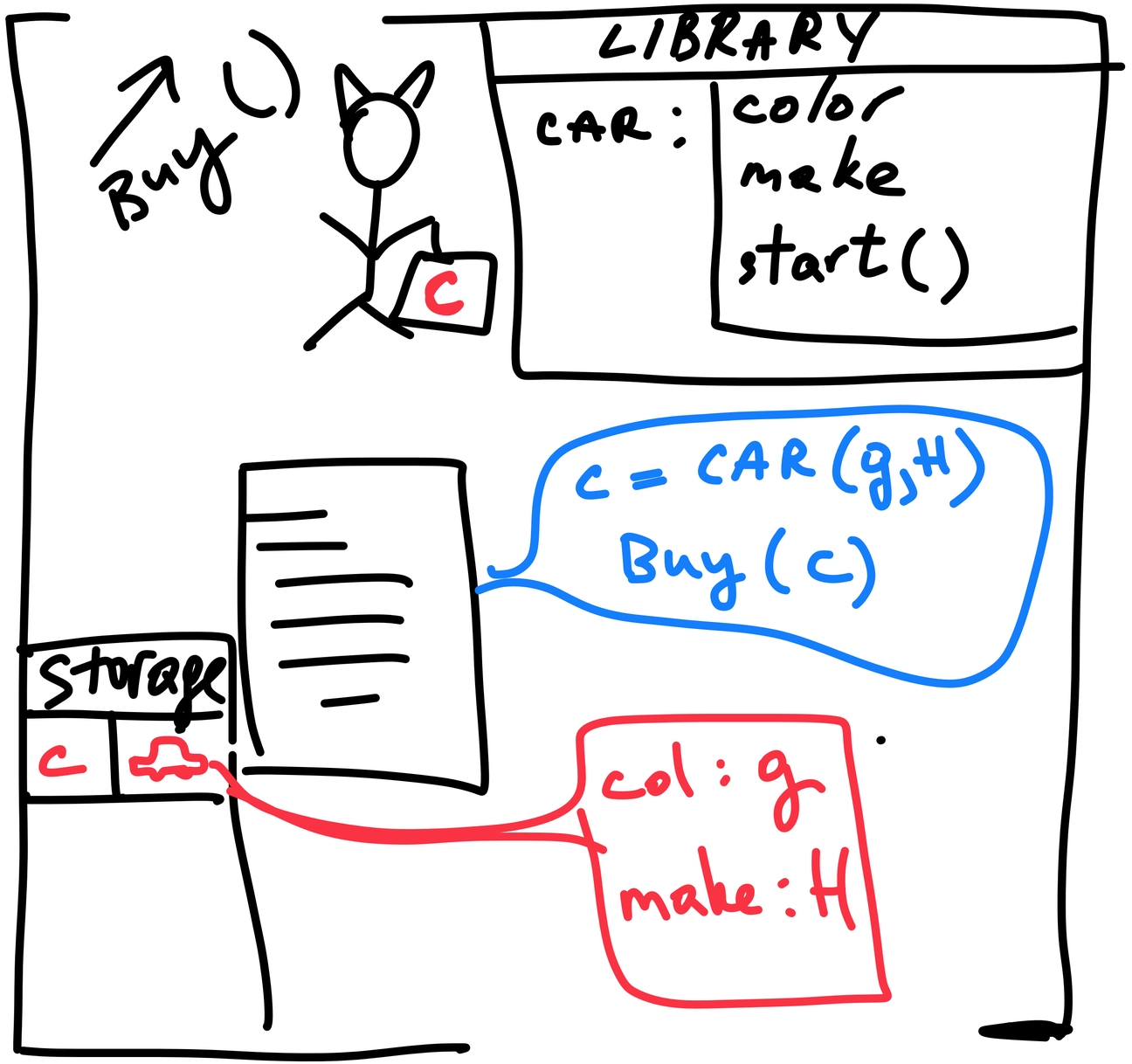
Classes
and their instances

- Classes have __init__() method
  - you can't change __init__()'s name
  - c = Car()
    - calls Car's __init__()
  - required args
    - c = Car("green", "Honda")
  - self is a special word
    - omitted when calling the initialization method
      - assumed by default

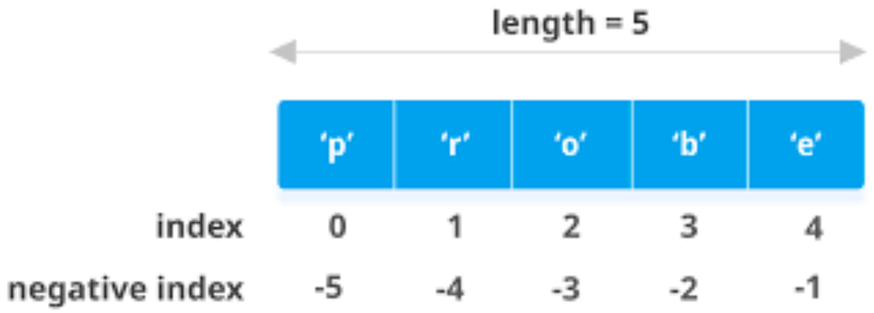


Classes
and their instances

- After creating an instance of Car named c, we can access:
  - c.col
  - c.make
  - and c.start()
  - We can also call other methods like Buy() on c.
  - We cannot call Buy on Car
    - but can on Car(g,H)
      - Buy (Car(g,h))
    - why?



Manipulating lists



https://www.programiz.com/python-programming/list

- Lists are a type of data structure
  - aka arrays
- Each element in a list has an index
  - elements are accessed by those indices
  - negative indices
    - -1 means the last element, etc.
  - list len()
    - length; a critical property of any list in any programming language
- In python:
  - lots of intricate ways of manipulating lists
  - for this class, the most relevant are:
    - access/reassignment, appending, iterating

Lists demo (in commnand line)

Pathlib

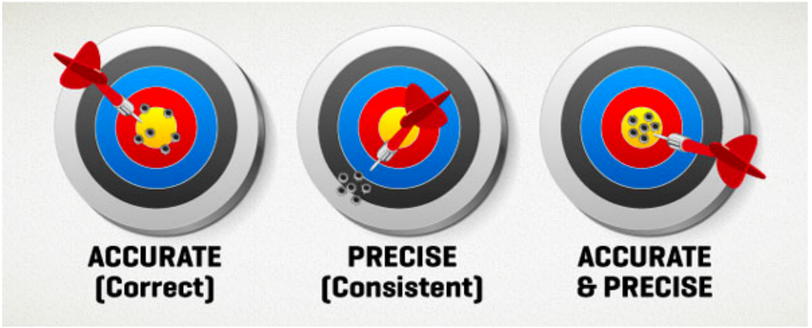
python module for filepaths

- Assignment 3:
  - Iterate over files in the directory
- A modern python library for robust handling of directories and fliepaths
- Many ways of doing it:
  - https://www.techiedelight.com/iterate-over-files-directory-python/

Pathlib demo

Evaluation metrics

Accuracy



https://itsadeliverything.com/accuracy-vs-precision-in-estimation

- How many data points out of total did the system **get right**?
  - e.g., 6 movie reviews:
    - Actual:
      - [POS,POS,POS,NEG,NEG,NEG]
    - System:
      - [POS,NONE,NEG,NEG,NEG,POS]
      - System's accuracy = 3/6 = 50%
- Accuracy can be computed when you don't care about **classes** of items
  - e.g. you **don't** care how good the system does in recognizing **bad** reviews in particular
  - you **only** care how well it did **overall**, wrt **all types** of reviews

“Accuracy paradox”

- Suppose you have System A and System B
  - Both systems output medical diagnoses:
    - POS (has the disease) or NEG (does not have it)
  - Which system has higher accuracy?

	System A		System B	
	neg	pos	neg	pos
neg	999	0	990	9
pos	1	0	0	1

“Accuracy paradox”

	System A		System B	
	neg	pos	neg	pos
neg	999	0	990	9
pos	1	0	0	1

- Suppose you have System A and System B
  - Both systems output medical diagnoses:
    - POS (has the disease) or NEG (does not have it)
  - Which system has higher accuracy?
    - System A:
      - Accuracy = 999/1000
    - System B:
      - Accuracy = 991/1000

“Accuracy paradox”

- Suppose you have System A and System B
  - Both systems output medical diagnoses:
    - POS (has the disease) or NEG (does not have it)
  - Which system has higher accuracy?
    - **System A:**
      - Accuracy = **999**/1000
    - System B:
      - Accuracy = 991/1000
  - Which system would you prefer in your hospital?

	System A		System B	
	neg	pos	neg	pos
neg	999	0	990	9
pos	1	0	0	1

System A was **unable** to diagnose the **ill** patient!

Precision and Recall

- Some systems need to be **accurate**
  - e.g. a part-of-speech tagger
- Some systems need more **precision**
  - e.g. voice assistant
    - want it to understand me correctly
    - don't care if it doesn't react sometimes
- Some systems need better recall
  - e.g. medical diagnoses
    - want to diagnose **ALL** patients who are sick **correctly**
    - care **less** about those who I said were sick but they are actually healthy

- Context: ~~Object~~ apple 🍎 **retreival**
  - Array of objects: [0, 1, 2,3, 4,5, 6, 7]
  - Ground Truth: [🍎🍎🍎🍎🍊🍊🍊🍊]
  - System A: [🍎🍊🍎🍎🍊🍊🍊🍊]
    - perfect precision, but not recall
  - System B: [🍎🍊🍊🍊🍊🍊🍊🍊]
    - Still perfect precision, but poor recall
  - System C: [🍎🍎🍎🍎🍎🍎🍎🍎]
    - perfect recall, but poor precision

Precision and Recall

- How well does the system do wrt **relevant** items?
  - If you have **different types of items** and want to know how well the system does wrt **each**, separately
  - e.g. how good is my system at recognizing **bad** reviews?
    - may be important for business considerations
    - ethical considerations
      - Is the system biased to do well on one class of items?

- Context: ~~Object~~ apple 🍎 **retreival**
  - Array of objects: [0, 1, 2,3, 4,5, 6, 7]
  - Ground Truth: [🍎🍎🍎🍎🍊🍊🍊🍊]
  - System A: [🍎🍊🍎🍎🍊🍊🍊🍊]
    - perfect precision, but not recall
  - System B: [🍎🍊🍊🍊🍊🍊🍊🍊]
    - Still perfect precision, but poor recall
  - System C: [🍎🍎🍎🍎🍎🍎🍎🍎]
    - perfect recall, but poor precision

Metrics tensions

- An **ideal** system:
  - Is accurate, precise, and has high recall
- In **reality**:
  - Precision and Recall are in **tension** with each other
    - Why?
      - It is **trivial** to have a system which has 100% precision and unacceptably low recall (and vice versa)

- Context: ~~Object~~ apple 🍎 **retreival**
  - Array of objects: [0, 1, 2,3, 4,5, 6, 7]
  - Ground Truth: [🍎🍎🍎🍎🍊🍊🍊🍊]
  - System A: [🍎🍊🍎🍎🍊🍊🍊🍊]
    - perfect precision, but not recall
  - System B: [🍎🍊🍊🍊🍊🍊🍊🍊]
    - Still perfect precision, but poor recall
  - System C: [🍎🍎🍎🍎🍎🍎🍎🍎]
    - perfect recall, but poor precision

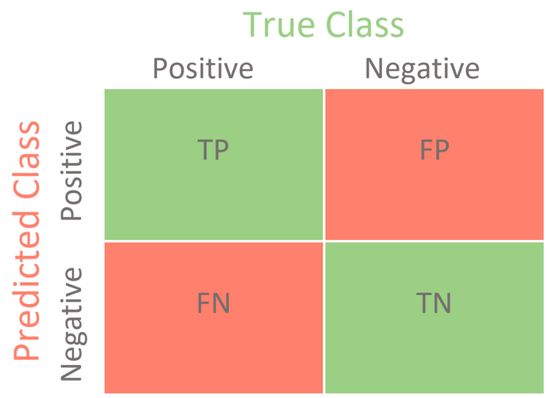
Confusion Matrix

visualizing statistics

- Context: ~~Object~~ apple 🍎 **retreival**
  - Array of objects: [0, 1, 2,3, 4,5, 6, 7]
  - Ground Truth: [🍎🍎🍎🍎🍊🍊🍊🍊]
  - Our System: [🍎🍊🍎🍎🍊🍊🍎🍊]
- Reference table for the **four types of label**
- **True Positive:** 0,2,3
- **False Positive:** 6
- **True Negative:** 4,5,7
- **False Negative:** 1
- Compute Precision and Recall as per **definitions**

	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal ✉)	
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉✉ **320**	FALSE NEGATIVE (FN) ✉ **43**	*Recall* $= \frac{TP}{TP+FN} = \frac{320}{320+43} = 0.882$
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉✉ **20**	TRUE NEGATIVE (TN) ✉ **538**	
	Precision $= \frac{TP}{TP+FP} = \frac{320}{320+20} = 0.941$		

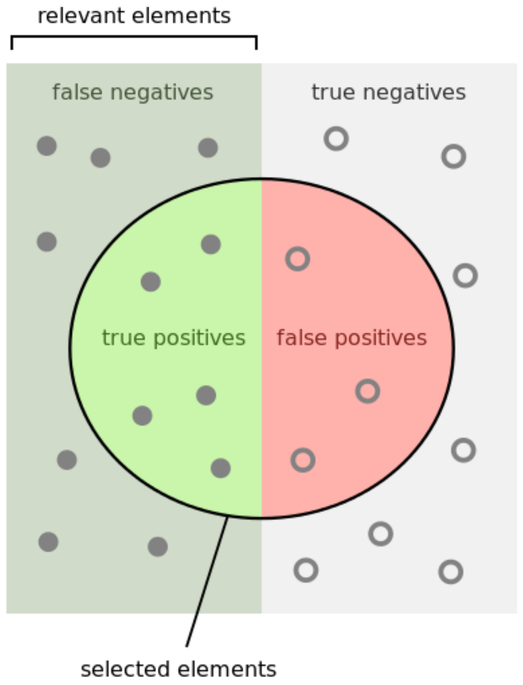
https://www.knime.com/blog/from-modeling-to-scoring-confusion-matrix-and-class-statistics

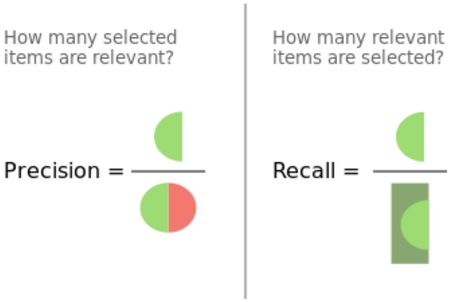


https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Computing Precision and Recall
for “positive” and “negative” movie reviews

- Assignment 3
- The use of words “positive”/“negative” is **confusing** 🤯
  - The name of the **review type** (“positive”/“negative”) has **nothing** to do with the **error types** (e.g. “false positive”/“false negative”
  - Try thinking of reviews as “good”/“bad” instead
    - rename variables accordingly
    - compute carefully, wrt **each** class
    - **Test** on the tiny datasets provided!
      - If you don’t test, you will likely **not** get correct results

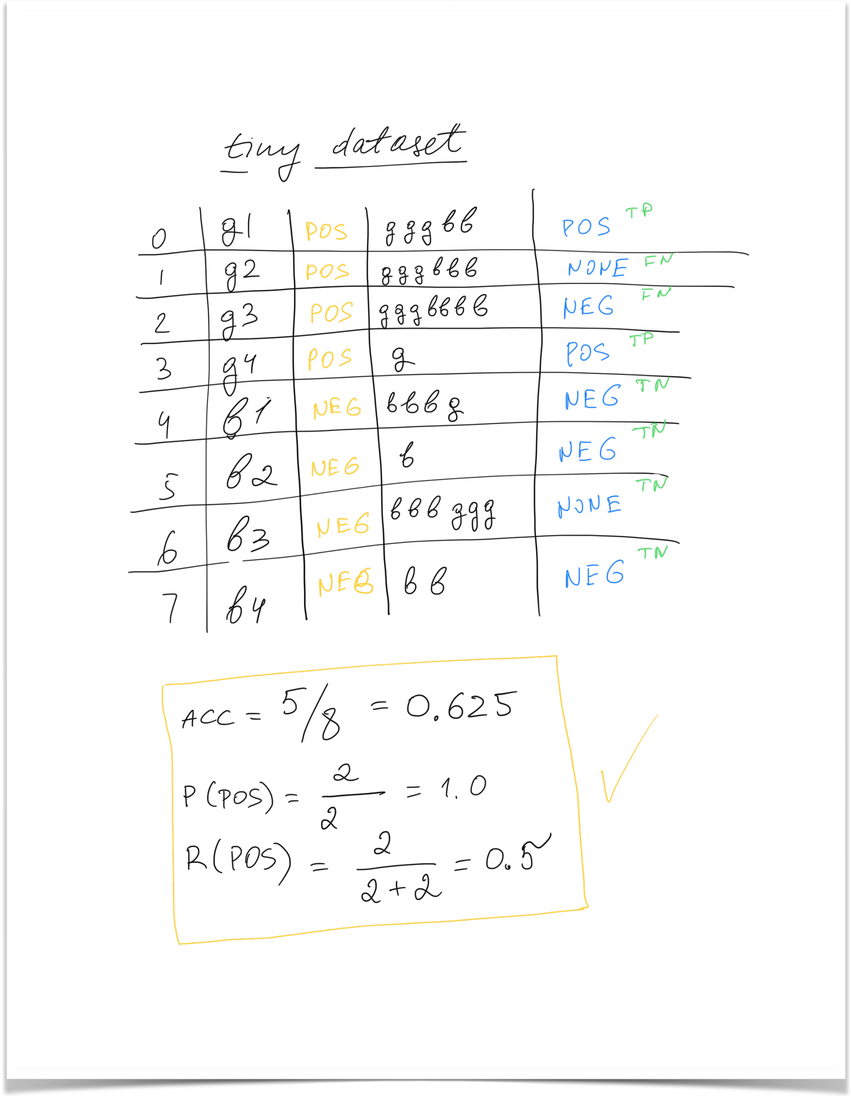




https://en.wikipedia.org/wiki/Precision_and_recall

Tiny dataset

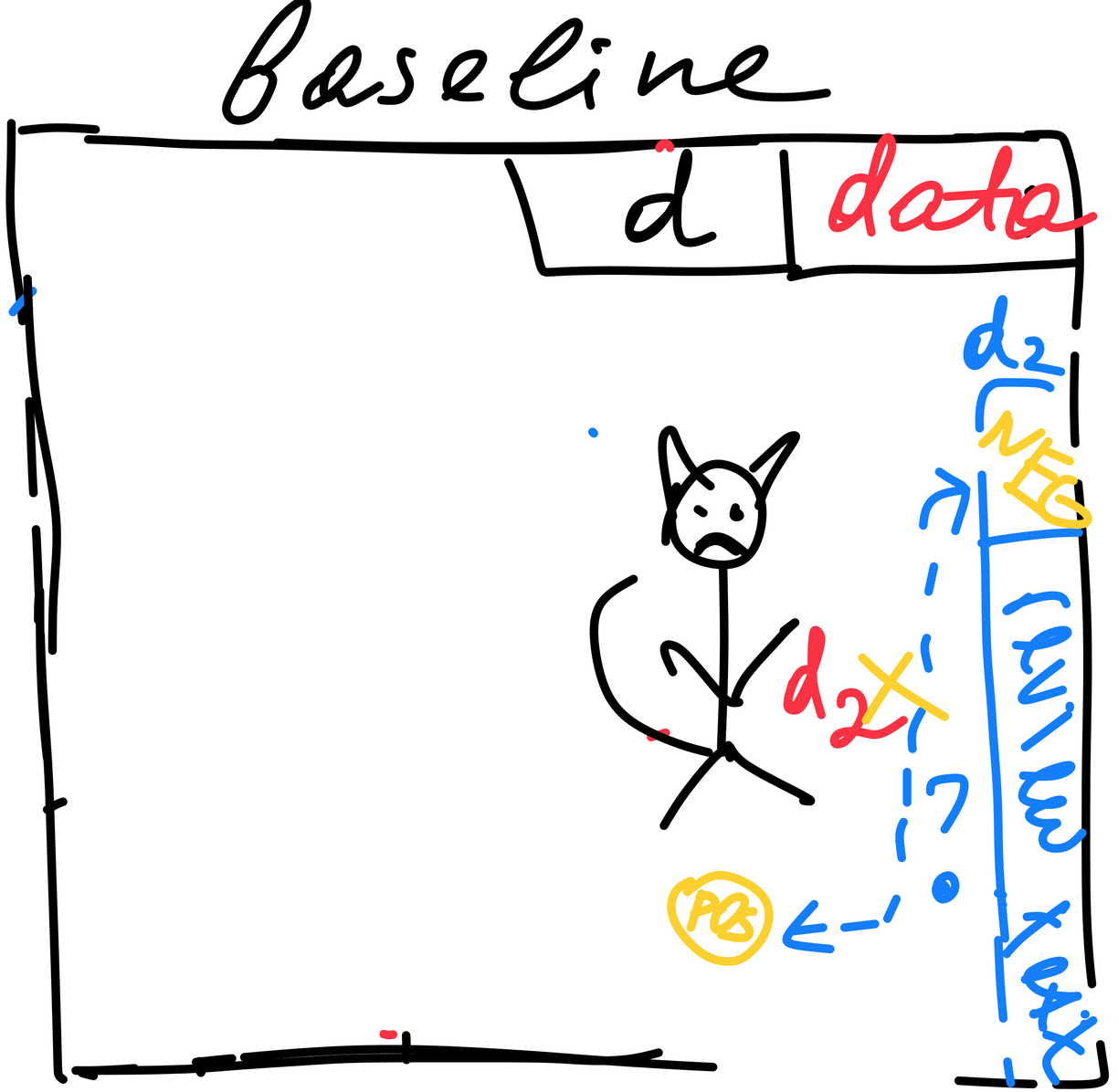
- Comes with Assignment 3 repo
- **Do** use it instead of the main dataset at first
  - You don't want to manually debug thousands of files if your numbers are wrong
- Use tiny dataset to **make sure** you **understand** how to compute P&R



Error Analysis

Error Analysis

- Evaluate the system
- Inspect data points which it gets wrong
- Make conclusions for further development
  - and for current deployment!



Error Analysis

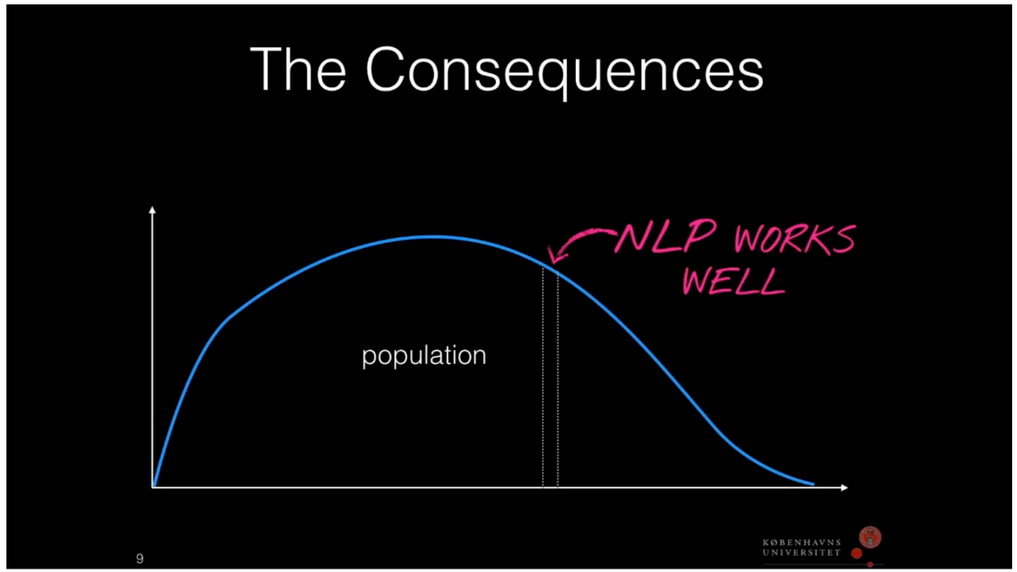
- Evaluate the system
- Inspect data points which it gets wrong
- Make conclusions for further development
  - and for current deployment!

Error Analysis

- Evaluate the system
- Inspect data points which it gets wrong
- Make conclusions for further development
  - and for current deployment!

Error Analysis
and computer science

- Why is there little error analysis in NLP?
- Recall from March 30 lecture:
  - NLP emphasis on “raw data”
    - “because computer science”
      - actually, because would like to delay “the WSJ effect”
- Computer science aspires for full automation
- EA usually involves “manual work”
- Computer science like “generic data”
  - should not matter what the data are like!



https://hch19.cl.uni-heidelberg.de/program/slides/l/HCH19_lecture_Dirk_Hovy.pdf

In your presentations, talk about any EA that the authors did, or what they could do, if they didn’t do any.

This lecture survey: chat window

Please fill out Midterm Course Evaluations!