

Machine Learning for Behavioral Data (CS-421)

Introduction

February 20, 2023

Today

- What is ML for Behavioral Data?
- Course Logistics
- Active Learning
- Projects: EdTech StartUp(s)

Today

- What is ML for Behavioral Data?
- Course Logistics
- Active Learning
- Projects: EdTech StartUp(s)



This will be an interactive course...

- More on this later
- For now: take your phone (or laptop) and join us on SpeakUp

<https://go.epfl.ch/speakup-mlbd>



About Me

- Assistant professor at EPFL since May, 2020
 - Head of the ML4ED lab
 - In the past, I was a
 - senior data scientist at the SDSC
 - postdoc at Stanford University
 - postdoc at ETH Zurich/consultant for Disney research Zurich
 - PhD student at ETH Zurich
-

Students – Shake Hands



What is ML for Behavioral Data?

SpeakUp Chat!

0
0 votes

14/02/2022 21:25, by me

0 comments

The image shows a single message card from a 'SpeakUp Chat' platform. The message content is 'What is ML for Behavioral Data?'. It was posted on 14/02/2022 at 21:25 by the user 'me'. The message has 0 votes and 0 comments. There are icons for thumbs up and thumbs down on either side of the message.

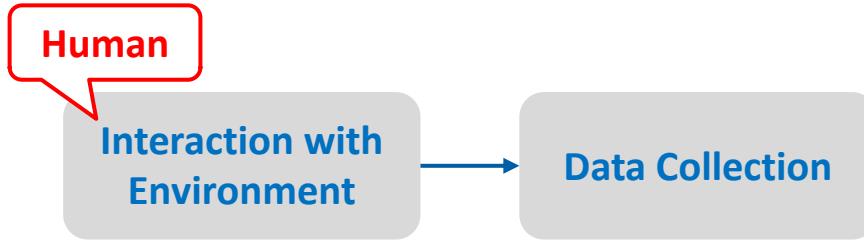
What is ML for Behavioral Data?

Human

Interaction with
Environment

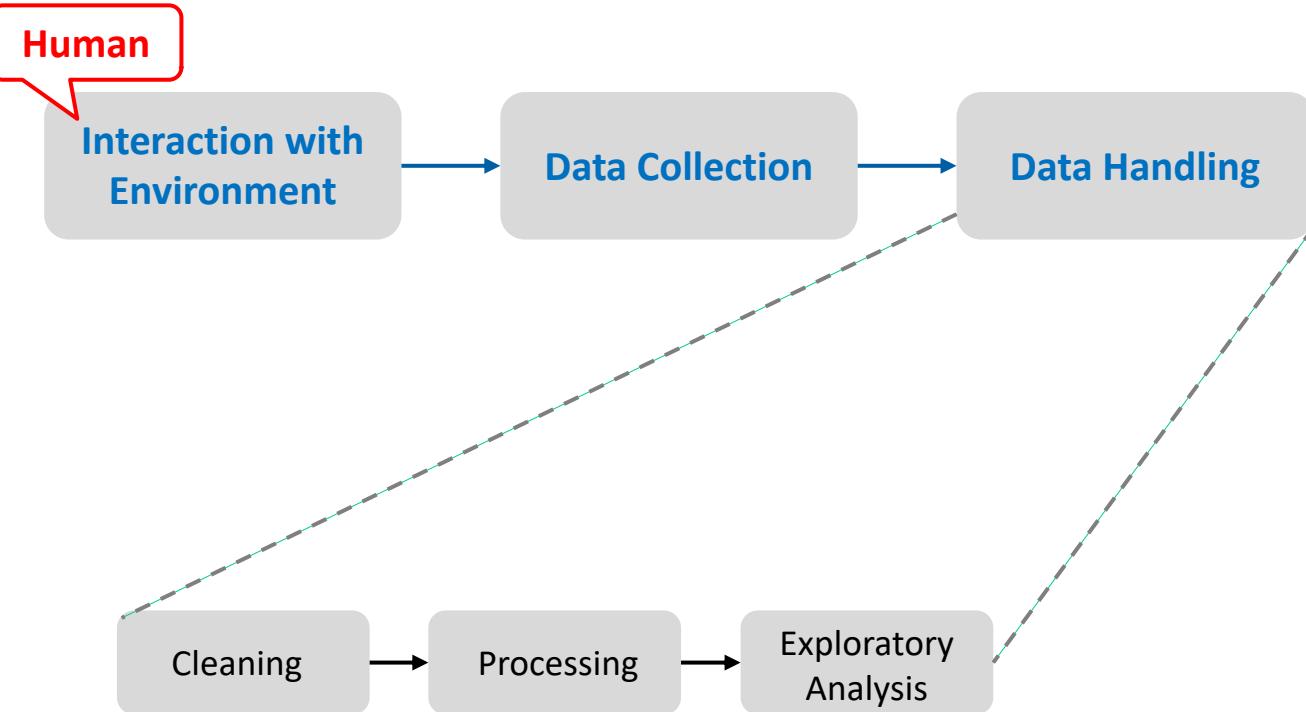


What is ML for Behavioral Data?

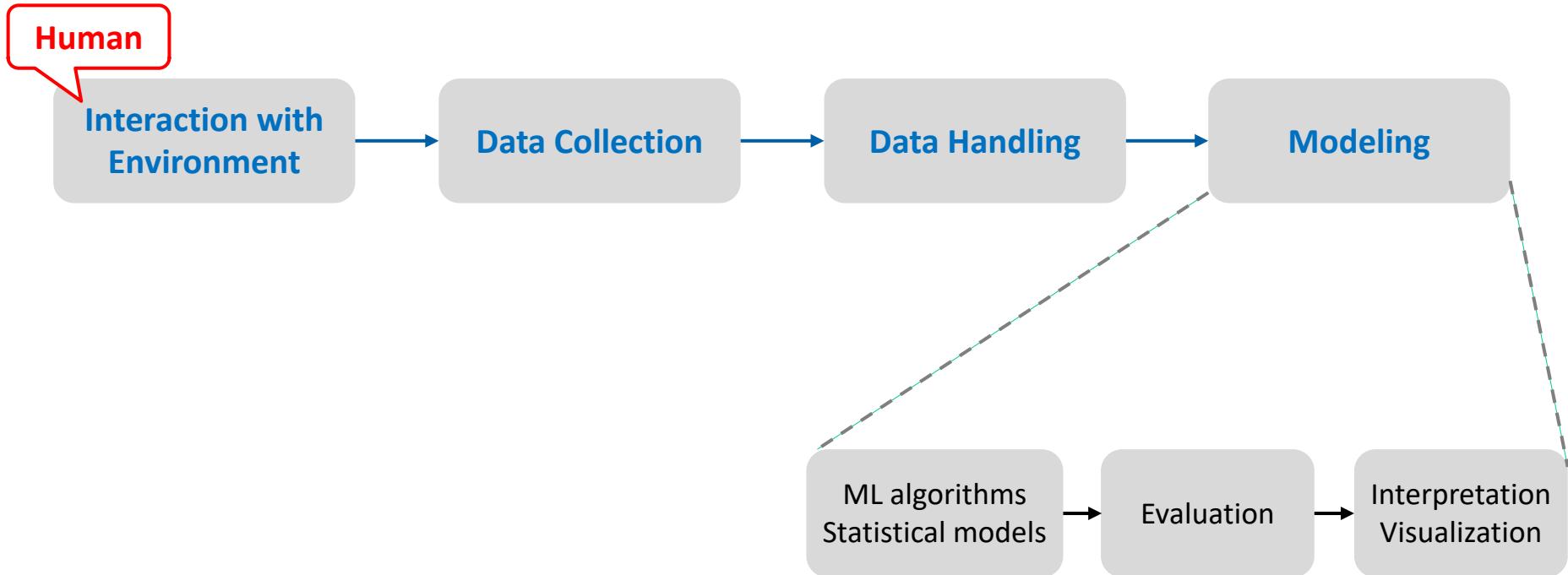


- Clickstream
- Text
- Categorical Data
- Images
- Video
- Sensor Data
- ...

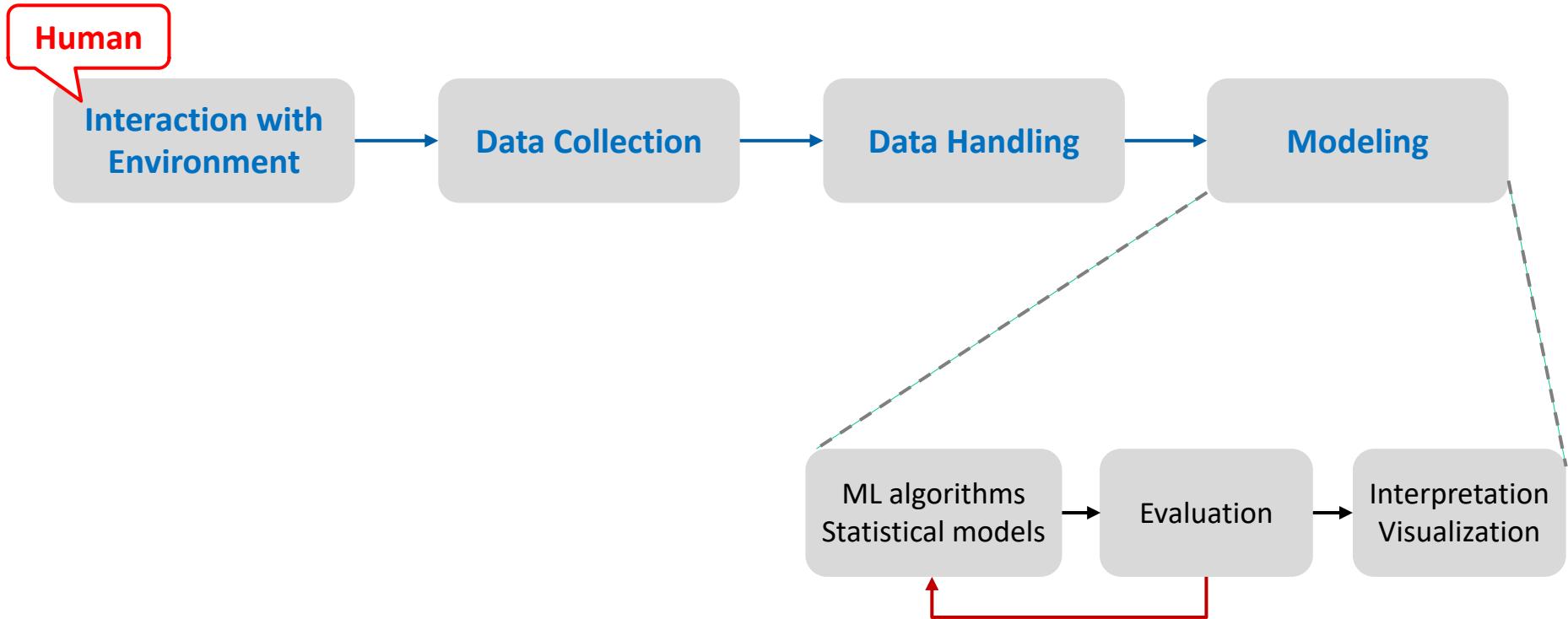
What is ML for Behavioral Data?



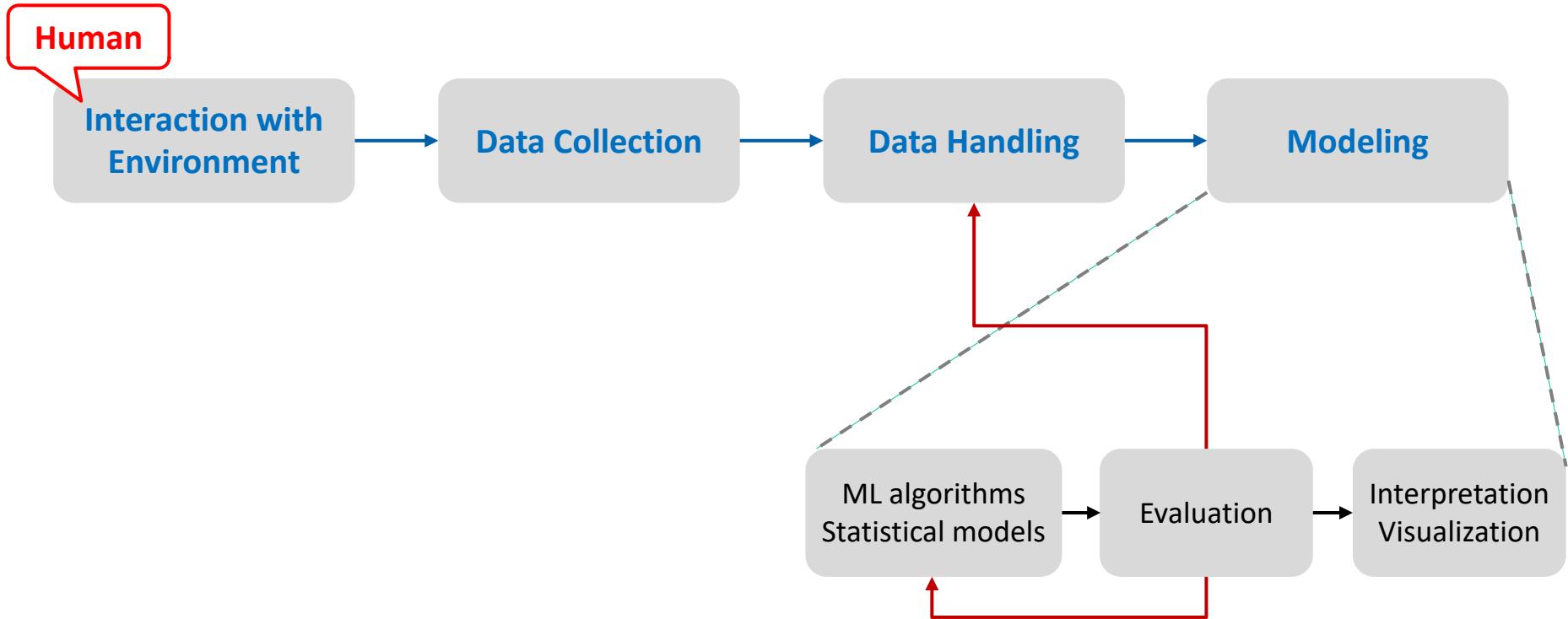
What is ML for Behavioral Data?



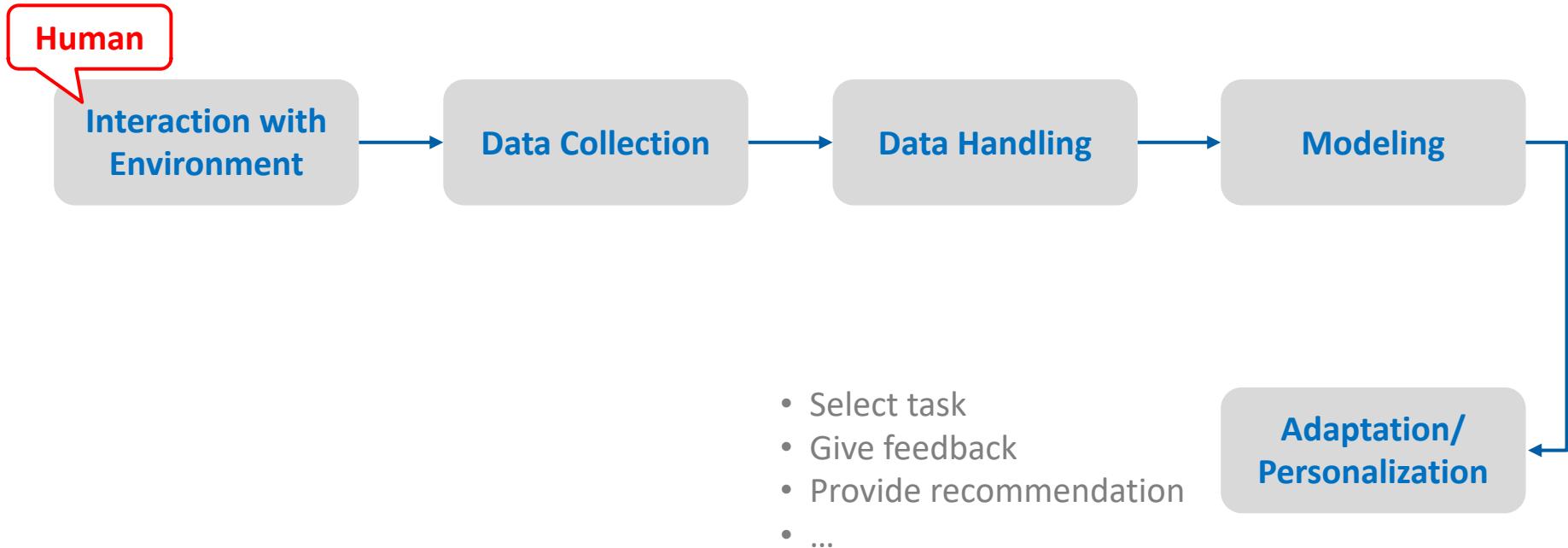
What is ML for Behavioral Data?



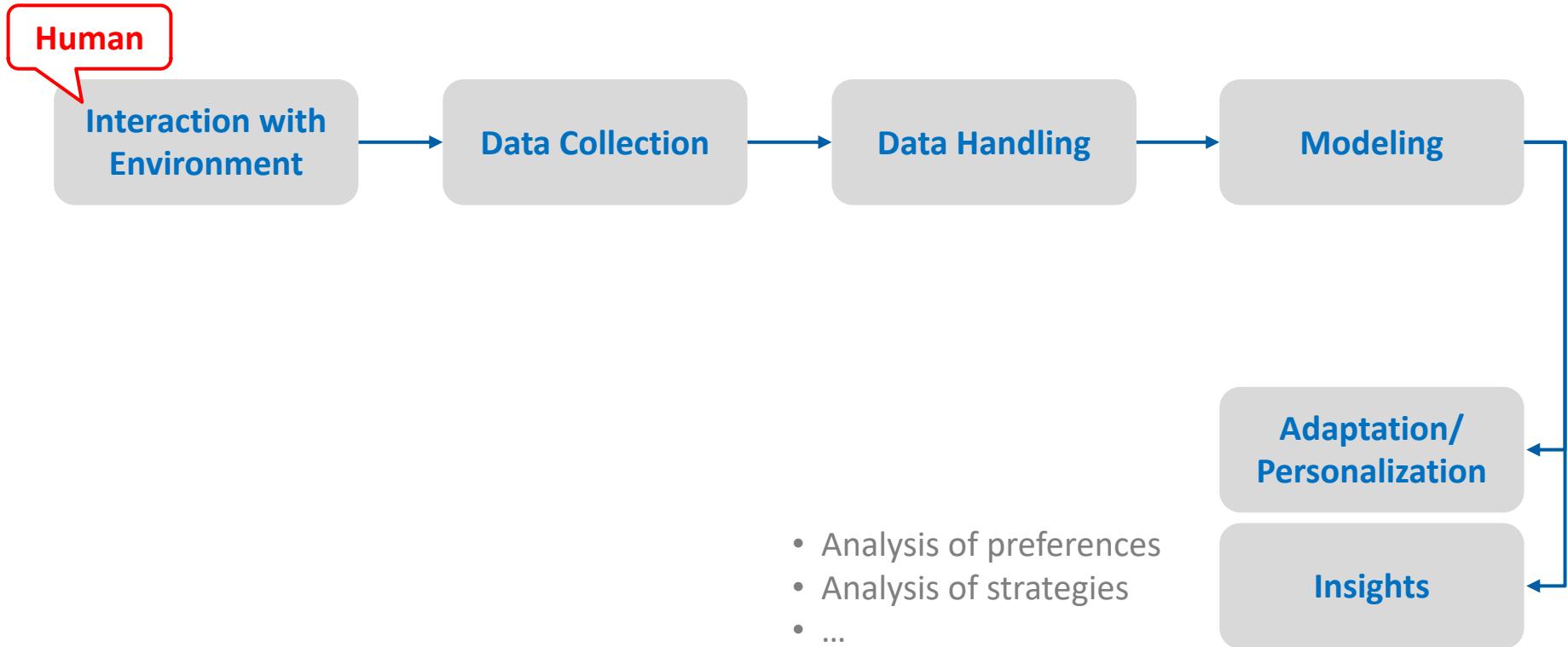
What is ML for Behavioral Data?



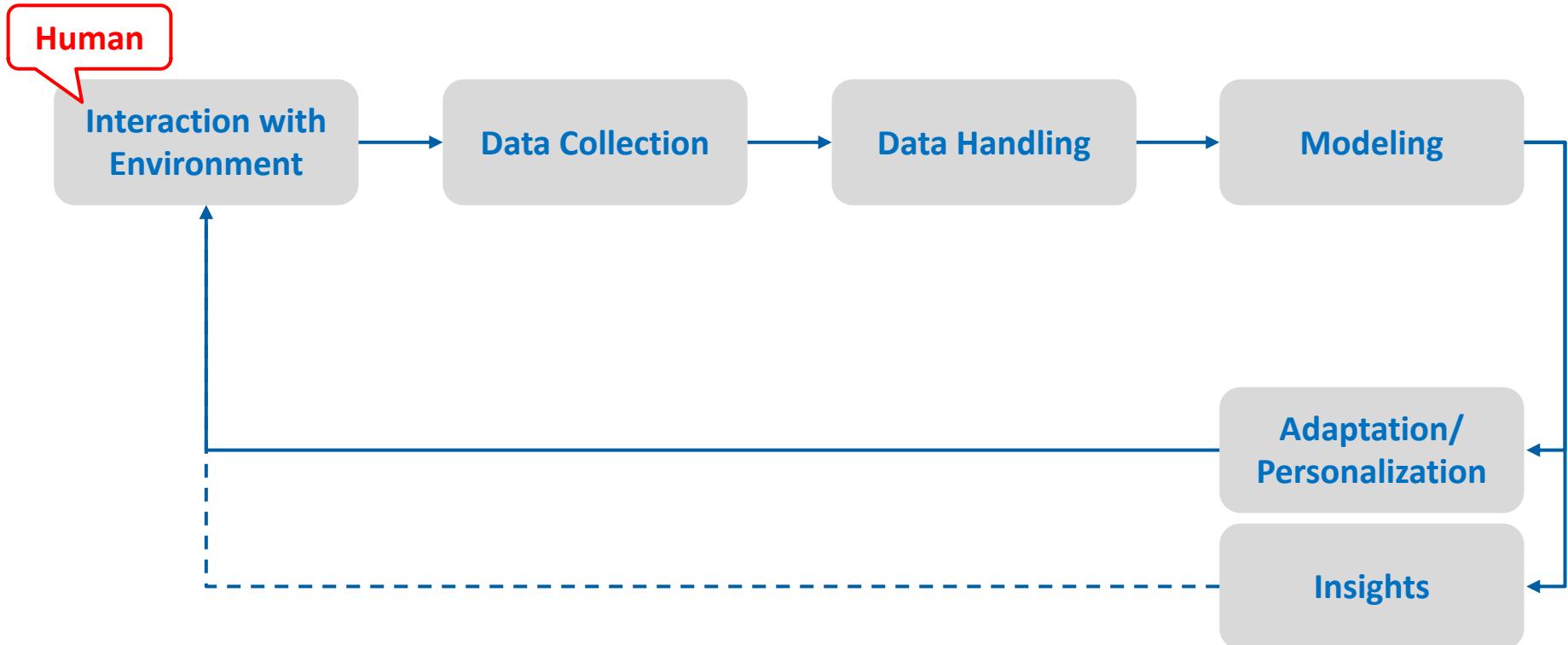
What is ML for Behavioral Data?



What is ML for Behavioral Data?

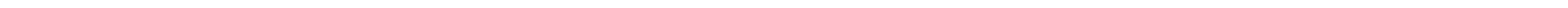


What is ML for Behavioral Data?



Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction



Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

- Exploring & visualizing data
- Time Series Exploration

Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

- Generalized Linear Models
- Mixture Models
- Regression for time series

Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

- Random Forest, nearest neighbors, etc.
- Classifying time series data

Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

- Cross validation, bootstrap, information scores
- Error metrics & visualization

Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

Complete pipeline for one use case:

- Data exploration
- Prediction
- Model evaluation

Lecture Syllabus

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction

Supervised learning on time series:

- Probabilistic graphical models
- Neural networks: LSTM, GRU, etc.

Lecture Syllabus

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Ethical Machine Learning
13	Ethical Machine Learning
14	Project Presentations
15	Whit Monday

Lecture Syllabus

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Ethical Machine Learning
13	Ethical Machine Learning
14	Project Presentations
15	Whit Monday

- 
- K-Means, Spectral Clustering
 - Choosing the optimal K*
 - Clustering time-series data

Lecture Syllabus

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Ethical Machine Learning
13	Ethical Machine Learning
14	Project Presentations
15	Whit Monday

- 
- Fairness
 - Explainability

Lecture/Lab

- Monday, 15:15 – 18:00
 - INR 113
 - Lecture + practice session
 - Slides will be uploaded to our GitHub
 - Jupyter Notebooks will be uploaded to our GitHub
 - Recording: we will make the recordings from the past year available
-

Project

- Teams of 3 people
 - We will provide the data sets (from EdTech Start-Ups)
 - We will provide example research questions
 - You will suggest an additional analysis/extension to the selected research question
 - We will give feedback during the semester (see milestones)
 - You will do a **poster presentation** in the penultimate week of the semester
 - Final project (Code + Report) delivered by **June 9, 2023 23:59 CET**
-

Project (Office) Hours

- Wednesday, 9:15-10:00
- INM 10
- Content:
 - Introduction to project tasks
 - Individual feedback meetings with teams
 - Drop-in office hours for questions regarding the lecture or project

Project Schedule

Week	Project Hours	Milestones
1	Environment setup	-
2	Introduction to tasks for M2	<i>M1: preferences on team members and start-up</i>
3	Office hours	
4	Introduction to tasks for M4	<i>M2: individual exploration of selected data set</i>
5	Office hours	<i>M3: selection of research question and approach</i>
6	Individual discussion with teams	
7	Office hours	
8	Spring Break	

Project Schedule

Week	Project Hours	Milestones
9	Team Coaching	
10	Office hours	<i>M4: submission of results for first research question M5: ideas for extension (+ approach)</i>
11	Individual discussion with teams	
12	Office hours	
13	Team Coaching	
14	Poster Presentations	<i>M6: poster session (in person, on campus)</i>
15	Office Hours	
16		<i>M7: Hand in report and code base</i>

Grading

- **50% Project**
 - Teams of 3 people
 - All milestones are mandatory
 - All individual feedback meetings and team coaching meetings are mandatory
 - 15% individual exploration (M2), 25% supervised learning (M4), 20% presentation (M6), 40% final results (M7)
 - **50% Final Exam (exam session)**
 - Individually, at the laptop
 - Mix of conceptual and coding questions
-

Course Goals

- Explain the main machine learning approaches to personalization, describe their advantages and disadvantages and explain the differences between them
 - Implement algorithms for these machine learning models
 - Apply them to real-world data
 - Assess / evaluate their performance
-

Which ML courses have you taken?

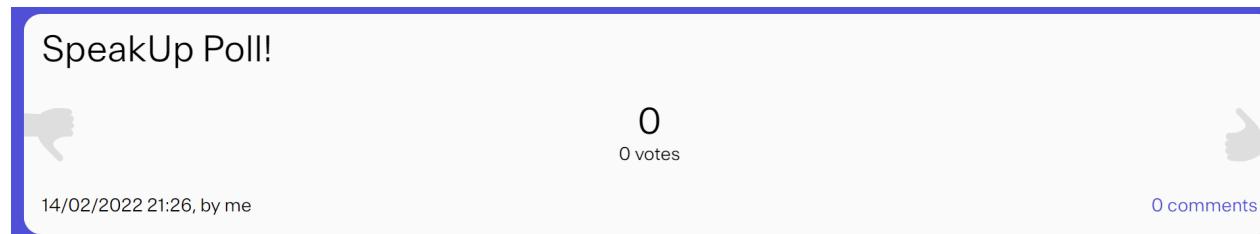
- A. Introduction to Machine Learning
- B. Machine Learning
- C. Applied Data Analysis
- D. Other

SpeakUp Poll!

0
0 votes

14/02/2022 21:26, by me

0 comments



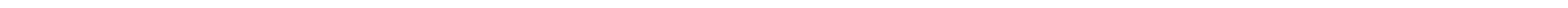
Course Prerequisites

- Probabilities and statistics
- Programming:
 - Project: Python
 - Exam: Python
- Foundations of machine learning



Important Websites

- Moodle: <https://moodle.epfl.ch/course/view.php?id=16434>
 - Contains all important information
 - Use forum for questions
 - For more personal questions contact teaching assistants
- Project:
 - GitHub: <https://github.com/epfl-ml4ed/mlbd-2023>
 - EPFL Noto: <https://noto.epfl.ch/>



Team

Instructor

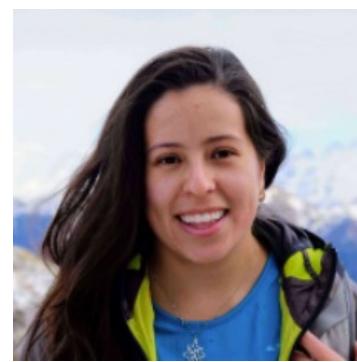


Tanja Käser
tanja.kaeser@epfl.ch

Teaching Assistants



Vinitra Swamy, Paola Mejia
vinitra.swamy@epfl.ch, paola.mejia@epfl.ch



Feedback

- We are committed to providing the best possible version of the course
- If you want to give us feedback, there will be a link on Moodle:

Feedback

We are fully committed to providing the best possible version of the course and we appreciate all constructive feedback.
We are looking forward to reading your comments and improving based on them.

[Feedback link \(anonymous\)](#)

Questions?



Today

- What is ML for Behavioral Data?
- Course Logistics
- **Active Learning**
- Projects: EdTech StartUp(s)

Active learning – what is it?

SpeakUp Chat!

0
0 votes

14/02/2022 21:25, by me

0 comments

The image shows a screenshot of a digital platform called "SpeakUp Chat". At the top left, the text "SpeakUp Chat!" is displayed. In the center, there is a message card with a blue border. The message content is "Active learning – what is it?", which is identical to the main title of the slide. Below the message text, there are two small icons: a thumbs-down icon on the left and a thumbs-up icon on the right. In the center of the card, the number "0" is prominently displayed above the text "0 votes". At the bottom left of the card, the timestamp "14/02/2022 21:25, by me" is visible. At the bottom right of the card, the text "0 comments" is present. The entire screenshot is set against a white background with a thin red horizontal line at the very bottom.

Active learning – what is it?

- Activities that students do to construct knowledge and understanding
 - Read
 - Write
 - Explore
 - Discuss
 - ...

Active learning in this course

SpeakUp

Collecting Ideas

Polls

Think – Pair - Share

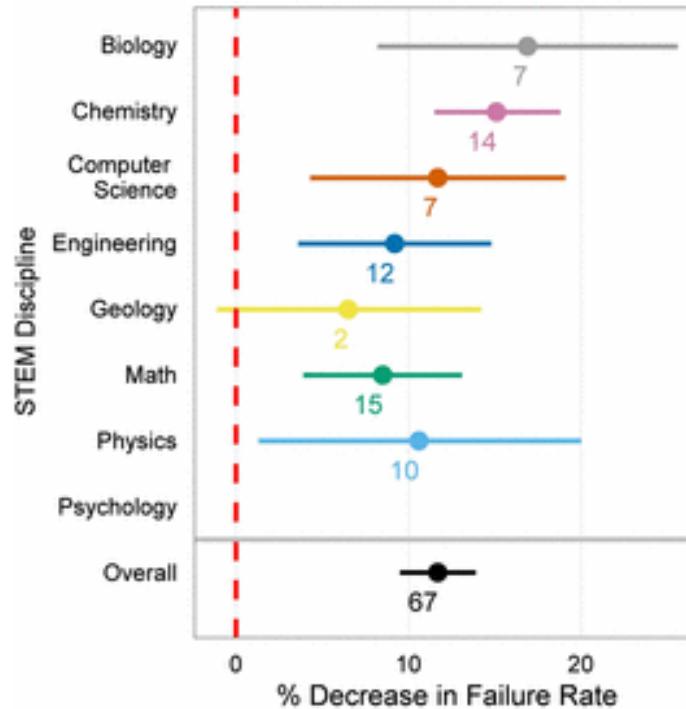
Jupyter Notebook

Demonstration

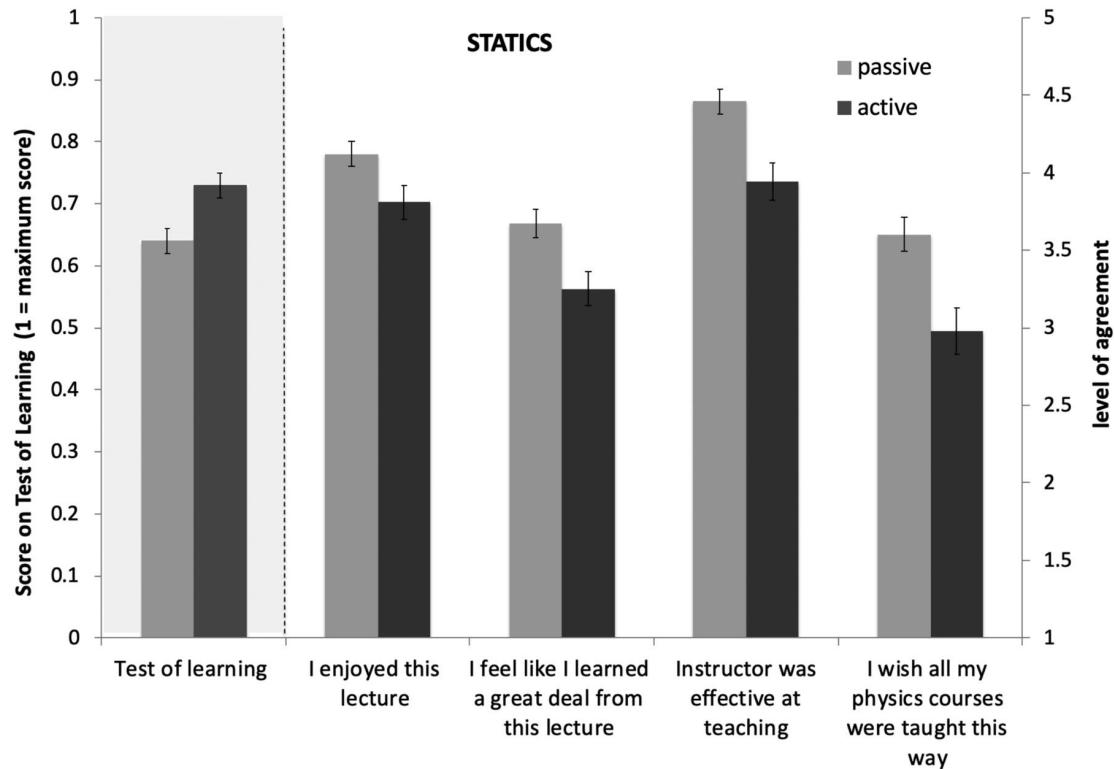
Worked examples

Small coding tasks

Active learning increases performance



Watch out: Feeling-of-Learning can deceive you!



The lecture will be interactive, thus

- we expect you to attend the lecture
- we expect you to participate in all the activities

Important: bring your laptop !

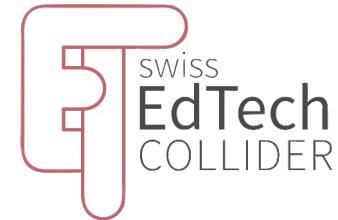
Questions?



Today

- What is ML for Behavioral Data?
- Course Logistics
- Active Learning
- **Projects: EdTech StartUp(s)**

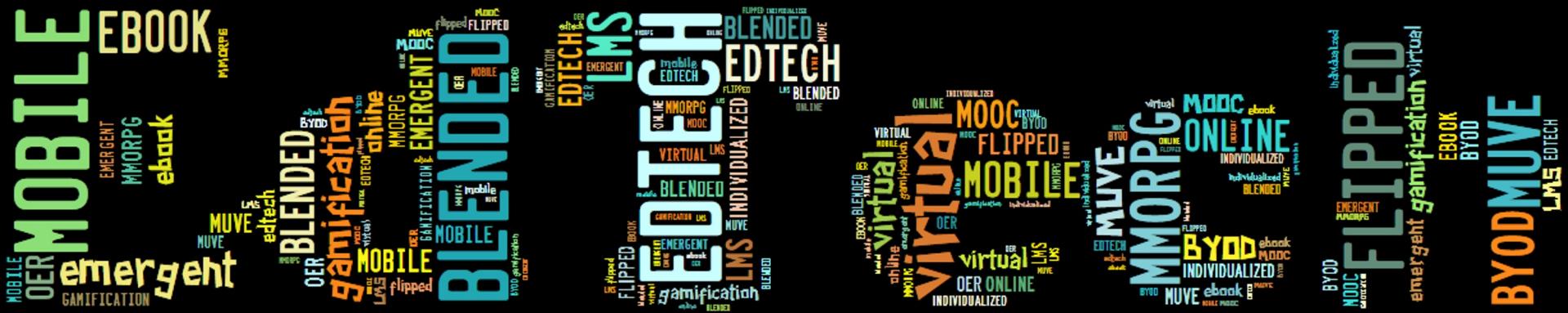
Swiss “EdTech” Hub



Why a Swiss “EdTech” Hub?

To support the digital transformation in **ED**ucation
with **TECH**nological solutions

EdTech Market – Highly Fragmented



Large diversity in the use of technology-enhanced solutions



Early Childhood Education



Compulsory Education



**Upper Secondary /
Higher Education**



University/VET



**Continuous Training &
Education**



Corporate Training & Learning

Mission and Vision

- Bring players in EdTech together in one place in order to create a market place around Education and EdTech
- Focus on future learning solutions / future skills
- Long-term partnerships (not a short-term incubator)
- Help to accelerate growth
- Sustainability / long term positive impact on society

EdTech Collider - Facts



ECOSYSTEM

Establishing a unique ecosystem around EdTech



PROXIMITY EPFL

Close to EPFL Research (Digital Learning / Learning Sciences)

LEARN - Center for Learning Sciences EPFL

SPACE

Physical co-working space / Virtual space



Independent Association/NPO (04/17)

Four EPFL Professors

93+ Members (EdTech startups)

Support: EPFL, Jacobs Foundation, Credit Suisse

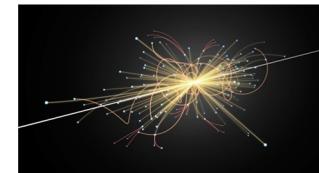
VISIBILITY

Create an image / branding and reputation Swiss EdTech Collider



COLLISIONS

Nourish the 'collision' of knowledge / sharing ideas / creating co-operations / collaborations (members/external)



+175 Collisions/Events

93 StartUps (status: 01.2022)



Follow Us

Twitter: [@SwissEdTech](#)

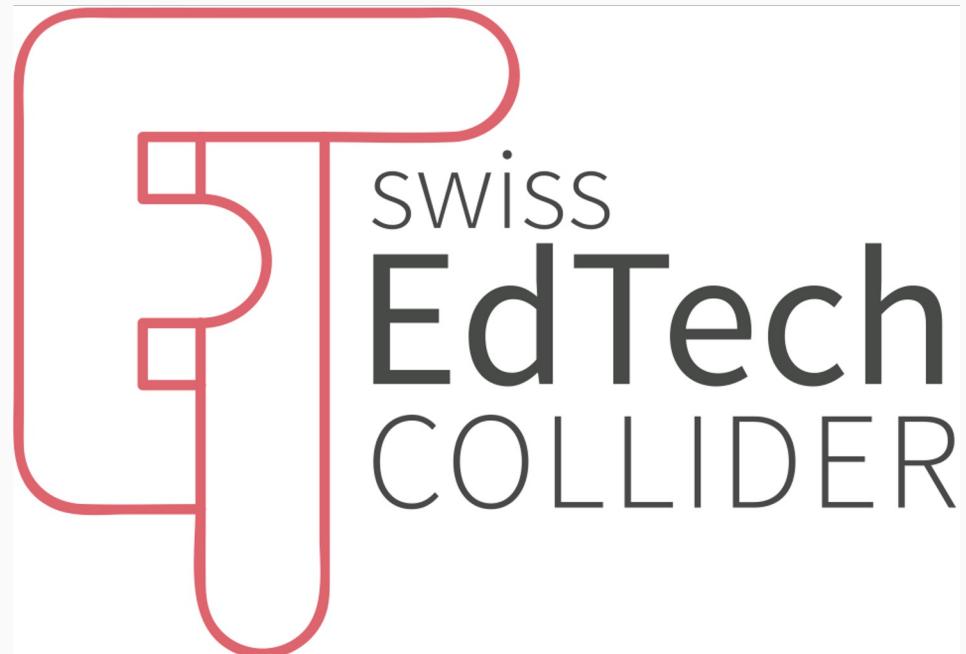
LinkedIn: [Swiss-EdTech-Collider](#)

Instagram: [Swiss_EdTech_Collider](#)

Facebook: [SwissEdTech](#)

contact@edtech-collider.ch

www.edtech-collider.ch



The two participating StartUps

- Dybuster Alemira (Marco Bär)
- Taskbase (Anette Hunziker)



Up next...

- Detailed information regarding the project: milestones, guidelines, grading, data sets, etc. [lab session today]
- Setting up GitHub and Jupyter notebook for the lecture and project [lab session on Wednesday]

Remember

- Register for the course on IS Academia
- Bring your laptop!
- You find everything on...

Moodle:

<https://moodle.epfl.ch/course/view.php?id=16434>

Data Exploration

Machine Learning for Behavioral Data
February 27, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction
8	Time Series Prediction

Complete pipeline for one use case:

- Data exploration
- Prediction
- Model evaluation

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace..
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>

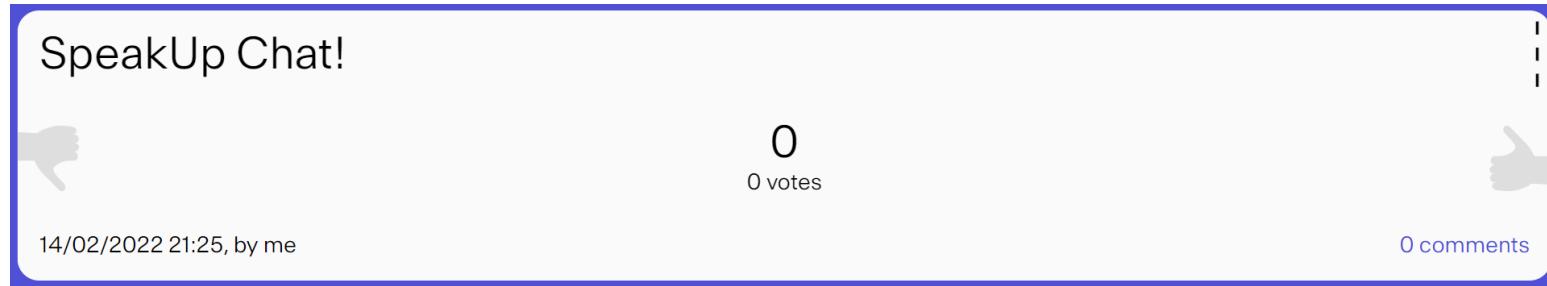


Noto: Student notebook

- Go to <https://noto.epfl.ch/>
- Login with your GASPAR
- Go to Git → Clone
- Clone the course repository: <https://github.com/epfl-ml4ed/mlbd-2023>

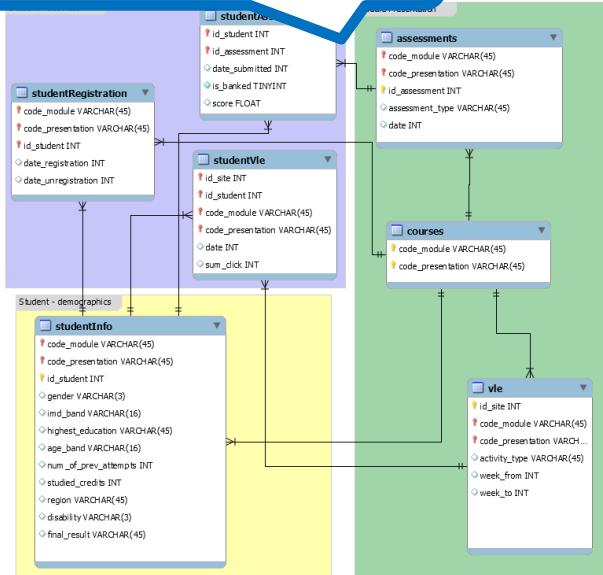
Why is data handling important?

- Why do we not just use the *raw data*?



Different types of input data

Relational databases



Clickstream data

Session1	A8
Session2	A14 A4 A8 A11 A12
Session3	A14 A4 A8 A11 A12
Session4	A14 A4 A9 A8 A9 A8 A11 A12
Session5	A14 A4 A9 A8 A11 A24 A9 A9 A1 A14 A4 A8 A11 A12

logdate	url	ip	city	state	country	category	age	gender
2 2012-03-12	http://www.acme.com/SH55126545/VD55179433	76.166.167.172	oxnard	CA	usa	shoes	29	F
3 2012-03-12	http://www.acme.com/SH55126545/VD55179433	76.166.167.172	oxnard	CA	usa	shoes	29	F
4 2012-03-12	http://www.acme.com/SH55126545/VD55179433	12.132.157.137	opelika	AL	usa	shoes	28	M
5 2012-03-15	http://www.acme.com/SH55126545/VD55179433	24.184.60.95	brooklyn	NY	usa	shoes		
6 2012-03-15	http://www.acme.com/SH55126545/VD55179433	24.184.60.95	brooklyn	NY	usa	shoes		
7 2012-03-15	http://www.acme.com/SH55126545/VD55179433	24.184.60.95	brooklyn	NY	usa	shoes		
8 2012-03-15	http://www.acme.com/SH55126545/VD55179433	24.184.60.95	brooklyn	NY	usa	shoes		
9 2012-03-15	http://www.acme.com/SH55126545/VD55179433	24.184.60.95	brooklyn	NY	usa	shoes		
10 2012-03-12	http://www.acme.com/SH55126545/VD55179433	24.58.5.10	ithaca	NY	usa	shoes		
11 2012-03-12	http://www.acme.com/SH55126545/VD55179433	24.58.5.10	ithaca	NY	usa	shoes		
12 2012-03-12	http://www.acme.com/SH55126545/VD55179433	24.58.5.10	ithaca	NY	usa	shoes		
13 2012-03-12	http://www.acme.com/SH55126545/VD55179433	24.58.5.10	ithaca	NY	usa	shoes		
14 2012-03-05	http://www.acme.com/SH55126545/VD55177927	208.190.165.82	laredo	TX	usa	clothing		
15 2012-03-05	http://www.acme.com/SH55126545/VD55177927	208.190.165.82	laredo	TX	usa	clothing		
16 2012-03-05	http://www.acme.com/SH55126545/VD55177927	208.190.165.82	laredo	TX	usa	clothing		
17 2012-03-05	http://www.acme.com/SH55126545/VD55177927	208.190.165.82	laredo	TX	usa	clothing		
18 2012-03-05	http://www.acme.com/SH55126545/VD55177927	75.138.250.116	spring hill	TN	usa	clothing	25	M
19 2012-03-05	http://www.acme.com/SH55126545/VD55177927	75.138.250.116	spring hill	TN	usa	clothing	25	M
20 2012-03-05	http://www.acme.com/SH55126545/VD55177927	75.138.250.116	spring hill	TN	usa	clothing	25	M
		75.250.116	spring hill	TN	usa	clothing	25	M
		75.250.116	spring hill	TN	usa	clothing	25	M

Events over time

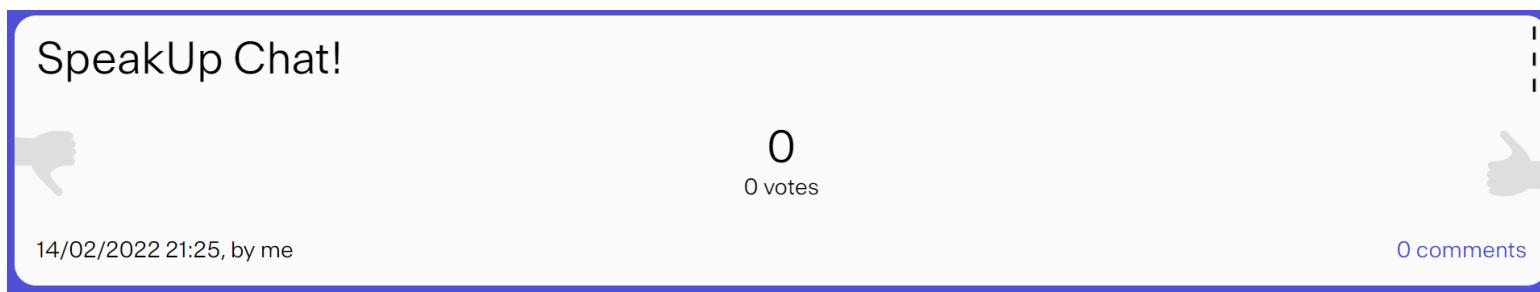
Data Problems

- Incorrect data
- Duplicates
- Inconsistent data
- Missing data
- Outliers



Why is data handling important?

- What is the purpose of *data exploration*?



Today: Data Exploration

- **Univariate Analysis**
- Multivariate Analysis
- Time Series

Today's Use Case: Flipped Classroom Course

- Participants: 157 EPFL students of a course taught in *flipped classroom* mode with a duration of 10 weeks
 - Structure:
 - Preparation: watch videos (and solve simple quizzes) on **new content** at home as a preparation for the lecture
 - Lecture: discuss open questions and solve more complex tasks
 - Lab session: solve paper-and-pen assignments
 - Data: clickstream data (all interactions of the student with the system)
-

Today's Use Case: The Data

		Video_Info		Video_Events				
TimeStamp	DataPackageID	UniqueRowID	TableName	VideoID	EventType	SessionUserID		
1436539064	hwts-002	0000000773b50de2958e6128ca6a01dc	Video_Events	75	Video.Download	9e6622aa3440f144edb91a7d63973		
1348761147	progfun-2012-001	00000013631cd1107b9781b40c37ac07	Video_Events	37	Video.Play	a7e07c5f41369e0acdf08ec72794b		
1362266322	dsp-001	0000002363c3bd0f73b783e3adc44fb3	Video_Events	29	Video.Pause	bf85620e711cc570f95763d9768c0		
1430601717	reactive-002	00000059c6fb3e38eb5639e1b9e6c863	Video_Events	133	Video.Seek	ec35ab9103eb35ffcaf74f12c7e97		
1372391638	progfun-002	00000078c0f0685cc50a25a8d5734a88	Video_Events	33	Video.Play	ef64fb7b096008f7eaf8441684afdf9		
1348627928	progfun-2012-001	000000d6a01b089ecee6aea3ddb4589c	Video_Events	33	Video.Seek	f12fbe6298a9e46122ed11cfabc43t		
1366535543	progfun-002	0000013af9c71dde9e67332e9f2220f	Video_Events	39	Video.Load	8d7c72c0dfe78d0dbeb187c6c4643		
1361863559	dsp-001	00000146053bbf1daf5e74539b695ae6	Video_Events	43	Video.Play	c0b7417192e8b38e8f6cb641fc7bd		
1350842274	progfun-2012-001	0000016e472deac18413b2a7ccdc2e07	Video_Events	97	Video.Seek	0c8efe11945ef0f1d0017707ba930		
1400493317	progfun-004	0000017c871f54fd701333bd0acf7ba	Video_Events	77	Video.Play	2487d6899365bd5f704979f91995		
1426880606	villesafricaines-003	0000017ea64ccce0f405090cf7220b51	Video_Events	47	Video.Load	b27704ef3090a0f666907807c1d85		
1417881517	intropooprojava-001	0000019fa8f938d69cc019e7805edcba	Video_Events	67	Video.Pause	8ae201009a69aa6ee8c0ae7909279		
1395399921	java-fr-2013-001	000001cb3ef0ccf281d3b9f1c00e7d60	Video_Events	13	Video.Stalled	817fc9f1ede5e69d36641c8b2d937		
1400786471	microcontroleurs-003	000001d606e9a4bea4544c1827275b89	Video_Events	19	Video.Pause	6c06a76c20df00c17f1d83e7c1832		

Characteristics of a Variable/Feature

ID	Grade	Gender	Category	# Sessions	Time in videos	Time in problems	# clicks on weekdays	# clicks on weekends	Content alignment	Mean pause duration	Mean playback speed	# problem submissions	# correct submissions
1	4.5	M	Suisse. Autres	57	9227	1698	179	4	0.75	50	1.1	9	5.9
2	5.25	M	Suisse. Autres	41	10801	2340	129	95	0.35	231	0.8	6.1	3
3	4.5	F	Suisse. PAM	33	8185	2737	46	14	0.37	92	0.5	4.6	3.2
4	4.75	F	France	47	7040	3787		58	0.03	62	0.85	0.3	0.1

- 
- Center of the data?
 - Spread of the data?
 - Shape/distribution of the data?

Descriptive Statistics

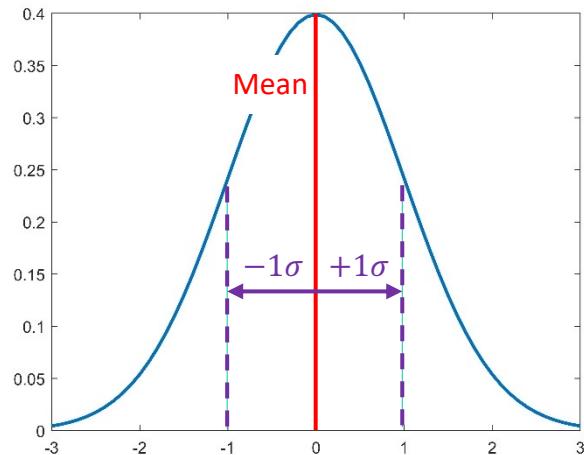
	Mean	Median	Mode	Variance	Std	Minimum	25%	75%	Maximum
grade	4.05	4.25	5.0	1.49e+00	1.22	1.00	3.25	5.00	6.00
sessions	33.89	34.00	36.0	2.38e+02	15.42	6.00	22.00	43.00	97.00
time_in_problem	28022.04	24209.50	0.0	4.83e+08	21980.95	0.00	10029.00	41756.75	111238.00
time_in_video	82851.62	81735.50	26699.0	2.20e+09	46942.02	0.00	48823.25	111431.25	274917.00
lecture_delay	820.27	0.00	0.0	1.85e+09	43010.20	-159250.48	-22921.90	24249.25	144964.21
content_anticipation	0.11	0.09	0.0	1.02e-02	0.10	0.00	0.01	0.20	0.31
mean_playback_speed	0.94	0.92	0.9	9.37e-02	0.31	0.00	0.80	1.11	1.76
relative_video_pause	0.22	0.23	0.0	1.05e-02	0.10	0.00	0.14	0.30	0.43
submissions	46.05	35.50	0.0	1.77e+03	42.12	0.00	9.75	77.00	171.00
submissions_correct	25.01	18.00	0.0	5.24e+02	22.90	0.00	4.75	41.00	89.00
clicks_weekend	679.80	465.00	0.0	4.93e+05	702.04	0.00	160.50	1012.75	4546.00
clicks_weekday	1130.64	930.50	108.0	8.13e+05	901.44	0.00	495.00	1534.00	6223.00

Center of the data

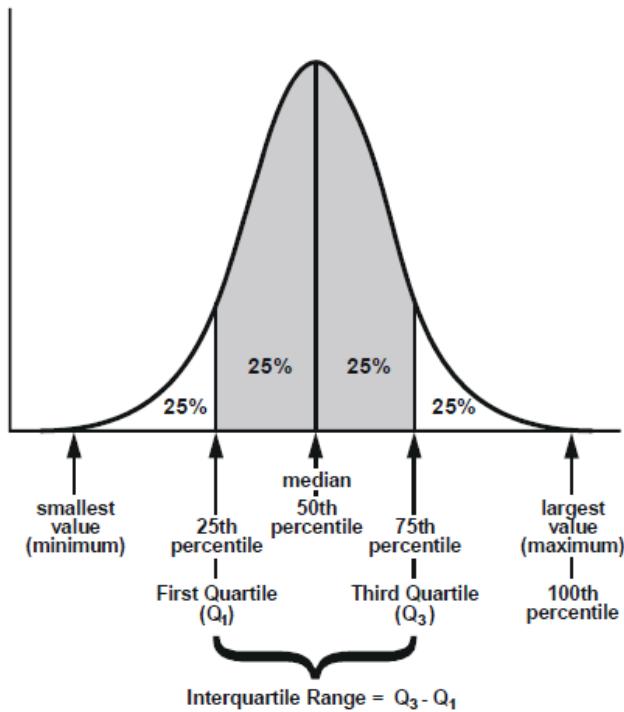
Spread of the data

Example: Normal Distribution

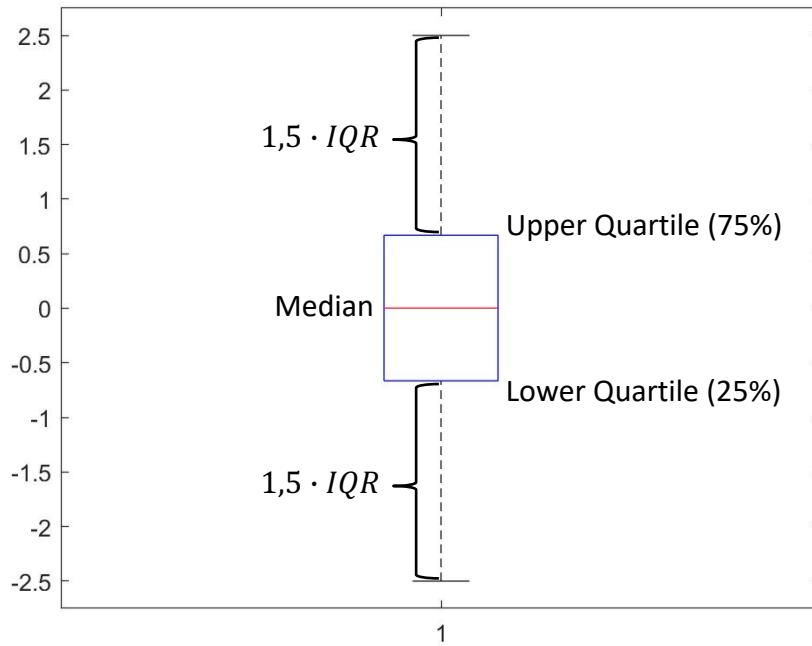
- Sample mean: $\mu_{\bar{x}} = \frac{1}{n} \sum_{i=1}^n x_i$
- Sample variance: $\sigma_{\bar{x}}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_{\bar{x}})^2$
- Mode: most frequent value in data set
- Median: separates the lower and upper half of the data (1, 2, 2, 3, 4, 7, 9)



Example: Normal Distribution



Boxplot



Descriptive Statistics

	Mean	Median	Mode	Variance	Std	Minimum	25%	75%	Maximum
grade	4.05	4.25	5.0	1.49e+00	1.22	1.00	3.25	5.00	6.00
sessions	33.89	34.00	36.0	2.38e+02	15.42	6.00	22.00	43.00	97.00
time_in_problem	28022.04	24209.50	0.0	4.83e+08	21980.95	0.00	10029.00	41756.75	111238.00
time_in_video	82851.62	81735.50	26699.0	2.20e+09	46942.02	0.00	48823.25	111431.25	274917.00
lecture_delay	820.27	0.00	0.0	1.85e+09	43010.20	-159250.48	-22921.90	24249.25	144964.21
content_anticipation	0.11	0.09	0.0	1.02e-02	0.10	0.00	0.01	0.20	0.31
mean_playback_speed	0.94	0.92	0.9	9.37e-02	0.31	0.00	0.80	1.11	1.76
relative_video_pause	0.22	0.23	0.0	1.05e-02	0.10	0.00	0.14	0.30	0.43
submissions	46.05	35.50	0.0	1.77e+03	42.12	0.00	9.75	77.00	171.00
submissions_correct	25.01	18.00	0.0	5.24e+02	22.90	0.00	4.75	41.00	89.00
clicks_weekend	679.80	465.00	0.0	4.93e+05	702.04	0.00	160.50	1012.75	4546.00
clicks_weekday	1130.64	930.50	108.0	8.13e+05	901.44	0.00	495.00	1534.00	6223.00

Variable Types

- Categorical
- Ordinal
- Numerical

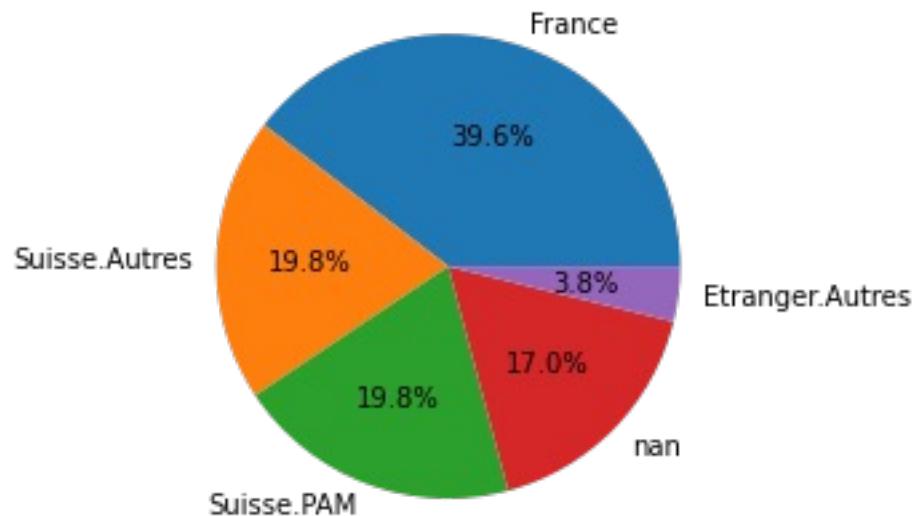
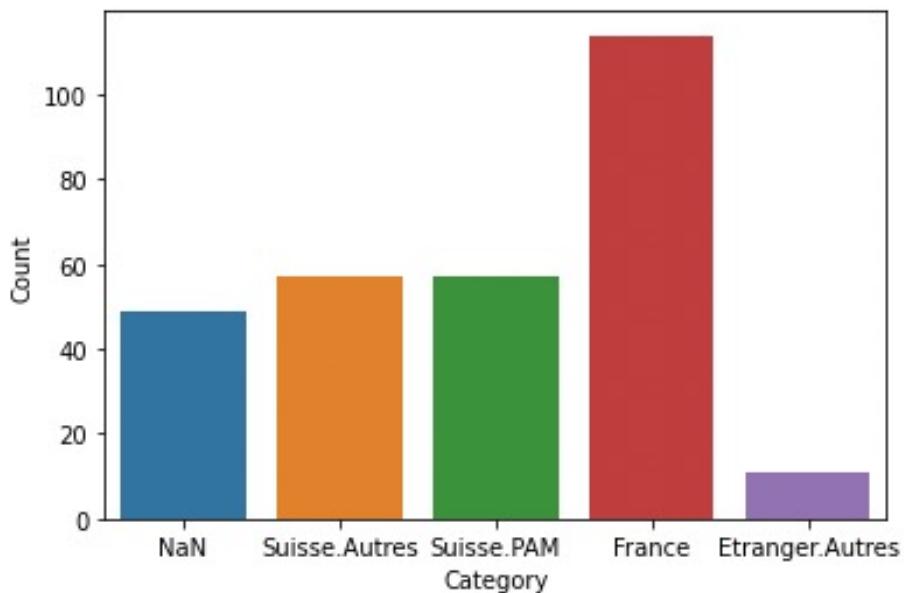


Categorical Variables

Category	Count	Count %
France	114	0.40
Suisse.Autres	57	0.20
Suisse.PAM	57	0.20
NaN	49	0.17
Etranger.Autres	11	0.04

Gender	Count	Count %
M	156	0.54
F	83	0.29
NaN	49	0.17

Number of students per category

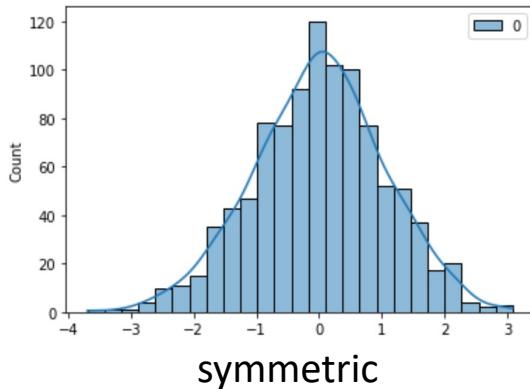


Characteristics of a Variable/Feature

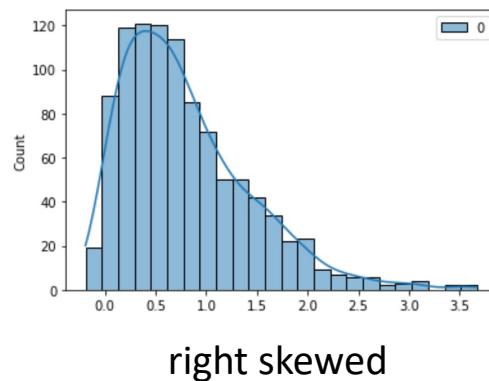
ID	Grade	Gender	Category	# Sessions	Time in videos	Time in problems	# clicks on weekdays	# clicks on weekends	Content alignment	Mean pause duration	Mean playback speed	# problem submissions	# correct submissions
1	4.5	M	Suisse. Autres	57	9227	1698	179	4	0.75	50	1.1	9	5.9
2	5.25	M	Suisse. Autres	41	10801	2340	129	95	0.35	231	0.8	6.1	3
3	4.5	F	Suisse. PAM	33	8185	2737	46	14	0.37	92	0.5	4.6	3.2
4	4.75	F	France	47	7040	3787		58	0.03	62	0.85	0.3	0.1

- 
- Center of the data?
 - Spread of the data?
 - Shape/distribution of the data?

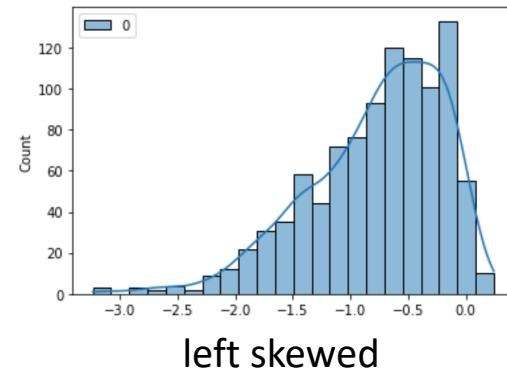
Does my data follow a normal distribution?



symmetric



right skewed



left skewed

Normal test $p = 0.39$

Normal test $p = 8.7\text{e-}43$

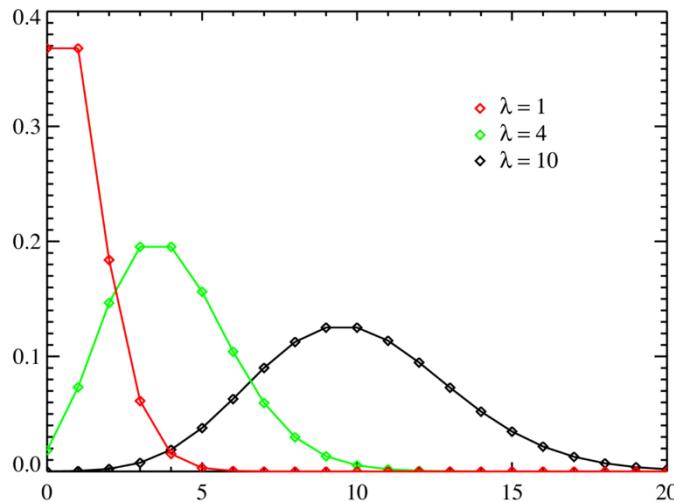
Normal test $p = 6.0\text{e-}26$

Important Distributions

- **Normal distribution** : (*continuous*) see previous slides
 - **Poisson distribution**: (*discrete*) expresses the probability of a given number of events occurring in a fixed interval of time or space
 - **Exponential distribution** (*continuous*) distribution of times between events in a Poisson process
 - **Binomial distribution**: (*discrete*) models the number of successes in a sequence of independent experiments
 - **Bernoulli distribution**: (*discrete*) special case of binomial distribution ($n=1$)
-

Important Distributions | Poisson

Models the number of events occurring within a given time interval.



Properties:

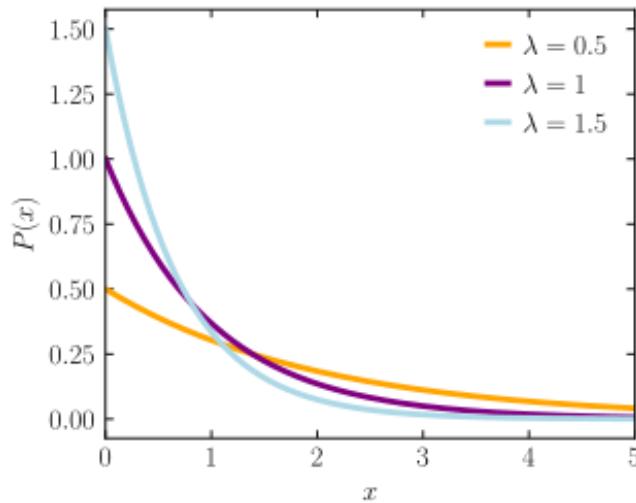
- Discrete (not continuous)
- Greater or equal to zero.

Examples:

- Number of calls a call center receives per minute
- Number of students that join the zoom meeting per minute during the first 15 minutes of the class

Important Distributions | Exponential

Probability distribution of time between events of a **Poisson** process.



Properties:

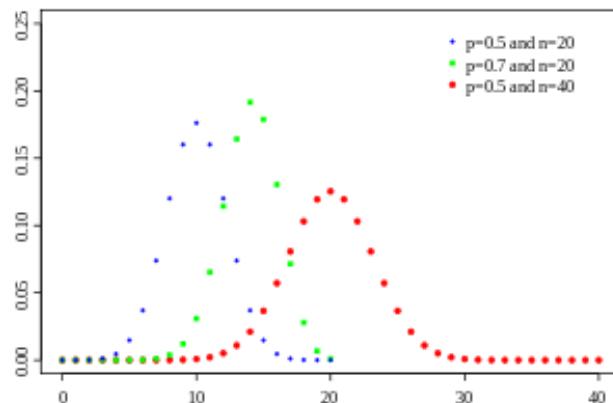
- Continuous
- Greater or equal to zero.

Examples:

- The time before the next telephone call in a call center.
- The time before the next student joins the zoom call.

Important Distributions | Binomial

Models the number of successes in a sequence of independent experiments.



Properties:

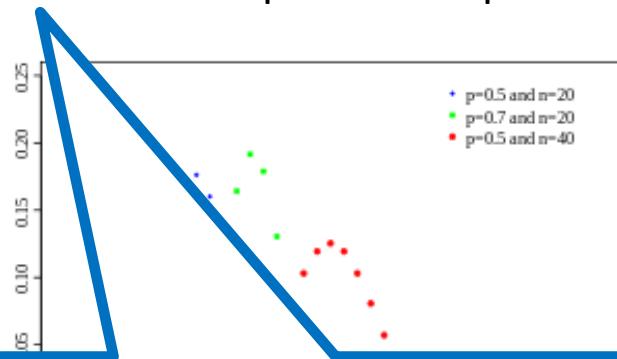
- Discrete (not continuous)
- Greater or equal to zero.

Examples:

- Number of passed tests in a course with 20 tests.
- Number of customers that redeemed a coupon.

Important Distributions | Binomial

Models the number of successes in a sequence of independent experiments.



Bernoulli is a special case of the Binomial distribution with one experiment: $n = 1$

Properties:

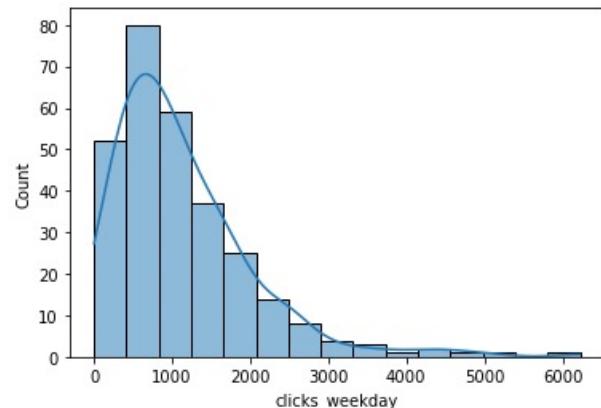
- Discrete (not continuous)
- Greater or equal to zero.

Examples:

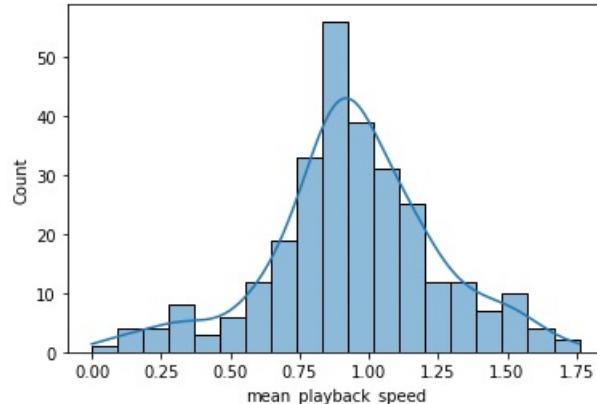
- Number of passed tests in a course with 20 tests.
- Number of customers that redeemed a coupon.

Visual Inspection

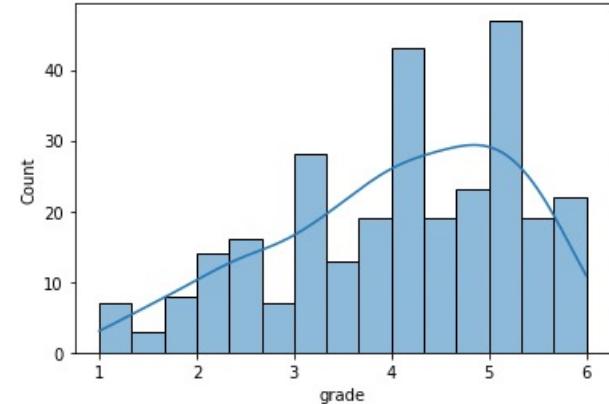
$p = 6.20579e-29$
The null hypothesis can be rejected



$p = 0.0216998$
The null hypothesis cannot be rejected



$p = 5.78191e-05$
The null hypothesis can be rejected



Data Exploration

- Univariate Analysis
- **Multivariate Analysis**
- Time Series



Multivariate Analysis

How can we explore the relationship between two variables?

SpeakUp Chat!

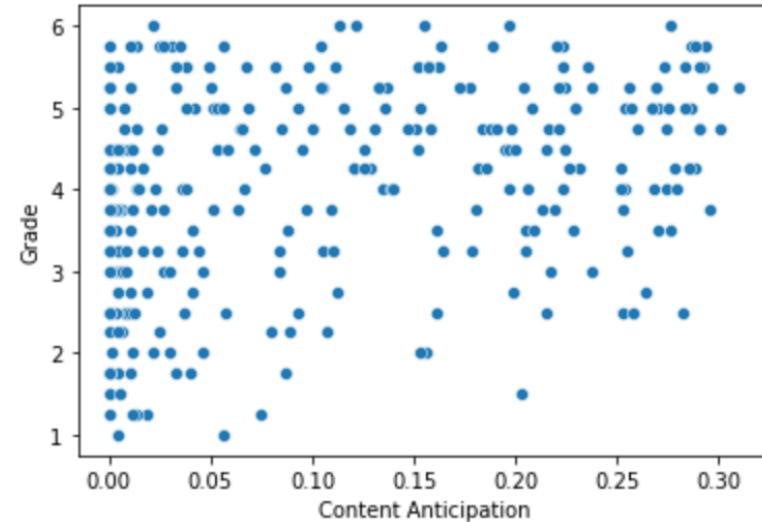
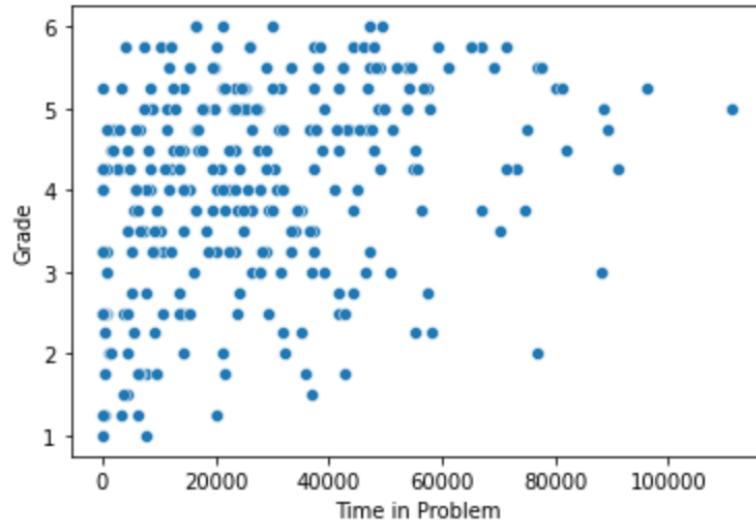
0
0 votes

14/02/2022 21:25, by me

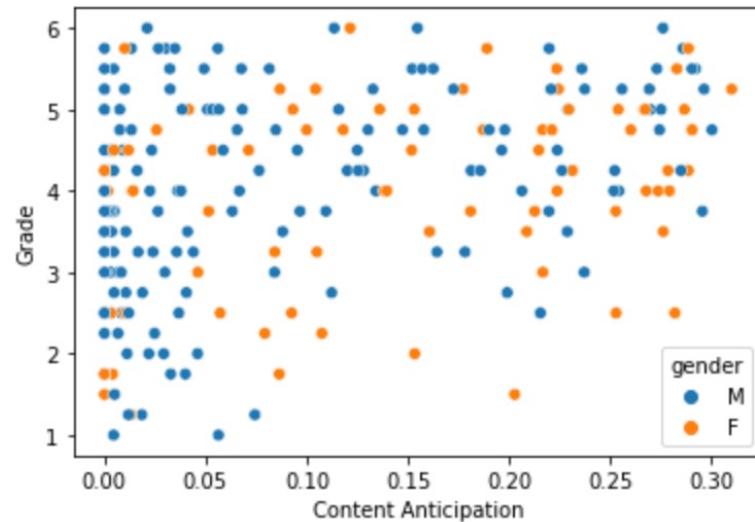
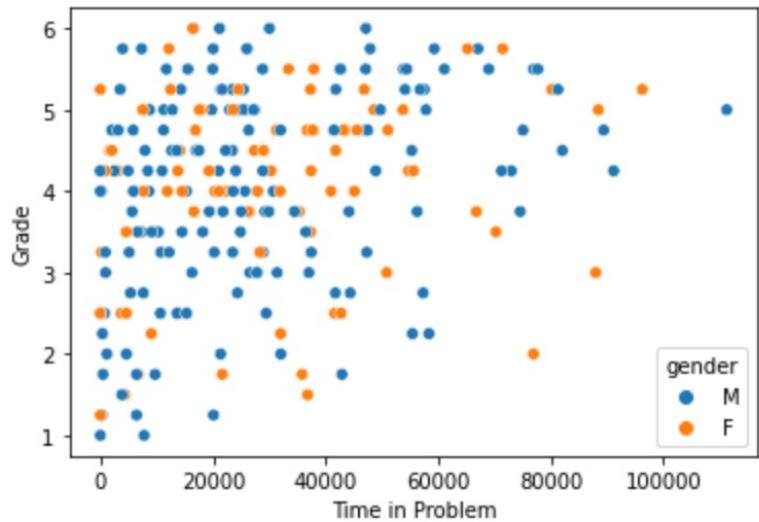
0 comments



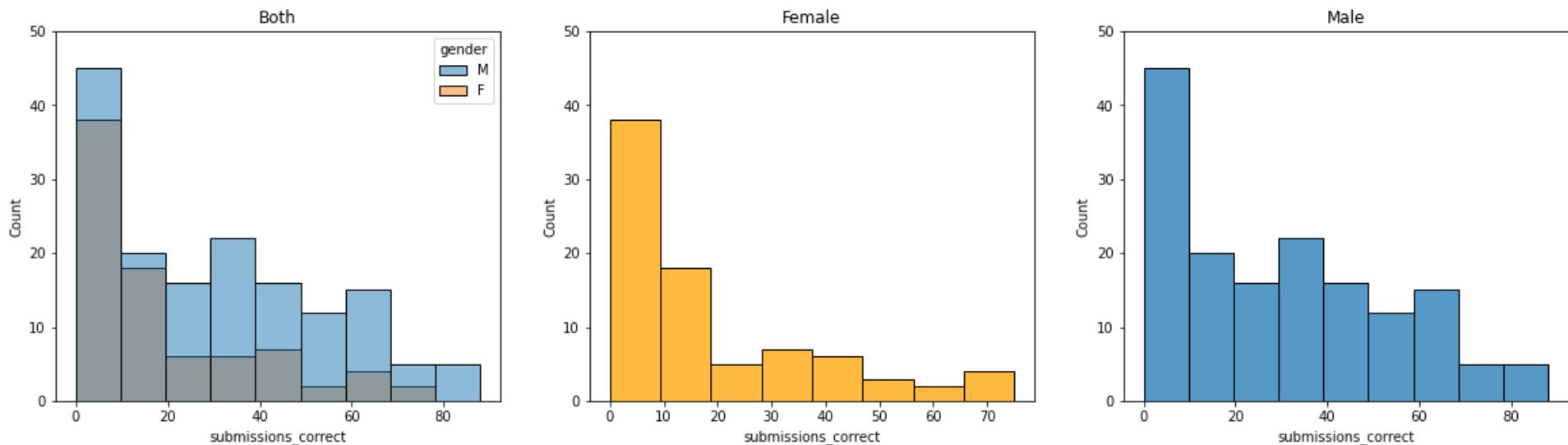
Relation between numerical variables



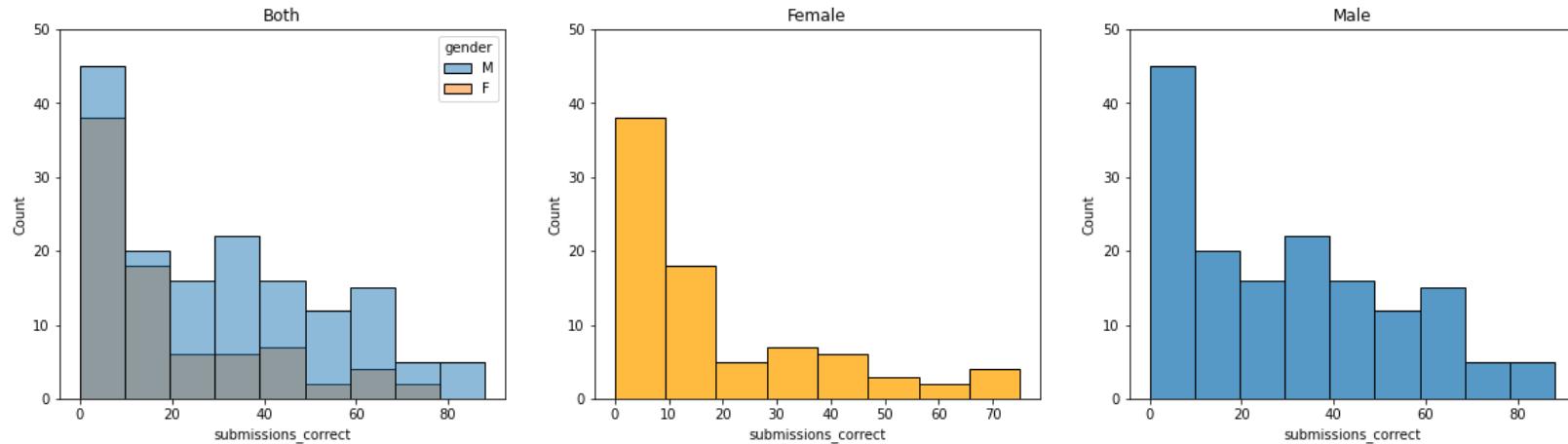
Relation between numerical & categorical variables



Submissions Correct by Gender



Who is more likely to have correct submissions?



- a) Students identifying as male are more likely to have a correct submission.
- b) Students identifying as female are more likely to have a correct submission.
- c) I cannot answer based on the visualization.



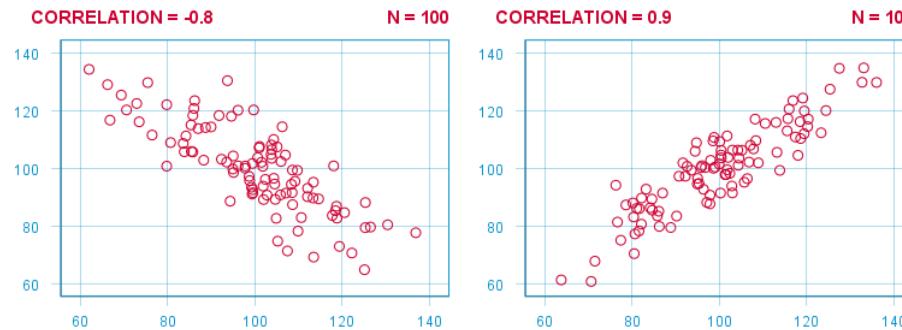
<https://go.epfl.ch/speakup-mlbd>

Pearson's Correlation

Linear correlation between two sets of data.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Where $\text{cov}(X, Y)$ is the covariance
 σ_X is the standard deviation on X
 σ_Y is the standard deviation on Y



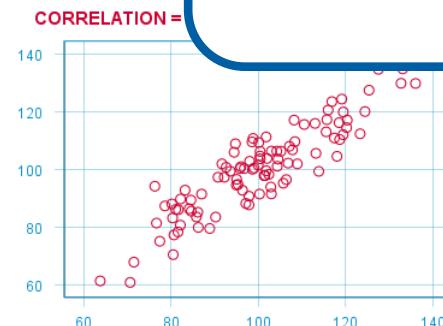
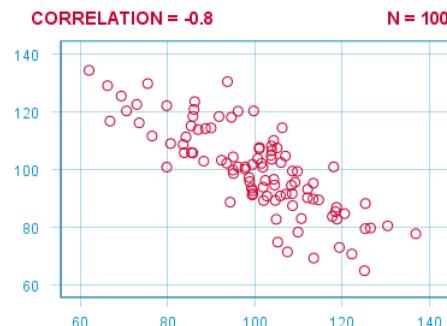
Pearson's Correlation

Linear correlation between two sets of data.

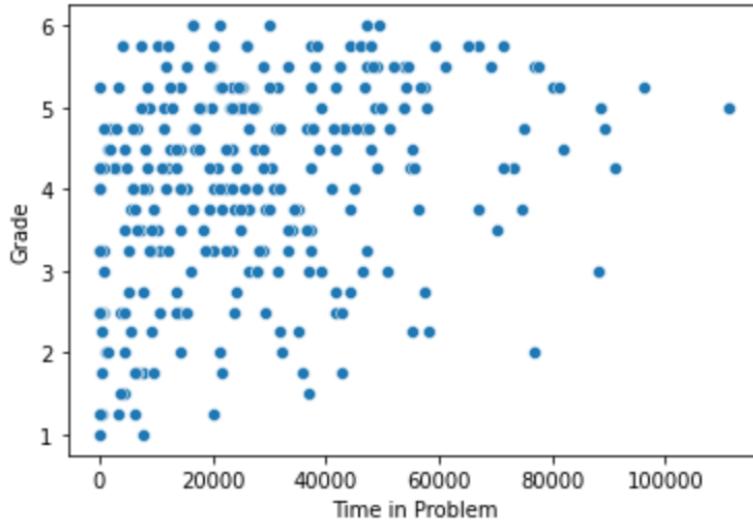
$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Where cov is the covariance,
 σ_X is the standard deviation of X ,
 σ_Y is the standard deviation of Y .

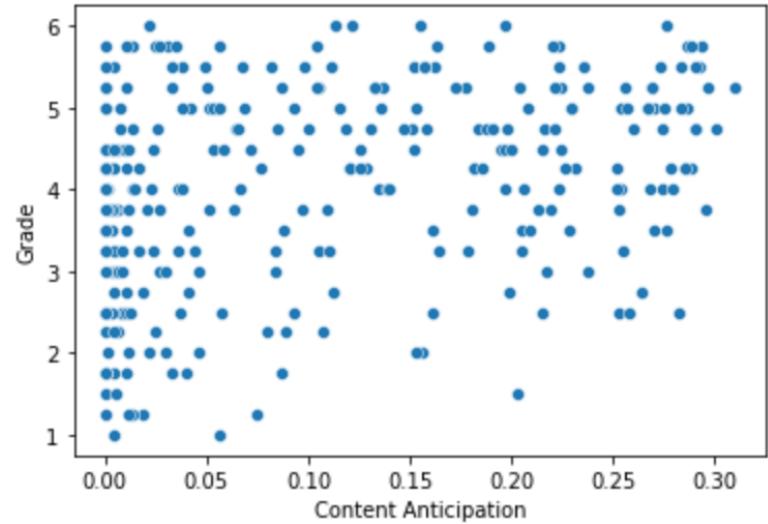
X and Y need to be numerical or at least ordinal variables



Correlation between variables



$$\rho = 0.31 \ (p = 6.8e - 8)$$



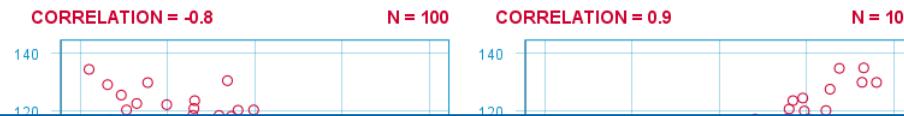
$$\rho = 0.32 \ (p = 1.5e - 08)$$

Pearson's Correlation

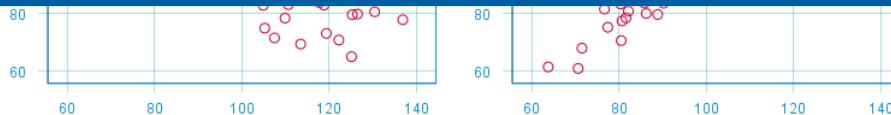
Linear correlation between two sets of data.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Where $\text{cov}(X, Y)$ is the covariance
 σ_X is the standard deviation on X
 σ_Y is the standard deviation on Y



No correlation = variables are independent?

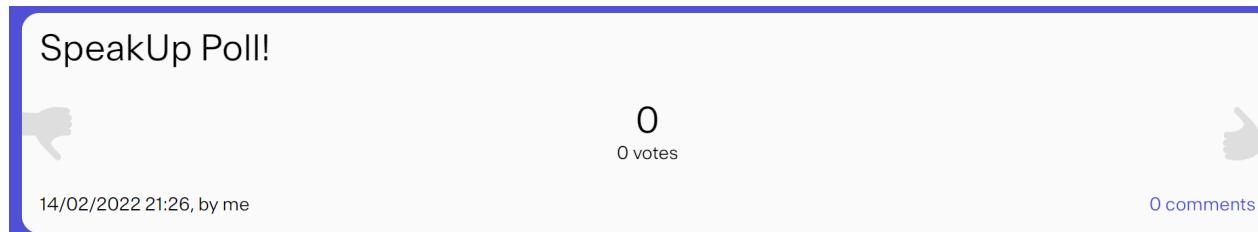


Pearson's Correlation

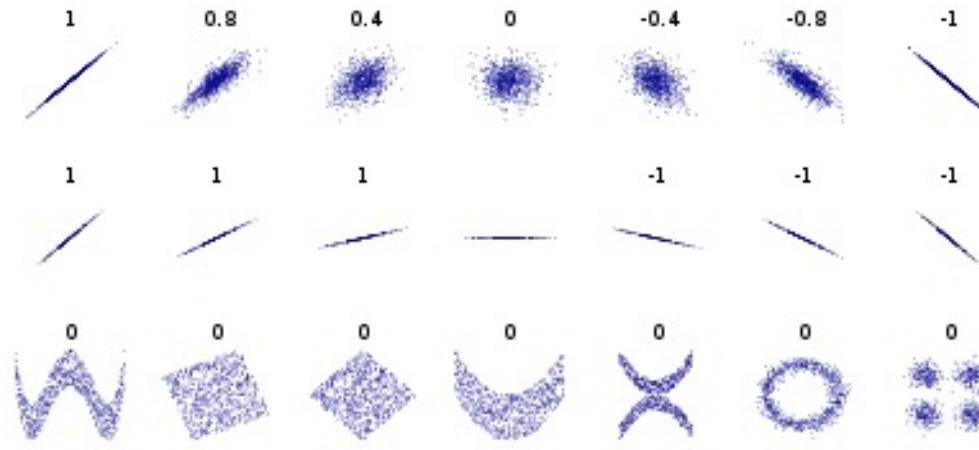
Linear correlation between two sets of data.

No correlation = variables are independent?

- a) Yes
- b) No



Pearson's Correlation



X, Y independent $\rightarrow \rho_{X,Y} = 0$
 $\rho_{X,Y} = 0 \not\Rightarrow X, Y$ independent

Mutual Information

- Dependence between two random variables: “Amount of information” obtained about one random variable through observing the other random variable

$$I(X;Y) = D_{KL}(P_{(X,Y)} \parallel P_X \otimes P_Y)$$

where X and Y are random variables, $P_{(X,Y)}$ is their joint distribution, P_X and P_Y are the marginal distributions, and D_{KL} is the Kullback-Leibler divergence.

Mutual Information

- Dependence between two random variables: “Amount of information” obtained about one random variable through observing the other random variable

$$I(X; Y) = D_{KL}(P_{(X,Y)} || P_X \otimes P_Y)$$

where X and Y are random variables, $P_{(X,Y)}$ is their joint distribution, P_X and P_Y are the marginal distributions, and D_{KL} is the Kullback-Leibler divergence.

- For discrete distributions

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot \log\left(\frac{p(x, y)}{p(x) \cdot p(y)}\right)$$

Mutual Information - Motivation

- For discrete distributions

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot \log\left(\frac{p(x, y)}{p(x) \cdot p(y)}\right)$$

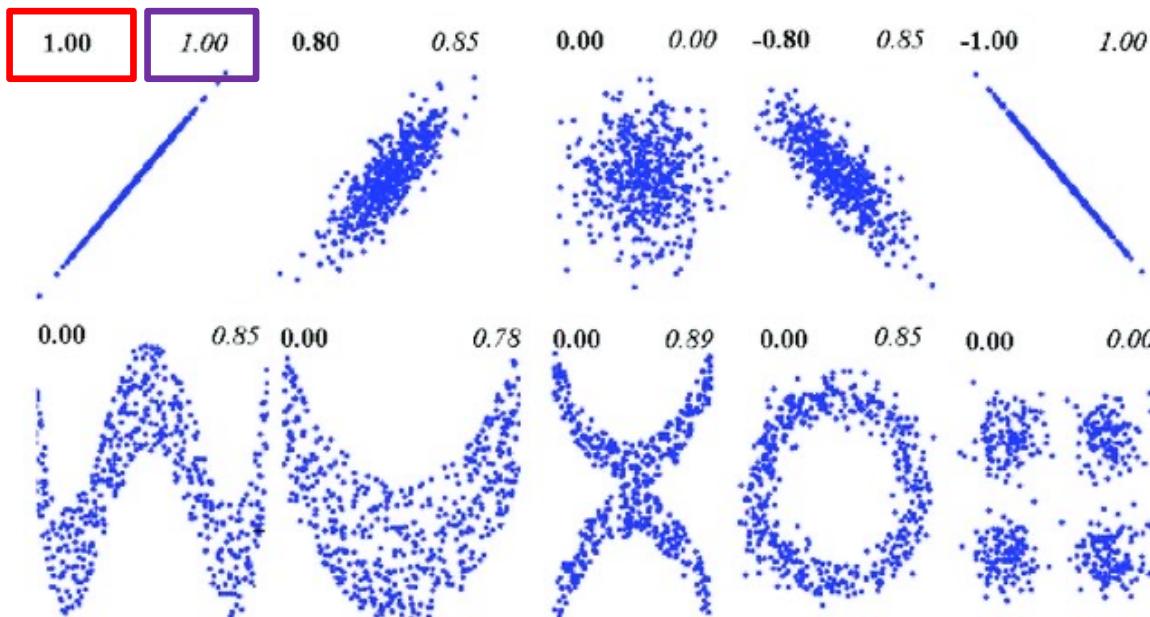
- If X and Y are *independent*, then $p(x, y) = p(x) \cdot p(y)$ and therefore:

$$\log\left(\frac{p(x, y)}{p(x) \cdot p(y)}\right) = \log(1) = 0$$

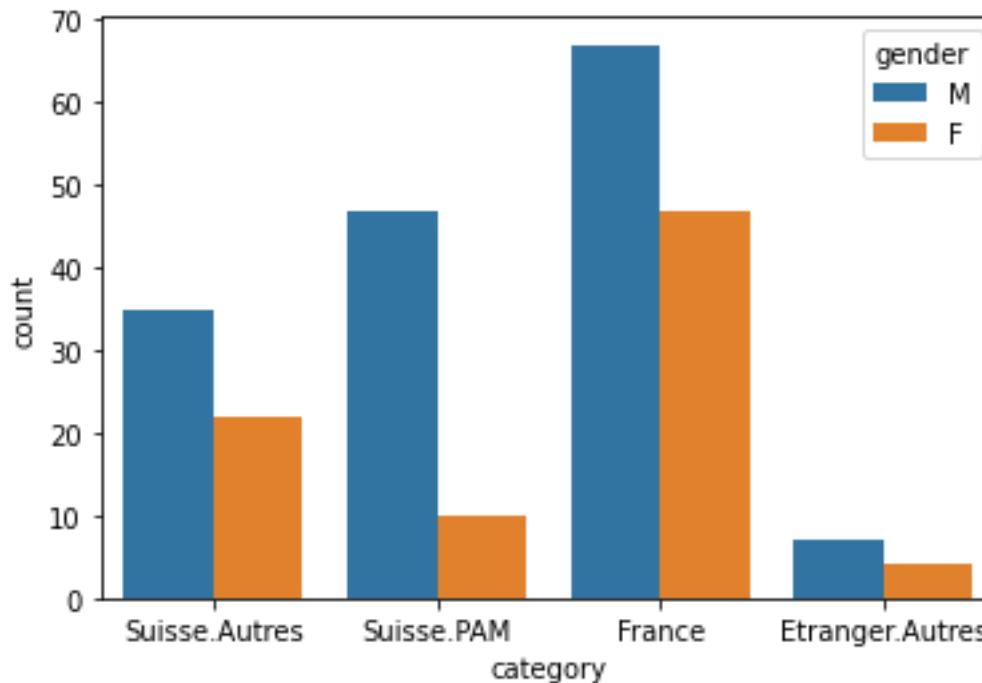
Pearson Correlation vs Mutual Information

Pearson's Correlation
Coefficient

Mutual Information



Mutual Information – Discrete



Mutual Information - Discrete

$P(X, Y)$		Y: Category			
X: Gender		France	Suisse.PAM	Suisse. Autres	Etranger.Autres
		Male	0.28	0.20	0.15
	Female	0.20	0.04	0.09	0.02

Mutual Information - Discrete

$P(X, Y)$

X: Gender

Y: Category

	France	Suisse.PAM	Suisse. Autres	Etranger.Autres
Male	0.28	0.20	0.15	0.02
Female	0.20	0.04	0.09	0.02

$P(Y)$

France	Suisse.PAM	Suisse. Autres	Etranger.Autres
0.48	0.24	0.24	0.04

$P(X)$

Female	Male
0.35	0.65

Mutual Information - Discrete

$P(X, Y)$

X: Gender

Y: Category

	France	Suisse.PAM	Suisse. Autres	Etranger.Autres
Male	0.28	0.20	0.15	0.02
Female	0.20	0.04	0.09	0.02

$P(Y)$

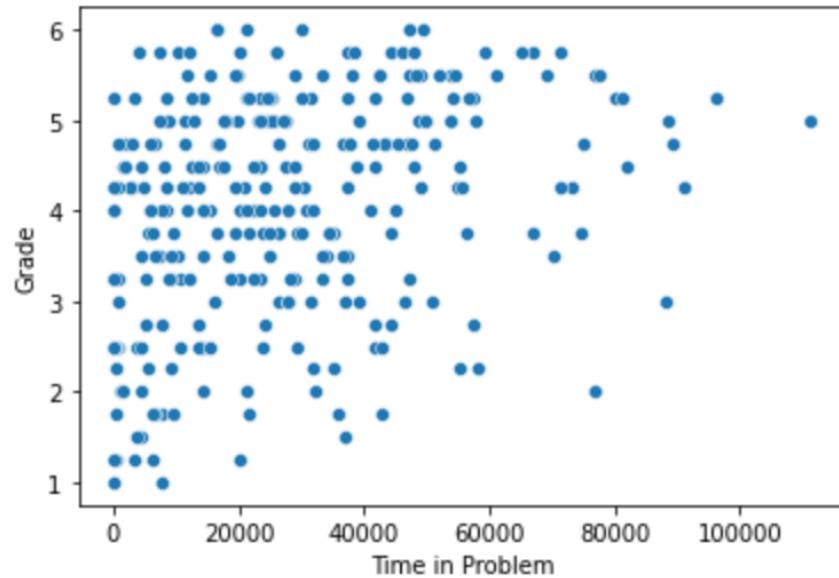
France	Suisse.PAM	Suisse. Autres	Etranger.Autres
0.48	0.24	0.24	0.04

$P(X)$

Female	Male
0.35	0.65

$$I(X; Y) = 0.02$$

Mutual Information - Continuos



$$I(X; Y) = 0.12$$

$$\rho = 0.31 \ (p = 6.8e - 8)$$

Data Exploration

- Univariate Analysis
- Multivariate Analysis
- **Time Series**



Time Series Data

Records, which are measured sequentially over time:

- **Business:** sales figures, production numbers, customer frequencies, ...
- **Economics:** stock prices, exchange rates, interest rates, ...
- **Official Statistics:** census data, personal expenditures, road casualties, ...
- **Natural Sciences:** population sizes, sunspot activity, chemical process data, ...
- **Environmetrics:** precipitation, temperature or pollution recordings, ...

Time Series – Behavioral Data

Records of **user behavior**, which are measured sequentially over time:

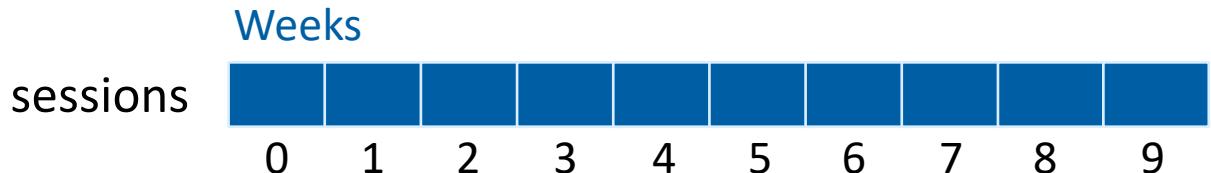
- we usually deal with multiple time series (i.e. one time series per user u)
- a record $r_{u,t}$ of a user u at time t can consists of multiple variables

We might be interested in representing, analyzing, and predicting behavior of single users or of group of users:

- Visualization and exploration of time series data (this lecture)
 - Modeling time series data (later...)
-

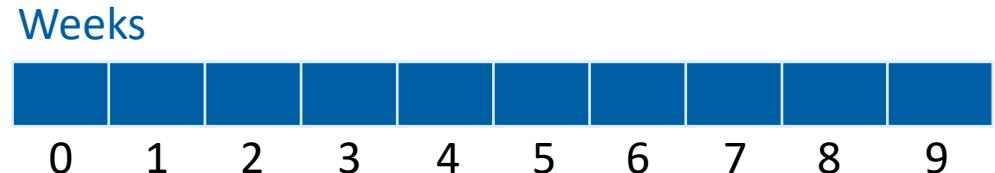
Time Series – Our flipped classroom case

Student n



•
•
•

submissions_correct



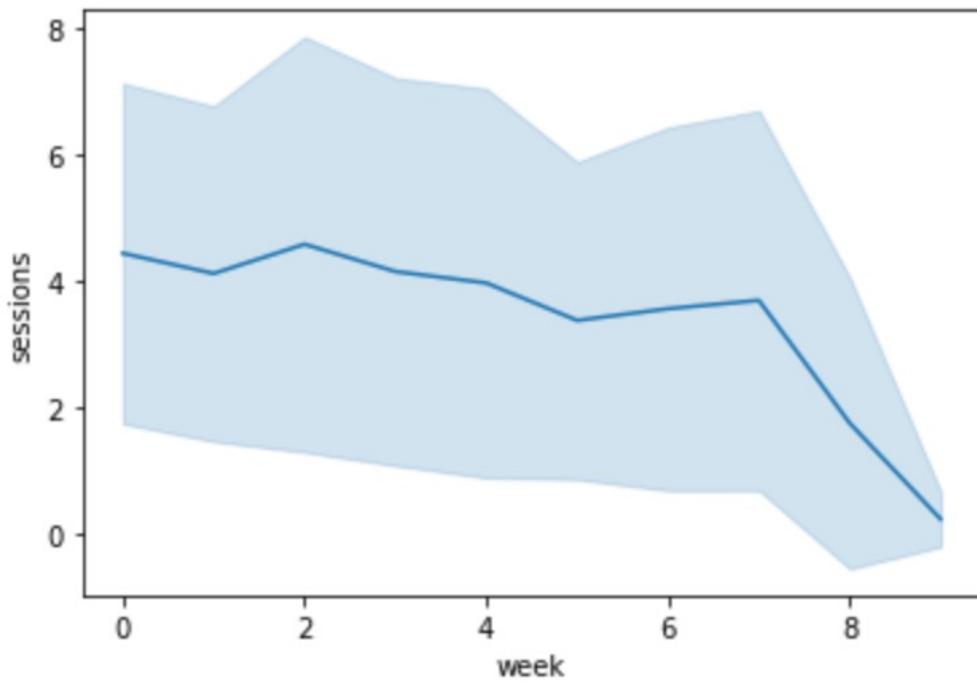
Hypothesis 1

The number of sessions will decrease over the course of the semester.



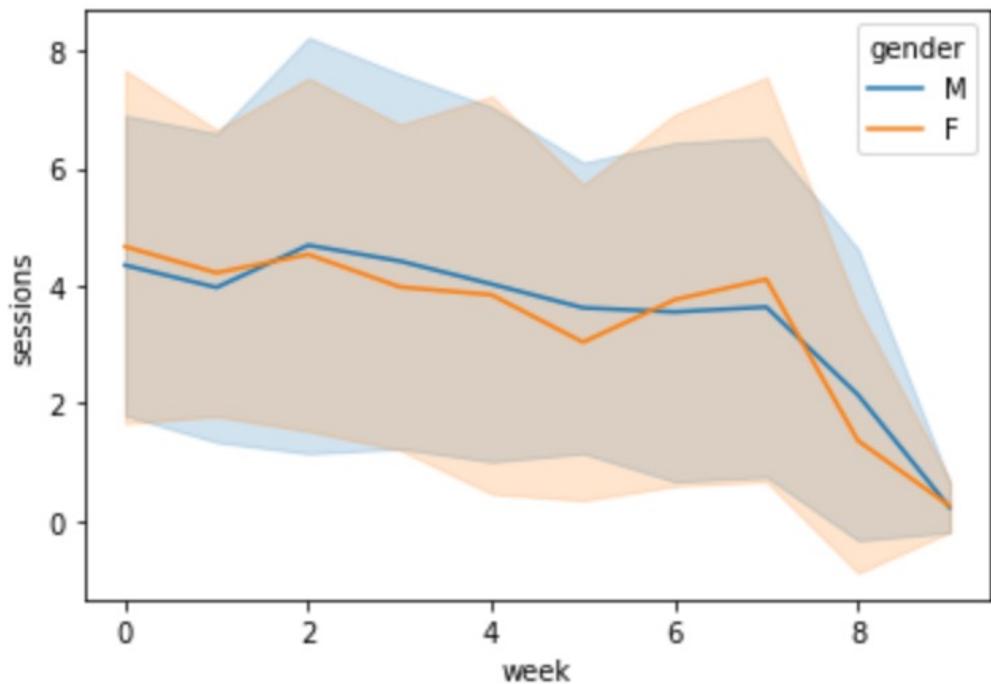
Hypothesis 1

The number of sessions will decrease over the course of the semester.



Hypothesis 2

There is no difference between males and females in terms of the number of sessions.



Your turn!

- Come up with a hypothesis on your own
- Produce a visualization
- Describe: what do you observe? Can your hypothesis be confirmed?



Your turn!

- Come up with a hypothesis on your own
- Produce a visualization
- Describe: what do you observe? Can your hypothesis be confirmed?

Do you want feedback or have questions?

(Optional) Upload your Jupyter Notebook here:

<https://go.epfl.ch/notebooks-mlbd>

Summary

- Compute descriptive statistics
 - Visualize, visualize, visualize,...
 - Different types of visualizations or representations help to identify different types of problems
 - Different types of visualizations help to identify different patterns/properties in the data
 - Try to gain as much knowledge as possible about the domain and the data collection
-

Regression

Machine Learning for Behavioral Data
March 6, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction
8	Spring Break

Complete pipeline for one use case:

- Data exploration
- Prediction
- Model evaluation

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



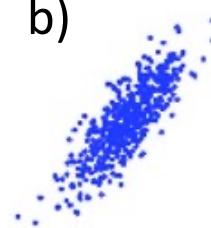
Short quiz about the past...

- Which of the four graphs have the following properties:
High Pearson's Correlation, High Mutual Information

a)



b)

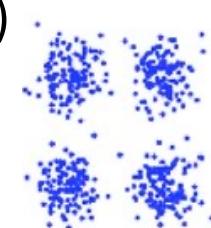


c) None

d)



e)



SpeakUp Poll!



14/02/2022 21:26, by me

0

votes



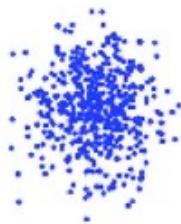
0 comments

Short quiz about the past...

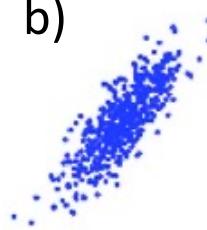
- Which of the four graphs have the following properties:

High Pearson's Correlation, Low Mutual Information

a)



b)

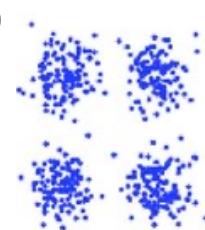


c) None

d)



e)



SpeakUp Poll!



14/02/2022 21:26, by me

0

votes



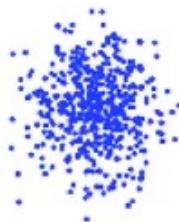
0 comments

Short quiz about the past...

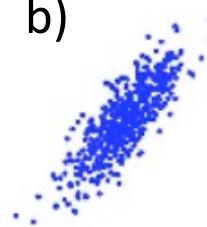
- Which of the four graphs have the following properties:

Low Pearson's Correlation, Low Mutual Information

a)



b)

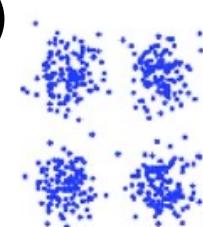


c) None

d)



e)



SpeakUp Poll!



14/02/2022 21:26, by me

0

votes



0 comments

Today's Use Case: Flipped Classroom Course

- Participants: 288 EPFL students of a course taught in *flipped classroom* mode with a duration of 10 weeks
 - Structure:
 - Preparation: watch videos (and solve simple quizzes) on **new content** at home as a preparation for the lecture
 - Lecture: discuss open questions and solve more complex tasks
 - Lab session: solve paper-an-pen assignments
 - Data: clickstream data (all interactions of the student with the system)
-

Agenda

- **Linear Regression**
- Generalized Linear Models
- Mixed-Effect Models
- Performance Metrics
- Regression for Time-Series



Idea

- In regression, a single aspect of the data (output variable) is modeled by some combination of other aspects of the data (input variables)



More formal

- In regression, a single aspect of the data (output variable) is modeled by some combination of other aspects of the data (input variables)
 - Given: N data points (y_n, \mathbf{x}_n) , where y_n is the n 'th output variable and \mathbf{x}_n is a D-dimensional vector of input variables
 - Goal: $y_n \approx f(\mathbf{x}_n)$
-

Usage

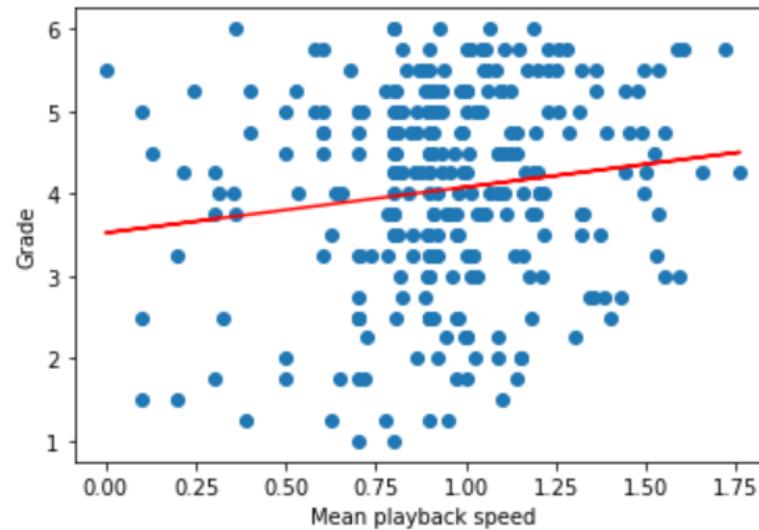
- *Prediction*: predict the output for a new (unseen) input vector x
 - *Interpretation*: analyze the relationships between the variables (what effect the input variables have on the output variable)
-

Example | Mean playback speed

x-axis: Mean playback speed of videos

y-axis: Course grade

Each point is one student



Students who watch the videos faster tend to have better grades.

Linear Regression

The output variable y_n with $n = 1, \dots, N$ is modeled by a **linear** combination of the input variables $x_{n,d}$ with $d = 1, \dots, D$.

$$y_n = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D} + \epsilon_n$$

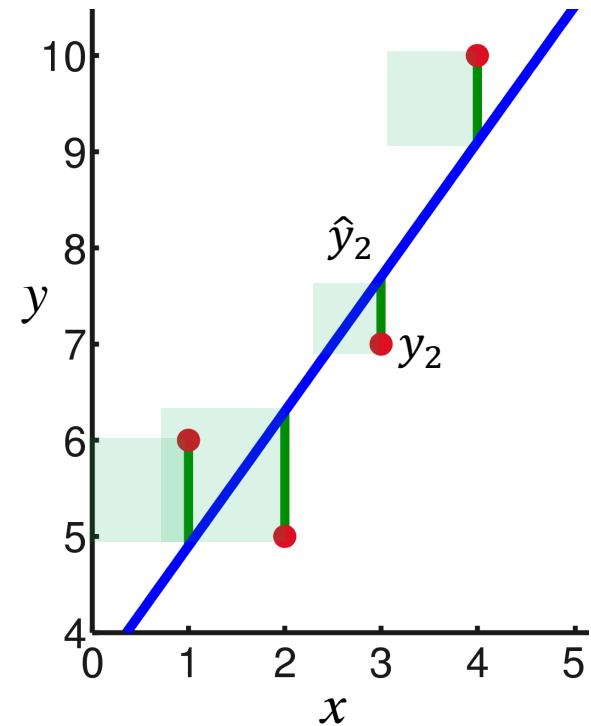
where ϵ_n are error terms that should be as small as possible and $\epsilon_n \sim N(0, \sigma^2)$.

Goal: find optimal parameters

Find parameters $\hat{\beta}$ that minimize

$$\sum_{n=1}^N (y_n - \tilde{x}_n^T \cdot \hat{\beta})^2$$

with $\tilde{x}_n = \begin{bmatrix} 1 \\ x_{n,1} \\ \dots \\ x_{n,D} \end{bmatrix}$ and $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \dots \\ \hat{\beta}_D \end{bmatrix}$



✓

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2 \quad SS_{\text{res}} = \sum_i (y_i - f(x_i))^2$$

Fitting the parameters

X

$$\text{grade} = \beta_0 + \beta_1 \cdot \text{time_in_problem} + \beta_2 \cdot \text{percentage_correct}$$

Formula: grade ~ ch_time_in_prob_sum + wa_num_subs_perc_correct

Family: gaussian

Estimator: OLS

Std-errors: non-robust

CIs: standard 95%

Number of observations: 288

R^2: 0.110

Inference: parametric

Log-likelihood: -449.516

AIC: 905.031

R^2_adj: 0.104

BIC: 916.020

Fixed effects:

$p(y|x)$

" $2k - 2LL$

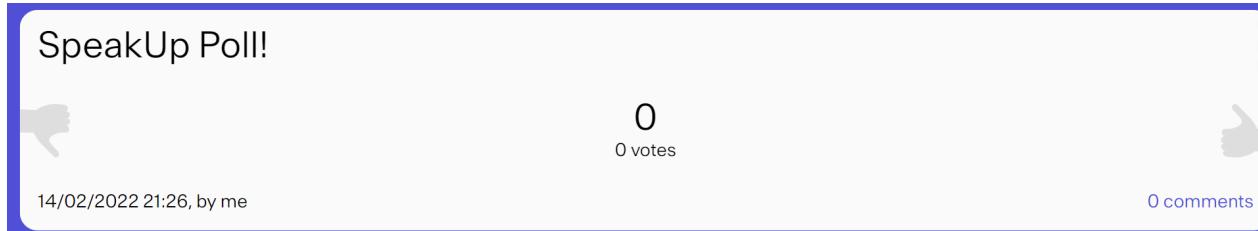
" $k \cdot h(n) - 2LL$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \rightarrow = 1 - \frac{(1-R^2)(n-1)}{(n-k-1)}$$

	Estimate	2.5_ci	97.5_ci	SE	DF	T-stat	P-val	Sig
Intercept	3.410119	3.148091	3.672148	0.133123	285	25.616335	0.000000	***
ch_time_in_prob_sum	0.000157	0.000094	0.000220	0.000032	285	4.921856	0.000001	***
wa_num_subs_perc_correct	0.716132	0.035683	1.396581	0.345700	285	2.071542	0.039208	*

Influence of input variables

$$grade = 3.4 + 0.000016 \cdot time_in_problem + 0.72 \cdot percentage_correct$$

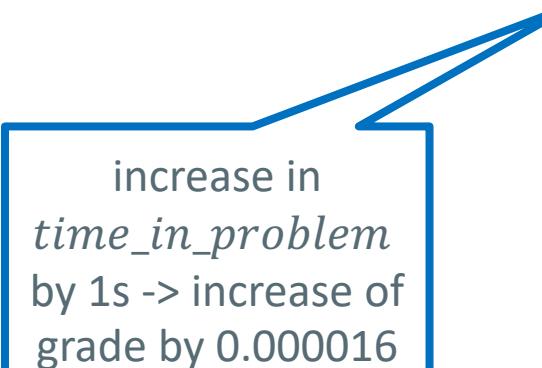


Which of the input variables has the largest impact on *grade*?

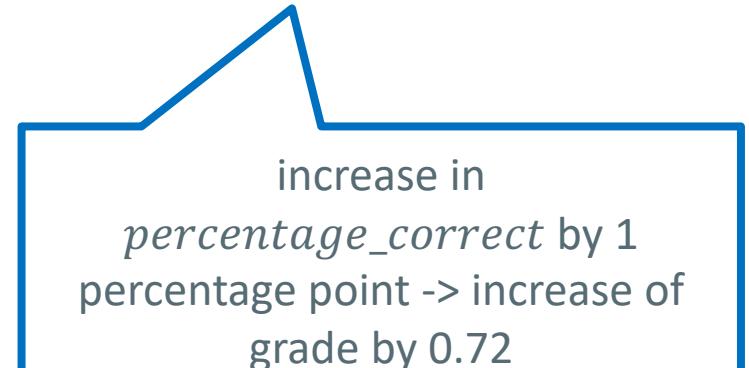
- a) *time_in_problem*
- b) *percentage_correct*
- c) I don't know

Different units of measurements

$$grade = 3.2 + 0.000016 \cdot time_in_problem + 0.72 \cdot percentage_correct$$



increase in
time_in_problem
by 1s -> increase of
grade by 0.000016



increase in
percentage_correct by 1
percentage point -> increase of
grade by 0.72

Transformation: Z-Scores

$$\tilde{x}_{n,d} = \frac{x_{n,d} - \bar{x}_d}{\sigma(x_d)}$$

$d = 1, \dots, D$

$n = 1, \dots, N$

- Standardization via z-score: $\tilde{x}_{n,d}$ denotes the distance between the raw feature $x_{n,d}$ and the sample mean \bar{x}_d (in units of the standard deviation)

Transformation: Example

$$grade = 4.05 + 0.35 \cdot time_in_problem + 0.15 \cdot percentage_correct$$

Example in Jupyter Notebook



Transformation: Summary

- Lets us compare the impact of input variables with different scales/units of measurements (e.g., time in problem in *seconds* and percentage correct)
- Reduces interpretability of individual input variables



Interpretation: Caveat

$$bodyfat = -45.95 + 0,99 \cdot abdomen - 0,33 \cdot weight$$

Can I conclude that heavier people (higher weight) have a lower bodyfat percentage?

Interpretation: Caveat

$$bodyfat = -45.95 + 0,99 \cdot abdomen - 0,33 \cdot weight$$

Can I conclude that heavier people (higher weight) have a lower bodyfat percentage?

- a) Yes
- b) No
- c) I don't know

SpeakUp Poll!!

0
0 votes

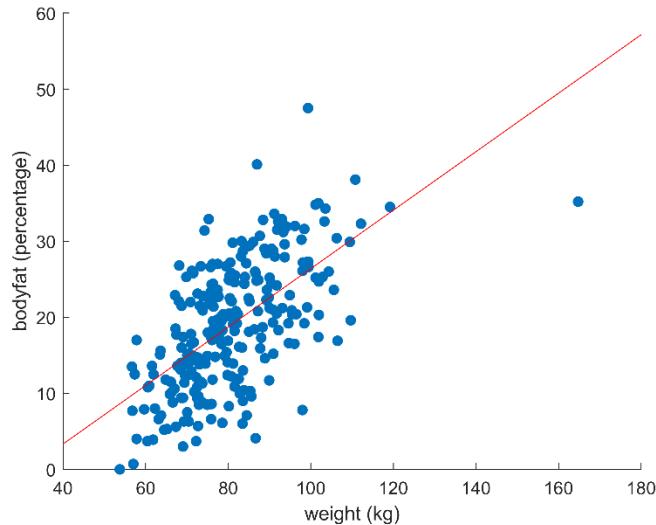
14/02/2022 21:26, by me

0 comments



Interpretation: Caveat

- There is a positive correlation between *weight* and *bodyfat* ($r = 0.61, p < .001$).



Interpretation: Caveat

- There is a positive correlation between *weight* and *bodyfat* ($r = 0.61, p < .001$).
 - *weight* only has a negative coefficient β in the context of *abdomen*, i.e. for fixed *abdomen* predictor
 - a predictor can only be interpreted **in the context** of the other predictors in the model

What means linear?

$$y_n = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D} + \epsilon_n$$

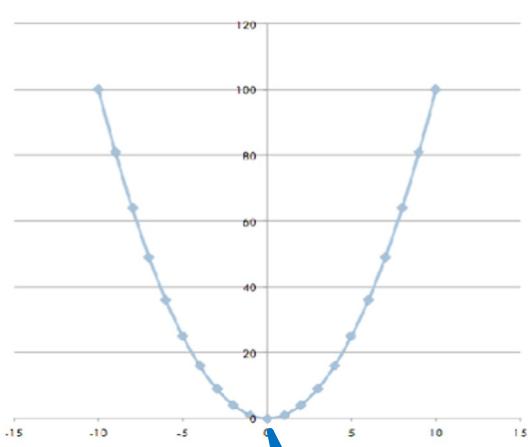
*Linear in the **parameters*** -> we can apply arbitrary functions to the raw input variables, e.g.,

- logarithms, exponentials
- polynomials
- inverse

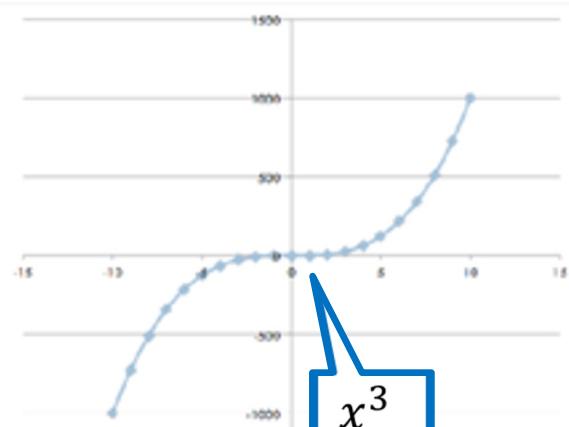
(time in problem)²

What means linear

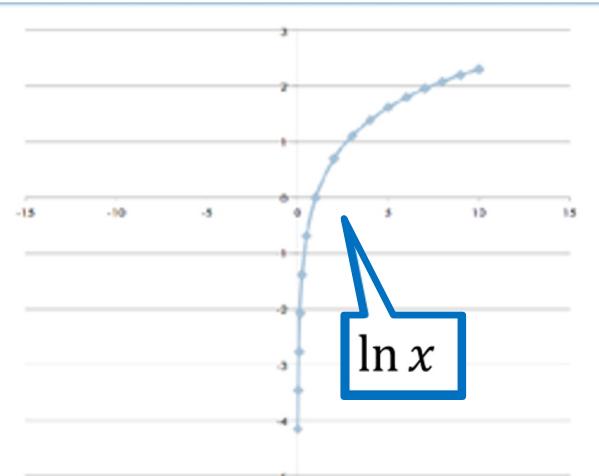
Different transformations



$$x^2$$



$$x^3$$

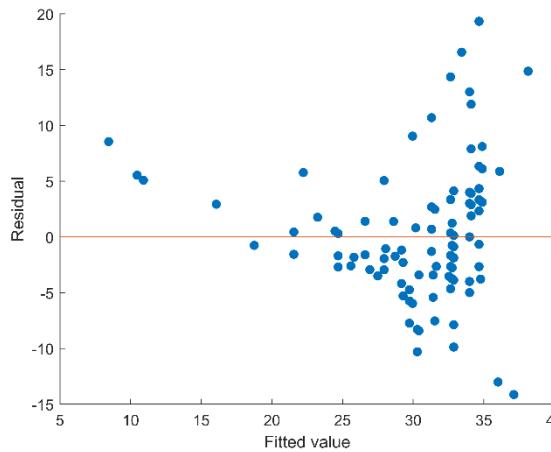
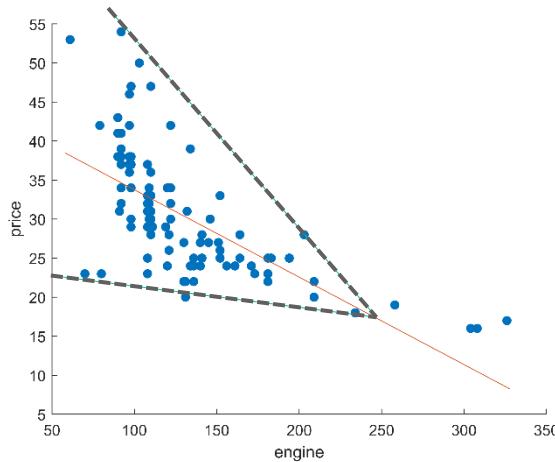


$$\ln x$$

Restrictions of linear models

For some cases, linear regression models are not appropriate:

- the variance of y depends on the mean



Assumption for statistics (t-test, chi, etc.):
 $\epsilon \sim N(0, \sigma^2)$

Restrictions of linear models

For some cases, linear regression models are not appropriate:

- the variance of y depends on the mean
- the range of y is restricted

$$\#bycycles = -2291 + 83 \cdot maxTemp - 13 \cdot minTemp - 890 \cdot precipitation$$

→ prediction \hat{y} can be negative...

Agenda

- Linear Regression
- **Generalized Linear Models**
- Mixed-Effect Models
- Performance Metrics
- Regression for Time-Series



Generalized Linear Models

A generalized linear model is composed of a **linear predictor**

$$\pi_n = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D}$$

and a **link function**

$$g(\mu_n) = \pi_n$$

with $\mu_n = E[Y|X = x_n]$

 conditional expectation

Generalized Linear Models

Conditional expectation: the mean μ_n depends on the values of independent variables x_n

is composed of a **linear predictor**

$x_{n,1}$

Each y_n represents the realization of the random variable Y , which is distributed according to a specific probability distribution

and a

function

$$g(\mu_n) = \pi_n$$

$$\text{with } \mu_n = E[Y|X = x_n]$$

\hat{y}_n

,

Generalized Linear Models

A generalized linear model is composed of a **linear predictor**

$$\pi_n = \beta_0 + \beta_1 x_{n,1}$$

In practice (for parameter fitting): observed values y_n are assumed to represent μ_n

and a **link function**

$$g(\mu_n) = \pi_n$$

with $\mu_n = E[Y|X = x_n]$

Logistic Regression

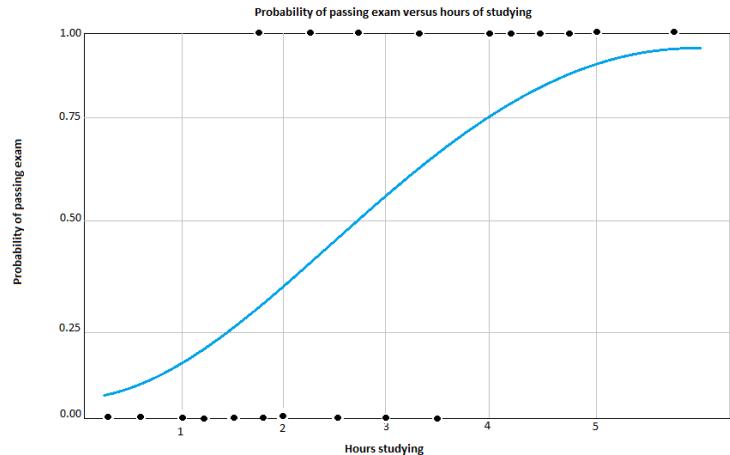
In logistic regression, the link function is

$$g(\mu_n) = \log\left(\frac{\mu_n}{1 - \mu_n}\right)$$

and therefore (for fitting)

$$\log\left(\frac{y_n}{1 - y_n}\right) = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D}$$

$$y_n = \frac{1}{1+e^{-\beta x}}$$



$$\text{Supp } \mathcal{T} : \{0, 1\}$$

Poisson Regression

In Poisson regression, the link function is

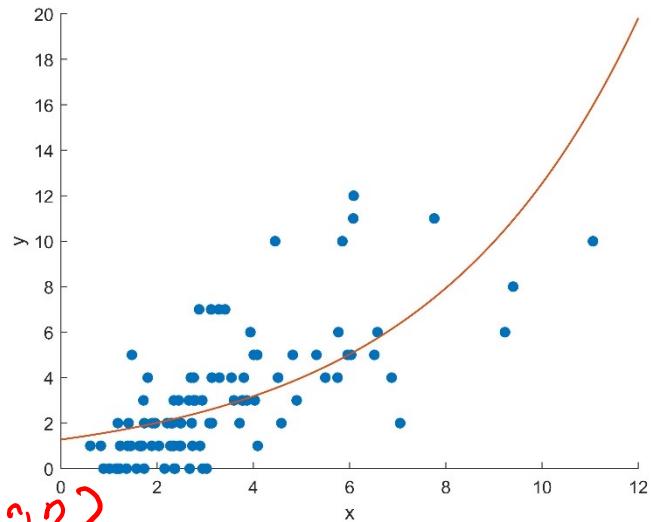
$$g(\mu_n) = \log(\mu_n)$$

and therefore (for fitting)

$$\log(\mu_n) = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D}$$

$$Y_n = e^{(\beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D})}$$

Support: $0, 1, 2, \dots,$



Linear Regression as a special case

For the linear regression, the link function is

$$g(\mu_n) = \mu_n$$

and therefore (for fitting)

$$y_n = \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_D x_{n,D}$$



Example

What type of model would you use for the following tasks?

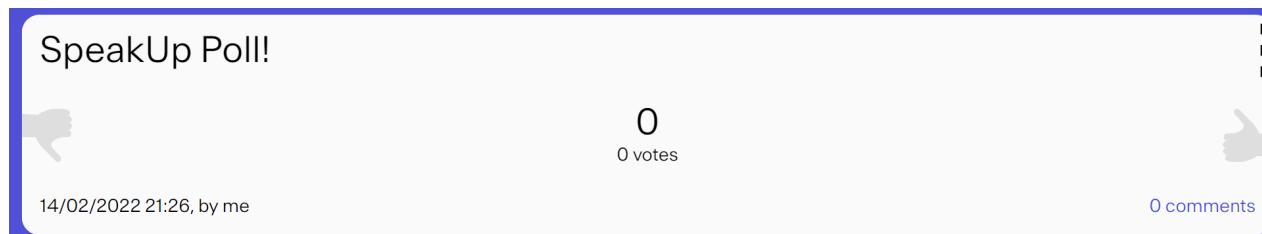
1. Predict the **number of awards** earned by students at one high school.

Predictors include the type of program in which the student was enrolled (e.g., vocational, general or academic) and the score on their final exam in math.

(a) Generalized Linear Model

(b) Logistic Regression

(c) Poisson Regression



Example

What type of model would you use for the following tasks?

2. Predict whether a student will **solve a task correctly**. Predictors include the difficulty of the task and the number of tasks the student has already solved.

(a) Generalized Linear Model

(b) Logistic Regression

(c) Poisson Regression

Linear Regression

SpeakUp Poll!



0
0 votes



14/02/2022 21:26, by me

0 comments

Example

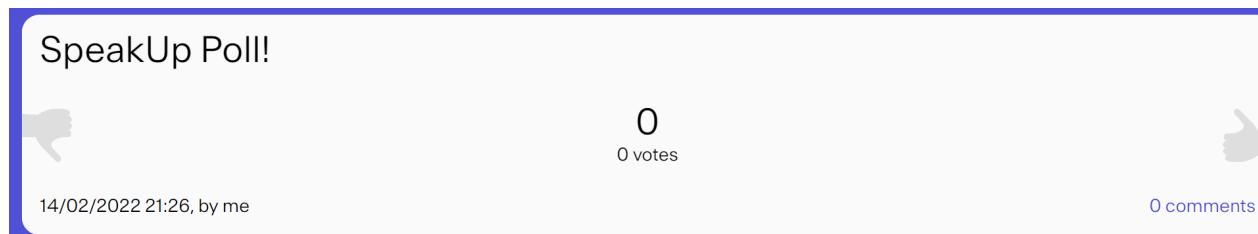
What type of model would you use for the following tasks?

3. Predict the **profit (in \$)** of a company based on their advertising budget on Youtube.

(a) Linear Regression

(b) Logistic Regression

(c) Poisson Regression



Agenda

- Linear Regression
- Generalized Linear Models
- **Mixed-Effect Models**
- Performance Metrics
- Regression for Time-Series

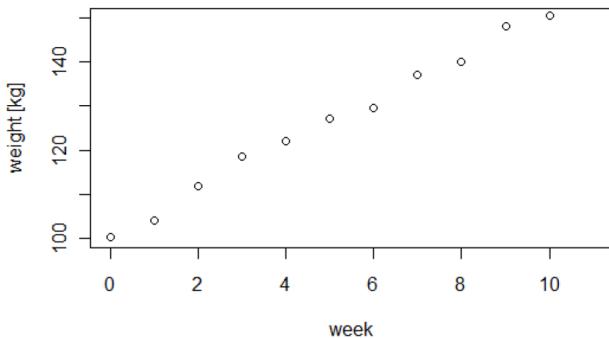


Why mixed-effect models?

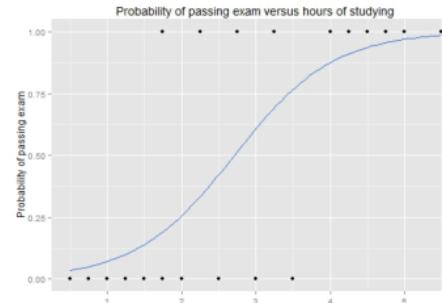
- Useful when we are dealing with **correlated** samples
 - **Grouping** of subjects (e.g., students within a classroom)
 - **Repeated measurements** on each subject over time (e.g., student in flipped classroom course over 10 weeks)

Generalized Linear Models

- Example 1: strength gain by weight training
- Example 2: probability of passing exam of a course c depending on the hours studied



$$y_n = \beta_0 + \beta_1 x_{n,1}$$

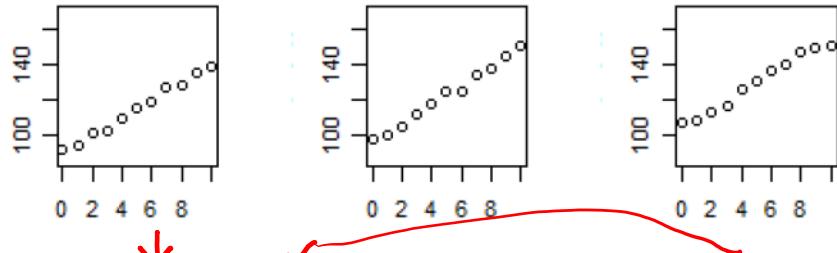


$$\log\left(\frac{y_n}{1 - y_n}\right) = \beta_0 + \beta_1 x_{n,1}$$

“Fixed” Effects

Generalized Linear Mixed Effects Model

- Example 1: strength gain by weight training
 - Each person has individual starting strength



$$y_n = \beta_0 + u_n + \beta_1 x_{n,1} \quad u_n \sim N(0, \sigma_u^2)$$

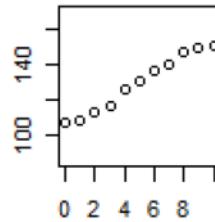
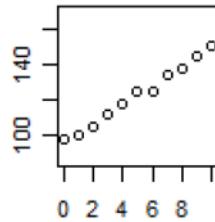
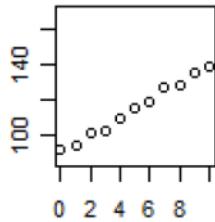
Random Intercept

“Fixed” Effects

“Random” Effect

Generalized Linear Mixed Effects Model

- Example 1: strength gain by weight training
 - Each person has individual starting strength



$$y_n = \beta_0 + u_n + \beta_1 x_{n,1} \quad u_n \sim N(0, \sigma_u^2)$$

Fitting the parameters:

- Fixed effects only: linear least squares
- Mixed effects: maximum likelihood estimation

“Fixed” Effects

“Random” Effect

“Mixed” Effects

Generalized Linear Mixed Effects Model

- In our case, students come from different origins and we assume that students from the same origin are more similar (same education system)
- We therefore use origin (*category*) as a proxy for prior knowledge and add a random intercept to the model

$$passed \sim 1 | category + percentage_correct$$

Agenda

- Linear Regression
- Generalized Linear Models
- Mixed-Effect Models
- **Performance Metrics**
- Regression for Time-Series



Usage

- *Interpretation*: analyze the relationships between the variables (what effect the input variables have on the output variable)
 - *Prediction*: predict the output for a new (unseen) input vector x
-

Regression: R²

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where $SS_{res} = \sum_i (y_i - f(x_i))^2$ and $SS_{tot} = \sum_i (y_i - \bar{y})^2$

- Can be interpreted as the fraction of explained variability of the data
- Often used when the goal is *interpretation*
- Often used in the fields of Psychometrics, Learning Sciences, Psychology, etc.

Regression: MAE and RMSE

- Mean absolute error:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - f(x_i)|$$

- Root mean squared error:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2}$$

- Often used when the goal is *prediction*
- **RMSE is largely preferred to MAE**

Hypothetical Example

- Given:
 - Student giving correct answers 70% of the time
 - Model A: predicts correct 70% of the time
 - Model B: predicts 100% correctness



MAE: Model B is better

- 70% of the time the student gives a correct answer (response = 1)
 - Model A: absolute error = 0.3
 - Model B: absolute error = 0.0
 - 30% of the time the student answers wrong (response = 0)
 - Model A: absolute error = 0.7
 - Model B: absolute error = 1.0
 - $MAE_A = 0.42$, $MAE_B = 0.30$
-

RMSE: Model A is better

- $RMSE_A = 0.21$
- $RMSE_B = 0.30$
- **$RMSE$ penalizes large errors heavier**

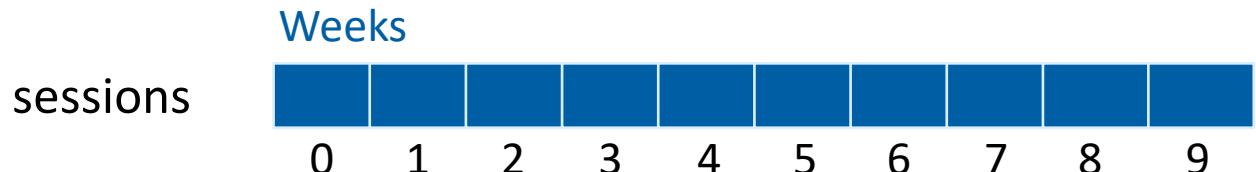
Agenda

- Linear Regression
- Generalized Linear Models
- Mixed-Effect Models
- Performance Metrics
- **Regression for Time-Series**



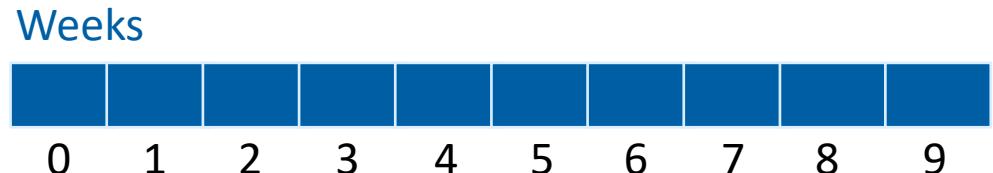
Time Series – Our flipped classroom case

Student i



•
•
•

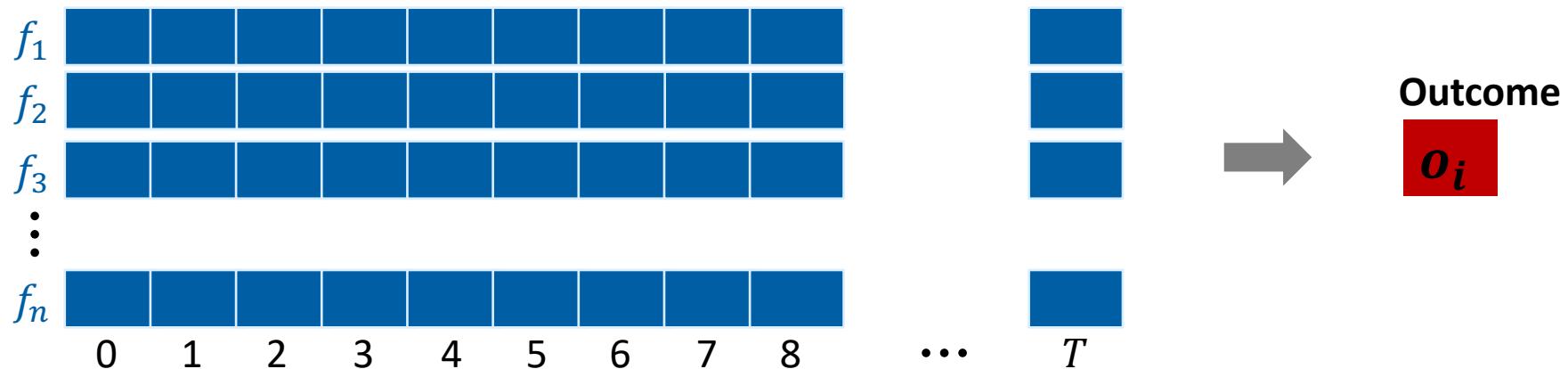
submissions_correct



Time Series – Possible Tasks

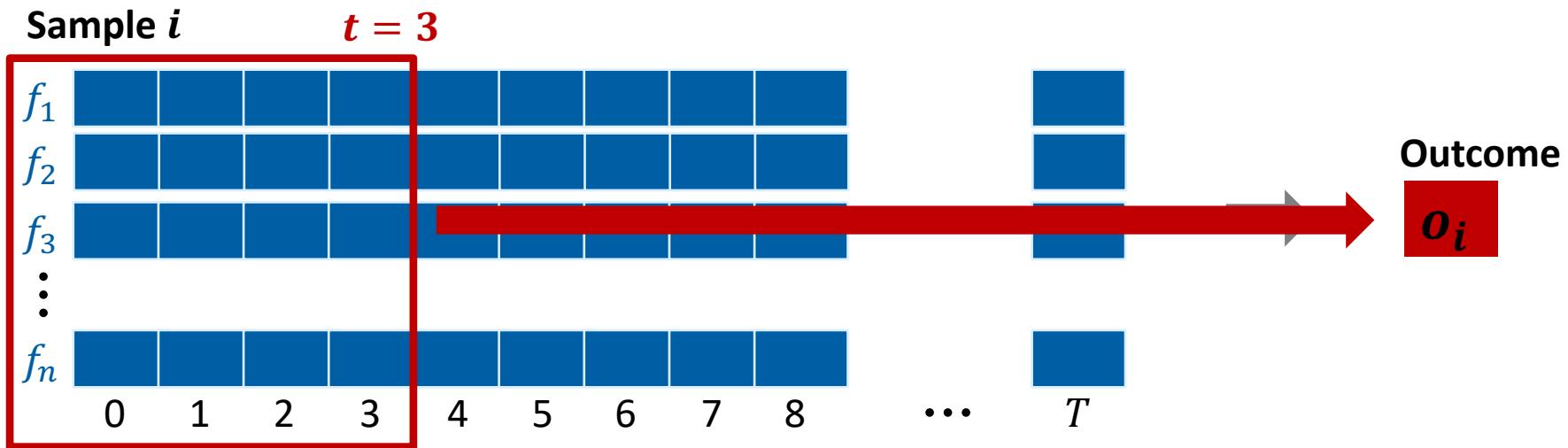
- Prediction of a target variable after $t < T$ time steps, where T is the total number of time steps

Sample i



Time Series – Possible Tasks

- Prediction of a target variable after $t < T$ time steps, where T is the total number of time steps



Handling Time Series Data

- Flattening

$$\text{grade} = \beta_0 + \beta_1 \cdot \text{time_old_week} \\ + \beta_2 \cdot \text{time_prob_week} + \dots$$

- The number of parameters of the model depends on the number of time steps of the model .

- Aggregation

- Averaging across weeks
- Accumulating across weeks

$$\text{grade} = \beta_0 + \beta_1 \cdot \text{average_time_prob} \\ + \beta_2 \cdot \text{variance_in_problem}$$

Example – Prediction of Grade

- Prediction of grade after $t < T$ weeks
- We will try to predict after 5 weeks and after 10 weeks

$$grade \sim (1 | category) + average_percentage_correct [week n]$$

Your Turn – Prediction of Passing

- Adjust the example equation to predict after week 5 and then, whether students will pass the exam
- Extension (if you have time):
 - Improve the accuracy of the model by adding more features
 - Justify, why you selected the chosen features and send us your RMSEs.

Your Turn – Feedback

Do you want feedback or have questions?

Upload your Jupyter Notebook here:

<https://go.epfl.ch/notebooks-mlbd>

Summary

- Linear regression is a useful framework for interpreting data and making predictions
 - Caveat: be careful when interpreting the models
 - Linear regression is flexible, i.e. arbitrary functions can be applied to the raw input data
 - Generalized linear models are a more general framework appropriate for response variables from exponential family distributions
 - Mixed models allow for capturing correlation in the data
 - Modeling time series data requires some type of aggregation
-

Classification

Machine Learning for Behavioral Data
March 13, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction
8	Time Series Prediction

Complete pipeline for one use case:

- Data exploration
- Prediction
- Model evaluation

Getting ready for today's lecture...

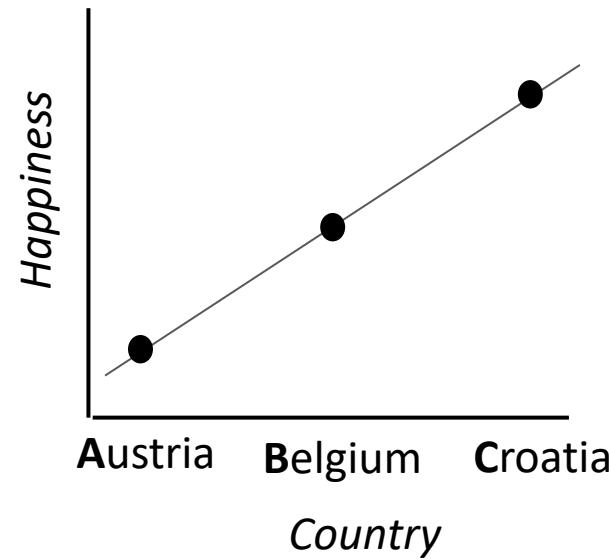
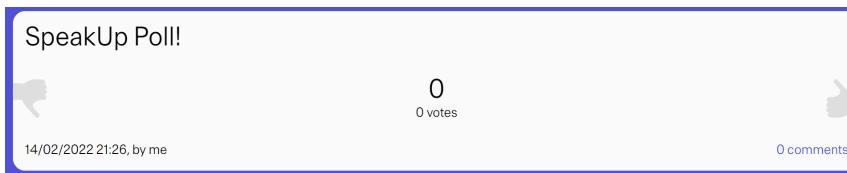
- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

- [Exploration] Based on the provided graph, what can you say about the relationship between country and happiness?

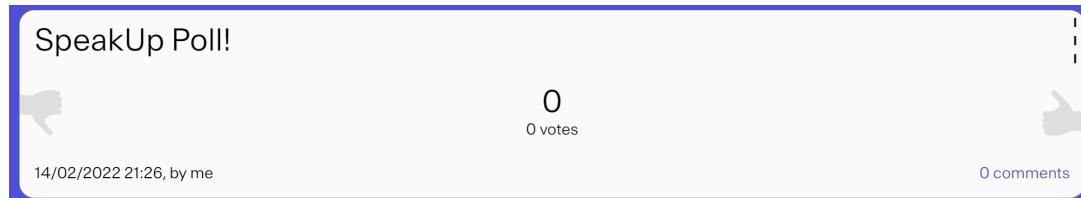


- a) Happiness increases with country
- b) Happiness decreases with country
- c) It is not possible to compute a correlation between country and happiness

Short quiz about the past...

[Regression] Which GLM family should you use when the output variable is continuous?

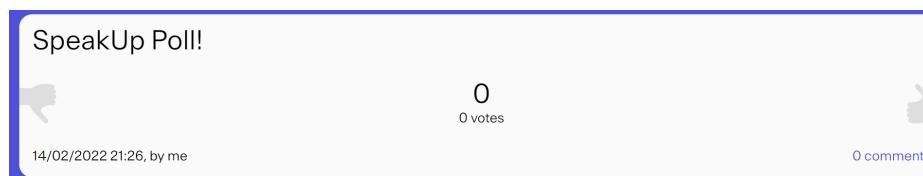
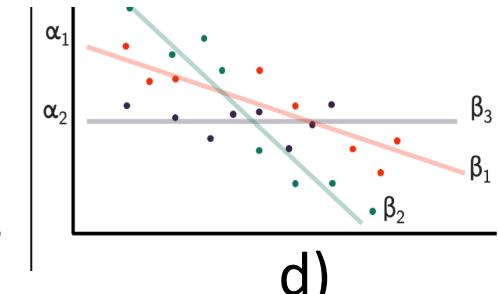
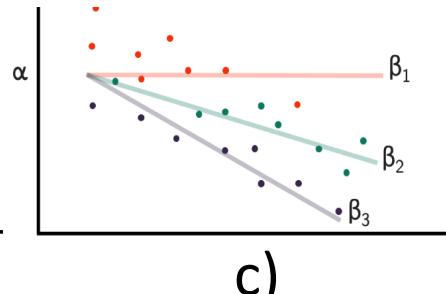
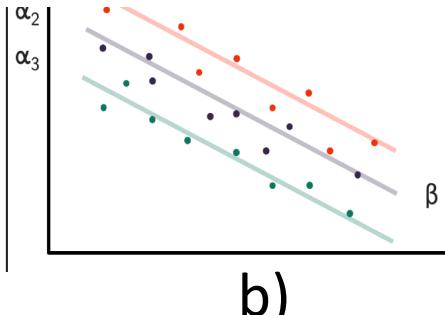
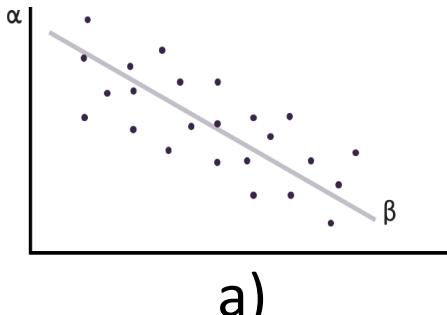
- a) Binomial
- b) Poisson
- c) Gaussian



```
model = lmer("grade ~ (1|user) + submissions_wrong",
             data=df_train, family='??????')
```

Short quiz about the past...

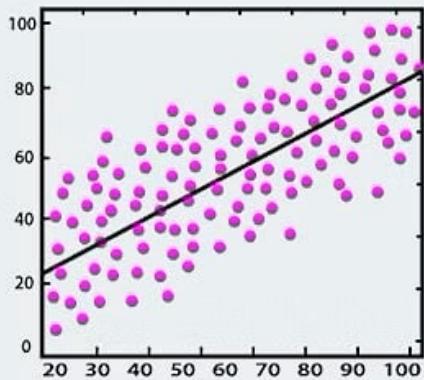
[Regression] Which of the following are examples of models with **only fixed effects**? In the plots, α denotes intercept, β denotes slope.



Idea

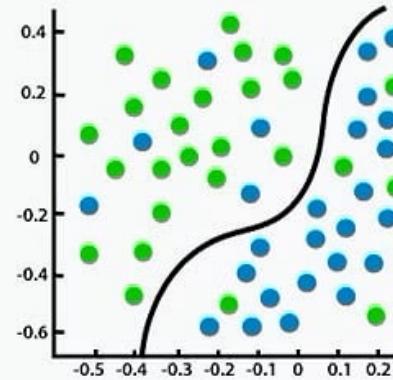
- In classification, a single aspect of the data (predicted variable) is modeled by some combination of other aspects of the data (features)
 - The predicted variable is **categorical** (set of classes)
 - Examples:
 - Prediction of dropout in massive open online courses (binary)
 - Exploration of user categories in a simulation (multiclass)
-

Classification



Regression

versus



Classification

Today's Use Case: Flipped Classroom Course

- Participants: 288 EPFL students of a course taught in *flipped classroom* mode with a duration of 10 weeks
 - Structure:
 - Preparation: watch videos (and solve simple quizzes) on **new content** at home as a preparation for the lecture
 - Lecture: discuss open questions and solve more complex tasks
 - Lab session: solve paper-an-pen assignments
 - Data: clickstream data (all interactions of the student)
 - Binary classification: 2 Classes to predict: Pass, Fail
-

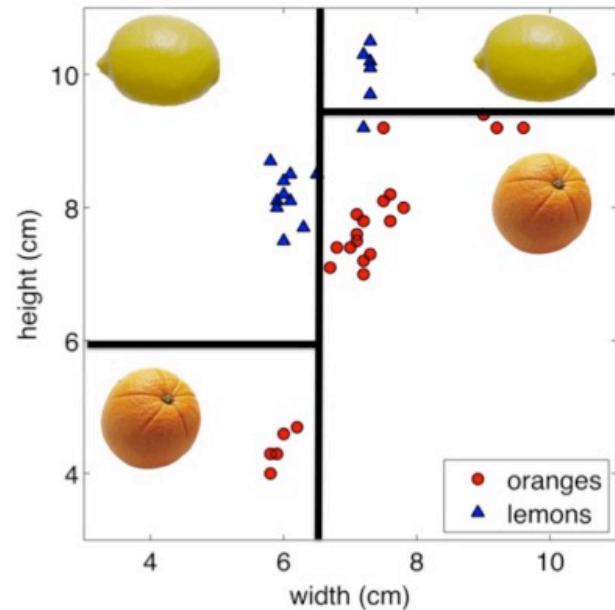
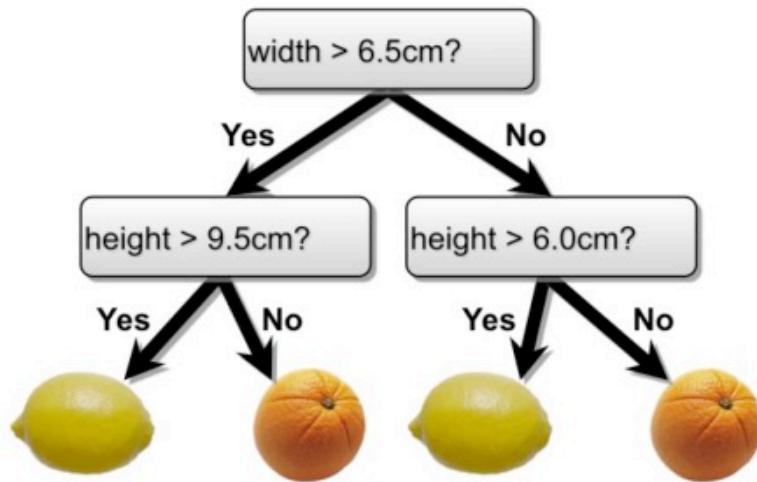
Agenda

- Traditional Classification Methods:
 - Decision Trees
 - Random Forest
 - K-Nearest Neighbor
 - Logistic Regression
 - Performance Metrics
 - Classification of Time Series
-

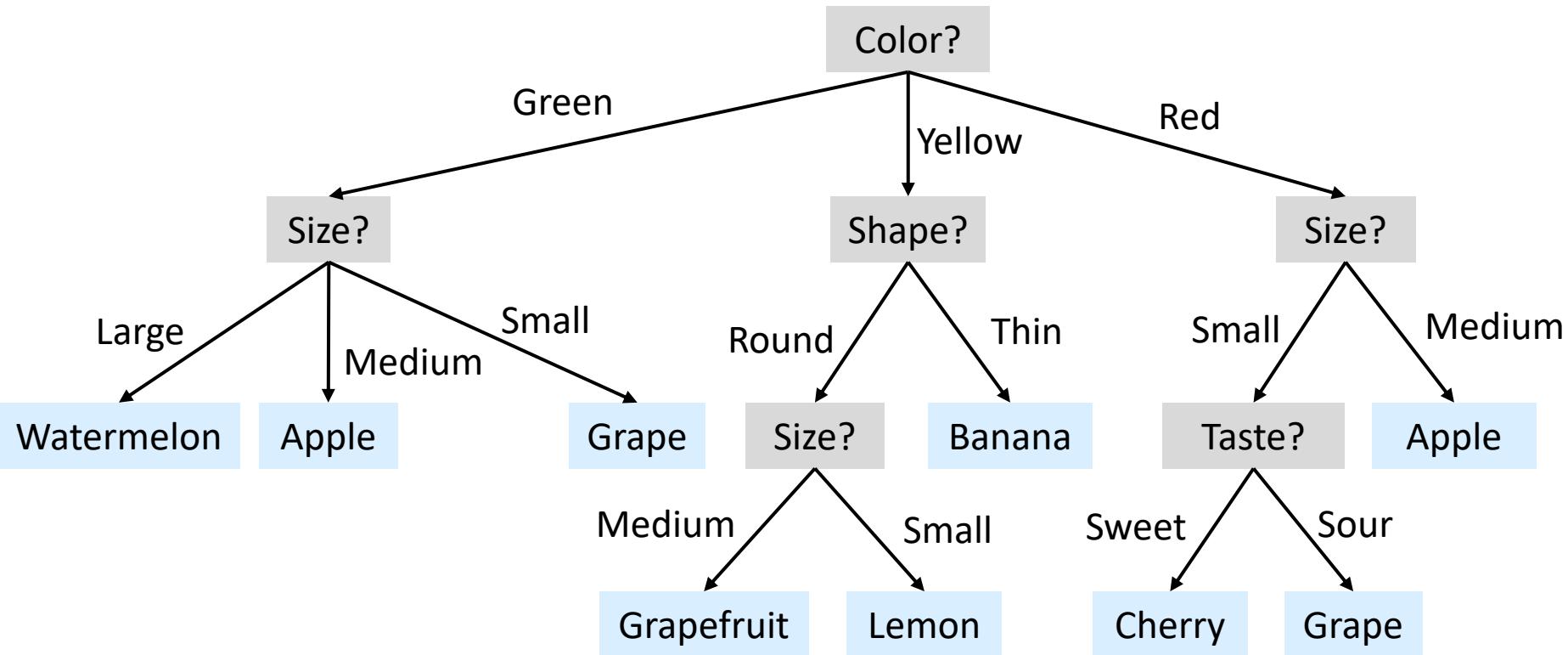
Decision Trees - Idea

1. Pick an attribute/feature, do a simple yes/no test
 2. Conditioned on a choice, pick another attribute, do another test
 3. In the leaves, assign a class with majority vote
 4. Do other branches as well
-

Decision Trees - Example



Decision Trees - Categorical Features



Decision Trees - Construction

- Which attributes do we choose?
 - In which order?
 - In the case of continuous attributes, how do we choose the threshold value?
-

Construction Algorithm

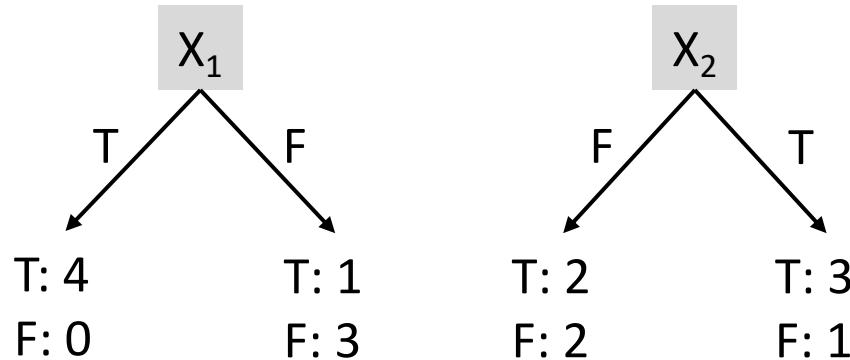
- Learning the simplest (smallest) decision tree is an NP complete problem (translation: it is hard !!!)
 - Greedy heuristic:
 1. Pick an attribute to split at a non-terminal node
 2. Split example into groups based on attribute value
 3. For each group:
 - No examples -> return majority of parent node
 - All examples from same class -> return class
 - Else: loop to step 1
-

Construction Algorithm

- Learning the simplest (smallest) decision tree is an NP complete problem (translation)
- Greedy heuristic:
 1. Pick an attribute to split at a non-terminal node
 2. Split example into groups based on attribute value
 3. For each group:
 - No examples -> return majority of parent node
 - All examples from same class -> return class
 - Else: loop to step 1

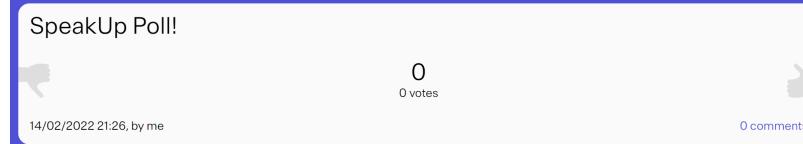
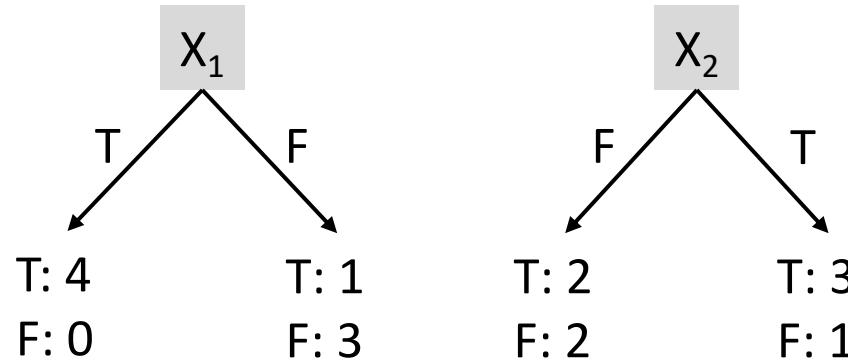
How do we pick the best attribute ?

Picking the best attribute



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

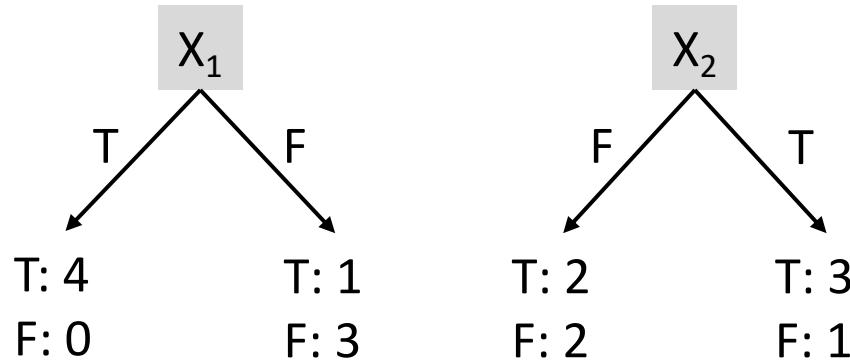
On which attribute would you split?



- a) X_1
- b) X_2

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Picking the best attribute

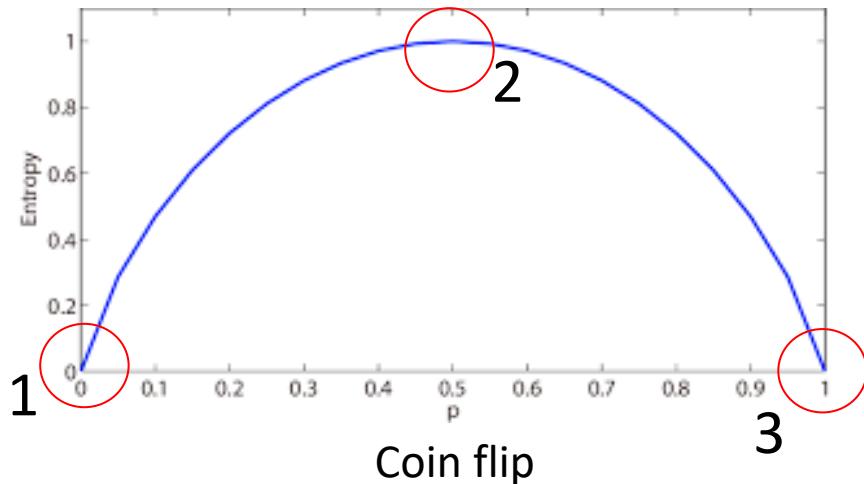


X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

How to do it formally?
-> Information theory !!!

Entropy

- Describes the level of “**uncertainty**” or “**surprise**” about a random variable’s possible outcome



1. Will always land on heads
2. 50/50
3. Will always land on tail

Entropy

$$H(X) = - \sum_{x \in X} p(x) \cdot \log_2 p(x)$$

Information content or surprise of event x .

Intuition: low probability = high surprise/information content.

Entropy: Expected surprise/information content.

Conditional Entropy

P(X,Y)

X\Y	Cloudy	Not Cloudy
Rain	0.24	0.01
No Rain	0.25	0.50

- Specific Conditional Entropy: what is the entropy of cloudiness Y, given that **it is raining?**

$$H(Y|X = x) = - \sum_{y \in Y} p(y|x) \cdot \log_2 p(y|x)$$

Conditional Entropy

$P(X,Y)$

X \ Y	Cloudy	Not Cloudy
X		
Rain	0.24	0.01
No Rain	0.25	0.50

- Specific Conditional Entropy: what is the entropy of cloudiness Y, given that **it is raining?**

$$H(Y|X = x) = - \sum_{y \in Y} p(y|x) \cdot \log_2 p(y|x)$$

- Expected Conditional Entropy: what is the **expected entropy** of cloudiness Y, given “raininess” X?

$$H(Y|X) = \sum_{x \in X} p(x) \cdot H(Y|X = x)$$

Information Gain

$P(X,Y)$

X \ Y	Cloudy	Not Cloudy
X		
Rain	0.24	0.01
No Rain	0.25	0.50

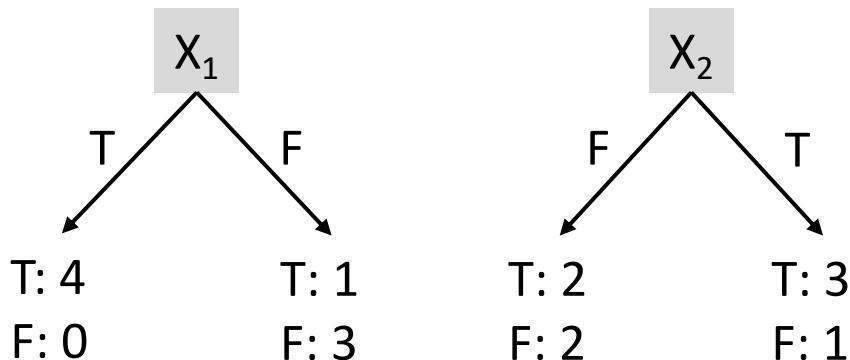
- **Information Gain**: how much information about cloudiness (Y) do we get by discovering whether it is raining (X)?

$$I(Y; X) = H(Y) - H(Y|X)$$

Picking the best attribute

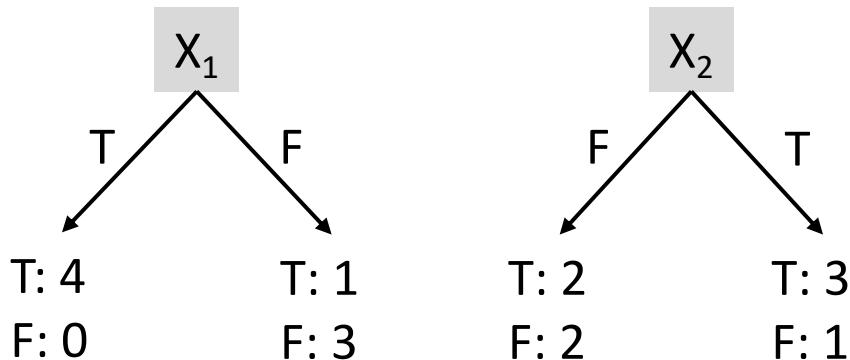
Should I pick X_1 or X_2 to gain the most information about Y ?

Is $I(Y; X_1) > I(Y; X_2)$?



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Picking the best attribute



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

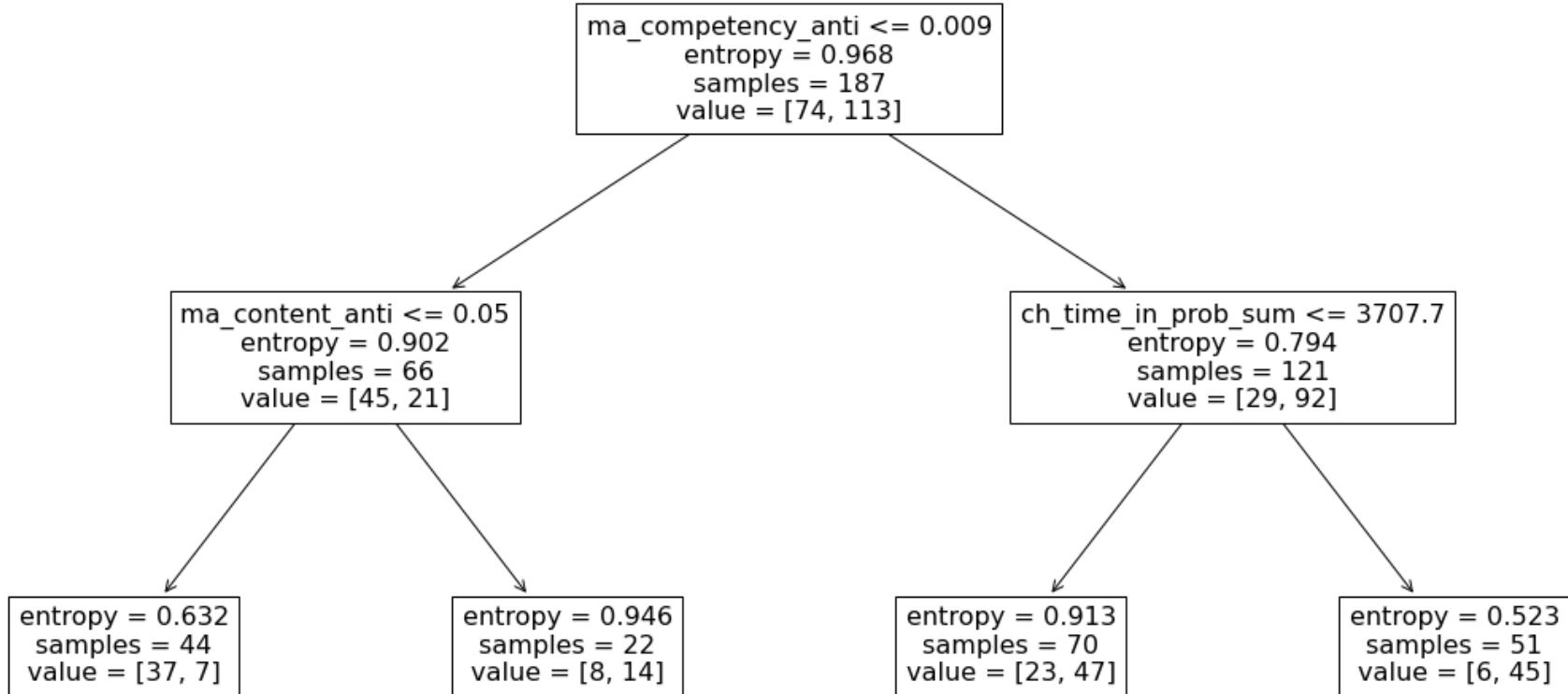
What makes a good tree?

- Not too small: needs to handle important, but possibly subtle distinctions in data
- Not too big:
 - Avoid overfitting to training examples
 - Computational efficiency (avoid redundant, spurious attributes)

→ Pruning strategies:

- Stop splitting a node when the number of samples falls below a certain threshold
- Grow a full tree, do bottom-up cross-validation: two leaves can be merged and labeled with the majority class if classification accuracy (on a validation set!) does not get worse

Decision Tree - Example



Agenda

- Traditional Classification Methods:
 - Decision Trees
 - Random Forest
 - K-Nearest Neighbor
 - Logistic Regression
 - Performance Metrics
 - Classification of Time Series
-

Random Forest - Idea

- **Ensemble Method:**
 - Take a collection of weak (simple) learners
 - Combine their predictions to obtain a better result
 - **Bagging:** train learners on different samples of the data and then combine their output (e.g., majority vote or average)
-

Random Forest - Algorithm

Grow K trees on datasets sampled from the original data set (size N) with replacement (bootstrap samples), $d =$ number of features.

- Draw K bootstrap samples of size N (bootstrap means that different samples can have elements in common)
 - Grow each decision tree by selecting a *random set of m out of d features* at each node, and choosing the best feature to split on.
 - Aggregate the predictions of the trees (majority vote or average) to produce the final prediction
-

Random Forest - Algorithm

Grow K trees on datasets sampled (size N) with replacement (bootstrap) features.

Each tree is trained on different data

- Draw K bootstrap samples of size N (bootstrap means that different samples can have elements in common)
- Grow each decision tree by selecting a *random set of m out of d features* at each node, and choosing the best feature to split on.
- Aggregate the predictions of the trees (majority vote or average) to produce the final prediction

Random Forest - Algorithm

Grow K trees on datasets sampled from the original data set (size N) w

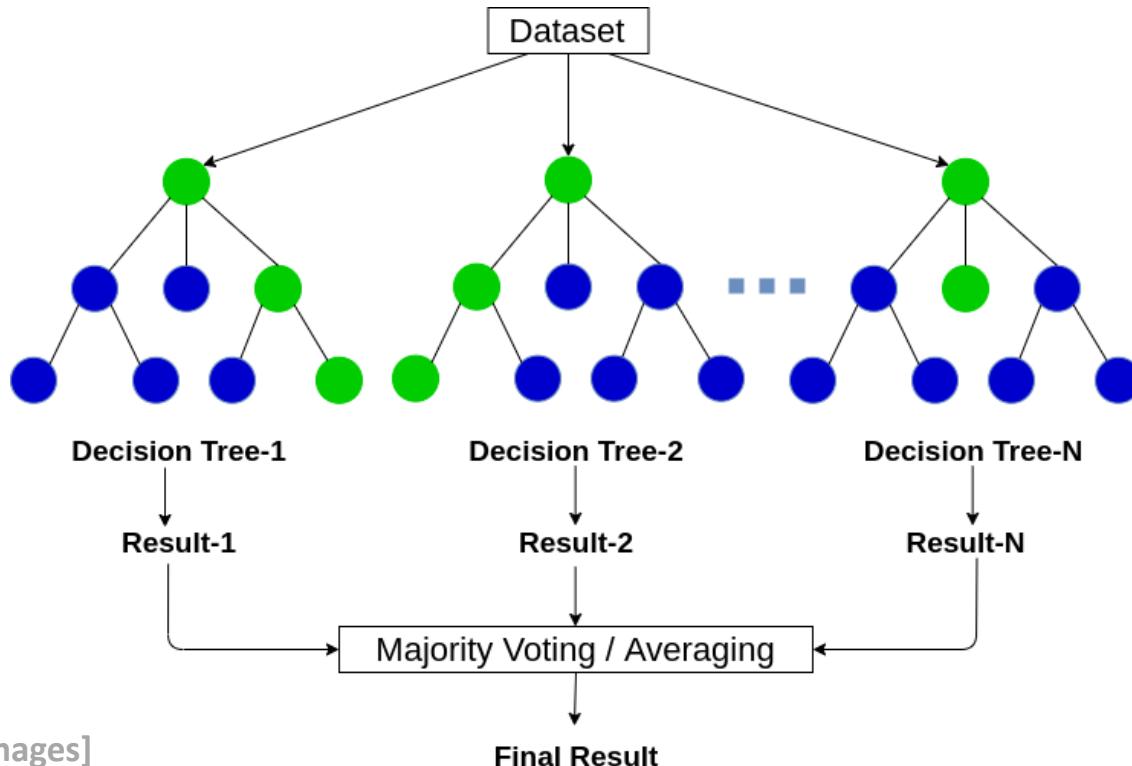
Corresponding nodes in different trees

feat will (usually) not use the same feature

for splitting

- D
- Grow each decision tree by selecting a *random set of m out of d features* at each node, and choosing the best feature to split on.
- Aggregate the predictions of the trees (majority vote or average) to produce the final prediction

Random Forest - Illustration



[Image Source: Google Images]

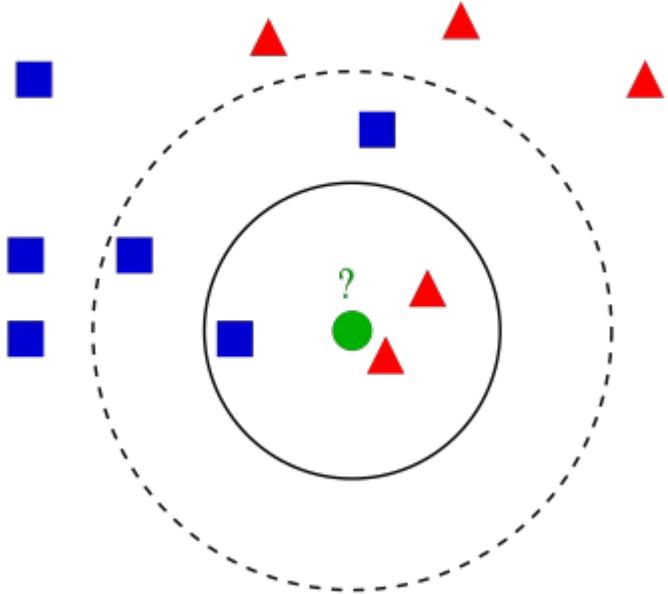
Summary

- Decision trees are simple, but
 - sensitive to small perturbations in the data
 - tend to overfit
 - Random forests
 - Reduce overfitting in decision trees and can improve accuracy
 - Are versatile (classification, regression, continuous/categorical variables)
 - Easy to implement and parallelize
 - Still a popular algorithm in practice (for dense data)
 - Not so easy to interpret...
-

Agenda

- Traditional Classification Methods:
 - Decision Trees
 - Random Forest
 - K-Nearest Neighbor
 - Logistic Regression
 - Performance Metrics
 - Classification of Time Series
-

kNN- Illustration



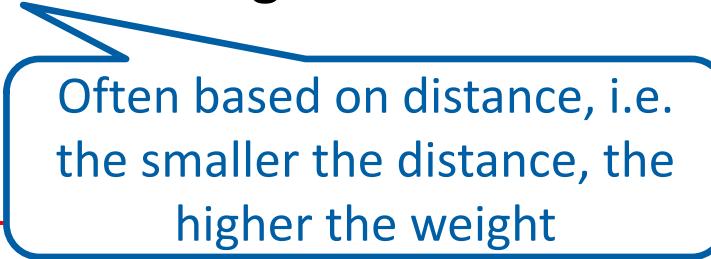
- Alternative to parametric models are **non-parametric** models
- Learning amounts to simply storing training data: ■ ▲
- **Test instances** are classified using similar training instances
- kNN: k-nearest neighbors

kNN- Algorithm

- Training data: samples x_n ($n = 1, \dots, N$) with class labels c ($c = 1, \dots, C$)
- Classification of test sample x^*
 - Find the k nearest neighbors of x^*
 - Predicted class $\hat{c}^* =$ majority vote of k nearest neighbors
 - Option: give nearest neighbors different weights

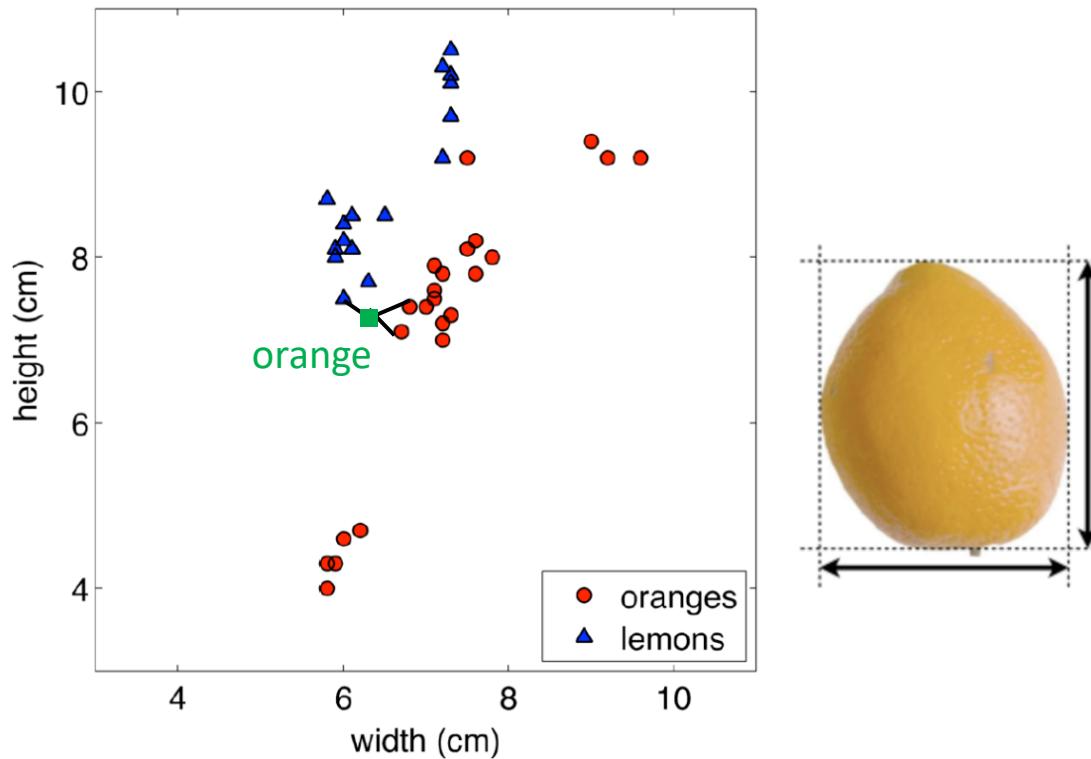
kNN- Algorithm

- Training data: samples x_n ($n = 1, \dots, N$) with class labels c ($c = 1, \dots, C$)
- Classification of test sample x^*
 - Find the k nearest neighbors of x^*
 - Predicted class $\hat{c}^* =$ majority vote of k nearest neighbors
 - Option: give nearest neighbors different weights



Often based on distance, i.e.
the smaller the distance, the
higher the weight

Example: 3-nearest neighbor



How do we choose k ?

- Larger k may lead to better performance
 - But if we set k too large we may end up looking at samples that are not neighbors (are far away from the query)
 - Find k using leave-one-out cross-validation:
 - For each point x_n in the training data set, find the k nearest neighbors from the set of all *other* training samples
 - Predict \hat{c}_n as the majority vote of the k nearest neighbors
 - Measure the classification accuracy for different values of k
 - Choose k with the highest classification accuracy on the training data set
-

kNN: what distance could we use?

- Comparing the number of sessions students had in a MOOC:

$u_1: [5,3,7,8,10,2]$ $u_2: [1,9,10,2,3,2]$ $u_3: [4,5,2,8,7,6]$

- Comparing users by the movies they have watched:

$u_1: \{"Frozen", "The Horse Whisperer", "Follow Me", "Notting Hill"\}$

$u_2: \{"Die Hard 1", "The Father", "Frozen", "Black Panther", "Casablanca"\}$

$u_3: \{"The Dark Knight", "Die Hard 1", "Wonder Woman", "Black Panther", "Logan", "Up"\}$

- Comparing weeks days of users (W: Work, O: Off, S: Sick)

$u_1: [O, W, W, O, W]$ $u_2: [W, W, W, W, W]$ $u_3: [W, W, S, W, O]$

- Comparing sequences of actions of users

$u_1: [O, A, P, E, T, F, G, F, G, H, I, O, N, K, U, P, E, L]$ $u_2: [O, S, I, E, P, L]$ $u_3: [O, R, C, C, T, A, A, S, S, P, L]$

- Comparing the relative amount of time users spent on watching videos, solving quizzes, etc.

$u_1: [0.2, 0.3, 0.1, 0.1, 0.3]$ $u_2: [0.8, 0.1, 0.0, 0.0, 0.1]$ $u_3: [0.1, 0.5, 0.3, 0.0, 0.1]$

kNN: similarity metrics

- Euclidean Distance: simple & fast for numbers

$$d(x, y) = \|x - y\|$$

- Jaccard Distance: for set data

$$d(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

- Hamming distance: for strings (with the same length)

$$d(x, y) = \sum_{i=1}^n (x_i \neq y_i)$$

kNN: similarity metrics

- **Levenshtein distance**: minimal number of single character edits (insertion, deletion, substitution) to change one string into the other
- **Longest common subsequence (LCS)**: string similarity measure, find the longest common subsequence between two sequences
- **Kullback-Leibler Divergence**: measures difference between two probability distributions (relative entropy)

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \cdot \log\left(\frac{P(x)}{Q(x)}\right)$$

kNN: what similarity metric could we use?

- Comparing the number of sessions students had in a MOOC:
 $u_1: [5,3,7,8,10,2]$ $u_2: [1,9,10,2,3,2]$ $u_3: [4,5,2,8,7,6]$
 - Comparing users by the movies they have watched:
 $u_1: \{\text{"Frozen"}, \text{"The Horse Whisperer"}, \text{"Follow Me"}, \text{"Notting Hill"}\}$
 $u_2: \{\text{"Die Hard 1"}, \text{"The Father"}, \text{"Frozen"}, \text{"Black Panther"}, \text{"Casablanca"}\}$
 $u_3: \{\text{"The Dark Knight"}, \text{"Die Hard 1"}, \text{"Wonder Woman"}, \text{"Black Panther"}, \text{"Logan"}, \text{"Up"}\}$
 - Comparing weeks days of users (W: Work, O: Off, S: Sick)
 $u_1: [O, W, W, O, W]$ $u_2: [W, W, W, W, W]$ $u_3: [W, W, S, W, O]$
 - Comparing sequences of actions of users
 $u_1: [O, A, P, E, T, F, G, F, G, H, I, O, N, K, U, P, E, L]$ $u_2: [O, S, I, E, P, L]$
 $u_3: [O, R, C, C, T, A, A, S, S, P, L]$
 - Comparing the relative time users spent on watching videos, solving quizzes, etc.
 $u_1: [0.2, 0.3, 0.1, 0.1, 0.3]$ $u_2: [0.8, 0.1, 0.0, 0.0, 0.1]$ $u_3: [0.1, 0.5, 0.3, 0.0, 0.1]$
-
- The diagram illustrates five examples of kNN similarity metrics, each involving three user vectors (u_1 , u_2 , u_3) and a corresponding similarity metric:
- Numerical: Euclidean**: Compares session counts. Vectors: $u_1: [5,3,7,8,10,2]$, $u_2: [1,9,10,2,3,2]$, $u_3: [4,5,2,8,7,6]$.
 - Sets: Jaccard**: Compares movie sets. Vectors: $u_1: \{\text{"Frozen"}, \text{"The Horse Whisperer"}, \text{"Follow Me"}, \text{"Notting Hill"}\}$, $u_2: \{\text{"Die Hard 1"}, \text{"The Father"}, \text{"Frozen"}, \text{"Black Panther"}, \text{"Casablanca"}\}$, $u_3: \{\text{"The Dark Knight"}, \text{"Die Hard 1"}, \text{"Wonder Woman"}, \text{"Black Panther"}, \text{"Logan"}, \text{"Up"}\}$.
 - Same length Strings: Hamming/LCS/Levenshtein**: Compares weekly days. Vectors: $u_1: [O, W, W, O, W]$, $u_2: [W, W, W, W, W]$, $u_3: [W, W, S, W, O]$.
 - Strings: LCS/Levenshtein**: Compares action sequences. Vectors: $u_1: [O, A, P, E, T, F, G, F, G, H, I, O, N, K, U, P, E, L]$, $u_2: [O, S, I, E, P, L]$, $u_3: [O, R, C, C, T, A, A, S, S, P, L]$.
 - Proba: KL**: Compares relative time spent. Vectors: $u_1: [0.2, 0.3, 0.1, 0.1, 0.3]$, $u_2: [0.8, 0.1, 0.0, 0.0, 0.1]$, $u_3: [0.1, 0.5, 0.3, 0.0, 0.1]$.

kNN: similarity metrics

- There are many more similarity metrics (e.g., Cosine distance, Manhattan distance, Mahalanobis distance)
- These are all standard metrics – we will look detailed into metrics for comparing sequences in later lectures

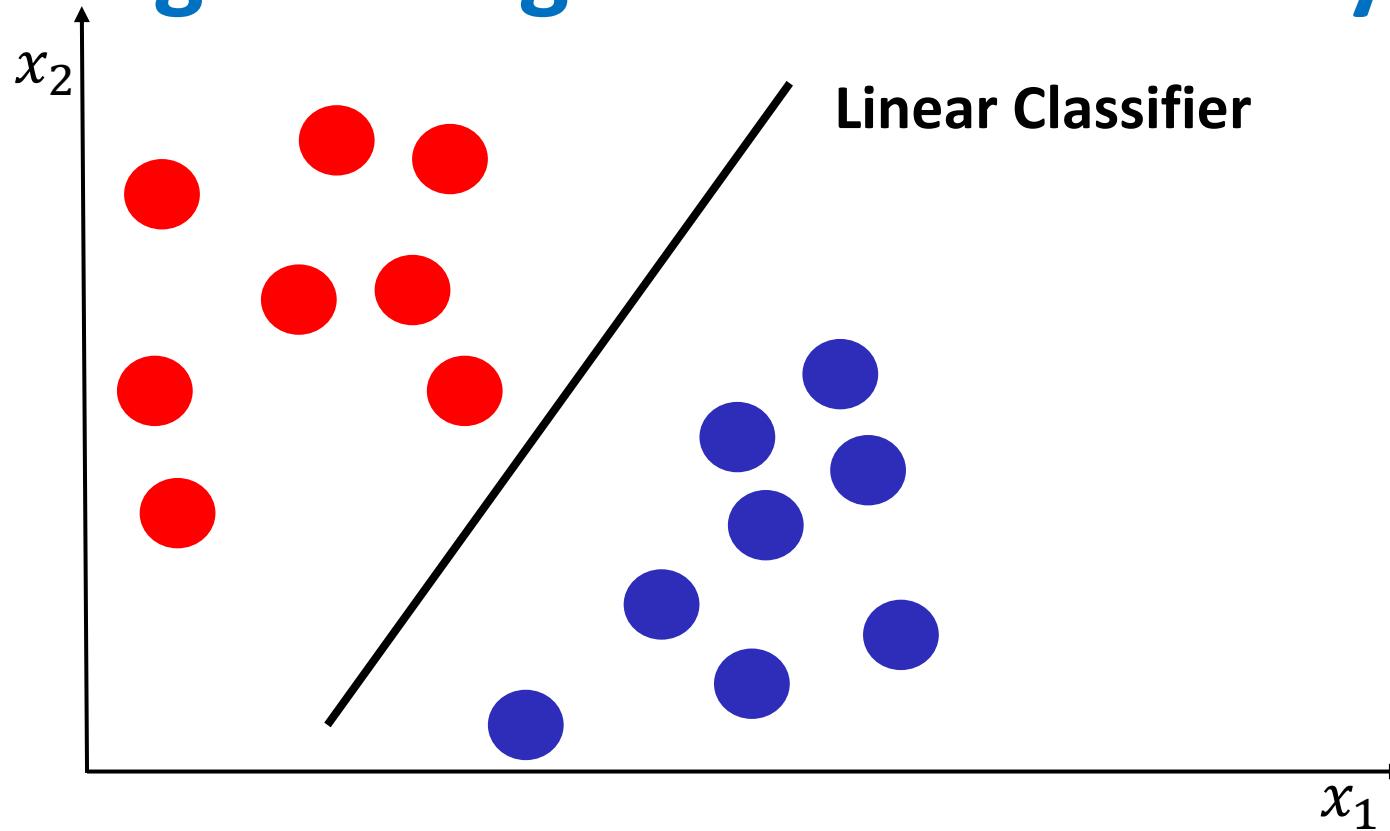
Summary

- Naturally forms complex decision boundaries
- Works well if we have a lot of samples
- Issues:
 - Complexity Scales linearly with the number of samples
 - High-dimensional data: to work well, we need a number of samples that grows exponentially with dimension

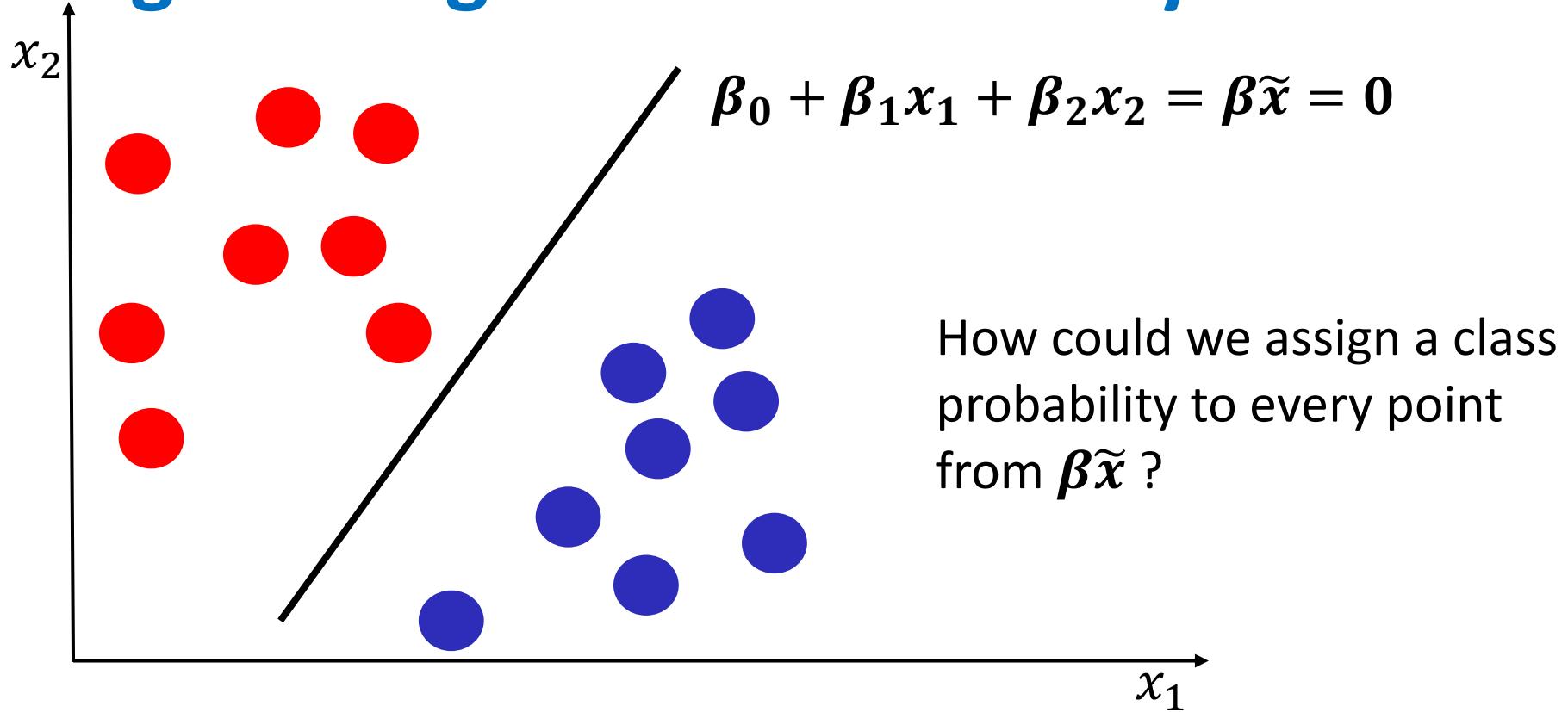
Agenda

- Traditional Classification Methods:
 - Decision Trees
 - Random Forest
 - K-Nearest Neighbor
 - **Logistic Regression**
 - Performance Metrics
 - Classification of Time Series
-

Logistic Regression as a binary classifier



Logistic Regression as a binary classifier



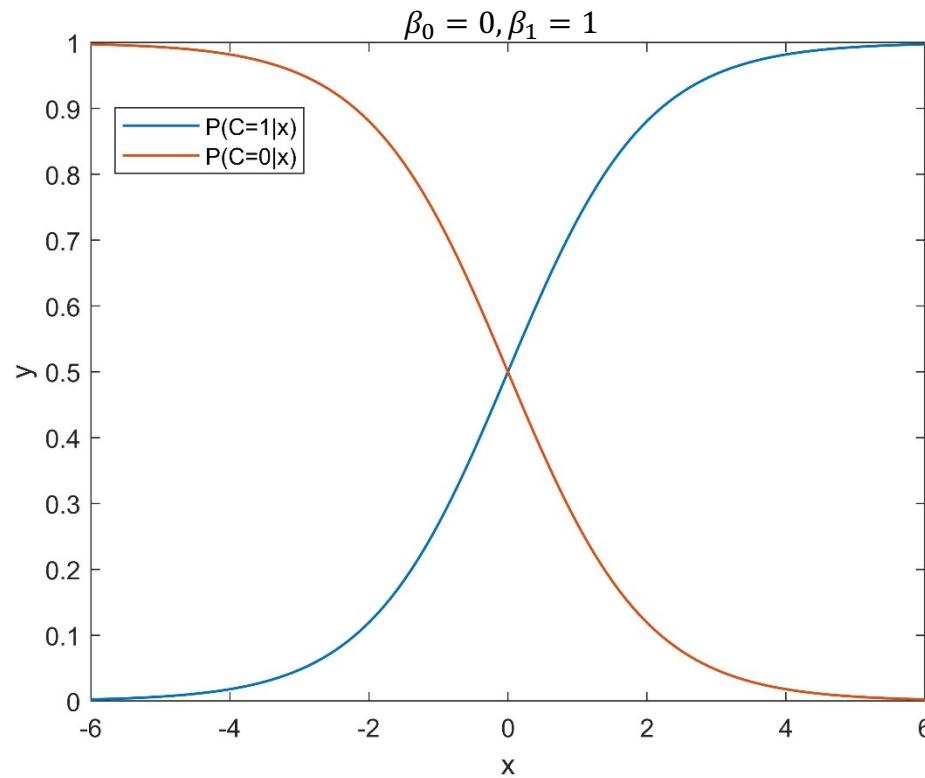
Logistic regression as a binary classifier

- We use a logistic function !!!

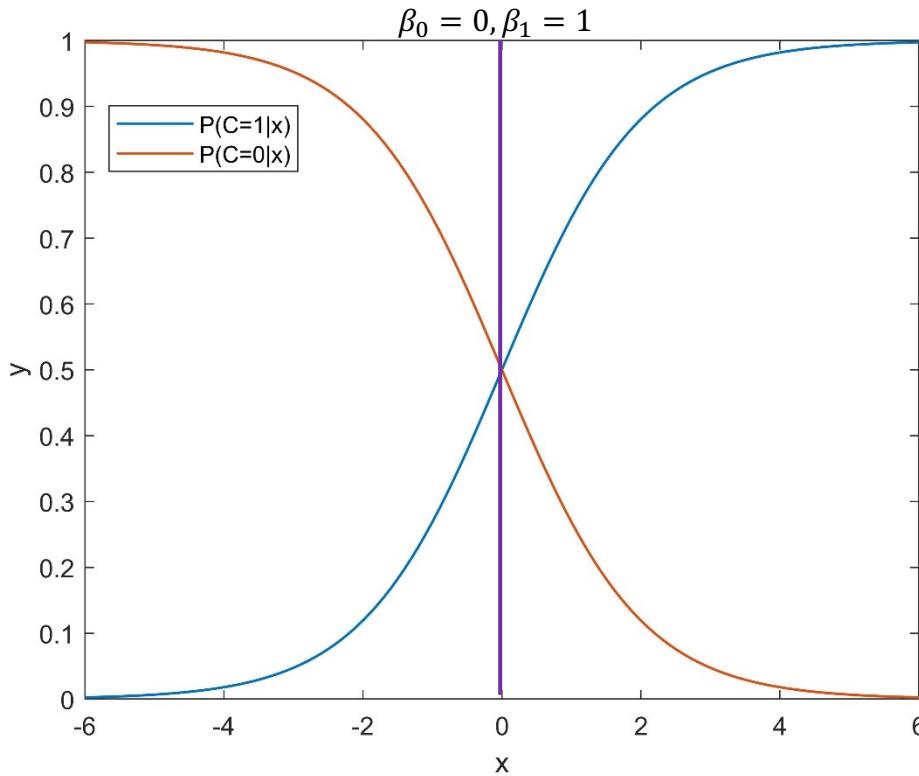
$$P(C = 1|x) = \frac{1}{1 + e^{-\beta \tilde{x}}} \quad \rightarrow \quad P(C = 0|x) = 1 - P(C = 1|x)$$

- Binary classification: We predict $C = 1$ if $P(C = 1|x) > 0.5$ and $C = 0$ otherwise.
- Why logistic function ?
 - Finding the parameters: convex optimization problem (easy to solve)
 - Smooth function

Example with 1 dimension



Example with 1 dimension



Set threshold to 0.5, i.e. predict
 $C = 1$ if $P(C = 1|x) > 0.5$

Logistic Regression as a binary classifier

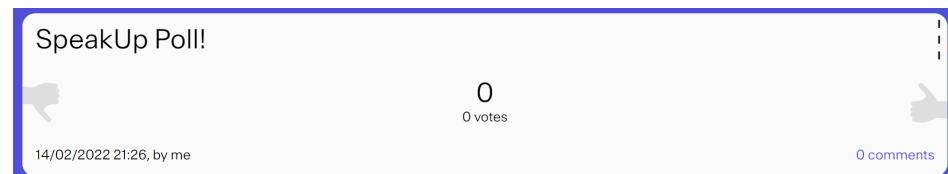
$$0.5\beta_1x_1 + 0.5\beta_2x_2 = 0$$

$$1\beta_1x_1 + 1\beta_2x_2 = 0$$

$$2\beta_1x_1 + 2\beta_2x_2 = 0$$

These 3 equations define the same boundary (same solutions).
If we use them for logistic regression and set the threshold to 0.5, will the classification be different ?

- 1) Yes
- 2) No



Logistic Regression as a binary classifier

$$0.5\beta_1x_1 + 0.5\beta_2x_2 = 0$$

$$1\beta_1x_1 + 1\beta_2x_2 = 0$$

$$2\beta_1x_1 + 2\beta_2x_2 = 0$$

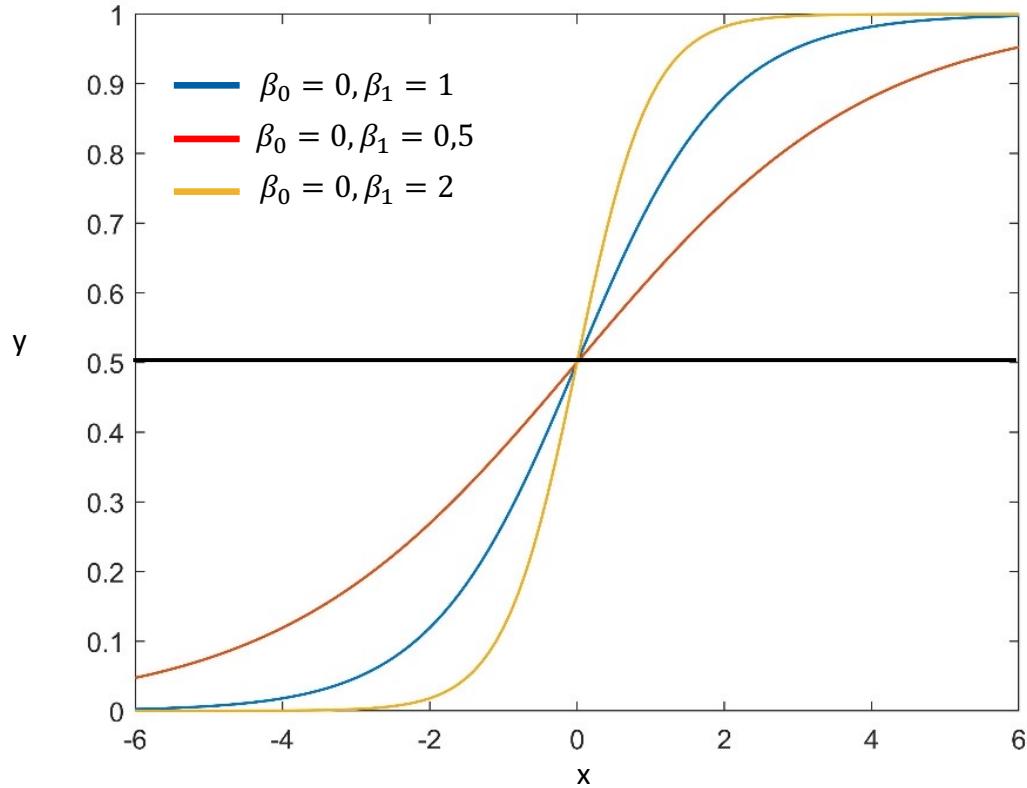
These 3 equations define the same boundary (same solutions).
If we use them for logistic regression and set the threshold to 0.5, will the resulting classifier be different ?

Answer: No. it is the same decision boundary.

But if we set the threshold to a different value than 0.5, it will make a difference.

Example with 1 dimension

$$y_n = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_n}}$$



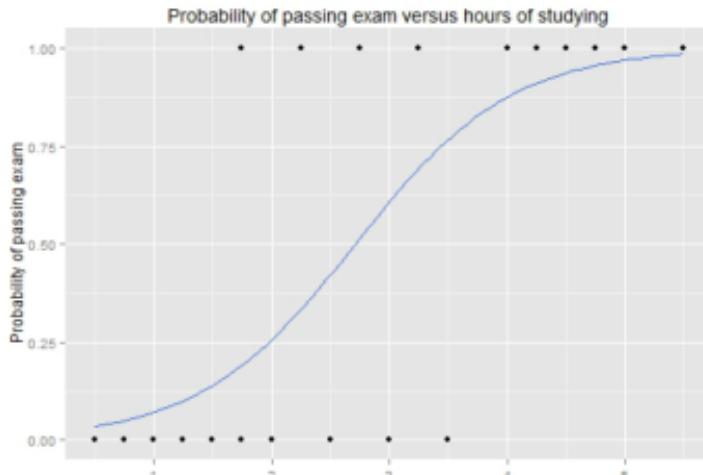
Logistic regression with multiple classes

- Multi-class classification: we do one regression per class:
 - $\beta_1 \tilde{x} = \beta_{1,0} + \beta_{1,1}x_1 + \beta_{1,2}x_2 + \cdots + \beta_{1,D}x_D$
 - $\beta_2 \tilde{x} = \beta_{2,0} + \beta_{2,1}x_1 + \beta_{2,2}x_2 + \cdots + \beta_{2,D}x_D$
 - ...
 - $\beta_K \tilde{x} = \beta_{K,0} + \beta_{K,1}x_1 + \beta_{K,2}x_2 + \cdots + \beta_{K,D}x_D$
- There exists multiple methods to compute the probability of each classes.
- One of them is the SoftMax function: $P(C = k|x) = \frac{e^{\beta_k \tilde{x}}}{\sum_{j=1}^K e^{\beta_j \tilde{x}}}$

Example: passing the exam

- **Problem:** given the number of hours the student spent learning, will (s)he pass the exam?

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



Hours of study	Probability of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Summary

- Advantages:
 - Natural probabilistic view of class predictions
 - Quick to train
 - Good accuracy for many simple data sets
 - Interpretability of model coefficients
- Disadvantages:
 - Linear decision boundary.

Traditional Methods

- Decision Trees and Random Forest
- K-Nearest Neighbor
- Logistic Regression

These are just examples of classification algorithms, there are others out there (e.g., Naïve Bayes, Support Vector Machines)

Agenda

- Traditional Methods:
 - Decision Trees and Random Forest
 - K-Nearest Neighbor
 - Logistic Regression
 - **Performance Metrics**
 - Classification of Time Series
-

Classification: Accuracy

$$Acc = \frac{|agreements|}{N}$$

where *agreements* means that the predicted outcome is equal to the observed outcome

- Easy to interpret
 - Works for binary and multi-class
 - General agreement across fields: accuracy is not a good standalone performance metric
-

Classification: Accuracy

$$Acc = \frac{|agreements|}{N}$$

where *agreements* means that the predicted outcome is equal to the observed outcome

- Easy to interpret
- Works for binary and multi-class
- General agreement across fields: accuracy is not a good standalone performance metric



Why?

Classification: Balanced accuracy

$$Acc_{bal} = \frac{1}{|C|} \cdot \sum_{c \in C} Acc_c$$

where $|C|$ denotes the number of classes

- Easy to interpret: average accuracy over all classes
- Takes into account class imbalance
- Works also for the multi-class case



Classification: Confusion Matrix

		True Outcome	
		Positive	Negative
Predicted Outcome	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

Confusion matrix for a binary classification task (two classes denoted as positive and negative class)

Classification: Confusion Matrix

		True Outcome		
		A	B	C
Predicted Outcome	A	15	7	5
	B	5	82	1
	C	1	6	13

Confusion matrix for a classification task with 3 classes: A, B, C

Classification: Confusion Matrix

		True Outcome		
		A	B	C
Predicted Outcome	A	21	0	0
	B	0	95	0
	C	0	0	19

Confusion matrix for a **Perfect** classifier

Classification: Specificity, sensitivity,...

		True Outcome	
		Positive	Negative
Predicted Outcome	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

$$\text{specificity} = \frac{tn}{tn + fp} \quad \text{sensitivity} = \frac{tp}{tp + fn}$$

Classification: Specificity, sensitivity,...

		True Outcome	
		Positive	Negative
Predicted Outcome	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

$$\text{specificity} = \frac{tn}{tn + fp} \quad \text{sensitivity} = \frac{tp}{tp + fn}$$

- **Sensitivity:** what is the proportion of **positive** I identified correctly ?
 - High Sensitivity: Most of the positive were identified
- **Specificity:** what is the proportion of **negative** I identified correctly ?
 - High Specificity: Most of the negative were identified

Classification: Specificity, sensitivity,...

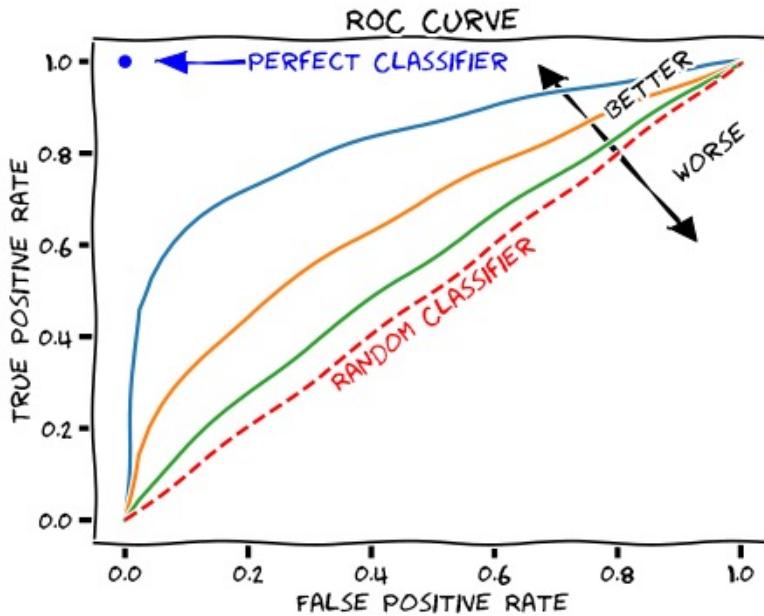
		True Outcome	
		Positive	Negative
Predicted Outcome	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

$$\text{specificity} = \frac{tn}{tn + fp} \quad \text{sensitivity} = \frac{tp}{tp + fn}$$

- These metrics are used in addition to accuracy
- Sensitivity and specificity are often used in the fields of Psychology, Learning Sciences, etc.

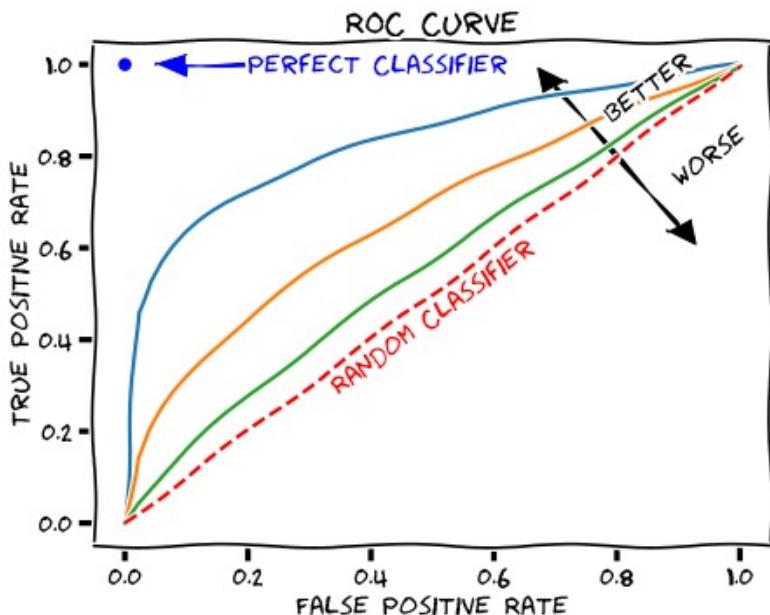
ROC curve

$$TPR = \frac{tp}{P}$$



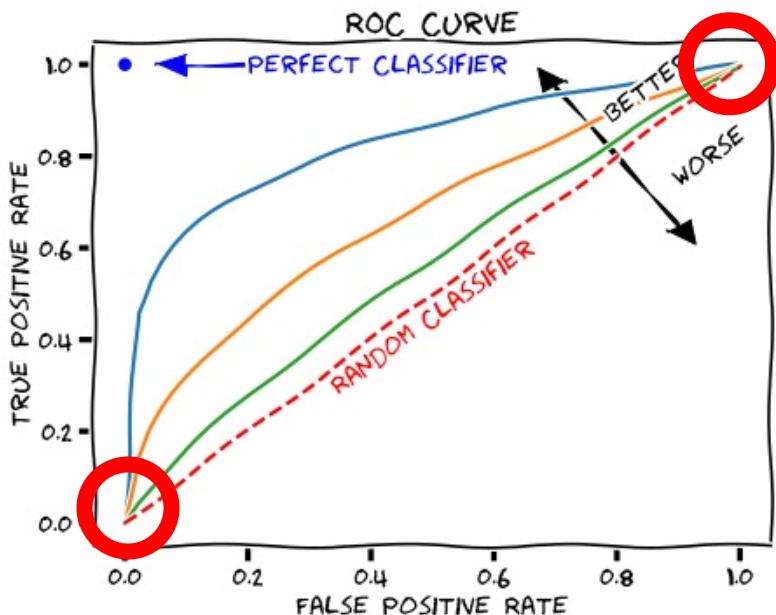
$$FPR = \frac{fp}{N}$$

Computing the ROC curve



- Given a classifier $f(x)$ predicting some type of score (e.g., a probability)
- Choose a threshold t :
 - $f(x_i) \geq t$: predict positive class
 - $f(x_i) < t$: predict negative class
- Compute TPR and FPR given t
- Repeat for different thresholds (e.g., in case of probabilities choose $t = 0, 0.1, \dots, 1$)

Computing the ROC curve



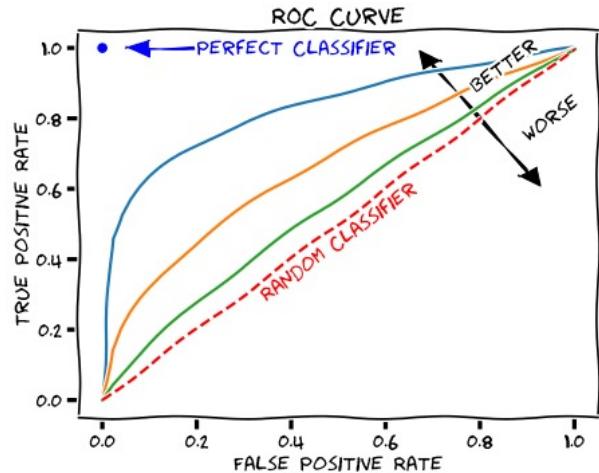
If we set $t = 1$, to which part of the graph will it corresponds to?

1. Bottom left ($\text{TPR}=\text{FPT}=0$)
2. Top Right ($\text{TPR}=\text{FPT}=1$)



Area under the ROC curve (AUC)

- The AUC denotes the area under the ROC curve
- A perfect classifier has an AUC of 1
- A random classifier has an AUC of 0.5
- Often used as a performance metric in more technical fields (e.g., educational data mining)
- The AUC can be extended to the multi-class case by considering all possible pairs of classes



Classification: Summary

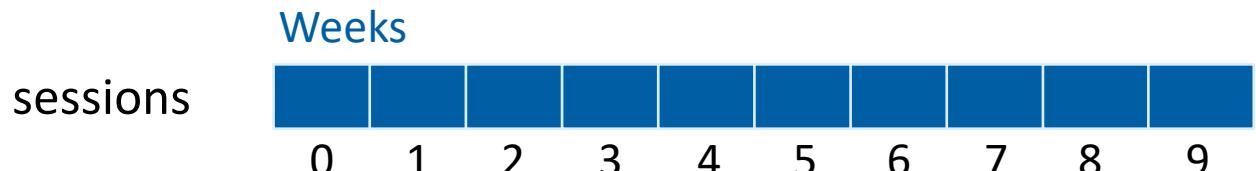
- Do:
 - Carefully think about the choice of metric (or combination)
 - Some ideas:
 - Use accuracy plus sensitivity and specificity [binary]
 - Use (balanced) accuracy plus AUC [binary + multi-class]
 - Use just AUC [binary + multi-class]
 - Don’t:
 - Use accuracy as a standalone metric
 - Compute “all” possible metrics that come to your mind
-

Agenda

- Traditional Methods:
 - Decision Trees and Random Forest
 - K-Nearest Neighbor
 - Logistic Regression
 - Performance Metrics
 - **Classification of Time Series**
-

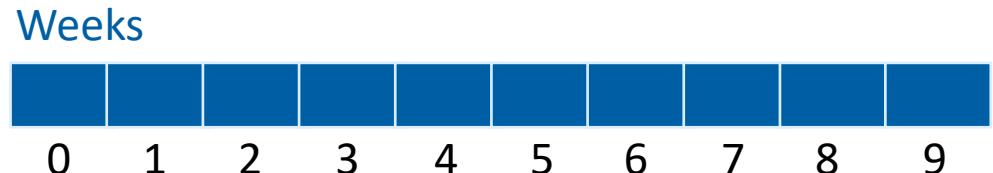
Time Series – Our flipped classroom case

Student i



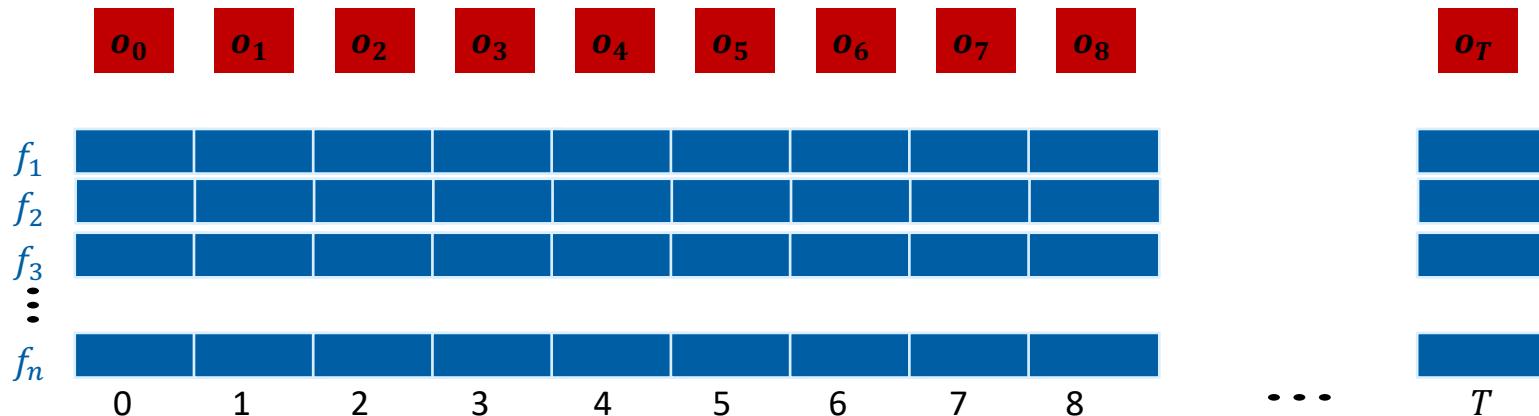
•
•
•

submissions_correct



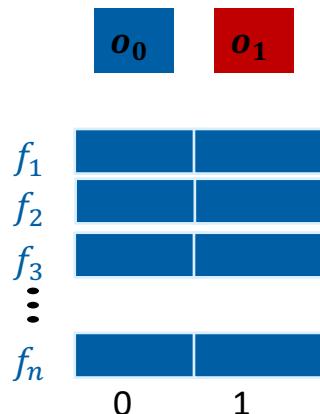
Time Series – Tracing Task

- Prediction of a **categorical** target variable after $t < T$ time steps, where T is the total number of time steps
- Prediction of a variable in time step $t + 1$, based on time steps $0, \dots, t$.



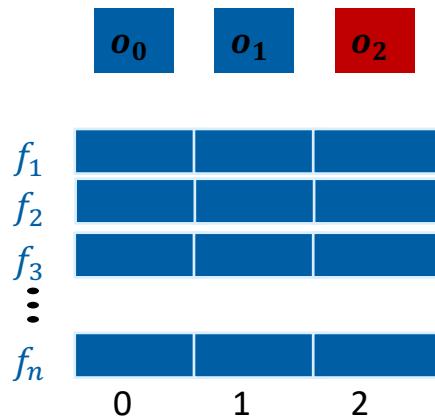
Time Series – Tracing Task

- Prediction of a **categorical** target variable after $t < T$ time steps, where T is the total number of time steps
- Prediction of a variable in time step $t + 1$, based on time steps $0, \dots, t$.



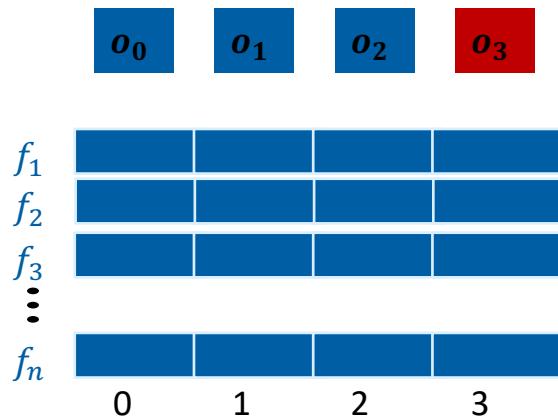
Time Series – Tracing Task

- Prediction of a **categorical** target variable after $t < T$ time steps, where T is the total number of time steps
- Prediction of a variable in time step $t + 1$, based on time steps $0, \dots, t$.



Time Series – Tracing Task

- Prediction of a **categorical** target variable after $t < T$ time steps, where T is the total number of time steps
- Prediction of a variable in time step $t + 1$, based on time steps $0, \dots, t$.



Time Series – Tracing Task

Last Week:

- we have solved this task for a *numerical* target variable (students' performance in weekly quizzes) using a linear mixed effect model

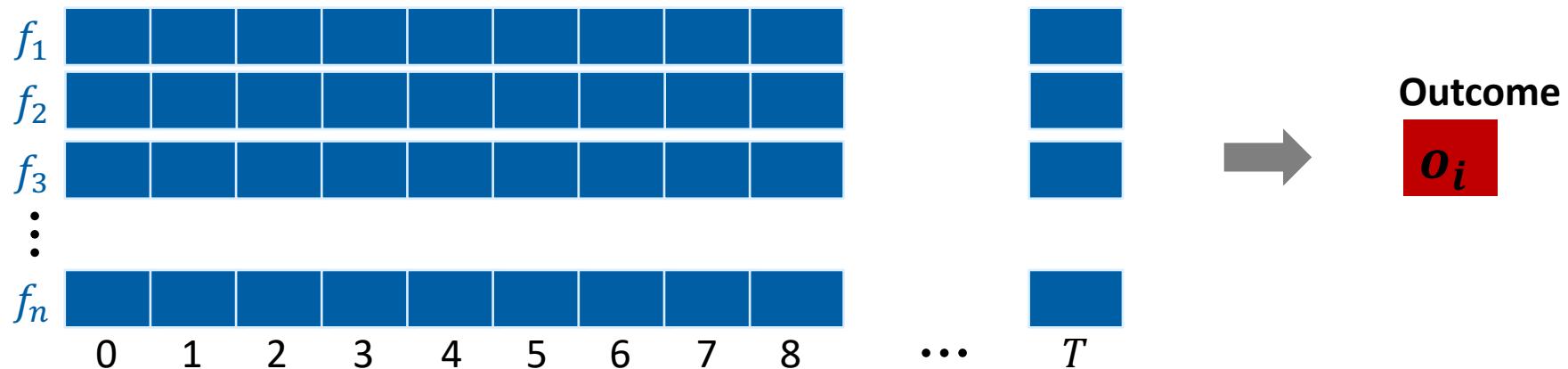
This Week:

- for a *binary* target variable, this task can be solved using a logistic (mixed effect) model
- The other presented algorithms are not suitable for this task

Time Series – Prediction Task

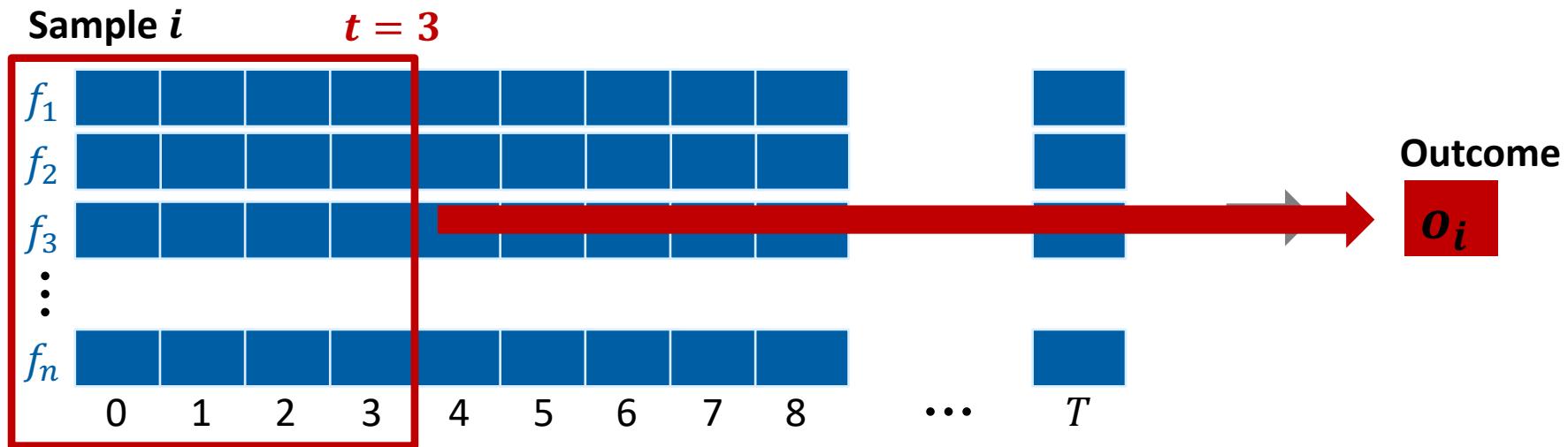
- Prediction of a **binary** target variable after $t < T$ time steps, where T is the total number of time steps

Sample i



Time Series – Prediction Task

- Prediction of a **binary** target variable after $t < T$ time steps, where T is the total number of time steps



Your Turn

- Predict whether students will pass the course after $t = 5$ weeks (i.e. after half of the course)
 - We provide you the train-test split to use in the Jupyter Notebook
 - You can choose the classifier: Decision Tree, Random Forest, or k-Nearest Neighbor
-

Your Turn – Some Hints

- For Decision Tree and Random Forest you will need to use one of the following:
 - Flattening
 - Aggregation (hint: we have aggregated features last week)
 - For k-Nearest Neighbor, you can compute pairwise distances between vectors
 - If you have several features, compute a pairwise distance matrix separately for each feature and then sum the distance matrices up
 - Distance matrices can have different scales (hint: MinMaxScaler from *sklearn*)
-

Your Turn – Feedback

Do you want feedback or have questions?

Upload your Jupyter Notebook here:

<https://go.epfl.ch/mlbd-activities>

Model Evaluation

Machine Learning for Behavioral Data

March 20, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Time Series Prediction
7	Time Series Prediction
8	Spring Break

Complete pipeline for one use case:

- Data exploration
- Prediction
- Model evaluation

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

[Classification] Which of the following is an example of an ensemble method?



- a) Decision Tree
- b) k-Nearest Neighbor
- c) Logistic Regression
- d) Random Forest

Short quiz about the past...

[Classification] We are going to predict the species of an Iris Flower. The dependent variable (species) contains three possible values: Setosa, Versicolor, and Virginica. Which classification methods are appropriate for this task?

SpeakUp Poll!

0
0 votes

14/02/2022 21:26, by me

0 comments

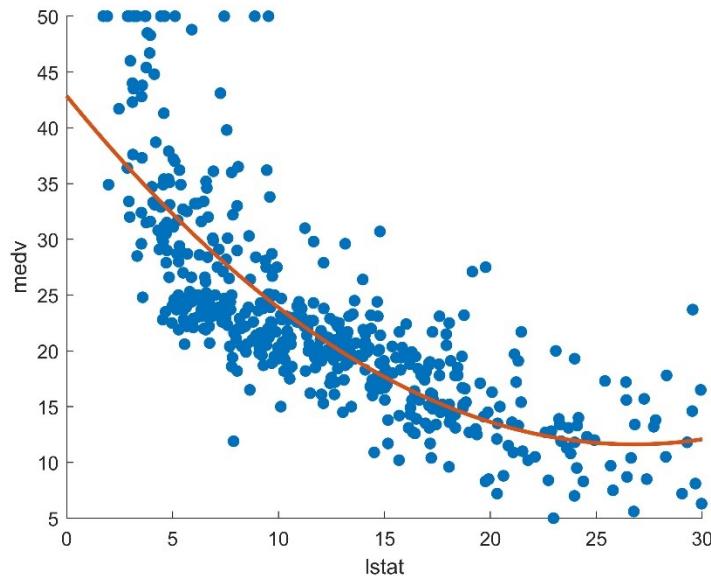
- a) Decision Tree
- b) k-Nearest Neighbor
- c) Logistic Regression
- d) Random Forest

Today

- **Model Assessment and Selection**
- Reporting of Results

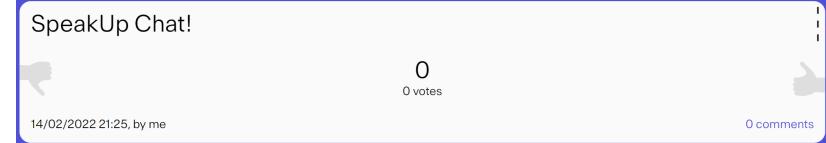


How good is my model?



$$medv = 42.86 - 2.33 \cdot lstat + 0.04 \cdot lstat^2$$

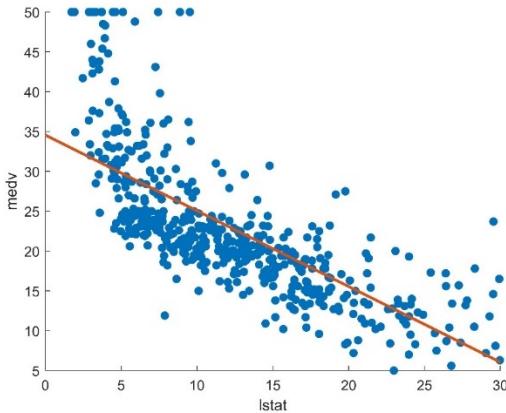
$$R^2 = 0.64$$



Istat: Percentage of lower status of the population

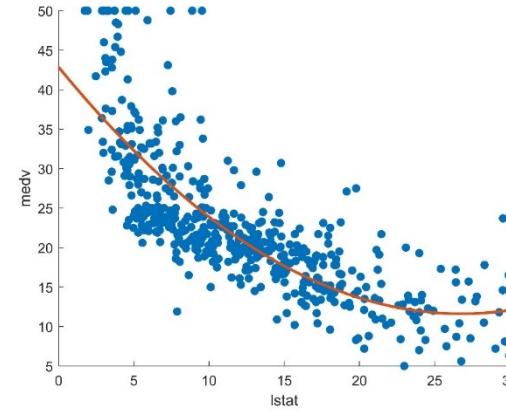
medv: Median value of owner-occupied homes in \$1000s

Which model is better?



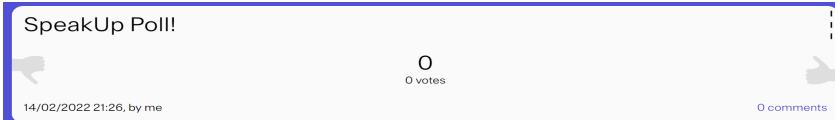
$$medv = 34.55 - 0.95 \cdot lstat$$

$$R^2 = 0.54$$



$$medv = 42.86 - 2.33 \cdot lstat + 0.04 \cdot lstat^2$$

$$R^2 = 0.64$$



- a) The left model
- b) The right model
- c) I need more information

Model Evaluation

- **Model assessment:** having chosen a final model, estimating its prediction error on new data (*generalization*).



Model Evaluation

- **Model assessment:** having chosen a final model, estimating its prediction error on new data (*generalization*).
 - **Model selection:** estimating the performance of different models in order to choose the best one.
-

Theory: Notation

- We are given a sample data set:

$$T = \{(y_n, \mathbf{x}_n)\} \text{ with } n = 1, \dots N$$

- The sample data set T has been drawn from an (**unknown**) underlying data model D with range $X \times Y$:

$$(y_n, \mathbf{x}_n) \text{ i.i.d.} \sim D$$

- We have learnt a model f for predicting y based on \mathbf{x} . **How good is f ?**
-

Theory: Expected Loss

$$Err_D(f) = \mathbb{E}_D[L(y, f(x))]$$

where y and x are randomly drawn from D , i.e. $Err_D(f)$ is the *expected* error of f over all samples chosen according to D .
 $Err_D(f)$ is denoted as *expected/true risk/loss*.



Theory: Expected Loss

$$Err_D(f) = \mathbb{E}_D[L(y, f(x))]$$

where y and x are randomly drawn from D , i.e. $Err_D(f)$ is the *expected* error of f over all samples chosen according to D .
 $Err_D(f)$ is denoted as *expected/true risk/loss*.

- We cannot compute $Err_D(f)$, since we don't know D .
- We are, however, given a sample data set T , drawn from D .
We can use T to approximate the expected loss.

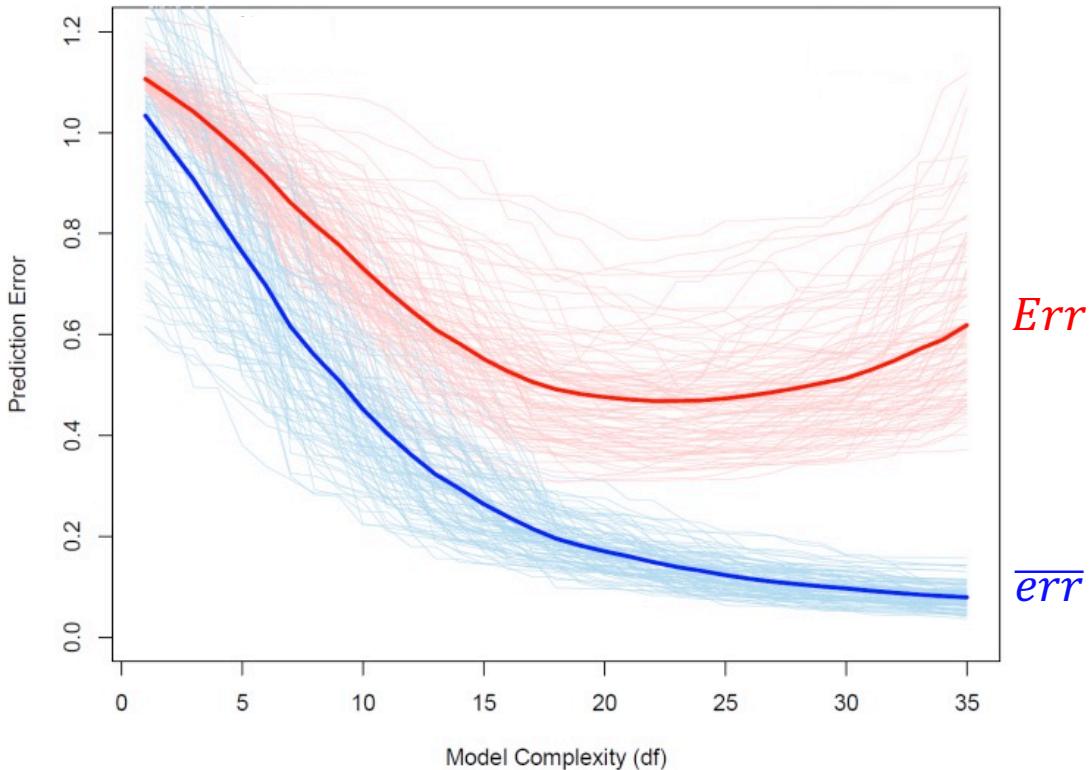
Theory: Training Error

- First idea: we compute the error of f on T to *empirically* approximate $Err_D(f)$:

$$\overline{err}(f) = \frac{1}{|T|} \sum_{(x_n, y_n) \in T} L(y_n, f(x_n))$$

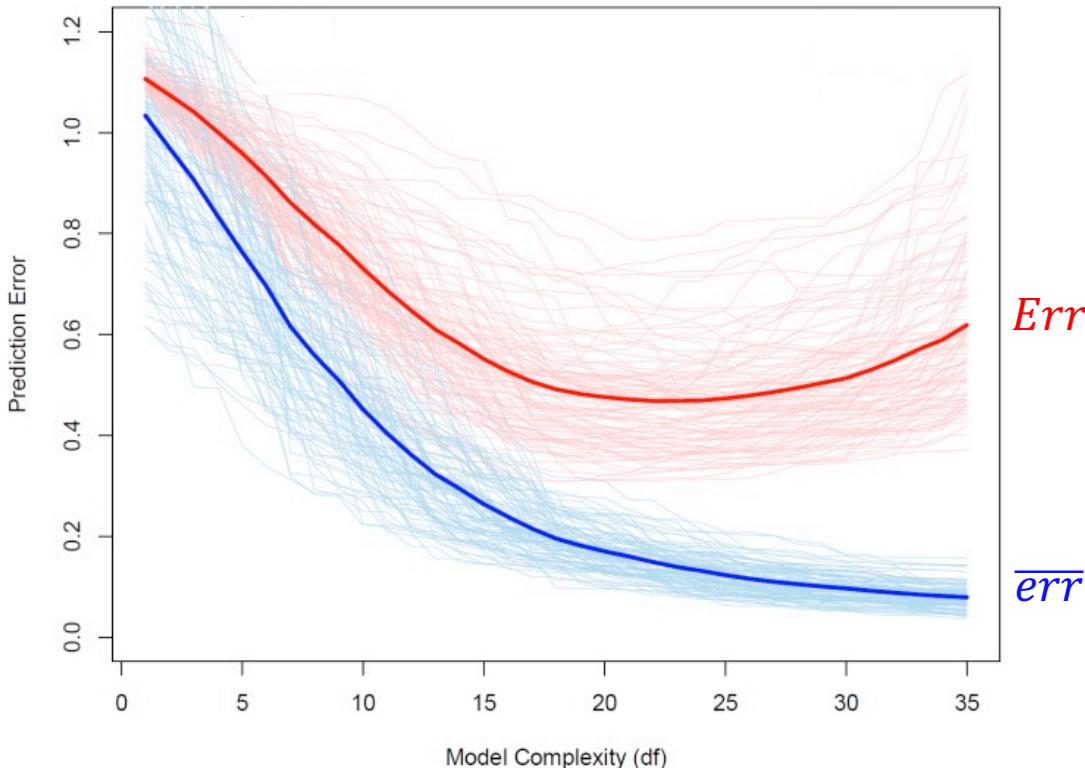
- We call $\overline{err}(f)$ the *empirical error* (or simply the *training error*).
-

Training error underestimates expected loss



- Using synthetic data, we can compute the expected loss (we **know** D).

Training error underestimates expected loss



- Using synthetic data, we can compute the expected loss (we **know** D).
- The training error underestimates the expected loss



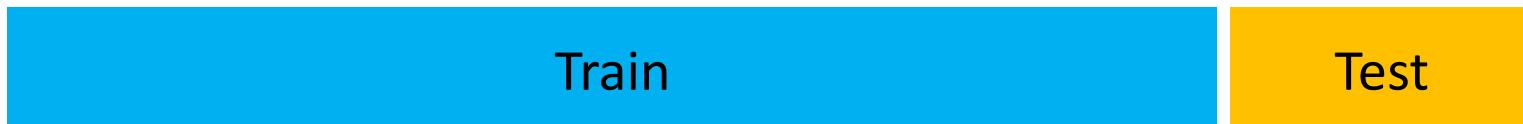
We need a better solution to estimate Err_D

Model Assessment & Selection

- ① Train, Validate, Test
 - ② Resampling Methods
 - ③ Information Criteria
-

Fixed assignment

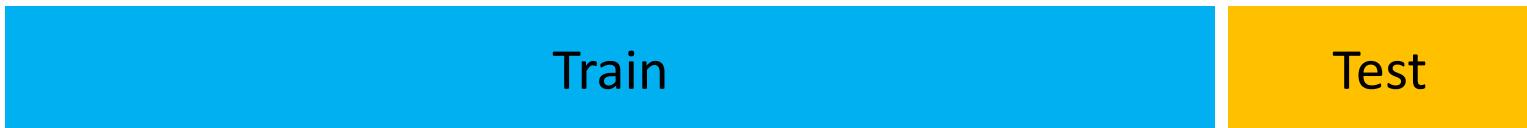
- We randomly split the data into a training data set and a test data set (e.g., 80/20 split or 70/30 split)



- *Training set*: used for fitting the model
- *Test set*: assessment of the generalization error of the model

Test Error

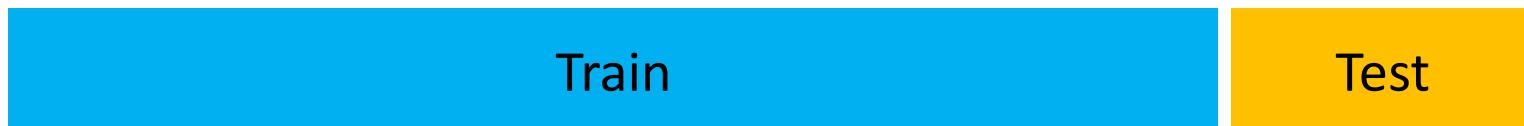
- We randomly split the data set T into a training data set and a test data set (e.g., 80/20 split or 70/30 split)



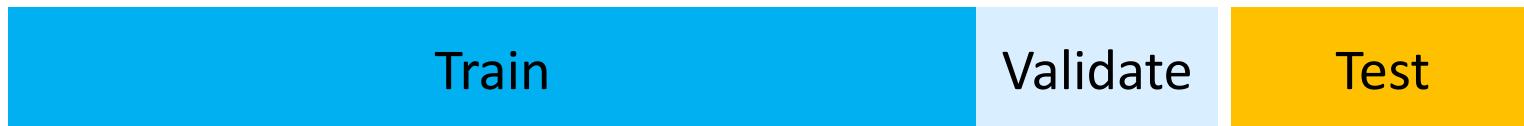
$$Err_{test}(f_{train}) = \frac{1}{|T_{test}|} \sum_{(x_n, y_n) \in T_{test}} L(y_n, f_{train}(x_n))$$

Fixed assignment: Selection & Generalization

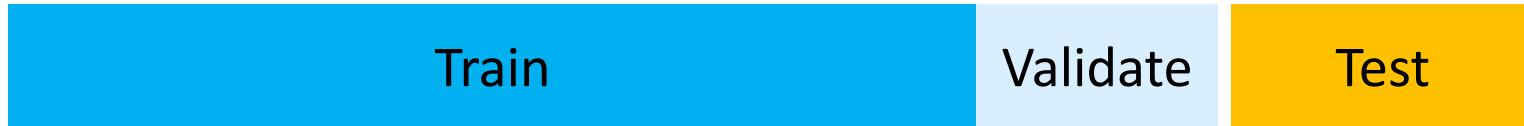
- We randomly split the data set T into a training data set and a test data set (e.g., 80/20 split or 70/30 split)



- We further (randomly) split the training data set T_{train} and reserve a part of it for validation (e.g., 80/20 split)

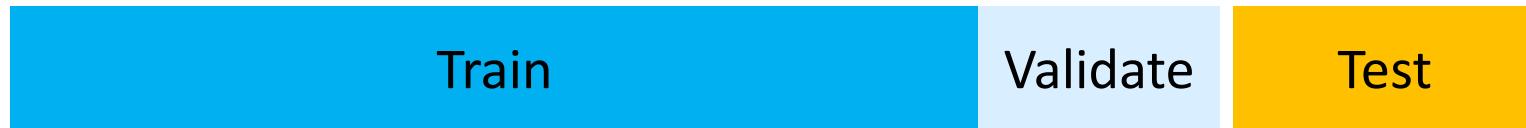


Fixed assignment: Selection & Generalization



- *Training set*: used for fitting the models
- *Validation set*: used to estimate the prediction error for model selection (e.g., hyperparameter tuning)
- *Test set*: assessment of the generalization error of the chosen model

Inefficient use of data



- Not very efficient use of data (requires a large amount of data)
- How large? Depends on
 - Signal to noise ratio of data set
 - Complexity of the models we want to fit

Assessment & Selection

- ① Train, Validate, Test
- ② Resampling Methods
 - Cross Validation
 - Bootstrapping
- ③ Information Criteria



K-Fold Cross-Validation

- Randomly divide data into K parts (folds)
- For $k = 1, \dots, K$
 - Fit the model to the other $K - 1$ folds $\{1, \dots, K\} \setminus \{k\}$
 - Compute the prediction error on k
- Combine the K estimates of prediction error

$K = 5$ 

$k = 1$	Test	Train	Train	Train	Train
$k = 2$	Train	Test	Train	Train	Train
$k = 3$	Train	Train	Test	Train	Train
$k = 4$	Train	Train	Train	Test	Train
$k = 5$	Train	Train	Train	Train	Test

Combining the K estimates of prediction error

$K = 5 \rightarrow$

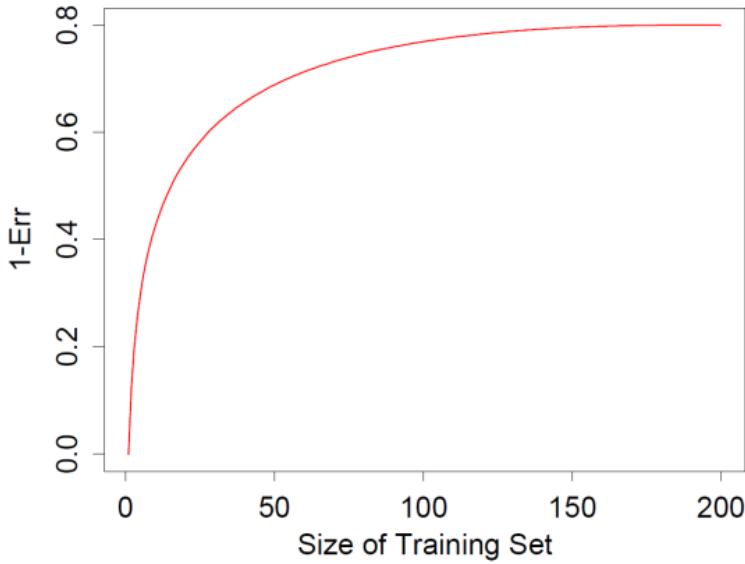
k = 1	Test	Train	Train	Train	Train
k = 2	Train	Test	Train	Train	Train
k = 3	Train	Train	Test	Train	Train
k = 4	Train	Train	Train	Test	Train
k = 5	Train	Train	Train	Train	Test

$$PE^{CV} = \frac{1}{K} \sum_{k \leq K} PE_k = \sum_{i=1}^N L(y_i, f^{-\kappa(i)}(x_i))$$

Leave-one-out cross validation

- $K = N$, i.e. each fold just contains one sample
 - For a sample i , we train f on all samples but i and then use f to predict on i
 - Can have high variance (the N training sets are very similar to each other)
 - High computational burden (f needs to be fit N times)
 - Avoids issues with fold selection
-

How do we choose K?



- We don't know the learning curve of our model
- In practice: 5-fold and 10-fold cross validation are recommended as a good compromise

Variants of cross validation

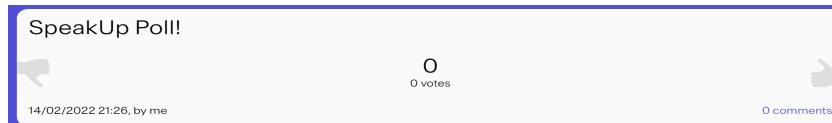
- Flat
 - Each sample has equal chance of being placed into each fold
 - Stratified
 - Fold selection is biased such that some variable is equally represented in each fold
 - This can be the variable we are trying to predict (y)
 - This can be a variable that is thought to be an important context
-

Stratification

- Other levels of stratification are possible (e.g., school, learning environment, demographic information)
 - Consideration:
 - Where will the model be used (e.g., what is the potential application)?
 - Make sure you stratify at this level
 - Can also be combined (e.g., we can stratify by predicted variable y and by user)
 - Stratification can also be used in the fixed assignment (train – validate – test) setting
-

Example: Classification Problem

- Given: classification problem, synthetic data set with
 - binary** class label y (balanced)
 - 5000 features (*independent* of class labels y)
- What is the expected loss of any classifier f on this data set?



- a) 10%
- b) 30%
- c) 50%
- d) 70%
- e) 90%

Example: Suggested Procedure

- 1) Select the 100 features that have the highest correlation with the class labels y
 - 2) Use a 1-nearest neighbor classifier based on just these 100 selected features
 - 3) Use cross validation to estimate the prediction error of the classifier
- Over 50 simulations of this procedure: $PE^{CV} = 0.03$

Be careful!

- Multistep modeling procedure
 - Cross-validation must be applied to the **entire sequence** of modeling steps. In particular, **samples must be “left out” before any selection or filtering steps** are applied.
 - Exception: **unsupervised** cleaning or screening steps can be done before samples are “left out”.
 - This of course holds also for the train-validate-test case!
-

Your Turn – Model Assessment

- In your student notebook, we provide examples on Train-Test Split and Cross Validation model evaluation.
- Your task:
 - Run the Train-Test Split and the Cross Validation model evaluations
 - What is the difference in accuracy/AUC between the two methods?
Where does this difference come from?

Assessment & Selection

- ① Train, Validate, Test
- ② Resampling Methods
 - Cross Validation
 - Bootstrapping
- ③ Information Criteria

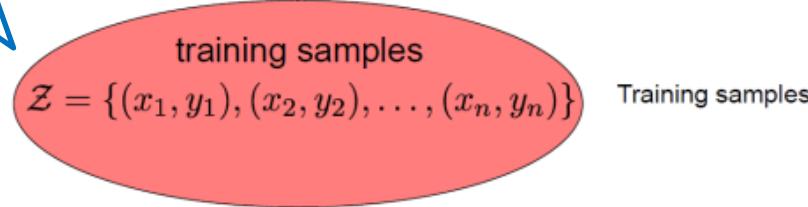


Bootstrapping

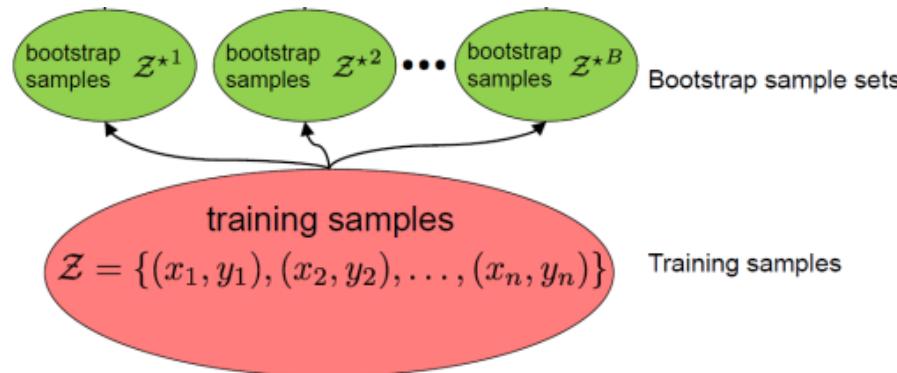
- General tool for assessing statistical accuracy
 - Basic idea:
 - Given: training data set $\mathbf{Z} = \{z_1, \dots, z_N\}$ with $z_i = (\mathbf{x}_i, y_i)$, where N is the number of samples
 - Randomly draw N pairs (\mathbf{x}_i, y_i) with replacement from the training data set
 - Repeat B times (e.g., $B = 100$) -> B bootstrap data sets
 - Fit model to each bootstrap data set, observe behavior across bootstrap data sets
-

Bootstrapping

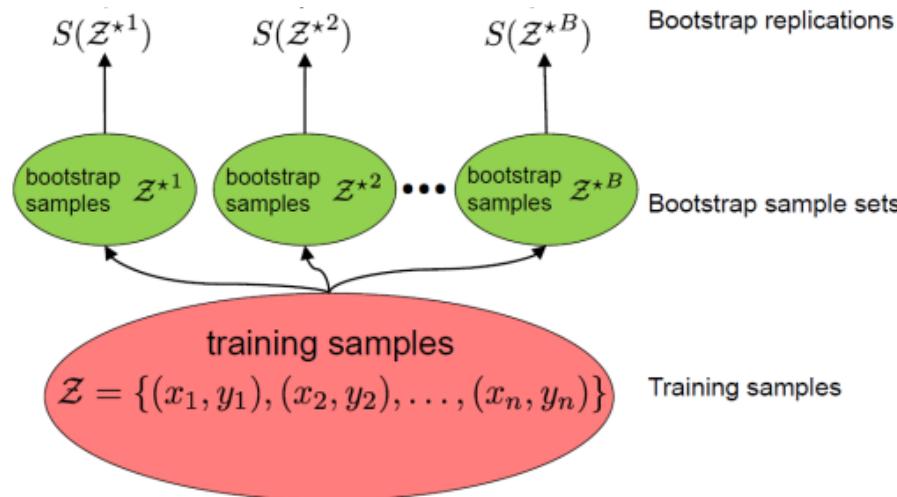
- We are interested in $S(Z)$
- $S(Z)$ can be any quantity computed from the data Z , e.g., the prediction at some input point
- We want to compute an aspect of the distribution of $S(Z)$ (e.g., the variance)



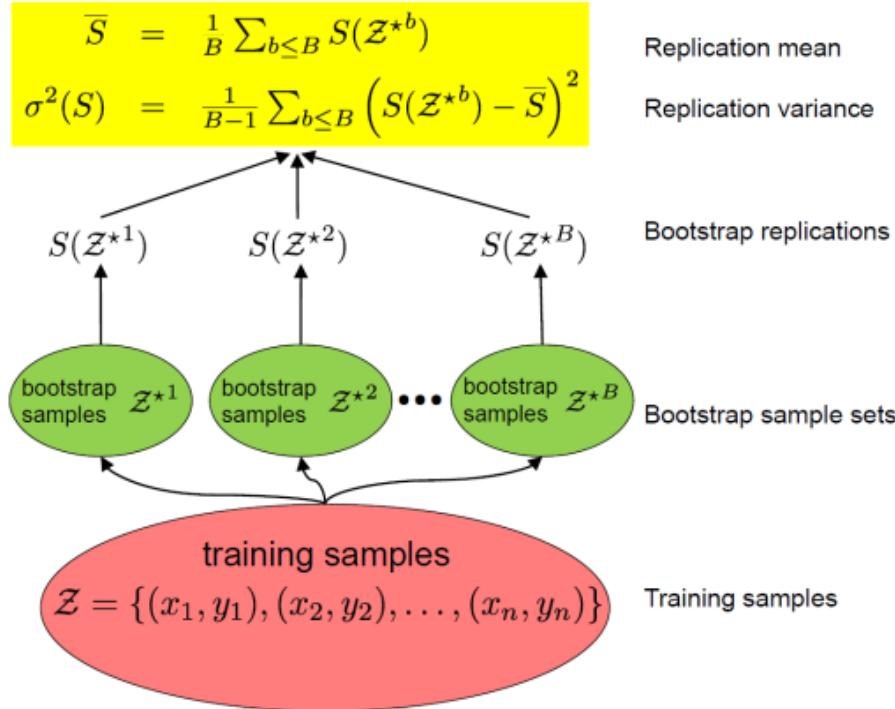
Bootstrapping



Bootstrapping



Bootstrapping



Bootstrapping for prediction

- Idea: leave-one-out bootstrap

$$\widehat{Err}^{(1)} = \frac{1}{N} \sum_{i=1}^N \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, f^{*b}(x_i))$$

where C^{-i} denotes the set of indices of bootstrap samples b , that do **not** contain observation i

.632 bootstrap error

- It can be shown that the average number of distinct samples in each bootstrap sample b is about $0.632 \cdot N$
- $\widehat{Err}^{(1)}$ will therefore tend to overestimate the true loss
- Solution:

$$\widehat{Err}^{(.632)} = .368 \cdot \overline{err} + .632 \cdot \widehat{Err}^{(1)}$$

Assessment & Selection

- ① Train, Validate, Test
- ② Resampling Methods
- ③ **Information Criteria**
 - AIC
 - BIC



Information Criteria

- The training error is an overly optimistic estimate of the *expected loss*
- Information criteria try to estimate the *optimism* in the training error and correct for it

Akaike Information Criterion (AIC)

$$AIC = -\frac{2}{N} \cdot \loglik + 2 \cdot \frac{d}{N}$$

- d is the number of parameters of our model f
- \loglik is the log-likelihood (logarithm of the likelihood) of the sample data T given our model f
- N is the number of samples in the data set, i.e. $|T| = N$

Bayesian Information Criterion (BIC)

$$BIC = -2 \cdot \text{loglik} + \log(N) \cdot d$$

- d is the number of parameters of our model f
- loglik is the log-likelihood (logarithm of the likelihood) of the sample data T given our model f
- N is the number of samples in the data set, i.e. $|T| = N$
- BIC penalizes complex models more heavily than AIC

Information Criteria - Considerations

- Applicable for models where the fitting is done under a maximum likelihood setting
 - *Effective* number of parameters:
 - Choosing the best fitting model with d features, can result into more than d parameters being fit (e.g., regularization)
 - Determining the *effective* number of parameters can be difficult for more complex models (e.g., trees)
-

What method should we choose?

- ① Train, Validate, Test
- ② Resampling Methods
 - Cross Validation
 - Bootstrapping
- ③ Information Criteria
 - AIC
 - BIC

What method should we choose?

Depends on the purpose of the model:

- Interpretation: information criteria are useful for unsupervised cases or when the model is built for interpretation purposes (e.g., which features of the regression model explain the data better?)
 - Prediction:
 - All presented methods (train-test, cross validation, bootstrap) are reasonable choices
 - We can combine them freely to do model selection and assessment
 - In practice: while bootstrap provides accurate estimates of test error, it also requires a high amount of extra work
-

Example Combinations



Your Turn – Model Selection & Assessment

- In your student notebook, we provide an incorrect example to tune to hyperparameters of the model and then evaluate the prediction error of that model in terms of accuracy or AUC
- Your task:
 - Explain why it is incorrect.
 - Describe how it could be fixed.

Today

- Model Selection and Assessment
- **Reporting of Results**



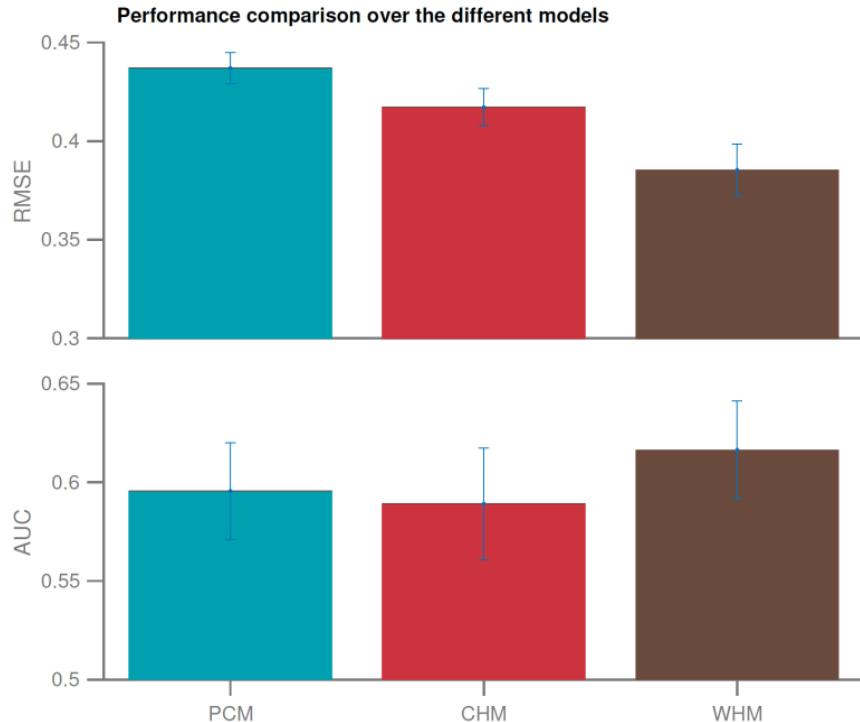
Back to the question: is my model any good?

- A friend tells you that he has found a great model for early prediction of drop out in online courses
- She gives you the following information:
 - The data set stems from N students taking course c
 - 30% of the students dropped the course
 - She has evaluated her model using cross validation
 - The accuracy of the model is 0.8 and the AUC is 0.83

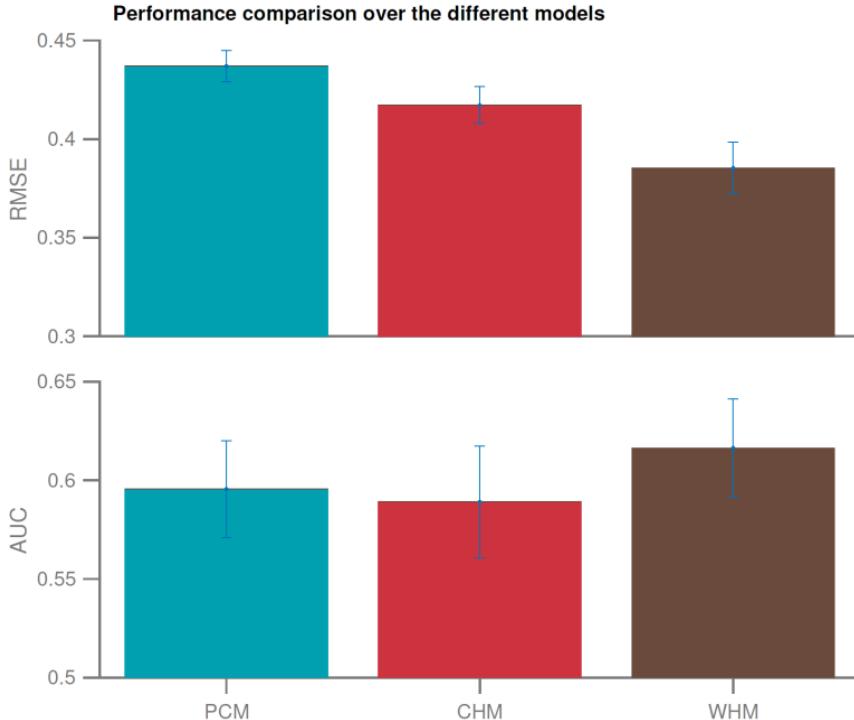
Provide comparisons!

- Reporting performance metrics for a model f on a data set T is not enough: achievable performance (what is good?) depends on the data set!
 - We need to provide comparisons to **baseline** models:
 - **Minimum** baseline: compare to a random model
 - Additional comparisons, depending on the goal:
 - Suggestion of a new type of model structure for a problem: need to compare to previously suggested structures
 - Suggestion of new features for a problem: need to compare performance between a model using and not using the new features
-

Quantification of Uncertainty



Quantification of Uncertainty



- We usually report the *mean* prediction error over all samples
- We should quantify the uncertainty of the performance metric!
- Uncertainty can be computed across samples where applicable (or across folds)
- Error bars can denote
 - Standard deviation
 - Standard error ($\sqrt{\sigma^2/N}$)

Today

- Model Assessment and Evaluation
- Reporting of Results



Summary: Pipeline

Design/choose an appropriate learning algorithm and features

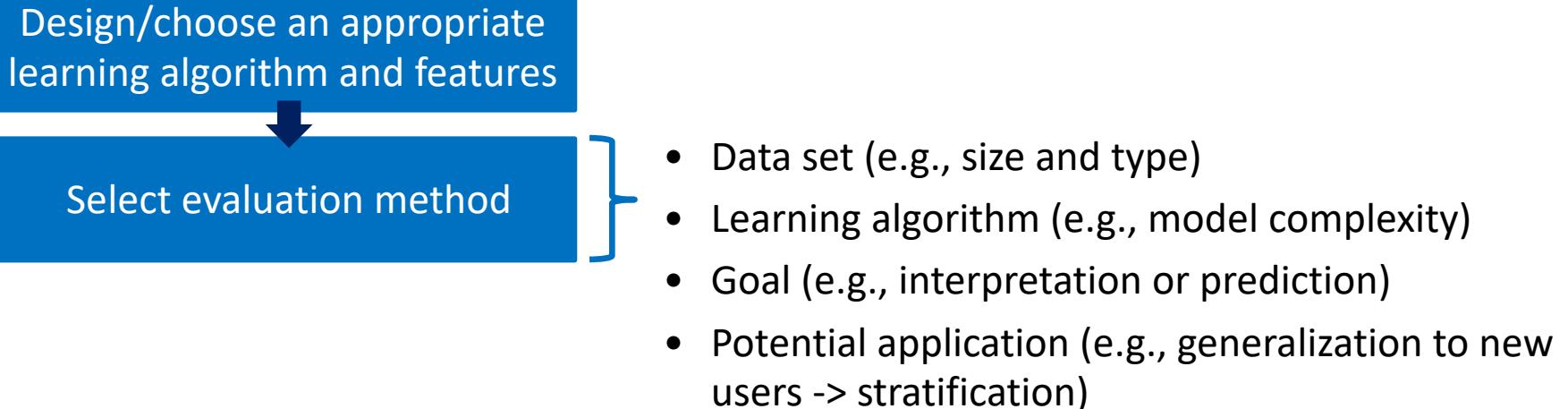


- Problem (e.g., classification or regression)
- Data set (e.g., size, type)
- Task (e.g., dropout prediction)

Summary: Pipeline

Design/choose an appropriate learning algorithm and features

Select evaluation method

- 
- Data set (e.g., size and type)
 - Learning algorithm (e.g., model complexity)
 - Goal (e.g., interpretation or prediction)
 - Potential application (e.g., generalization to new users -> stratification)

Summary: Pipeline

Design/choose an appropriate learning algorithm and features



Select evaluation method



Choose appropriate performance metrics



- Problem (e.g., classification or regression)
- Learning algorithm (e.g., model type)
- Potential application (i.e., what is important?)

Summary: Pipeline

Design/choose an appropriate learning algorithm and features



Select evaluation method



Choose appropriate performance metrics



Select baseline approaches for comparison



What do I want to show/prove, i.e. what is my claim (e.g., suggestion of new neural network architecture, suggestion of new features for a specific problem)?

Summary: Pipeline

Design/choose an appropriate learning algorithm and features



Select evaluation method



Choose appropriate performance metrics



Select baseline approaches for comparison



Report your results providing error bars



Summary: Pipeline

Design/choose an appropriate learning algorithm and features



Select evaluation method



Choose appropriate performance metrics



Select baseline approaches for comparison



Report your results providing error bars

There are many ways to solve a given task (e.g., predicting student performance). It is important that:

- You provide a clean and complete evaluation of your solution
- You are able to justify your decisions for each step

Knowledge Tracing

Machine Learning for Behavioral Data
March 27, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Knowledge Tracing
7	Knowledge Tracing
8	Spring Break

Supervised learning on time series:

- Probabilistic graphical models
- Neural networks: LSTM, GRU, etc.

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

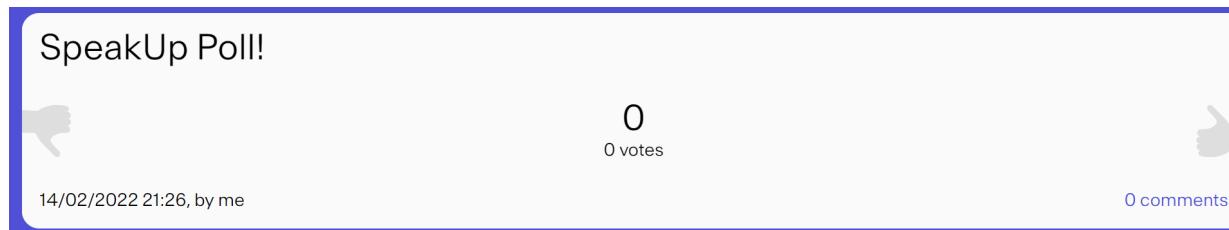
<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

[Model Evaluation] Given a data set $\{1,2,3,4\}$, one possible bootstrap set is $\{1,1,1,1\}$:

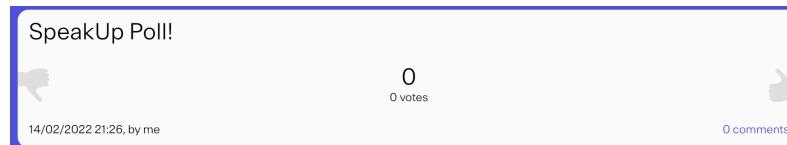
- a) True
- b) False



Short quiz about the past...

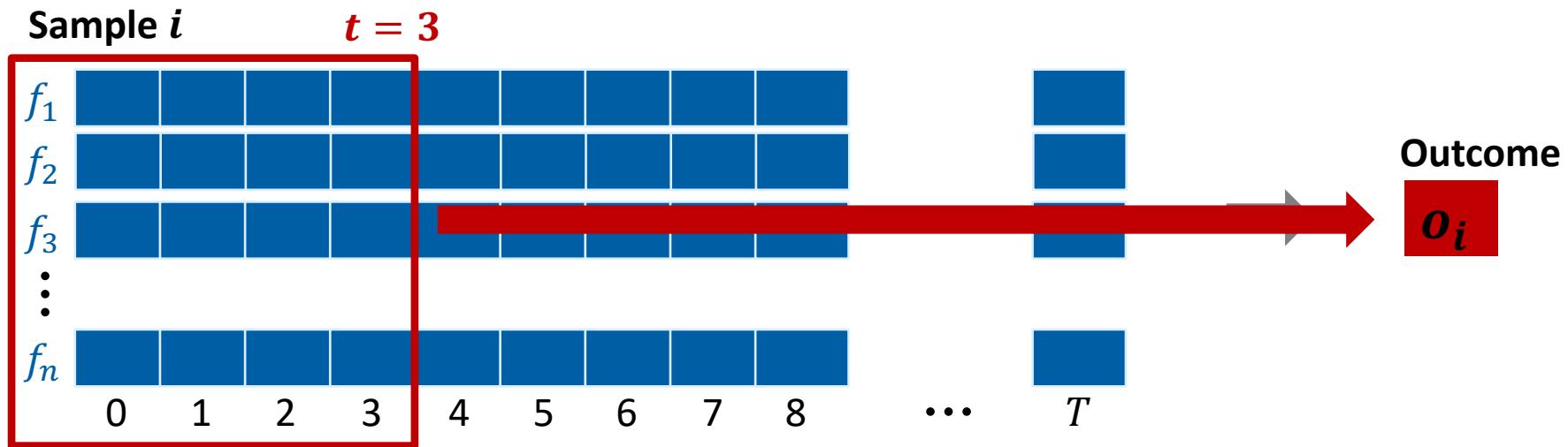
[Model Evaluation] Which of the following statements about k-fold cross validation are wrong? N denotes the number of samples in the data set, k the number of folds:

- a) k must always be smaller than N .
- b) The smaller k is, the more expensive it is to compute the error.
- c) Cross validation can be used to tune model hyperparameters.
- d) Cross validation is not a valid method for computing the generalization error of a model.



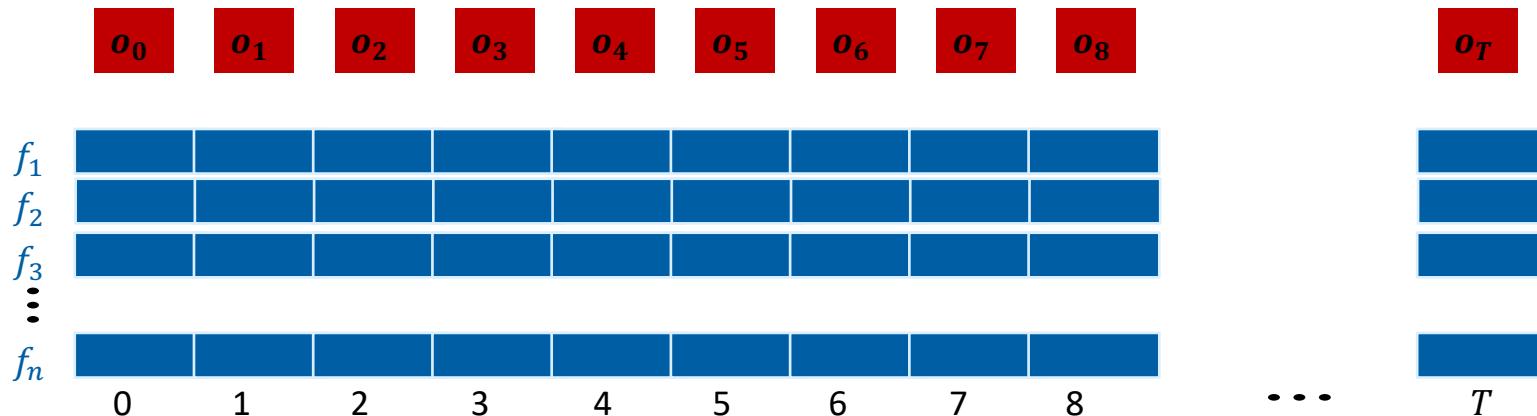
Time Series – Prediction Task

- Prediction of a target variable after $t < T$ time steps, where T is the total number of time steps



Time Series – Tracing Task

- Prediction of a target variable after $t < T$ time steps, where T is the total number of time steps
- Prediction of a variable in time step $t + 1$, based on time steps $0, \dots, t$



Today: Tracing Student Knowledge

- Is the student learning?
 - Measure what the student *knows* at a specific time t
 - More specifically: knowledge of the student about relevant knowledge components (skills)



Task:

$$50 - 23 = ?$$

$$75 - 12 = ?$$

$$38 - 14 = ?$$

Answer:

27

Tracing Knowledge – why is it useful?

- Is the student learning?
 - Measure what the student *knows* at a specific time t
 - More specifically: knowledge of the student about relevant knowledge components (skills)

→ Choose the next appropriate activity

→ Know which activities support learning

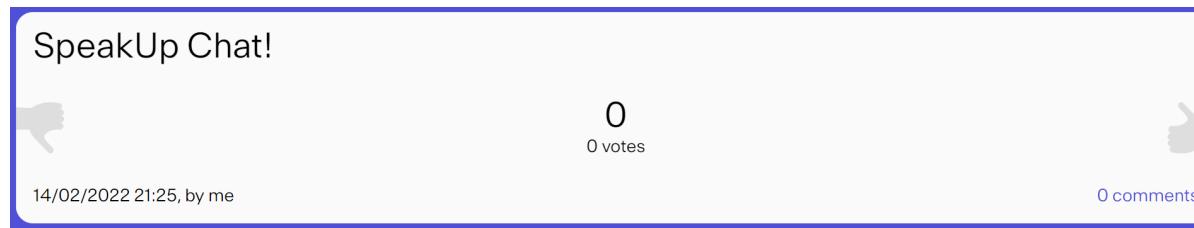
Today's Use Case

- ASSISTments is a free tool for assigning and assessing math problems and homework
 - All math problems (tasks/items) are associated to a specific skill/knowledge component
 - 4,217 middle-school students
 - 525,534 observations
-

Today: Tracing Student Knowledge

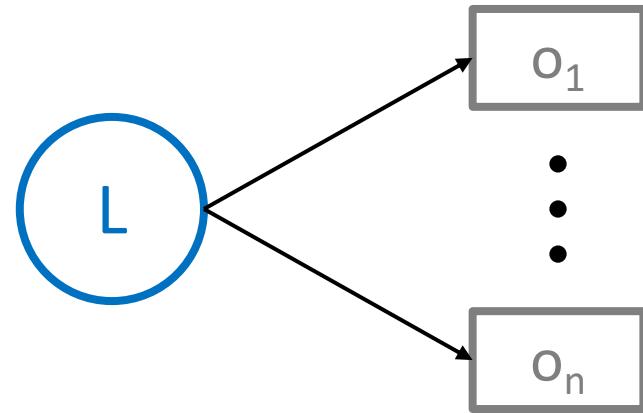
- Bayesian Knowledge Tracing (BKT)
 - Latent variables
 - BKT – Inference
 - Practical Example

What is a latent variable?



What is a latent variable?

- A **latent** variable L is a variable which is not directly observable/cannot be measured
- It is assumed to affect the outcome of other variables o , which can be **observed** (directly measured)

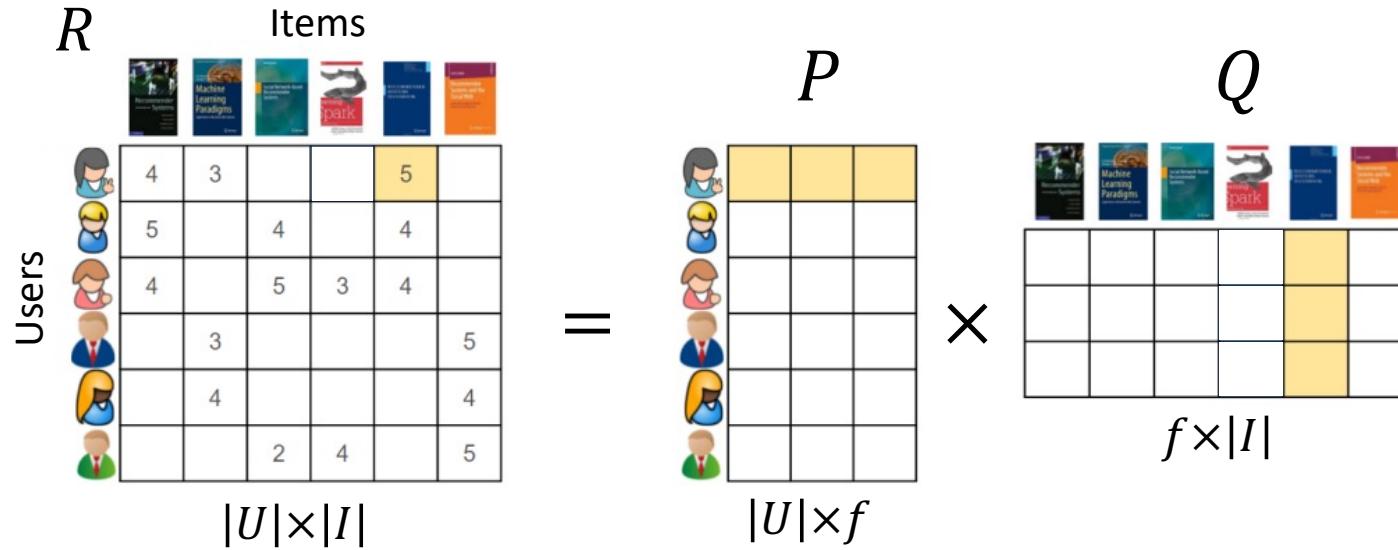


Why should we use latent variables?

- In many scientific fields, we are interested in concepts/factors that cannot directly be measured/observed:
 - Political sciences: leadership, political competence, etc.
 - Psychology: stress, self-worth, personality characteristics, talent, etc.
 - Education: memory, spatial ability, cognitive abilities, etc.
- We represent underlying concepts/factors by latent variables and infer them from the observed variables

Example 1: Recommender Systems

- Given: ratings of users u for items i (e.g., books)



Example 2: Education

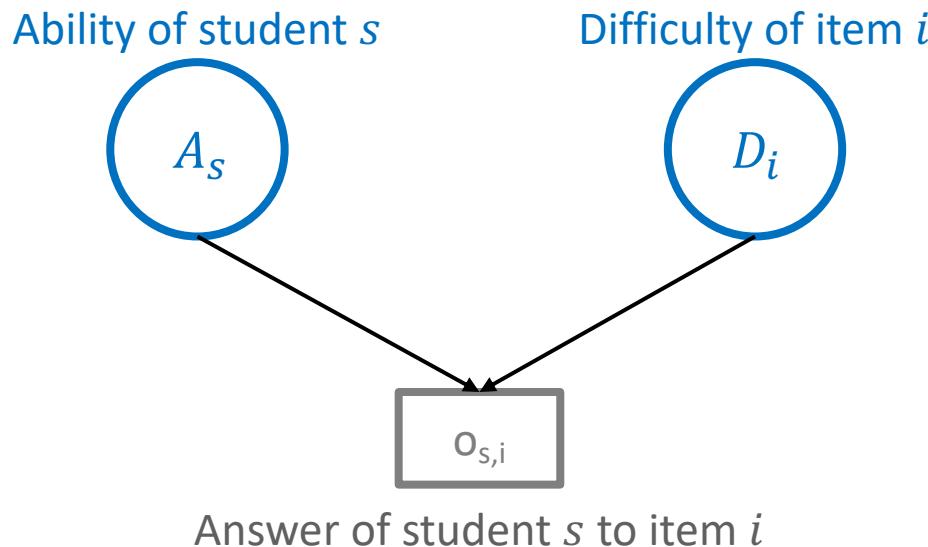
- Observations: binary answers (correct/wrong) of students to items (tasks)



Answer of student s to item i

Example 2: Education

- Observations: binary answers (correct/wrong) of students to items (tasks)



Is the student learning?



Task: $50 - 23 = ?$ $75 - 12 = ?$ $38 - 14 = ?$

Answer: 27 61 24

What are we measuring?



Task: $50 - 23 = ?$ $75 - 12 = ?$ $38 - 14 = ?$

Answer: 27 61 24

1

0

1

Binary observations of student answers



Subtraction 0-100

1

2

...

n

0

0

1

0

1

1

Predicting future performance



Subtraction 0-100

1

2

...

n

n+1

0

0

1

0

1

1

?

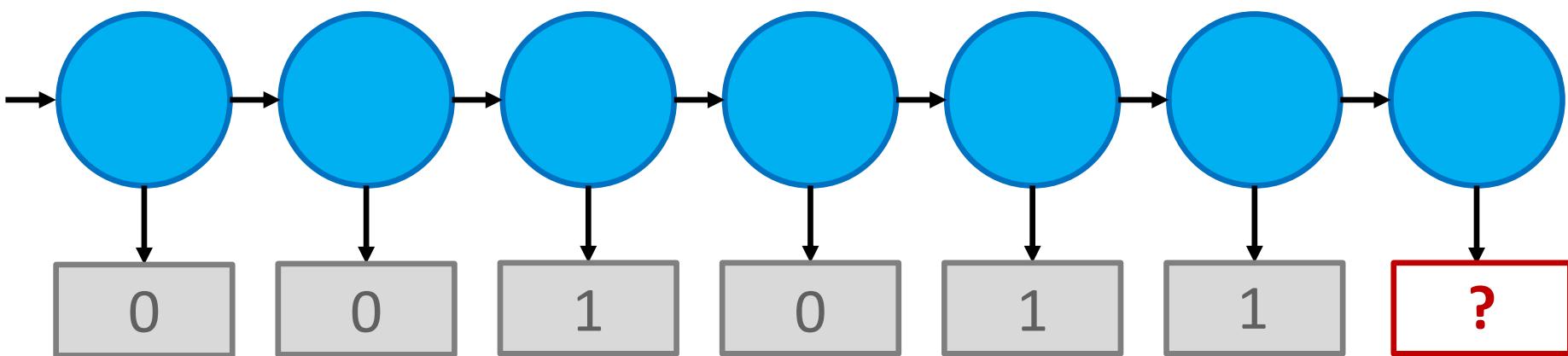
Bayesian Knowledge Tracing (BKT)



Latent variable



Observed variable

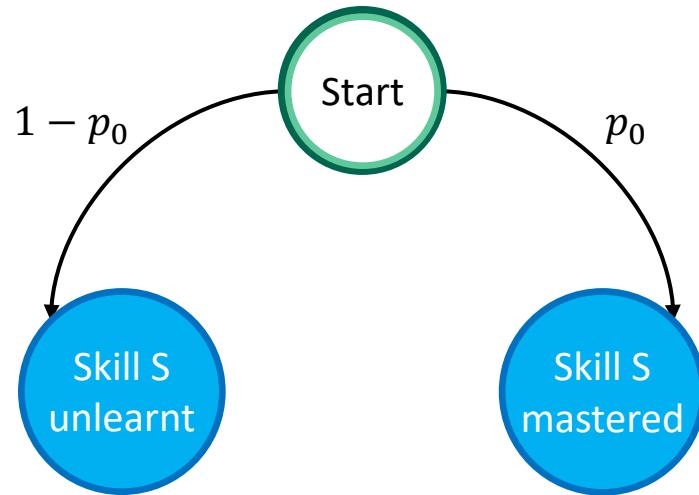


Bayesian Knowledge Tracing (BKT)



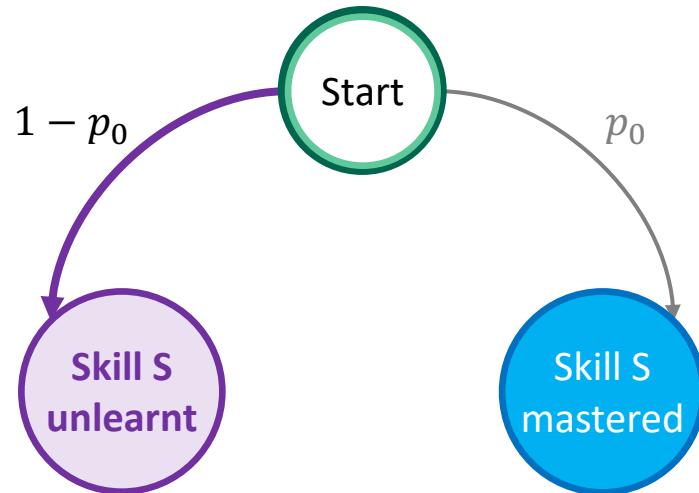
Observations for student s :

Bayesian Knowledge Tracing (BKT)



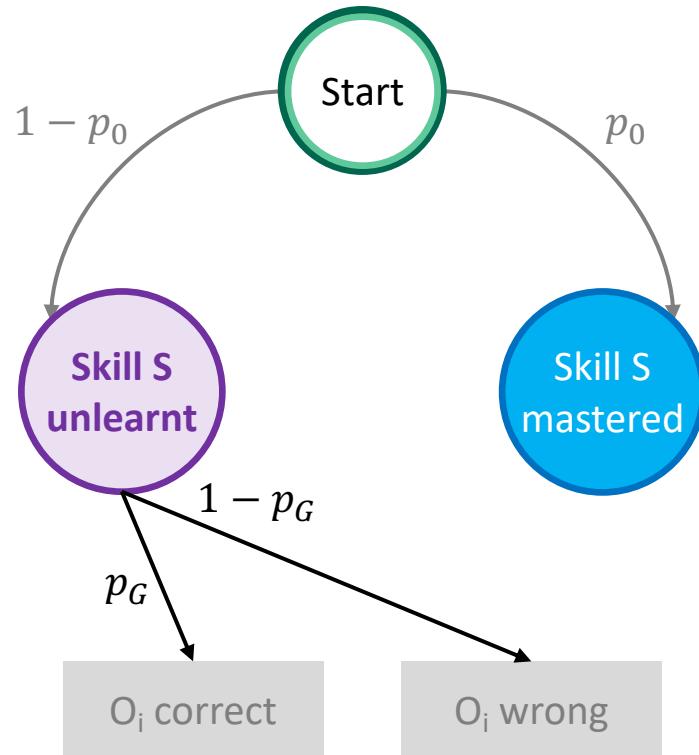
Observations for student s :

Bayesian Knowledge Tracing (BKT)



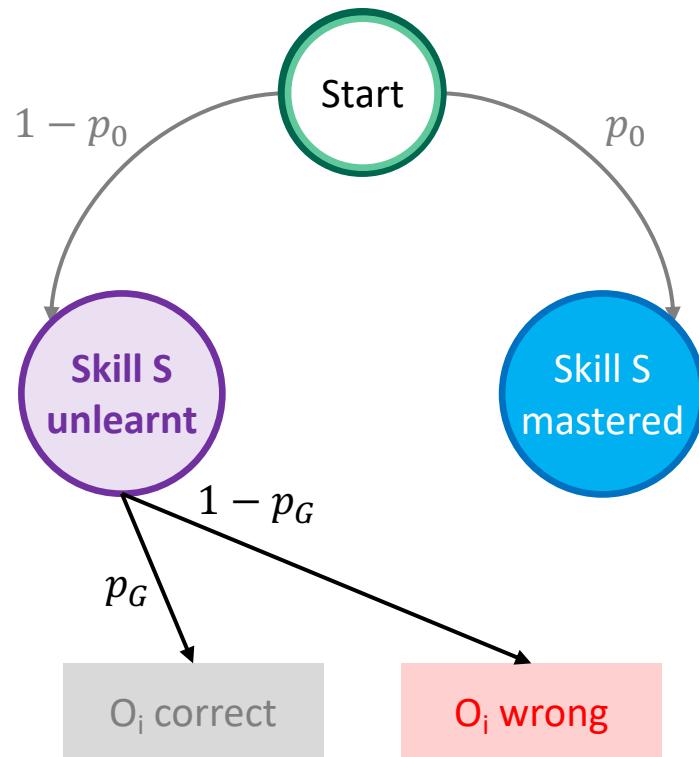
Observations for student s :
 $t = 0$:

Bayesian Knowledge Tracing (BKT)



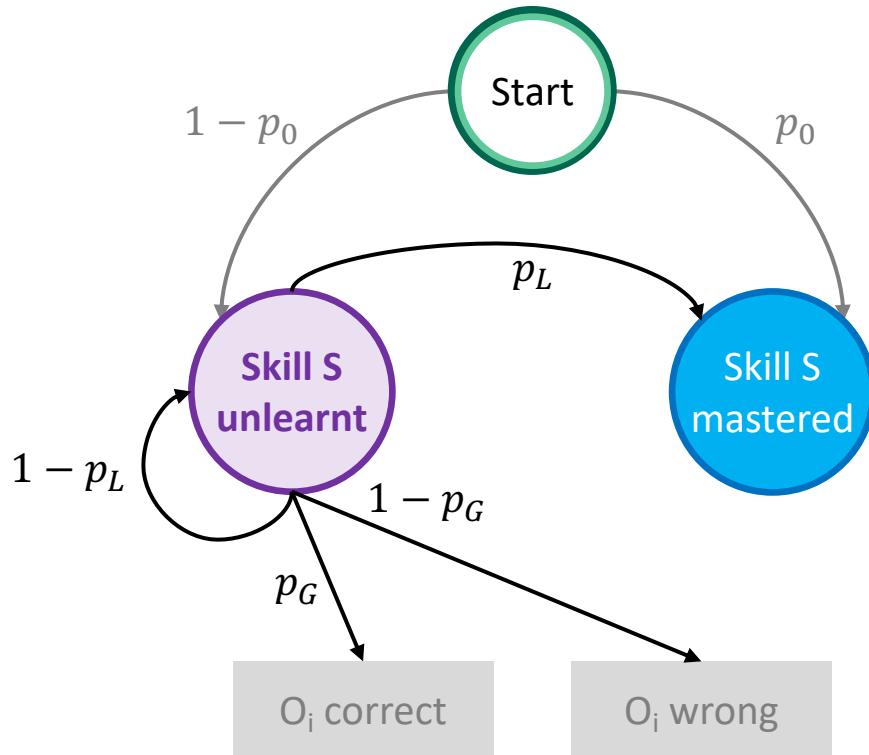
Observations for student s :
 $t = 0$:

Bayesian Knowledge Tracing (BKT)



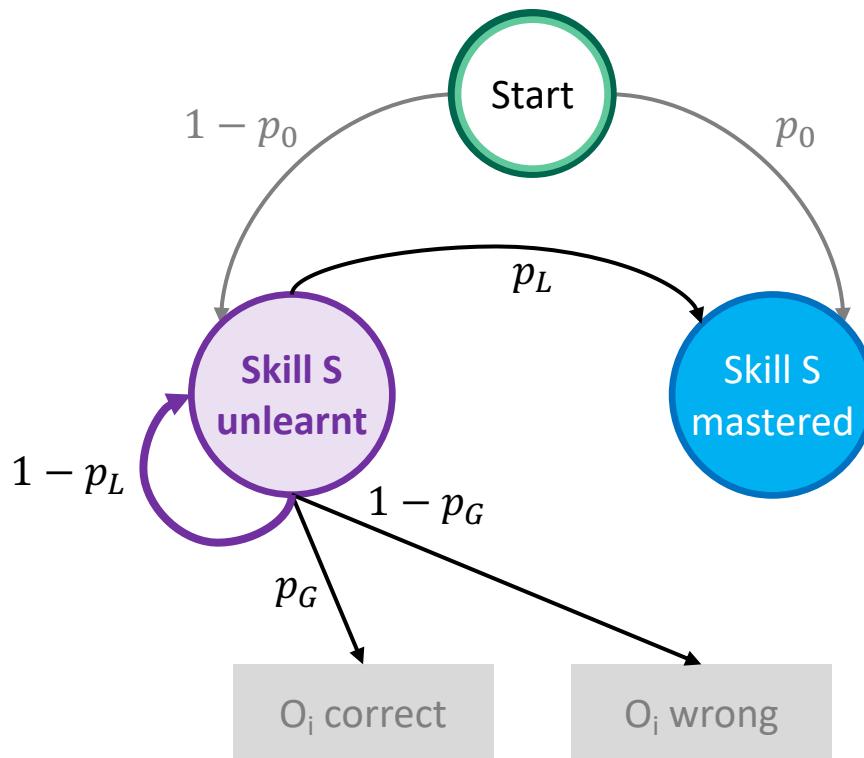
Observations for student s :
 $t = 0: 0$

Bayesian Knowledge Tracing (BKT)



Observations for student s :
 $t = 0: 0$

Bayesian Knowledge Tracing (BKT)

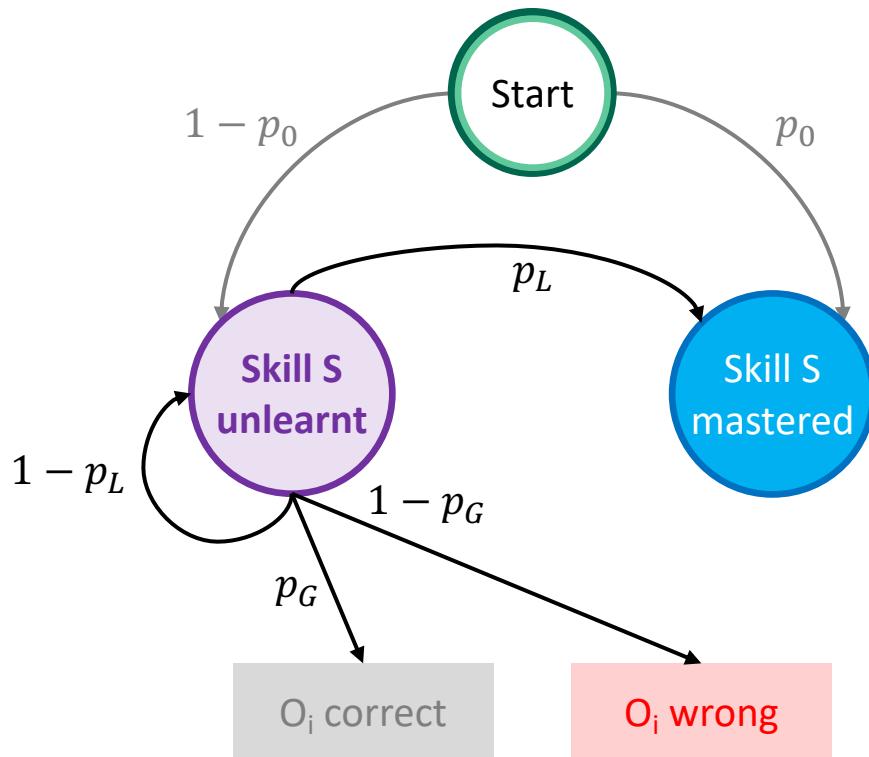


Observations for student s :

$t = 0: 0$

$t = 1:$

Bayesian Knowledge Tracing (BKT)

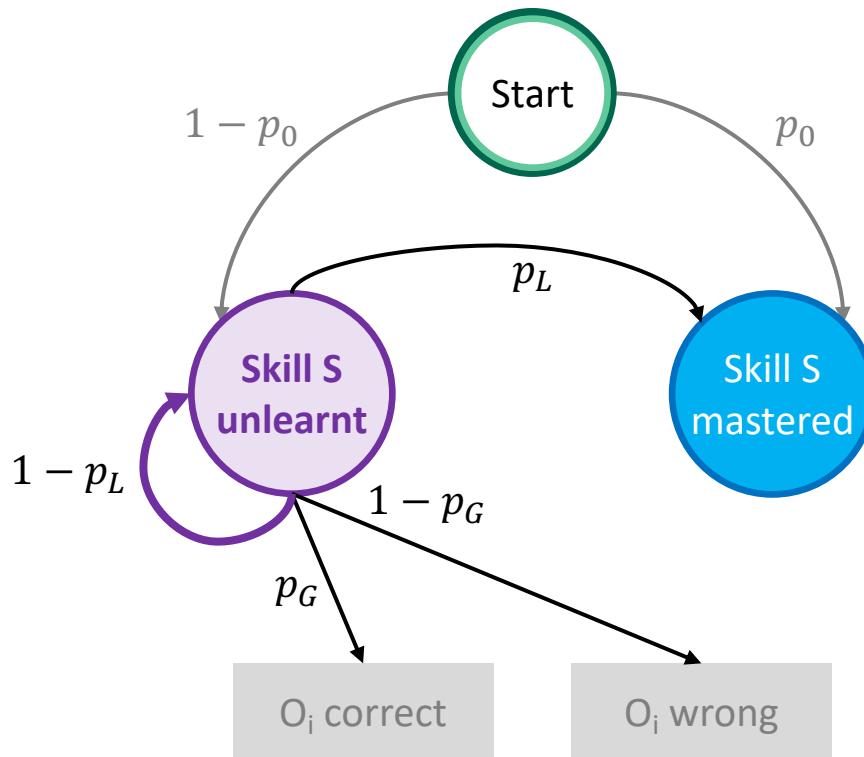


Observations for student s :

$t = 0: 0$

$t = 1: 0$

Bayesian Knowledge Tracing (BKT)



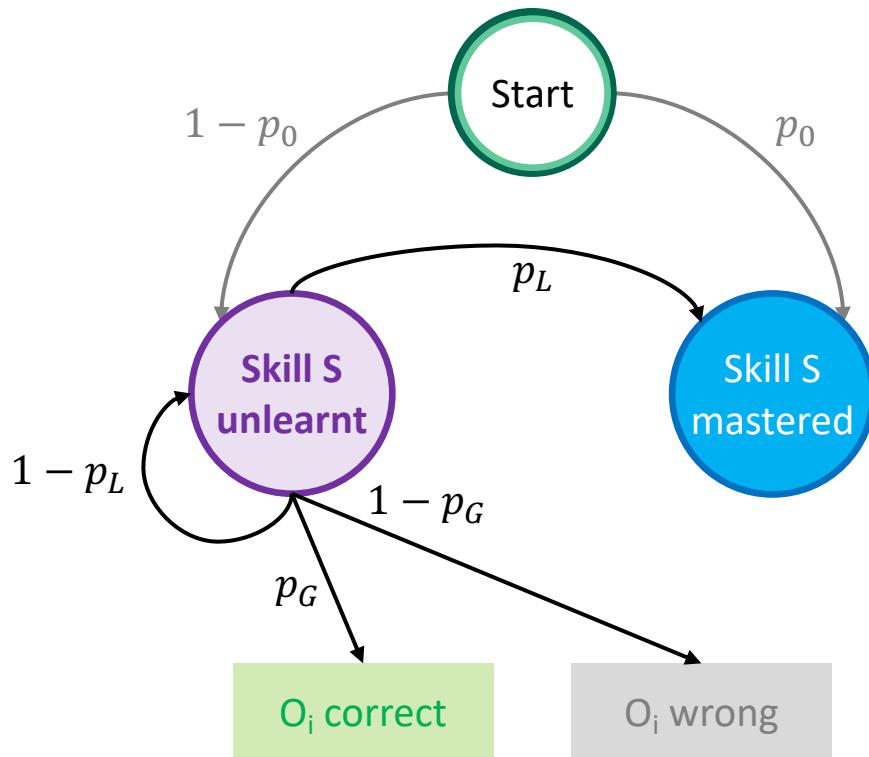
Observations for student s :

$t = 0: 0$

$t = 1: 0$

$t = 2:$

Bayesian Knowledge Tracing (BKT)



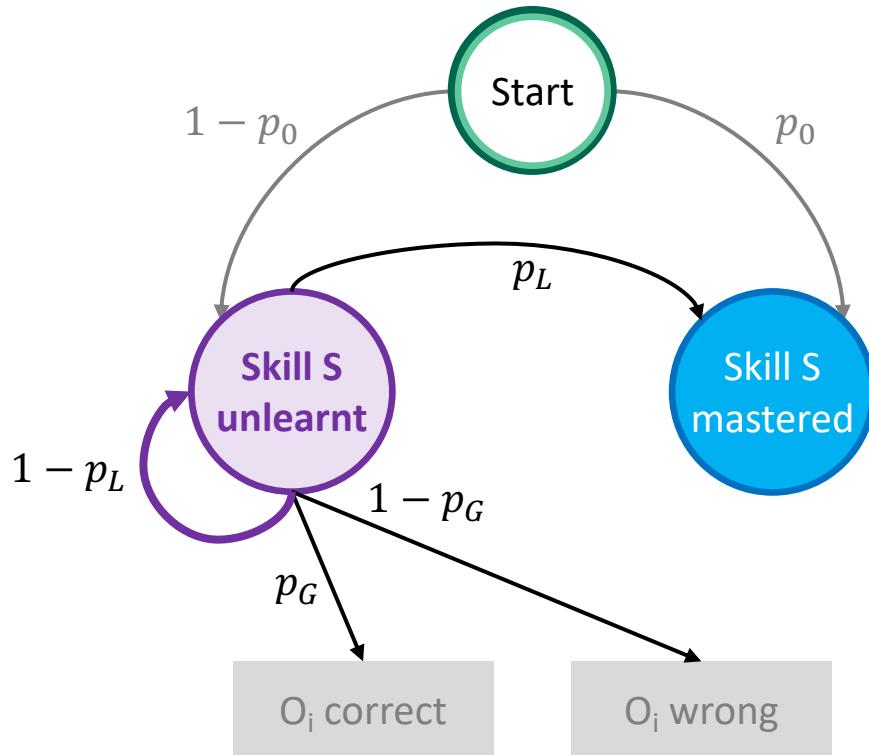
Observations for student s :

$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

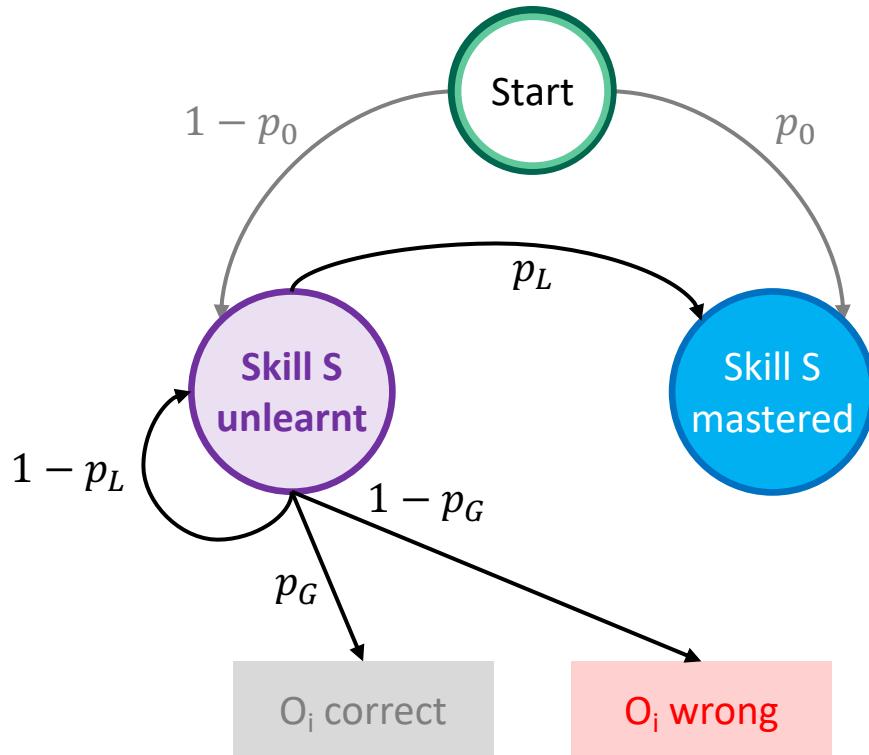
$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

$t = 3:$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

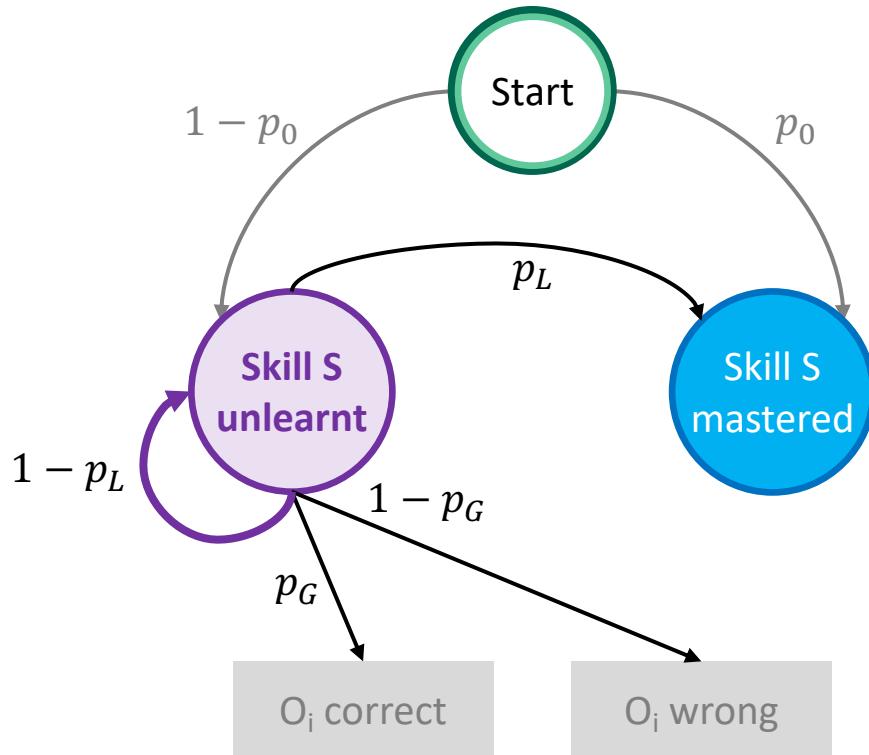
$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

$t = 3: 0$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

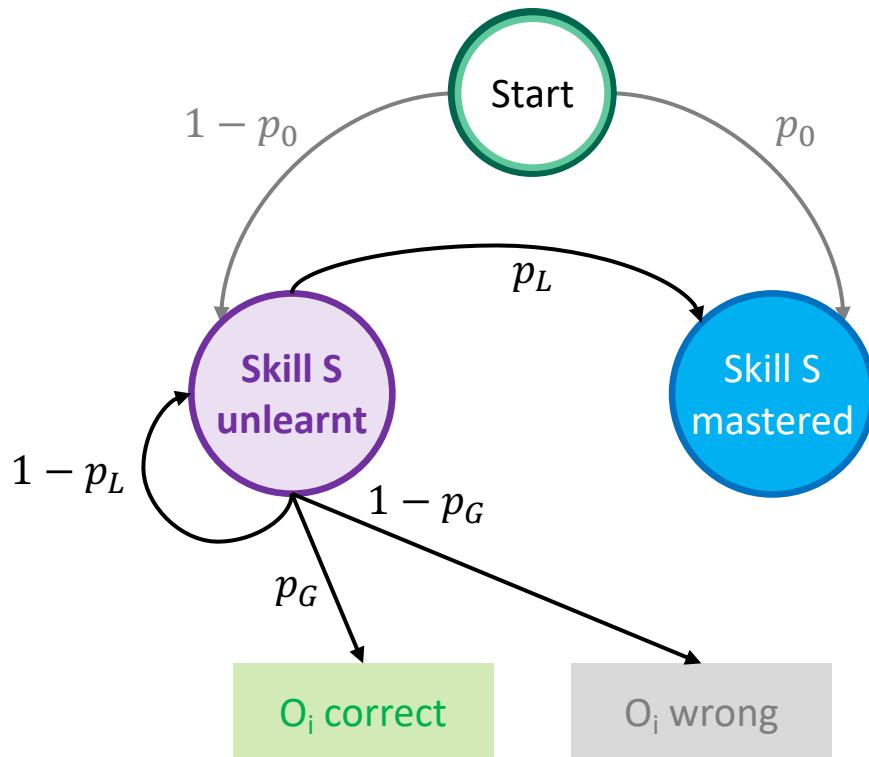
$t = 1: 0$

$t = 2: 1$

$t = 3: 0$

$t = 4:$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

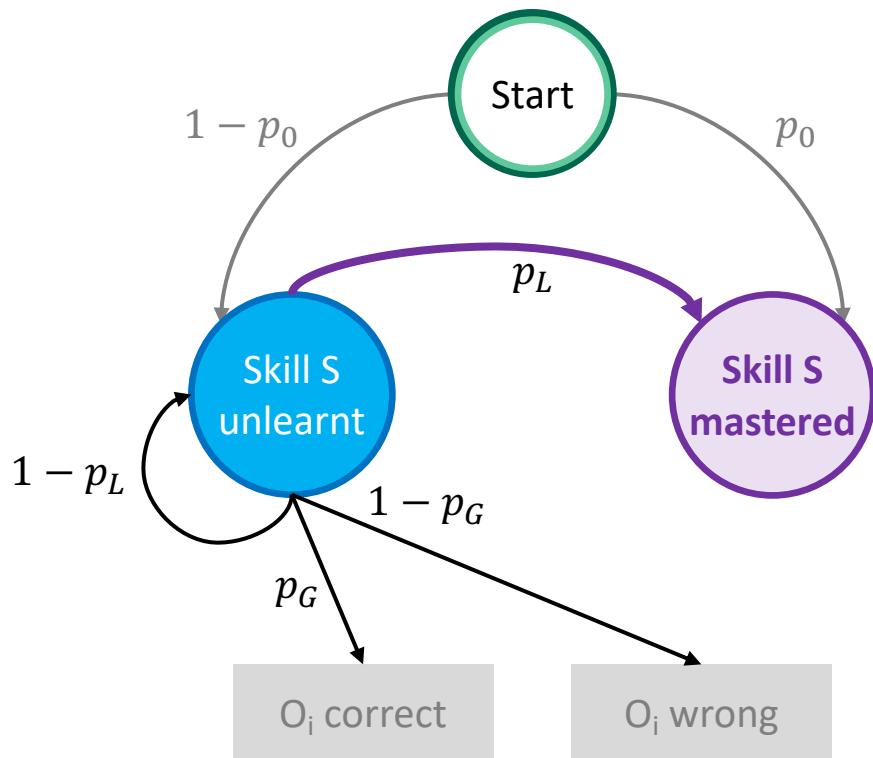
$t = 1: 0$

$t = 2: 1$

$t = 3: 0$

$t = 4: 1$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

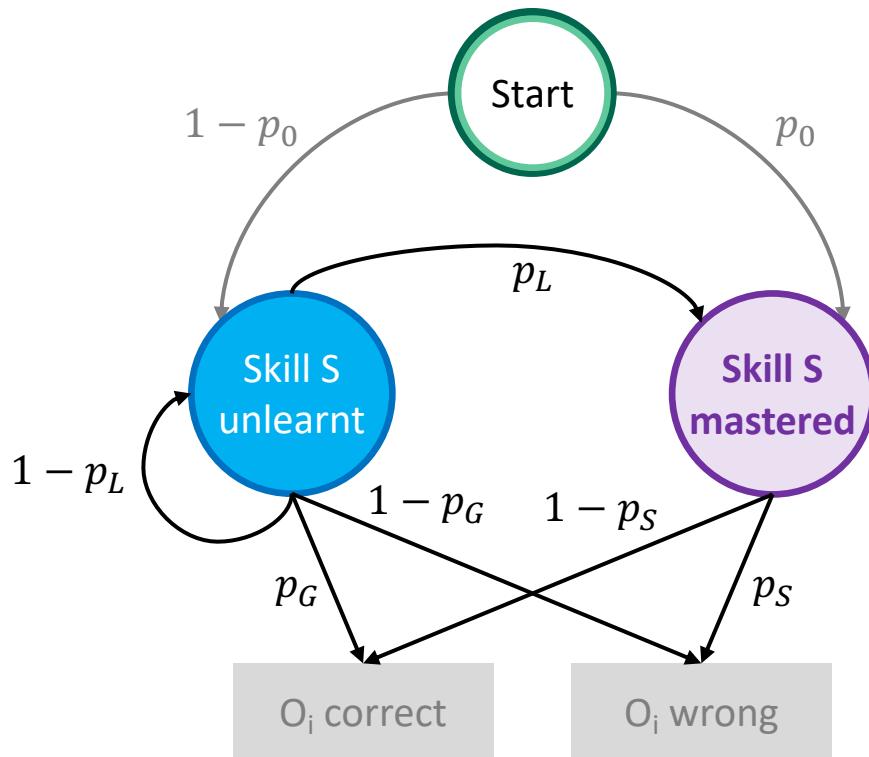
$t = 2: 1$

$t = 3: 0$

$t = 4: 1$

$t = 5:$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

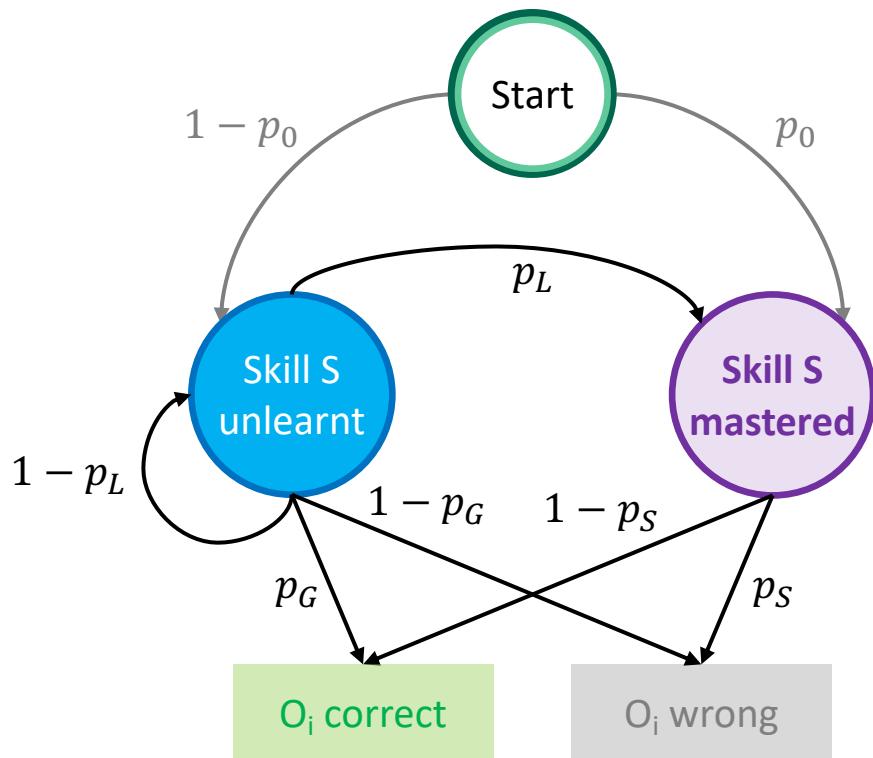
$t = 2: 1$

$t = 3: 0$

$t = 4: 1$

$t = 5:$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

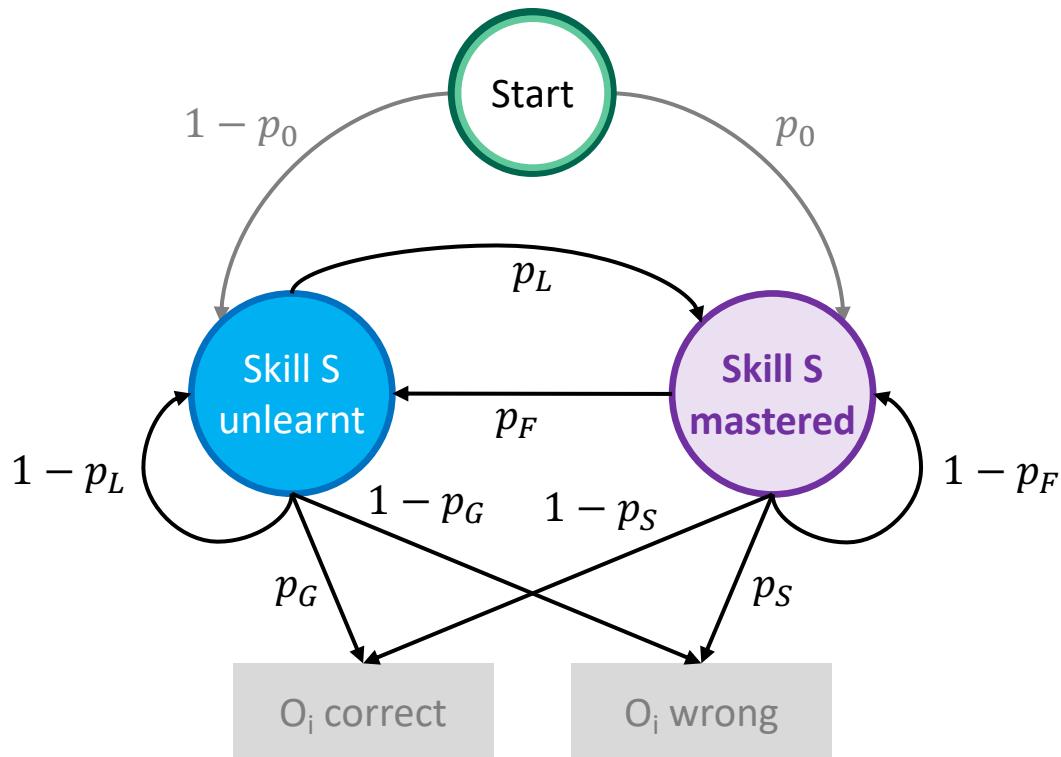
$t = 2: 1$

$t = 3: 0$

$t = 4: 1$

$t = 5: 1$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

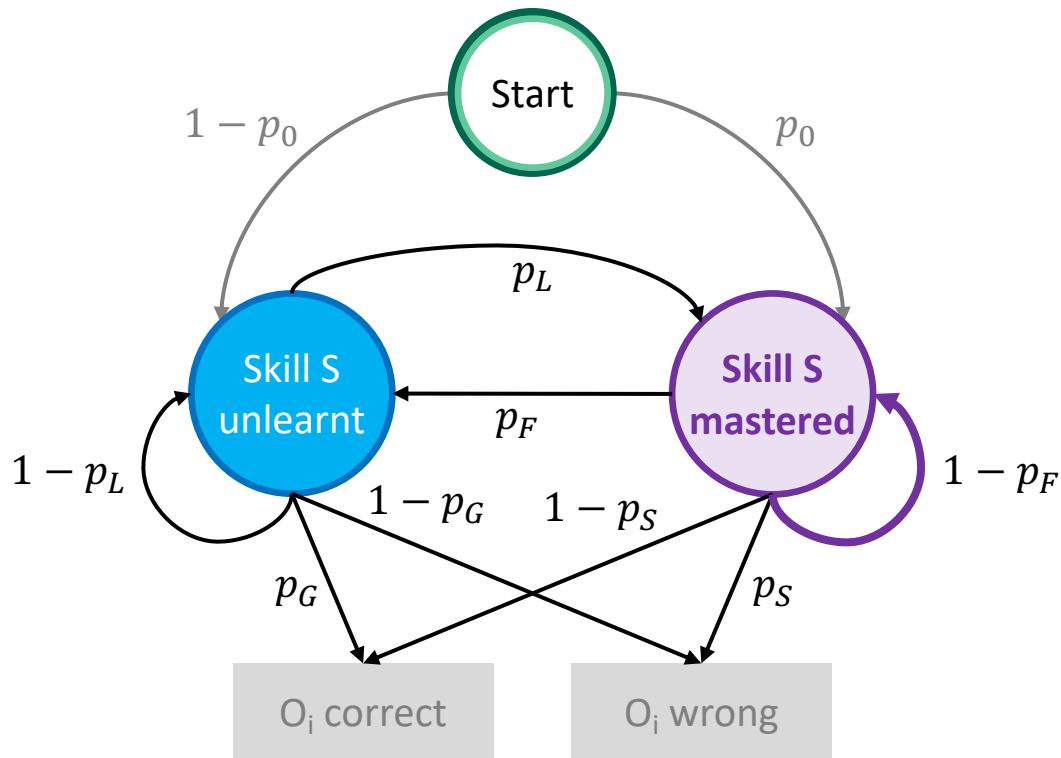
$t = 2: 1$

$t = 3: 0$

$t = 4: 1$

$t = 5: 1$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

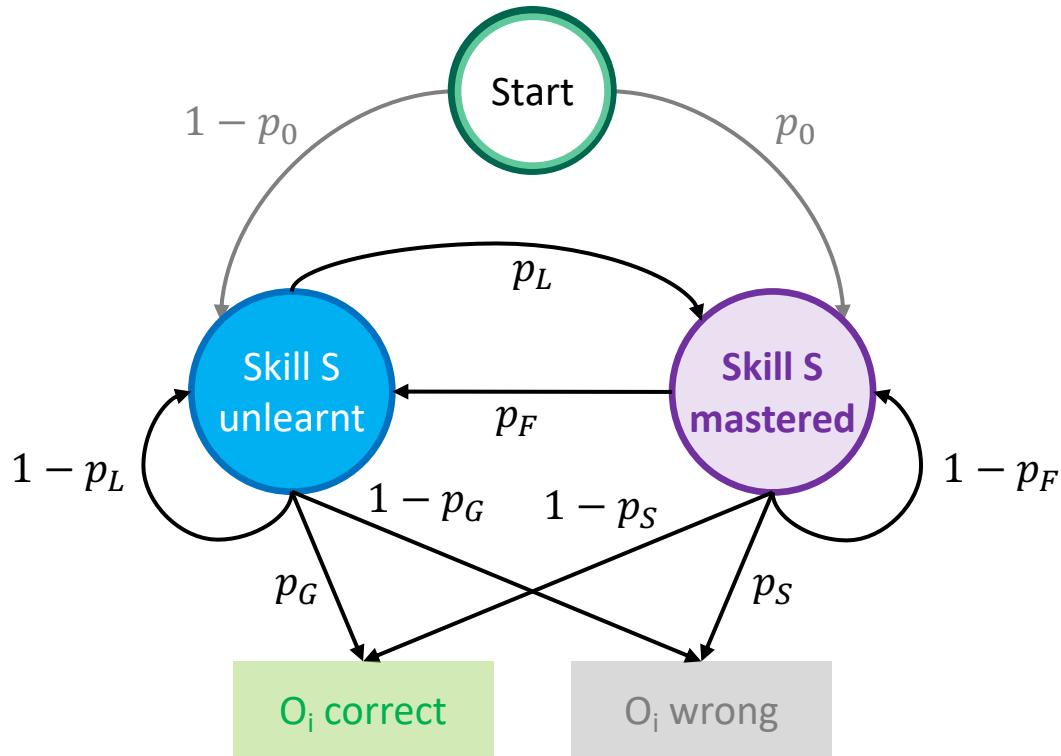
$t = 3: 0$

$t = 4: 1$

$t = 5: 1$

$t = 6:$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

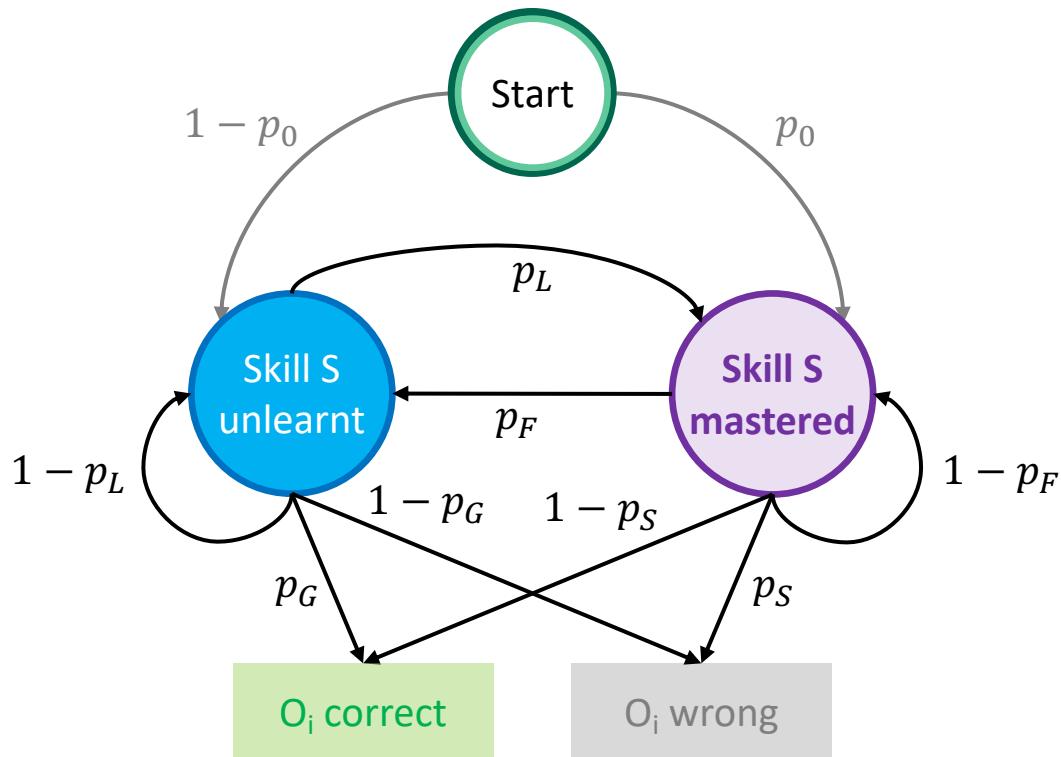
$t = 3: 0$

$t = 4: 1$

$t = 5: 1$

$t = 6: 1$

Bayesian Knowledge Tracing (BKT)



Observations for student s :

$t = 0: 0$

$t = 1: 0$

$t = 2: 1$

$t = 3: 0$

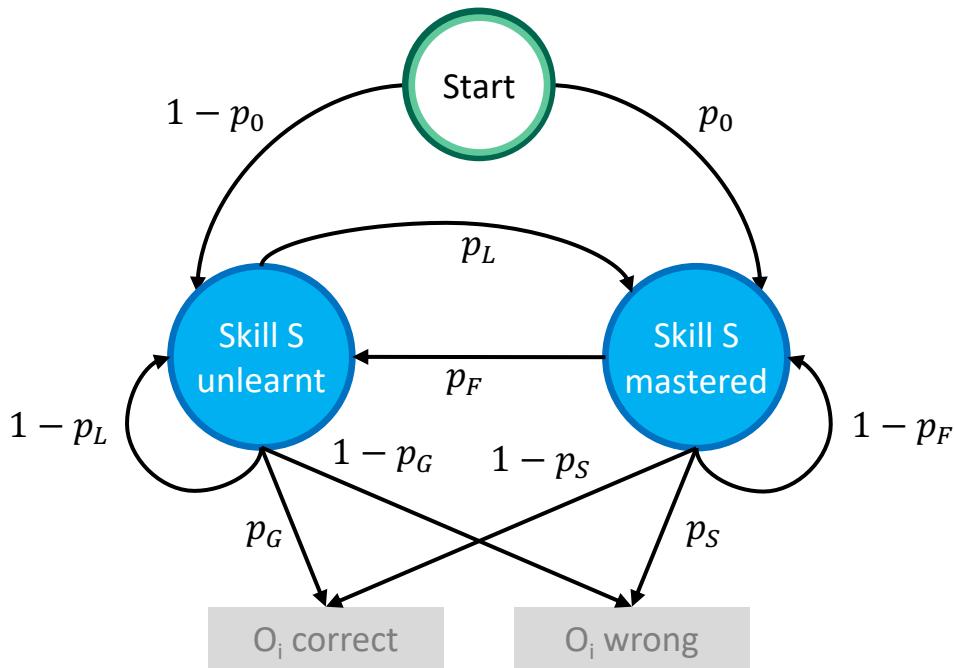
$\mathbf{o}_s = [0,0,1,0,1,1,1]$

$t = 4: 1$

$t = 5: 1$

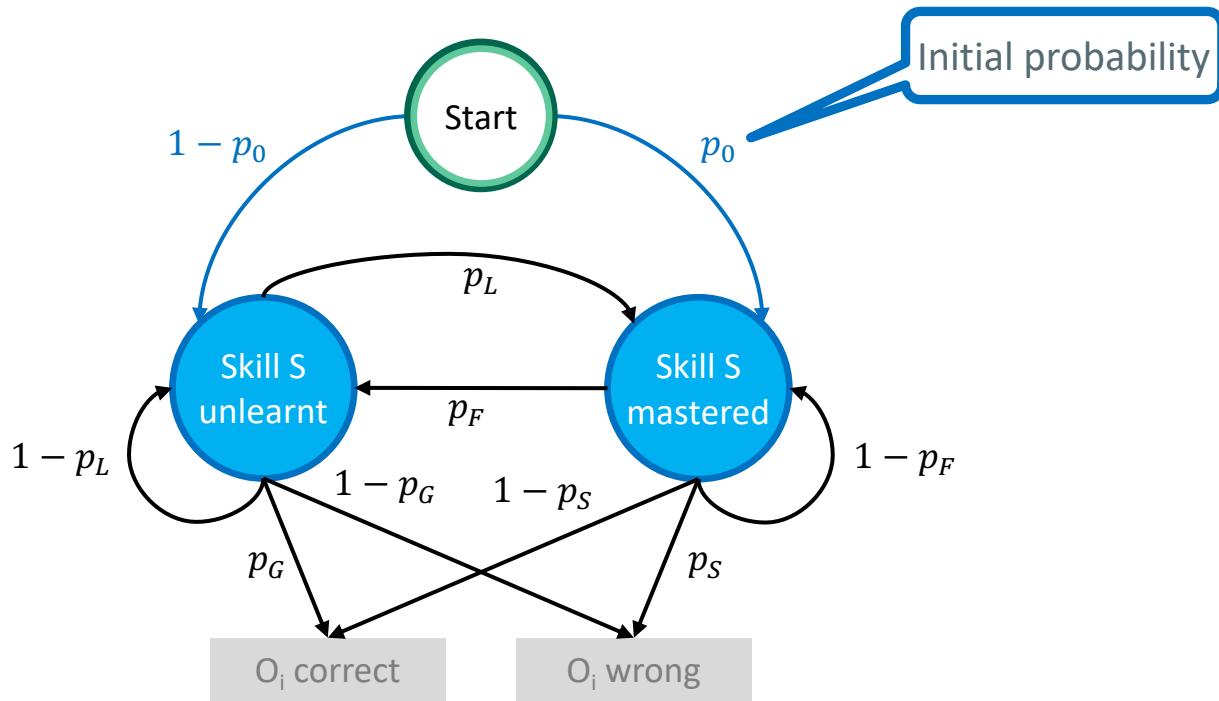
$t = 6: 1$

BKT - Terminology

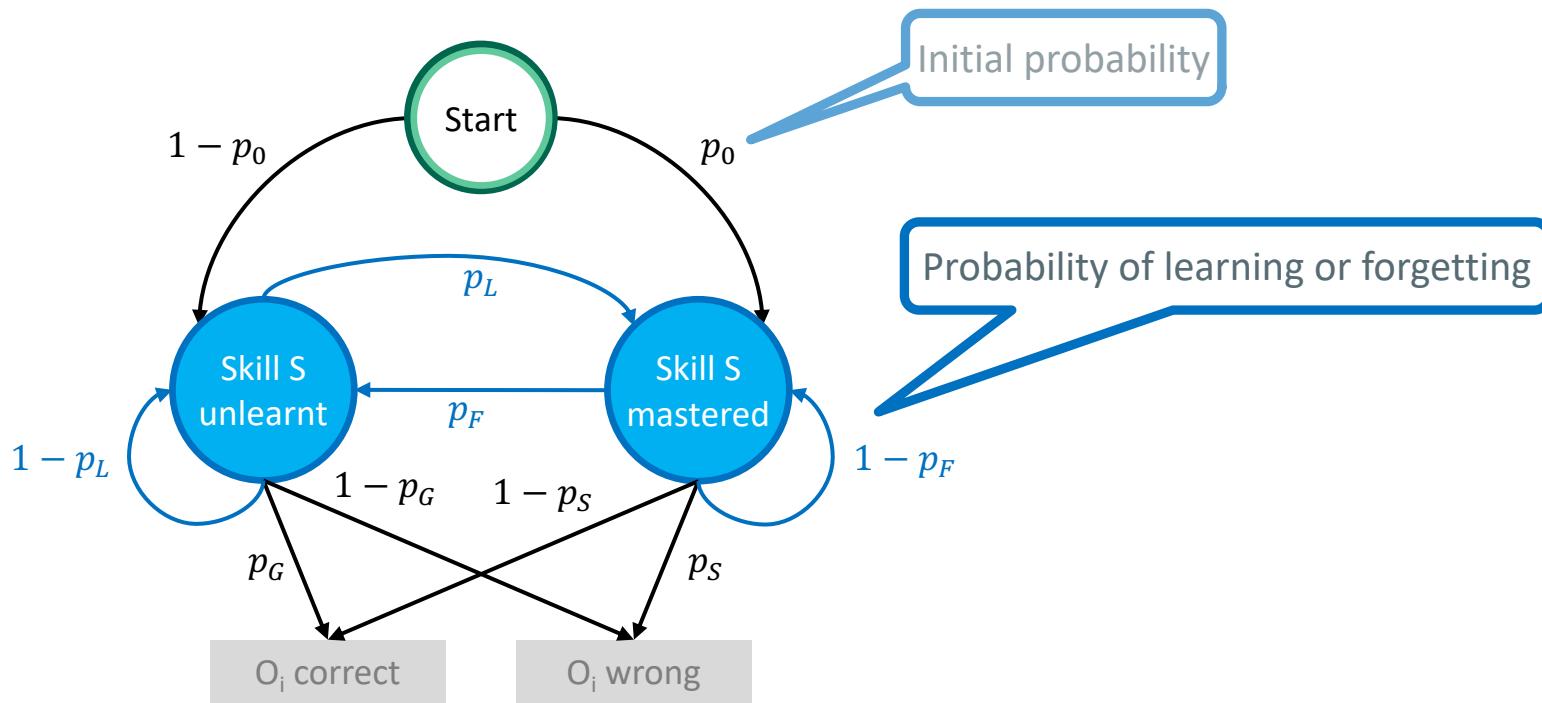


- One **latent** variable (S) with two possible states
- Observations (also binary)
- Five parameters:
 - Emission probabilities
 - Transition probabilities
$$\theta = \{p_0, p_L, p_F, p_S, p_G\}$$

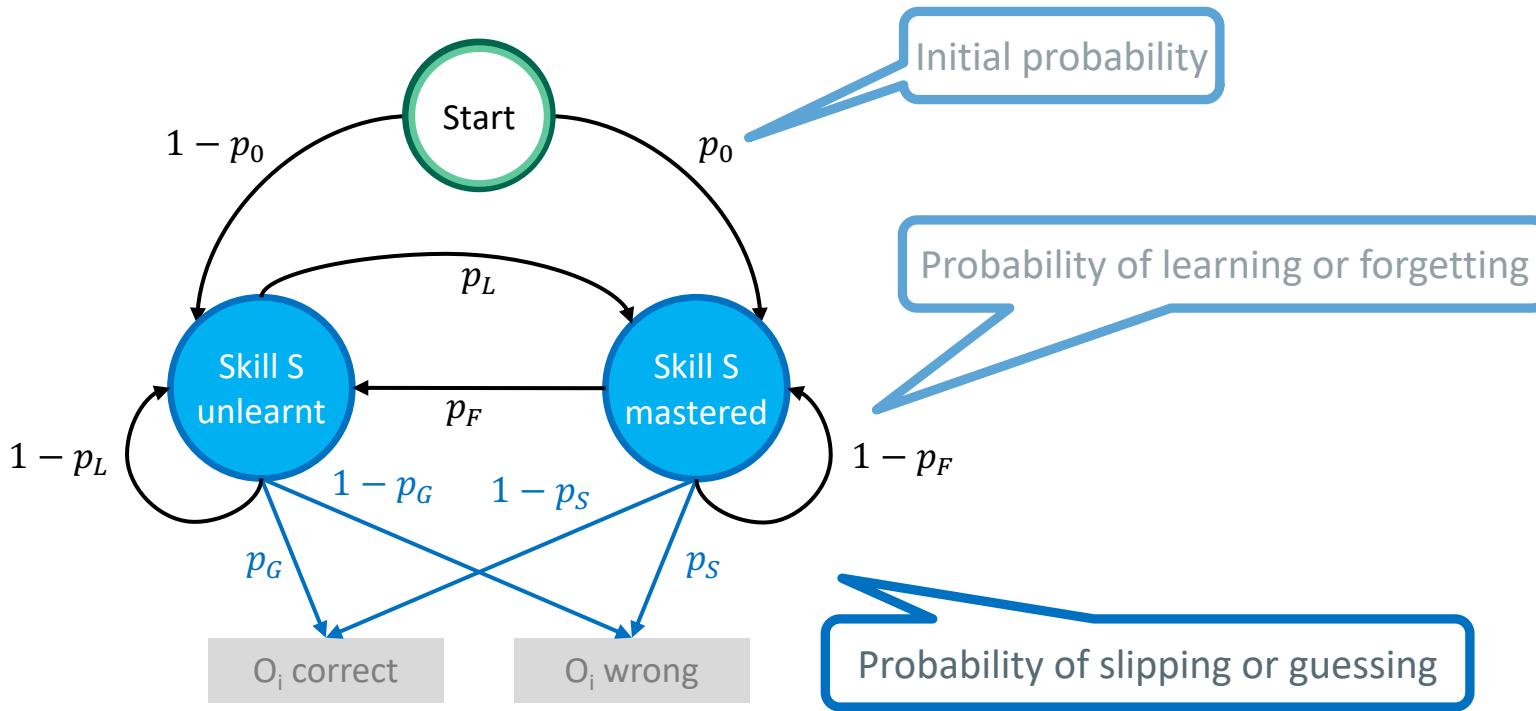
BKT parameters are interpretable



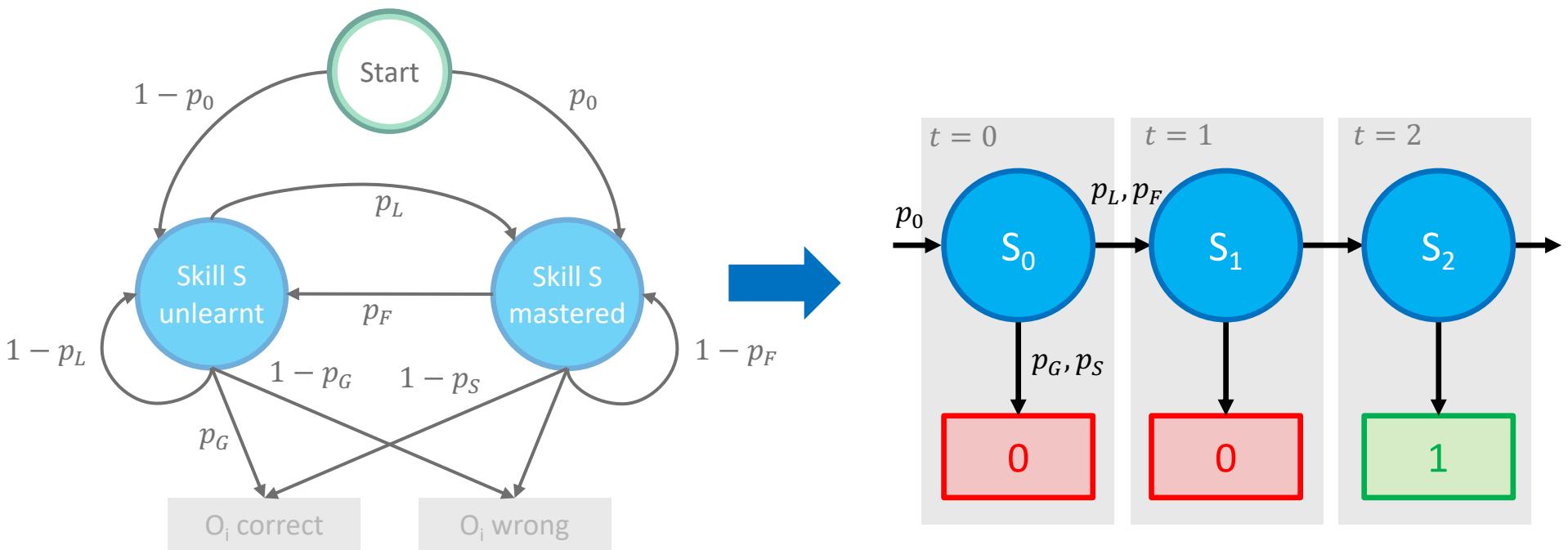
BKT parameters are interpretable



BKT parameters are interpretable

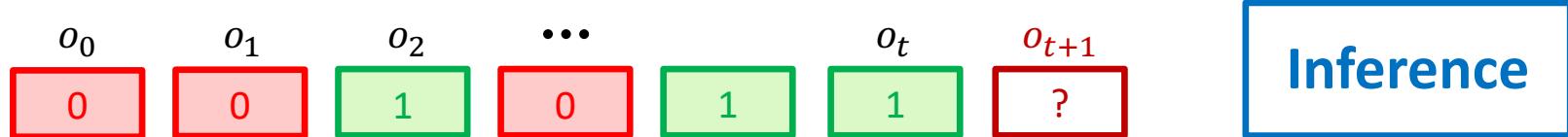


BKT – unrolled over time

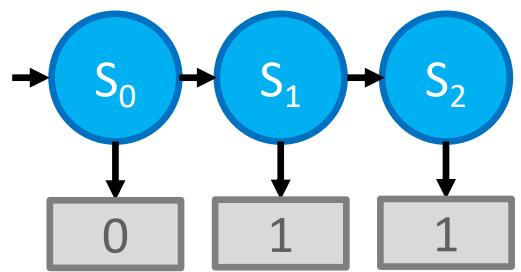


Two tasks need to be solved in practice

- Given a model with parameters $\theta = \{p_0, p_L, p_F, p_S, p_G\}$ and a sequence of observations $\mathbf{o} = [o_0, \dots, o_t]$ from a student s , predict o_{t+1}



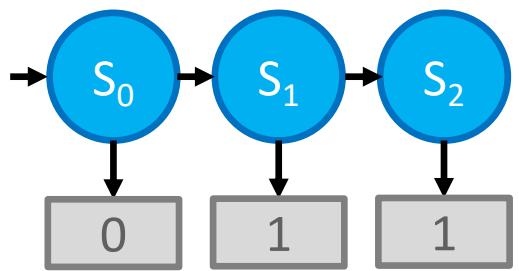
Inference Example



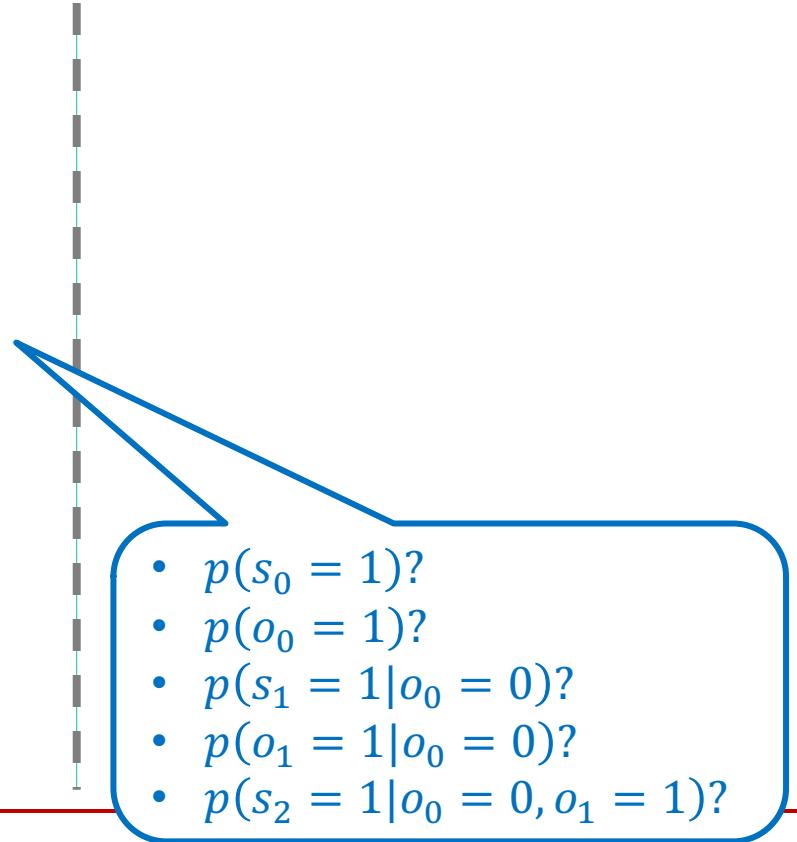
$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



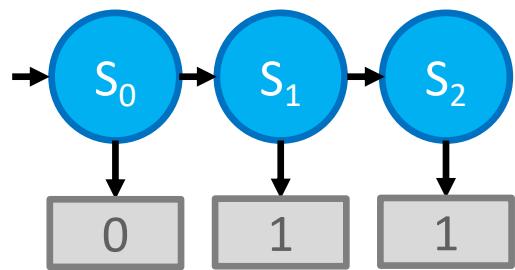
Inference Example – Your Turn



$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



Inference Example – Your Turn



$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$

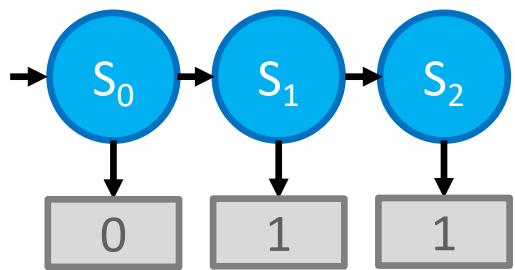
S_0	$p(S_0)$
1	p_0
0	$1-p_0$

S_t	S_{t+1}	$p(S_{t+1} S_t)$
0	0	$1 - p_L$
0	1	p_L
1	0	p_F
1	1	$1 - p_F$

S_t	O_t	$p(O_t S_t)$
0	0	$1 - p_G$
0	1	p_G
1	0	p_S
1	1	$1 - p_S$

- $p(s_0 = 1)?$
- $p(o_0 = 1)?$
- $p(s_1 = 1|o_0 = 0)?$
- $p(o_1 = 1|o_0 = 0)?$
- $p(s_2 = 1|o_0 = 0, o_1 = 1)?$

Inference Example – Your Turn



$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$

S_0	$p(S_0)$
1	p_0
0	$1-p_0$

S_t	S_{t+1}	$p(S_{t+1} S_t)$
0	0	$1 - p_L$
0	1	p_L
1	0	p_F
1	1	$1 - p_F$

S_t	O_t	$p(O_t S_t)$
0	0	$1 - p_G$
0	1	p_G
1	0	p_S
1	1	$1 - p_S$

Some useful rules:

$$p(A, B) = p(A|B) \cdot p(B)$$

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

$$\begin{aligned} p(A = 1) &= p(A = 1, B = 1) \\ &\quad + p(A = 1, B = 0) \end{aligned}$$

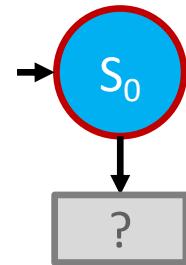
- $p(s_0 = 1)?$
- $p(o_0 = 1)?$
- $p(s_1 = 1|o_0 = 0)?$
- $p(o_1 = 1|o_0 = 0)?$
- $p(s_2 = 1|o_0 = 0, o_1 = 1)?$

Inference in BKT models

Equations for time step 0:

$$p(s_0 = 1) = p_0$$

$$p(s_0 = 0) = 1 - p_0$$

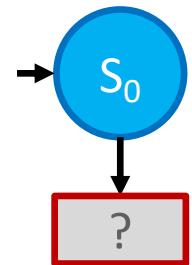


Inference in BKT models

Equations for time step 0:

$$p(s_0 = 1) = p_0$$

$$p(s_0 = 0) = 1 - p_0$$

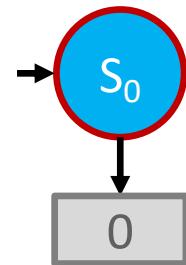


$$\begin{aligned} p(o_0 = 1) &= p(o_0 = 1, s_0 = 1) + p(o_0 = 1, s_0 = 0) \\ &= (1 - p_s) \cdot p_0 + p_G \cdot (1 - p_0) \end{aligned}$$

$$p(o_0 = 0) = p_S \cdot p_0 + (1 - p_G) \cdot (1 - p_0)$$

Inference in BKT models

$$p(s_0 = 1 | o_0 = 0) = \frac{p(o_0 = 0 | s_0 = 1) \cdot p(s_0 = 1)}{p(o_0 = 0)}$$
$$p_{s_0|0} = \frac{p_s \cdot p_0}{p_s \cdot p_0 + (1 - p_G) \cdot (1 - p_0)}$$



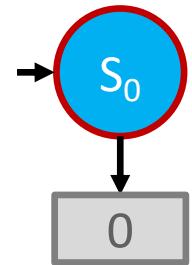
$$p(s_0 = 0 | o_0 = 0) = 1 - p_{s_0|0}$$

Inference in BKT models

$$p(s_0 = 1 | o_0 = 1) = \frac{p(o_0 = 1 | s_0 = 1) \cdot p(s_0 = 1)}{p(o_0 = 1)}$$

$p_{s_0|1}$

$$= \frac{(1 - p_s) \cdot p_0}{(1 - p_s) \cdot p_0 + p_G \cdot (1 - p_0)}$$



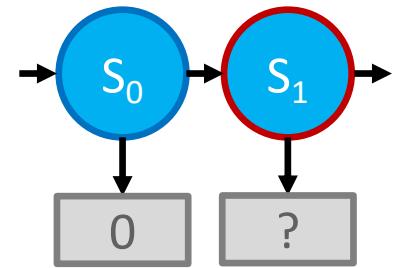
$$p(s_0 = 0 | o_0 = 1) = 1 - p_{s_0|1}$$

Inference in BKT models

Equations for time step 1:

$$\begin{aligned} p(s_1 = 1 | o_0 = 0) &= \frac{p(s_1 = 1, o_0 = 0)}{p(o_0 = 0)} \\ &= \frac{p(s_1 = 1, s_0 = 1, o_0 = 0)}{p(o_0 = 0)} + \frac{p(s_1 = 1, s_0 = 0, o_0 = 0)}{p(o_0 = 0)} \\ &= \frac{p(s_1 = 1 | s_0 = 1) \cdot p(o_0 = 0 | s_0 = 1) \cdot p(s_0 = 1)}{p(o_0 = 0)} \\ &\quad + \frac{p(s_1 = 1 | s_0 = 0) \cdot p(o_0 = 0 | s_0 = 0) \cdot p(s_0 = 0)}{p(o_0 = 0)} \end{aligned}$$

$$p(s_1 = 1 | o_0 = 0) = (1 - p_F) \cdot p_{s_0|0} + p_L \cdot (1 - p_{s_0|0})$$



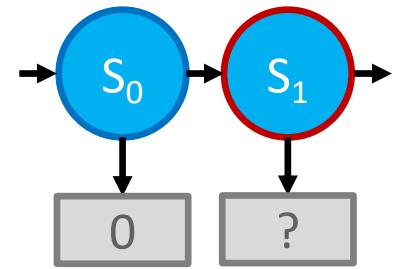
Inference in BKT models

$$p(s_1 = 1 | o_0 = 1) = (1 - p_F) \cdot p_{s_0|1} + p_L \cdot (1 - p_{s_0|1})$$

$$p(s_1 = 1 | o_0 = 0) = (1 - p_F) \cdot p_{s_0|0} + p_L \cdot (1 - p_{s_0|0})$$



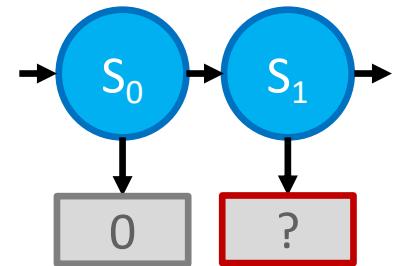
$$p_{s_1|o_0} = (1 - p_F) \cdot p_{s_0|o_0} + p_L \cdot (1 - p_{s_0|o_0})$$



Inference in BKT models

$$\begin{aligned}
 p(o_1 = 1 | o_0 = 0) &= \frac{p(o_1 = 1, o_0 = 0)}{p(o_0 = 0)} \\
 &\quad + \frac{p(o_1 = 1, s_1 = 1, s_0 = 1, o_0 = 0)}{p(o_0 = 0)} \\
 &\quad + \frac{p(o_1 = 1, s_1 = 1, s_0 = 0, o_0 = 0)}{p(o_0 = 0)} \\
 &\quad + \frac{p(o_1 = 1, s_1 = 0, s_0 = 1, o_0 = 0)}{p(o_0 = 0)} + \frac{p(o_1 = 1, s_1 = 0, s_0 = 0, o_0 = 0)}{p(o_0 = 0)} \\
 &= p(o_1 = 1 | s_1 = 1) \cdot (p(s_1 = 1 | s_0 = 1) \cdot p(o_0 = 0 | s_0 = 1) \cdot p(s_0 = 1) / p(o_0 = 0)) \\
 &\quad + p(o_1 = 1 | s_1 = 1) \cdot (p(s_1 = 1 | s_0 = 0) \cdot p(o_0 = 0 | s_0 = 0) \cdot p(s_0 = 0) / p(o_0 = 0)) \\
 &\quad + p(o_1 = 1 | s_1 = 0) \cdot (p(s_1 = 0 | s_0 = 1) \cdot p(o_0 = 0 | s_0 = 1) \cdot p(s_0 = 1) / p(o_0 = 0)) \\
 &\quad + p(o_1 = 1 | s_1 = 0) \cdot (p(s_1 = 0 | s_0 = 0) \cdot p(o_0 = 0 | s_0 = 0) \cdot p(s_0 = 0) / p(o_0 = 0))
 \end{aligned}$$

$$p(o_1 = 1 | o_0 = 0) = (1 - p_S) \cdot p_{s_1|o_0=0} + p_G \cdot (1 - p_{s_1|o_0=0})$$



Inference in BKT models

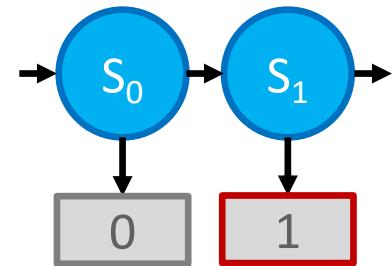
$$p(o_1 = 1 | o_0 = 1) = (1 - p_S) \cdot p_{s_1|o_0=1} + p_G \cdot (1 - p_{s_1|o_0=1})$$

$$p(o_1 = 0 | o_0 = 1) = p_S \cdot p_{s_1|o_0=1} + (1 - p_G) \cdot (1 - p_{s_1|o_0=1})$$



$$p(o_1 = 1 | o_0) = (1 - p_S) \cdot p_{s_1|o_0} + p_G \cdot (1 - p_{s_1|o_0})$$

$$p(o_1 = 0 | o_0) = p_S \cdot p_{s_1|o_0} + (1 - p_G) \cdot (1 - p_{s_1|o_0})$$



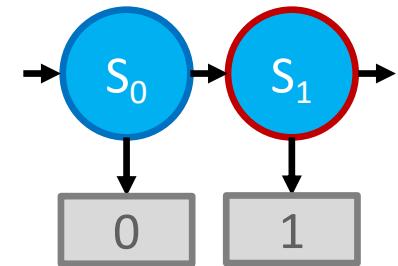
Inference in BKT models

$$p(s_1 = 1 | o_1 = 1, o_0) = \frac{p(s_1 = 1, o_1 = 1, o_0)}{p(o_1 = 1, o_0)}$$

$$\begin{aligned} p(o_1 = 1, o_0) &= p(o_1 = 1 | o_0) \cdot p(o_0) \\ &= ((1 - p_s) \cdot p_{s_1|o_0} + p_G \cdot (1 - p_{s_1|o_0})) \cdot p(o_0) \end{aligned}$$

$$\begin{aligned} p(s_1 = 1, o_1 = 1, o_0) &= p(o_1 = 1 | s_1 = 1) \cdot p(s_1 = 1 | s_0 = 1) \cdot p(o_0 | s_0 = 1) \cdot p(s_0 = 1) \\ &\quad + p(o_1 = 1 | s_1 = 1) \cdot p(s_1 = 1 | s_0 = 0) \cdot p(o_0 | s_0 = 0) \cdot p(s_0 = 0) \\ &= (1 - p_s) \cdot p_{s_1|o_0} \cdot p(o_0) \end{aligned}$$

$$p(s_1 = 1 | o_1 = 1, o_0) = \frac{(1 - p_s) \cdot p_{s_1|o_0}}{((1 - p_s) \cdot p_{s_1|o_0} + p_G \cdot (1 - p_{s_1|o_0}))}$$



Inference in BKT models

$$\mathbf{o}_{t-1} = [o_0, \dots, o_{t-1}]$$

Equations for time $t = 0$:

Belief about latent state before observation

$$p(s_0 = 1) = p_0$$

Predicted observation at time t

$$p(o_0 = 1) = (1 - p_s) \cdot p_0 + p_G \cdot (1 - p_0)$$

$$p(o_0 = 0) = p_s \cdot p_0 + (1 - p_G) \cdot (1 - p_0)$$

Posterior: belief about latent state after observation

$$p_{s_0|1} = \frac{(1 - p_s) \cdot p_0}{(1 - p_s) \cdot p_0 + p_G \cdot (1 - p_0)}$$

$$p_{s_0|0} = \frac{p_s \cdot p_0}{p_s \cdot p_0 + (1 - p_G) \cdot (1 - p_0)}$$

Equations for time steps $t = 1, \dots, T$:

$$p_{s_t|\mathbf{o}_{t-1}} = (1 - p_F) \cdot p_{s_{t-1}|\mathbf{o}_{t-1}} + p_L \cdot (1 - p_{s_{t-1}|\mathbf{o}_{t-1}})$$

$$p(o_t = 1|\mathbf{o}_{t-1}) = (1 - p_s) \cdot p_{s_t|\mathbf{o}_{t-1}} + p_G \cdot (1 - p_{s_t|\mathbf{o}_{t-1}})$$

$$p(o_t = 0|\mathbf{o}_{t-1}) = p_s \cdot p_{s_t|\mathbf{o}_{t-1}} + (1 - p_G) \cdot (1 - p_{s_t|\mathbf{o}_{t-1}})$$

$$p_{s_t|1,\mathbf{o}_{t-1}} = \frac{(1 - p_s) \cdot p_{s_t|\mathbf{o}_{t-1}}}{(1 - p_s) \cdot p_{s_t|\mathbf{o}_{t-1}} + p_G \cdot (1 - p_{s_t|\mathbf{o}_{t-1}})}$$

$$p_{s_t|0,\mathbf{o}_{t-1}} = \frac{p_s \cdot p_{s_t|\mathbf{o}_{t-1}}}{p_s \cdot p_{s_t|\mathbf{o}_{t-1}} + (1 - p_G) \cdot (1 - p_{s_t|\mathbf{o}_{t-1}})}$$

Making predictions using a BKT model

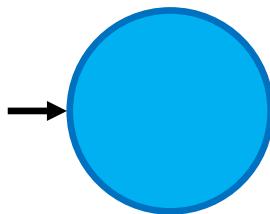
$$p_0 = 0.5$$

$$p_S = 0.2$$

$$p_G = 0.3$$

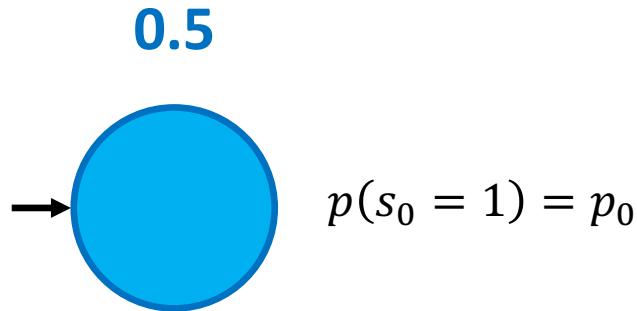
$$p_L = 0.4$$

$$p_F = 0.0$$



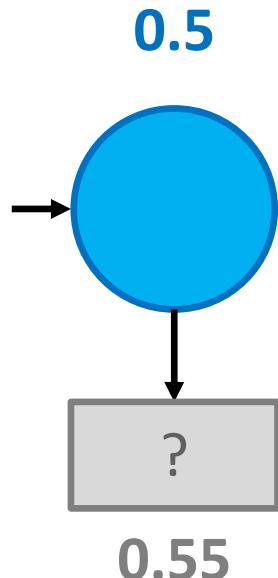
Making predictions using a BKT model

$p_0 = 0.5$
 $p_S = 0.2$
 $p_G = 0.3$
 $p_L = 0.4$
 $p_F = 0.0$



Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$

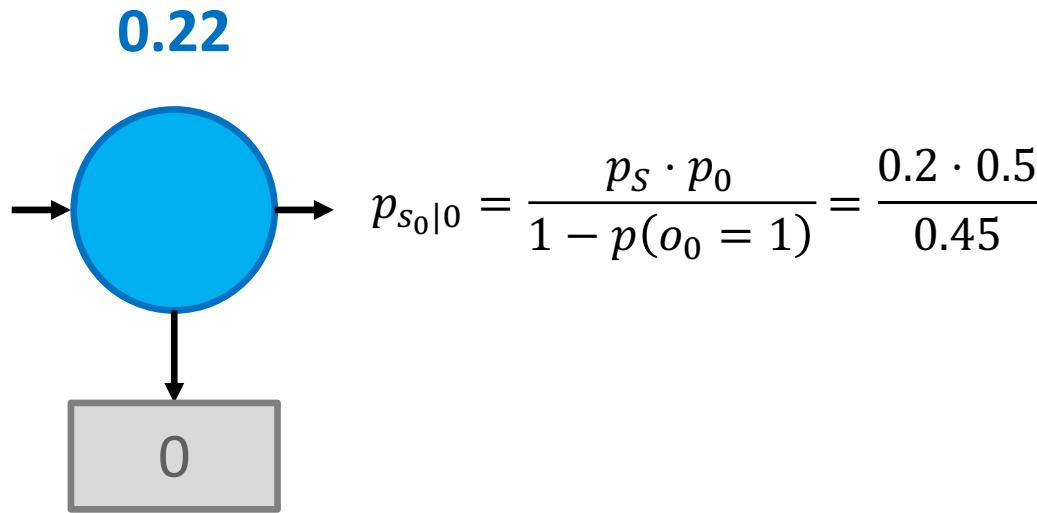


$$p(s_0 = 1) = p_0$$

$$p(o_0 = 1) = (1 - p_s) \cdot p_0 + p_G \cdot (1 - p_0)$$

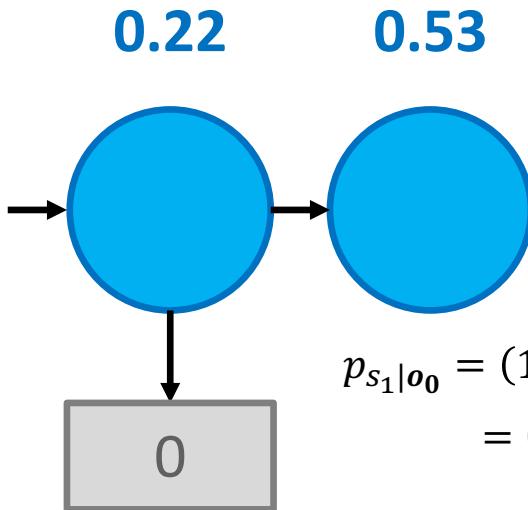
Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



Making predictions using a BKT model

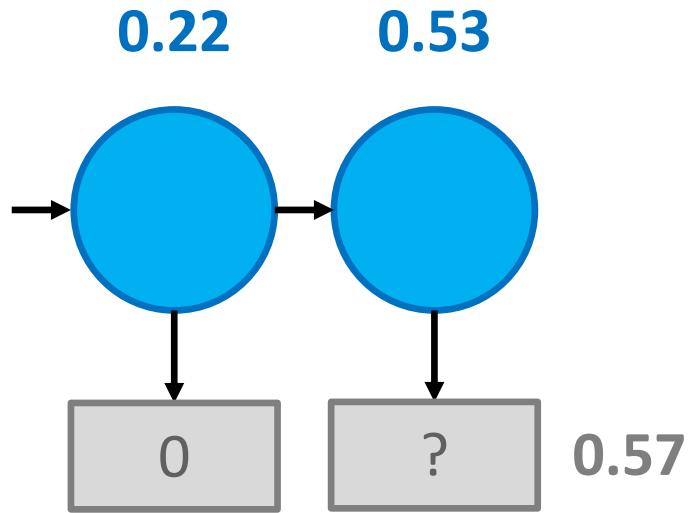
$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



$$\begin{aligned} p_{S_1|o_0} &= (1 - p_F) \cdot p_{S_1|o_0} + p_L \cdot (1 - p_{S_1|o_0}) \\ &= 0.22 + 0.4 \cdot 0.78 \end{aligned}$$

Making predictions using a BKT model

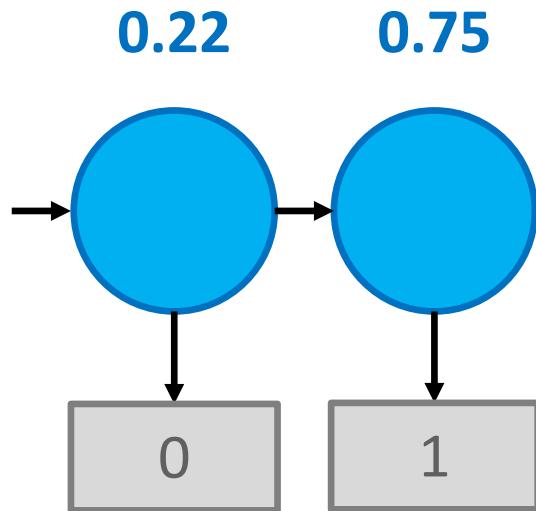
$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



$$\begin{aligned} p(o_1 = 1 | o_0) &= (1 - p_S) \cdot p_{s_1|o_0} + p_G \cdot (1 - p_{s_1|o_0}) \\ &= 0.8 \cdot 0.53 + 0.3 \cdot 0.47 \end{aligned}$$

Making predictions using a BKT model

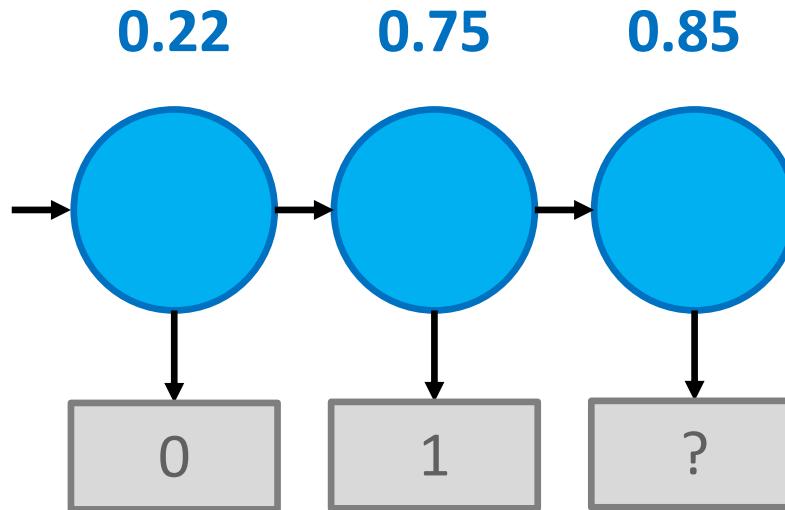
$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



$$\begin{aligned} p_{s_1|1,o_0} &= \frac{(1 - p_s) \cdot p_{s_1|o_0}}{(1 - p_S) \cdot p_{s_1|o_0} + p_G \cdot (1 - p_{s_1|o_0})} \\ &= \frac{0.8 \cdot 53}{0.57} \end{aligned}$$

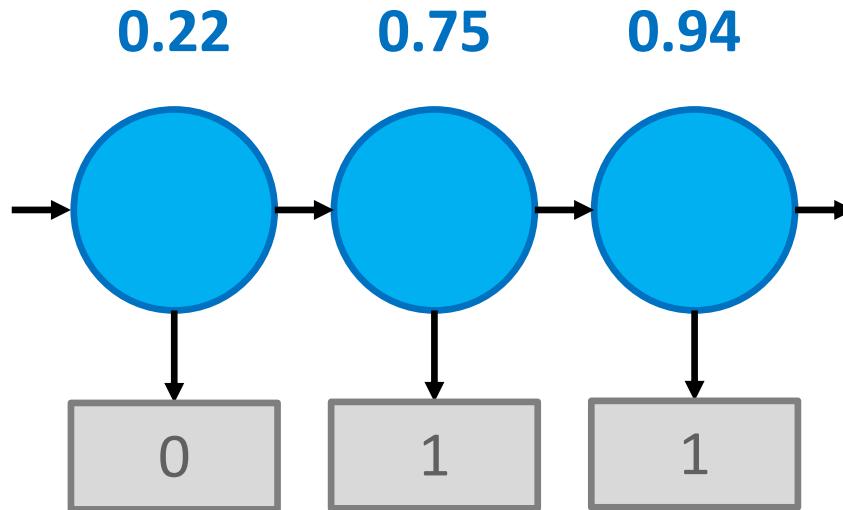
Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



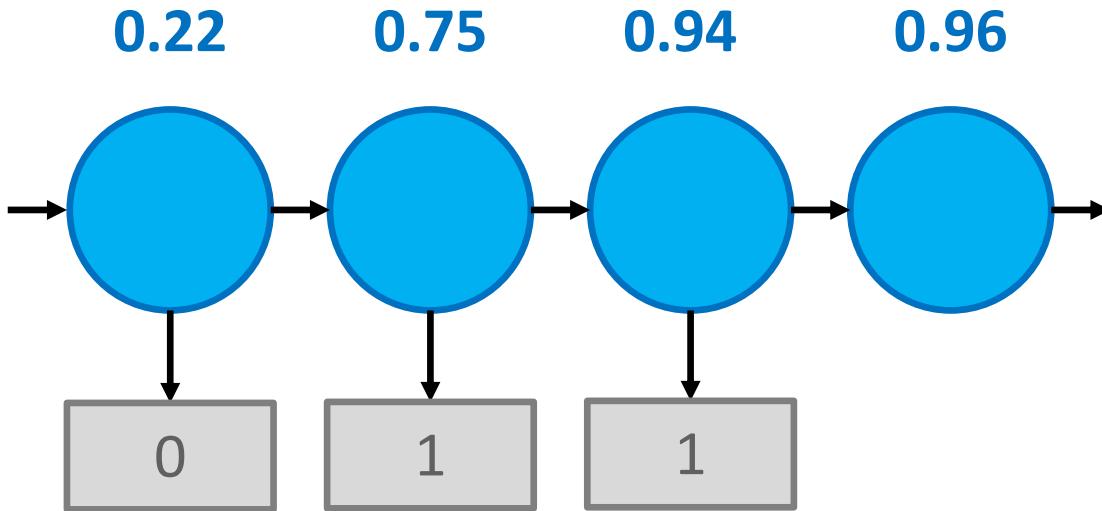
Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



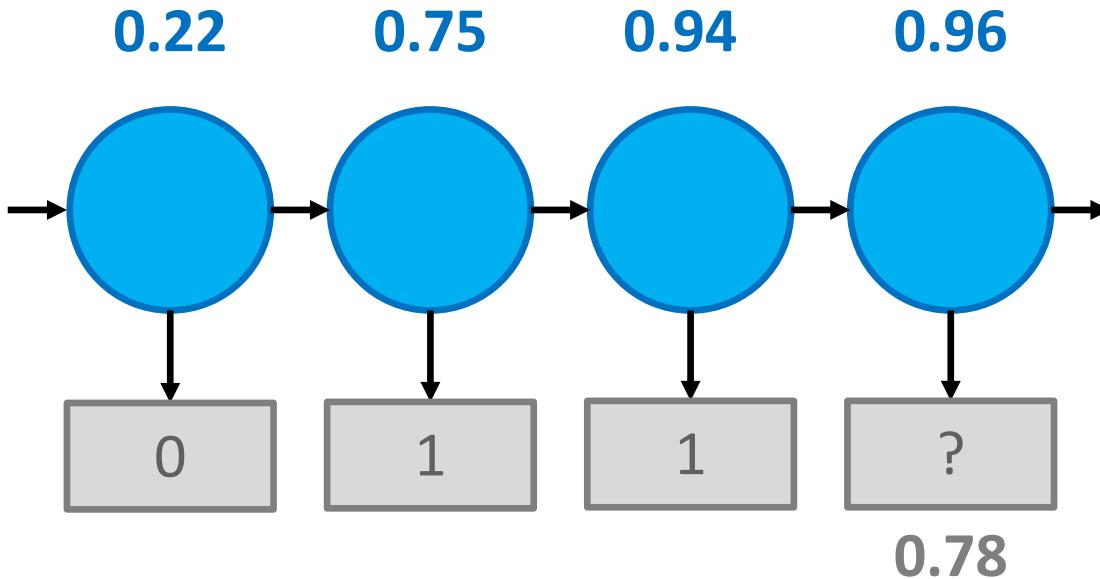
Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



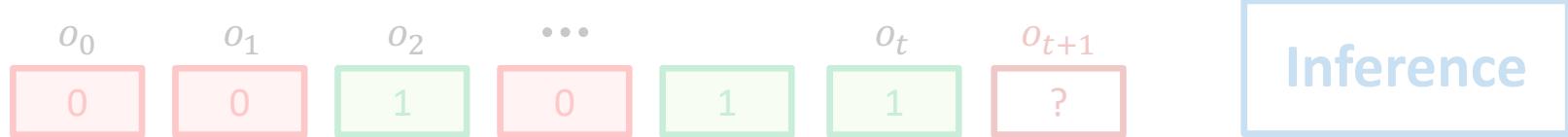
Making predictions using a BKT model

$$\begin{aligned} p_0 &= 0.5 \\ p_S &= 0.2 \\ p_G &= 0.3 \\ p_L &= 0.4 \\ p_F &= 0.0 \end{aligned}$$



Two tasks need to be solved in practice

- Given a model with parameters $\theta = \{p_0, p_L, p_F, p_S, p_G\}$ and a sequence of observations $\mathbf{o} = [o_0, \dots, o_t]$ from a student s , predict o_{t+1}



- Given sequences of observations $\mathbf{o} = [o_0, \dots, o_T]$ of N students, learn the parameters $\theta = \{p_0, p_L, p_F, p_S, p_G\}$ that maximize the likelihood of the observed data

Student 1:

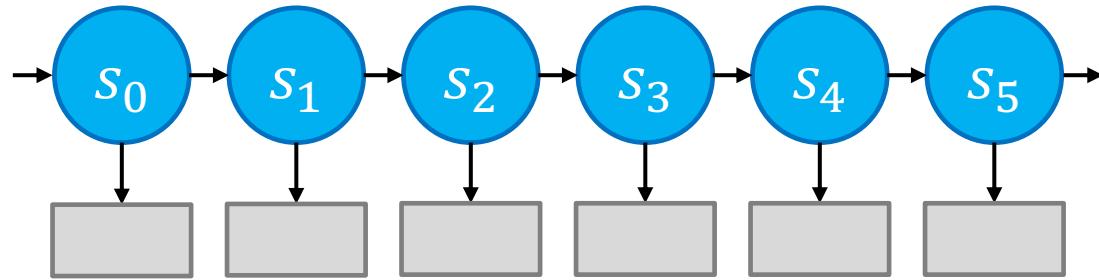
Student 2:

⋮

Student N :

Parameter
Learning

Training a BKT model



$$\theta = \{p_0, p_L, p_F, p_S, p_G\}$$

Student l_0 : $\mathbf{o}_{l_0} = [0,1,1]$

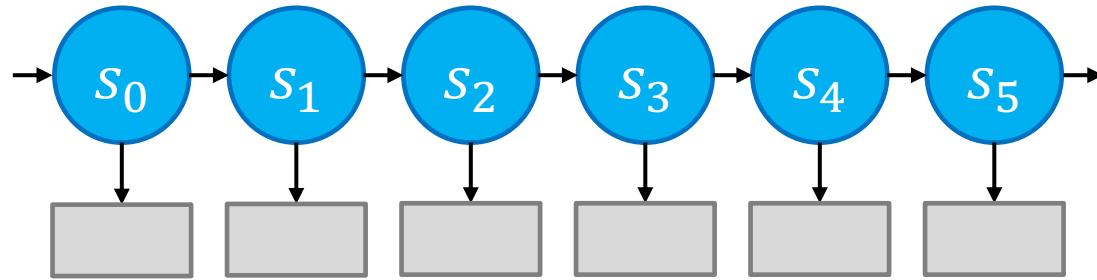
⋮

Student l_{N-1} : $\mathbf{o}_{l_{N-1}} = [1,0,1,1,1,0,0,1,1,1]$

Student l_N : $\mathbf{o}_{l_N} = [0,1,0,1]$

$$\max_{\theta} p(\mathbf{o}_{l_0}, \dots, \mathbf{o}_{l_{N-1}}, \mathbf{o}_{l_N})$$

Training a BKT model



$$\theta = \{p_0, p_L, p_F, p_S, p_G\}$$

Student l_0 : $\mathbf{o}_{l_0} = [0,1,1]$

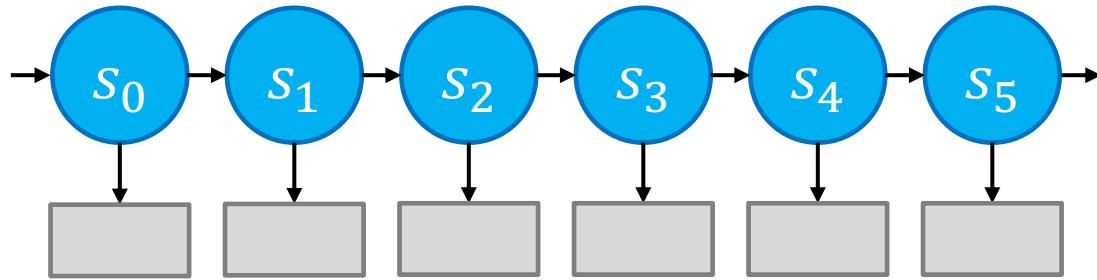
⋮

Student l_{N-1} : $\mathbf{o}_{l_{N-1}} = [1,0,1,1,1,0,0,1,1,1]$

Student l_N : $\mathbf{o}_{l_N} = [0,1,0,1]$

$$\max_{\theta} \prod_{i=1}^N p(\mathbf{o}_{l_i})$$

Training a BKT model



$$\theta = \{p_0, p_L, p_F, p_S, p_G\}$$

Student l_0 : $p(\mathbf{o}_{l_0}) = \sum_s p(\mathbf{o}_{l_0}, s)$

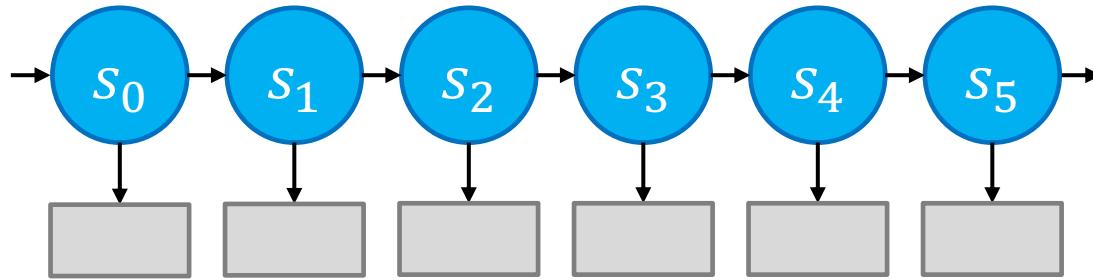
⋮

Student l_{N-1} : $p(\mathbf{o}_{l_0}) = \sum_s p(\mathbf{o}_{l_0}, s)$

Student l_N : $p(\mathbf{o}_{l_0}) = \sum_s p(\mathbf{o}_{l_0}, s)$

$$\max_{\theta} \prod_{i=1}^N \sum_s p(\mathbf{o}_{l_i}, s_{l_i})$$

Training a BKT model



$$\theta = \{p_0, p_L, p_F, p_S, p_G\}$$

$$\max_{\theta} \prod_{i=1}^N \sum_{s_{l_i}} p(\mathbf{o}_{l_i}, s_{l_i}) \quad \rightarrow \quad \min_{\theta} - \sum_{i=1}^N \log \left(\sum_{s_{l_i}} p(\mathbf{o}_{l_i}, s_{l_i}) \right)$$

- Brute-Force Grid Search
- Expectation Maximization
- Gradient Descent
- Nelder-Mead Optimization

Your Turn – Evaluating a BKT model

- In the student notebook, you have:
 - A trained BKT model for six selected skills
 - A data frame containing the predictions of the BKT model for each observation in the test set
 - Your task:
 - Compute the RMSE or AUC separately for each skill
 - Provide a visualization of the mean RMSE (or AUC) + standard deviation over all skills as well as the per skill RMSE (or AUC)
-

Summary – Why tracing knowledge?

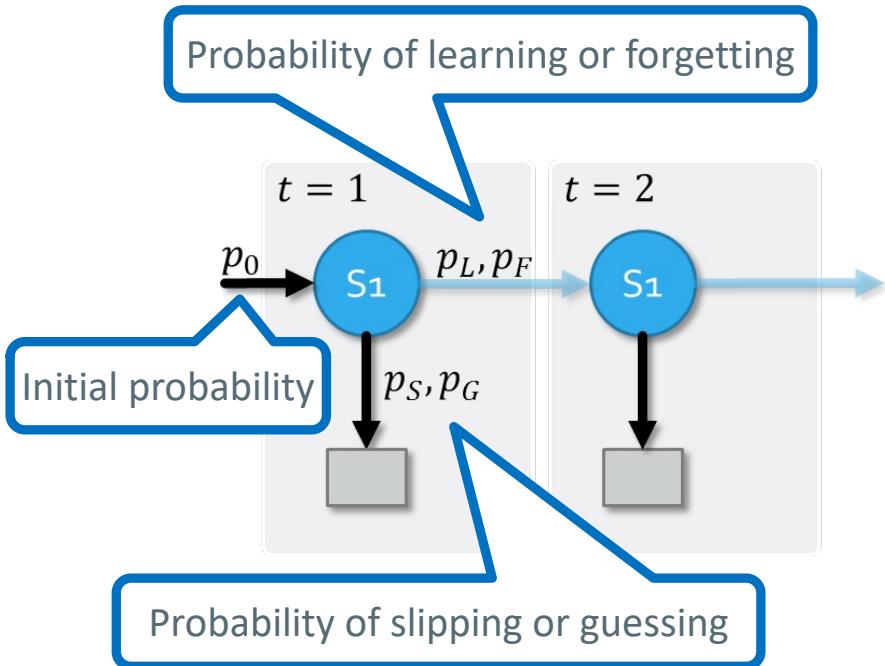
- Is the student learning?
 - Measure what the student *knows* at a specific time t
 - More specifically: knowledge of the student about relevant knowledge components (skills)

→ Choose the next appropriate activity

→ Know which activities support learning



Summary - BKT



- Predict $p(o_{i_{s1},t} | o_0, \dots, o_{t-1})$, the probability that the student will solve task i_{s1} correctly at time step t
- Predict $p(s_{1,t} | o_0, \dots, o_{t-1})$, the probability that the student has mastered skill s_1 at time step t

Summary - Assumptions behind BKT

- Knowledge can be divided into different skills
 - Definition of skills is accurate/detailed enough
 - Each task corresponds to a single skill (original)
 - There is **no** connection between the skills
 - Mastery can be achieved through practice
 - There is no forgetting: $p_F = 0$ (original)
-

Knowledge Tracing - Continued

Machine Learning for Behavioral Data
April 3, 2023

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Knowledge Tracing
7	Knowledge Tracing
8	Spring Break

Supervised learning on time series:

- Probabilistic graphical models
- Neural networks: LSTM, GRU, etc.

Agenda

- Short quiz about the past...
- Learning Curves
- Alternative approaches to knowledge tracing
- (Voluntary) participation in user study
- Lab session:
 - Easter Quiz!  *Win a chocolate Easter bunny!*
 - Practice on knowledge tracing



Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

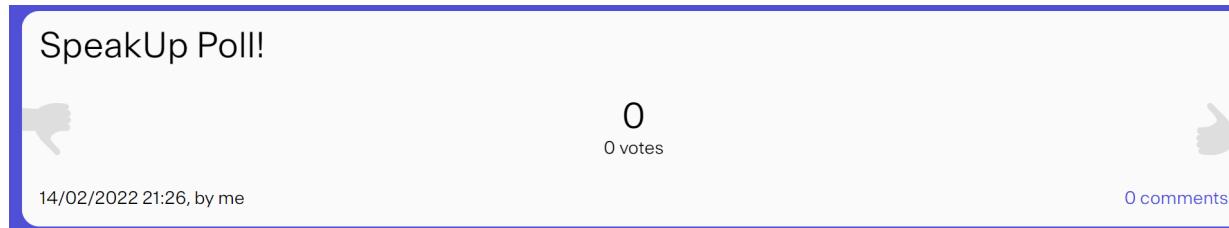
<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

[KT] BKT does account for students guessing the correct answer.

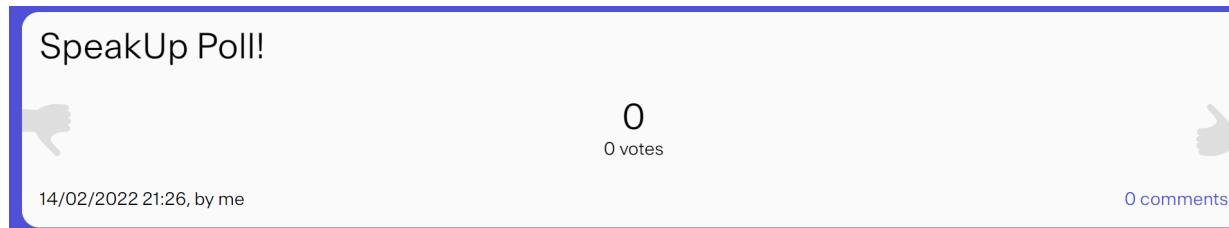
- a) True
- b) False



Short quiz about the past...

[KT] BKT can represent the relationships between different skills.

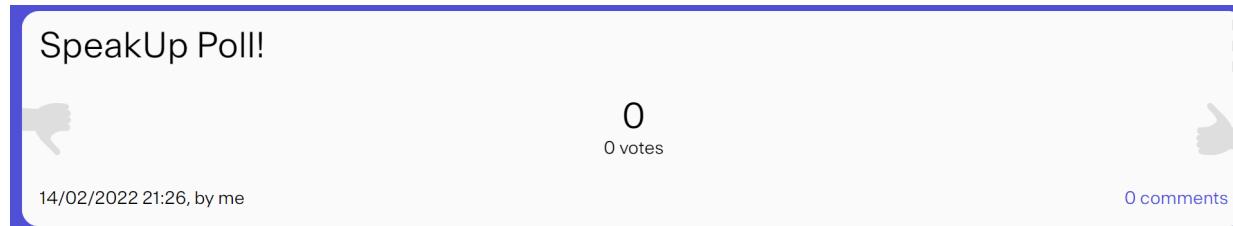
- a) True
- b) False



Short quiz about the past...

[Mixed Models] Mixed-effect models are useful when the samples in the data set are uncorrelated.

- a) True
- b) False



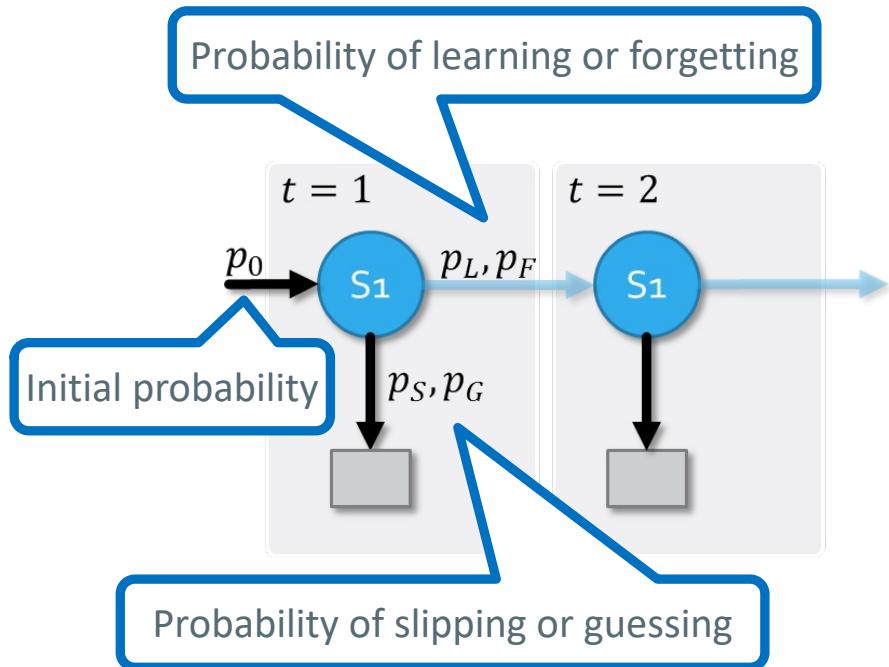
Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	Model Evaluation
6	Knowledge Tracing
7	Knowledge Tracing
8	Spring Break

Supervised learning on time series:

- Probabilistic graphical models
- Neural networks: LSTM, GRU, etc.

Last Week: Bayesian Knowledge Tracing



- Predict $p(o_{i_{s1},t} | o_0, \dots, o_{t-1})$, the probability that the student will solve task i_{s1} correctly at time step t
- Predict $p(s_{1,t} | o_0, \dots, o_{t-1})$, the probability that the student has mastered skill s_1 at time step t

Assumptions behind BKT

- Knowledge can be divided into different skills
 - Definition of skills is accurate/detailed enough
 - Each task corresponds to a single skill (original)
 - There is **no** connection between the skills
 - Mastery can be achieved through practice
 - There is no forgetting: $p_F = 0$ (original)
-

Today

- **Learning Curves**
- Alternative Models for Knowledge Tracing
 - AFM
 - PFA



Today's Use Case

- ASSISTments is a free tool for assigning and assessing math problems and homework
 - All math problems (tasks/items) are associated to a specific skill/knowledge component
 - 4,151 middle-school students
 - 525,534 observations
-

Tracing Knowledge – why is it useful?

- Is the student learning?
 - Measure what the student *knows* at a specific time t
 - More specifically: knowledge of the student about relevant knowledge components (skills)

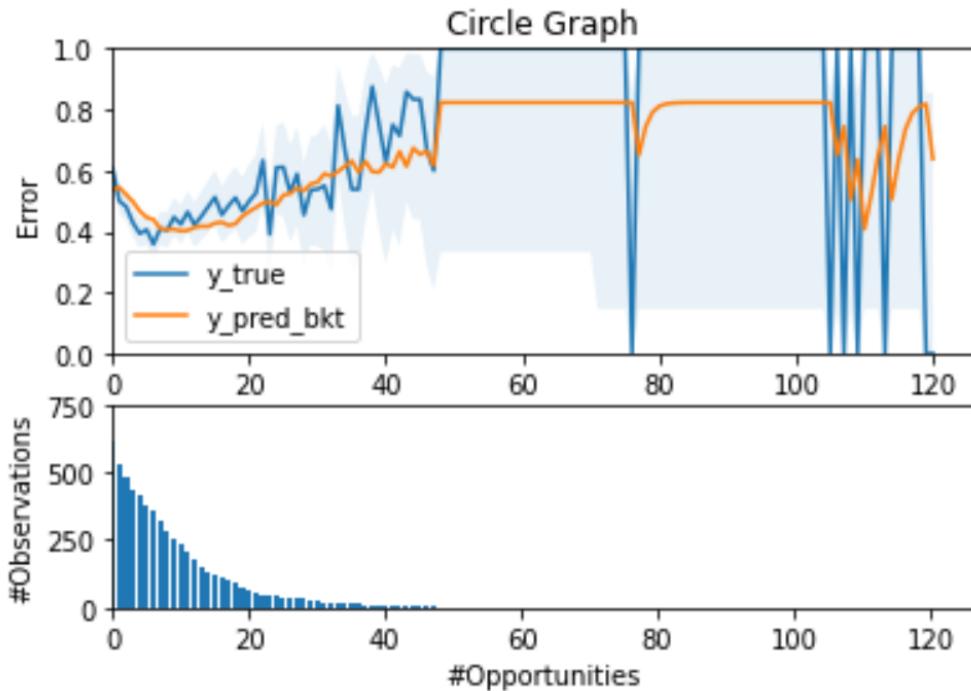
→ Choose the next appropriate activity

→ Know which activities support learning

Building a learning curve for skill s

Student	Opportunity	y_true	y_pred
0	0	0	0.3
0	1	0	0.5
0	2	1	0.7
0	3	1	0.9
1	0	0	0.3
1	1	1	0.5
2	0	0	0.3
2	1	1	0.5
2	2	1	0.7
3	0	1	0.3
3	1	0	0.7
3	2	1	0.5
3	3	1	0.9

What could this curve indicate?



Your Turn – Learning Curves

- In the student notebook, you have:
 - BKT model trained on all skills and students
 - List of available skills
 - Function for plotting learning curves and student numbers for a specific skill
 - Your task:
 - Pick 1-2 skills, generate the learning curves for them, and interpret them
 - Send us your plots and interpretations
-

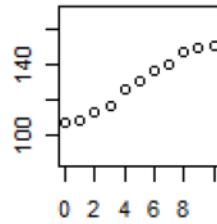
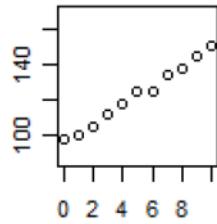
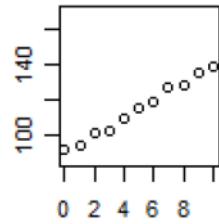
Today

- Learning Curves
- Alternative Models for Knowledge Tracing
 - AFM
 - PFA



Generalized Linear Mixed Effects Models revisited

- Example: strength gain by weight training
 - Each person has individual starting strength



$$y_n = \beta_0 + u_n + \beta_1 x_{n,1}$$

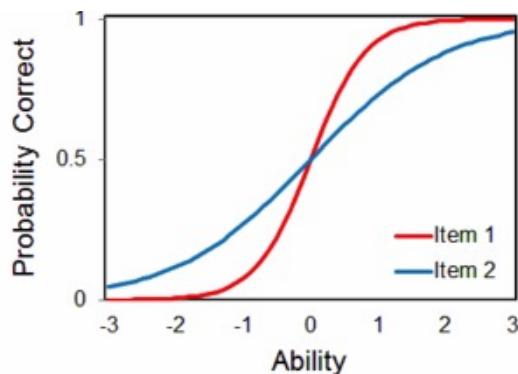
“Fixed” Effects

“Random” Effect

Rasch Model

$$\log\left(\frac{p_{i,n}}{1 - p_{i,n}}\right) = \theta_n - b_i$$

Probability that student n will solve item i correctly.



θ_n : student ability

b_i : difficulty of item i

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

Probability that student n will solve task i correctly.

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$



Student proficiency

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$

Student proficiency

$q_{ik} = 1$, if item i uses skill k

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$

Student proficiency

Difficulty of
skill k

$q_{ik} = 1$, if item i uses skill k

Additive Factors Model (AFM)

$$p_{n,i} = \frac{1}{1 + e^{-\pi_{n,i}}}$$

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$

Student proficiency

$q_{ik} = 1$, if item i uses skill k

Difficulty of
skill k

Learning rate
at skill k

Number of practice
opportunities
student n had at
skill k

AFM - Assumptions

- Students may initially know more or less
 - Students learn at the same rate
 - Some skills are more likely to initially be known
 - Some skills are easier to learn than others
 - Students learn with each practice opportunity
 - Each item belongs to one or more skills
-

Performance Factors Analysis (PFA)

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot T_{n,k})$$

Performance Factors Analysis (PFA)

$$\pi_{n,i} = \theta_n + \sum_k q_{i,k} \cdot (\beta_k + \gamma_k \cdot s_{n,k} + \rho_k \cdot f_{n,k})$$

Number of prior
successes student
n had at skill k

Number of prior
failures student n
had at skill k

PFA - Assumptions

- Students may initially know more or less
 - Students learn at the same rate
 - Some skills are more likely to initially be known
 - Some skills are easier to learn than others
 - Students learning rate differs for correct and wrong practice opportunities
 - Each item belongs to one or more skills
-

AFM/PFA in action...

➡ Jupyter Notebook

Cheat sheet for mixed effect models:

<https://go.epfl.ch/mlbd-mixed-effects>

Your Turn: Comparing Models

- We have evaluated AFM, PFA, and BKT on a subset of six skills. Your task:
 - Visualize the overall RMSE and AUC of the models such that it can easily be compared
 - Discuss the obtained results



Summary

- Learning Curves
- Alternative Models for Knowledge Tracing:
 - AFM
 - PFA

Final Project Presentations

- Poster Session
- May 22, 15.15-18.00 (location: BC Atrium)
- **Mandatory** presence of all team members
- There will be prizes and snacks/drinks...

Easter Quiz – Join us on Kahoot!



[www.kahoot.it](https://wwwkahootit)

Enter the game pin!



Win a chocolate Easter bunny!



Recurrent Neural Networks

Machine Learning for Behavioral Data
April 17, 2023

Today's Topic

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Ethical Machine Learning
13	Ethical Machine Learning
14	Project Presentations
15	Whit Monday

Supervised learning on time series:

- Probabilistic graphical models
- Neural networks: LSTM, GRU, etc.

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

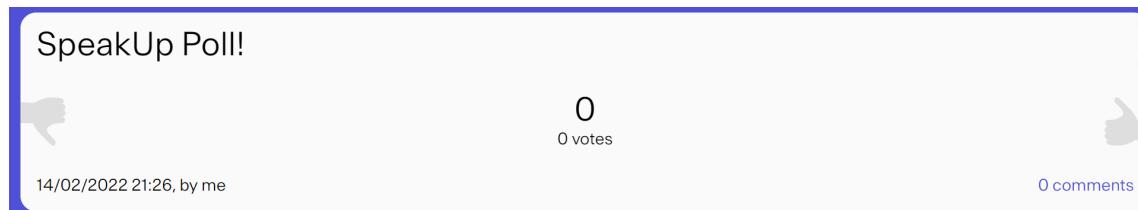
<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

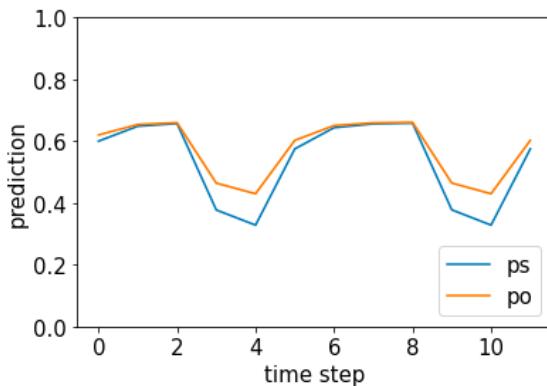
This KT model uses the # of opportunities the student had per skill and treats prior successes and failures the same.

- a) Additive Factors Model (AFM)
- b) Performance Factors Analysis (PFA)
- c) Bayesian Knowledge Tracing (BKT)

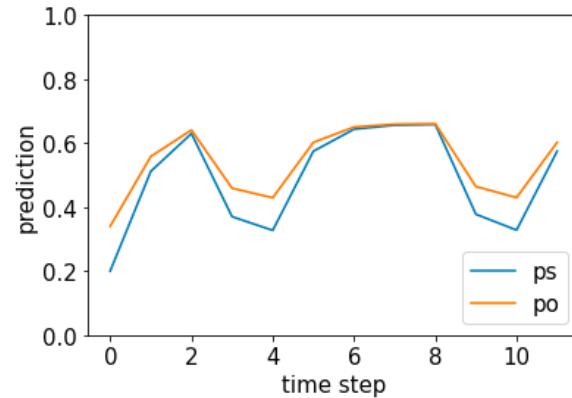


Short quiz about the past...

Which BKT parameter has been changed between the left and right plot (exactly one)?



$$\begin{aligned} p_0 &= 0.6, \\ p_s &= 0.1, \\ p_g &= 0.2, \\ p_l &= 0.3, \\ p_f &= 0.3 \end{aligned}$$



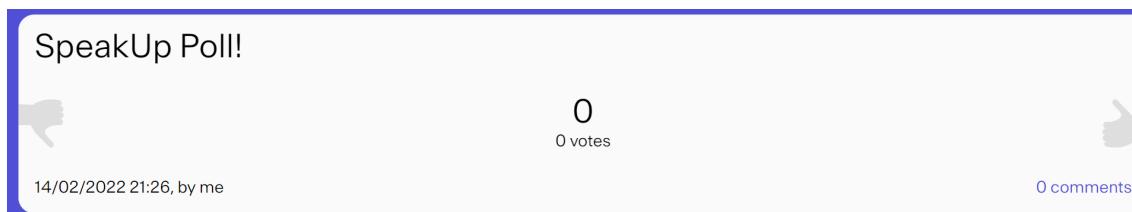
- a) p_g (guess probability)
- b) p_l (probability of learning)
- c) p_0 (initial probability)
- d) p_f (forget probability)



Short quiz about the past...

Which of the following statements about Pearson's correlation is true?

- a) If two variables X, Y have correlation = 0, then X, Y are dependent.
- b) If two variables X, Y have correlation = 0, then X, Y are independent.
- c) If X, Y are dependent variables, then their correlation = 0.
- d) If X, Y are independent variables, then their correlation = 0.



Knowledge Tracing – Predicting Future Performance



Subtraction 0-100

1

2

...

n

n+1

A rectangular box with a red border and a light pink background, containing the number "0" in a large, black font.A rectangular box with a red border and a light pink background, containing the number "0" in a large, black font.A rectangular box with a green border and a light green background, containing the number "1" in a large, black font.A rectangular box with a red border and a light pink background, containing the number "0" in a large, black font.A rectangular box with a green border and a light green background, containing the number "1" in a large, black font.A rectangular box with a green border and a light green background, containing the number "1" in a large, black font.A rectangular box with a red border and a white background, containing a large question mark "?" in a black font.

Today – Recurrent Neural Networks

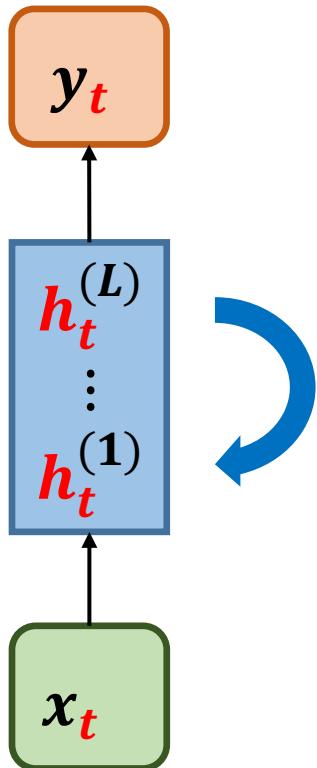
- Deep Knowledge Tracing
- Parameters and hyperparameter tuning
- Different architectures
- Different tasks:
 - “Many-to-many” versus “Many-to-one”
 - Classification versus Regression

Neural Networks

- Neural networks are able to represent non-linear functions, i.e. $y_n \approx f(x_n)$ can be non-linear
- Neural networks are able to *learn* the features and the weights (parameters) from the data
- Tutorial: <https://go.epfl.ch/tutorial-nn>

Recurrent Neural Network

Output Layer



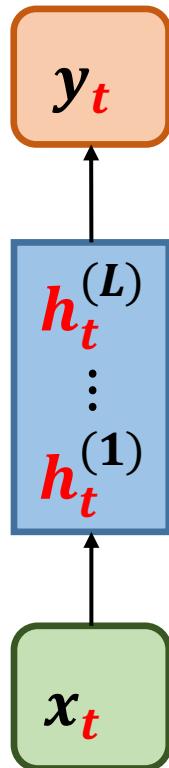
Hidden Layer(s)

Input Layer

$$h_t^{(1)} = \phi_h (W_{hx}x_t + W_{h^{(1)}h^{(1)}}h_{t-1}^{(1)} + b^{(1)})$$

Recurrent Neural Network

Output Layer



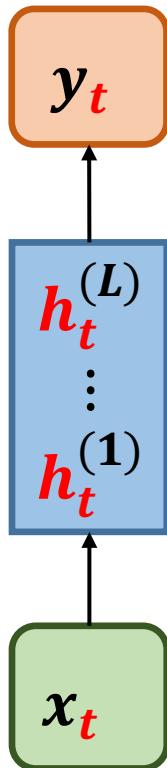
Hidden Layer(s)

Input Layer

$$h_t^{(l)} = \phi_h(W_{h^{(l)} h^{(l-1)}} h_t^{(l-1)} + W_{h^{(l)} h^{(l)}} h_{t-1}^{(l)} + b^{(l)})$$
$$h_t^{(1)} = \phi_h (W_{hx} x_t + W_{h^{(1)} h^{(1)}} h_{t-1}^{(1)} + b^{(1)})$$

Recurrent Neural Network

Output Layer



Hidden Layer(s)

Input Layer

$$y_t = \phi_y(W_{yh} h_t^L + b^{(y)})$$

$$h_t^{(l)} = \phi_h(W_{h^{(l)} h^{(l-1)}} h_t^{(l-1)} + W_{h^{(l)} h^{(l)}} h_{t-1}^{(l)} + b^{(l)})$$

$$h_t^{(1)} = \phi_h (W_{hx} x_t + W_{h^{(1)} h^{(1)}} h_{t-1}^{(1)} + b^{(1)})$$

Deep Knowledge Tracing

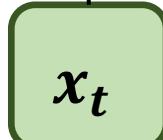
Output Layer



Hidden Layer(s)

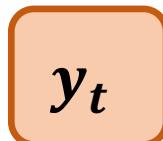


Input Layer



Deep Knowledge Tracing

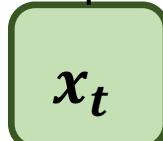
Output Layer



Hidden Layer(s)



Input Layer



Skills or
Items

0	0
1	1
2	0
0	0
1	0
2	0

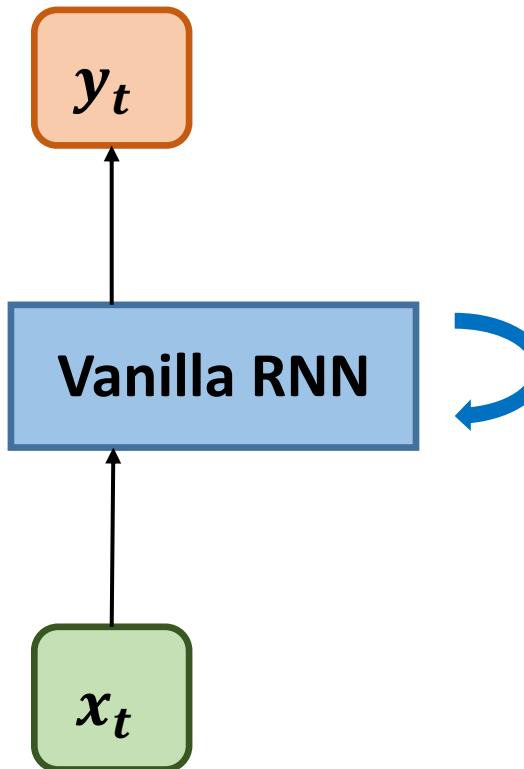
Input: one-hot encoded
observation at time step t

Deep Knowledge Tracing

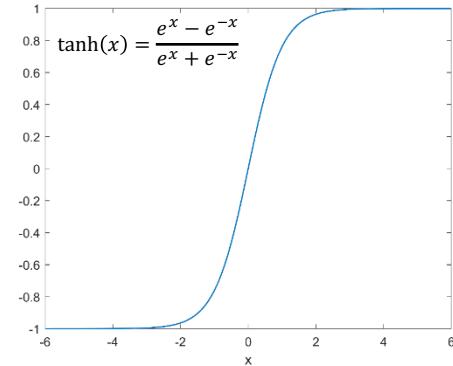
Output Layer

Hidden Layer(s)

Input Layer

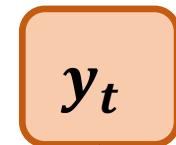


$$h_t = \tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$



Deep Knowledge Tracing

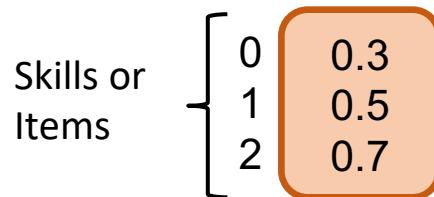
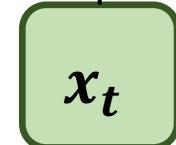
Output Layer



Hidden Layer(s)

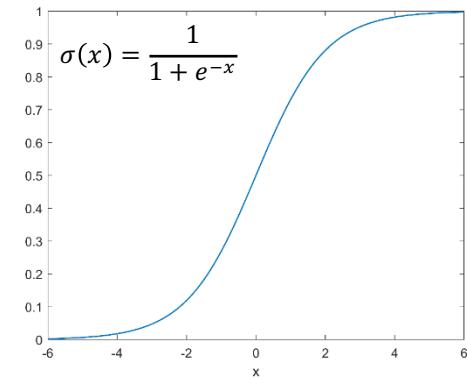


Input Layer

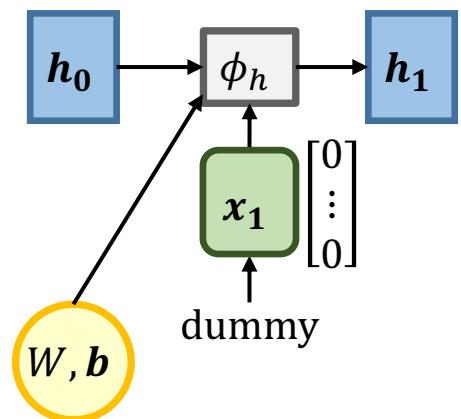


Output: probability for
answering skill (or item)
correct at time step t

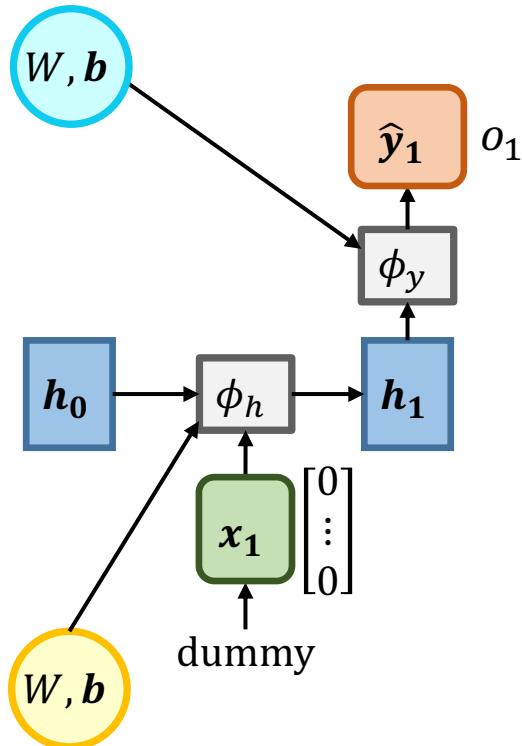
$$y_t = \sigma(W_{yh}h_t + b_y)$$



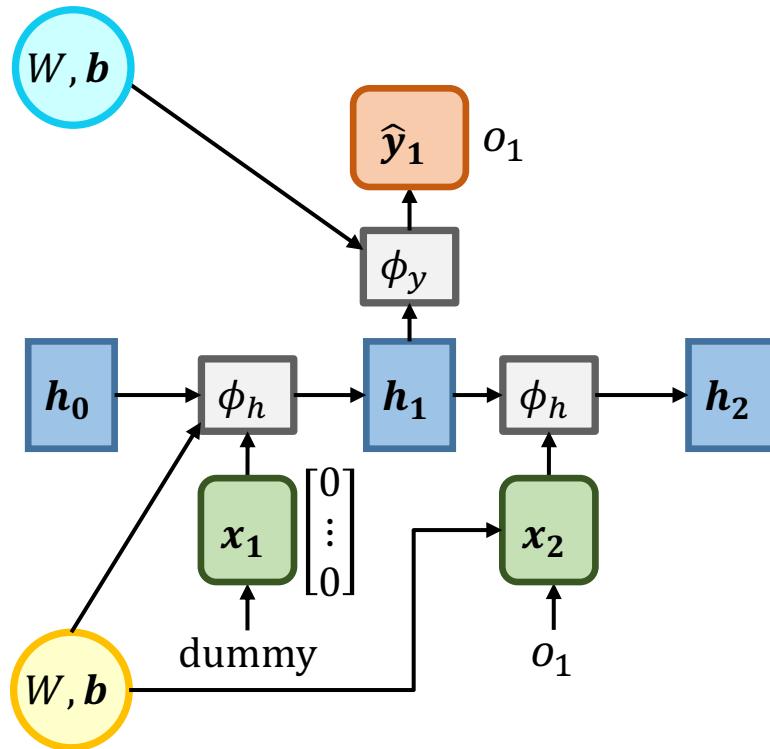
Deep Knowledge Tracing – Computational Graph



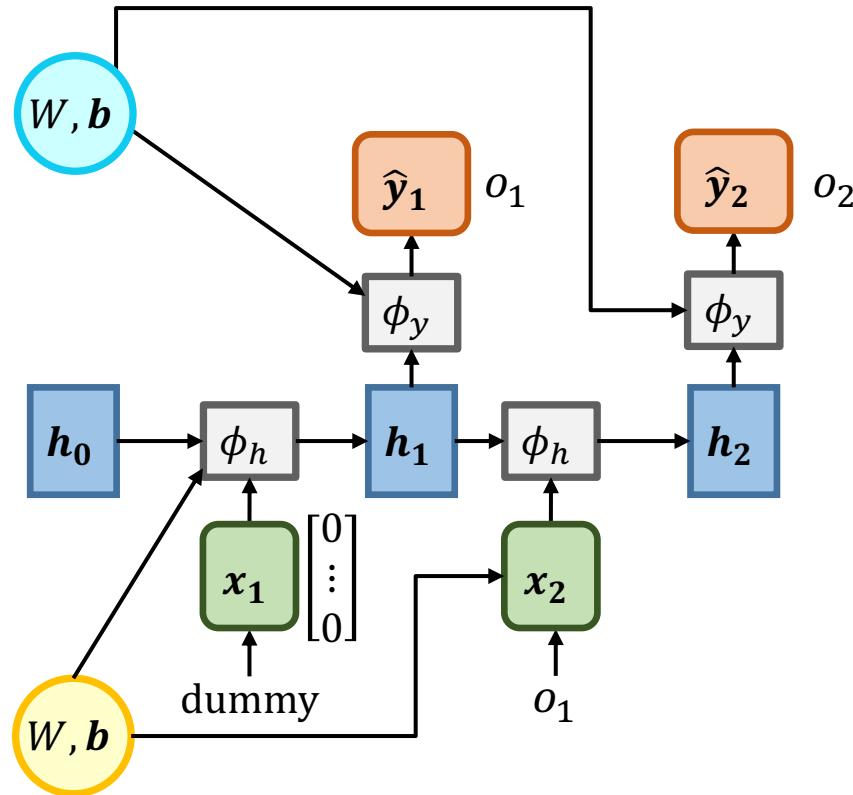
Deep Knowledge Tracing – Computational Graph



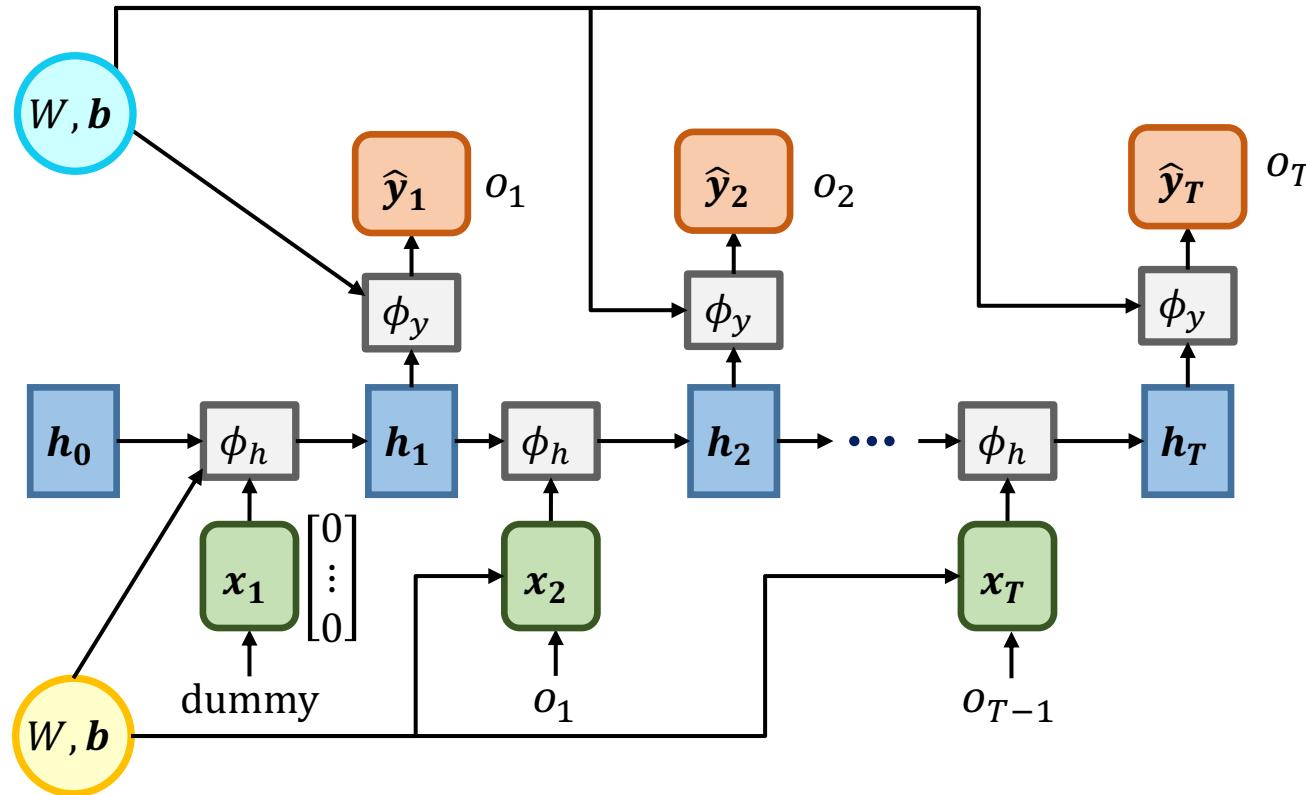
Deep Knowledge Tracing – Computational Graph



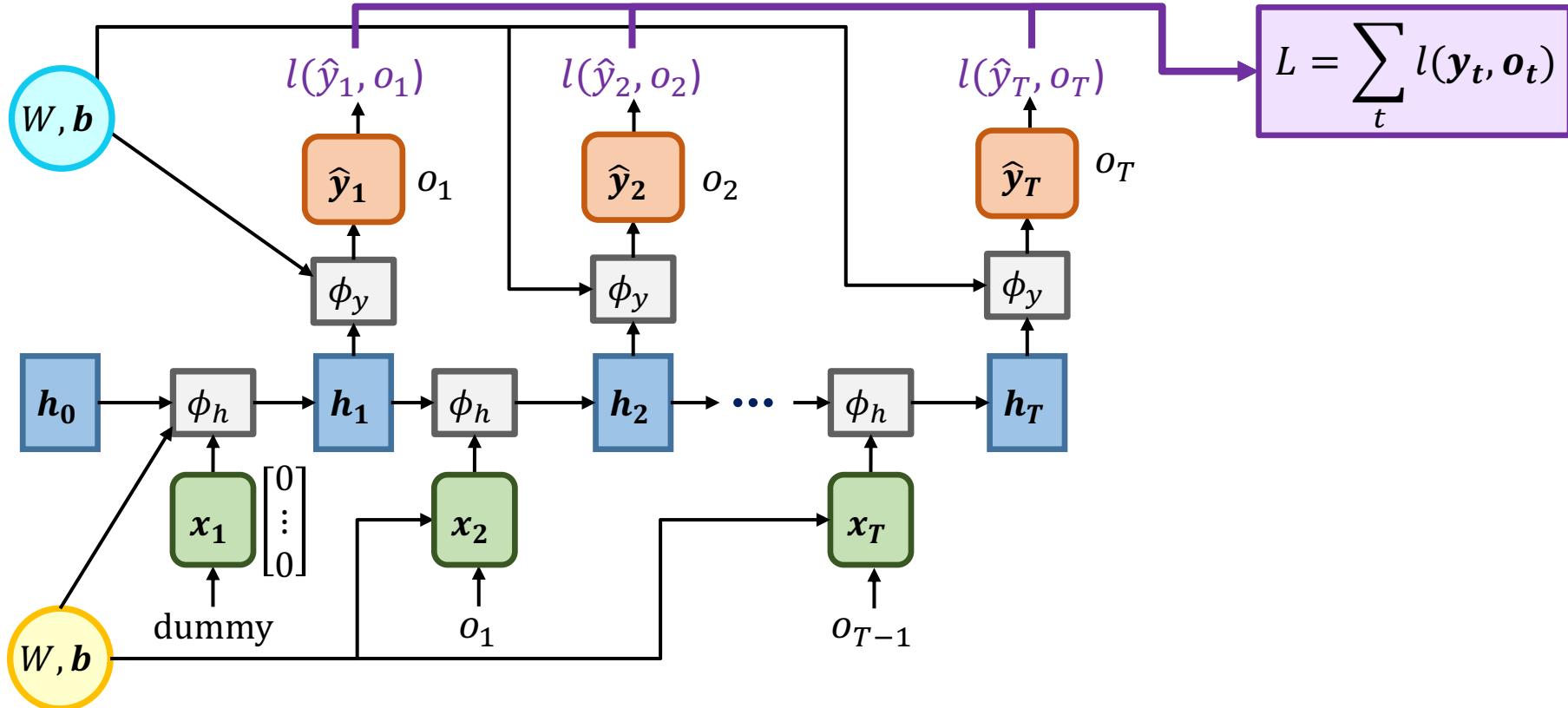
Deep Knowledge Tracing – Computational Graph



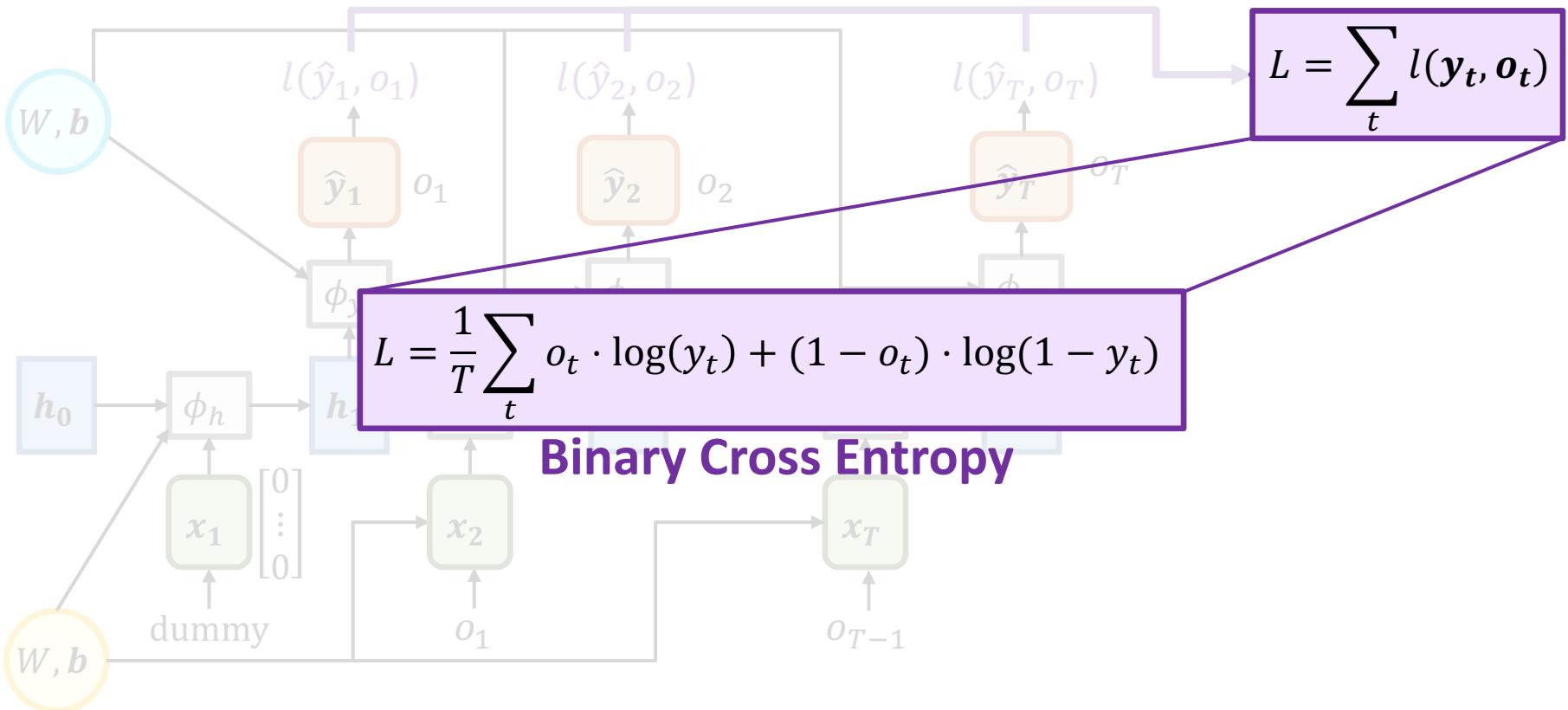
Deep Knowledge Tracing – Computational Graph



Deep Knowledge Tracing – Computational Graph



Training a DKT model: Binary Crossentropy Loss



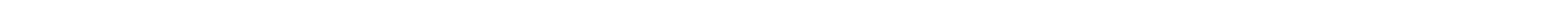
Training and Prediction using DKT

- Training: gradient descent
- Prediction: compute inference in the network (see computational graph)



Today – Recurrent Neural Networks

- Deep Knowledge Tracing
- **Parameters and hyperparameter tuning**
- Different architectures
- Different tasks:
 - “Many-to-many” versus “Many-to-one”
 - Classification versus Regression



RNNs – Specifying Parameters

```
[ ] # Specify the model hyperparameters. Full descriptions included in the demo notebook!
params = {}

params['batch_size'] = 32
params['mask_value'] = -1.0
params['verbose'] = 1
params['best_model_weights'] = 'weights/bestmodel'
params['optimizer'] = 'adam'
params['recurrent_units'] = 16
params['epochs'] = 20
params['dropout_rate'] = 0.1
```

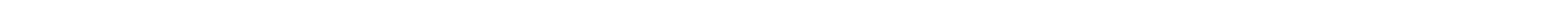
RNNs – Tuning hyperparameters

```
[ ] # Specify the model hyperparameters. Full descriptions included in the demo notebook!
params = {}

params['batch_size'] = 32
params['mask_value'] = -1.0
params['verbose'] = 1
params['best_model_weights'] = 'weights/bestmodel'
params['optimizer'] = 'adam'
params['recurrent_units'] = 16
params['epochs'] = 20
params['dropout_rate'] = 0.1
```

RNNs – Tuning hyperparameters

- Optimal number of epochs can be found using callbacks
- Other parameters can be tuned using for example:
 - a) Train-Validation-Test split
 - b) Train-Test split, using a k-fold cross validation on the training data to determine the optimal parameters

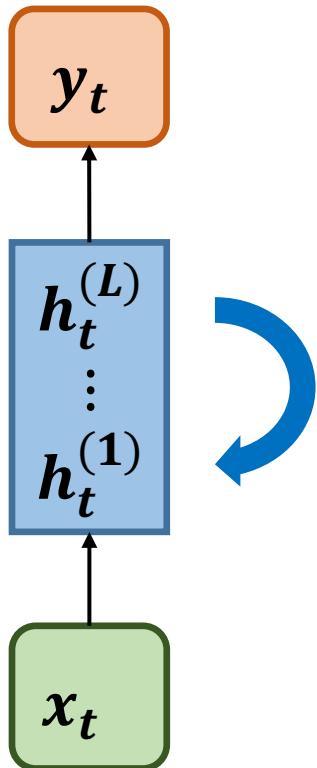


Today – Recurrent Neural Networks

- Parameters and hyperparameter tuning
- **Different architectures**
- Different tasks:
 - “Many-to-many” versus “Many-to-one”
 - Classification versus Regression

Recurrent Neural Network

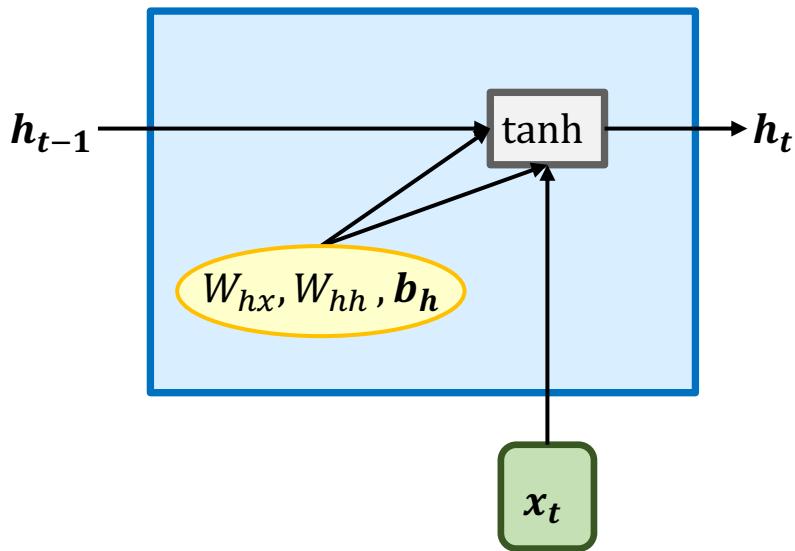
Output Layer



Hidden Layer(s)

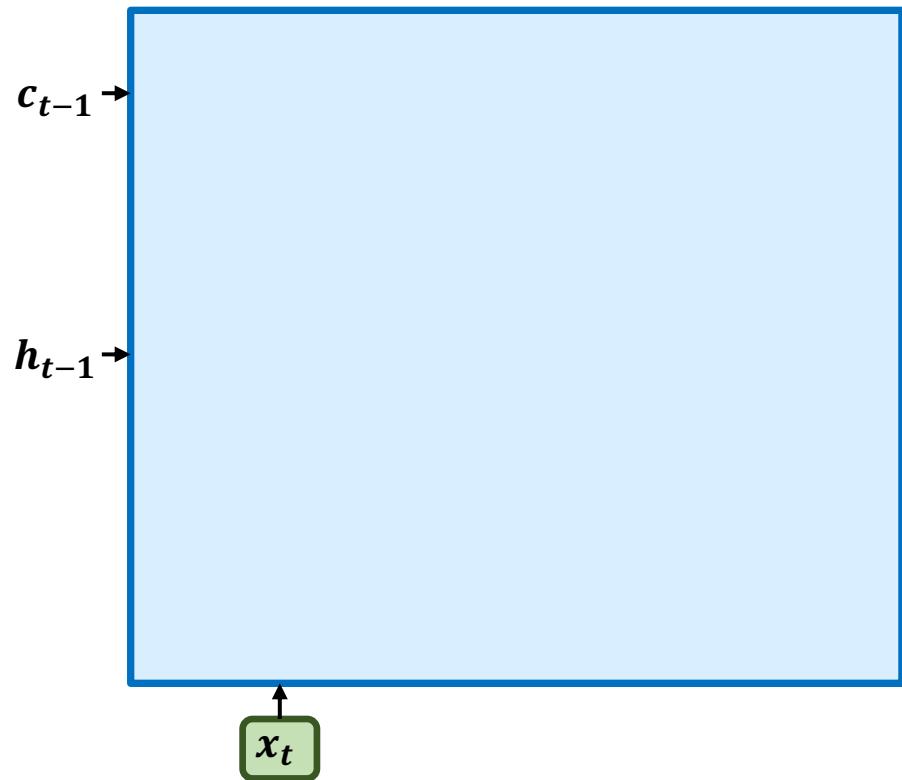
Input Layer

Vanilla RNN - revisited



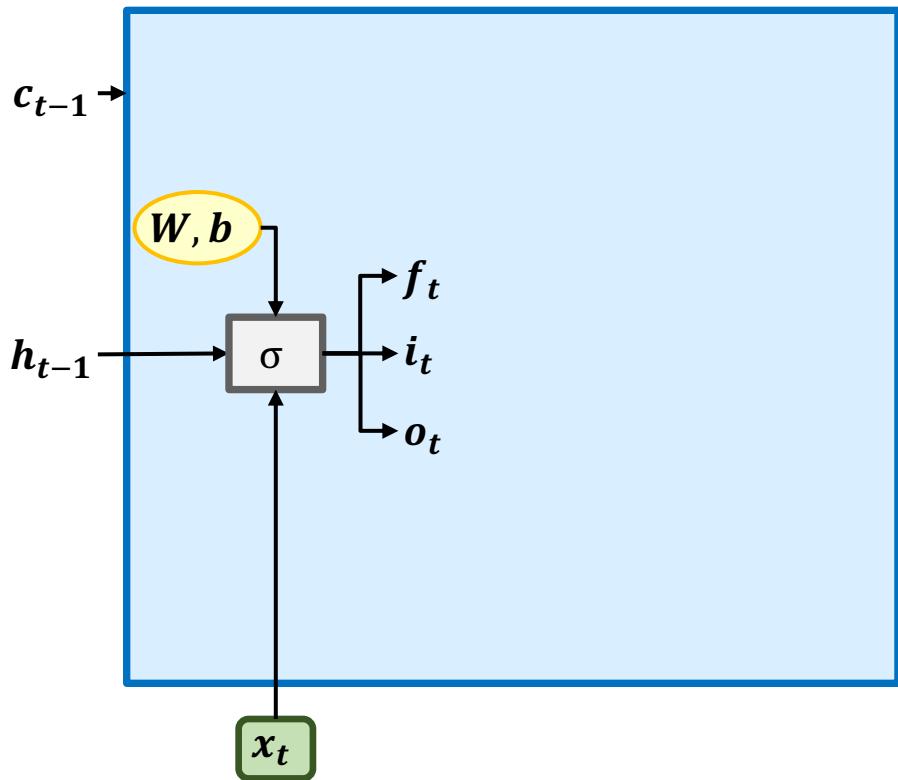
$$h_t = \tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$

Long-Short Term Memory Network (LSTM)



- Two states:
 - Hidden state h_{t-1}
 - Cell state c_{t-1}

Long-Short Term Memory Network (LSTM)



① Updating the gates:

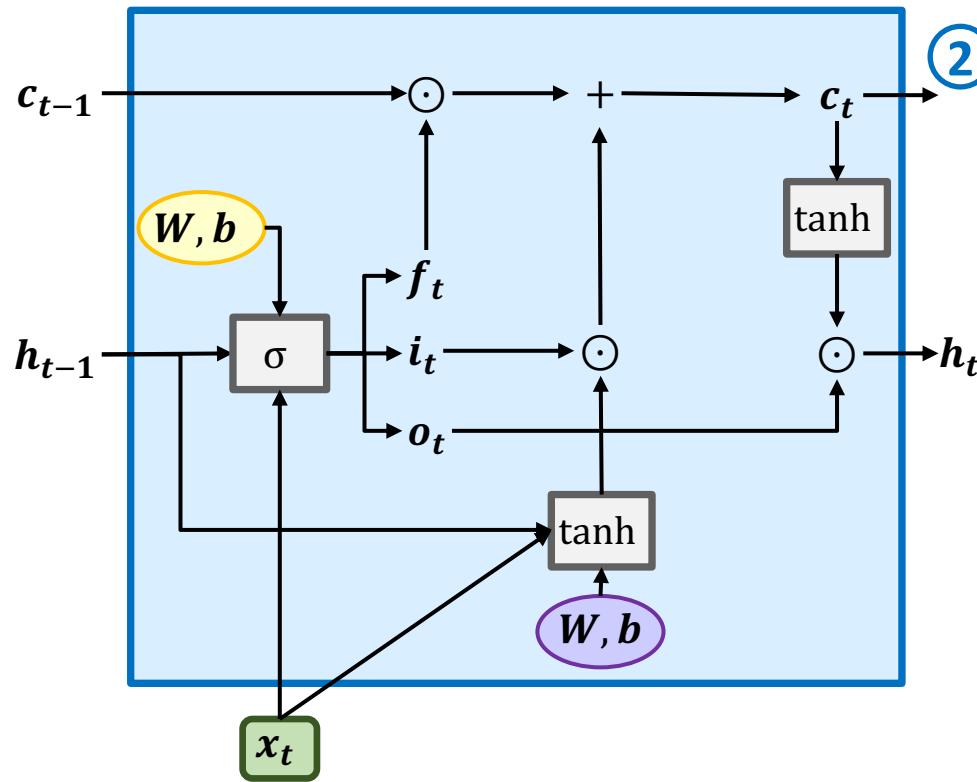
- f forget gate: whether to erase cell
- i input gate: whether to write to cell
- o output gate: how much to reveal cell

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

Long-Short Term Memory Network (LSTM)



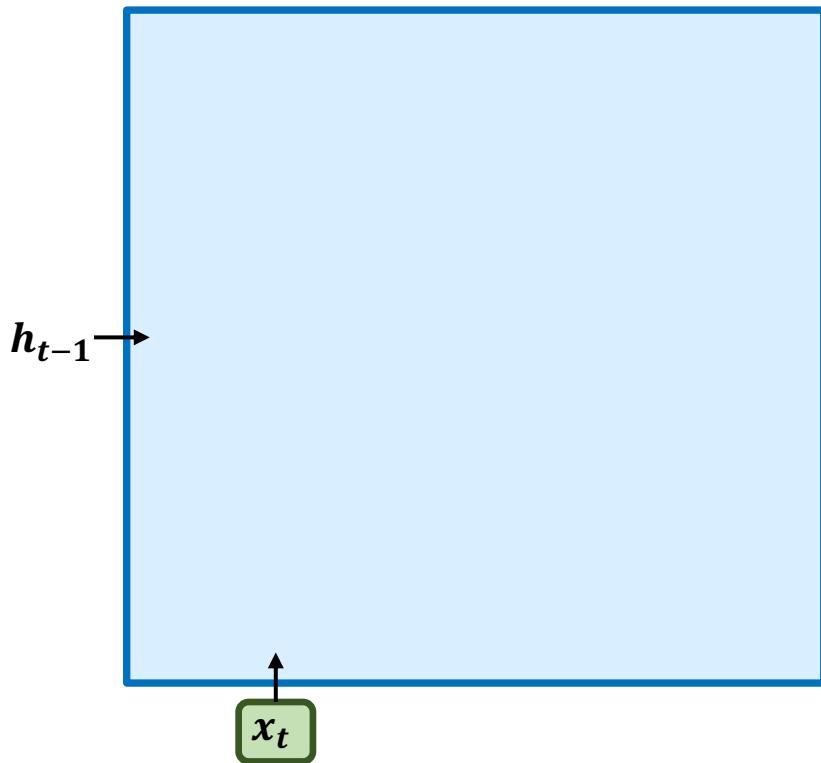
②

Updating the states:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$$

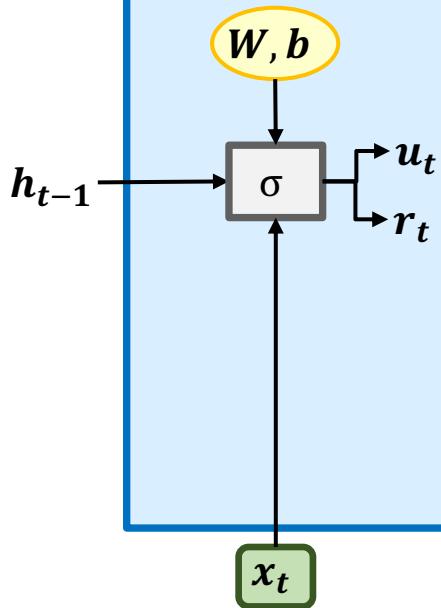
$$h_t = o_t \odot \tanh(c_t)$$

Gated Recurrent Units (GRU)



- Only one state (got rid of cell):
 - Hidden state h_{t-1}

Gated Recurrent Units (GRU)



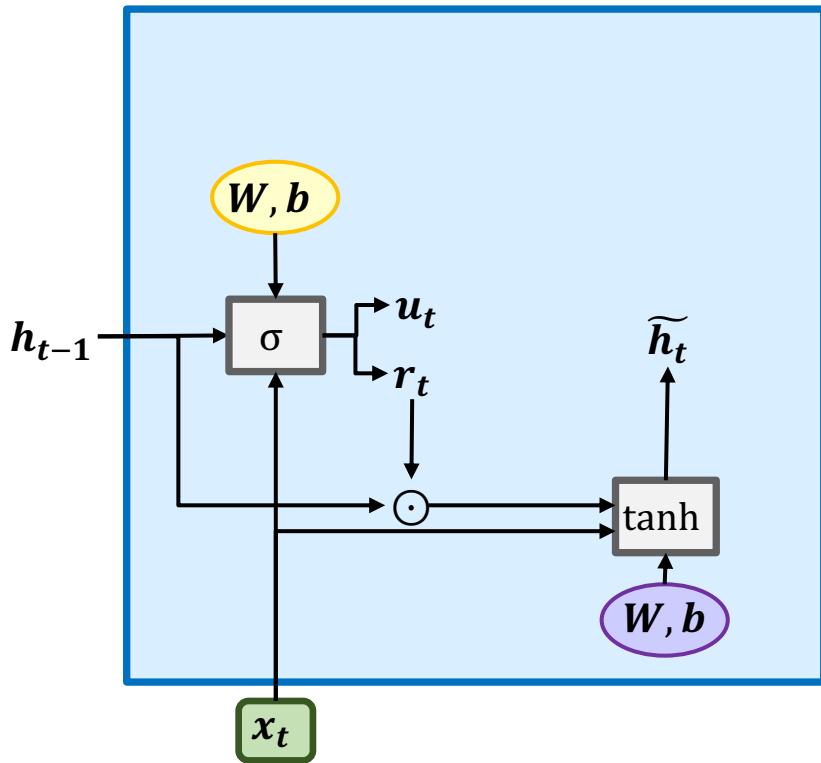
① Updating the gates:

- r reset gate: how much of the previous state to remember
- u update gate: how much of the new state is just a copy of the old state

$$r_t = \sigma(W_{rx}x_t + W_{th}h_{t-1} + b_r)$$

$$u_t = \sigma(W_{ux}x_t + W_{uh}h_{t-1} + b_u)$$

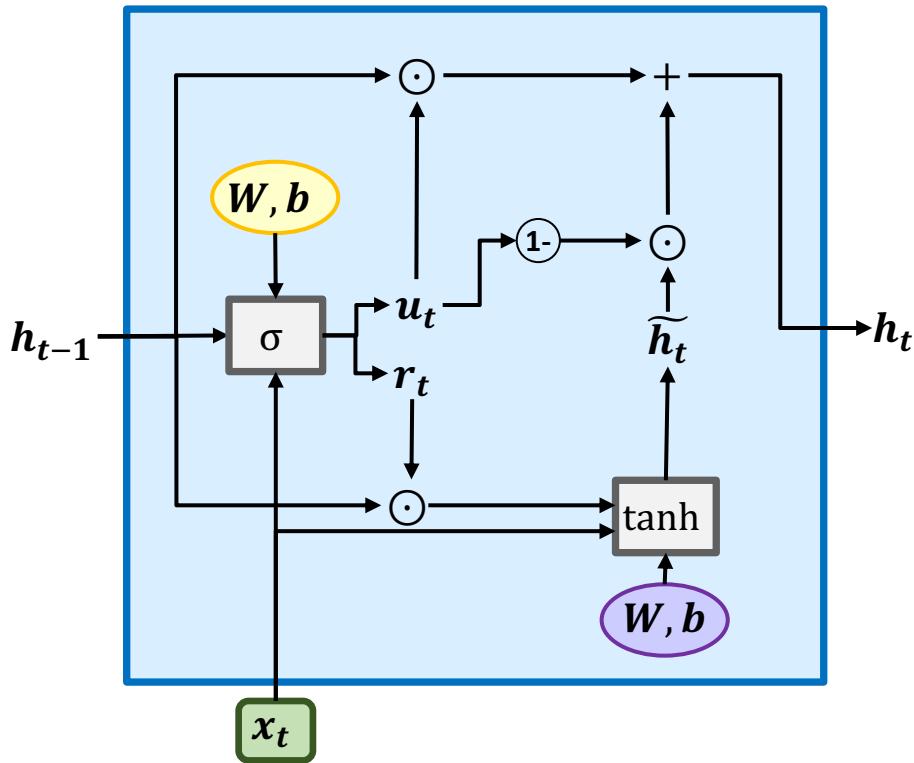
Gated Recurrent Units (GRU)



② Get candidate hidden state:

$$\tilde{h}_t = \tanh(W_{hx}x_t + W_{ht}(r_t \odot h_{t-1}) + b_h)$$

Gated Recurrent Units (GRU)

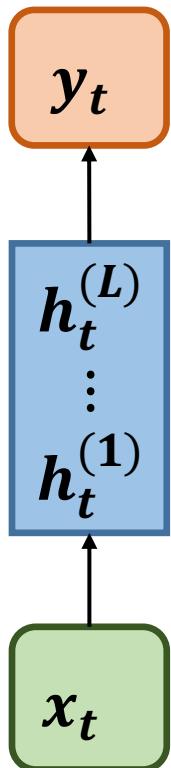


③ Updating the state:

$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot \tilde{h}_t$$

Same input/output – different architectures

Output Layer



Hidden Layer(s)

Input Layer

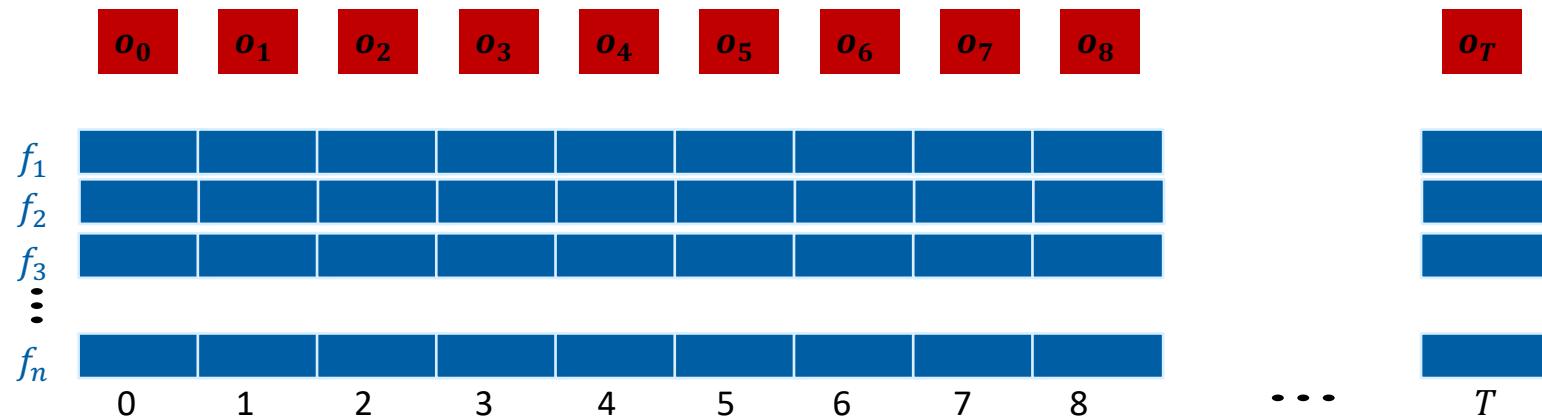
- Vanilla RNN
- LSTM
- GRU

Today – Recurrent Neural Networks

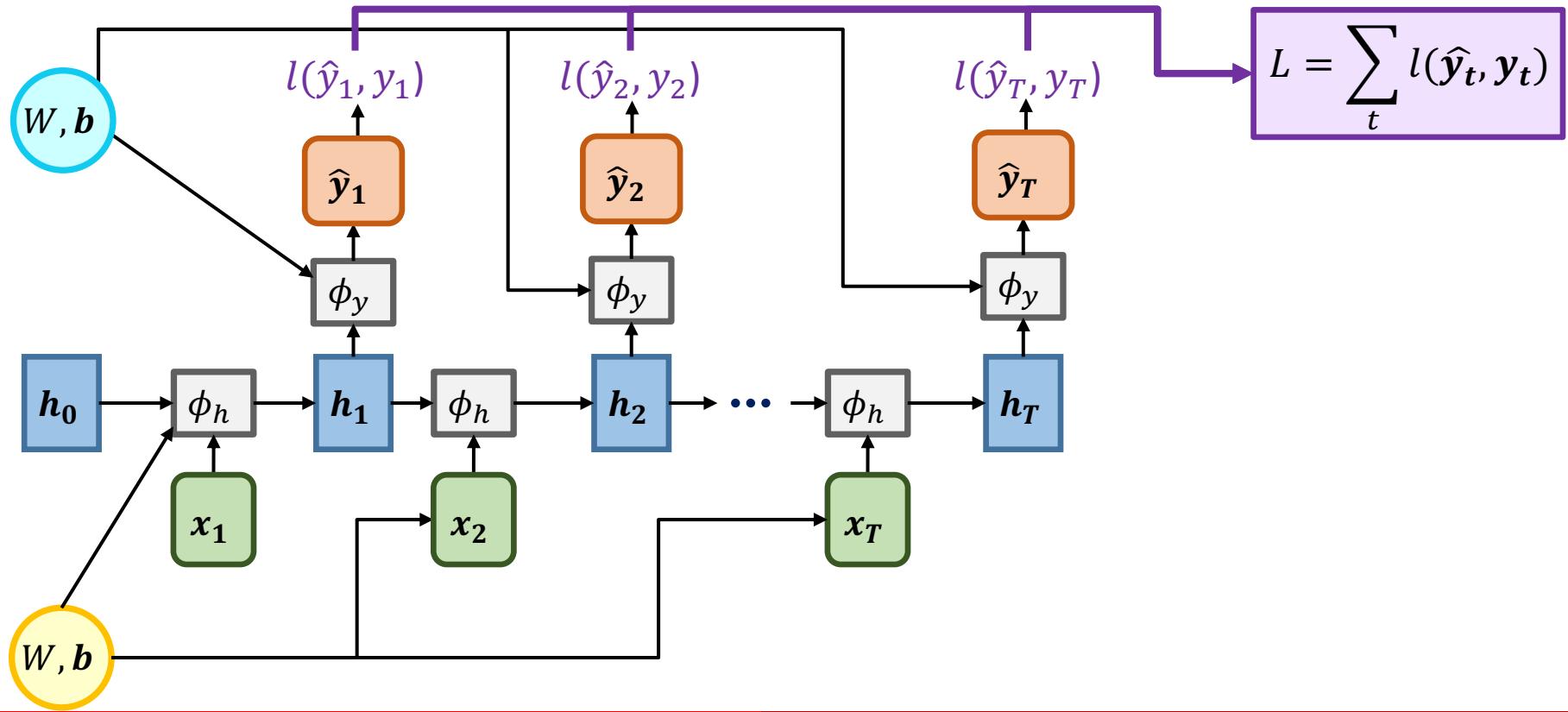
- Deep Knowledge Tracing
- Parameters and hyperparameter tuning
- Different architectures
- **Different tasks:**
 - “Many-to-many” versus “Many-to-one”
 - Classification versus Regression

Many-to-many aka the Tracing Task

- Prediction of a target variable o_t at each time step t

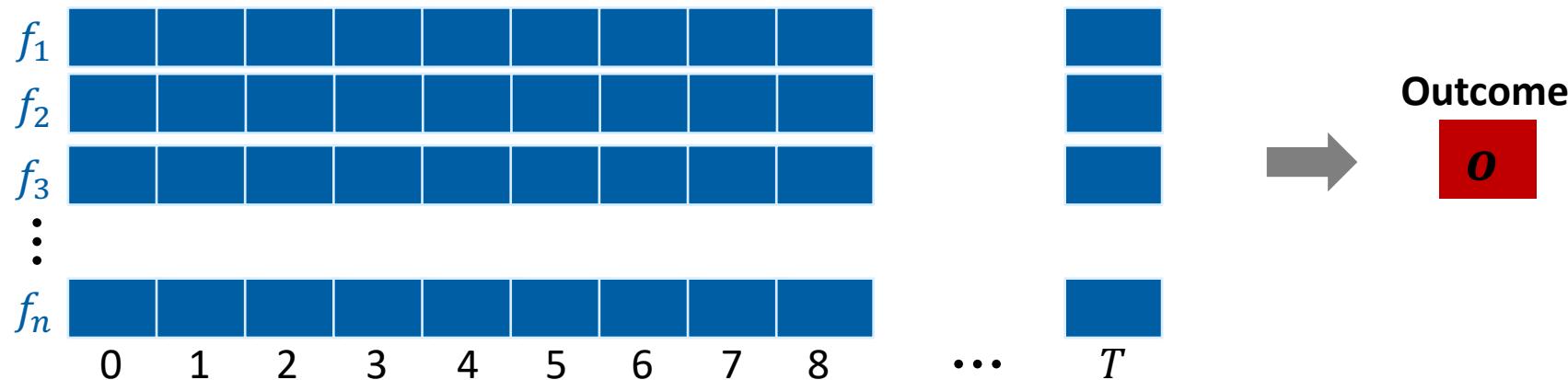


Computational Graph – Many-to-many

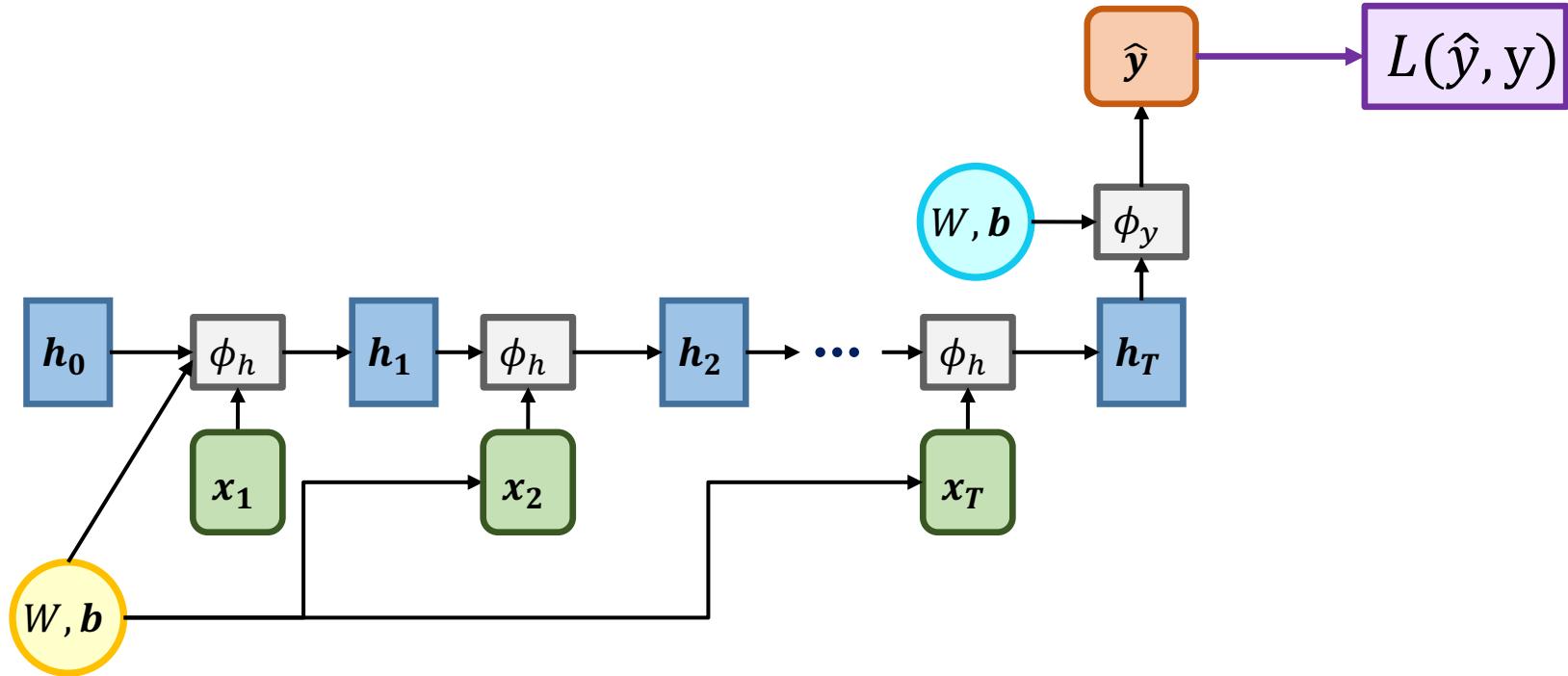


Many-to-one aka the Time-Series Prediction Task

- Prediction of a target variable o after $t \leq T$ time steps, where T is the total number of time steps



Computational Graph – Many-to-one



Today – Recurrent Neural Networks

- Deep Knowledge Tracing
- Parameters and hyperparameter tuning
- Different architectures
- **Different tasks:**
 - “Many-to-many” versus “Many-to-one”
 - **Classification versus Regression**

Classification vs. Regression

Output Layer

$$y_t$$

$$y_t = \phi_y(W_{yh}h_t^L + b^{(y)})$$

$$L(\hat{y}_t, y_t)$$

Hidden Layer(s)

$$\begin{matrix} h_t^{(L)} \\ \vdots \\ h_t^{(1)} \end{matrix}$$

Input Layer

$$x_t$$

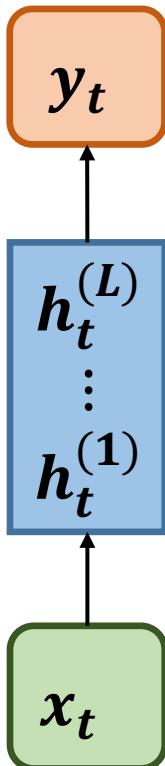
$$y_t$$

$$y_t = \phi_y(W_{yh}h_t^L + b^{(y)})$$

$$L(\hat{y}_t, y_t)$$

Classification vs. Regression: Output Layer

Output Layer



$$y_t = \phi_y(W_{yh}h_t^L + b^{(y)})$$

$$L(\hat{y}_t, y_t)$$

Classification

Regression

Hidden Layer(s)

Input Layer

Sigmoid activation:

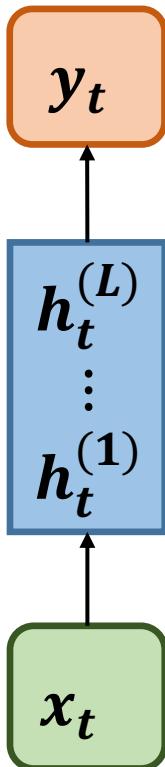
$$y_t = \sigma(W_{yh}h_t^L + b^{(y)})$$

Linear activation:

$$y_t = W_{yh}h_t^L + b^{(y)}$$

Classification vs. Regression: Training Loss

Output Layer



$$y_t = \phi_y(W_{yh}h_t^L + b^{(y)})$$

$$L(\hat{y}_t, y_t)$$

Classification

- Binary crossentropy
- Categorical crossentropy

Regression

Mean squared error

Hidden Layer(s)

Input Layer

Your Turn

- Given:
 - Data from a MOOC
 - An LSTM for predicting quiz performance of a student for every week of the course (tracing task)
- Your Task:
 - 1) Adjust the `create_model` function in order to predict pass/fail after 5 weeks of the course (time series prediction task) and send us the binary accuracy + AUC
 - Hint 1: `return_sequences=False`
 - Hint 2: what does `TimeDistributed(...)` do?
 - 2) Tune hyperparameters of your choice and send us binary accuracy and AUC

Summary

- Deep Knowledge Tracing
- Parameters and hyperparameter tuning
- Different architectures
- Different tasks:
 - “Many-to-many” versus “Many-to-one”
 - Classification versus Regression

Structure Discovery

Machine Learning for Behavioral Data
April 24, 2023

Today's Topic

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Fairness
13	Explainability
14	Project Presentations
15	Whit Monday

- 
- K-Means, Spectral Clustering
 - Choosing the optimal K*
 - Clustering time-series data

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

Which of the following NN architecture has the largest number of parameters?

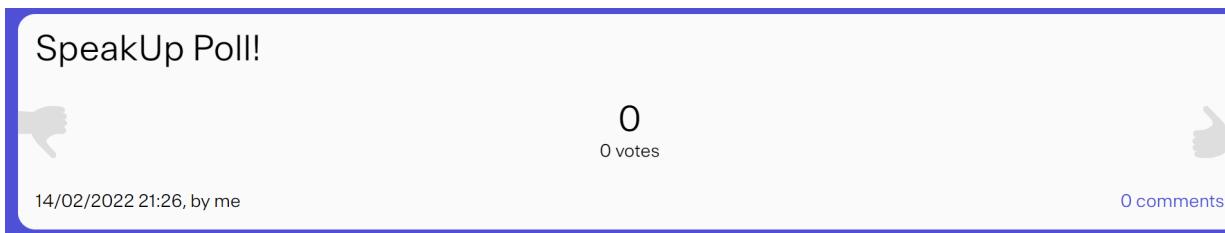
- a) RNN
- b) GRU
- c) LSTM

SpeakUp Poll!

0
0 votes

14/02/2022 21:26, by me

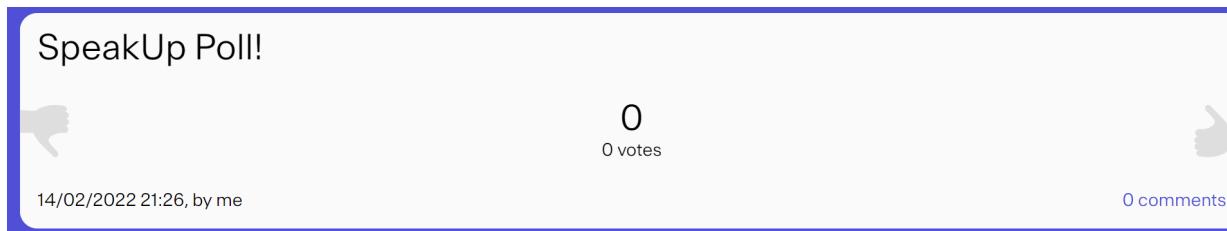
0 comments



Short quiz about the past...

In contrast to RNNs, GRU units include an additional memory cell.

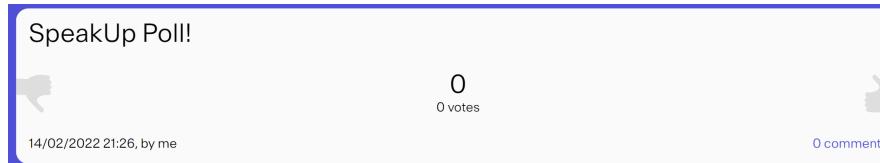
- a) True
- b) False



Short quiz about the past...

Which of the following should be changed to adapt a NN used for classification to a regression task?

- a) The dimension of the hidden layer
- b) The activation function of the output layer
- c) The batch size
- d) The drop-out rate



Why doing structure discovery?

- We are interested in finding different groups of users
 - for analytical purposes (e.g., to analyze how different types of users use our services)
 - to adapt the environment to different user types
 - Examples:
 - Finding groups of students with similar strategies
 - Identifying different types of new users on Snapchat (personalized retention)
 - Grouping tourists by their mobility patterns (recommendation)
-

Agenda

- Clustering Algorithms
 - K-Means Clustering
 - Spectral Clustering
- Choosing the optimal number of clusters

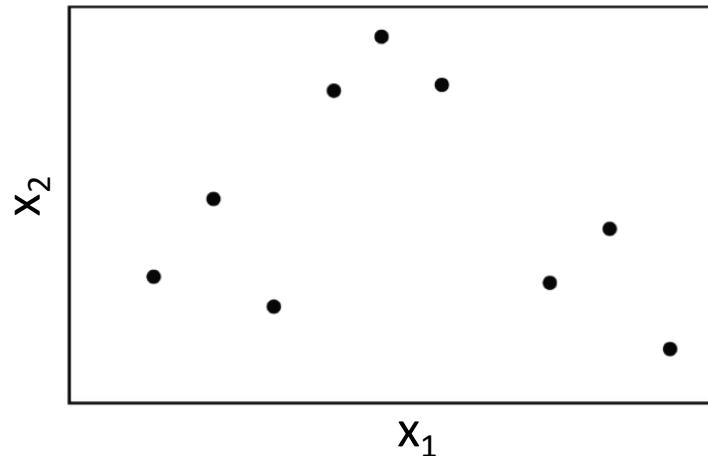


Agenda

- Clustering Algorithms
 - K-Means Clustering
 - Spectral Clustering
- Choosing the optimal number of clusters



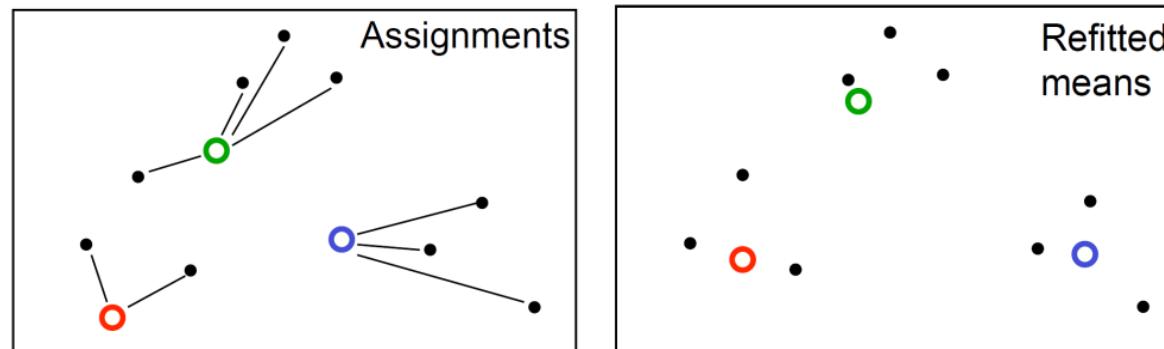
K-Means Clustering



- Assume the data $\{x_1, \dots, x_N\}$ lives in a **Euclidean** space, $x_n \in \mathbb{R}^D$
- Assume the data belongs to K different classes (groups)
- How can we identify those classes (data points that belong to each class)?

K-Means Algorithm

- **Initialization:** randomly assign cluster centers
- Algorithm iteratively alternates between two steps:
 - **Assignment** step: assign each data point to the closest cluster
 - **Update** step: move each cluster center to the center of gravity of the data assigned to it



[Image credit: Urtasun, CSC-411]

K-Means Algorithm

- **Initialization:** set K cluster means $\mathbf{m}_1, \dots, \mathbf{m}_K$ to random values
- Repeat until convergence (until assignments do not change):
 - **Assignment:** each data point x_n assigned to nearest mean

$$\hat{k}^n = \arg \min_k d(\mathbf{m}_k, x_n)$$

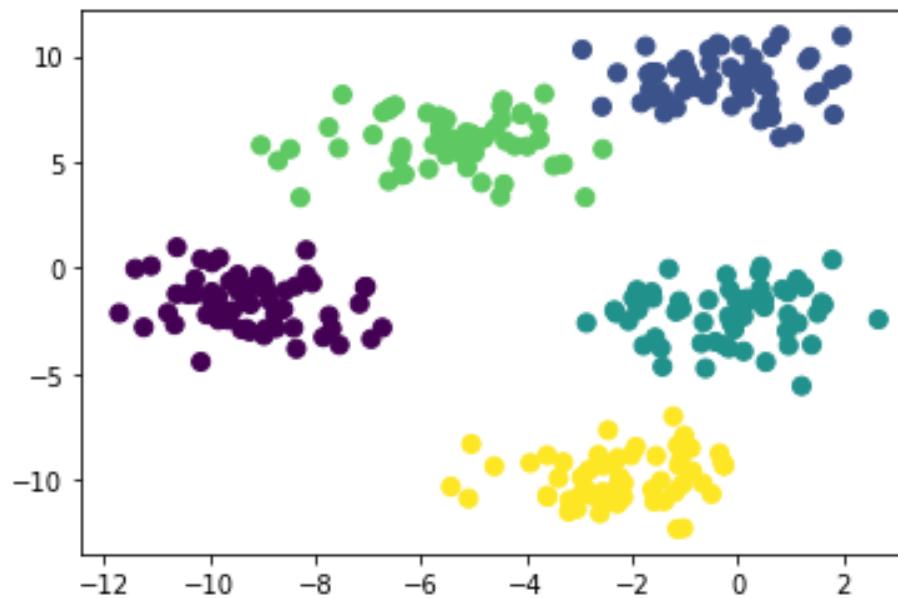
(e.g, with Euclidean distance: $d(\mathbf{m}_k, x_n) = \|\mathbf{m}_k - x_n\|_2$)

- **Update:** adjust means to match sample means of data points they are responsible for:

$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)} \mathbf{x}_n}{\sum_n r_k^{(n)}}, \text{ where } r_k^{(n)} = 1, \text{ if } \hat{k}^n = k$$

K-Means Example

Synthetic data with k=5 clusters



Observations

- Solution (goodness of solution) depends on the initial positioning of the cluster centers
 - Solution (goodness of solution) depends on the choice of k (the number of clusters)
- How should we initialize the cluster centers?
- How to choose the optimal number of clusters?

Initialization of cluster centers

- Random restarts:
 - Run to convergence using different random initializations
 - Choose the one that minimizes distortion (squared distance of data to cluster means)
- Distortion D (measure of in-cluster variance):

$$D = \sum_n \left(d(\mathbf{m}_{\hat{k}^n}, \mathbf{x}_n) \right)^2$$

(e.g., with Euclidean distance: $d(\mathbf{m}_k, \mathbf{x}_n) = \|\mathbf{m}_k - \mathbf{x}_n\|_2$)

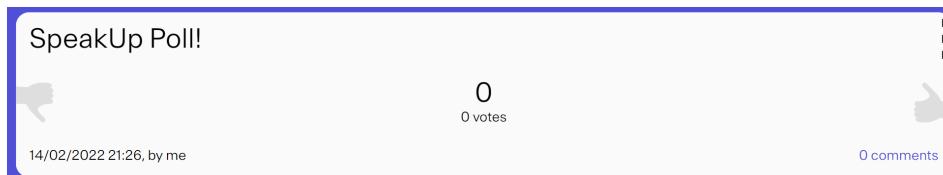
Minimizing Distortion

- Distortion D (measure of in-cluster variance):

$$D = \sum_n \left(d(\mathbf{m}_{\hat{k}^n}, \mathbf{x}_n) \right)^2$$

(e.g, with Euclidean distance: $d(\mathbf{m}_k, \mathbf{x}_n) = \|\mathbf{m}_k - \mathbf{x}_n\|_2$)

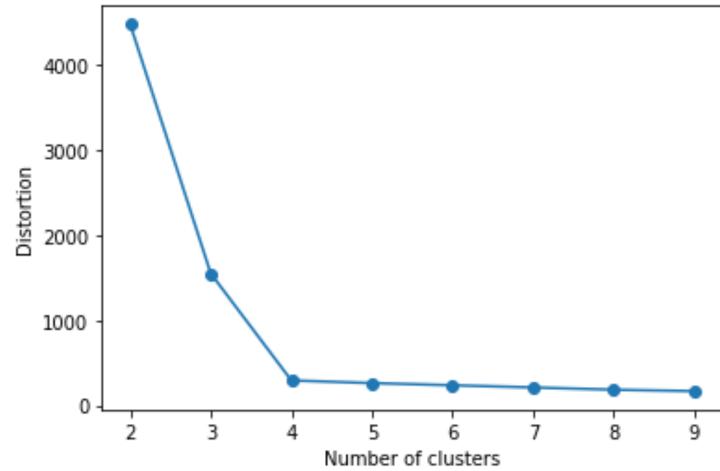
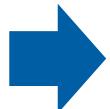
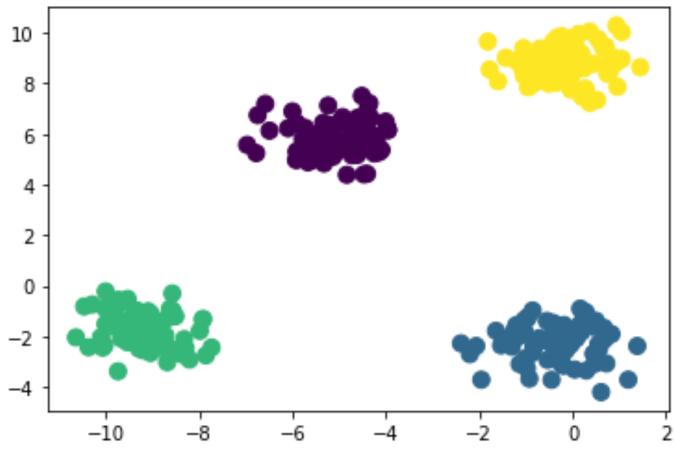
- Could we determine the optimal number of clusters by minimizing D?



- a) Yes
- b) No

Selecting k^* - Elbow Method

- Elbow Method (Heuristic): choose k^* such that adding another cluster does not lead to a much better model of the data



Selecting k^* - Silhouette Score

- Silhouette width ($-1 \leq s \leq 1$): Silhouette width measures how similar a data point is to its own cluster (cohesion) compared to other clusters (separation)
- Can be computed for $k = 2, \dots, N$
- For a data point \mathbf{x}_i in cluster C_k :

$$a(i) = \frac{1}{|C_k| - 1} \sum_{j \in C_k, i \neq j} d(\mathbf{x}_i, \mathbf{x}_j) \quad b(i) = \min_{l \neq k} \frac{1}{|C_l|} \sum_{j \in C_l} d(\mathbf{x}_i, \mathbf{x}_j)$$

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \text{ if } |C_k| > 1 \quad s(i) = 0 \text{ if } |C_k| = 1$$

Selecting k^* - Silhouette Score

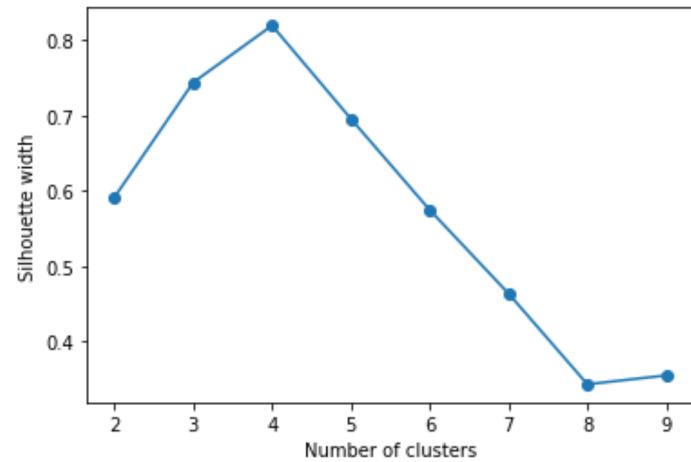
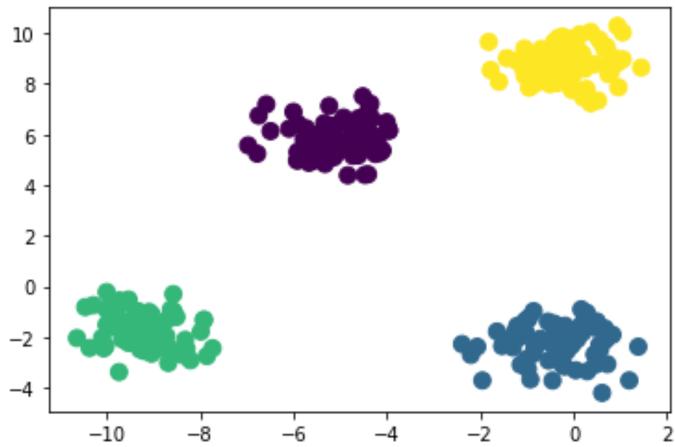
- Overall average silhouette width: average over all data points x_i , given a cluster number k :

$$\bar{s}_k = \frac{1}{N} \sum_{i=1}^N s(i)$$

- Find k^* that maximizes the overall average silhouette width:

$$k^* = \arg \max_k \bar{s}_k$$

Example: Silhouette Width



Selecting k^* - BIC Score

- Assumptions of K-Means:
 - Data points live in an Euclidean space
 - Data points are spherically distributed around centroid of clusters (spherical Gaussians)
- We can compute the likelihood of our cluster solution for a given k
- We can use the *BIC* to determine k^*

Bayesian Information Criterion (BIC)

$$BIC = -2 \cdot LL + \log(N) \cdot d$$

- d is the number of parameters of our model f
- LL is the log-likelihood (logarithm of the likelihood) of the sample data T given a specific k
- N is the number of samples in the data set, i.e. $|T| = N$

Schwarz Criterion (BIC)

$$BIC = -2 \cdot LL + \log(N) \cdot d$$



$$BIC = LL - \frac{d}{2} \cdot \log(N)$$

Likelihood for one data point

- Probability for a data point x_i , where \hat{r}^i denotes the cluster assignments for x_i (spherical gaussian assumption):

$$p(x_i) = \frac{|C_{\hat{r}^i}|}{N} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2\sigma^2} \cdot \|x_i - \mathbf{m}_{\hat{r}^i}\|_2^2}$$

- Computing the variance σ^2 of the data:

$$\sigma^2 = \frac{1}{N - k} \cdot \sum_{i=1}^N (x_i - \mathbf{m}_{\hat{r}^i})^2$$

Computing the log-likelihood

- Compute the log-likelihood over all data points:

$$\begin{aligned} LL &= \log \prod_{i=1}^N p(x_i) \\ &= \sum_{i=1}^N \log \frac{|C_{\hat{r}^i}|}{N} + \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \cdot \|x_i - m_{\hat{r}^i}\|_2^2 \end{aligned}$$

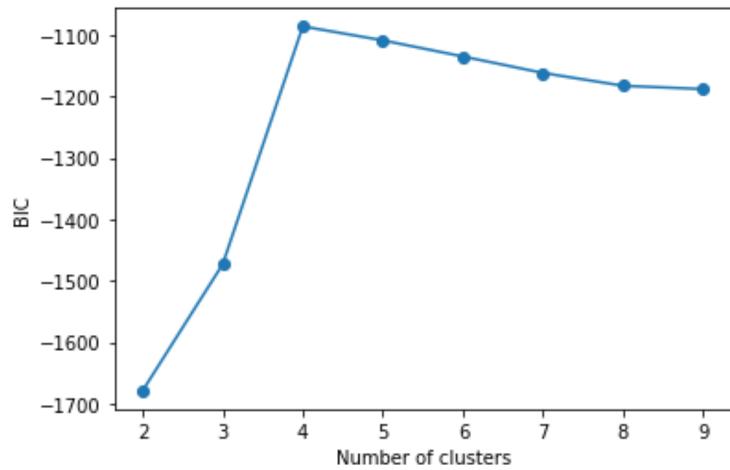
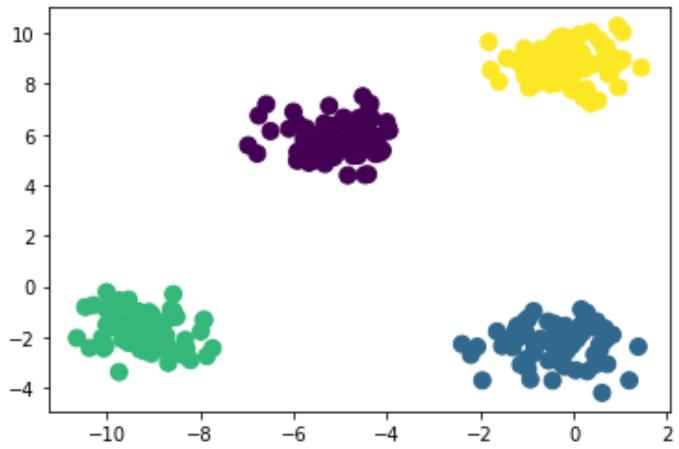
Computing the number of parameters d

- Here, number of parameters is equivalent to degrees of freedom:

$$d = (k - 1) + 1 + k \cdot D$$

- k is the number of clusters, D is the number of dimensions of the data points x_i
- We estimate the following:
 - $k - 1$ prior probabilities (for the k clusters)
 - 1 variance estimate (σ^2)
 - $k \cdot D$ centroid coordinates

BIC Example



Agenda

- Clustering Algorithms
 - K-Means Clustering
 - **Spectral Clustering**
- Choosing the optimal number of clusters

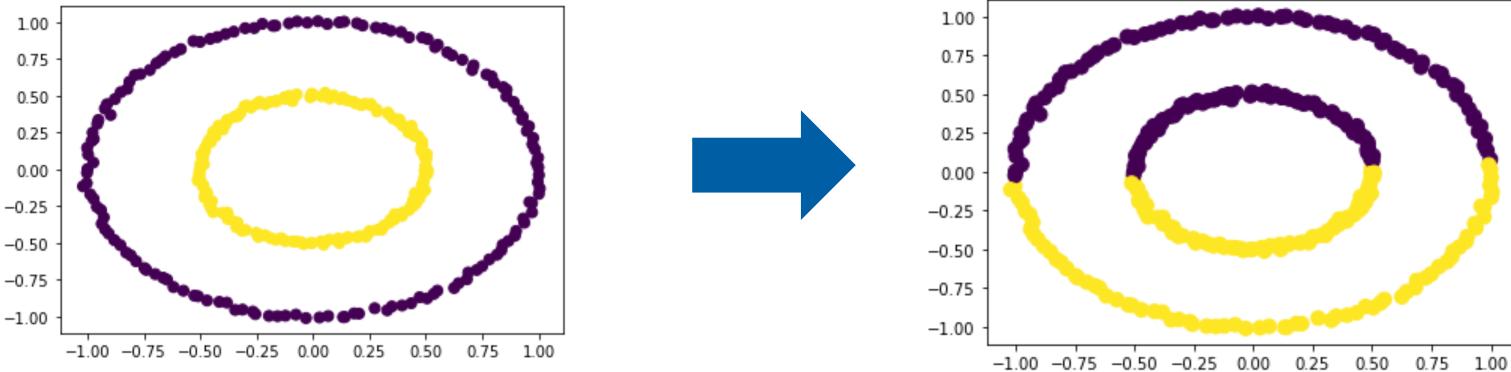


Two broad assumptions for clustering

- **Compactness:** Points that lie **close** to each other fall in the same cluster and are compact around the cluster center. The closeness can be measured by the distance between the observations.
 - **Connectivity:** Points that are **connected** or immediately next to each other are put in the same cluster. Even if 2 points are close together, if they are not connected, they are not clustered together.
-

Assumptions of K-Means

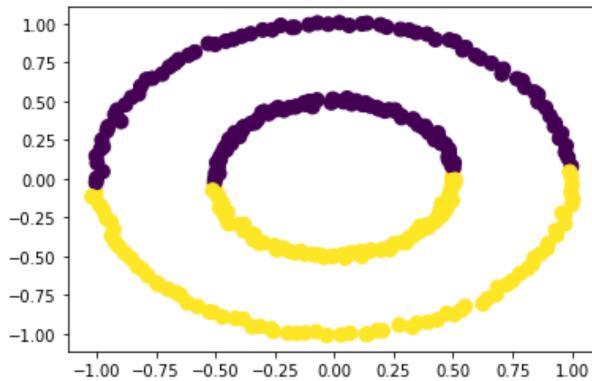
- K-Means assumes a Euclidean space
- K-Means assumes that the variance of the distribution of each cluster is spherical



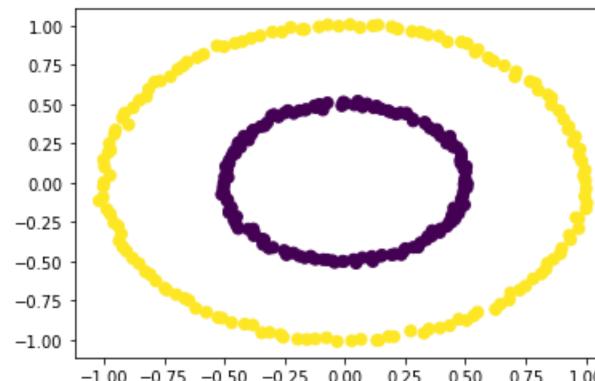
Assumptions of K-Means

- K-Means assumes a Euclidean space
- K-Means assumes that the variance of the distribution of each cluster is spherical

K-Means



Spectral Clustering



Spectral Clustering

- No assumption is made about the form/shape of the clusters
 - Data points are treated as nodes of graphs
 - Algorithm consists of three steps:
 1. Compute the pairwise similarities $s(x_i, x_j)$ between all pairs of data points i and j
 2. Construct a similarity graph
 3. Compute first k eigenvectors (k is the number of clusters) of graph Laplacian
 4. Perform clustering on transformed data
-

Spectral Clustering

- No assumption is made about the form/shape of the clusters
 - Data points are treated as nodes of graphs
 - Algorithm consists of three steps:
 1. Compute the pairwise similarities $s(x_i, x_j)$ between all pairs of data points i and j
 2. Construct a similarity graph
 3. Compute first k eigenvectors (k is the number of clusters) of graph Laplacian
 4. Perform clustering on transformed data
-

Similarity Measures

- Quantify similarity between two samples
- No single definition exists, can usually be seen as the inverse of distance metrics



Similarity Measures

- **Cosine Similarity:** for vectors, often used for document comparison

$$S_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

- **Jaccard Similarity:** for set data

$$d(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$



Similarity Measures

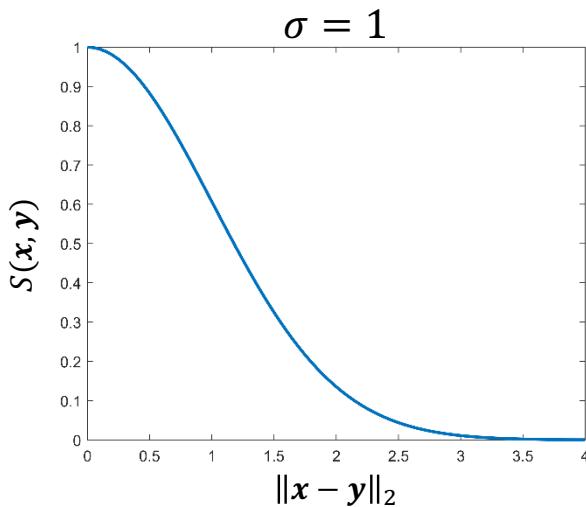
- Gaussian Kernel with Euclidean distance (takes into account local neighborhood)

$$S(x, y) = e^{-\frac{\|x-y\|_2^2}{2\sigma^2}}$$

Similarity Measures

- Gaussian Kernel with Euclidean distance (takes into account local neighborhood)

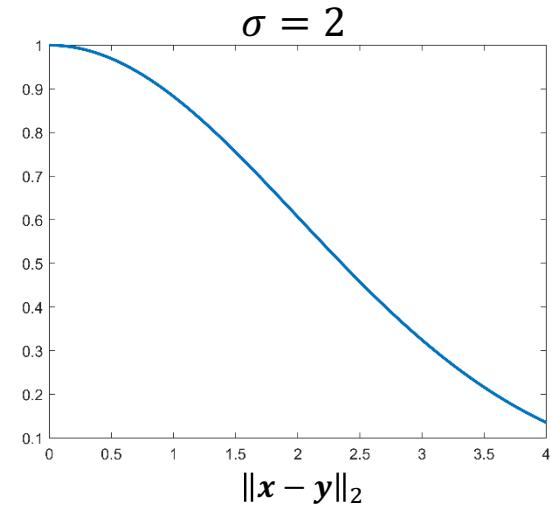
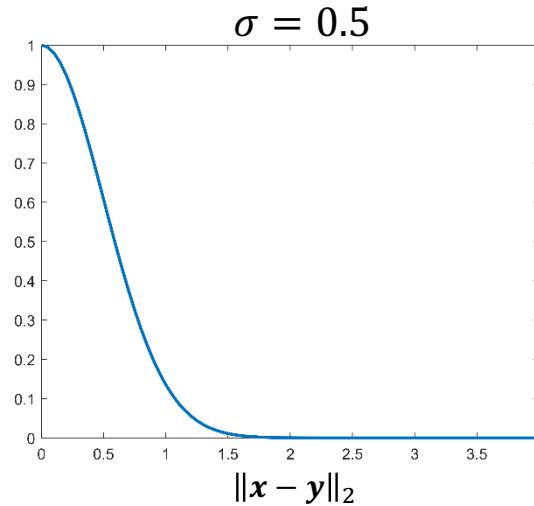
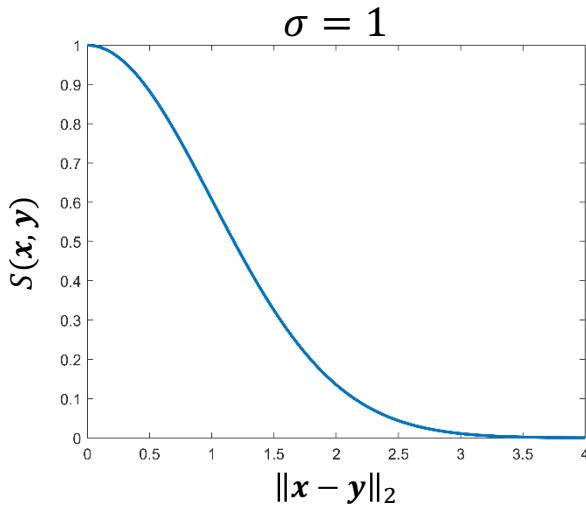
$$S(x, y) = e^{-\frac{\|x-y\|_2^2}{2\sigma^2}}$$



Similarity Measures

- **Gaussian Kernel** with Euclidean distance (takes into account local neighborhood)

$$S(x, y) = e^{-\frac{\|x-y\|_2^2}{2\sigma^2}}$$



Spectral Clustering

- No assumption is made about the form/shape of the clusters
 - Data points are treated as nodes of graphs
 - Algorithm consists of three steps:
 1. Compute the pairwise similarities $s(x_i, x_j)$ between all pairs of data points i and j
 2. **Construct a similarity graph**
 3. Compute first k eigenvectors (k is the number of clusters) of graph Laplacian
 4. Perform clustering on transformed data
-

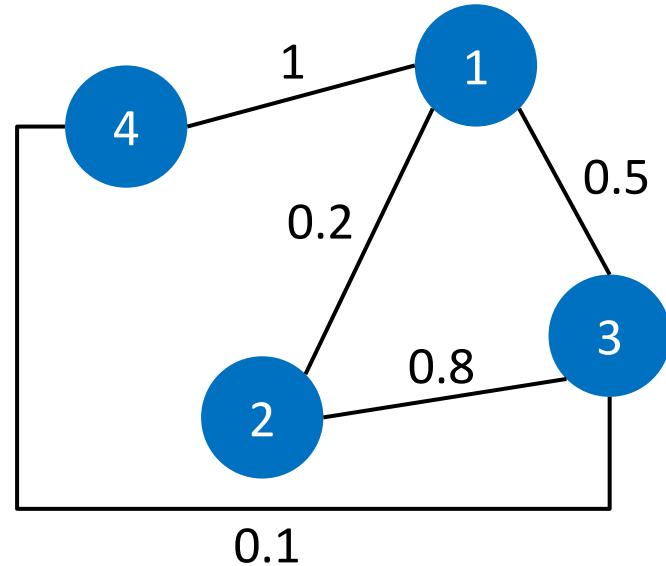
Undirected Graphs - Notation

- Weighted adjacency matrix W

$$W = \begin{pmatrix} 0 & 0.2 & 0.5 & 1 \\ 0.2 & 0 & 0.8 & 0 \\ 0.5 & 0.8 & 0 & 0.1 \\ 1 & 0 & 0.1 & 0 \end{pmatrix}$$

- Degree d_i of a node i : $d_i = \sum_{j=1}^n w_{ij}$

- Degree matrix D : $D = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_n \end{pmatrix}$



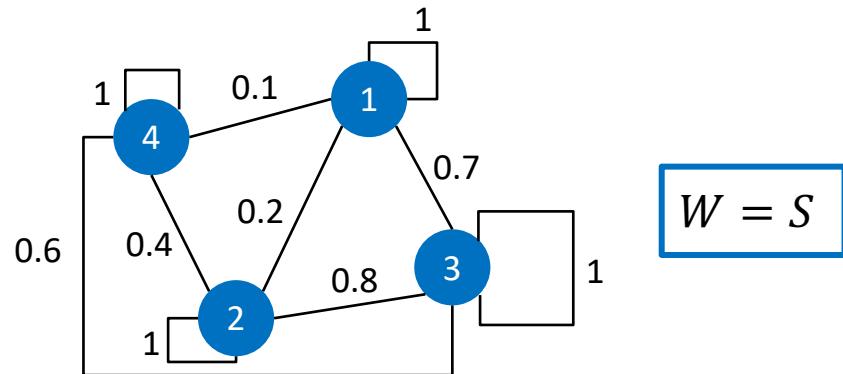
Constructing Similarity Graphs

- Given: a set of data points x_1, \dots, x_n and some notion of similarity $s_{ij} \geq 0$ between all pairs of data points x_i and x_j
 - We assume that
 - Each data point x_i represents a vertex of a graph
 - Two “vertices” x_i and x_j are connected, if s_{ij} is larger than a threshold (or zero), and the edge is weighted with s_{ij}
-

Constructing Similarity Graphs

- *Fully connected graph*: simply connect all data points x_i and x_j with positive similarity s_{ij} and weight the edges with s_{ij}
- Example (S denotes the similarity matrix):

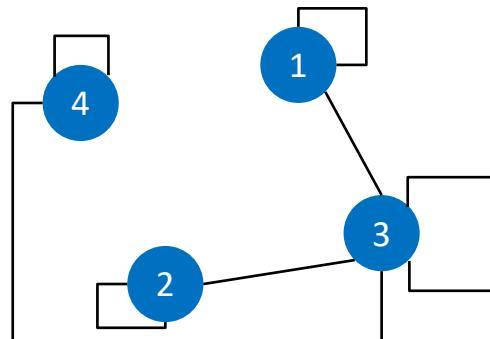
$$S = \begin{pmatrix} 1 & 0.2 & 0.7 & 0.1 \\ 0.2 & 1 & 0.8 & 0.4 \\ 0.7 & 0.8 & 1 & 0.6 \\ 0.1 & 0.4 & 0.6 & 1 \end{pmatrix}$$



Constructing Similarity Graphs

- ε –neighborhood graph: we connect all data points x_i and x_j with similarity $s_{ij} > \varepsilon$ and treat the graph as unweighted
- Example with $\varepsilon = 0.5$ (S denotes the similarity matrix):

$$S = \begin{pmatrix} 1 & 0.2 & 0.7 & 0.1 \\ 0.2 & 1 & 0.8 & 0.4 \\ 0.7 & 0.8 & 1 & 0.6 \\ 0.1 & 0.4 & 0.6 & 1 \end{pmatrix}$$

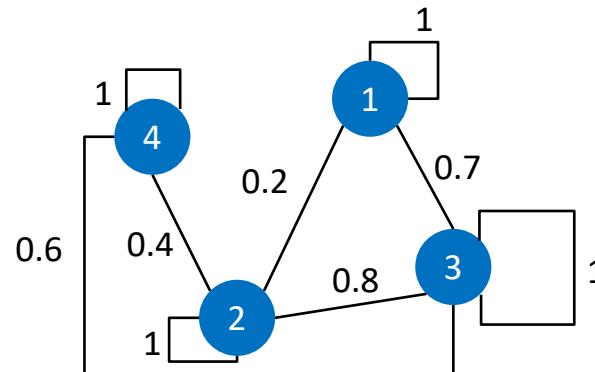


$$W = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Constructing Similarity Graphs

- *k –nearest neighbor graph*: we connect all data points x_i and x_j if x_i is among the k nearest neighbors of x_j **or** x_j is among the k nearest neighbors of x_i
- *mutual k –nearest neighbor graph*: we connect all data points x_i and x_j if x_i is among the k nearest neighbors of x_j **and** x_j is among the k nearest neighbors of x_i
- Example with $k = 2$ (S denotes the similarity matrix):

$$S = \begin{pmatrix} 1 & 0.2 & 0.7 & 0.1 \\ 0.2 & 1 & 0.8 & 0.4 \\ 0.7 & 0.8 & 1 & 0.6 \\ 0.1 & 0.4 & 0.6 & 1 \end{pmatrix}$$



$$W = \begin{pmatrix} 1 & 0.2 & 0.7 & 0 \\ 0.2 & 1 & 0.8 & 0.4 \\ 0.7 & 0.8 & 1 & 0.6 \\ 0 & 0.4 & 0.6 & 1 \end{pmatrix}$$

Spectral Clustering - Algorithm

Unnormalized spectral clustering

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct

- Construct a similarity graph and compute W (weighted adjacency matrix) and D (degree matrix)
- Compute the unnormalized graph Laplacian $L = D - W$
- Compute the first k eigenvectors u_1, \dots, u_k of L
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U
- Cluster the points y_i into clusters C_1, \dots, C_k using k-means clustering (e.g., using Euclidean distance)

Output: Clusters C_1, \dots, C_k

Spectral Clustering - Algorithm

Unnormalized spectral clustering

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, number of clusters k

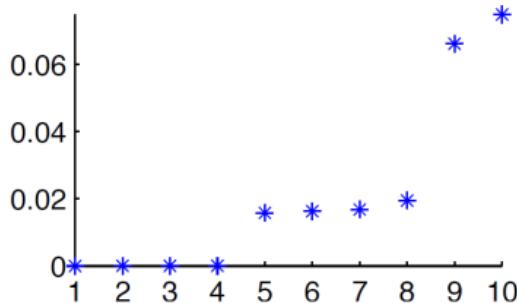
- Construct a similarity graph and compute its adjacency matrix W and degree matrix D
- Compute the unnormalized graph Laplacian $L = D - W$
- Compute the first k eigenvectors u_1, \dots, u_k of L
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U
- Cluster the points y_i into clusters C_1, \dots, C_k using k-means clustering (e.g. using Euclidean distance)

Output: Clusters C_1, \dots, C_k

Normalized Laplacian: $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$
Random Walk Laplacian: $L = I - D^{-1}W$

Selecting the optimal number of clusters k^*

- *Eigengap Heuristic*: choose k^* such that the first k eigenvalues $\lambda_1, \dots, \lambda_k$ are very small, but λ_{k+1} is relatively large



- We can also use the *Silhouette* score to select the optimal number of clusters (on the embedding space)

Agenda

- Clustering Algorithms
 - K-Means Clustering
 - Spectral Clustering
- **Choosing the optimal number of clusters**



Choosing k^* - Flipped Classroom Data

- Participants: 288 EPFL students of a course taught in *flipped classroom* mode with a duration of 10 weeks
 - Structure:
 - Preparation: watch videos (and solve simple quizzes) on **new content** at home as a preparation for the lecture
 - Lecture: discuss open questions and solve more complex tasks
 - Lab session: solve paper-an-pen assignments
 - Data: clickstream data (all interactions of the student with the system)
-

Choosing k^* - Your Turn

- In practice, clusters are not always as well separable...
 - Your Task:
 1. Choose one of the suggested feature groups (Effort or Proactivity)
 2. Cluster the students based on these feature groups
 3. Compute (and visualize) the eigengap heuristic as well as the Silhouette score
 4. Discuss your findings: what number of clusters would you choose?
Why?
 - If you have time: repeat for the second feature group
-

Summary

- K-Means Clustering
 - Popular, easy to implement
 - Assumptions: data points live in Euclidean space, spherical distribution around cluster centroids
 - Need to choose: initialization, distance measure (e.g., Euclidean distance), number of clusters k
- Spectral Clustering
 - Flexible, no assumptions about shape/form of clusters
 - Need to choose: similarity measure, computation of similarity graph, computation of graph Laplacian, number of clusters k

Apply for Fall Projects at ML4ED

- Deadline: May 12th

<https://go.epfl.ch/ml4ed-projects>

Time Series Clustering

Machine Learning for Behavioral Data
May 1, 2023

Today's Topic

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Fairness
13	Explainability
14	Project Presentations
15	Whit Monday

- 
- K-Means, Spectral Clustering
 - Choosing the optimal K*
 - Clustering time-series data

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

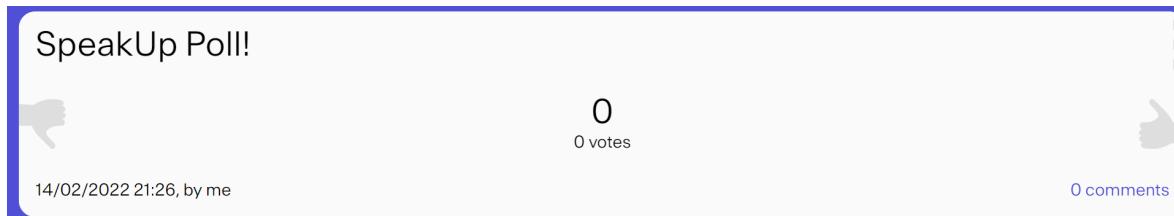
<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

In K-Means, which of the following parameters affect the goodness of the solution?

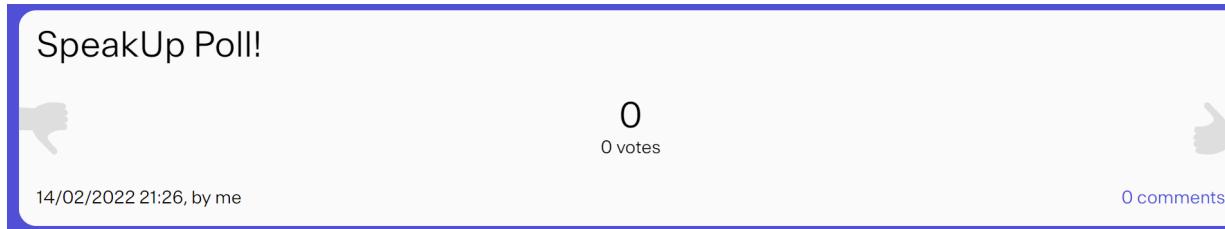
- a) Number of iterations
- b) Initial positioning of cluster centers
- c) Choice of k



Short quiz about the past...

K-Means is useful when dealing with non-convex clusters:

- a) True
- b) False



Short quiz about the past...

In a binary classification problem, it is appropriate to use the following activation function for the output layer:

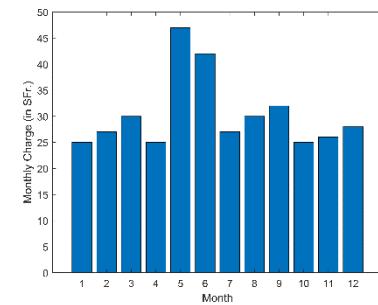
- a) Linear
- b) Tanh
- c) Sigmoid



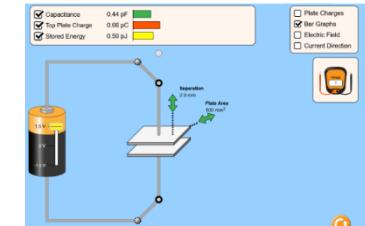
Today – Clustering Time Series Data

1. Aggregating features over time
2. Defining fixed time intervals (weeks, levels in a game, etc.)

3. Dynamic Time Warping



-
4. String Metrics
 5. Markov Models



Action Sequences

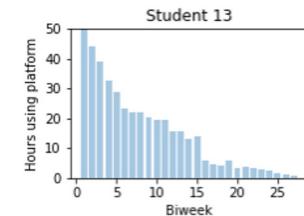
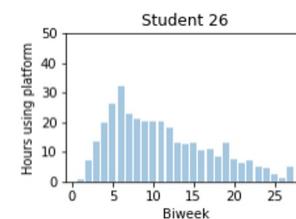
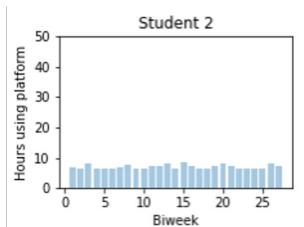
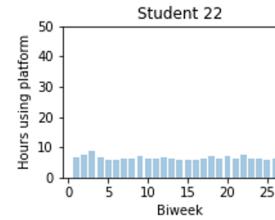
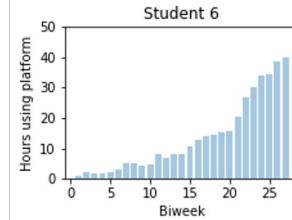
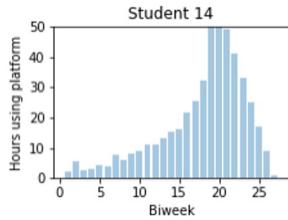
Learning Objectives

You should be able to:

- Explain the different approaches to time series clustering
 - Describe their advantages and disadvantages and when it is appropriate to use them
 - Implement these approaches (lecture/lab session)
 - Apply them to real-world data (lab session)
-

Today's Use Case

- Synthetic data of 30 high school students
- Time spent on an e-learning platform over one year (computed per biweek)
- Three clusters: 1) precrastinators, 2) regular, 3) procrastinators



Agenda

- **Aggregating features over time**
- Defining fixed time intervals (weeks, levels in a game, etc.)
- Dynamic Time Warping
- String Metrics
- Markov Models



Aggregating features over time

- We compute the value of the feature over the whole time series (average, maximum, range, standard deviation)
 - We do not explicitly represent changes in features over time
- We can use standard distance/similarity measures

Your Turn – Aggregated Data

Run spectral clustering on the average number of hours:

- Can we interpret the different clusters?
- Are we able to retrieve the procrastination patterns? If not, why not?

SpeakUp Chat!



0
0 votes



0 comments

14/02/2022 21:25, by me

Agenda

- Aggregating features over time
- **Defining fixed time intervals (weeks, levels in a game, etc.)**
- Dynamic Time Warping
- String Metrics
- Markov Models
- Additional Practice



Using fixed time intervals

- Compute the feature value at fixed points in time (e.g., weeks, level in a game)
 - We obtain feature vectors with the same length for every student
- We can use standard distance measures



Your Turn – Fixed Time Intervals

Run spectral clustering on the vectors of biweeks (dimension = 27) using Euclidean distance:

- What is the optimal number of clusters?
- How do the results differ from the aggregated feature results?

SpeakUp Chat!



0
0 votes



0 comments

14/02/2022 21:25, by me

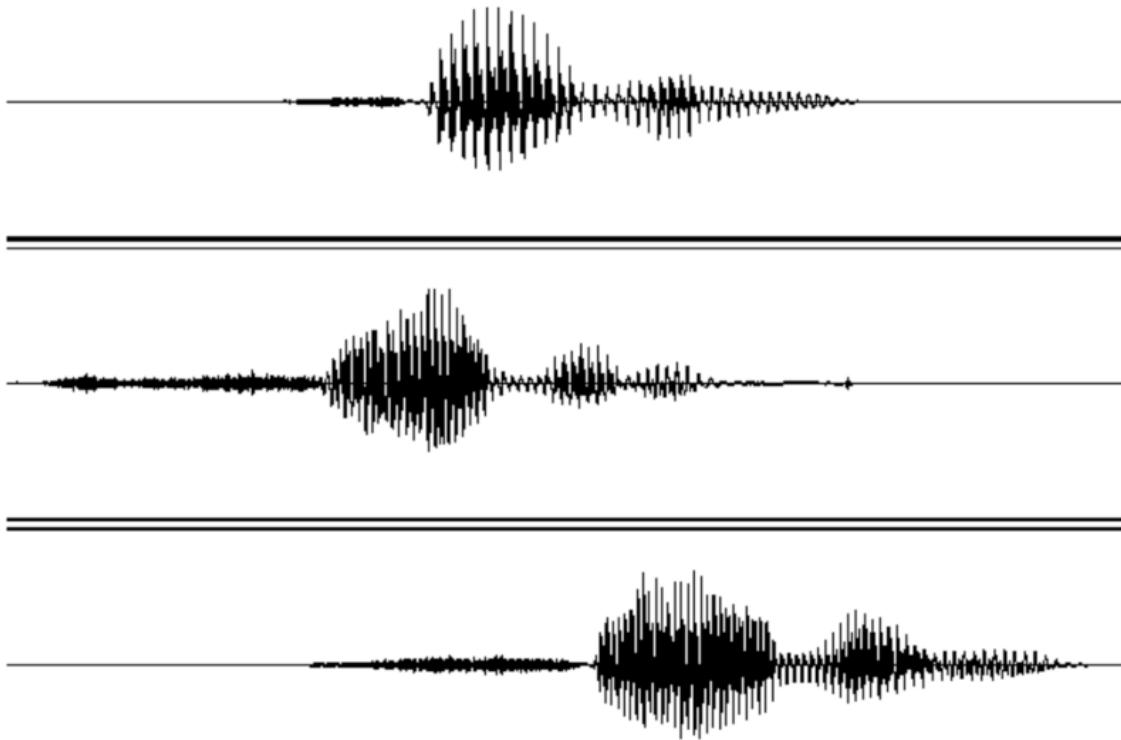
Agenda

- Aggregating features over time
 - Defining fixed time intervals (weeks, levels in a game, etc.)
 - **Dynamic Time Warping**
 - String Metrics
 - Markov Models
 - Additional Practice (if time permits)
-

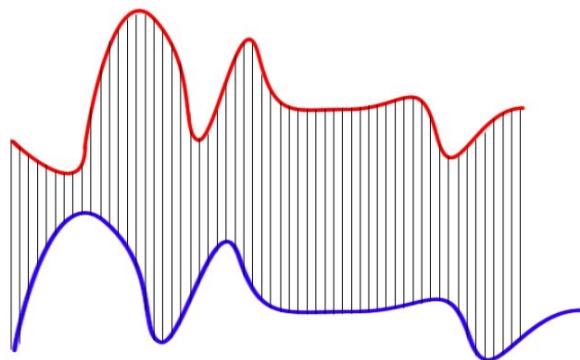
Dynamic Time Warping

- Compute distance between two time series, which may vary in speed
- Time series can have different lengths
- Develop a **one-to-many** match, i.e. find an *optimal alignment* between two time series

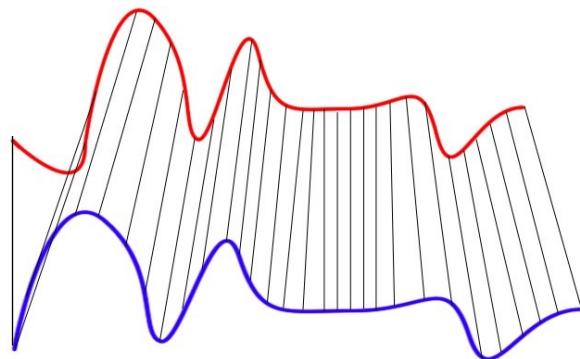
Example: Spoken Digits



Dynamic Time Warping vs. Euclidean Distance



Euclidean Distance



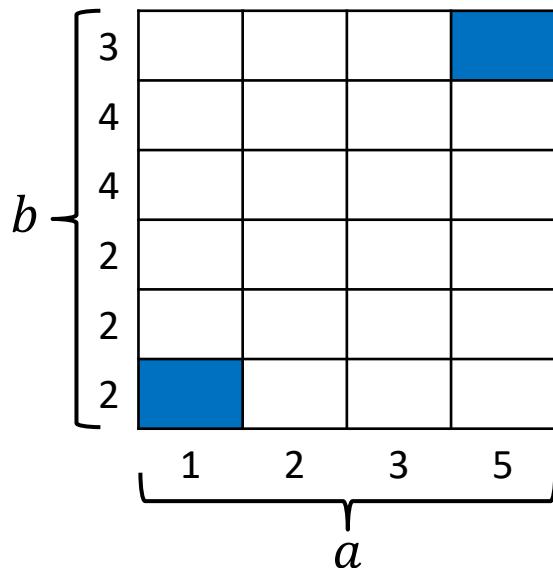
Dynamic Time Warping

Dynamic Time Warping: Rules

- **Goal:** minimize $D(a, b) = \min_{\emptyset} \sum_k d(a_{\emptyset(k)}, b_{\emptyset(k)})$
 - **Rules** (given two sequences a and b):
 - Every index of a must be matched with one or more indices from b , and vice versa
 - The first index from a must be matched with the first index from b (but it does not have to be its only match)
 - The last index from a must be matched with the last index from b (but it does not have to be its only match)
 - The mapping of the indices from a to indices from b must be monotonically increasing, and vice versa, i.e. if $j > i$ are indices from a , then there must not be two indices $m > n$ in b , such that index i is matched with index m and index j is matched with index n , and vice versa
-

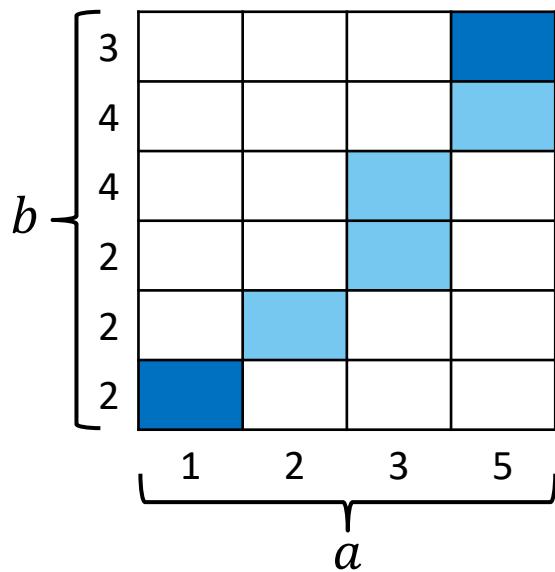
Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



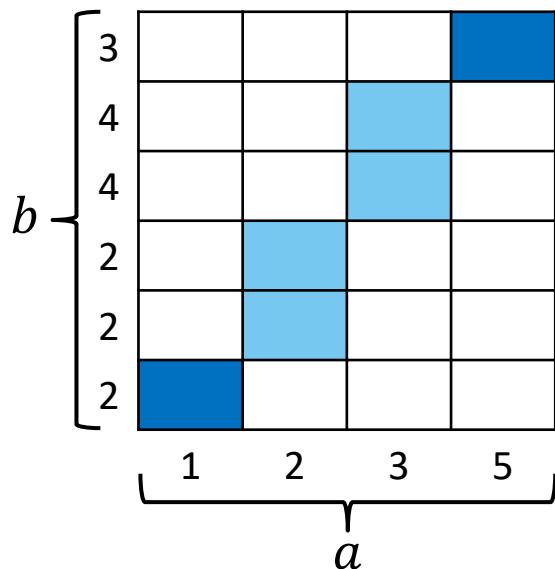
Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



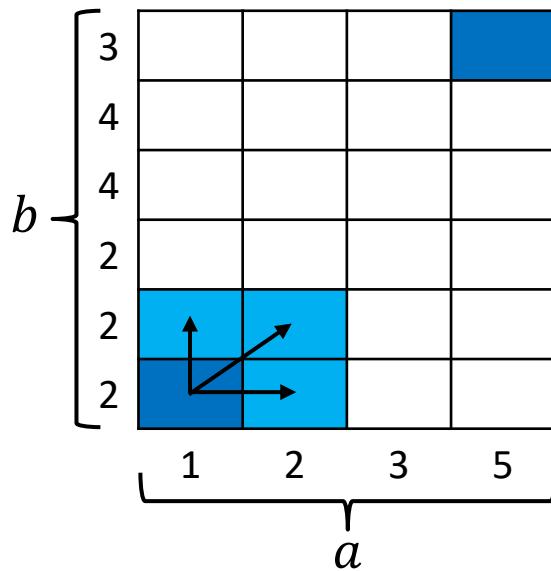
Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



Dynamic Time Warping: Possible Paths

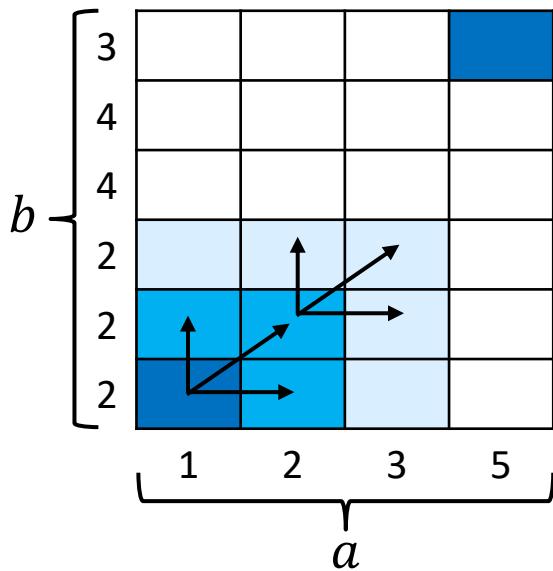
$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



- Three possible paths from each square

Dynamic Time Warping: Possible Paths

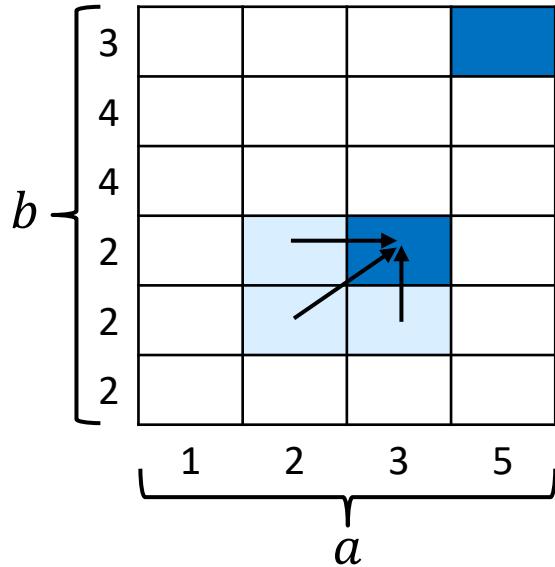
$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



- Three possible paths from each square
 - Every choice leads to three more possible paths
- ➡ $\approx 3^{4 \cdot 6}$ options

Dynamic Time Warping: Minimum Path

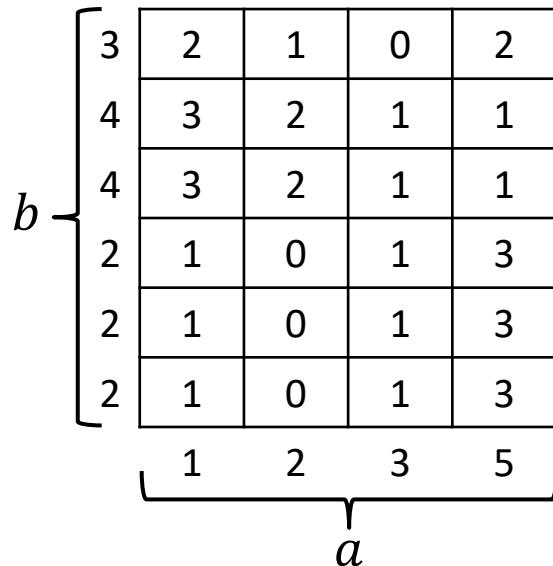
$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



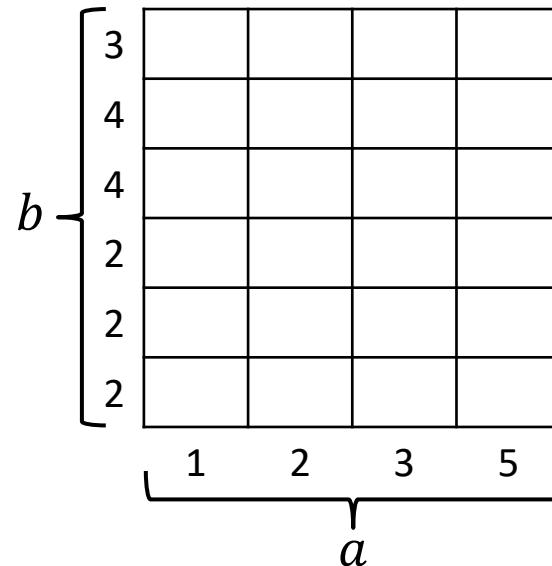
- For any cell C (matching indices i, j): three possible precursor cells
 - Minimum cost (distance) for getting to C
- $$d(i, j) + \min(D(i - 1, j), D(i - 1, j - 1), D(i, j - 1))$$

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



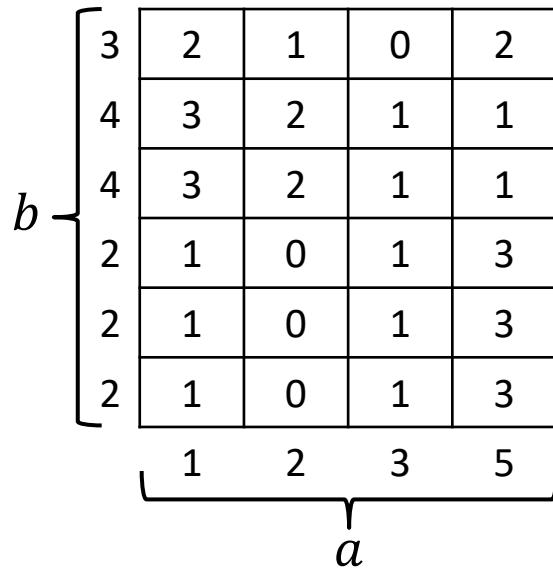
1. Compute pairwise distances



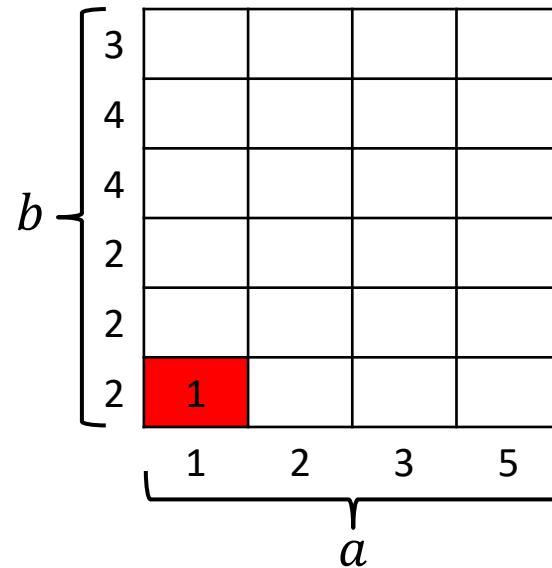
2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



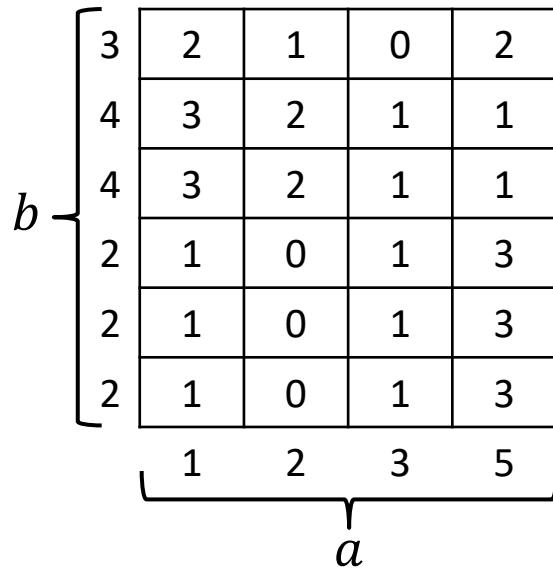
1. Compute pairwise distances



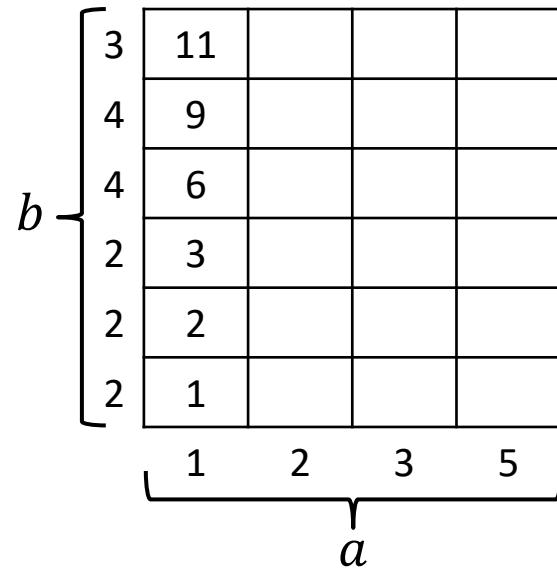
2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



1. Compute pairwise distances



2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	11			
4	9			
4	6			
2	3			
2	2			
2	1	1	2	5

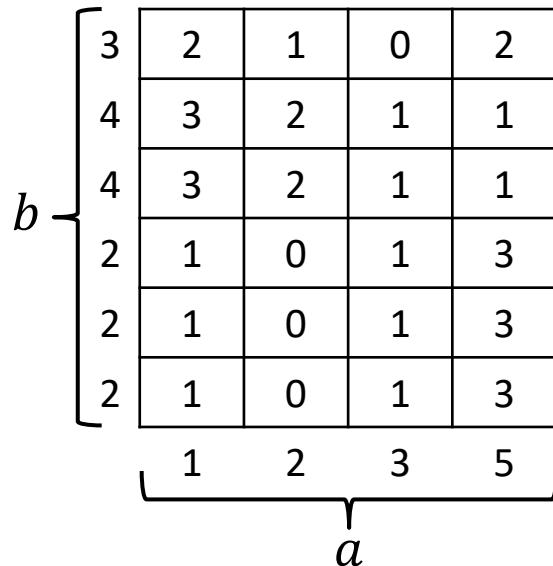
a

1. Compute pairwise distances

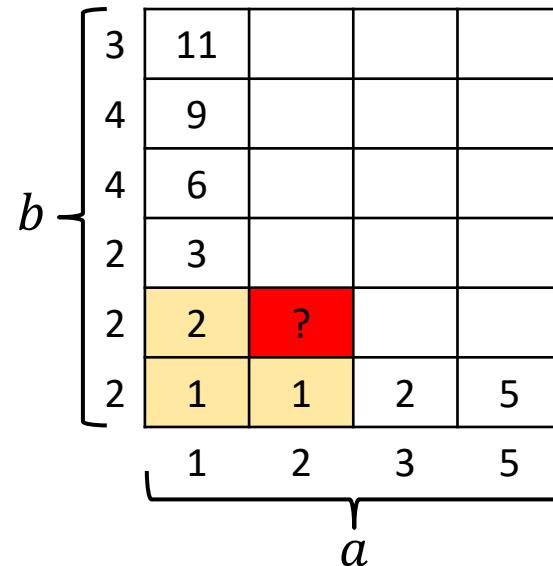
2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



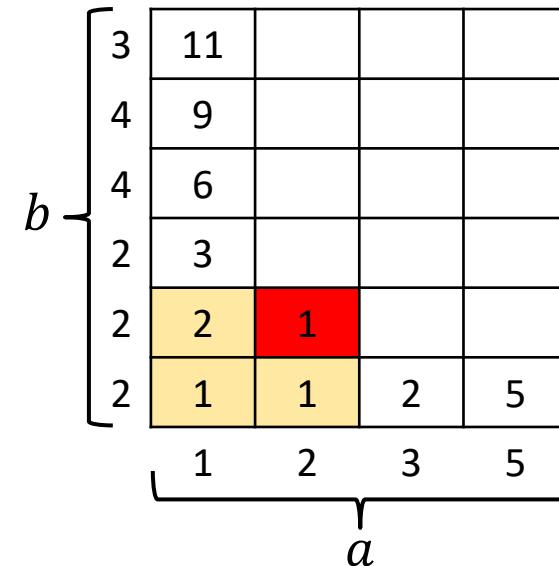
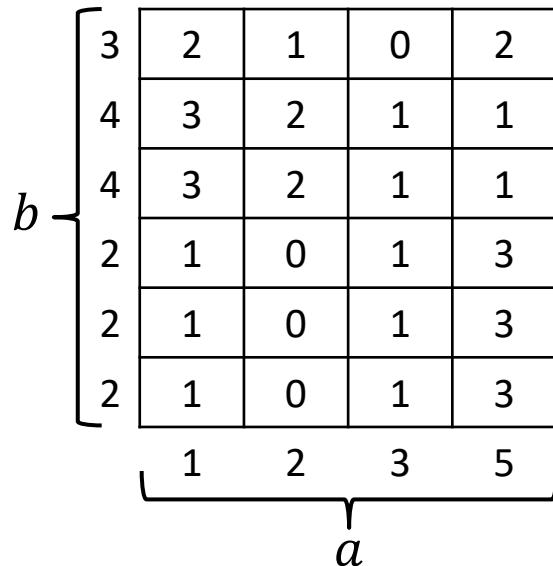
1. Compute pairwise distances



2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

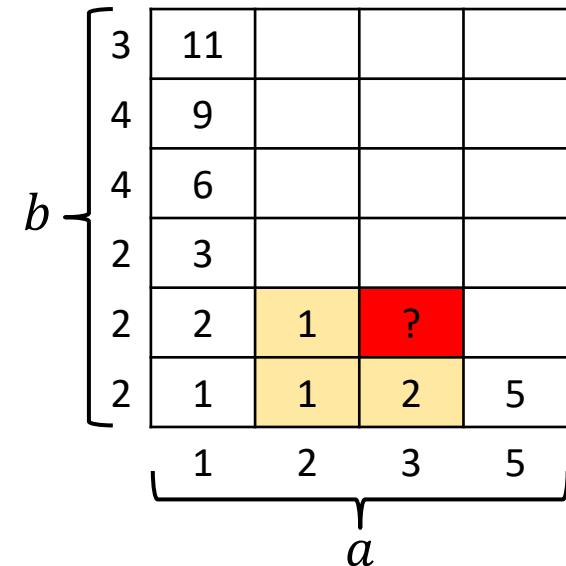
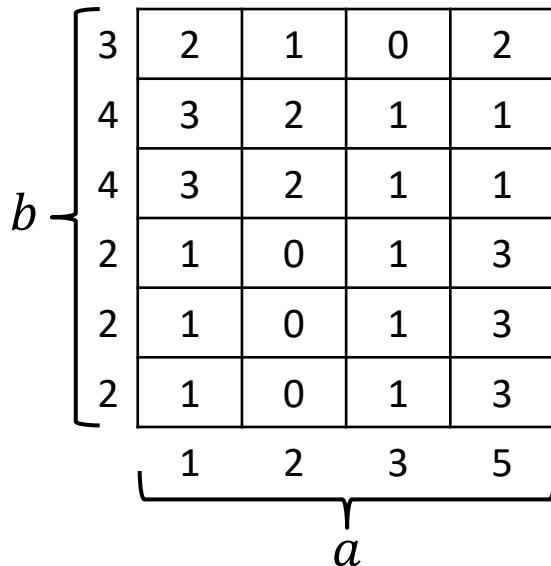


1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	11			
4	9			
4	6			
2	3			
2	2	1	2	
2	1	1	2	5

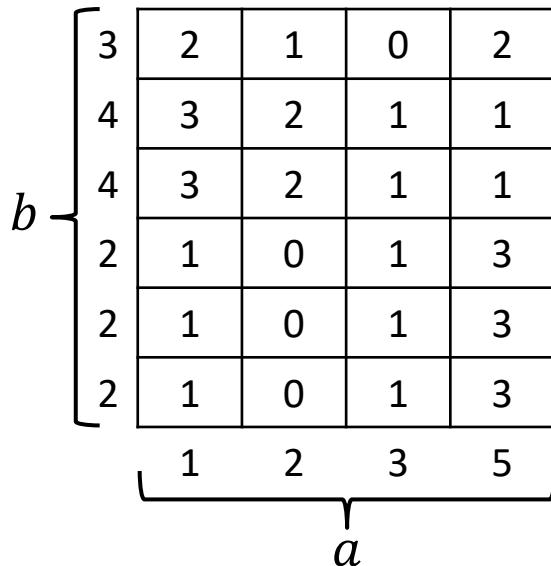
a

1. Compute pairwise distances

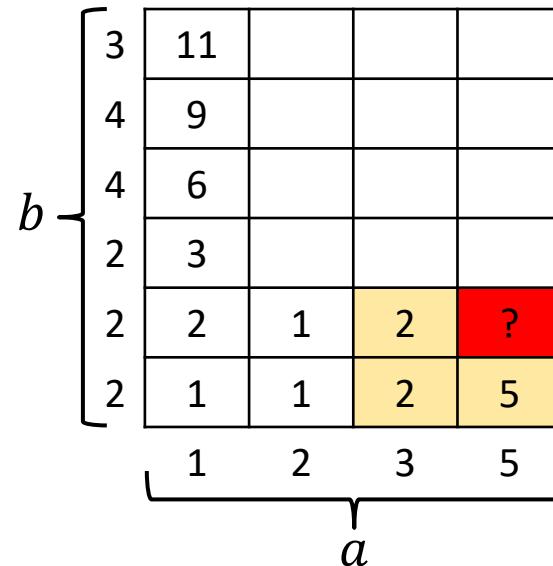
2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



1. Compute pairwise distances



2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	11			
4	9			
4	6			
2	3			
2	2	1	2	5
2	1	1	2	5

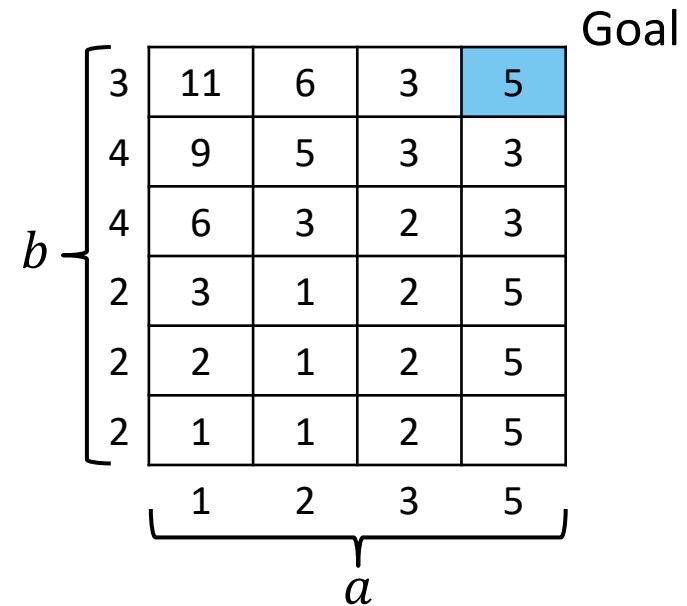
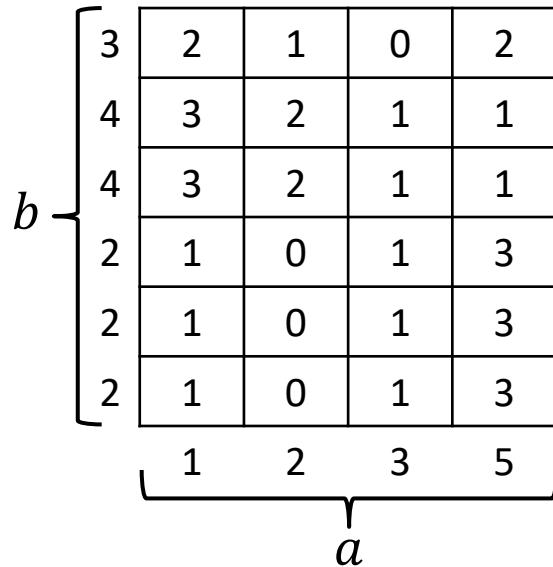
a

1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

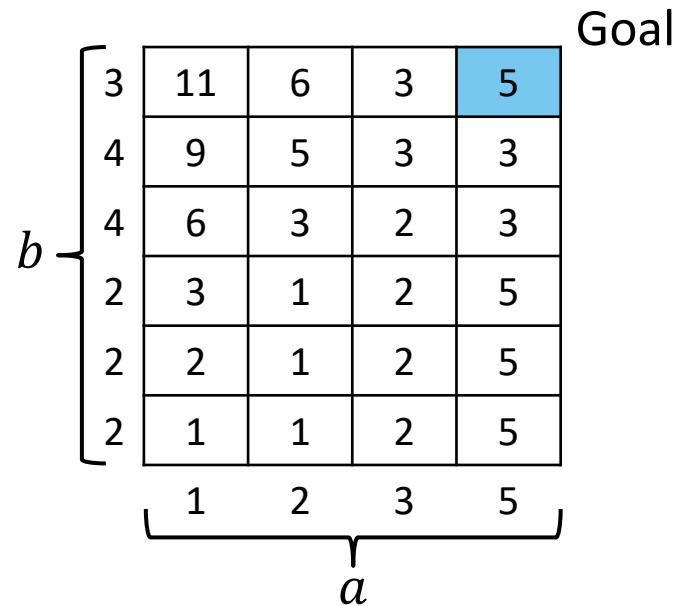


1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: Example

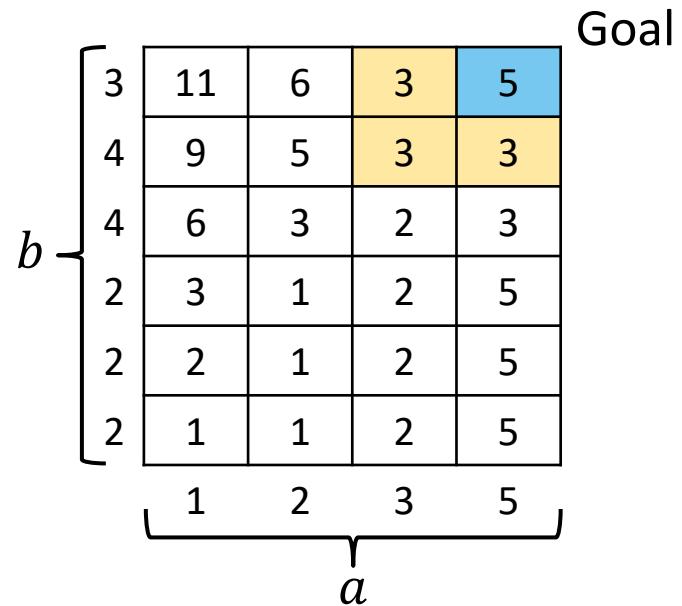
$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



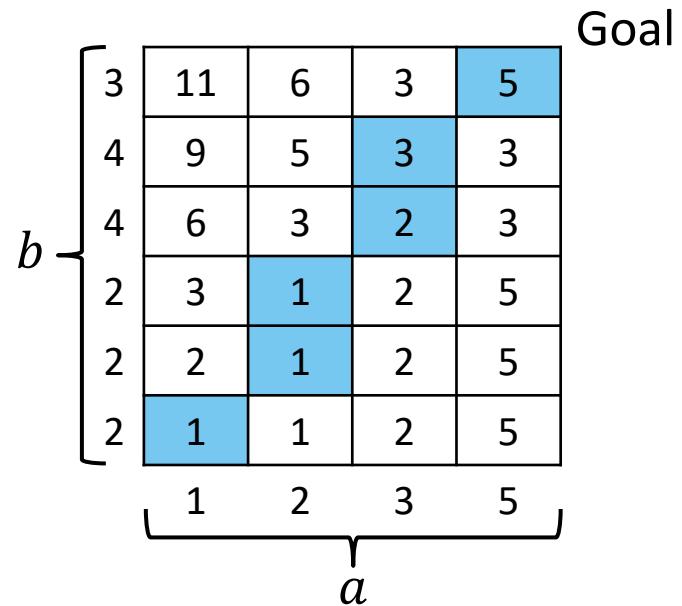
2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

$$a = [1,2,3,5]$$


$$b = [2,2,2,4,4,3]$$

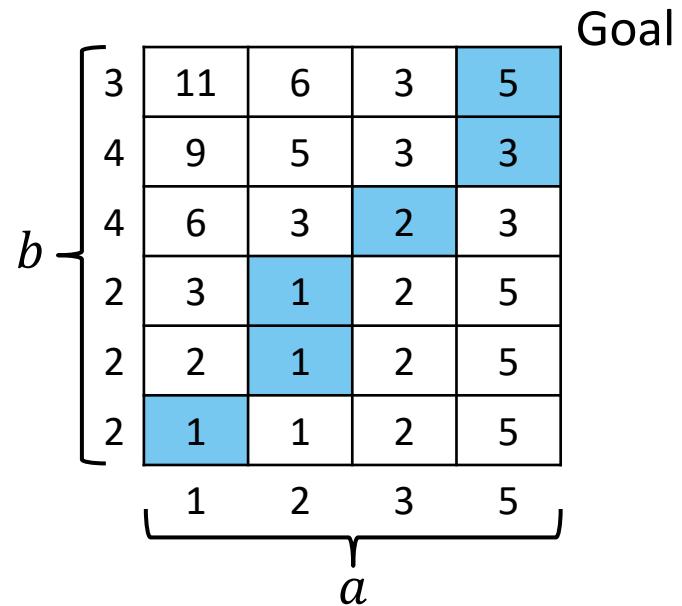


2. Compute minimum distance path

Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

$$a = [1,2,3,5]$$
$$b = [2,2,2,4,4,3]$$



2. Compute minimum distance path

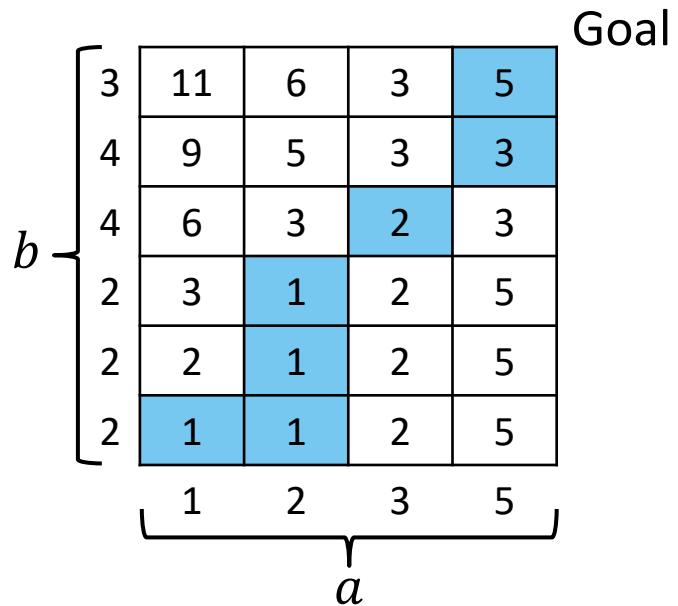
Dynamic Time Warping: Example

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

$$a = [1,2,3,5]$$

The diagram shows two sequences, a and b . Sequence a is $[1,2,3,5]$ and sequence b is $[2,2,2,4,4,3]$. Blue arrows connect the elements of a to the elements of b , illustrating a possible warping path.

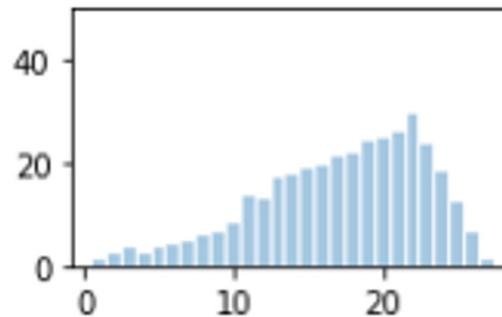
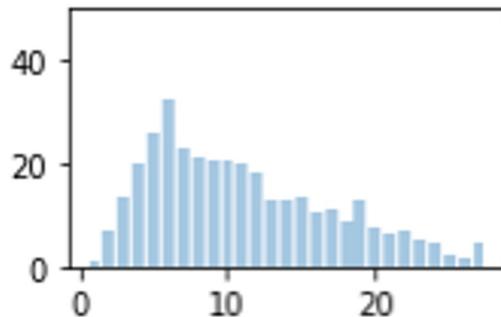
$$b = [2,2,2,4,4,3]$$



2. Compute minimum distance path

Dynamic Time Warping: Window

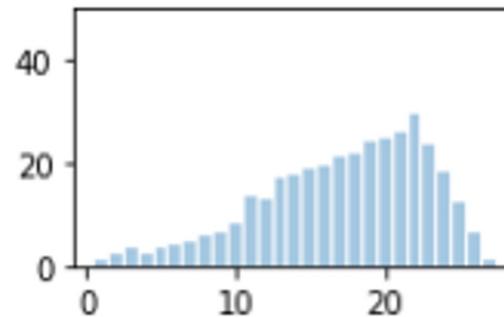
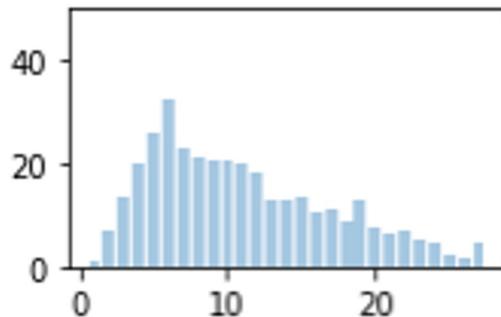
- Sometimes, we might want to constrain the mapping



.

Dynamic Time Warping: Window

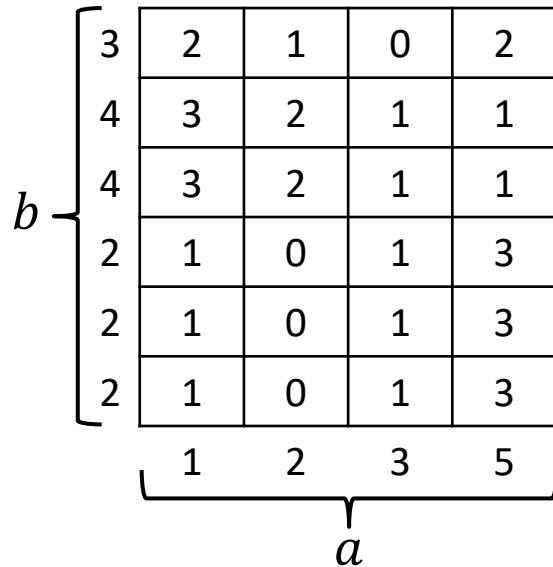
- Sometimes, we might want to constrain the mapping



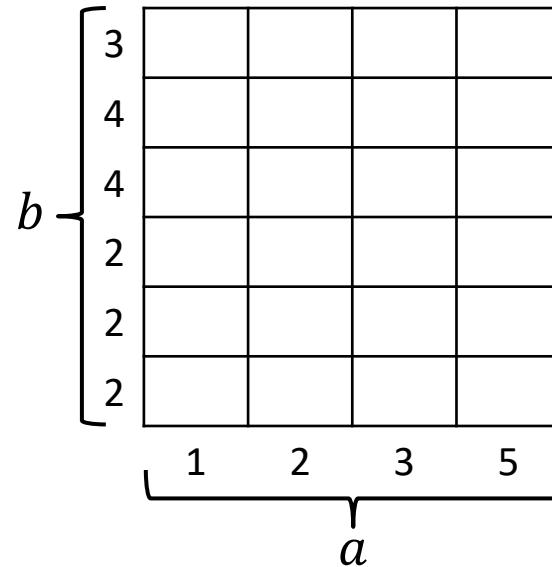
- We introduce a **window size w** : an element in sequence a at index i can only be mapped to elements at index $i - w, \dots, i + w$ in sequence b

Dynamic Time Warping: $w = 2$

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



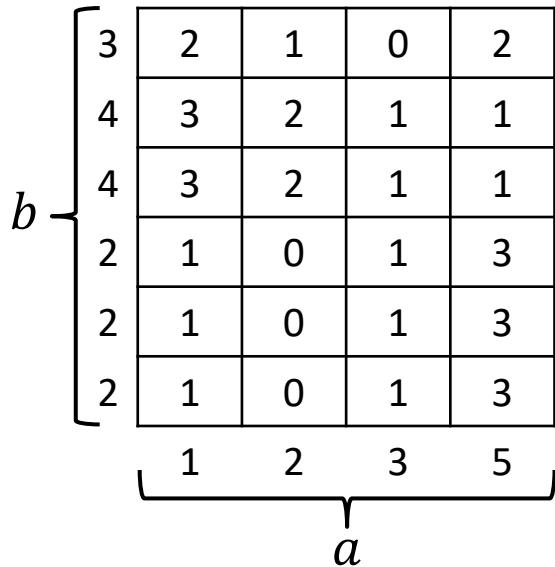
1. Compute pairwise distances



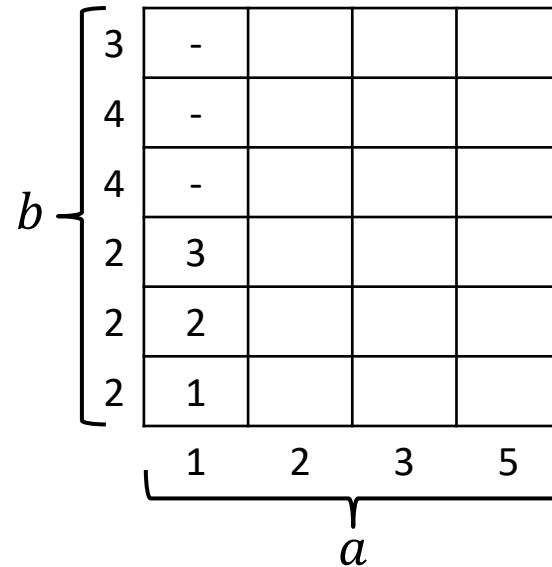
2. Compute minimum distance path

Dynamic Time Warping: $w = 2$

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$



1. Compute pairwise distances



2. Compute minimum distance path

Dynamic Time Warping: $w = 2$

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	-	-		
4	-	-		
4	-	3		
2	-	1		
2	2	1		
2	1	1		

a

1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: $w = 2$

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	-	-	-	
4	-	-	3	
4	-	-	2	
2	-	1	2	
2	2	1	2	
2	1	1	2	

a

1. Compute pairwise distances

2. Compute minimum distance path

Dynamic Time Warping: $w = 2$

$$a = [1,2,3,5] \quad b = [2,2,2,4,4,3]$$

b

3	2	1	0	2
4	3	2	1	1
4	3	2	1	1
2	1	0	1	3
2	1	0	1	3
2	1	0	1	3

a

b

3	-	-	-	5
4	-	-	-	3
4	-	-	2	3
2	-	1	2	5
2	2	1	2	5
2	1	1	2	-

a

1. Compute pairwise distances

2. Compute minimum distance path

Your Turn – Dynamic Time Warping

Run spectral clustering using DTW with a window size of $w = 3$:

- How do the results differ from previous results?
- What happens if you set $w = 0$?
- And if you set $w = 27$?

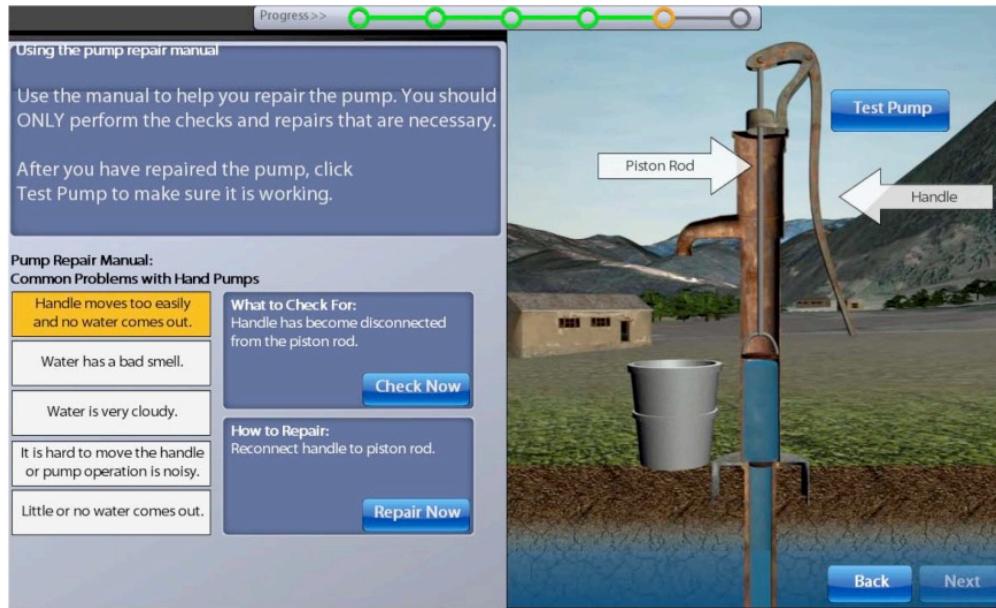


Agenda

- Aggregating features over time
- Defining fixed time intervals (weeks, levels in a game, etc.)
- Dynamic Time Warping
- **String Metrics**
- Markov Models



Example from Research: String Metrics



$C1 \rightarrow C2 \rightarrow C3 \rightarrow C4 \rightarrow R4 \rightarrow P \rightarrow C5 \rightarrow R5 \rightarrow P$

$C4 \rightarrow R4 \rightarrow P \rightarrow C5 \rightarrow R5 \rightarrow P$

Example from Research: String Metrics

- Levenshtein distance: minimal number of single character edits (insertion, deletion, substitution) to change one string into the other
- Longest common subsequence (LCS): string similarity measure, find the longest common subsequence between two sequences

Agenda

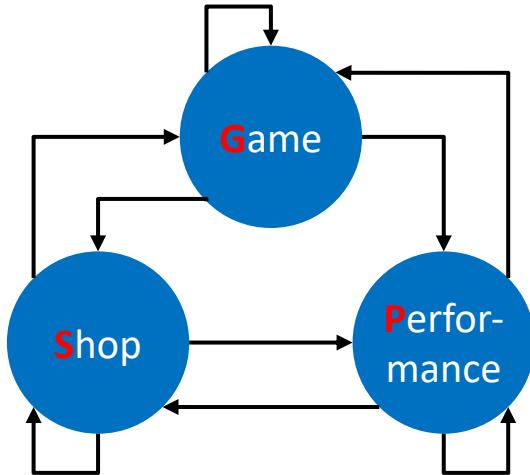
- Aggregating features over time
- Defining fixed time intervals (weeks, levels in a game, etc.)
- Dynamic Time Warping
- String Metrics
- **Markov Models**



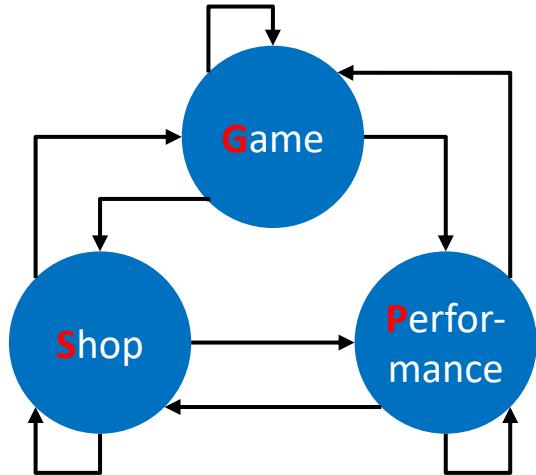
Markov Models

- Detailed action sequences provide rich temporal information
 - Might contain a considerable amount of noise
 - We might be interested not in the detailed sequence, but in patterns (which actions tend to follow each other)
-

Markov Models



Markov Models



$G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow S \rightarrow G$

$G \rightarrow G \rightarrow G \rightarrow G \rightarrow P \rightarrow G \rightarrow G$

$G \rightarrow P \rightarrow S \rightarrow G \rightarrow P \rightarrow S$

Parameters: Maximum Likelihood Estimation

$G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow S \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow S \rightarrow G \rightarrow S$

$$p(S|G) = \frac{10}{15} = 0.67$$

$$p(G|G) = \frac{2}{15} = 0.13$$

$$p(P|G) = \frac{3}{15} = 0.20$$

$$\begin{matrix} & G & S & P \\ G & \left(\begin{array}{ccc} 0.13 & 0.67 & 0.20 \end{array} \right) \\ S & \left(\begin{array}{ccc} 0.79 & 0.11 & 0 \end{array} \right) \\ P & \left(\begin{array}{ccc} 0.33 & 0.67 & 0 \end{array} \right) \end{matrix}$$

Stationary Distribution

$G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S$

$$\begin{array}{ccc} & G & S & P \\ G & \left(\begin{array}{ccc} 0.13 & 0.67 & 0.20 \\ 0.89 & 0.11 & 0 \\ 0.33 & 0.67 & 0 \end{array} \right) \end{array}$$

$$\pi^T = \pi$$

Stationary Distribution

$G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow P \rightarrow G \rightarrow P \rightarrow S \rightarrow G \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S \rightarrow G \rightarrow S$

$$\begin{array}{ccc} & G & S & P \\ G & \left(\begin{array}{ccc} 0.13 & 0.67 & 0.20 \\ 0.89 & 0.11 & 0 \\ 0.33 & 0.67 & 0 \end{array} \right) \\ S & & & \\ P & & & \end{array}$$

$$\pi^T = \pi$$

$$\pi = [0.48 \quad 0.43 \quad 0.09]$$

Expected Frequencies

- When sequences get very long (n gets large), how often do we expect to observe the transitions?

$$\begin{array}{ccc} & G & S & P \\ G & \left(\begin{array}{ccc} 0.06 & 0.32 & 0.10 \end{array} \right) \\ S & \left(\begin{array}{ccc} 0.38 & 0.05 & 0 \end{array} \right) \\ P & \left(\begin{array}{ccc} 0.03 & 0.06 & 0 \end{array} \right) \end{array}$$

Distance Metrics

- Based on **Frobenius Norm**: equivalent to Euclidean distance over vectors

$$D_2(A, B) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m (a_{ij} - b_{ij})^2}$$

Distance Metrics

- Kullback-Leibler Divergence: measures difference between two probability distributions

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \cdot \log\left(\frac{P(x)}{Q(x)}\right)$$

- Jensen-Shannon Divergence: measures difference between two probability distributions

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(M||Q) \quad M = \frac{1}{2}(P + Q)$$

Distance Metrics

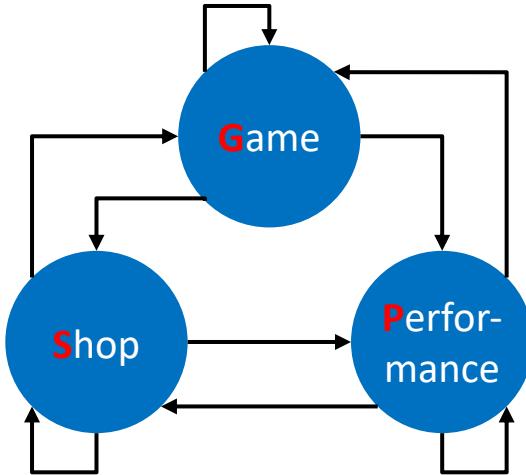
- Hellinger Distance: measures difference between two probability distributions

$$D_H(P||Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2}$$

Distance between samples: Options

- Compute distance between stationary distributions:
use Hellinger Distance (or Jensen-Shannon Divergence)
 - Compute distance between transition matrices: use Frobenius Distance
 - Compute distance between expected frequencies:
use Hellinger Distance (or Jensen-Shannon Divergence)
-

Example from Research: Spelling Learning

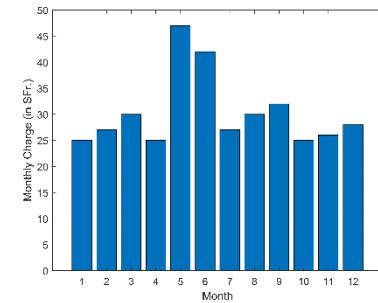


Three clusters:

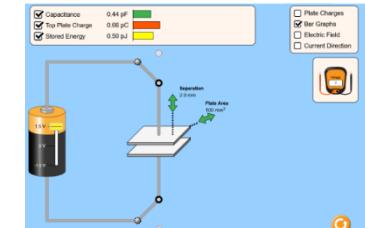
- Focused on the task
- Children, who frequently check performance/shop in-between tasks
- Spend long amounts of time off-task

Summary - Handling Time Series Data

1. Aggregating features over time
2. Defining fixed time intervals (weeks, levels in a game, etc.)
3. Dynamic Time Warping



-
4. String Measures
 5. Markov Models



Action Sequences

Fairness

Machine Learning for Behavioral Data
May 8, 2023

Today's Topic

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Fairness
13	Explainability
14	Project Presentations
15	Whit Monday



- What is fairness?
- Fairness metrics
- Interpreting neural networks

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



Agenda

- 1) Introduction to fairness – Cécile Hardebolle
- 2) Fairness in machine learning:
 - Sources of unfairness
 - Fairness metrics – evaluating model predictions
- 3) Example on real world data



Agenda

- 1) Introduction to fairness – Cécile Hardebolle
- 2) Fairness in machine learning:
 - Sources of unfairness
 - Fairness metrics – evaluating model predictions
- 3) Example on real world data

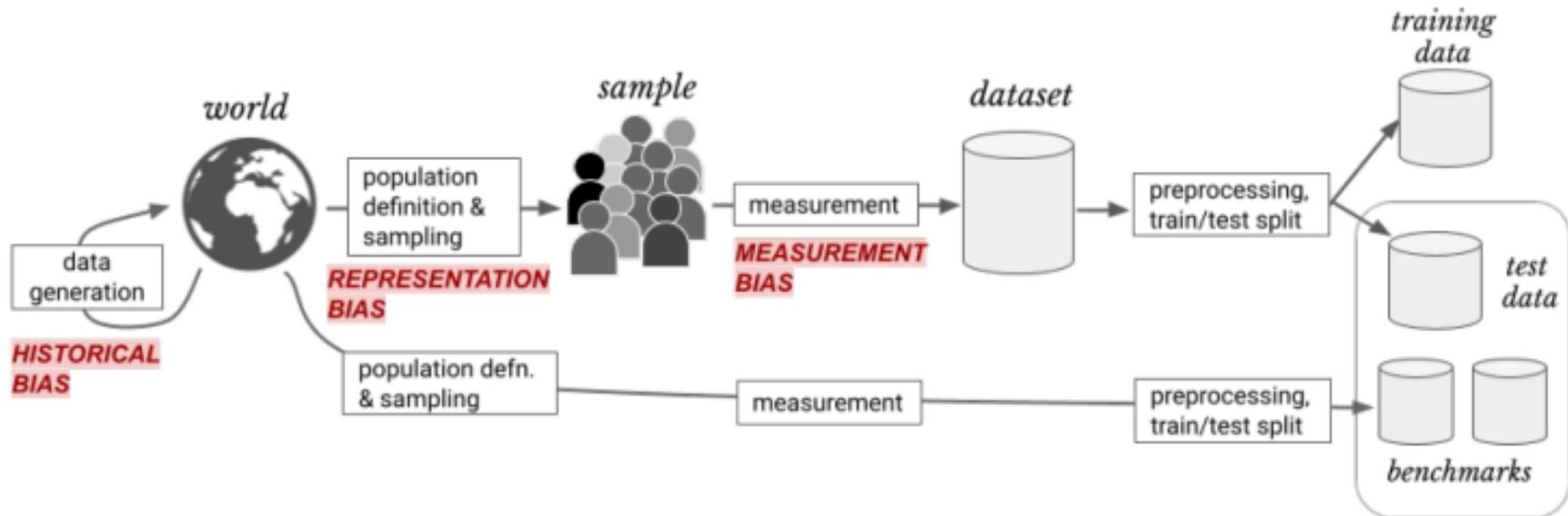


Learning Objectives

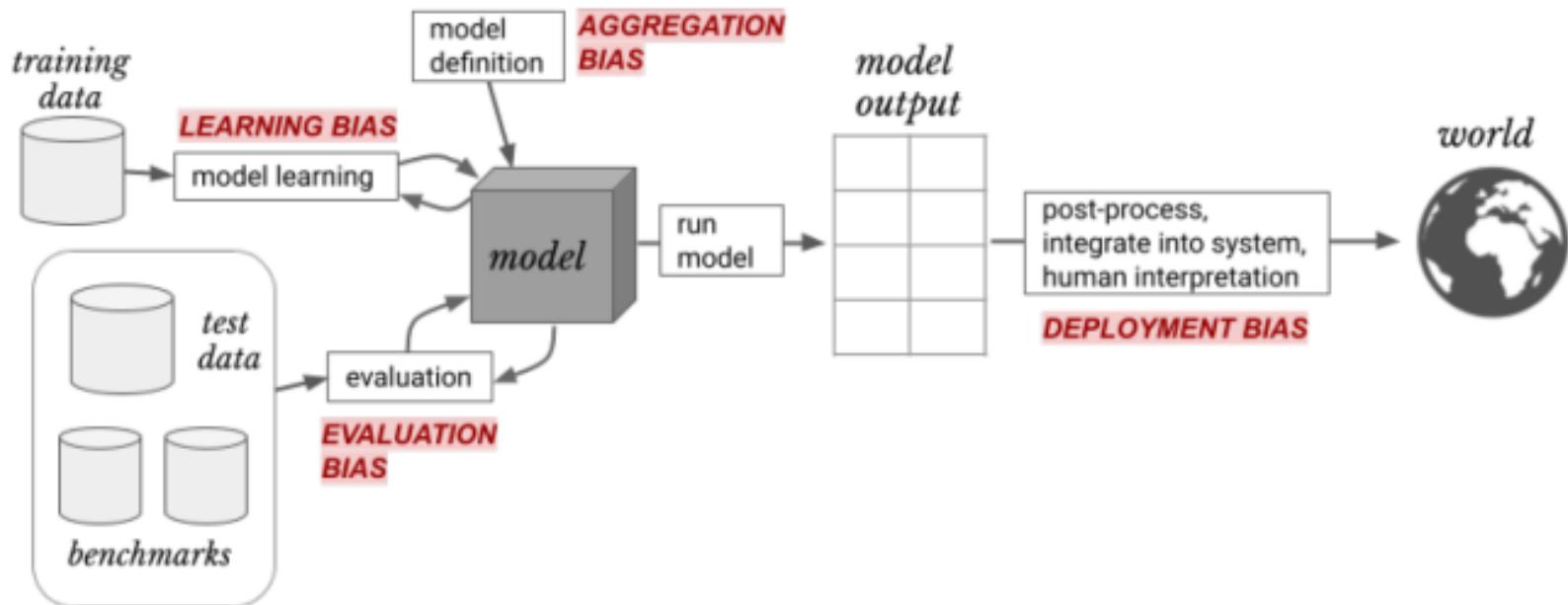
You should be able to:

- Name and explain the sources of unfairness in a machine learning pipeline
 - Explain and implement the most popular metrics for fairness
 - Perform a fairness evaluation of a machine learning model using an appropriate fairness metric
-

Sources of Unfairness – Data Generation

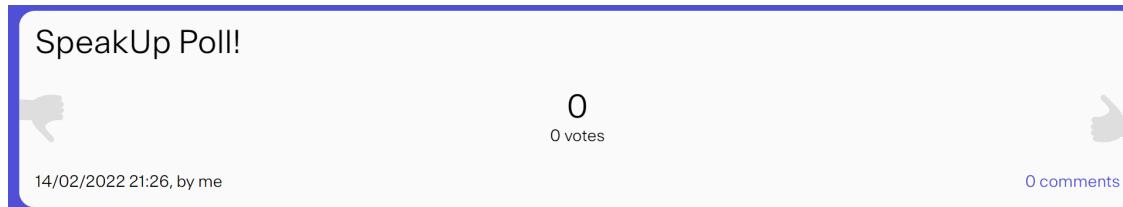


Sources of Unfairness – Model Building



Fairness Through Blindness

- Idea: we ignore all protected attributes in our model (e.g., we do not use protected attributes such as gender, race, etc. as features)



Will this idea lead to a fair model?

- a) Yes
- b) No

Fairness Through Awareness

- There is not one mathematically agreed definition of fairness
- Popular fairness metrics are
 - model-agnostic
 - defined for classification problems

Problem Formalization

Notation:

- X is the input to the model
- \hat{Y} is the prediction of the model
- T is the true label
- A is the protected attribute



Confusion Matrix

		True Label	
		$T = 1$	$T = 0$
Predicted Label	$Y = 1$	True Positive (TP)	False Positive (FP)
	$Y = 0$	False Negative (FN)	True Negative (TN)

Demographic Parity

- Requires equal proportion of positive predictions in each group

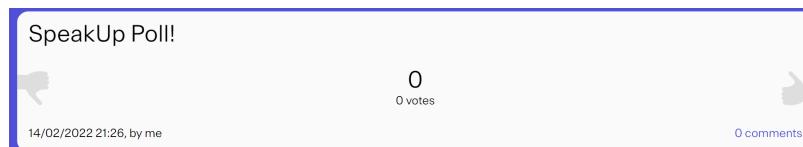
$$p(\hat{Y} = 1 | A = 1) = p(\hat{Y} = 1 | A = 0)$$

Demographic Parity - Example

- Admittance to *Fruits* University
- Students from two schools apply: *Apple* and *Peach*
- School *Peach* has a better program, resulting in more qualified students

<i>Peach</i>	Qualified	Unqualified
Admitted	45	2
Rejected	45	8

<i>Apple</i>	Qualified	Unqualified
Admitted	5	18
Rejected	5	72



Is demographic parity fulfilled?

- a) Yes b) No

Equalized Odds

- For any label and attribute, a classifier predicts the label equally well for all values of that attribute

$$p(\hat{Y} = 1 | A = 1, T = 1) = p(\hat{Y} = 1 | A = 0, T = 1)$$

$$p(\hat{Y} = 1 | A = 1, T = 0) = p(\hat{Y} = 1 | A = 0, T = 0)$$

Equalized Odds - Example

- Admittance to *Fruits* University
- Students from two schools apply: *Apple* and *Peach*
- School *Peach* has a better program, resulting in more qualified students

<i>Peach</i>	Qualified	Unqualified
Admitted	45	2
Rejected	45	8

<i>Apple</i>	Qualified	Unqualified
Admitted	5	18
Rejected	5	72



Are equalized odds fulfilled?

- a) Yes b) No

Predictive Value Parity

- Probability of a sample with positive (negative) predictive value to truly belong to the positive (negative) class should be the same across attributes

$$p(T = 1|A = 1, \hat{Y} = 1) = p(T = 1|A = 0, \hat{Y} = 1)$$

$$p(T = 0|A = 1, \hat{Y} = 0) = p(T = 0|A = 0, \hat{Y} = 0)$$

Predictive Value Parity - Example

- Admittance to *Fruits* University
- Students from two schools apply: *Apple* and *Peach*
- School *Peach* has a better program, resulting in more qualified students

<i>Peach</i>	Qualified	Unqualified
Admitted	45	2
Rejected	45	8

<i>Apple</i>	Qualified	Unqualified
Admitted	5	18
Rejected	5	72



Is predictive value parity fulfilled?

- a) Yes b) No

Impossibility Result

- Any two of the three criteria are mutually exclusive
 1. If A and T are not independent, then *demographic parity* and *predictive value parity* cannot simultaneously hold
 2. If A and \hat{Y} are not independent of T , then *demographic parity* and *equalized odds* cannot simultaneously hold
 3. If A and T are not independent, then *equalized odds* and *predictive value parity* cannot simultaneously hold
-

Impossibility Result

Note that these requirements hold for *most* classifiers in real contexts:

- Base-rates of outcomes rarely are equal across groups
- A and T are usually associated when issues of fairness are relevant for the group in question
- \hat{Y} and T are usually associated, if your classifier is any good



Agenda

- 1) Introduction to fairness – Cécile Hardebolle
- 2) Fairness in machine learning:
 - Sources of unfairness
 - Fairness metrics – evaluating model predictions
- 3) Example on real world data



Flipped Classroom – Your Turn

- Participants: 214 EPFL students of a course taught in *flipped classroom* mode with a duration of 10 weeks
- We have trained a classifier to predict whether a student will pass or fail the course based on their clickstream data
- Your task:
 1. Choose one of the fairness metrics introduced in class and compute the metric for the flipped classroom classifier
 2. Tell us: is the classifier fair according to the selected metric? Why did you choose this metric?

Summary

- There are multiple sources of unfairness in a machine learning pipeline
 - There is no consensus on the mathematical definition of fairness metrics
 - Different metrics assess different aspects of the classifier
 - Often, fairness metrics are mutually exclusive
 - Fairness evaluation of a classifier includes exploration of relevant characteristics of our data
-

Explainability

Machine Learning for Behavioral Data
May 25, 2023

Today's Topic

Week	Lecture/Lab
8	Spring Break
9	Time Series Prediction
10	Unsupervised Learning
11	Unsupervised Learning
12	Fairness
13	Explainability
14	Project Presentations
15	Whit Monday



- What is fairness?
- Fairness metrics
- Interpreting neural networks

Getting ready for today's lecture...

- **If not done yet:** clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace.
- SpeakUp room for today's lecture:

<https://go.epfl.ch/speakup-mlbd>



Short quiz about the past...

In K-Means Clustering, how should you initialize the cluster centroids?

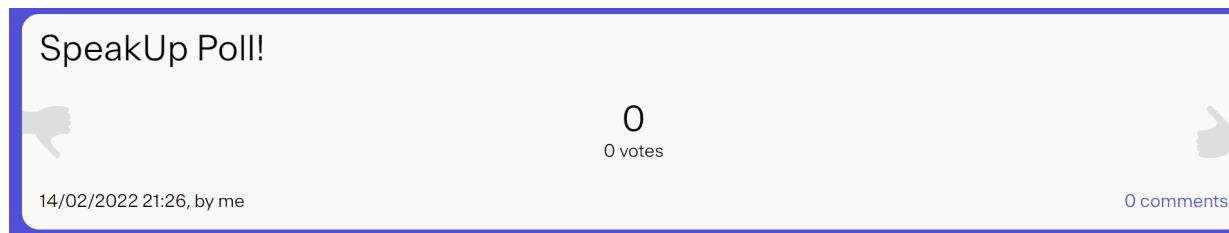
- a) Once, randomly
- b) Once, uniformly
- c) Visualizing the data and picking appropriate starting points
- d) Multiple times randomly and minimizing distortion

SpeakUp Poll!

0
0 votes

14/02/2022 21:26, by me

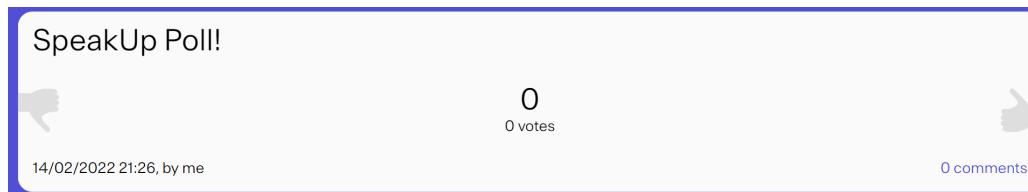
0 comments



Short quiz about the past...

When performing clustering on text data, which distance/similarity metric is appropriate?

- a) Silhouette Score
- b) Jaccard Similarity
- c) Cosine Similarity
- d) Euclidean Distance



Short quiz about the past...

If you use accuracy instead of balanced accuracy for a binary classification task (on an imbalanced data set), this is an example of:

- a) Historic Bias
- b) Evaluation Bias
- c) Measurement Bias
- d) Aggregation Bias



Short quiz about the past...

You are building a model for whether someone will pass a class based on their MOOC clickstream. You are concerned about whether your model's predictions of passing and predictions of failing are equally accurate across demographic groups. Which metric do you use?

- a) equalized odds
- b) demographic parity
- c) predictive (value) parity



Agenda

1) Introduction to Explainability

- Taxonomy of interpretability methods
- Deep Dive: PDP
- Deep Dive: LIME

2) Course Wrap-Up (project, exam)



Learning Objectives

You should be able to:

- Describe and categorize the explainability methods discussed in class
 - Explain their strength and weaknesses
 - Interpret their outputs
 - Apply the methods (using the APIs) to predictions of a model and discuss the results
-

Interpretability

Interpretability is the degree to which a human can understand the cause of a decision.

Interpretability

Interpretability is the degree to which a human can understand the cause of a decision.



The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made

Interpretability in Education



Taxonomy of Interpretability Methods

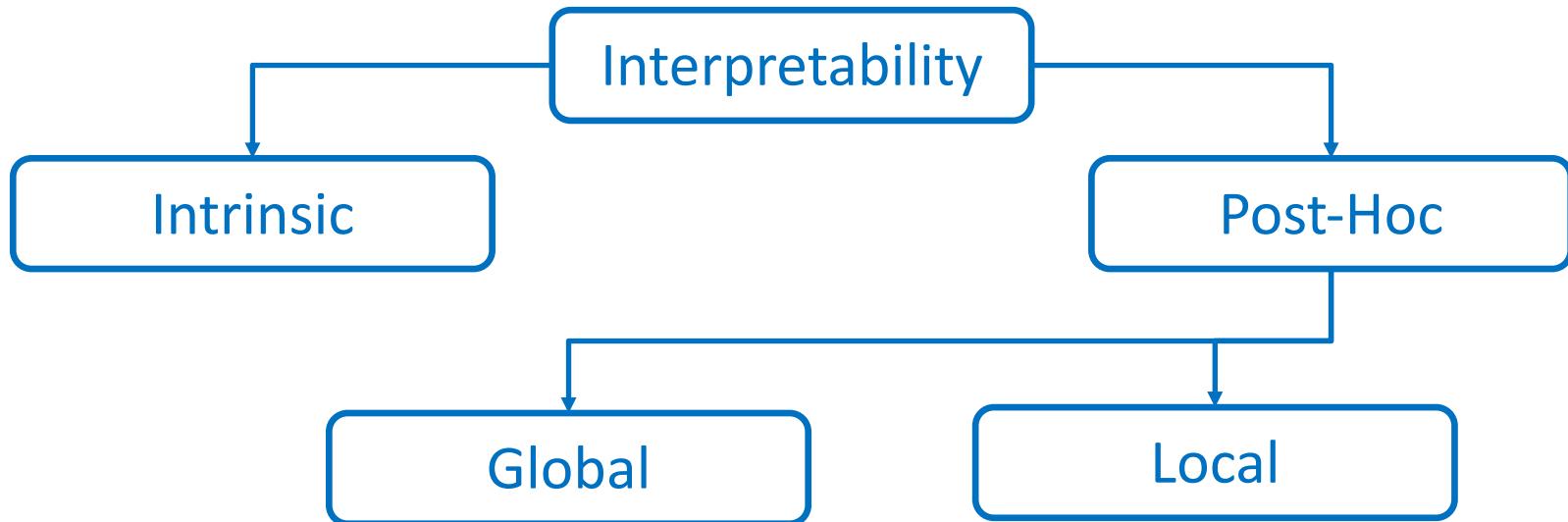


Taxonomy of Interpretability Methods

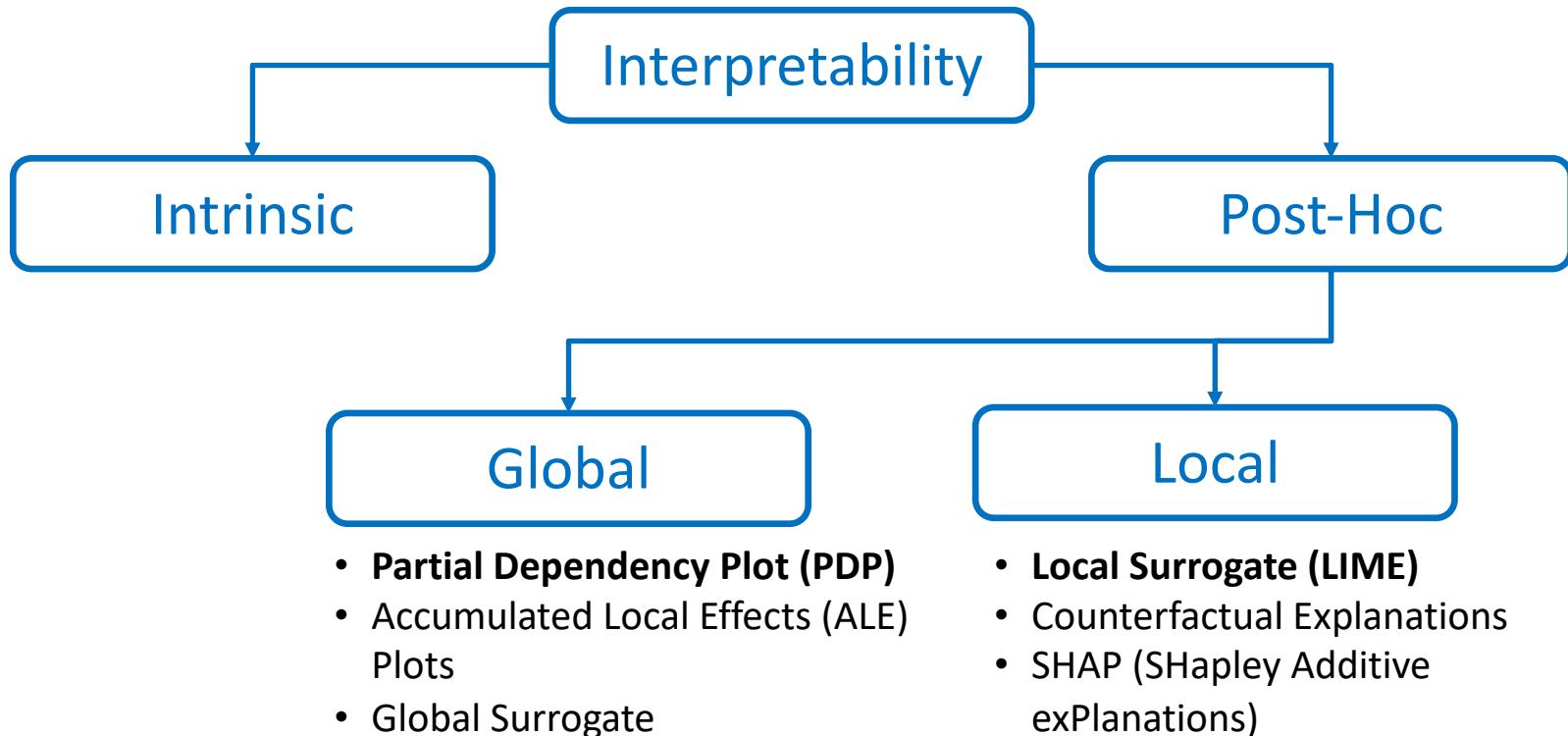


- Linear Regression
- Generalized Linear Models
(e.g., logistic regression)
- Decision Trees
- (k-Nearest Neighbors)

Taxonomy of Interpretability Methods



Taxonomy of Interpretability Methods



Global Method: Partial Dependency Plot (PDP)

- PDP is model-agnostic
- PDP show the marginal effects a subset of features have on the predicted outcome of a model
- The subset of features usually consists of one feature (resulting in a 2D-Plot) or two features (resulting in a 3D-Plot)

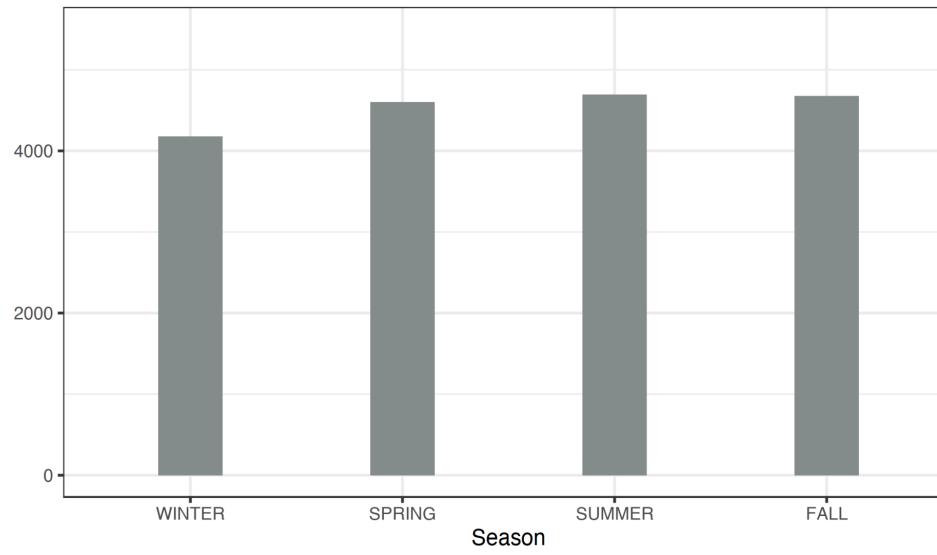
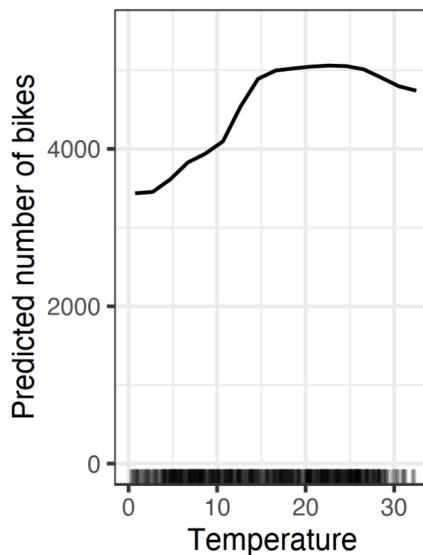


Example – Bike Rental Shop

- Y denotes the number of bikes that will be rented on a given day
- Features (X): season, work day, temperature, humidity, ...
- Given: model f such that $y = f(x)$

Example – Bike Rental Shop

- Y denotes the number of bikes that will be rented on a given day
- Features (X): season, work day, temperature, humidity, ...
- Given: model f such that $y = f(x)$



Partial Function - Regression

$$\widehat{f}_S(x_S) = E_{X_C}[\widehat{f}_S(x_S, X_C)] = \int_{X_C} \widehat{f}_S(x_S, X_C)$$

Partial Function - Regression

$$\widehat{f}_S(x_S) = E_{X_C}[\widehat{f}_S(x_S, X_C)] = \int_{X_C} \widehat{f}_S(x_S, X_C)$$



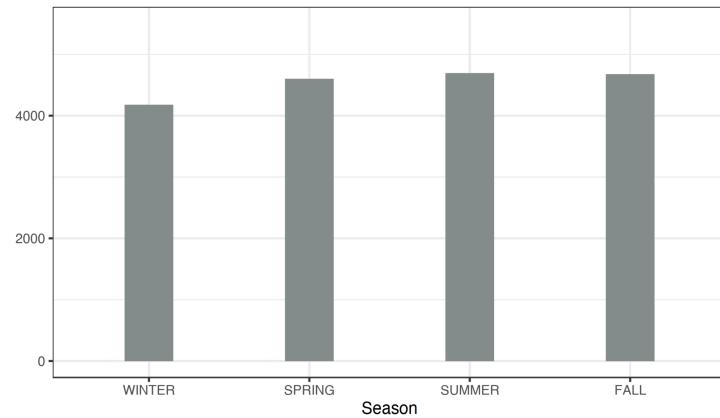
$$\widehat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \widehat{f}_S(x_S, x_c^{(i)})$$

Partial Function - Regression

$$\widehat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \widehat{f}_S \left(x_S, x_c^{(i)} \right)$$

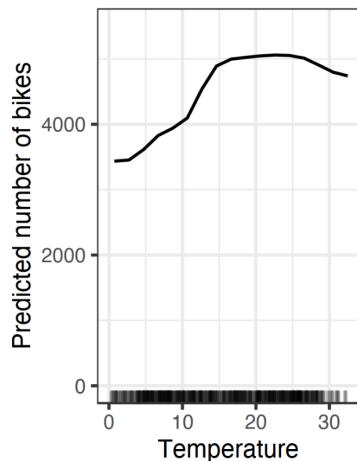
Partial Function - Regression

$$\widehat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \widehat{f}_S(x_S, x_c^{(i)})$$



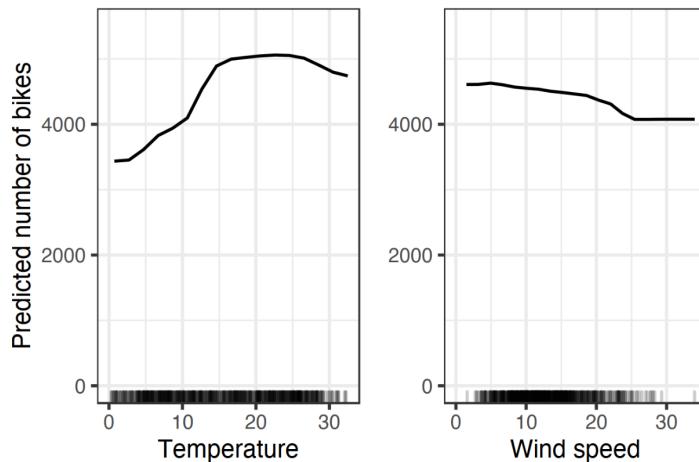
Partial Function - Regression

$$\widehat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \widehat{f}_S(x_S, x_c^{(i)})$$



Partial Function - Regression

$$\hat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}_S(x_S, x_c^{(i)})$$



Partial Function - Classification

$$\widehat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \widehat{f}_S \left(x_S, x_c^{(i)} \right)$$

- If classifier outputs a probability, the PDP displays the probability for a certain class given different values for feature(s) in S
- Dealing with multiple classes: draw one line or plot per class

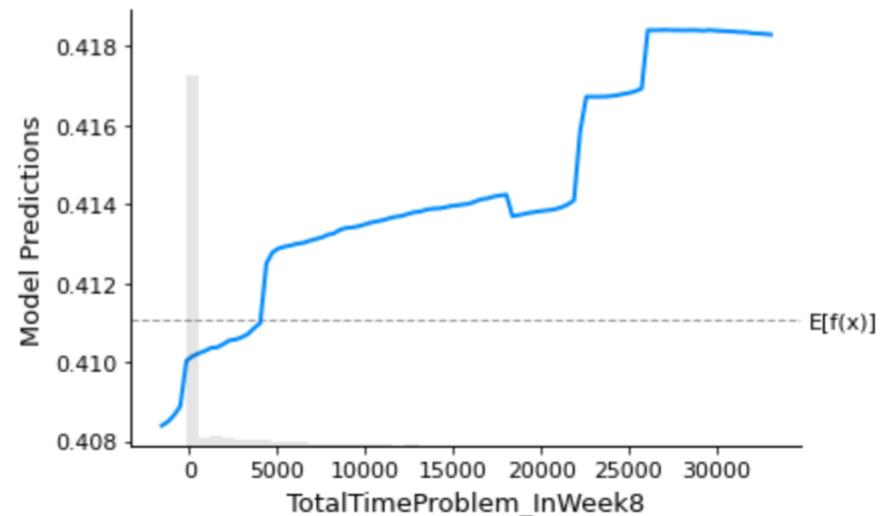
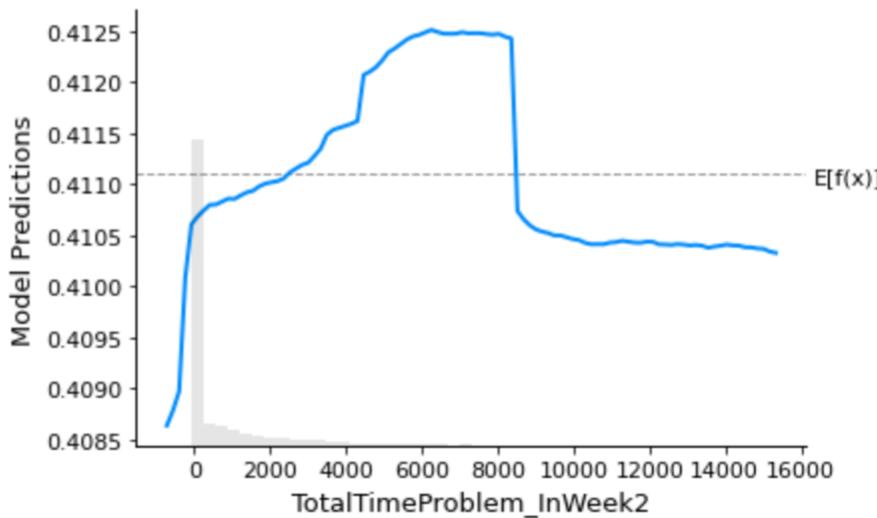
PDP – Strength & Weaknesses

- + Model-agnostic
 - + Computation is intuitive, interpretation is clear
 - + Easy to implement
 - + Causal interpretation
 - Maximum number of features in a PDP is two
 - Assumption of independence
 - Some PDP do not show feature distribution
-

PDP – Your Turn

- Participants: 8679 students of a of an EPFL MOOC with a duration of 10 weeks
 - We have trained a classifier to predict whether a student will pass or fail the course based on their clickstream data
 - Your Task:
 1. Investigate the PDPs for *TotalTimeProblem* in week 2 and week 8
 2. Discuss: how does this feature influence predictions? Is there a difference between week 2 and week 8? What about the distribution of feature values?
-

PDP Example – EPFL MOOC



Local interpretable model-agnostic explanations (LIME)

- Idea: use a local surrogate model (interpretable) to explain individual predictions of a black-box model

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

LIME - Recipe

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

1. Select your instance (sample) of interest



LIME - Recipe

$$\text{explanation}(x) = \operatorname{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

2. Perturb your data set: generate new samples that are variations of the selected sample



LIME - Recipe

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

3. Get the black-box model predictions for the new samples

LIME - Recipe

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

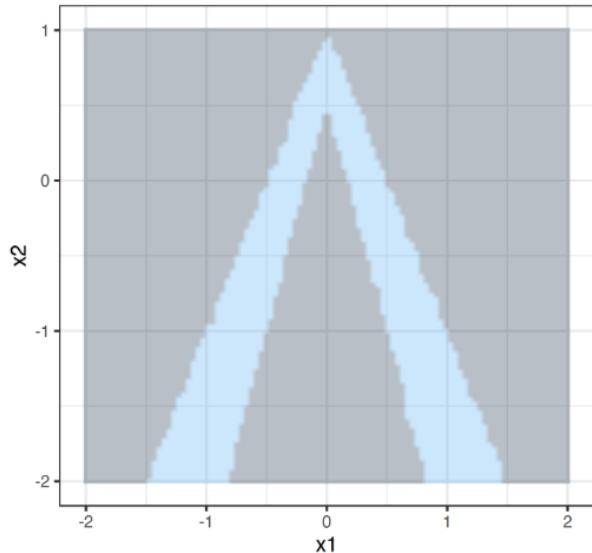
4. Train a weighted, interpretable model on the data set with variations

LIME - Recipe

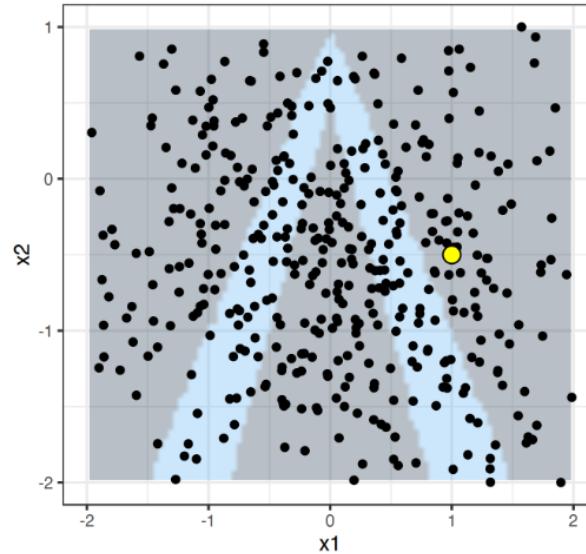
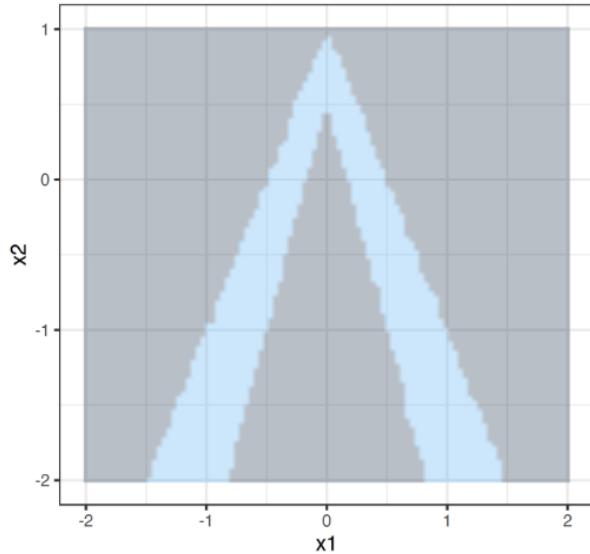
$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

5. Explain the prediction by interpreting the local model

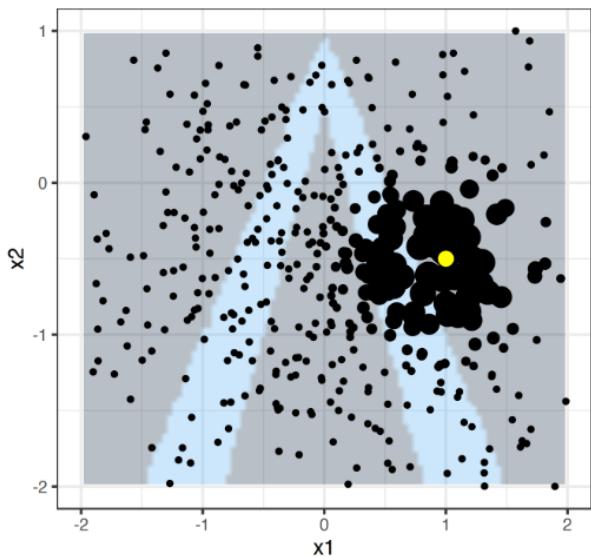
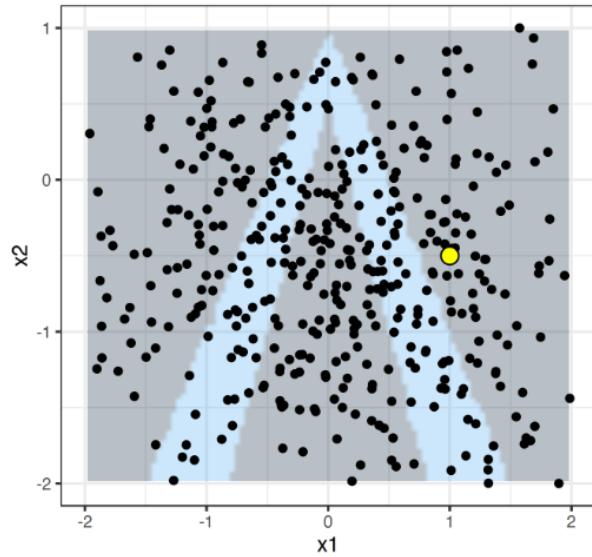
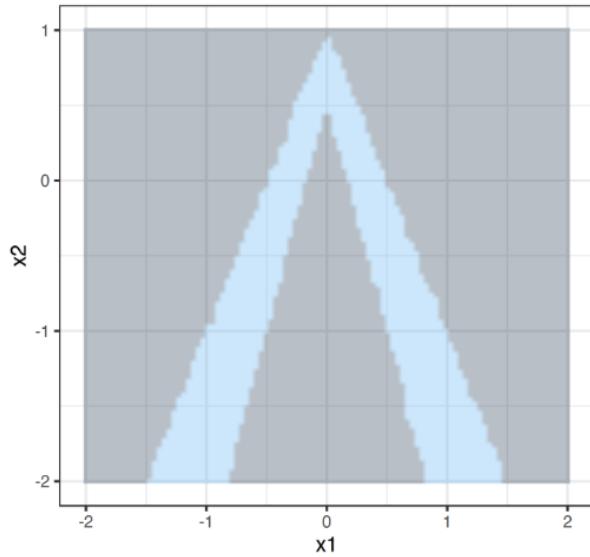
LIME – Perturbation of Sample



LIME – Perturbation of Sample



LIME – Perturbation of Sample



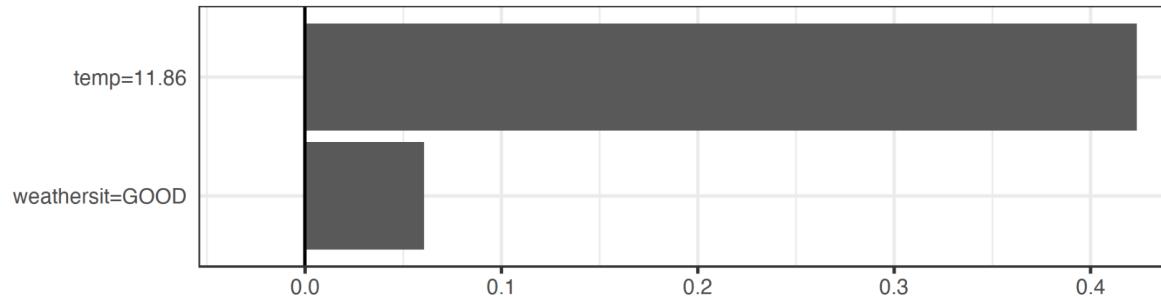
Example – Bike Rental Shop

- Y is binary and indicates, whether the number of bikes rented on a given day will be **above average** ($y = 1$)
- Features (X): season, work day, temperature, humidity, ...
- Given: model f such that $y = f(x)$

Example – Bike Rental Shop

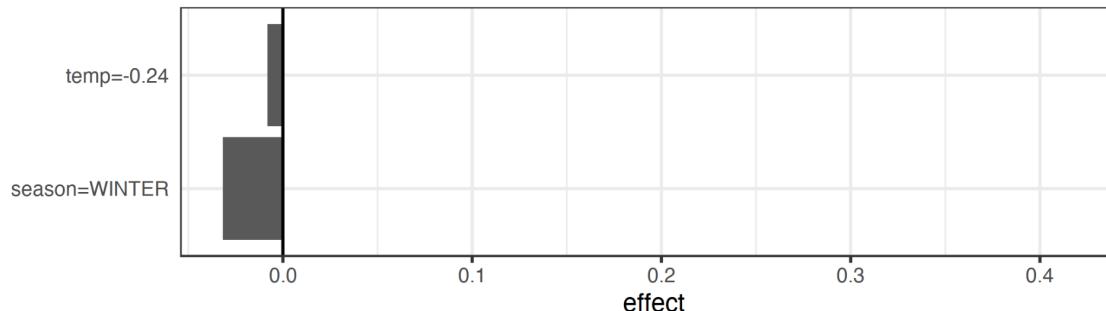
Actual prediction: 0.89

LocalModel prediction: 0.44



Actual prediction: 0.01

LocalModel prediction: -0.03



LIME – Strengths and Weaknesses

- + Model-agnostic (we can replace the underlying model and still use the same surrogate model)
 - + When using for example Lasso regression, explanations are short (= selective)
 - + Benefit from literature on training and interpreting interpretable models
 - + Fidelity measure gives us an idea of reliability
 - Definition of local neighborhood unsolved problem
 - Sampling ignores correlation between features (-> unlikely data points)
 - Instability of explanations
-

LIME – Your Turn

- Your Task:
 1. Run LIME on two instances of your choice
 2. Share the plots for the two instances with us as well as your observations (Are the same features important for both instances? Can you interpret the feature effects?)

Summary

- Interpretability is important (not only for education)
 - We can use intrinsic interpretable models or post-hoc methods to get interpretable predictions
 - Methods can be categorized into global and local
 - PDP is easy to interpret, but has an independence assumption and is limited to a low number of features
 - LIME leads to short explanations, but also ignores correlation between features and might lead to instable explanations
-

Agenda

- 1) Introduction to Explainability**
 - Taxonomy of interpretability method
 - Deep Dive: PDP
 - Deep Dive: LIME
 - 2) Course Wrap-Up (project, exam)**
-

In-Depth Evaluation

- The school of IC performs an in-depth evaluation of each course
 - The in-depth evaluation helps us to get more detailed feedback from you on the course
 - Student evaluations are also a criterion for evaluating the professors' teaching
 - For MLBD, the in-depth evaluation will take place during the poster session on May 22 (on paper)
-

Project – Poster Presentations

- Poster Presentations on May 22 in the BC atrium, starting at 15:00
 - Send us your posters by May 16 at 23:59 ([Google Form](#)) or print them yourselves
 - Each team will get a presentation slot assigned – if you don't sign up for the slot, we will assign you to a slot: [Sign Up Link](#)
 - You will have 5-6 minutes to present and 3-4 minutes for questions
 - There will be prizes by the start-ups as well as the teaching team
-

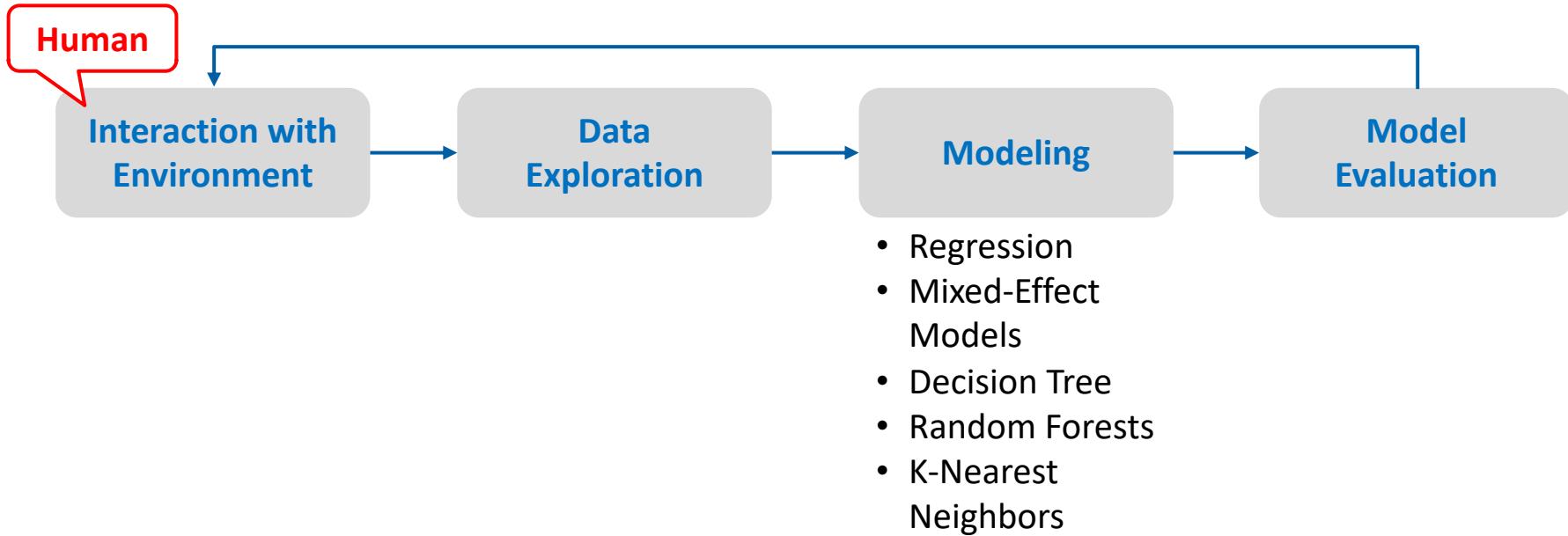
Project – Final Milestone

- Final project (Code + Report) to be delivered by **June 9, 2023 23:59 CET**
- Detailed guidelines (template and structure of report) have been posted on Moodle

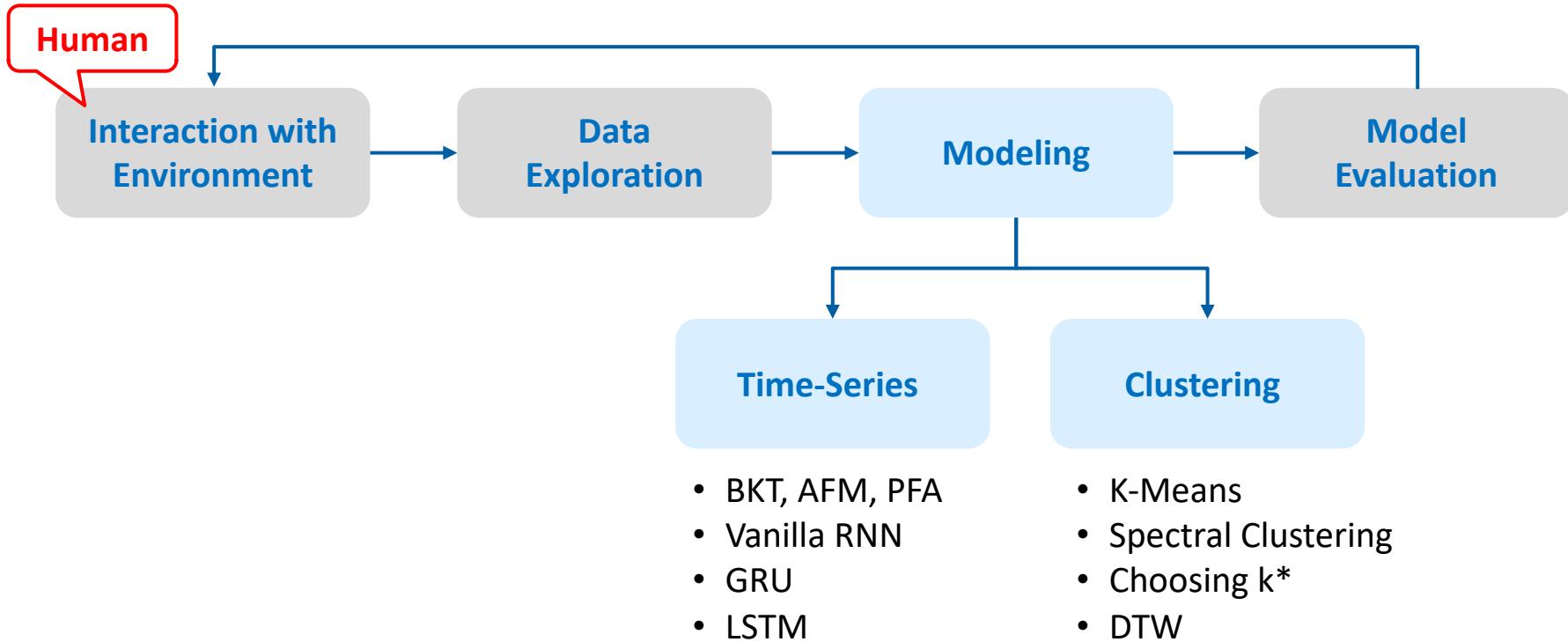
Final Exam - Content

- Mix of conceptual and coding questions
- In the exam: all topics covered in the lecture and tutorials until (including) May 22

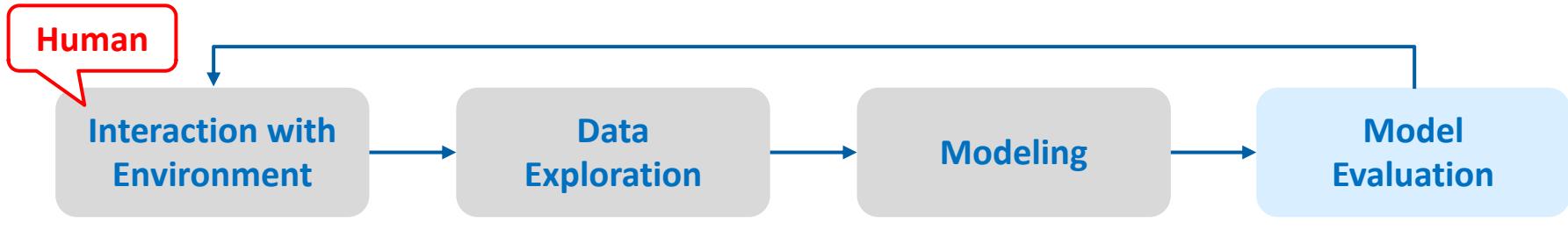
Final Exam - Content



Final Exam - Content



Final Exam - Content



- Fairness
- Explainability

Final Exam - Content

Design/choose an appropriate learning algorithm and features



Select evaluation method



Choose appropriate performance metrics



Select baseline approaches for comparison



Report your results providing error bars

There are many ways to solve a given task (e.g., predicting student performance). It is important that:

- You provide a clean and complete evaluation of your solution
- You are able to justify your decisions for each step

Final Exam - Administrative

- 50% of the final grade
 - Saturday, July 1, 9:15-12:15 (CO020 and CO021)
 - On campus:
 - Conceptual questions: on paper, 1 hour, counts 50% of the exam grade
 - Coding questions: at the computer, 2 hours, counts 50% of the exam grade
 - Environment:
 - Using EPFL NOTO
 - Packages will be pre-installed for you
-

Final Exam - Administrative

- For both the coding and conceptual questions, you are allowed to use the lecture slides, the lecture and lab notebooks, the internet, ...
 - You are not allowed to communicate with other people (and we count posting on forums like Stack Overflow as communicating with other people)
 - You are not allowed to use ChatGPT (or any other language model)
-

MOCK Exam

- We have posted the exam of last year
- **Lab session on May 31 (Wednesday):**
 - A TA will explain and discuss the solutions with you
 - If you plan to attend, try to solve the exam beforehand
- We will post the solutions of last year's exam on Moodle in the last week of the semester

Any Questions?

