Machine Learning

2025 (ML-2025)
Lecture 1. Intro to ML

by Alexei Kornaev, Dr. Sc., Assoc. Prof., Robotics and CV, Innopolis University
Researcher at the RC for AI, National RC for Oncology n.a. NN Blokhin

Agenda

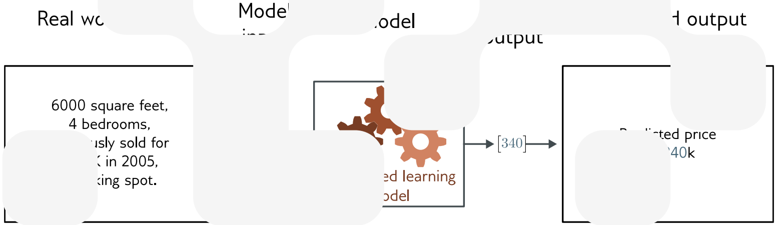
I. Logistics

II. ML Overview: ML = E + T + P

III. Experience (E) in ML

IV. ML Tasks (T)

V. Performance measure (P) in ML



All models are wrong, but some are useful.
/George Box/

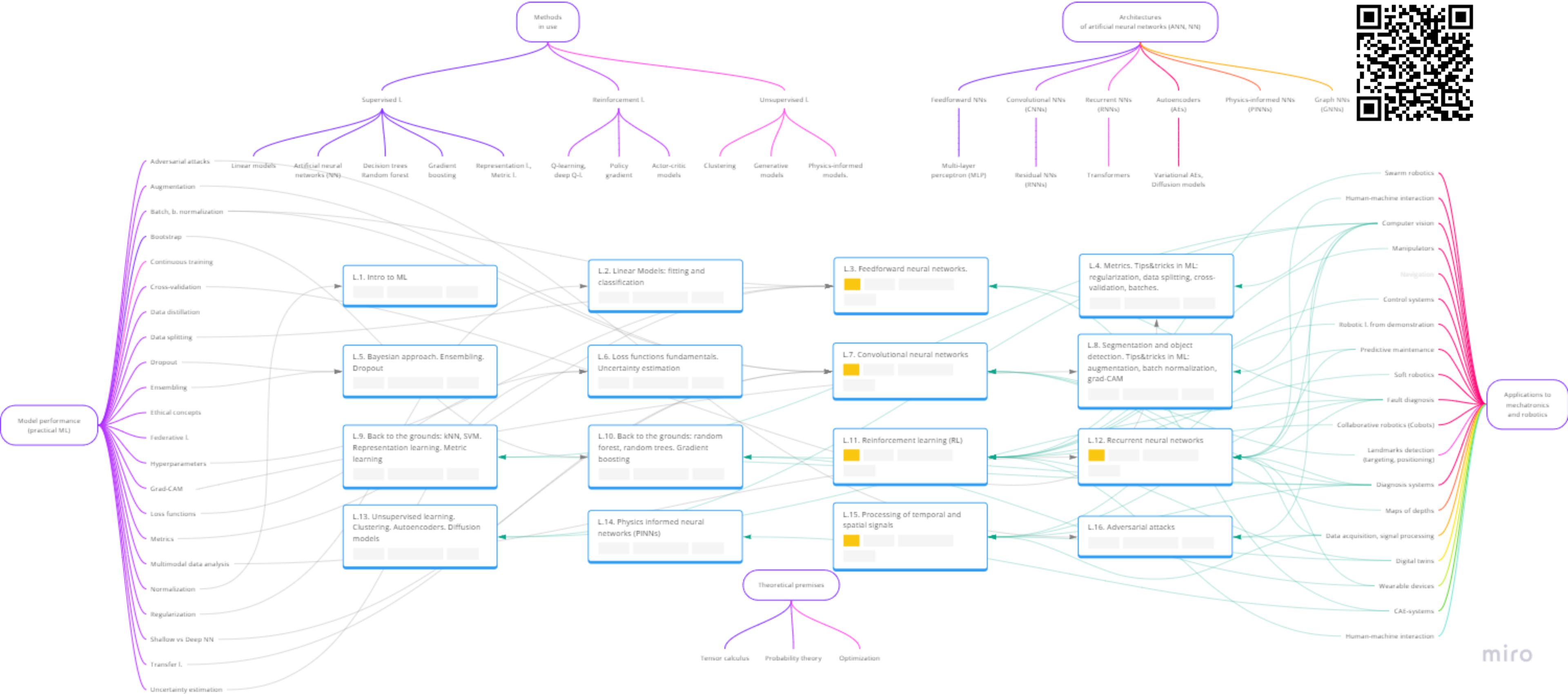
Course objectives

1. Explain how ML works: from basic ideas to real-world problems
2. Teach you how to build a model from scratch or use an open-source model to solve a problem
3. Help you take a few steps forward from educational problems to scientific ones

[1] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision:A survey. Ieee Access, 6:14410–14430, 2018.
[2] M. Artemyev and A. Ashukha. Handbook on Machine Learning (in Russian). Yandex, 2024.URL https://education.yandex.ru/handbook/ml.
[3] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing modeluncertainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR, 2016.
[4] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.http://www.deeplearningbook.org.
[5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. Nature Reviews Physics, 3(6):422–440, 2021.
[6] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for com-puter vision? Advances in neural information processing systems, 30, 2017.
[7] S. J. Prince. Understanding Deep Learning. The MIT Press, 2023. URLhttp://udlbook.com.
[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deeplearning framework for solving forward and inverse problems involving nonlinear partialdifferential equations. Journal of Computational physics, 378:686–707, 2019.
[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: vi-sual explanations from deep networks via gradient-based localization. International journalof computer vision, 128:336–359, 2020

Syllabus

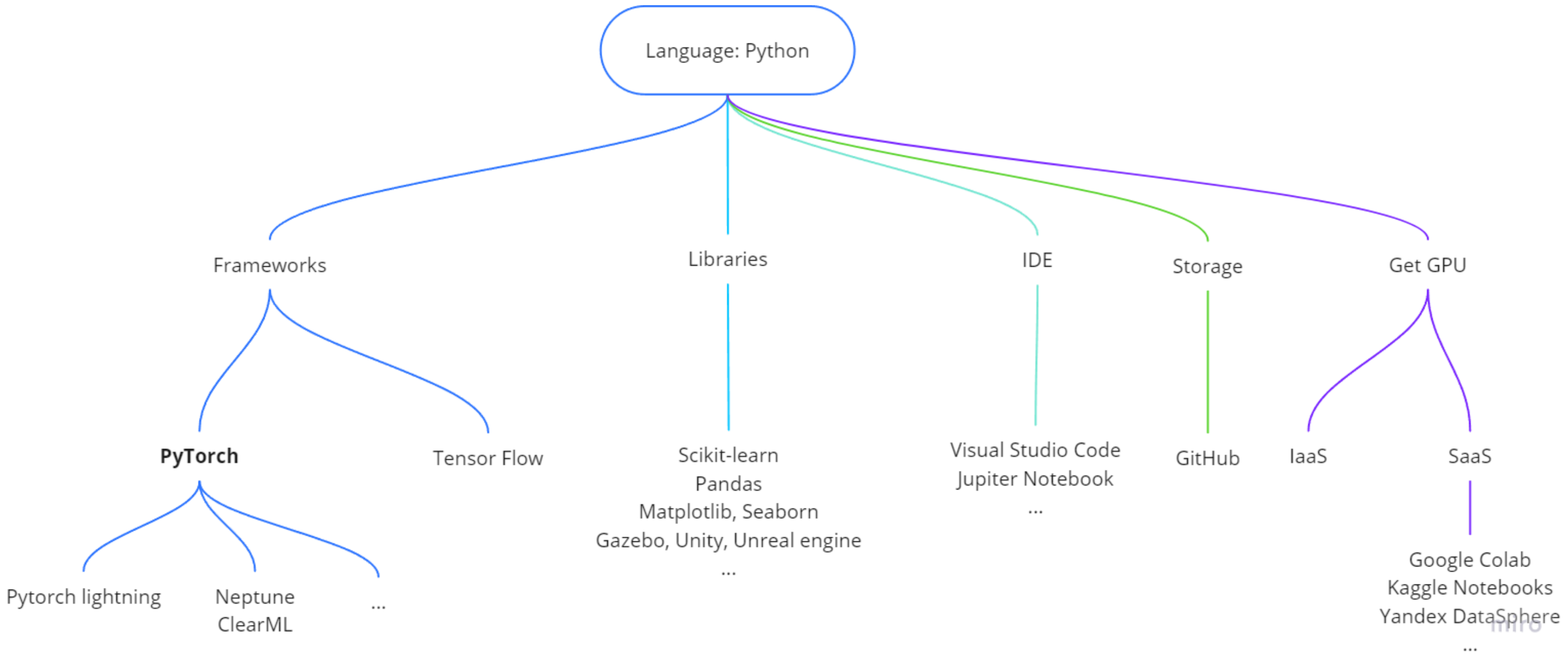
1. Intro to machine learning (ML) [2]
2. Linear models: fitting and classification [2]
3. Feedforward neural networks. Backpropagation [7, 2, 4]
4. Metrics. Tips and tricks in ML: regularization, data splitting, cross-validation, batches.[7, 2, 4]
5. Bayesian approach. Ensembling. Dropout [2, 3]
6. Loss functions fundamentals. Uncertainty estimation [7, 4, 2, 6]
7. Convolutional neural networks (CNNs). Residual neural networks [7, 2, 4]
8. Segmentation and object detection. Tips and tricks in ML: augmentation, batch normal-ization, grad-CAM [7, 2, 9]
9. Back to the grounds: kNN, SVM. Representation learning. Metric learning [2]
10. Back to the grounds: random forest, random trees. Gradient boosting [2]
11. Reinforcement learning (RL) [2, 7]
12. Recurrent neural networks (RNNs). Transformers [2, 7]
13. Unsupervised learning. Clustering. Autoencoders. Diffusion models [7, 2]
14. Physics informed neural networks (PINNs) [5, 8]
15. Processing of temporal and spatial signals [2]
16. Adversarial attacks [1]

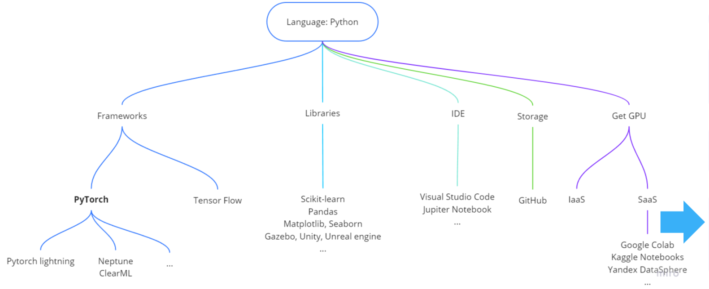


[Public Miro board]

Course objectives

1. Explain how ML works: from basic ideas to real-world problems
2. Teach you how to build a model from scratch or use an open-source model to solve a problem
3. Help you take a few steps forward from educational problems to scientific ones


Language: Python
Frameworks
PyTorch
Pytorch lightning
Neptune
ClearML
...
Tensor Flow
Libraries
Scikit-learn
Pandas
Matplotlib, Seaborn
Gazebo, Unity, Unreal engine
...
IDE
Visual Studio Code
Jupiter Notebook
...
Storage
GitHub
Get GPU
IaaS
SaaS
Google Colab
Kaggle Notebooks
Yandex DataSphere
...

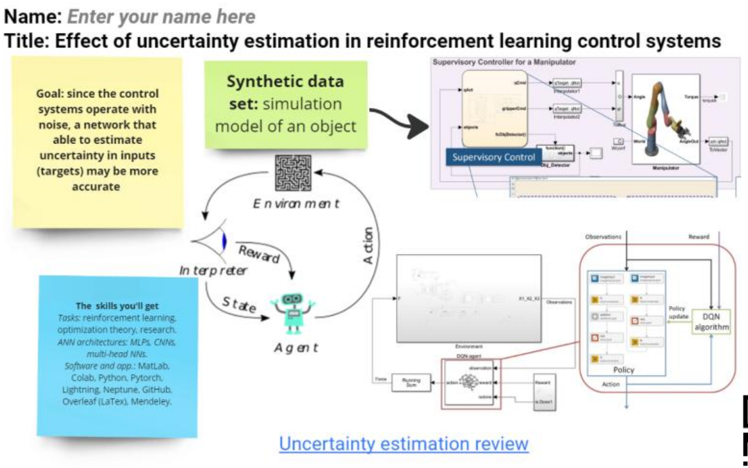


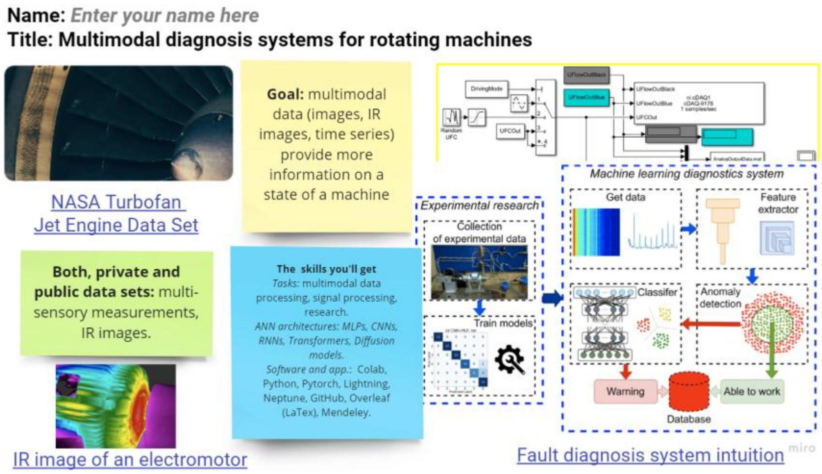
Критерий	Kaggle Notebooks	AWS SageMaker	Azure ML Studio	Google Colab	Yandex DataSphere
Вычислительные ресурсы	Ограничены	Масштабируемы, без серьезных ограничений	Масштабируемы, без серьезных ограничений	Ограничены	Масштабируемы, без ограничений времени использования
Ориентация	Соревнования по машинному обучению	Промышленная разработка, машинное обучение	Промышленная разработка, машинное обучение	Обучение, исследования	Промышленная разработка, машинное обучение
Интеграция с облачными хранилищами	Автоматическое монтирование датасетов	Создание датасетов из хранилищ AWS	Создание датасетов из хранилищ Azure	Упрощенный доступ к данным через Google Drive	Создание датасетов из хранилищ Yandex Cloud Object Storage, хранение датасетов в проекте
Поддержка коллективной работы	Обмен ноутбуками и датасетами с сообществом	Общие Docker-образы, унификация сред	Общие Docker-образы, унификация сред	Поддержка совместной работы через Google Drive	Коллективная работа над проектами и ресурсами
Интерфейс пользователя	Jupyter Notebook	Платформенно-ориентированный UI	Платформенно-ориентированный UI	Jupyter Notebook	Jupyter Notebook
Интеграция с Git	Доступна	Доступна	Доступна	Доступна через интерфейс командной строки	Доступна
Подход к данным и вычислительным ресурсам	Сеансы с ограниченным временем использования	Отделение данных от вычислительных ресурсов	Отделение данных от вычислительных ресурсов	Сеансы с ограниченным временем использования, необходимость сохранения данных в Google Drive	Отделение данных от вычислительных ресурсов, возможность легкого переключения между конфигурациями VM
Особенности для образования и исследований	Соревнования по машинному обучению, обмен решениями	—	—	Широко используется в академических и образовательных целях	Оптимизация пользовательского опыта для студентов, интеграция с учебными курсами
Уникальные функции	Соревнования, обширное сообщество	Интегрированные решения для ML-разработки и развертывания	Визуальное конструирование ML-моделей	Простота использования, бесплатный доступ к ограниченным вычислительным ресурсам	DataSphere Jobs для удобной работы с вычислениями в командной строке, датасеты как виртуальные диски

Handbook on ML by Yandex, 2022.

Course objectives

1. Explain how ML works: from basic ideas to real-world problems
2. Teach you how to build a model from scratch or use an open-source model to solve a problem
3. Help you take a few steps forward from educational problems to scientific ones





Public Miro board

ML → CV LLM RL

Books

Handbook on Machine Learning by M. Artemyev et al., Yandex, 2022 (in Russian)
Understanding Deep Learning by Simon J.D. Prince, 2024
Practical Deep Learning / FastAI book by Jeremy Howard
Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron Courville, 2016.

Online platforms, courses, resources

Sirius online courses on ML (in Russian)
Stepik online courses (in Russian)
Hugging Face online courses
Coursera is unavailable so far

MIT Introduction to Deep Learning, MIT, 2024
Lecture Hall of the Faculty of Applied Mathematics and Informatics (in Russian)
Fast AI, courses, software, book by Jeremy Howard
Deep Learning, course by Semyon Kozlov (in Russian), 2019

3Blue1Brown, Animated Math
PyTorch Tutorial by Patrick Loeber, 2020

#someLinks

Read here: https://arxiv.org/, https://scholar.google.ru/
Collect the references here: https://mendeley.com/
Draw here: https://miro.com/app/dashboard/
Write the text here: https://www.overleaf.com/project
Write the code here: https://colab.research.google.com/
Collect the code here: https://github.com/
Find the journal here: https://journalfinder.elsevier.com/
Find the conference here: https://portal.core.edu.au/conf-ranks/?search=A

Agenda

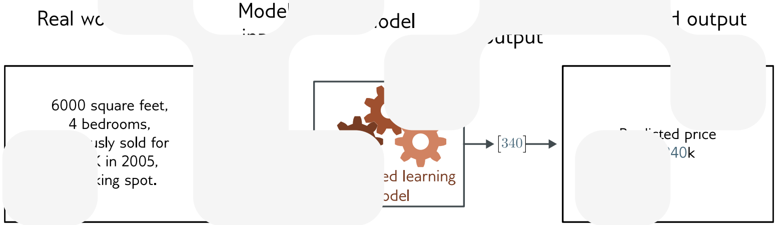
I. Logistics

II. ML Overview: ML = E + T + P

III. Experience (E) in ML

IV. ML Tasks (T)

V. Performance measure (P) in ML



All models are wrong, but some are useful.
/George Box/

$x \rightarrow \boxed{} \rightarrow y$

Deterministic vs Stochastic approaches to modeling

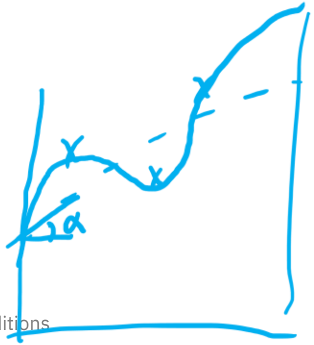
$$\frac{d^2x}{dt^2} = -g;$$

$$V(0) = V_0, \quad x(0) = x_0 .$$

$$a = -g,$$

$$V = -gt + V_0,$$

$$x = -\frac{gt^2}{2} + V_0 t + x_0.$$

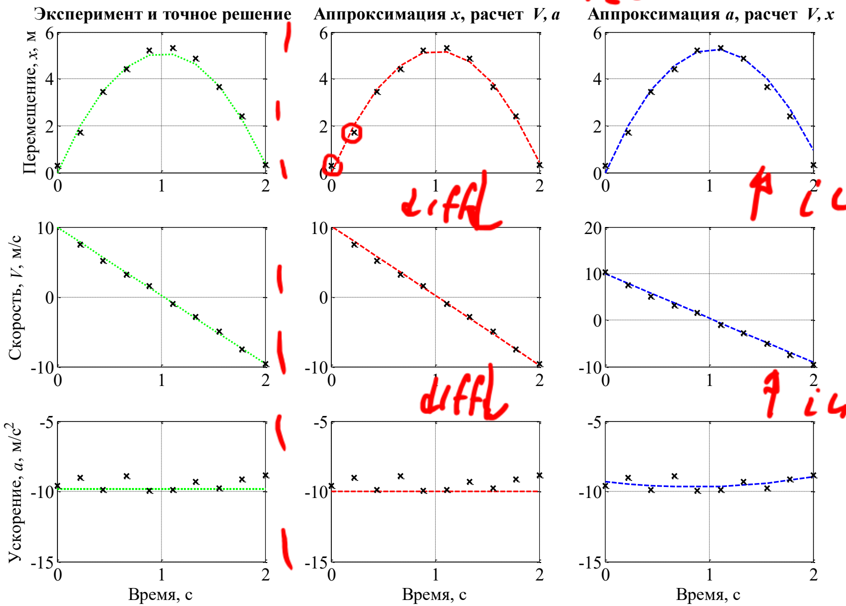


Deterministic approach: solution of diff. eq. + initial conditions

2

3

$x = \varphi_0 + \varphi_1 t + \varphi_2 t^2$



$a = \varphi_0 + \varphi_1 t + \varphi_2 t^2$

Stochastic approach: experiment + approximation

Terms

AGI

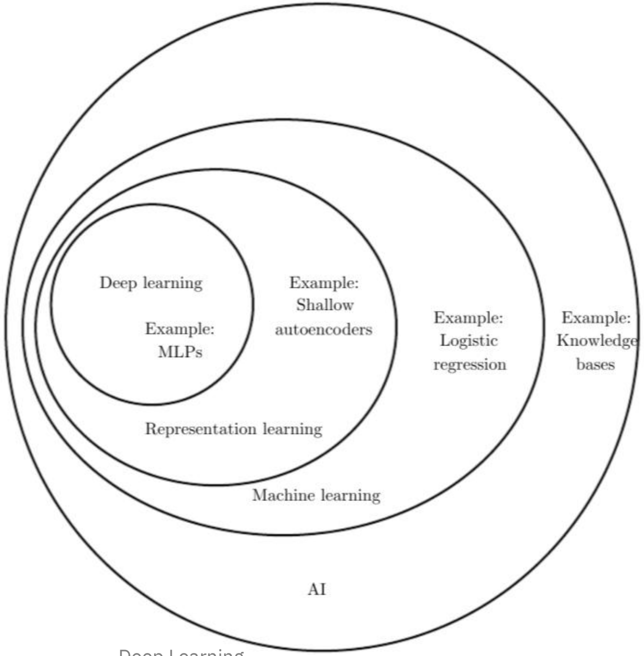
Искусственный интеллект / Artificial Intelligence (AI):

область информатики, занимающаяся моделированием разумного поведения в компьютерах / a branch of computer science dealing with the simulation of intelligent behavior in computers (Merriam-Webster)

Машинное обучение / Machine Learning (ML):

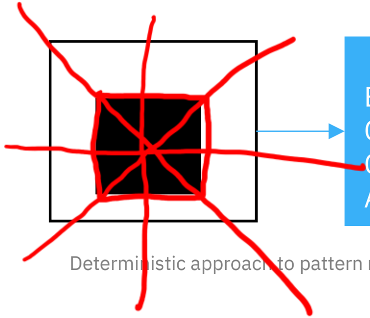
область знаний, в которой компьютеры обучаются без явного программирования/ field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959);

задача «З», в ходе решения которой программа обучается из опыта «О» и повышает меру качества «К» / well-posed learning problem: a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1998)



Deep Learning

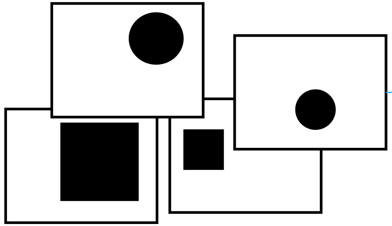
Intuition



Algorithm
Edge Detection;
Contour Construction;
Center Finding;
Axis of Symmetry Finding;

Logical Inference: This is a **square**

Deterministic approach to pattern recognition

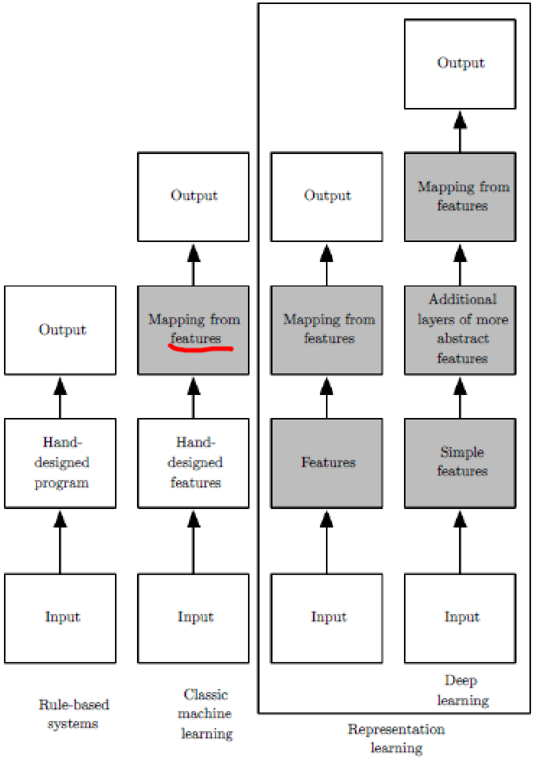


Algorithm
learn from experience E (from dataset of images), improve the performance measure P

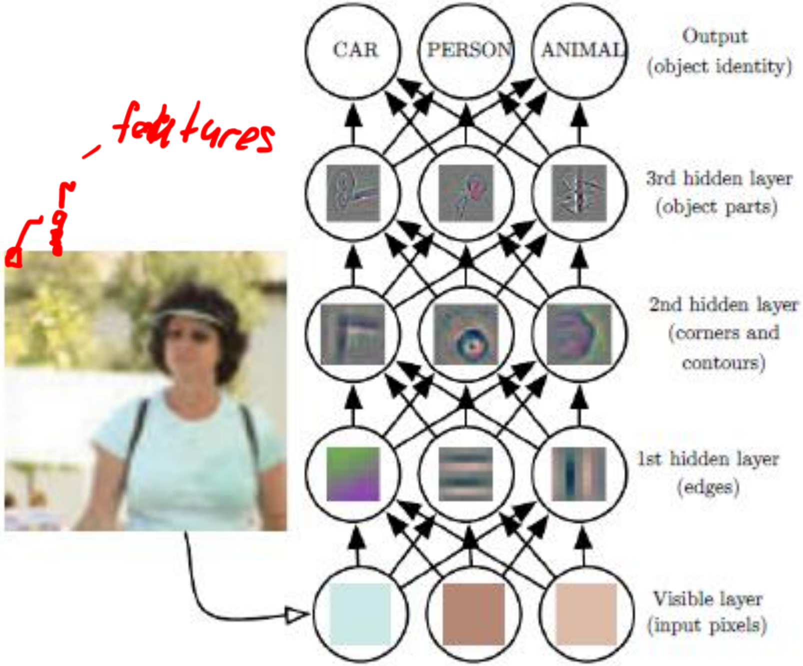
Trained Inference: This is a **square** (not a circle)

Stochastic ML approach to pattern recognition

Intuition



[Deep Learning]

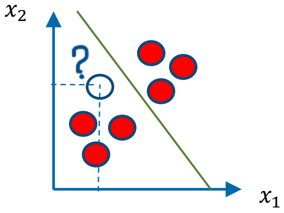


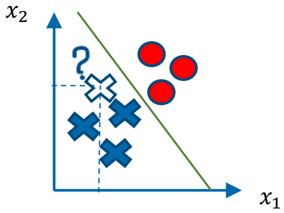
[Deep Learning]

Approaches to ML

ML

Unsupervised learning	...Reinforcement learning...	Supervised learning



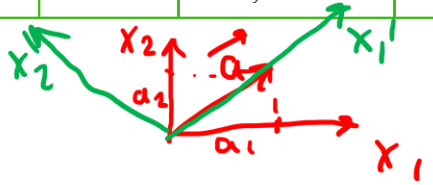


Prerequisites

ML prerequisites

Линейная алгебра / Linear algebra + Мат. анализ / Calculus + Теор. вер. / Probability theory

Titile	Geom. Analog	Notation in tensor / scalar forms	# of comp., in $\mathbb{R}^N$
Scalar		a	N^0
Vector		$\vec{a}, \boldsymbol{a}, [a_i]$, a_i	N^1
Tensor (rank 2)		$\boldsymbol{T}_a, [a_{ij}]$, a_{ij}	N^2



A *tensor* of rank n is a mathematical quantity characterized in N-dimensional space ($\mathbb{R}^N$) by N^n components, each of which transforms according to a specific rule when the coordinate system is rotated*:

$a' = a$ is for *scalar* (0-rank tensor);

$\boxed{a'_i = \alpha_{ij} a_j}$ is for *vector* (1-rank tensor),

$(a'_1 = \alpha_{11} a_1 + \alpha_{12} a_2 + \alpha_{13} a_3$,
$a'_2 = \alpha_{21} a_1 + \alpha_{22} a_2 + \alpha_{23} a_3$,
$a'_3 = \alpha_{31} a_1 + \alpha_{32} a_2 + \alpha_{33} a_3$);

$a'_{ik} = \alpha_{ij} \alpha_{km} a_{jm}$ is for *tensor* (2-rank tensor);

$a'_{i \ldots k} = \alpha_{ij} \ldots \alpha_{km} \, a_{j \ldots m}$ is for *tensor* in general (*n*-rank tensor);

____

* - The Einstein summation notation is used

Prerequisites

ML prerequisites

Линейная алгебра / Linear algebra + Мат. анализ / Calculus + Теор. вер. / Probability theory

Titile	Geom. Analog	Notation in tensor / scalar forms	# of comp., in $\mathbb{R}^N$
Scalar		a	N^0
Vector		$\vec{a}, \boldsymbol{a}, [a_i]$, a_i	N^1
Tensor (rank 2)		$\boldsymbol{T}_a, [a_{ij}]$, a_{ij}	N^2

$$\nabla a = \left[\left[\frac{\partial a}{\partial x_i}\right]\right].$$

The **ńØĐńį ẾģẾẬÃ** of some function $f(x)$ with respect to a probability distribution $p(x)$:

$$E(f(x)) = \sum_x p(x) f(x).$$

The conditional maximum likelihood estimator:

$$\Theta_{ML} = \operatorname{argmax} \sum_{i=1}^{m} \log\left(p\left(y^{(i)} | x^{(i)}; W\right)\right).$$

Bellman Expectation Equation for State-Action Value Function (Q-Function):

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a].$$

Google Parkour

Agenda

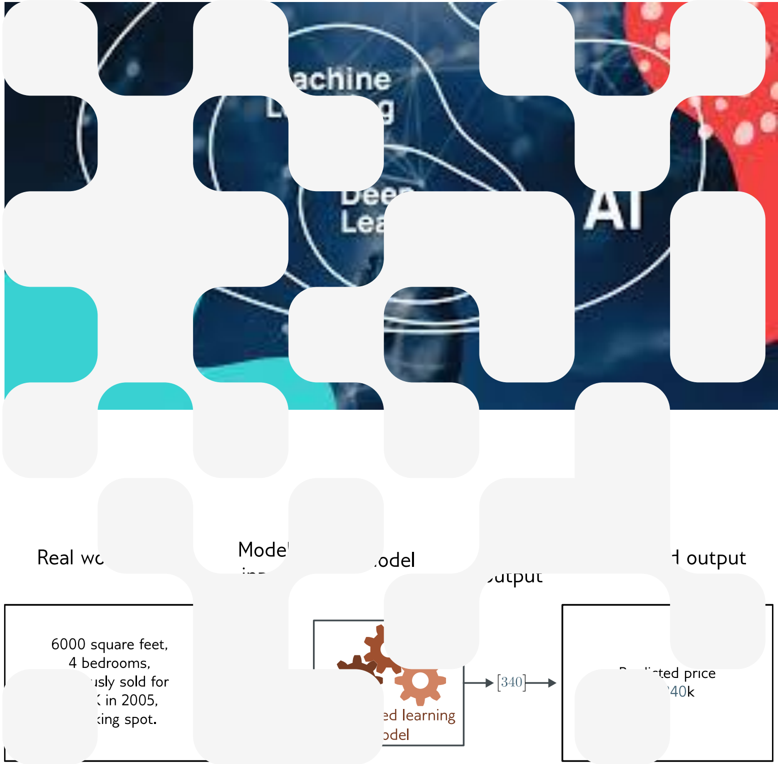
I. Logistics

II. ML Overview: ML = E + T + P

III. Experience (E) in ML

IV. ML Tasks (T)

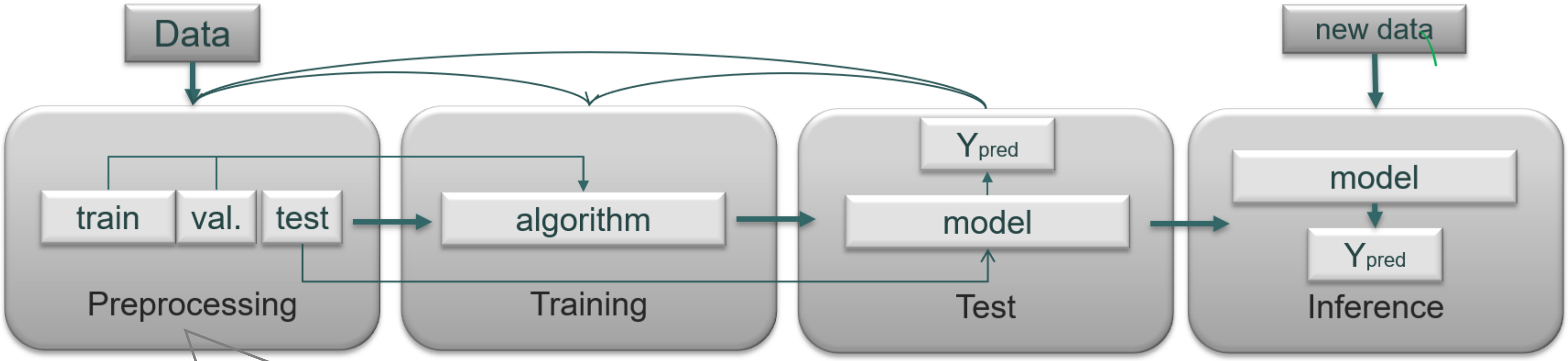
V. Performance measure (P) in ML



All models are wrong, but some are useful.
/George Box/

$\vec{x} \rightarrow \boxed{} \rightarrow y$

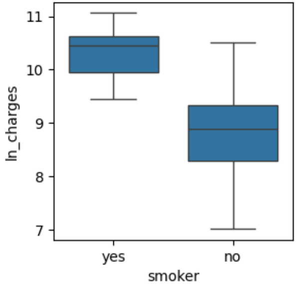
Flowchart for an ML model design (tabular data)

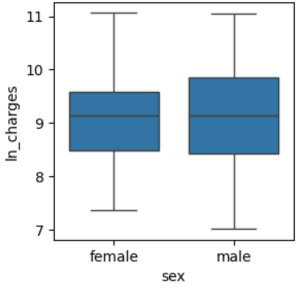


handling missing values
EDA: descriptive statistics, A/B tests, correlation
encoding categorical features
feature normalization
dimensionality reduction of the feature space/constructing new features
splitting the dataset

Exploratory data analysis (EDA)

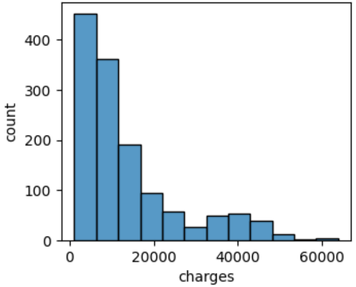
	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520



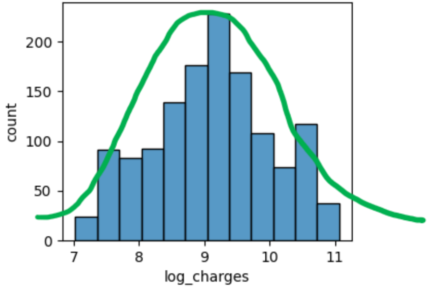


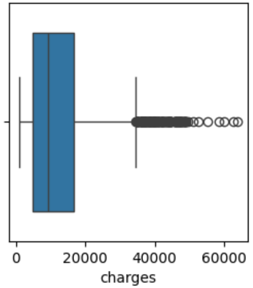
Medical insurance payout

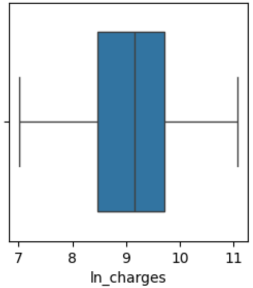
y

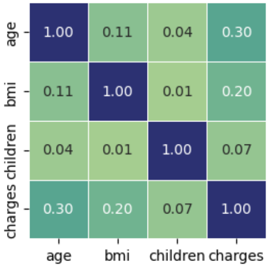


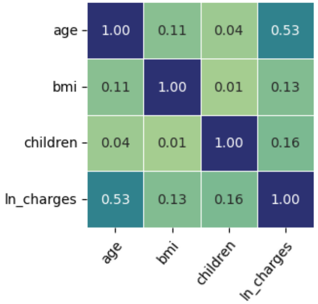
$\tilde{y} = \ln(1 + y)$











Encode categorical features

```
print(data.region.unique())
['southwest', 'southeast', 'northwest', 'northeast']
```

```
pd.get_dummies(data[['region']], dtype=int)
```

region
southwest
southeast
southeast
northwest
northwest

region_northeast	region_northwest	region_southeast	region_southwest
0	0	0	1
0	0	1	0
0	0	1	0
0	1	0	0
0	1	0	0

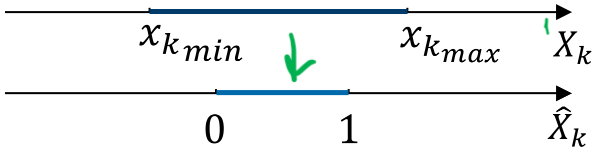
⚠ warning: $x_4 = 1 - \sum_{i=1}^{3} x_i$

```
pd.get_dummies(data[['region']], dtype=int,
drop_first=True)
```

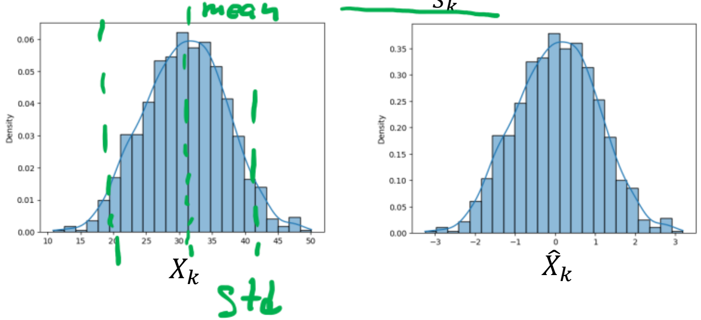
region_northwest	region_southeast	region_southwest
0	0	1
0	1	0
0	1	0
1	0	0
1	0	0

Normalize features

1. MinMaxScaler: $\hat{X}_k = \dfrac{x_k - x_{k_{min}}}{x_{k_{max}} - x_{k_{min}}}$



2. StandardScaler: $\hat{X}_k = \dfrac{X_k - \bar{x}_k}{S_k}$



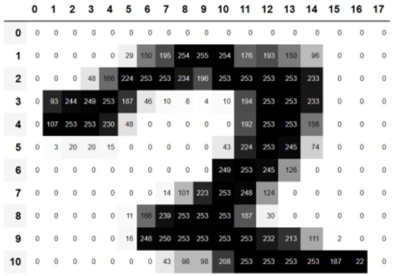

```python
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_num_norm = scaler.fit_transform(data[['age', 'bmi','children']])
```

	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest	age_norm	bmi_norm	children_norm	ln_charges
0	0.0	1.0	0.0	0.0	1.0	0.021739	0.321227	0.0	9.734236
1	1.0	0.0	0.0	1.0	0.0	0.000000	0.479150	0.2	7.453882
2	1.0	0.0	0.0	1.0	0.0	0.217391	0.458434	0.6	8.400763
3	1.0	0.0	1.0	0.0	0.0	0.326087	0.181464	0.0	9.998137
4	1.0	0.0	1.0	0.0	0.0	0.304348	0.347592	0.0	8.260455

Data: input(s) is (are) stationary

Numeric	Visual	Textual
Feature(s)	Image	Word

Feature	Value	Descript.
x_0 …	2	# rooms
x_m	5.5	Distance



“cat”

$$X = [x_i] = \begin{bmatrix} x_0 \\ \dots \\ x_m \end{bmatrix}$$

$$X = [x_{ij}] = \begin{bmatrix} x_{01} & \dots & x_{0n} \\ \dots & \dots & \dots \\ x_{0m} & \dots & x_{mn} \end{bmatrix}$$

$$X = [x_i] = \begin{bmatrix} x_0 \\ \dots \\ x_m \end{bmatrix}$$

Architecture:

Perceptron, decision tree, 1D-CNN, Transformer

Convolutional Neural Network (CNN), Transformer

Transformer

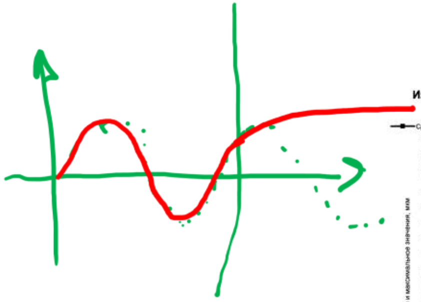
Data: input(s) is (are) non-stationary

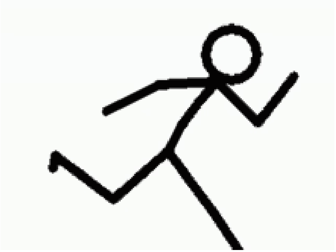
$\vec{a}\cdot\vec{b} = \underset{1}{\|\vec{a}\|}\,\underset{1}{\|\vec{b}\|}\,\underbrace{\cos(\widehat{\vec{a}\,\vec{b}})}$

↓

1 sim

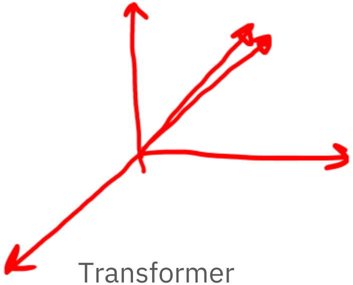
Numeric	Visual	Textual
Sequence of features	Sequence of images	Sequence of words





"The | cat's | eyes | sparkled | with | ..." → tokens

→ embeddings $\mathbb{R}^N$



Architecture:	Recurrent network (RN), Transformer	CNN + LSTM, Transformer	Transformer

Data: output(s) is (are) stationary

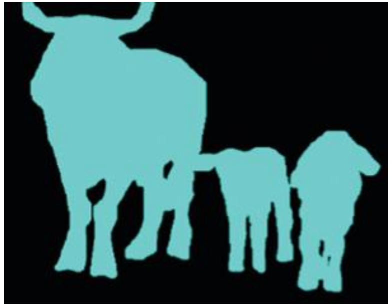
$x \to \boxed{} \to y$

Numeric	Visual	Textual
Number(s)	Image	Word

Numeric

Feature	…	Price, MRub
x_0	…	10
…		
x_m	…	

Visual



Textual

“curiosity”

$$y$$

$$Y = [y_{ij}] = \begin{bmatrix} y_{01} & \dots & y_{0n} \\ \dots & \dots & \dots \\ y_{0m} & \dots & y_{mn} \end{bmatrix}$$

$$Y = [y_i] = \begin{bmatrix} y_0 \\ \dots \\ y_m \end{bmatrix}$$

Architecture:

Numeric	Visual	Textual
Perceptron, decision tree, 1D-CNN, Transformer	Convolutional Neural Network (CNN)	Transformer

Agenda

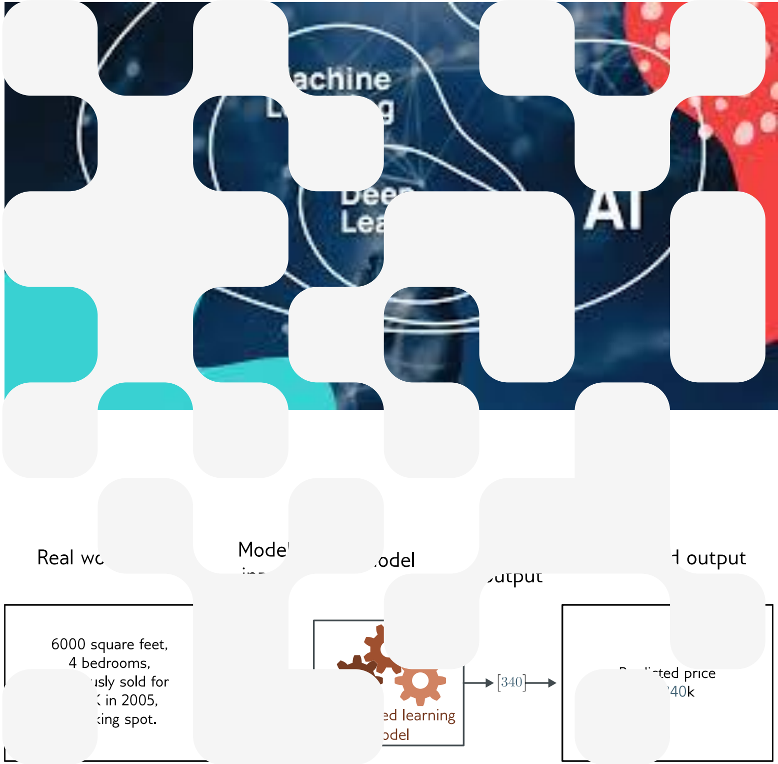
I. Logistics

II. ML Overview: ML = E + T + P

III. Experience (E) in ML

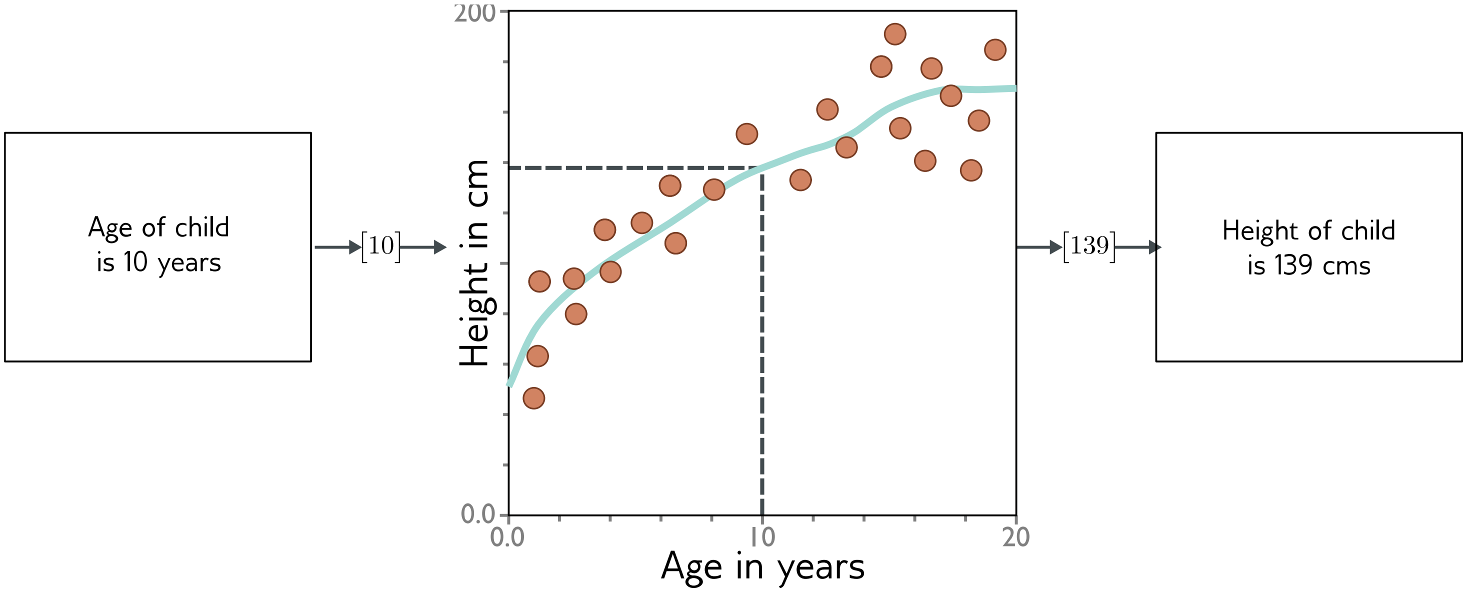
IV. ML Tasks (T)

V. Performance measure (P) in ML



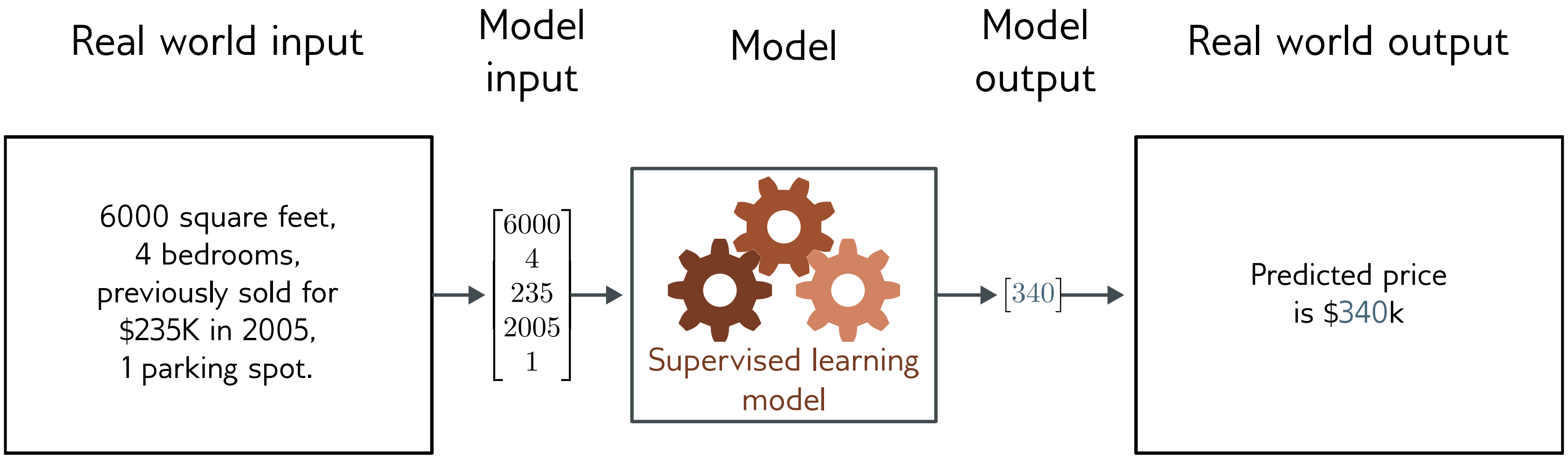
All models are wrong, but some are useful.
/George Box/

An ML model intuition



Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

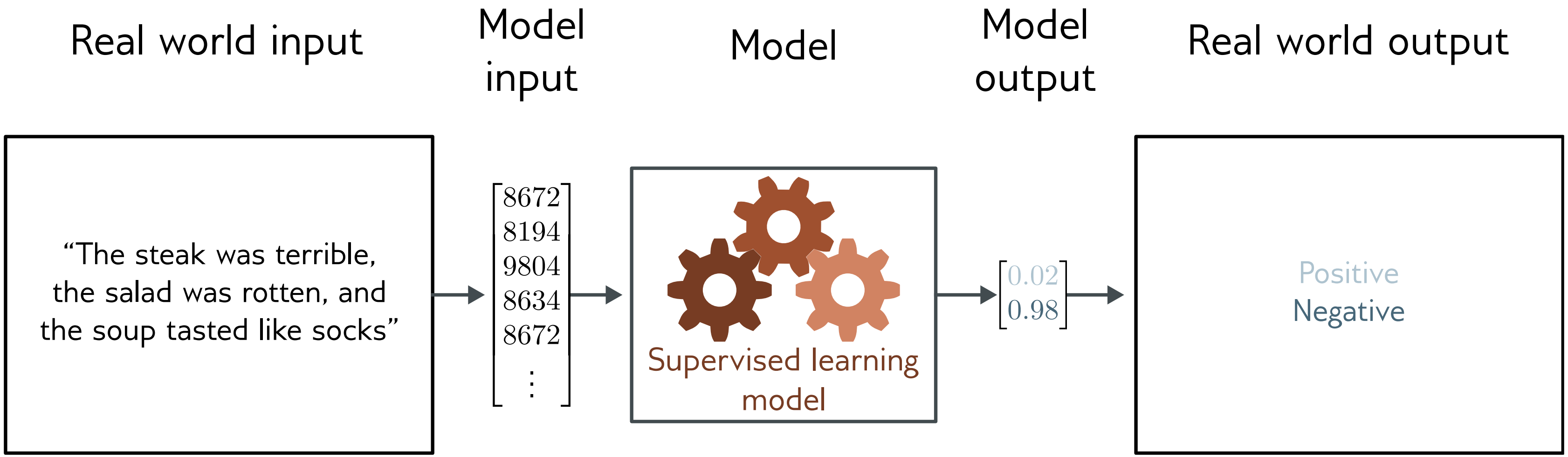
Regression



Univariate regression problem (one output, real value)
Fully connected network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

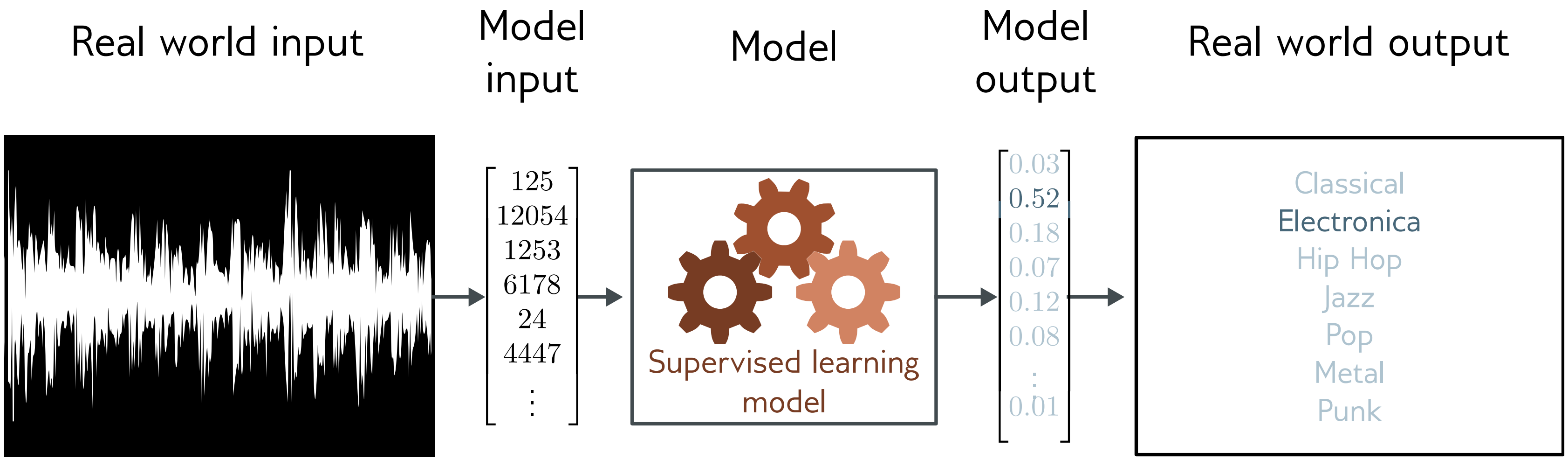
Text classification



Binary classification problem (two discrete classes)
Transformer network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

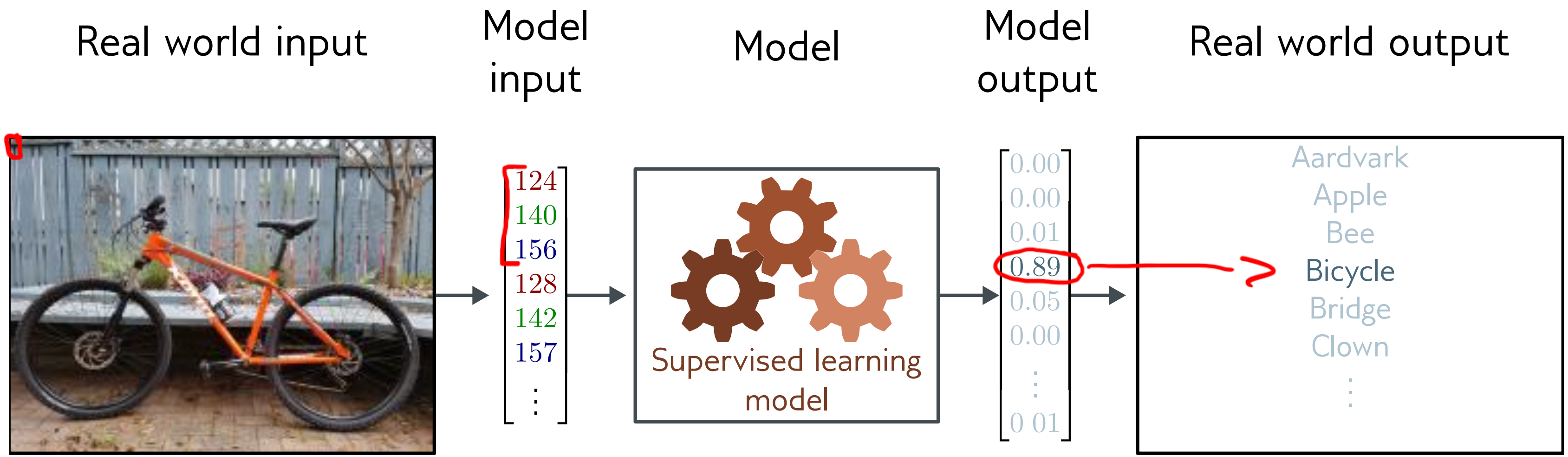
Music genre classification



Multiclass classification problem (discrete classes, >2 possible values)
Recurrent neural network (RNN), Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

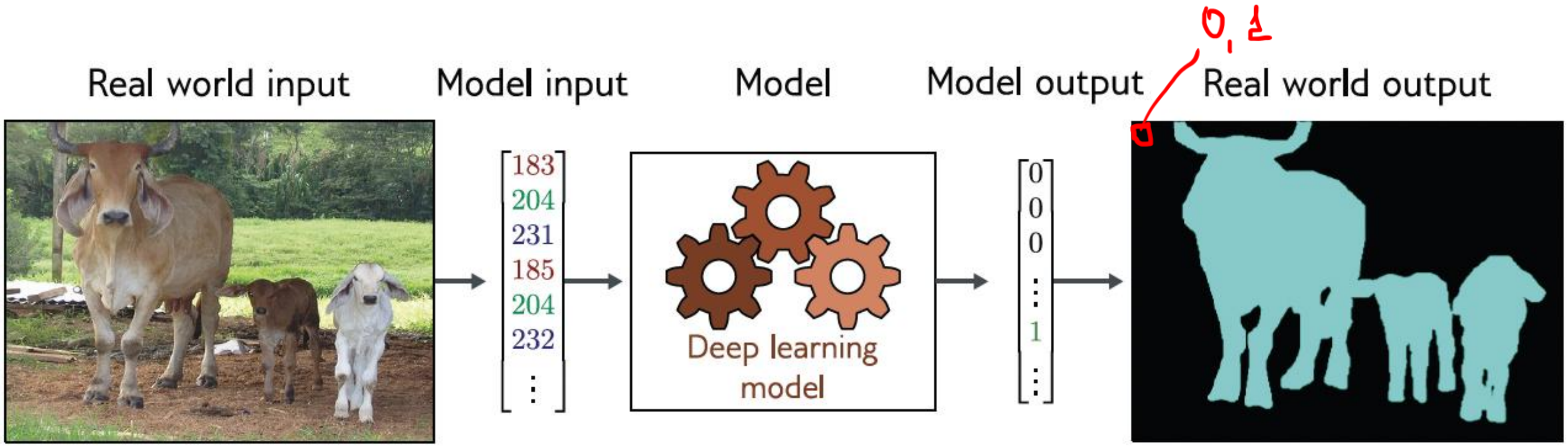
Image classification



Multiclass classification problem (discrete classes, >2 possible classes)
Convolutional networks, Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

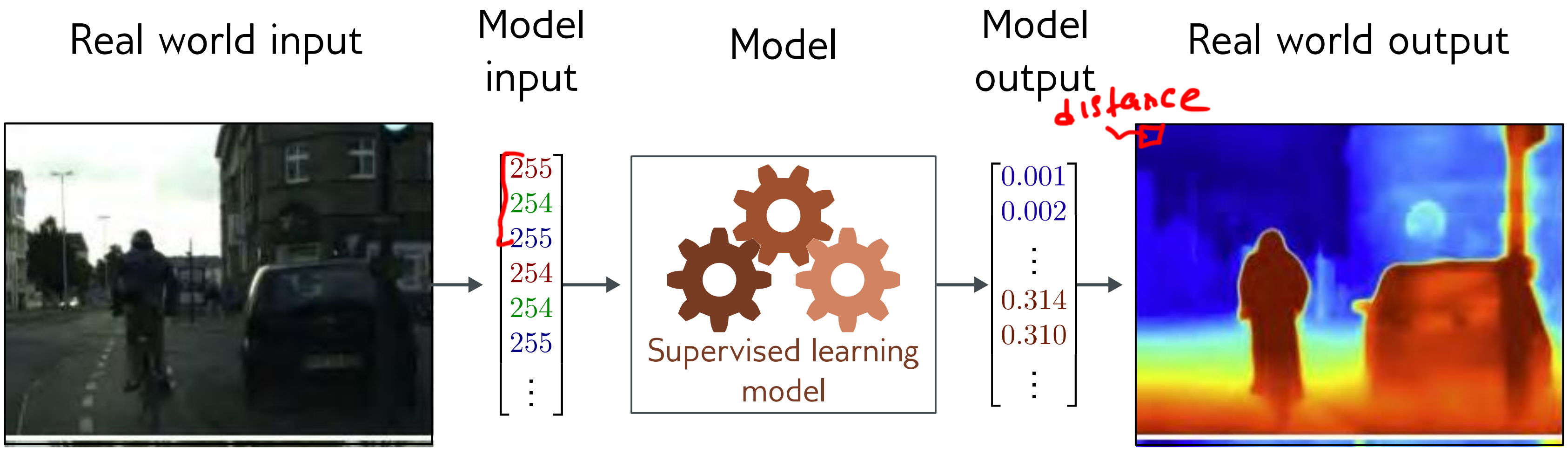
Image segmentation



Multivariate binary classification problem (many outputs, two discrete classes)
Convolutional encoder-decoder network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

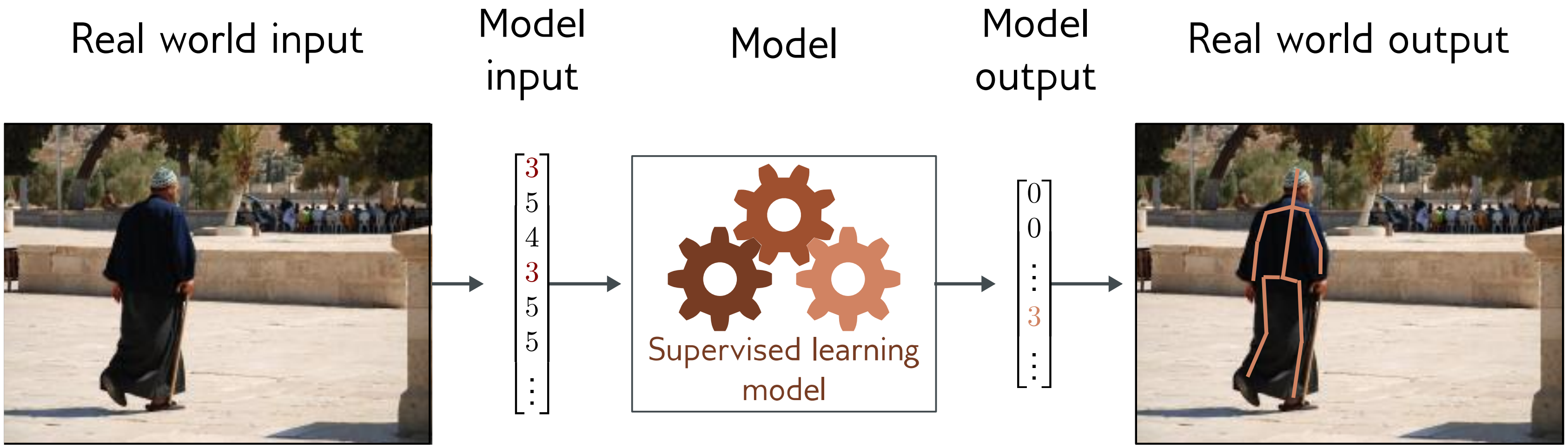
Depth estimation



Multivariate regression problem (many outputs, continuous)
Convolutional encoder-decoder network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

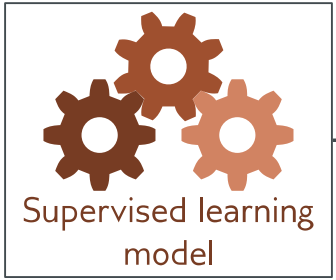
Pose estimation



Multivariate regression problem (many outputs, continuous)
Convolutional encoder-decoder network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

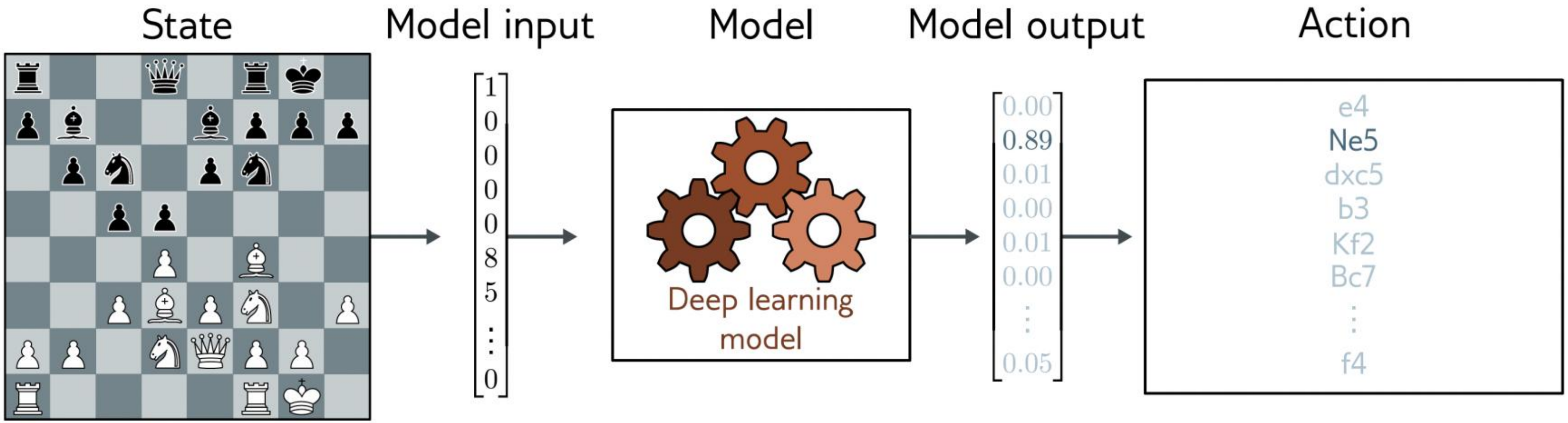
Image captioning

Real world input	Model input	Model	Model output	Real world output
	$\begin{bmatrix} 183 \\ 204 \\ 231 \\ 185 \\ 204 \\ 232 \\ \vdots \end{bmatrix}$		$\begin{bmatrix} 1 \\ 5593 \\ 7532 \\ 7924 \\ 1 \\ \vdots \end{bmatrix}$	"A Kazakh man on a horse holding a bird of prey"

Visual data analysis and text generation problem
Transformer networks

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

Games



Decision making problem
Feedforward networks, CNNs, transformer networks

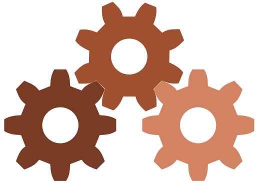
Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

Image generation from text

Real world input | Model input | Model | Model output | Real world output

"Teddy bears mixing sparkling chemicals as mad scientists, in a steampunk style."

$$\begin{bmatrix} 8300 \\ 532 \\ 7676 \\ 7898 \\ 883 \\ \vdots \end{bmatrix}$$



Supervised learning model

$$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ \vdots \end{bmatrix}$$

Etc., etc.

Supervised learning intuition

S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

Agenda

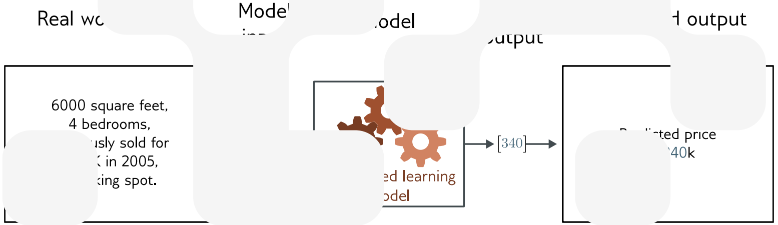
I. Logistics

II. ML Overview: ML = E + T + P

III. Experience (E) in ML

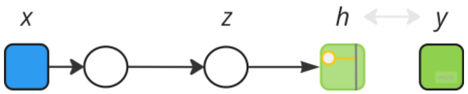
IV. ML Tasks (T)

V. Performance measure (P) in ML



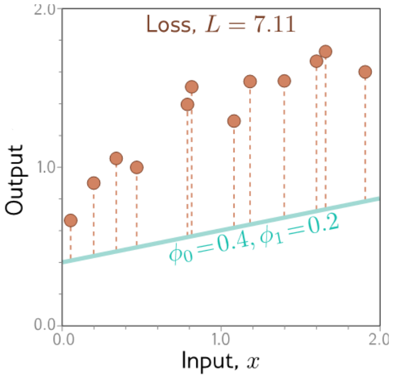
All models are wrong, but some are useful.
/George Box/

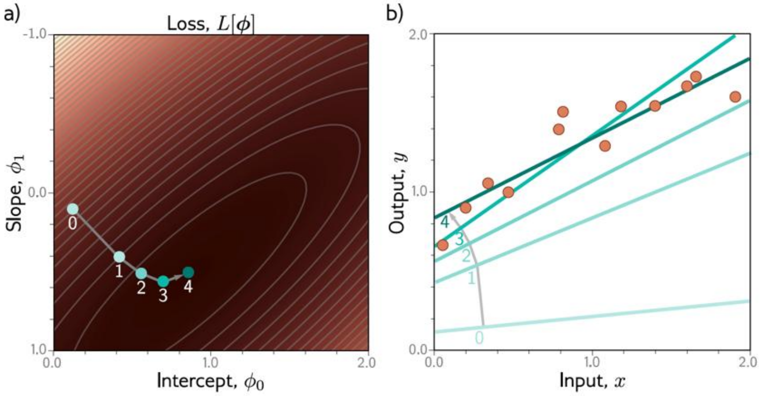
Performance Measure intuition



$$L_R = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h^{(i)}\right)^2$$

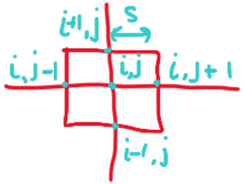
Model predicts output *h* given input *x*

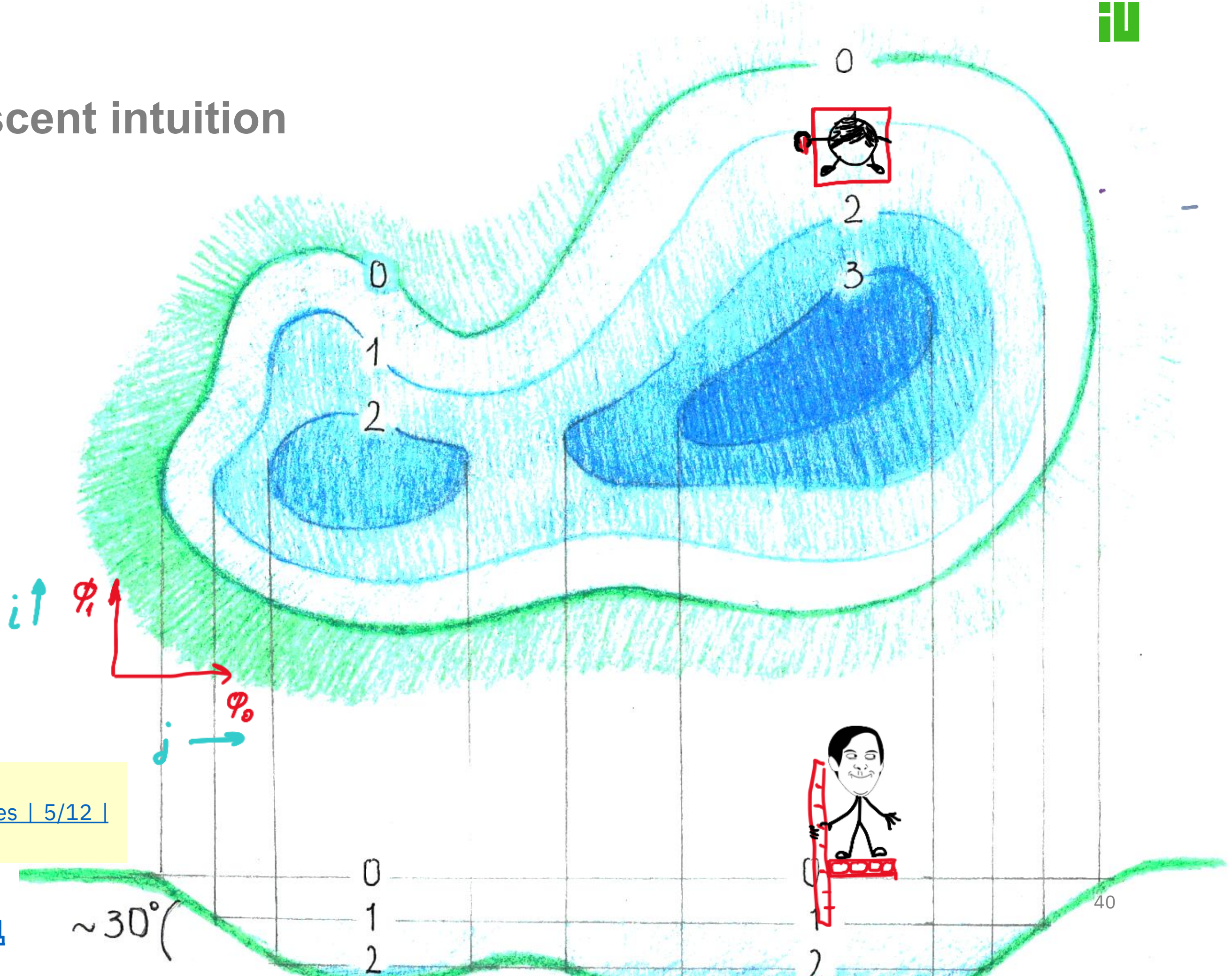




Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Gradient ascent/descent intuition

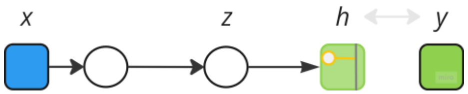




Want to know more?
[DeepMind x UCL | Deep Learning Lectures | 5/12 | Optimization for Machine Learning]

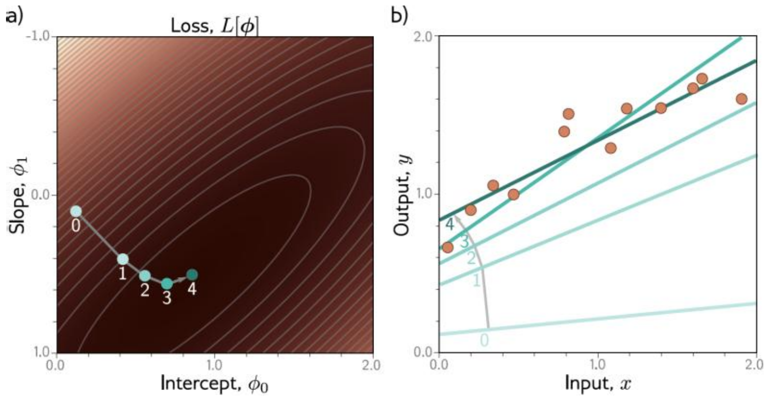
[Пруд]

Performance Measure intuition



$$L_R = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - h^{(i)} \right)^2$$

Model predicts output *h* given input *x*



Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Think about it

1. Why it is not recommended to differentiate functions obtained with approximation?
2. How can we estimate that *data* is enougth to solve a problem with an ML model?
3. Why small noise in *data* may be usefull?
4. Is it possible to make an AI model that solves all problems at once?
5. Is it possible to solve a problem using a complex *performance measure*? E.g. minimize something and maximize something else simultaneously?

Thank you for your attention!

a.kornaev@innopolis.ru, @avkornaev