

# Machine Learning

## Introduction. ML History

Aleksandr Petiushko

ML Research



# Content

## ① Introduction

# Content

- ➊ Introduction
- ➋ Course logistics and syllabus

# Content

- ① Introduction
- ② Course logistics and syllabus
- ③ Historic reference



# Intro

Time to introduce yourselves: what are your hobbies, motivation in ML, etc.: please go into “**Module 1 Students Introduction**” thread

# Sofia Plagiarism Policy

- It covers parts “*sourced from AI*”
  - ▶ Please read the “**Sofia Plagiarism Policy**” thread
  - ▶ **First offense:** students need to rewrite assignment
  - ▶ **Second offense:** students fail the course
  - ▶ **Third offense:** students are to be withdrawn from their program

# Note about xGPTx

- It can produce very plausible answers in 90% of cases



# Note about xGPTx

- It can produce very plausible answers in 90% of cases
- The caveats are the following:

# Note about xGPTx

- It can produce very plausible answers in 90% of cases
- The caveats are the following:
  - ▶ It can really hallucinate some things which are just untrue

# Note about xGPTx

- It can produce very plausible answers in 90% of cases
- The caveats are the following:
  - ▶ It can really hallucinate some things which are just untrue
  - ▶ It can produce very different information in comparison to the source used to ask question (e.g., book chapter)

## Note about discussions

- Discussion answers like “I agree because of bla-bla-bla” won’t be graded — they do not provide any value

## Note about discussions

- Discussion answers like “I agree because of bla-bla-bla” won’t be graded — they do not provide any value
- Only the answers with some non-trivial arguments that:
  - ▶ either contradict the initial post,
  - ▶ or add some non-obvious missing things to the initial messagewill be considered as graded ones

# Course logistics

- Course grading will be done based on attendance, assignments, discussions, (optional: some small programming tasks) and the final exam.

# Course logistics

- Course grading will be done based on attendance, assignments, discussions, (optional: some small programming tasks) and the final exam.
- Contribution:
  - 50%: attendance, assignments, discussions
  - 50%: exam

# Course logistics

- Course grading will be done based on attendance, assignments, discussions, (optional: some small programming tasks) and the final exam.
- Contribution:
  - 50%: attendance, assignments, discussions
  - 50%: exam
- Preliminary grading scale:

| Grade | Percent accumulated |
|-------|---------------------|
| A     | 90-100 %            |
| B     | 75-89 %             |
| C     | 60-74 %             |



# Late Submission Policy

Late submission deduction percent: **15% every day**;

# Late Submission Policy

Late submission deduction percent: **15% every day**;

- It means that if you're **7 days late** than no need to submit: you'll get **0 score** anyway.

# Makeup Policy

Missing onground class:

- Student's grade is dropped by 10%;

# Makeup Policy

Missing onground class:

- Student's grade is dropped by 10%;

Missing Exam:

- A student gets 0 for the Exam;

# Makeup Policy

Missing onground class:

- Student's grade is dropped by 10%;

Missing Exam:

- A student gets 0 for the Exam;

Missing Assignment in time (1 week, please refer to the *Late Submission Policy*):

- No makeups, i.e. 0 for the Assignment;

# Makeup Policy

Missing onground class:

- Student's grade is dropped by 10%;

Missing Exam:

- A student gets 0 for the Exam;

Missing Assignment in time (1 week, please refer to the *Late Submission Policy*):

- No makeups, i.e. 0 for the Assignment;

Unless:

- A Student has a serious medical condition, and this condition is validated by a hospital or licensed California physician (in English)
- The Student contacts in time our chair (Donna Dulo) and describes the situation and provides all the needed proofs
- The Student notifies in time Professor about the situation with the confirmation from our chair (Donna Dulo)

# Key ingredients of ML

- Current ML is: half Math, half Programming

# Key ingredients of ML

- Current ML is: half Math, half Programming
  - ▶ **Math:** for research and design of ML algorithms



# Key ingredients of ML

- Current ML is: half Math, half Programming
  - ▶ **Math:** for research and design of ML algorithms
  - ▶ **Programming:** usage and tuning of ML algorithms

# Key ingredients of ML

- Current ML is: half Math, half Programming
  - ▶ **Math:** for research and design of ML algorithms
  - ▶ **Programming:** usage and tuning of ML algorithms
- Hopefully we could touch on both a little

- Course page: <https://github.com/fatheral/sofia-ml-2024>
- The professor's lectures will be uploaded there

# What is Artificial Intelligence?

## Natural Intelligence (human)

- Able to perceive the information, analyze it, make decisions based on this analysis

# What is Artificial Intelligence?

## Natural Intelligence (human)

- Able to perceive the information, analyze it, make decisions based on this analysis

## Artificial Intelligence

- (Strong) The same as natural intelligence, but computer is instead of human

# What is Artificial Intelligence?

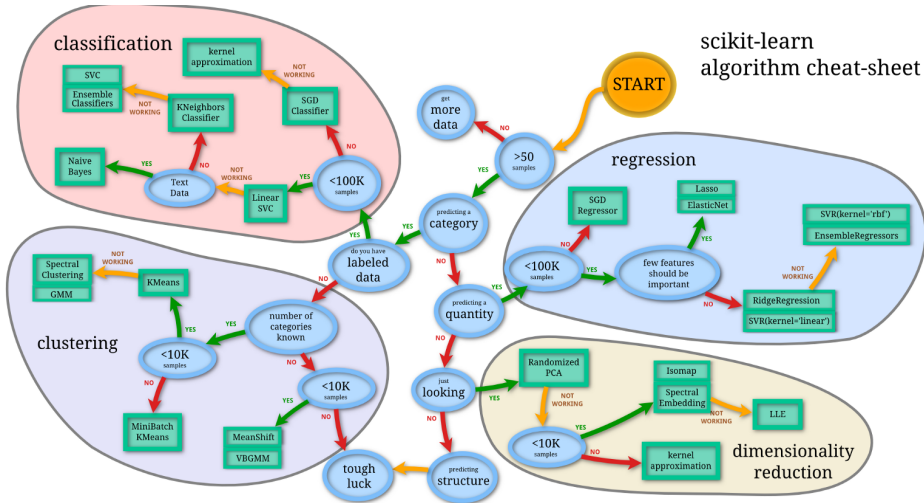
## Natural Intelligence (human)

- Able to perceive the information, analyze it, make decisions based on this analysis

## Artificial Intelligence

- (Strong) The same as natural intelligence, but computer is instead of human
- (**Weak**) Algorithm which is able to be trained using the input data in order to do tasks afterward — instead of human

# Scikit-Learn<sup>2</sup> Roadmap



<sup>2</sup>[https://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/](https://scikit-learn.org/stable/tutorial/machine_learning_map/)

# (Tentative) future content

## Theoretic part

- Quality metrics
  - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...



# (Tentative) future content

## Theoretic part

- Quality metrics
  - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Classification task and optimization
  - kNN, Linear Classifiers, Stochastic Gradient Descent, PCA, ...

# (Tentative) future content

## Theoretic part

- Quality metrics
  - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Classification task and optimization
  - kNN, Linear Classifiers, Stochastic Gradient Descent, PCA, ...
- Regression task
  - Linear Regression, Elastic Net, Ridge Regression, LASSO, ...

# (Tentative) future content

## Theoretic part

- Quality metrics
  - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Classification task and optimization
  - kNN, Linear Classifiers, Stochastic Gradient Descent, PCA, ...
- Regression task
  - Linear Regression, Elastic Net, Ridge Regression, LASSO, ...
- Ensembles (unlikely but will try)
  - Bootstrapping, Bagging, Boosting, ...

# (Tentative) future content

## Theoretic part

- Quality metrics
  - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Classification task and optimization
  - kNN, Linear Classifiers, Stochastic Gradient Descent, PCA, ...
- Regression task
  - Linear Regression, Elastic Net, Ridge Regression, LASSO, ...
- Ensembles (unlikely but will try)
  - Bootstrapping, Bagging, Boosting, ...

## Practice part

- Data processing and analysis by Python
  - Scikit-Learn, Numpy, ...

# What is Machine Learning

In 1959 Arthur Samuel introduced the term “machine learning” into scientific use.

## General definition

**Machine Learning** — the process leading computers to gain ability to show the behavior that wasn't explicitly programmed.

# What is Machine Learning

In 1959 Arthur Samuel introduced the term “machine learning” into scientific use.

## General definition

**Machine Learning** — the process leading computers to gain ability to show the behavior that wasn't explicitly programmed.

In 1997 Tom M. Mitchell introduced more formal definition of a machine learning algorithm.

## Formal definition

A **computer program** is said **to learn** from examples  $E$  for some set of problems  $T$  and a quality metric  $P$  if its performance on problems from  $T$ , as measured by  $P$ , is improved by using examples  $E$ .

# Forerunner of Machine Learning

- People have been trying to predict the future based on their experience since time immemorial.

# Forerunner of Machine Learning

- People have been trying to predict the future based on their experience since time immemorial.
- However, the scientific basis was laid by probability theory (statistics in particular) and linear algebra (as a tool).



# Forerunner of Machine Learning

- People have been trying to predict the future based on their experience since time immemorial.
- However, the scientific basis was laid by probability theory (statistics in particular) and linear algebra (as a tool).
- **1795**: Gauss first uses the least squares method (LSQ) to analyze astronomical observations. In **1805** Legendre first published this method for analyzing the shape of the Earth. At present, LSQ is the simplest way to solve an overdetermined system of linear equations.

# Forerunner of Machine Learning

- People have been trying to predict the future based on their experience since time immemorial.
- However, the scientific basis was laid by probability theory (statistics in particular) and linear algebra (as a tool).
- **1795**: Gauss first uses the least squares method (LSQ) to analyze astronomical observations. In **1805** Legendre first published this method for analyzing the shape of the Earth. At present, LSQ is the simplest way to solve an overdetermined system of linear equations.
- **1901**: Karl Pearson invented the Principal Component Analysis (PCA) — a master method for data dimensionality reduction.

# Forerunner of Machine Learning

- People have been trying to predict the future based on their experience since time immemorial.
- However, the scientific basis was laid by probability theory (statistics in particular) and linear algebra (as a tool).
- **1795**: Gauss first uses the least squares method (LSQ) to analyze astronomical observations. In **1805** Legendre first published this method for analyzing the shape of the Earth. At present, LSQ is the simplest way to solve an overdetermined system of linear equations.
- **1901**: Karl Pearson invented the Principal Component Analysis (PCA) — a master method for data dimensionality reduction.
- **1906**: Andrey Andreyevich Markov develops the apparatus of Markov chains, which in **1913** he uses to study the text “Eugene Onegin”. Markov chains are used to generate and recognize signals.

## Historic reference

- **1950:** Alan Turing creates the Turing test to evaluate the intelligence of a computer.

## Historic reference

- **1950**: Alan Turing creates the Turing test to evaluate the intelligence of a computer.
- **1951**: Marvin Minsky created the first SNARC learning machine with a randomly connected neural network. In **1959**, he co-founded the Artificial Intelligence Laboratory at MIT.

## Historic reference

- **1950:** Alan Turing creates the Turing test to evaluate the intelligence of a computer.
- **1951:** Marvin Minsky created the first SNARC learning machine with a randomly connected neural network. In **1959**, he co-founded the Artificial Intelligence Laboratory at MIT.
- **1952:** Arthur Samuel creates the first checkers program for the IBM 701. In **1955** Samuel adds self-learning capability to the program.

## Historic reference

- **1950:** Alan Turing creates the Turing test to evaluate the intelligence of a computer.
- **1951:** Marvin Minsky created the first SNARC learning machine with a randomly connected neural network. In **1959**, he co-founded the Artificial Intelligence Laboratory at MIT.
- **1952:** Arthur Samuel creates the first checkers program for the IBM 701. In **1955** Samuel adds self-learning capability to the program.
- **1958:** Frank Rosenblatt invented the Perceptron — the first artificial neural network — and built the first “Mark-1” brain computer. *New York Times: The Perceptron is “the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence”.*

## Historic reference

- **1950:** Alan Turing creates the Turing test to evaluate the intelligence of a computer.
- **1951:** Marvin Minsky created the first SNARC learning machine with a randomly connected neural network. In **1959**, he co-founded the Artificial Intelligence Laboratory at MIT.
- **1952:** Arthur Samuel creates the first checkers program for the IBM 701. In **1955** Samuel adds self-learning capability to the program.
- **1958:** Frank Rosenblatt invented the Perceptron — the first artificial neural network — and built the first “Mark-1” brain computer. *New York Times: The Perceptron is “the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence”.*
- **1963:** Lawrence Roberts formulated the thesis of computer vision in his dissertation at MIT.



## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.

## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.
- **1965:** One of the first books on machine learning (pattern classification) was published — Nilsson N. Learning Machines, McGraw Hill.

## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.
- **1965:** One of the first books on machine learning (pattern classification) was published — Nilsson N. Learning Machines, McGraw Hill.
- **1966:** Joseph Weizenbaum wrote a computerized conversation simulator ELIZA, able to imitate (or rather parody) a dialogue with a psychotherapist (the program owes its name to the heroine from the play by B. Shaw).

## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.
- **1965:** One of the first books on machine learning (pattern classification) was published — Nilsson N. Learning Machines, McGraw Hill.
- **1966:** Joseph Weizenbaum wrote a computerized conversation simulator ELIZA, able to imitate (or rather parody) a dialogue with a psychotherapist (the program owes its name to the heroine from the play by B. Shaw).
- **1967:** Alexey Ivakhnenko and Valentin Lapa publish the first general working learning algorithm for deep multilayer perceptrons for supervised learning problems.

## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.
- **1965:** One of the first books on machine learning (pattern classification) was published — Nilsson N. Learning Machines, McGraw Hill.
- **1966:** Joseph Weizenbaum wrote a computerized conversation simulator ELIZA, able to imitate (or rather parody) a dialogue with a psychotherapist (the program owes its name to the heroine from the play by B. Shaw).
- **1967:** Alexey Ivakhnenko and Valentin Lapa publish the first general working learning algorithm for deep multilayer perceptrons for supervised learning problems.
- **1986:** Rina Dechter introduced the term “Deep Learning” to the machine learning community.

## Historic reference

- **1963:** Vladimir Vapnik and Aleksey Chervonenkis invented the SVM algorithm.
- **1965:** One of the first books on machine learning (pattern classification) was published — Nilsson N. Learning Machines, McGraw Hill.
- **1966:** Joseph Weizenbaum wrote a computerized conversation simulator ELIZA, able to imitate (or rather parody) a dialogue with a psychotherapist (the program owes its name to the heroine from the play by B. Shaw).
- **1967:** Alexey Ivakhnenko and Valentin Lapa publish the first general working learning algorithm for deep multilayer perceptrons for supervised learning problems.
- **1986:** Rina Dechter introduced the term “Deep Learning” to the machine learning community.
- **1997:** The Deep Blue computer beat world chess champion Garry Kasparov.

## Historic reference

- **2010:** Founding of DeepMind.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)



## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.
- **2011:** IBM Watson AI supercomputer wins TV quiz show *Jeopardy!*

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.
- **2011:** IBM Watson AI supercomputer wins TV quiz show *Jeopardy!*
- **2014:** Facebook invented the DeepFace software algorithm for face recognition. The accuracy of the algorithm was 97%.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.
- **2011:** IBM Watson AI supercomputer wins TV quiz show *Jeopardy!*
- **2014:** Facebook invented the DeepFace software algorithm for face recognition. The accuracy of the algorithm was 97%.
- **2016:** AlphaGo, developed by (now) Google's DeepMind, won four out of five games against Korea's world Go champion Lee Se-dol. The computer won the last game with perfect information <sup>3</sup> against a human (an example of a game with incomplete information is poker, although robots are already beginning to perform successfully there).

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.
- **2011:** IBM Watson AI supercomputer wins TV quiz show *Jeopardy!*
- **2014:** Facebook invented the DeepFace software algorithm for face recognition. The accuracy of the algorithm was 97%.
- **2016:** AlphaGo, developed by (now) Google's DeepMind, won four out of five games against Korea's world Go champion Lee Se-dol. The computer won the last game with perfect information <sup>3</sup> against a human (an example of a game with incomplete information is poker, although robots are already beginning to perform successfully there).
- **2016:** OpenAI, a non-profit research company, is launched with the support of Elon Musk.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

## Historic reference

- **2010:** Founding of DeepMind.
- **2011:** Andrew Ng, Greg Corrado and Jeff Dean founded Google Brain.
- **2011:** IBM Watson AI supercomputer wins TV quiz show *Jeopardy!*
- **2014:** Facebook invented the DeepFace software algorithm for face recognition. The accuracy of the algorithm was 97%.
- **2016:** AlphaGo, developed by (now) Google's DeepMind, won four out of five games against Korea's world Go champion Lee Se-dol. The computer won the last game with perfect information <sup>3</sup> against a human (an example of a game with incomplete information is poker, although robots are already beginning to perform successfully there).
- **2016:** OpenAI, a non-profit research company, is launched with the support of Elon Musk.
- **2022:** OpenAI, a (not so) non-profit research company, provided the breakthrough in LLMs: ChatGPT.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Perfect\\_information](https://en.wikipedia.org/wiki/Perfect_information)

# Machine Learning Paradigms

## Definitions

- $X$  — set of objects
- $Y$  — set of (correct) answers/labels
- $y : X \rightarrow Y$  — the unknown dependency

# Machine Learning Paradigms

## Definitions

- $X$  — set of objects
- $Y$  — set of (correct) answers/labels
- $y : X \rightarrow Y$  — the unknown dependency

## Machine learning paradigms

- **Supervised** (now)
  - Sufficient amount of training material, i.e. pairs  $(x_i, y_i)$

# Machine Learning Paradigms

## Definitions

- $X$  — set of objects
- $Y$  — set of (correct) answers/labels
- $y : X \rightarrow Y$  — the unknown dependency

## Machine learning paradigms

- **Supervised** (now)
  - Sufficient amount of training material, i.e. pairs  $(x_i, y_i)$
- Semi-supervised
  - Few labeled data  $(x_i, y_i)$  and many unlabeled examples  $x_j$



# Machine Learning Paradigms

## Definitions

- $X$  — set of objects
- $Y$  — set of (correct) answers/labels
- $y : X \rightarrow Y$  — the unknown dependency

## Machine learning paradigms

- **Supervised** (now)
  - Sufficient amount of training material, i.e. pairs  $(x_i, y_i)$
- Semi-supervised
  - Few labeled data  $(x_i, y_i)$  and many unlabeled examples  $x_j$
- *Unsupervised* (in future lectures?)
  - No labeled pairs, only  $x_i$  examples

# Machine Learning Paradigms

## Definitions

- $X$  — set of objects
- $Y$  — set of (correct) answers/labels
- $y : X \rightarrow Y$  — the unknown dependency

## Machine learning paradigms

- **Supervised** (now)
  - Sufficient amount of training material, i.e. pairs  $(x_i, y_i)$
- Semi-supervised
  - Few labeled data  $(x_i, y_i)$  and many unlabeled examples  $x_j$
- *Unsupervised* (in future lectures?)
  - No labeled pairs, only  $x_i$  examples
- Reinforcement
  - Action generation based on interaction with the environment

## Mandatory external links to read

- 1 Please read the entire IBM booklet on the Introduction to Machine Learning. This is a quick and basic introduction but really gives a solid overview of the field of machine learning to help you understand the scope of practice or machine learning as related to Artificial Intelligence.

## Mandatory external links to read

- 1 Please read the entire IBM booklet on the Introduction to Machine Learning. This is a quick and basic introduction but really gives a solid overview of the field of machine learning to help you understand the scope of practice or machine learning as related to Artificial Intelligence.
- 2 Read the brief explanation of Occam's Razor

## Mandatory external links to read

- 1 Please read the entire IBM booklet on the Introduction to Machine Learning. This is a quick and basic introduction but really gives a solid overview of the field of machine learning to help you understand the scope of practice or machine learning as related to Artificial Intelligence.
- 2 Read the brief explanation of Occam's Razor
- 3 Read the wiki pages on Weak artificial intelligence, Artificial general intelligence, and the article named Weak AI vs Strong AI: Exploring Key Differences and Future Potential of AI (need either LinkedIn or Google account to read the full text)

## Mandatory external links to read

- 1 Please read the entire IBM booklet on the Introduction to Machine Learning. This is a quick and basic introduction but really gives a solid overview of the field of machine learning to help you understand the scope of practice or machine learning as related to Artificial Intelligence.
- 2 Read the brief explanation of Occam's Razor
- 3 Read the wiki pages on Weak artificial intelligence, Artificial general intelligence, and the article named Weak AI vs Strong AI: Exploring Key Differences and Future Potential of AI (need either LinkedIn or Google account to read the full text)
- 4 Read History and evolution of machine learning: A timeline and the Timeline of Machine Learning pages

# Takeaway notes

- 1 Please go through all the materials of **Module 0**

# Takeaway notes

- 1 Please go through all the materials of **Module 0**
- 2 Please introduce yourself, complete the **Assignment 1** and discuss the question inside “**Module 1 DQ**”



# Takeaway notes

- 1 Please go through all the materials of **Module 0**
- 2 Please introduce yourself, complete the **Assignment 1** and discuss the question inside “**Module 1 DQ**”
- 3 We are going to cover the most important things needed for ML, and will have small optional programming tasks

# Takeaway notes

- ➊ Please go through all the materials of **Module 0**
- ➋ Please introduce yourself, complete the **Assignment 1** and discuss the question inside “**Module 1 DQ**”
- ➌ We are going to cover the most important things needed for ML, and will have small optional programming tasks
- ➍ ML History is intriguing!

# Thank you!