AIcademy

Introduction to Machine Learning and AI

What is AI today ?

Created with the
Bing Image Creator
[DALL-E]

Prompt:
An artificial steam punk brain
as the core of a computer

Artificial Intelligence predicts machine failures, and automatically asks for replacement parts before the machine halts.

A Paper Machine that can foresee the strength of the paper and adapt the machine speed to avoid failure using AI.

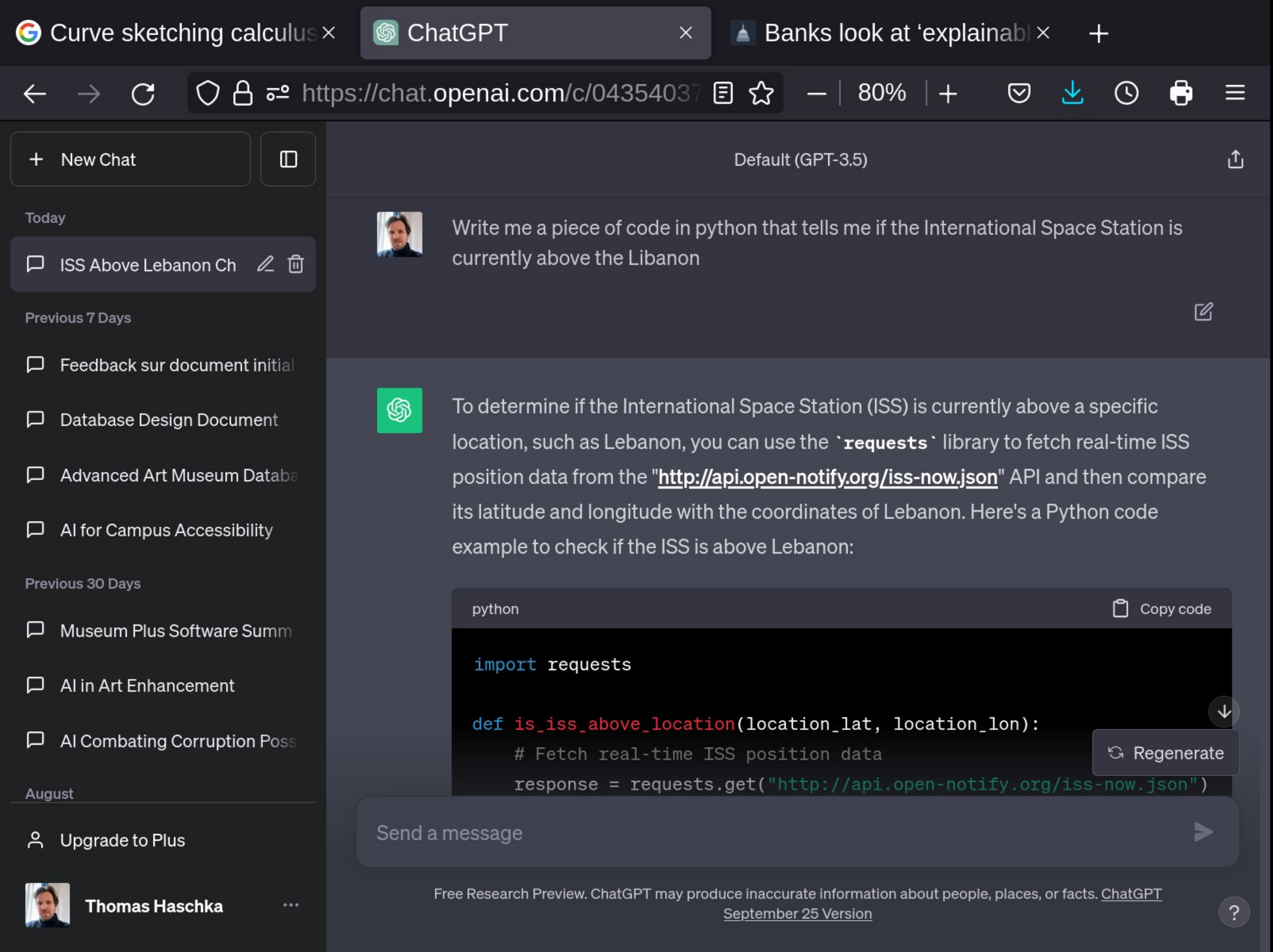
Self Driving
Taxis

are a reality
today in
China

thanks to
artificial
intelligence

AI protects
Credit Card users

by automatically
detecting fraudulent
behavior in
billions of transactions
every day.



AI enables sophisticated chatbots like ChatGPT:

They can write computer code,

give juridical advice,

translate texts,

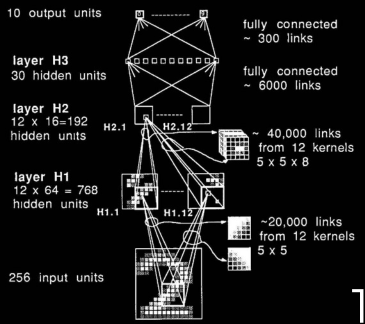
and do many more things.

AI Timeline

Artificial Intelligence

Machine Learning

Cognitive Computing:



1950
Theoretical Concepts
i.e. John Neumann
The computer and
the Brain

1980
Better algorithms
and progress in
computing power
leads to first results
i.e. OCR with SVM

2010
“Easily” programmable
GPUs perform large
parallel computations
which enables
recent breakthroughs

2020s AI becomes a topic of the General Public

Breakthroughs in Generative AI using the transformer model brought:

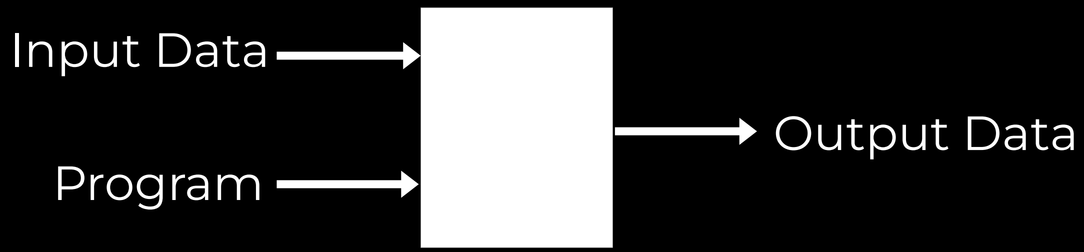
- Chatgpt, a conversational chatbot.
- Dall-E, Stable Diffusion, etc: Image generators powered by neural networks.

Machine Learning

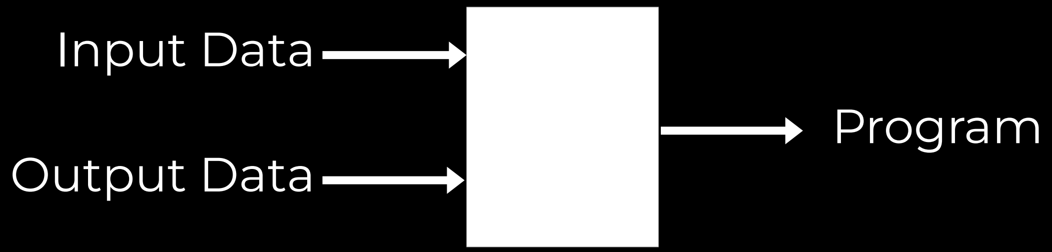
- Further broken down into:
  - *Supervised:*
    Trained with Data and Expected result.
  - *Unsupervised:*
    i.e. clustering, only input data is shown to the model.
  - *Reinforcement Learning:*
    Allow the computer to train itself optimizing a reward function.

Allows for new computing paradigms

Traditional Computing:



Supervised ML:



Machine Learning Definition:

Training Data:   X_train

Labels: Dog Cat Y_train

Builds a Model: f(X) → Y

In the unsupervised case no labels do exist but are inferred by the algorithm.

Machine Learning Design Cycle
Building Models and Evaluating them

begin

↓

collect data

↓

i.e. Size, Height — choose features

↓

ML algorithm — choose model

↓

train model

↓

<u>Evaluate model</u>

↓

done

How do you
evaluate your model ?

How to evaluate the model

- Depends on the data available and use case
- **Training Set / Test Set** Split is good
- **Cross Validation** is better
- Further methods exist

Training / Testset Split

- If labeled data is available:
  - Train the data on a part of the data
  - Test the trained model with data that was not used during training.
  - Training and Testing set choice depends on the dataset. Imagine Fraud detection, where you have a fraudulent case only once a year but daily data is available.
    - In this case you have to make sure that enough fraudulent cases are part of the training set and that you still can detect them in the testset.

Cross Validation

- You chose different splits of the dataset and validate:

Yields multiple model performance values depending on split.

Using basic statistics i.e. Mean, Deviation one can evaluate a models performance.

Training/Test set splits generally chosen at constant size but with random datapoints.

Languages for AI/ML

- Python
- Julia
- C/C++
- Java
- Lisp

Languages for AI/ML

- Depends largely on use case:
  - **Historically LISP was dominant** in the AI community for code morphing technologies and lambda syntax which allowed programs to manipulate themselves.
  - Today **on the application side python dominates**, but most of the code is written in **C/C++ for performance reasons on the backend**.
  - A scikit learn call to a support vector machine calls a C function training the support vector machine.

Building a simple machine learning model.
Linear Regression

Blood Pressure Dataset

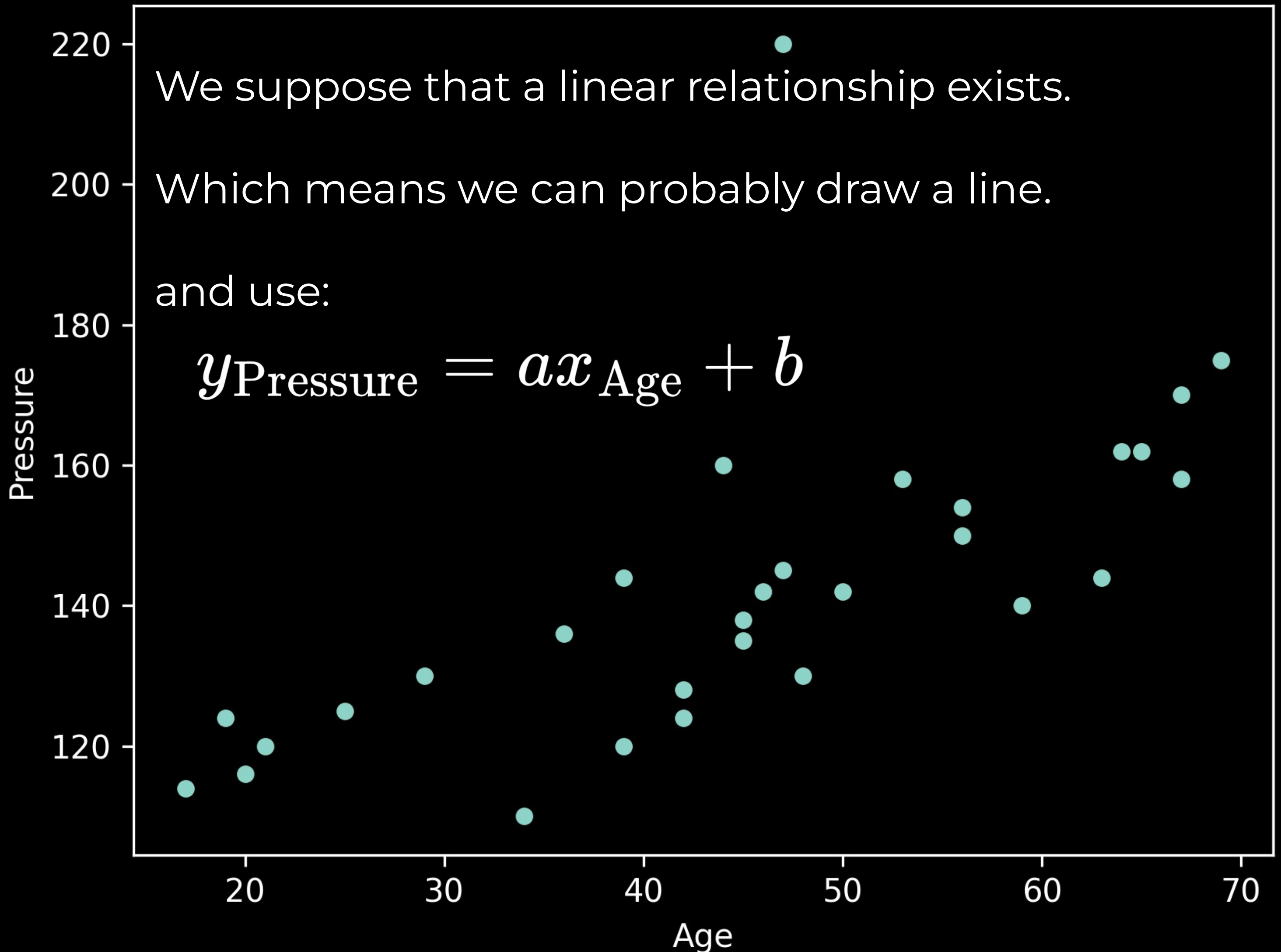
Index	Age	Pressure
1	39	144
2	47	220
3	45	138
4	47	145
5	65	162
6	46	142
7	67	170
8	42	124
9	67	158
10	56	154

We suppose that a linear relationship exists.

Which means we can probably draw a line.

and use:

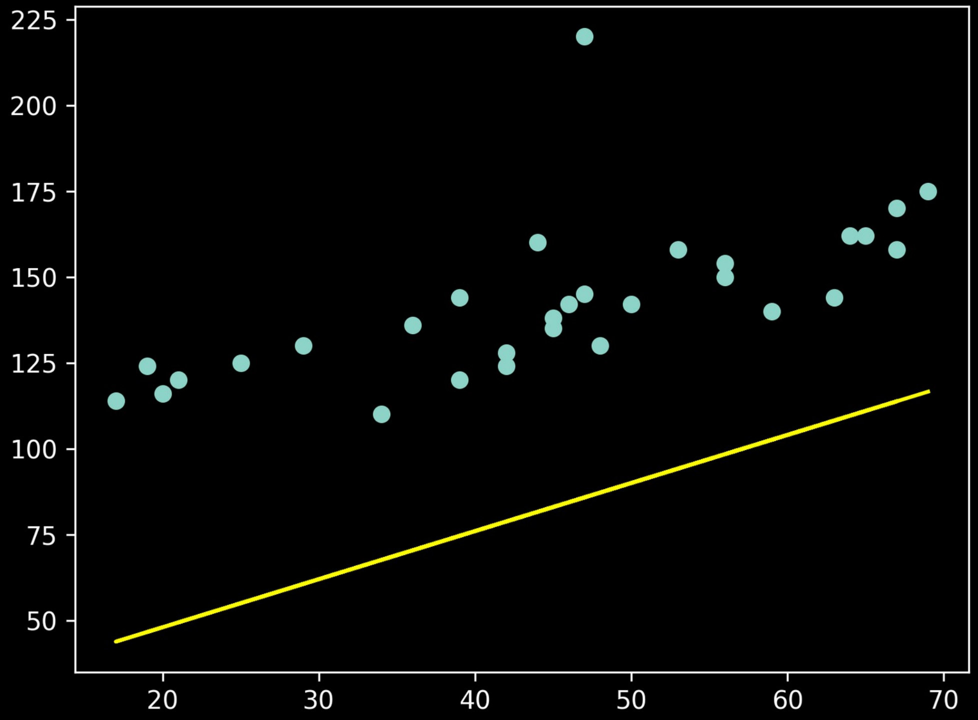
$$y_{\text{Pressure}} = a x_{\text{Age}} + b$$

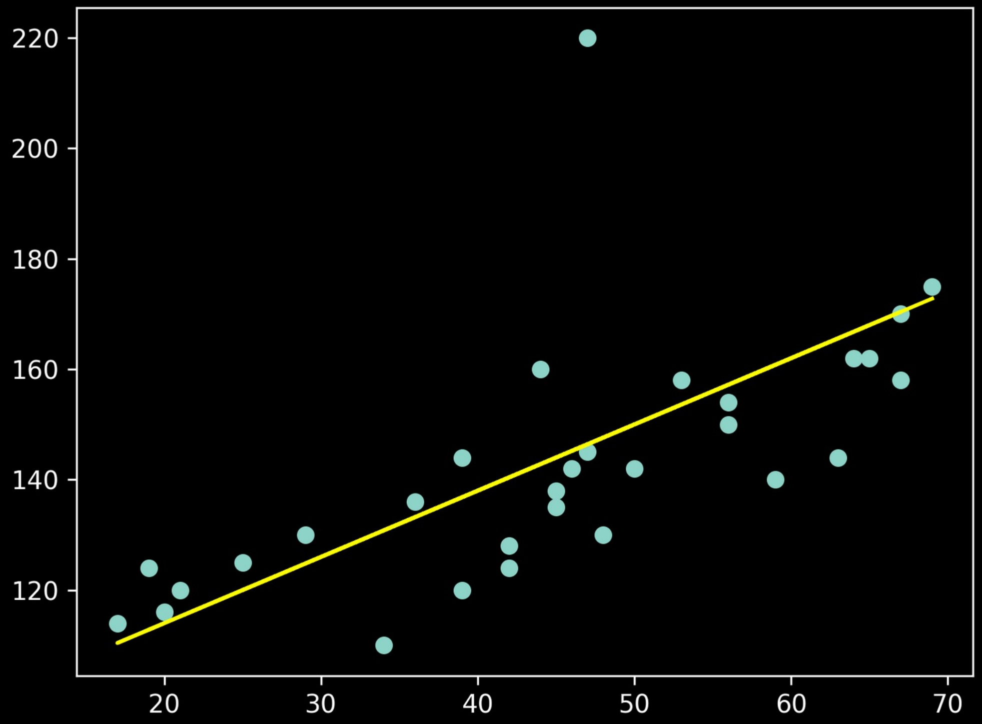


We try some values for a and b:

a = 1.2 … steepness of the curve
b = 20 … starting point (Y at X=0)

a = 1.1
b = 80





But we want to find optimal parameters for our forecast model!

Linear Models

A linear model is in the most generalized terms one that can be modeled using the equation:

$$f(x) = AX + B \approx Y$$

Linear Models

$$f(x) = AX + B \approx Y$$

X Dataset: In general, columns contain the features of a sample and the number of rows is the number of samples in the dataset.

Y Labels: For instance X can contain the features of a House, like number of bathrooms, living area size and Y contains the Prize Y can in most general forms also contain multiple labels.

Linear Models

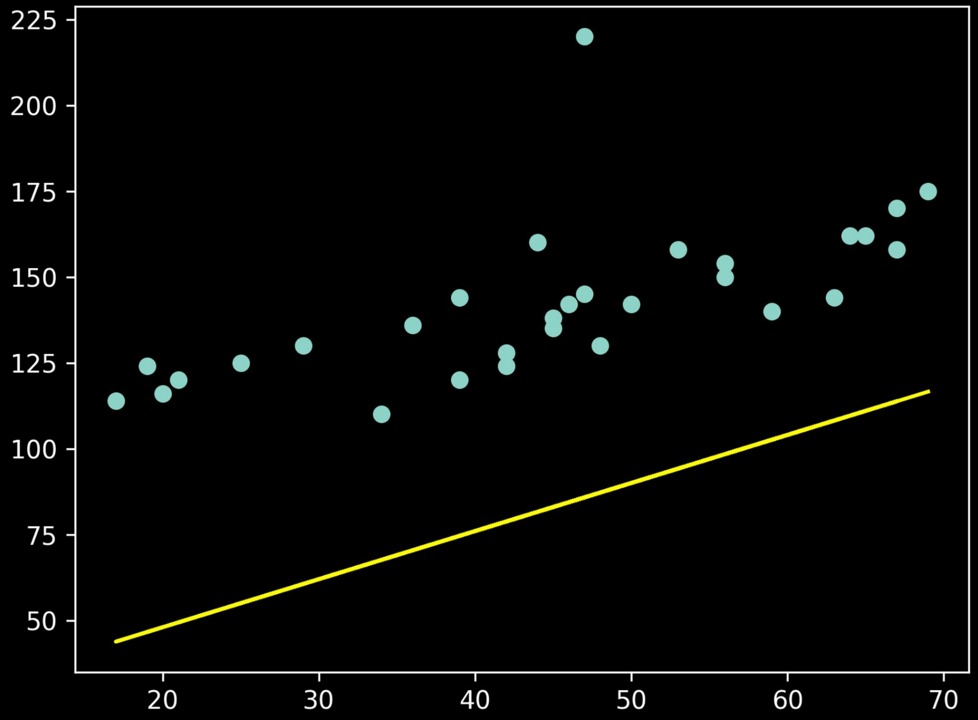
$$f(x) = AX + B \approx Y$$

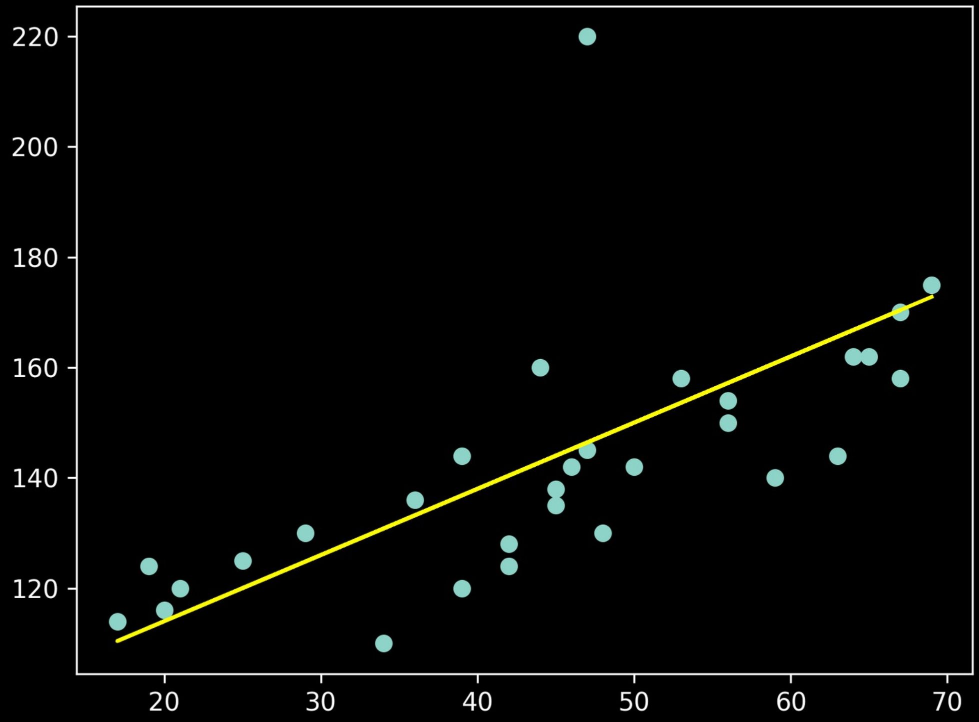
A
B

Matrices that define a linear model. These are learned by a loss function / objective function and an optimization routine, i.e. Gradient Descent.

For linear models with a loss function modeled after the euclidean norm, A and B are analytically solvable. (χ^2 optimization)

In order to find the optimal curve we need first to measure the error we make:

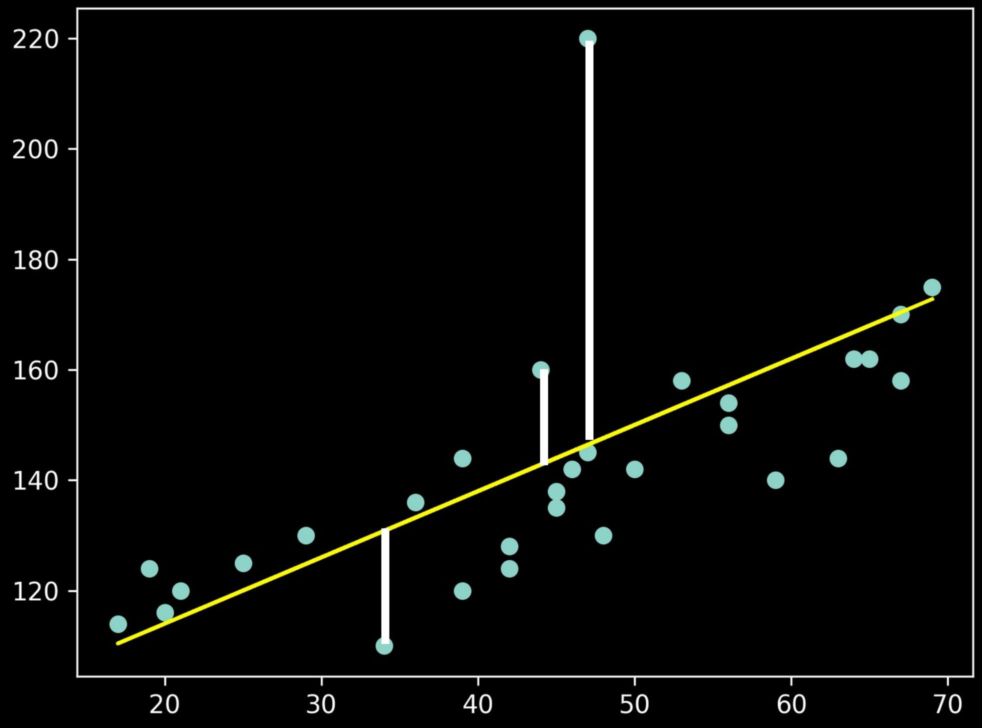




How can we measure how far our curve is from the points.

Loss function

Describes the error between the expected outcome and the prediction of the model.



The Loss function is some sort of sum of the distances of the actual points to the curve i.e.:

- Sum of squared distances (least squared)
- sum of absolute distances (least absolutes)

Loss Function

Choose your model parameters, in this case A, B so that:

$$\min L = \min |AX + B - Y|$$

In a more general case, i.e. even non linear models one can write:

$$\min L = \min |f(X) - Y|$$

Loss function choices

L_p norm:

- p=2 Euclidean norm (least squares)
- p=1 Manhattan distance (least absolutes)

$$L_p = \left[\sum_{i=1}^{n} (f(X)_i - Y_i)^p \right]^{\frac{1}{p}}$$

n … number of samples, of humans in the dataset.

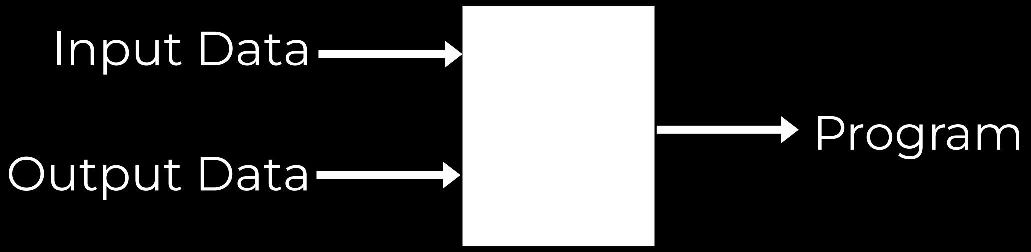
i … sample index, one human i with age Xi and blood pressure Yi

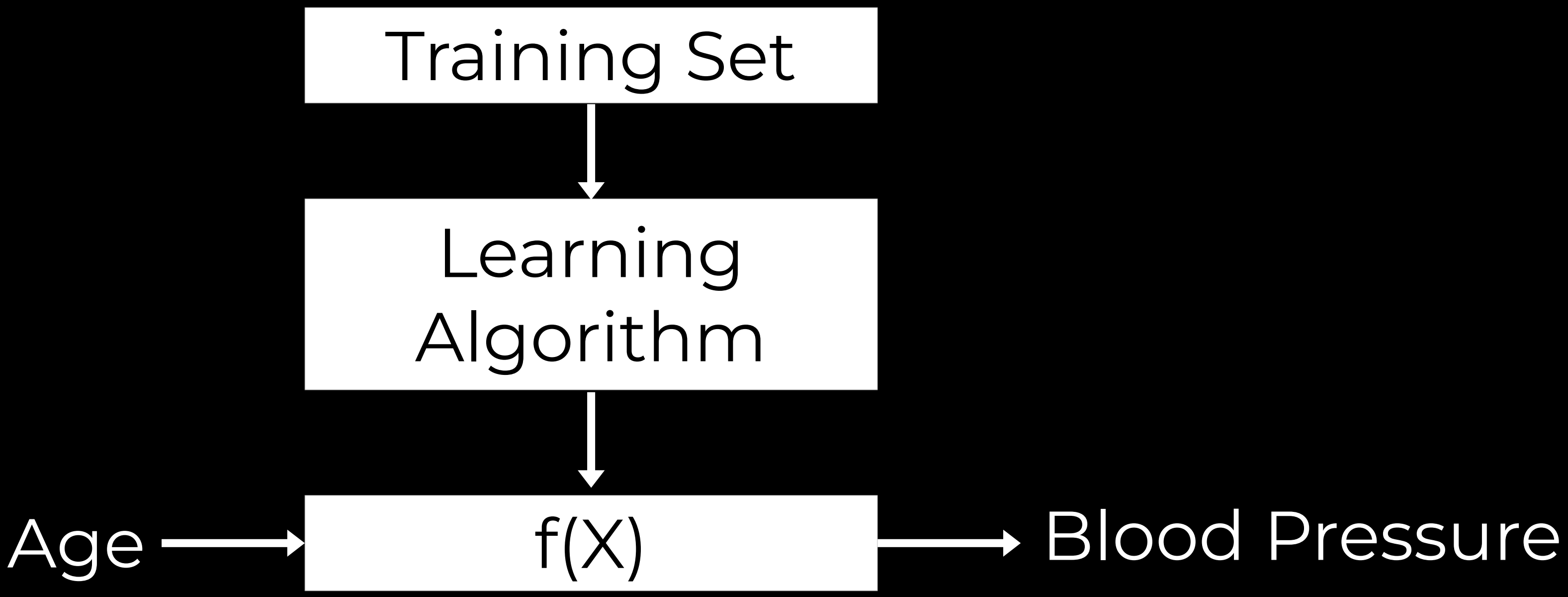
Most simple machine learning model: linear curve fitting

- $f(x) = ax+b$: Linear hypotheses with 2 parameters (a,b).
- Blood pressure dataset:
  - n: 30 participants
  - x: Age [feature]
  - y: Blood pressure [target or label]
- Can we predict one’s blood pressure from its age ?

Building a ML Model

- Supervised Learning
- Train the model with known inputs (X) and outputs (Y)




Training Set
Learning Algorithm
Age
f(X)
Blood Pressure

Optimize model parameters

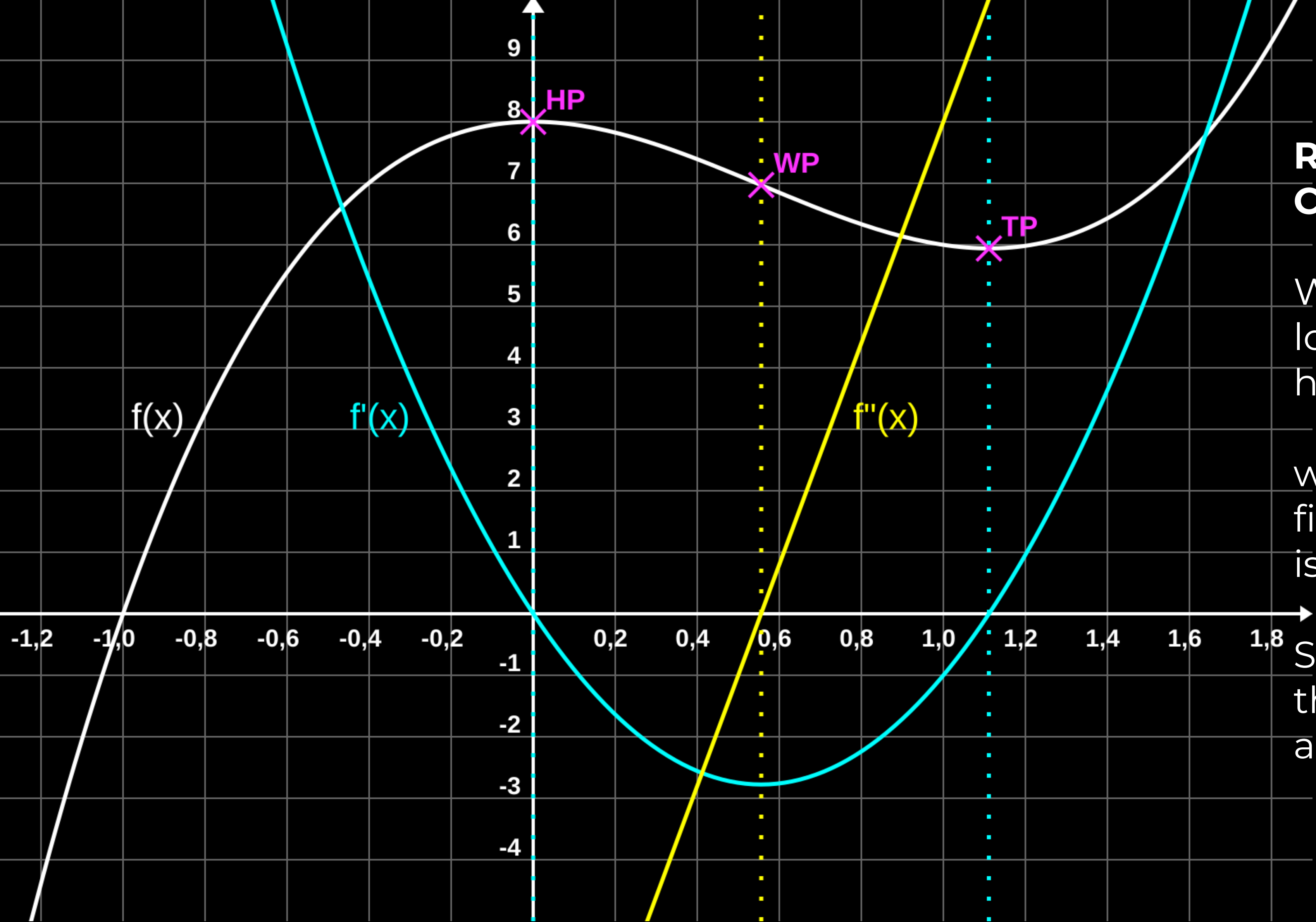
$$\|f(X, w) - Y\|$$

Choose model parameters,
a,b so that the above
becomes minimal.

Minimum of the Loss function

Least Squares Case (p=2):

$$L_p = \left[\sum_{i=1}^{n}(f(X)_i - Y_i)^p\right]^{\frac{1}{p}}$$



Remember Curve sketching:

We find the local minimum here: TP

where the first derivative is 0

So let us take the derivative and set it 0

Can be analytically derived:

$$\frac{\partial}{\partial a}\sum_{i=1}^{n}(aX_i + b - Y_i)^2 = 0$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{n}(aX_i + b - Y_i)^2 = 0$$

Taking the partial derivatives setting them 0 and solving for a and b.

Solution:

Mean of all Yi
(Blood Pressures)

$$a = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=0}^{n}(X_i - \bar{X})^2}$$

Mean of all Xi
(Ages)

$$b = \bar{Y} - a\bar{X}$$

Minimum for the Loss function

Least Absolutes Case (p=1):
More resistant to outliers !

$$L_p = \left[\sum_{i=1}^{n} (f(X)_i - Y_i)^p \right]^{\frac{1}{p}}$$

No analytical solution exists

But numerical solutions might help:

- Gradient Descent
  - Simple First Order Method
- Broyden–Fletcher–Goldfarb–Shanno (BFGS)
  - 2nd Order Method based on Newtons Method
- And many others do exist