NYU CS-GY 6923 Machine Learning

Prof. Pavel Izmailov

AlphaFold

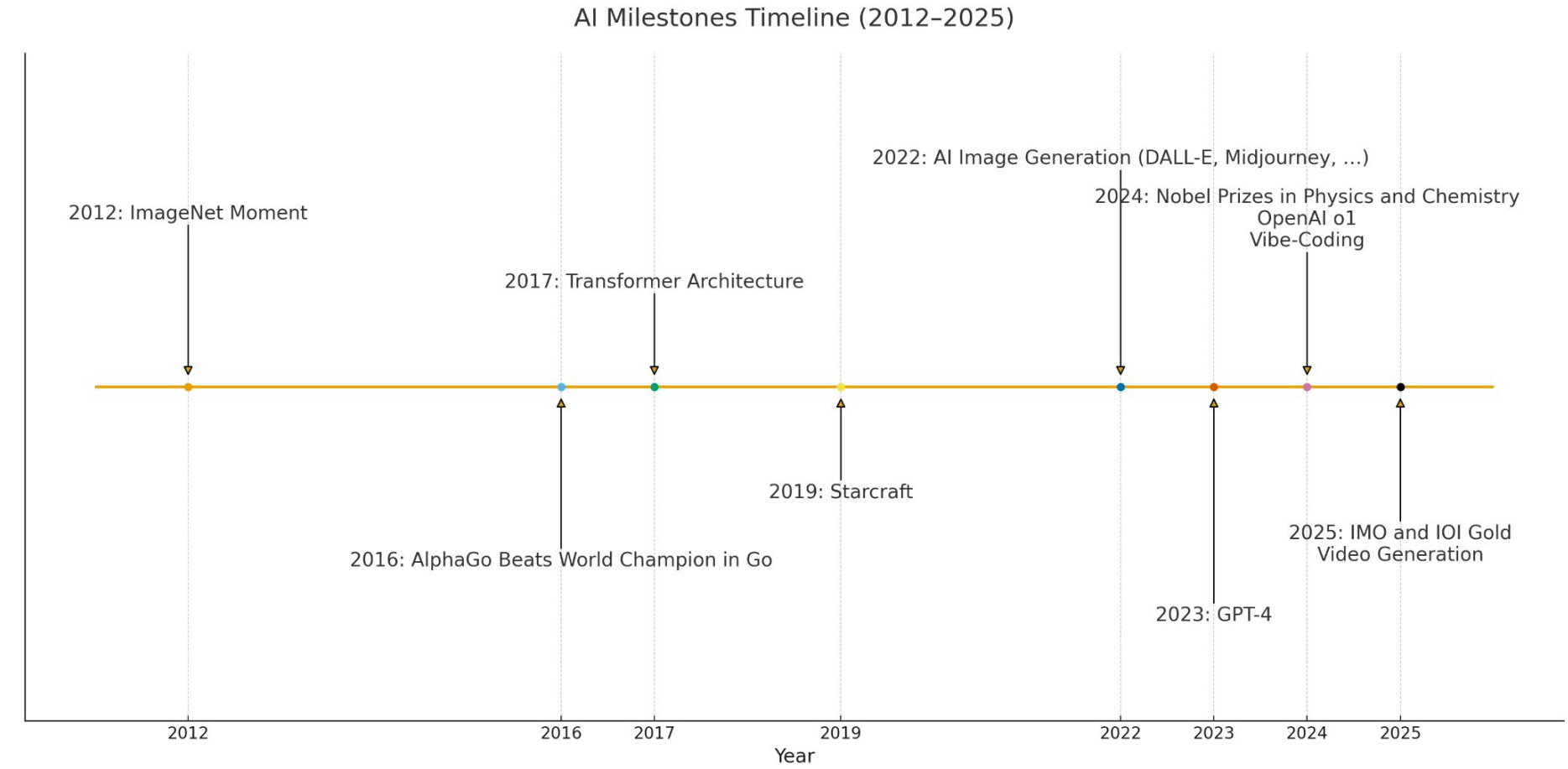
CLAUDE CODE

ChatGPT

deepseek

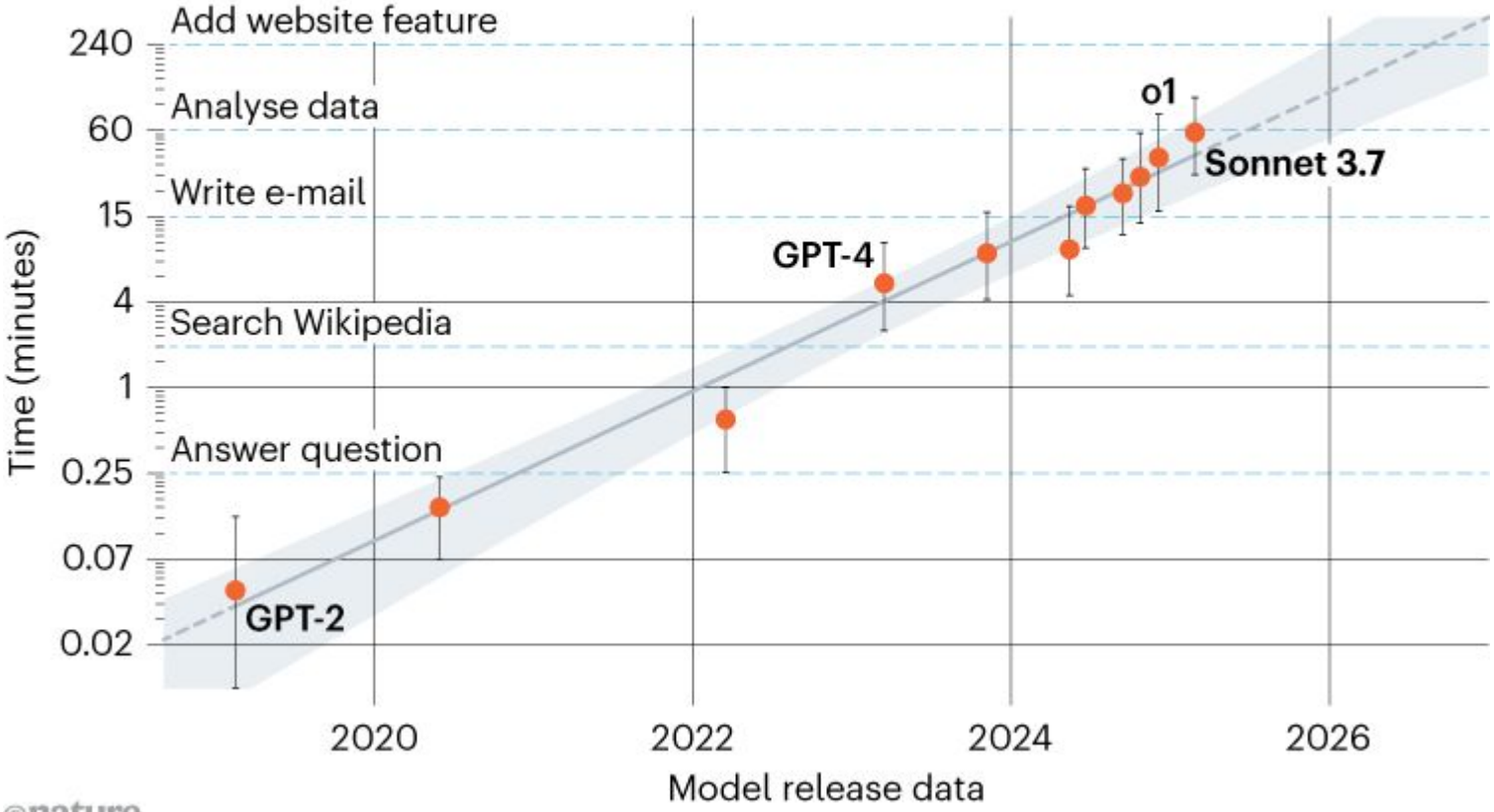
Highlights

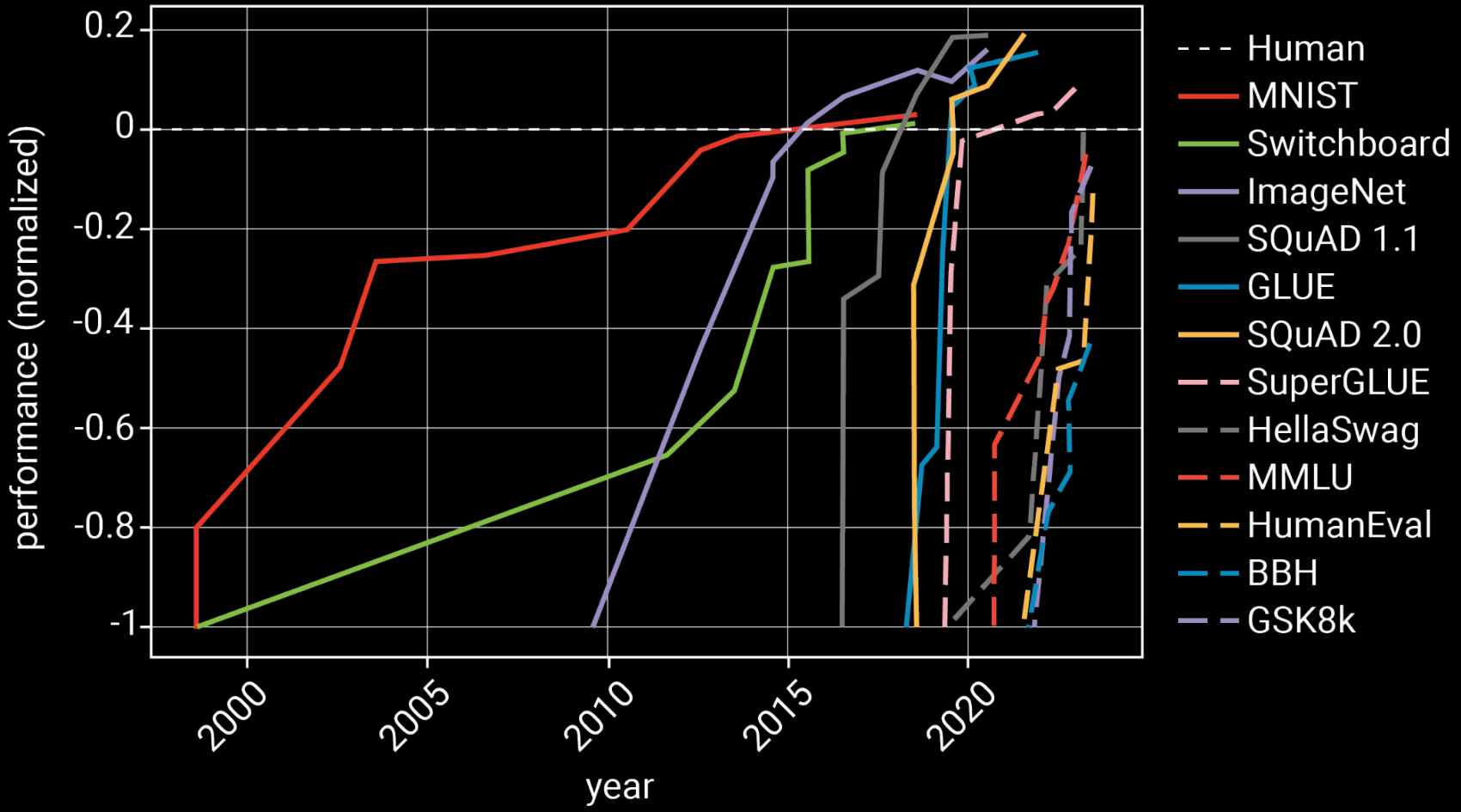
- 2012: ImageNet Moment
- 2016: AlphaGo Beats World Champion in Go
- 2017: Transformer Architecture
- 2019: Starcraft
- 2022: AI Image Generation (DALL-E, Midjourney, …)
- 2023: GPT-4
- 2024:
  - Nobel Prizes in Physics and Chemistry
  - OpenAI o1
  - Vibe-Coding
- 2025 IMO and IOI Gold


AI Milestones Timeline (2012–2025)
2012: ImageNet Moment
2016: AlphaGo Beats World Champion in Go
2017: Transformer Architecture
2019: Starcraft
2022: AI Image Generation (DALL-E, Midjourney, ...)
2023: GPT-4
2024: Nobel Prizes in Physics and Chemistry
OpenAI o1
Vibe-Coding
2025: IMO and IOI Gold
Video Generation
2012
2016
2017
2019
2022
2023
2024
2025
Year

IMPROVING PERFORMANCE

Since 2019, the time humans take to complete tasks that AI models can complete at 50% accuracy has doubled every seven months.





[Kiela et al. 2023]

Goals of this class

- Give you a foundation to understand the main ideas in modern ML
- By the end of the class, you should be able to:
  - Read NeurIPS papers
  - Implement and apply most of the core ML methods
  - Understand basic theoretical analysis and math behind these methods

Grading

- Coding
  - Take-home labs: 20%
- Theory
  - Problem sets: 20%
  - Midterm exam: 25%
  - Final exam: 25%
- Participation: 10%
  - Ask questions, attend office hours, post on Ed.

Team

Majid Daliri
Course Assistant

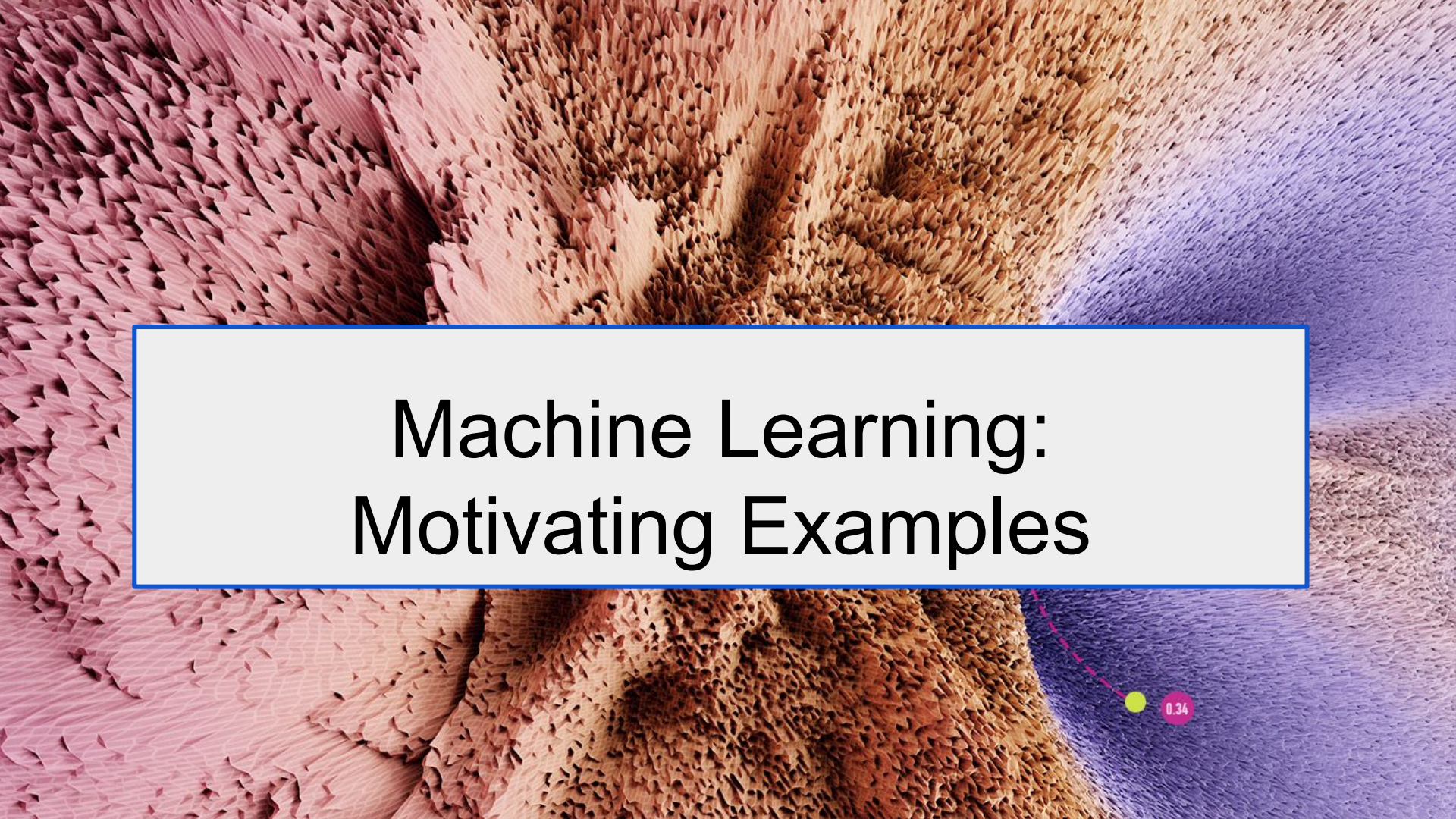
Harsha Mupparaju
Course Assistant

Shubham Rastogi
Course Assistant

- Up to date info: izmailovpavel.github.io/ny_fall25_ml_grad
- Follow Ed discussion, go to office hours

Policy

- Exams
  - Midterm: 10/17
  - Final: 12/19
- Collaboration
  - Fine to discuss problem sets and assignments, but **must write solutions independently**
  - Zero-tolerance policy for copied solutions
- AI tools allowed, make sure they don’t impede your learning



Machine Learning: Motivating Examples

MNIST

Goal: recognize handwritten digits

- Images: 28x28 pixels, black and white
- Labels: 0, ..., 9

How can a computer classify them?

- Idea: hand-craft rules, e.g. if contains two circles then 8, …

Rule-based Systems

- Dominant in AI in 1970s and 80s
- Hard to implement heuristics
  - How to define a circle? A line?
- A lot of manual work
- Unreliable: Impossible to cover all corner cases
- Not scalable
  - What is your heuristic for telling a cheetah from a lion?


ELIZA
He says I am depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
INPUT
It's true. I am unhappy.
DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
INPUT
I need some help, that much seems certain.
WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
INPUT
Perhaps I could learn to get along with my mother.
TELL ME MORE ABOUT YOUR FAMILY
INPUT
My mother takes care of me.
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU
INPUT
My father.
YOUR FATHER
INPUT

Rule-based Systems

ELIZA:

- Try here: https://www.masswerk.at/elizabot/
- Built in 1966
- Uses hand-crafted rules
- Impressive for the time, but by modern standards very bad

Rule-Based Systems

- Used in Siri, Alexa, …

Rule-based Systems

- Dominant in AI in 1970s and 80s
- Hard to implement heuristics
  -
- A lot of manual work
- Unreliable: Impossible to cover all corner cases
- Not scalable

Machine Learning: sketch

Use data to train computers to make predictions.

- Construct a **training dataset**
- Define a **model**: mathematical function that maps from inputs (28x28 black and white pixels) to outputs (labels 0,...,9)
  - The model depends on **parameters**
- **Train** the model: set the parameters so that predictions on training data are good
- Use the model to make predictions on new data

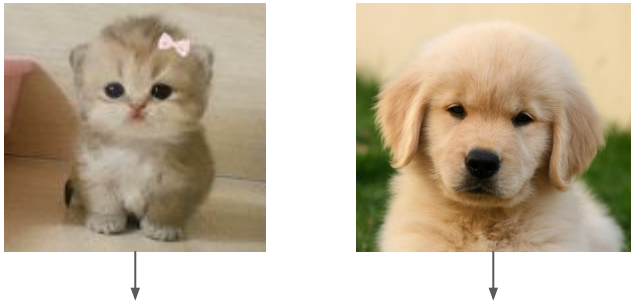
Whiteboard: logistic regression example

Machine Learning: challenges

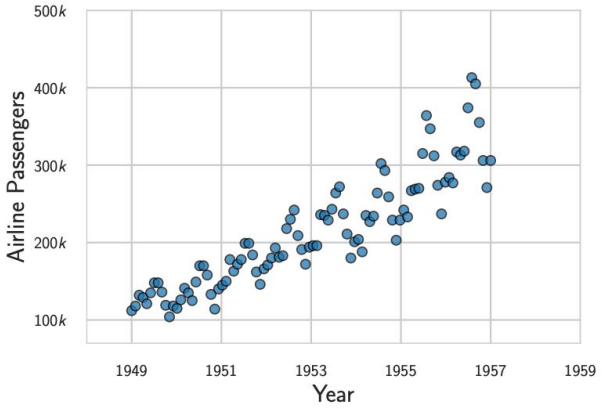
- Data: collect good training data
- Model: decide on the model class
- Training: find good parameters
- Generalization: make sure good train accuracy translates into good test predictions

Supervised learning

Classification: predict a discrete label

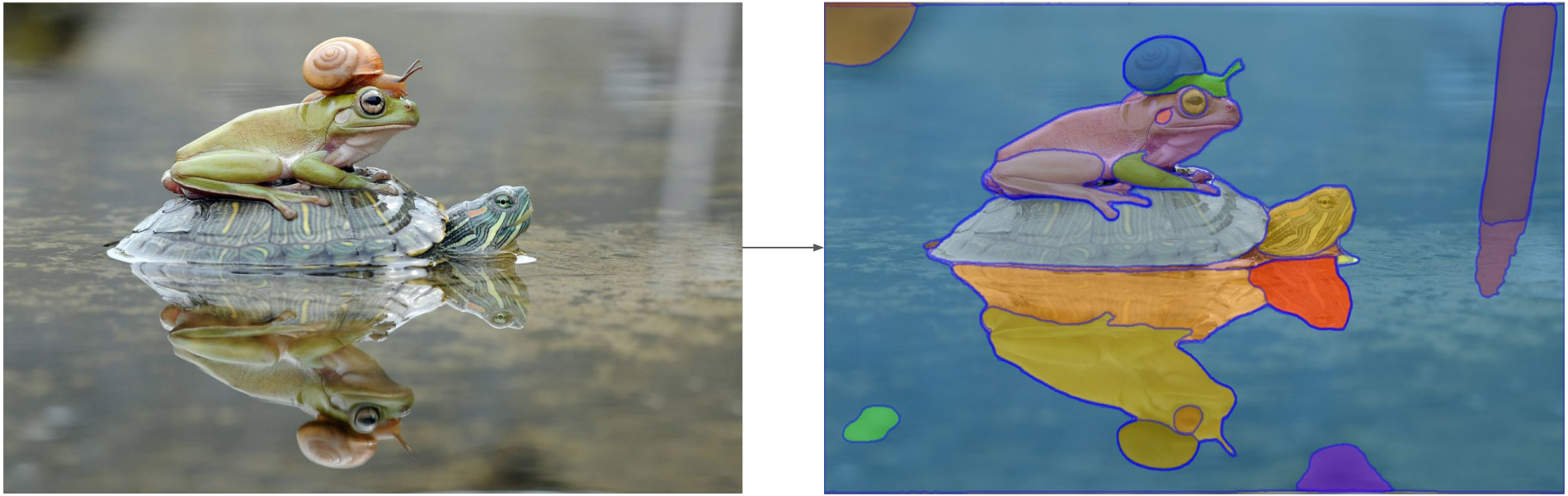


Regression: predict a continuous value



Supervised learning

Segmentation: identify locations of objects



https://segment-anything.com/demo

Other types of learning

- Unsupervised learning: no labels
- Self-supervised learning: also no labels
- Reinforcement learning: learn to act from interaction with environment
- …

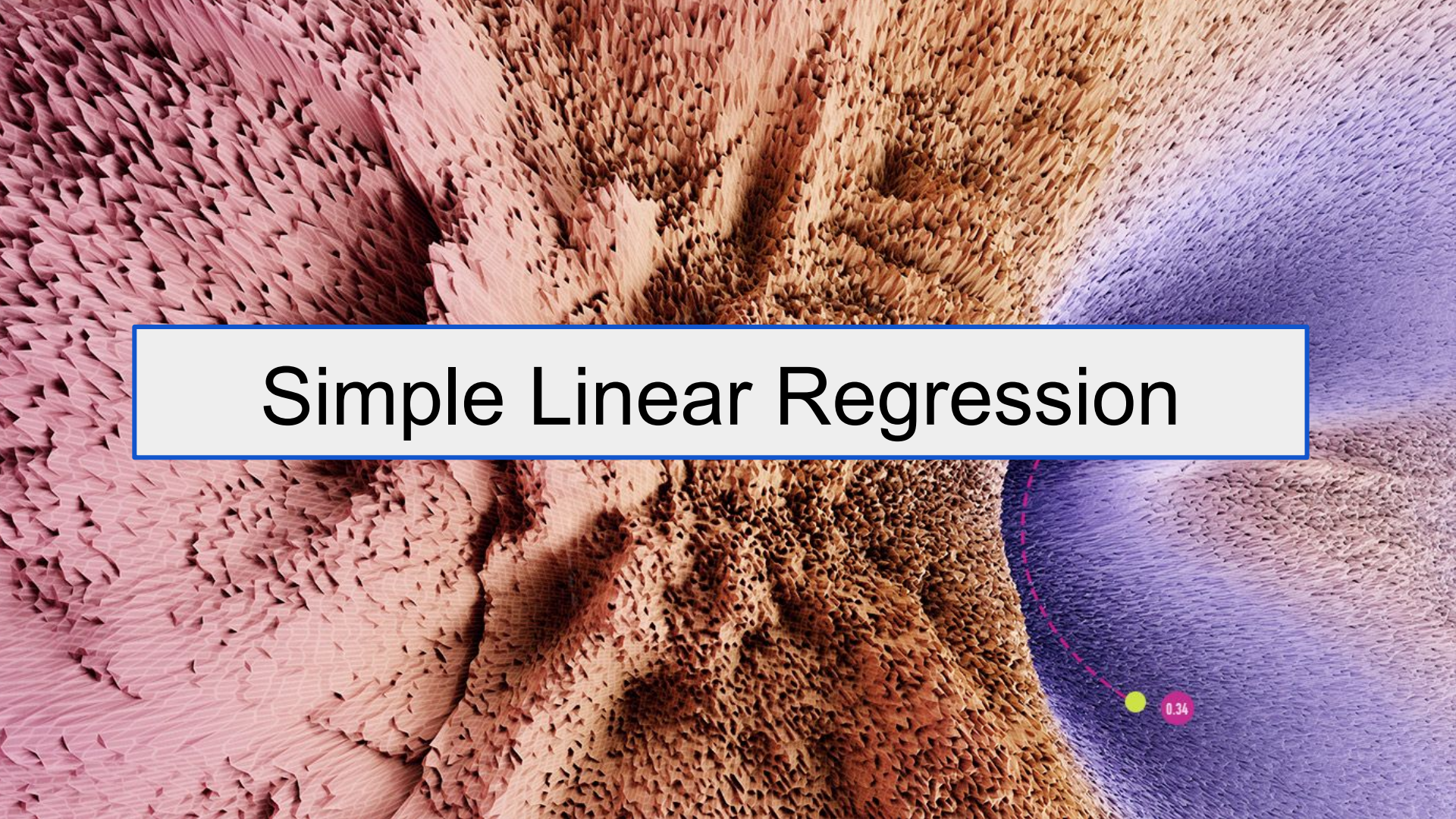
What is ChatGPT?

Other types of learning

- Unsupervised learning: no labels
- Self-supervised learning: also no labels
- Reinforcement learning: learn to act from interaction with environment
- …

What is ChatGPT?

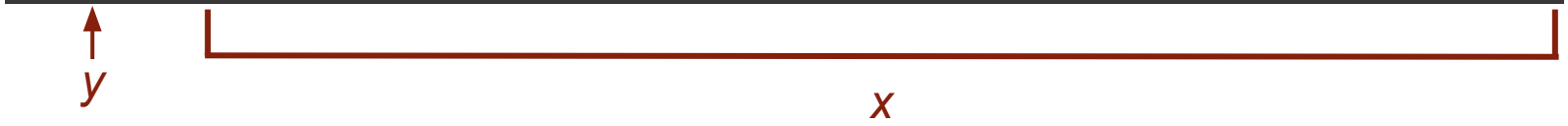
- Pre-training is supervised learning / classification
- Post-training is reinforcement learning (RLHF)


Simple Linear Regression
0.34

Simple linear regression

- Demo in: `demo_auto_mpg.ipynb`
- Predict highway miles per gallon of a car given engine characteristics

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino
5	15.0	8	429.0	198.0	4341.0	10.0	70	1	ford galaxie 500



Simple linear regression

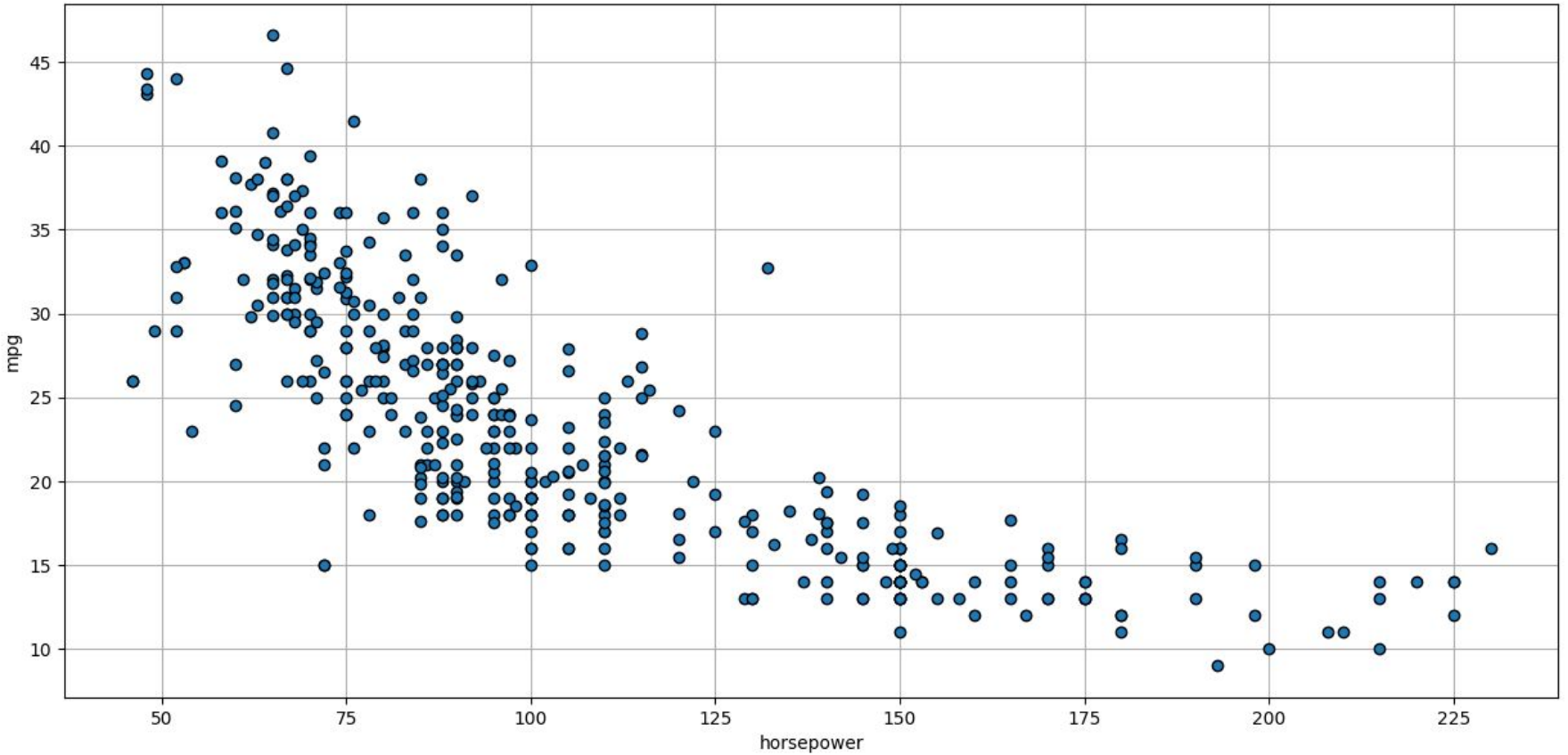
- Demo in: `pavel_demo_auto_mpg.ipynb`
- Predict highway miles per gallon of a car given engine characteristics

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino
5	15.0	8	429.0	198.0	4341.0	10.0	70	1	ford galaxie 500

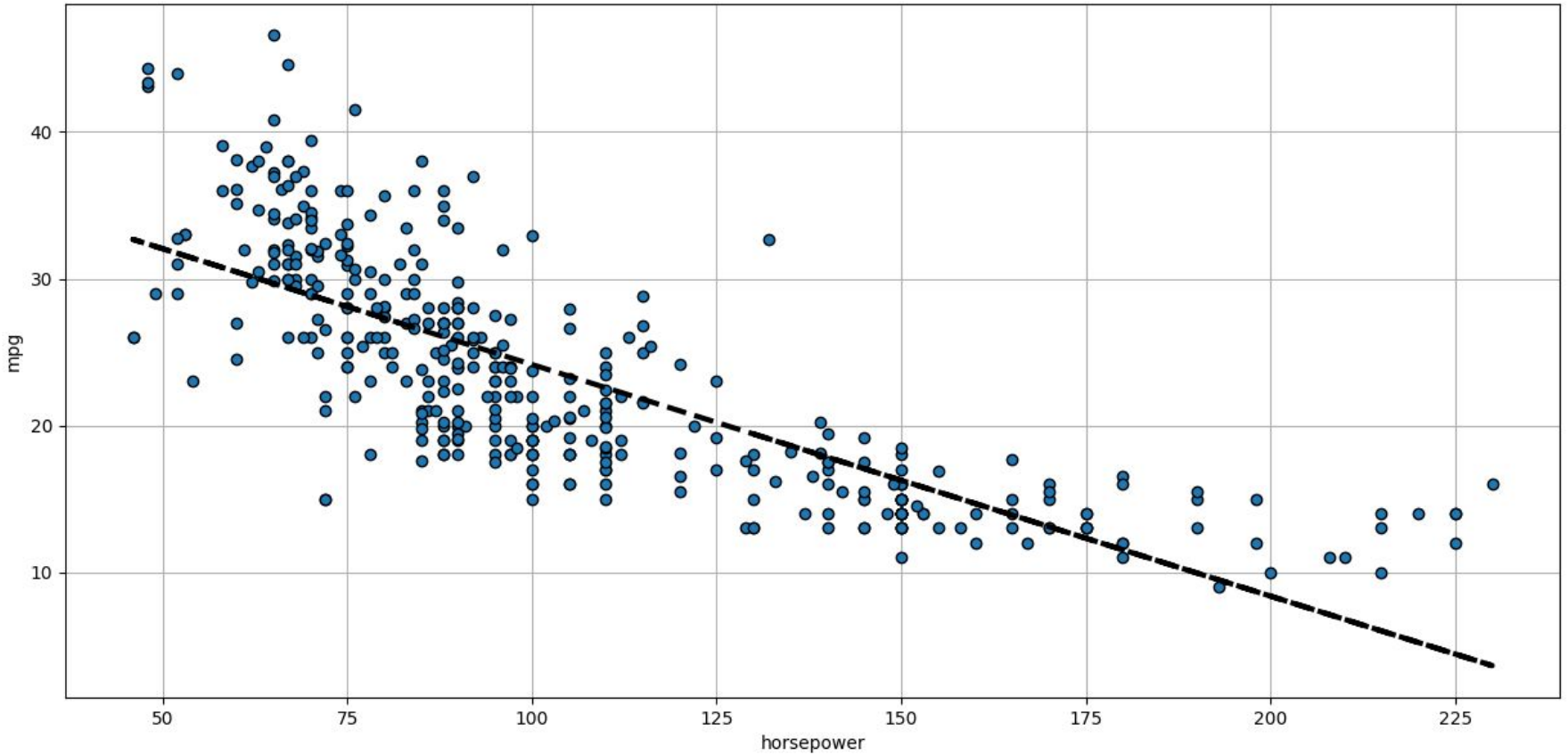
y

x

Simple linear regression



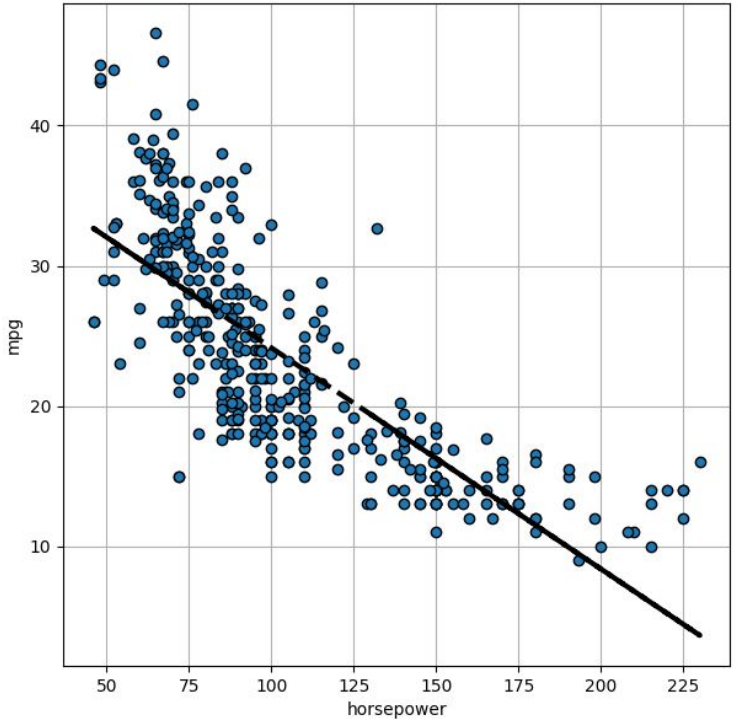
Simple linear regression



Simple linear regression

Whiteboard:

- *Dataset*
- *Model*
- *Model Parameters*
- *Loss Functions*
- *Training (ERM)*

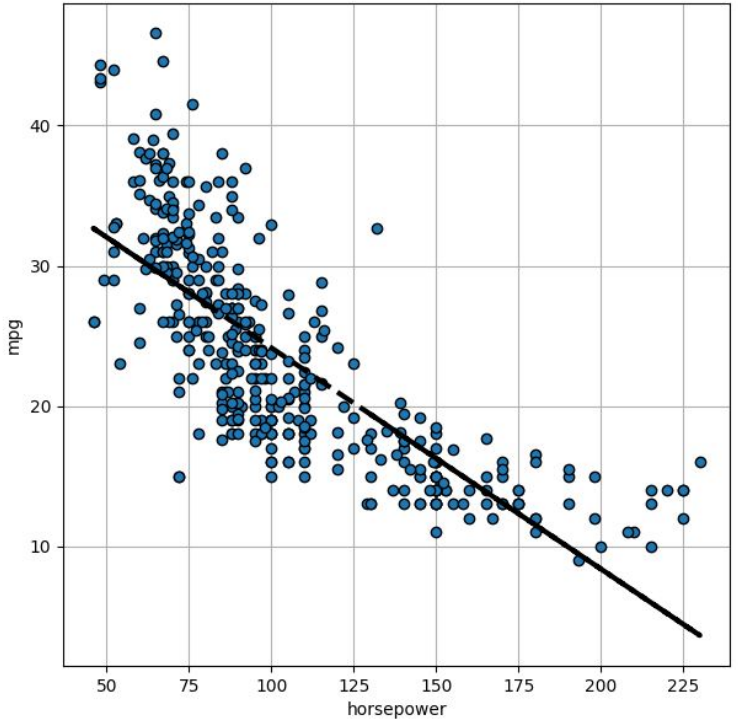


Simple linear regression

$$L(w,b) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - wx_i - b)^2$$

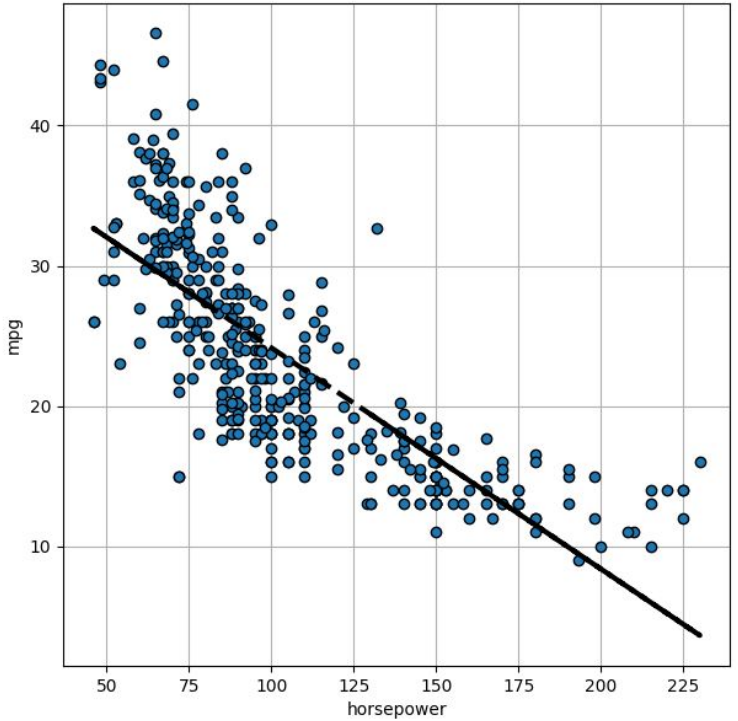
$$\hat{w} = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - \hat{w}\bar{x}$$



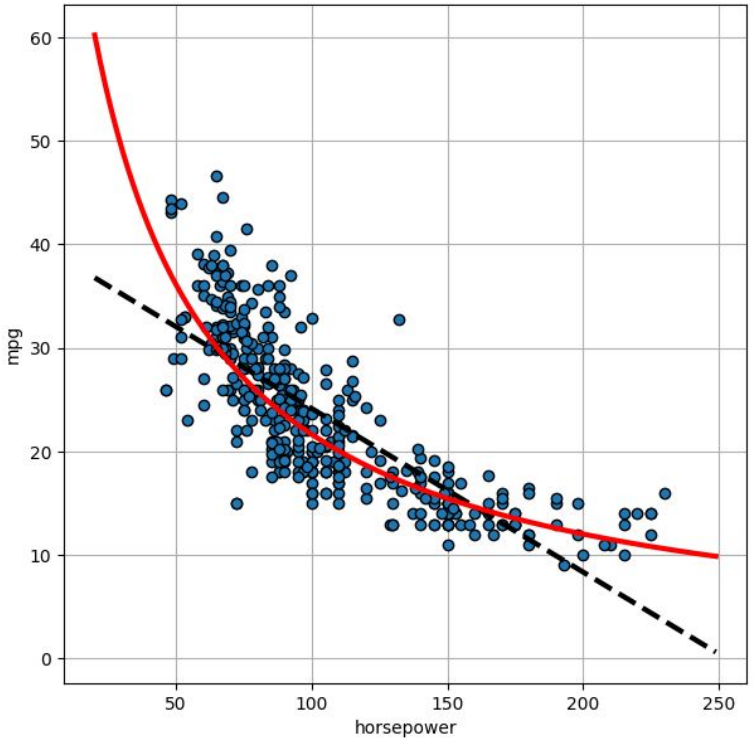
Simple linear regression: comments

- In general we cannot always find optimal parameters analytically
- Data can be non-linear



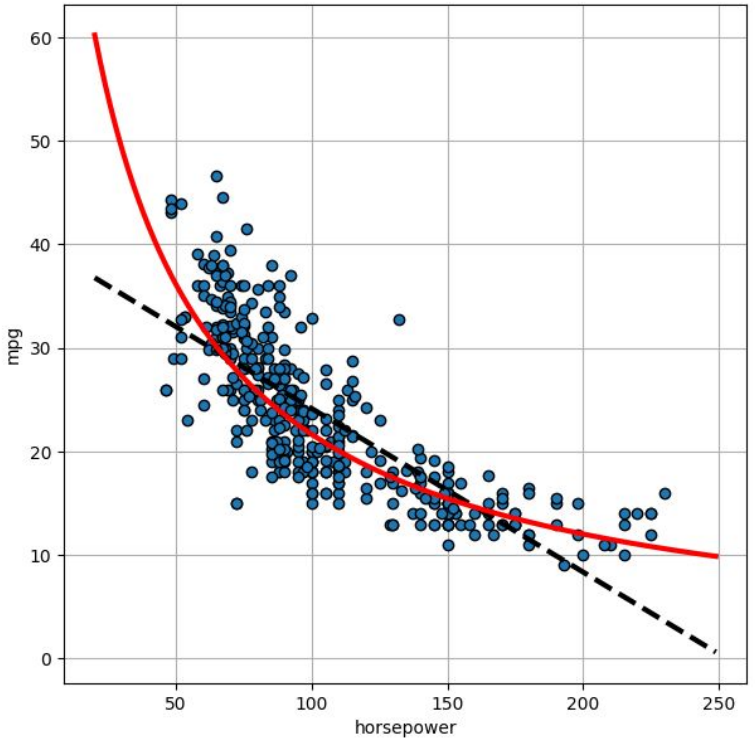
Simple linear regression: comments

- In general we cannot always find optimal parameters analytically
- Data can be non-linear
  - Can preprocess x and y, e.g. y'=1/y



Simple linear regression: comments

- In general we cannot always find optimal parameters analytically
- Data can be non-linear
  - Can preprocess x and y, e.g. y'=1/y
- You may need more than one feature
  - Think about MNIST

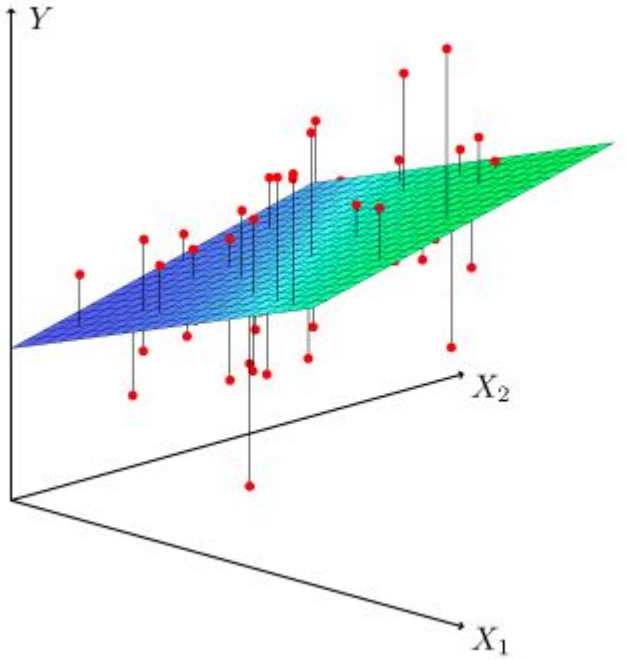


Linear Regression With Multiple Features

General linear regression

Whiteboard:

- *Dataset*
- *Model*
- *Model Parameters*
- *Loss Functions*
- *Training (ERM)*



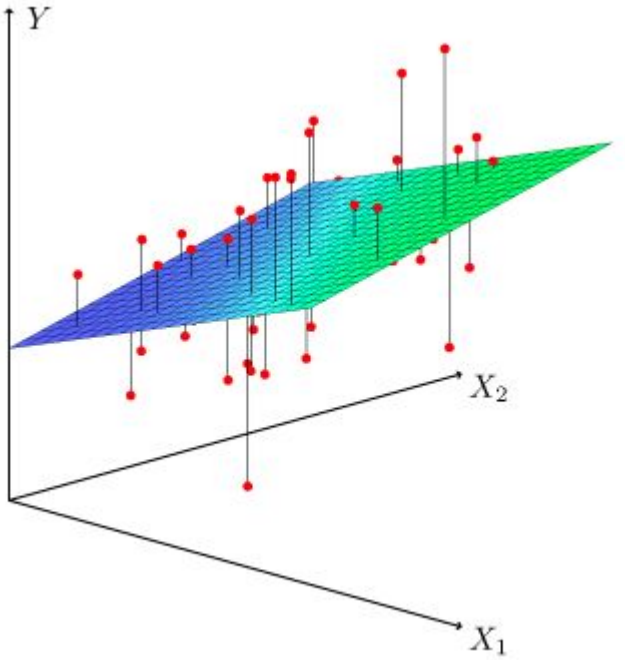
General linear regression

$$\hat{w} = (X^T X)^{-1}(X^T Y)$$

Compare to the formulas we got for the simple linear regression:

$$\hat{w} = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - \hat{w}\bar{x}$$



General linear regression

$$\hat{w} = (X^T X)^{-1}(X^T Y)$$

Compare to the formulas we got for the simple linear regression:

$$\hat{w} = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - \hat{w}\bar{x}$$

General linear regression

```
import pandas as pd
from sklearn.linear_model import LinearRegression

y_col = "mpg"
x_cols = ["cylinders", "displacement", "horsepower",
"weight", "acceleration", "model year"]
df_selected = df[[y_col] + x_cols]
df_selected = df_selected.dropna()

reg = LinearRegression()
reg.fit(X=df_selected[x_cols], y=df_selected[y_col])
print(reg.coef_)
```

General linear regression

```
# Create DataFrame with Ferrari 308 GTB/GTS data
data = {
    'mpg': [17],  # Combined estimate
    'cylinders': [8],
    'displacement': [177],
    'horsepower': [240],
    'weight': [3050],  # pounds, mid-range estimate
    'acceleration': [7.0],  # 0-60 mph in seconds
    'model year': [78]  # Representative year
}

ferrari df = pd.DataFrame(data)
reg.predict(ferrari_df[x_cols])
```

Predicts 22!

General linear regression

Whiteboard:

- *Categorical features*
- *Transformations of features*