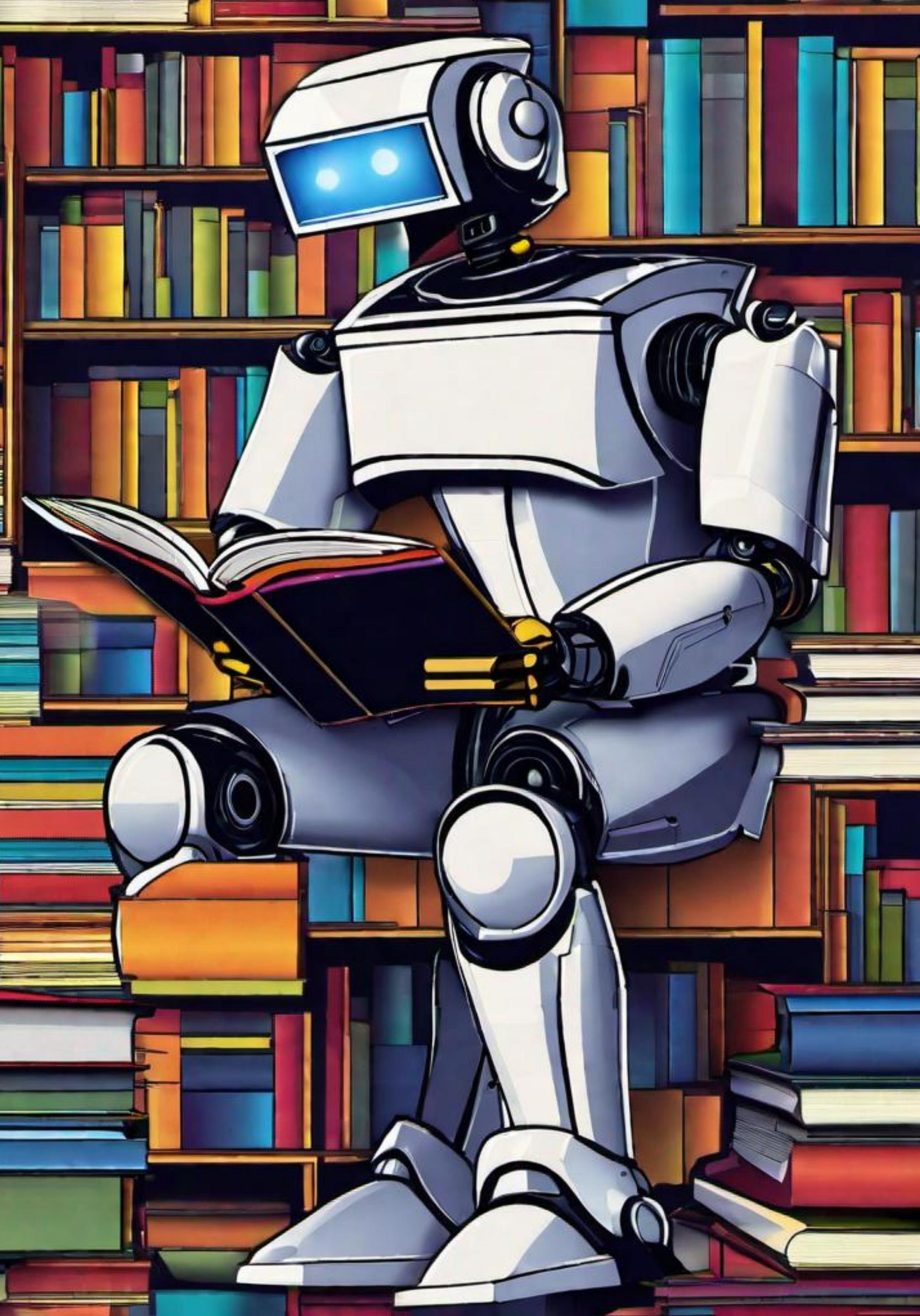


# What the hell is Machine Learning

And how **can anyone**  
get involved in it?



le wagon



# Since 2014

## Le Wagon

Has been providing candidates with the skillset necessary to future-proof their career.



le wagon

A teaching of excellence

Most voted

#1 ranked bootcamp worldwide



Official Partners

International campuses

22,000+

People educated at Le Wagon

40

Cities around the world

Focused on practice & product

2300+

Products build

100+

Startups launched

# Tech is just a tool.

Tools are used for building



The grid displays 45 different tech tools, each with a small profile picture below it:

- Row 1: NEEDL, BRETEL, MISSION BIBERON, MAKIWARS, LA RÉSERVE, SHARKRANK
- Row 2: BOOMBOX, IMPACT, KRAWD, WOOM, KWAALA, FOODKICK
- Row 3: KUDOZ, ROADSTR, OPEN LOGE, MEDPICS, FIVEMARKS, MEDIAPRONOS
- Row 4: MA SHARE ÉCOLE, LOVELY HOOD, FIREBNN, FRSHST, EXPLORERS, MONMECANICIEN.FR

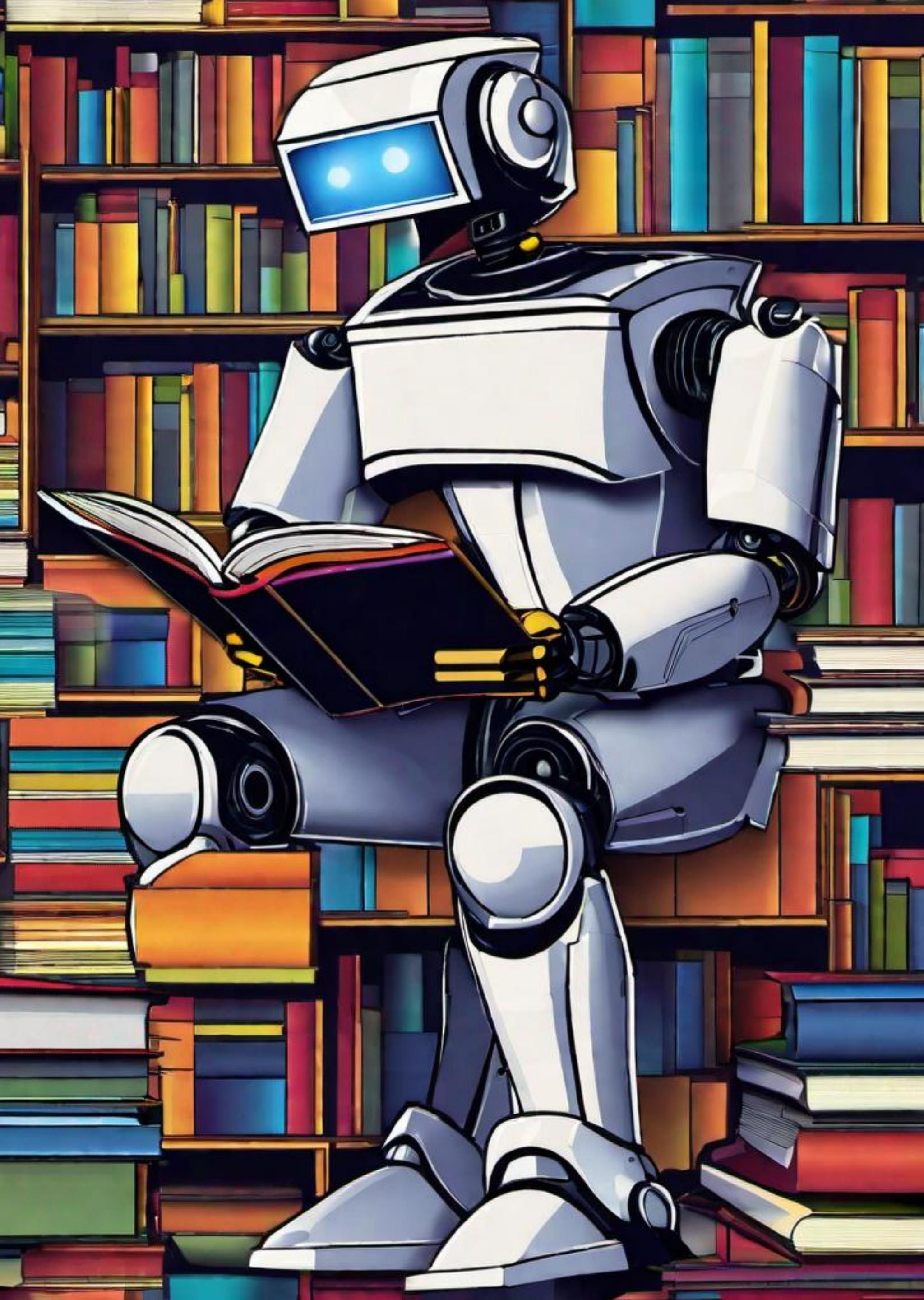


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

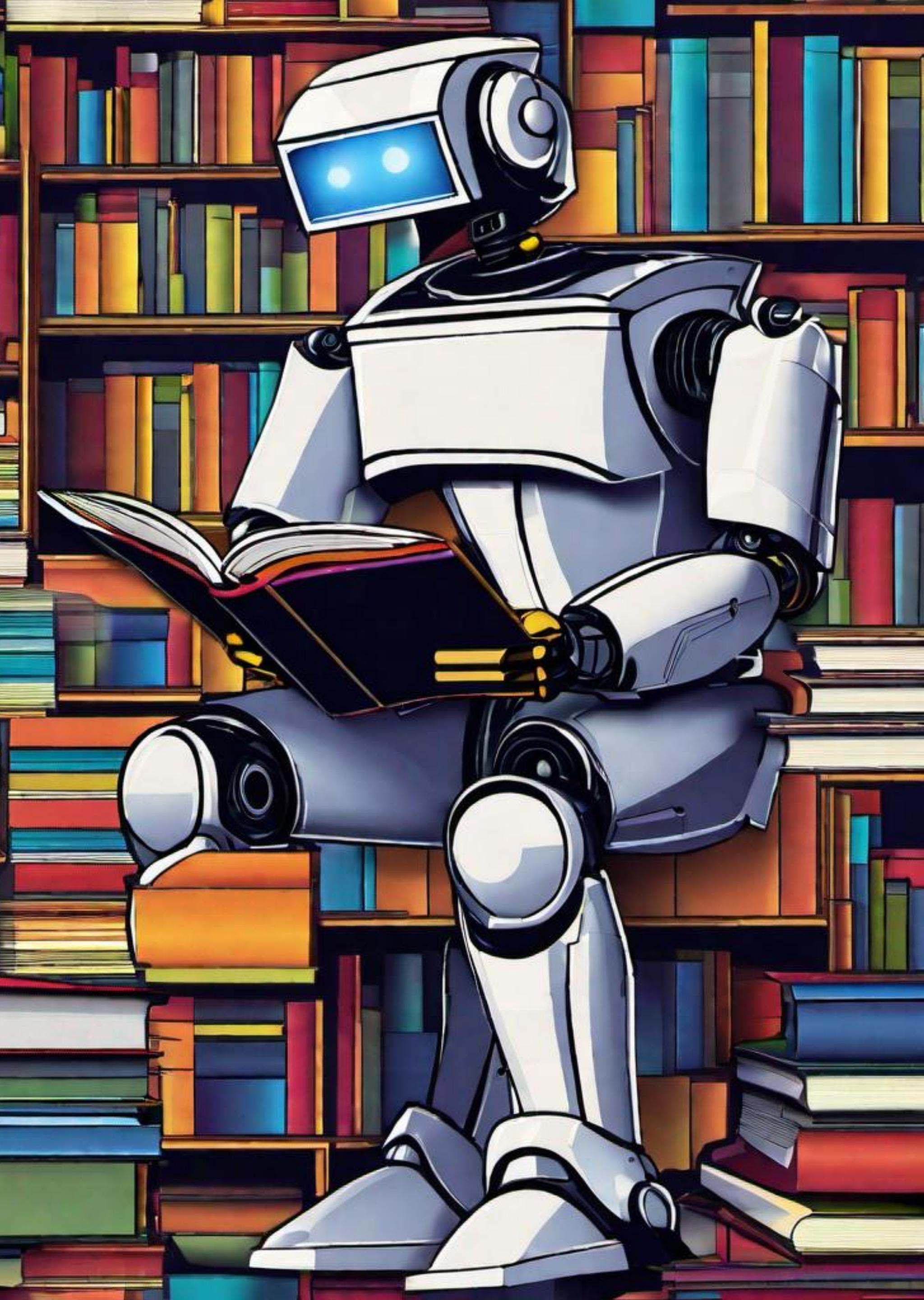


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



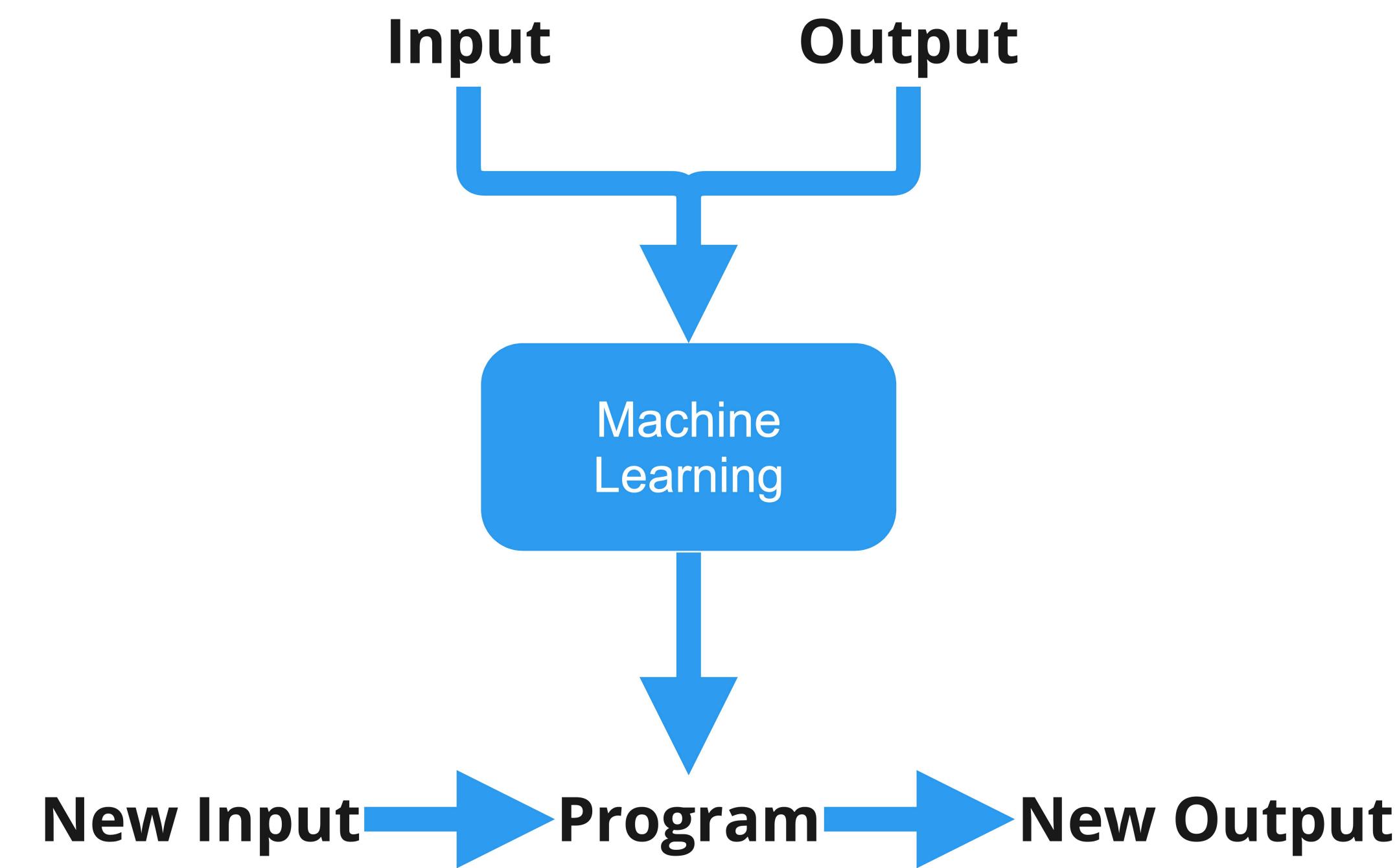
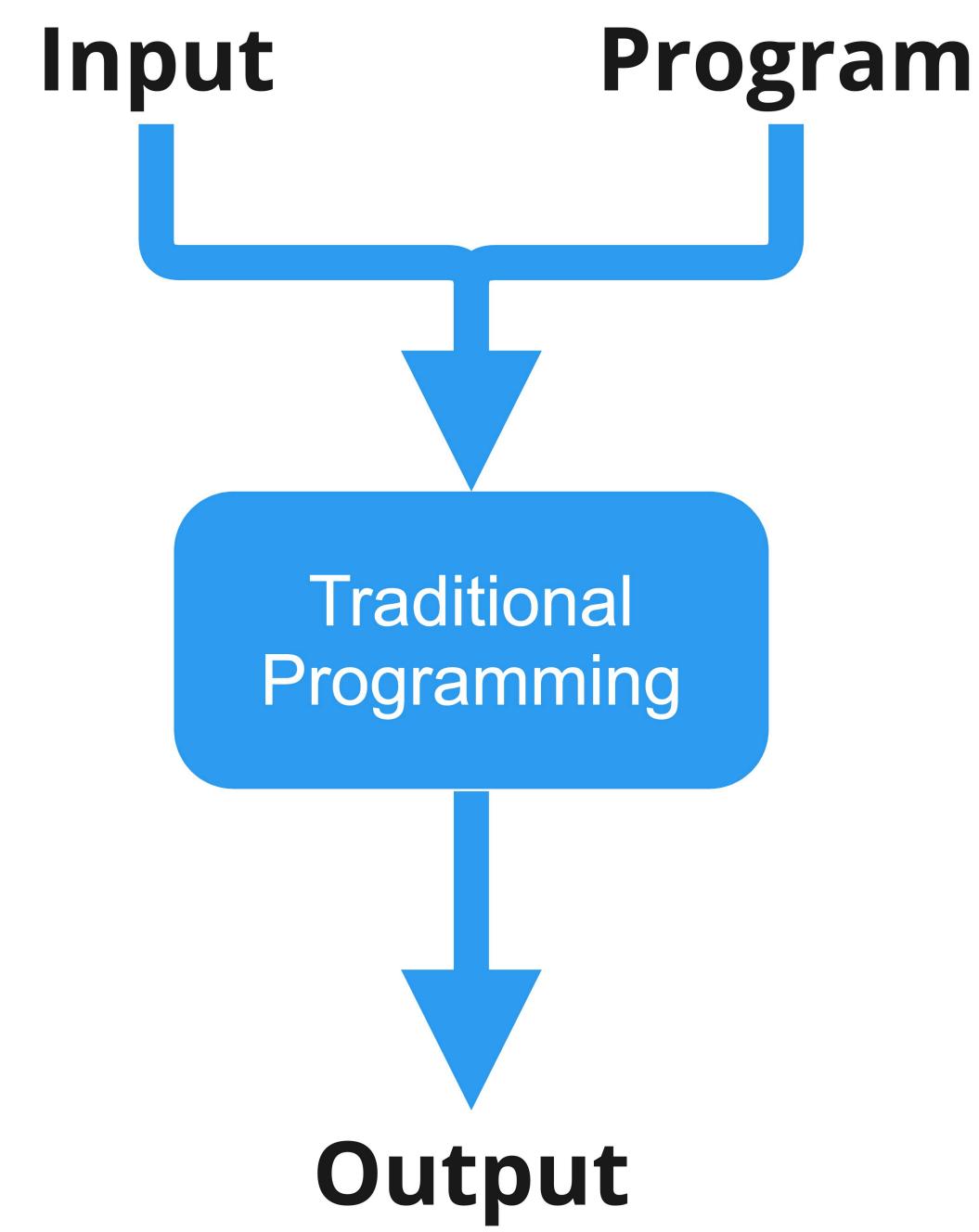
le wagon



“[a] field of study that gives computers the **ability to learn without being explicitly programmed”**

*Arthur Samuel (1959)*

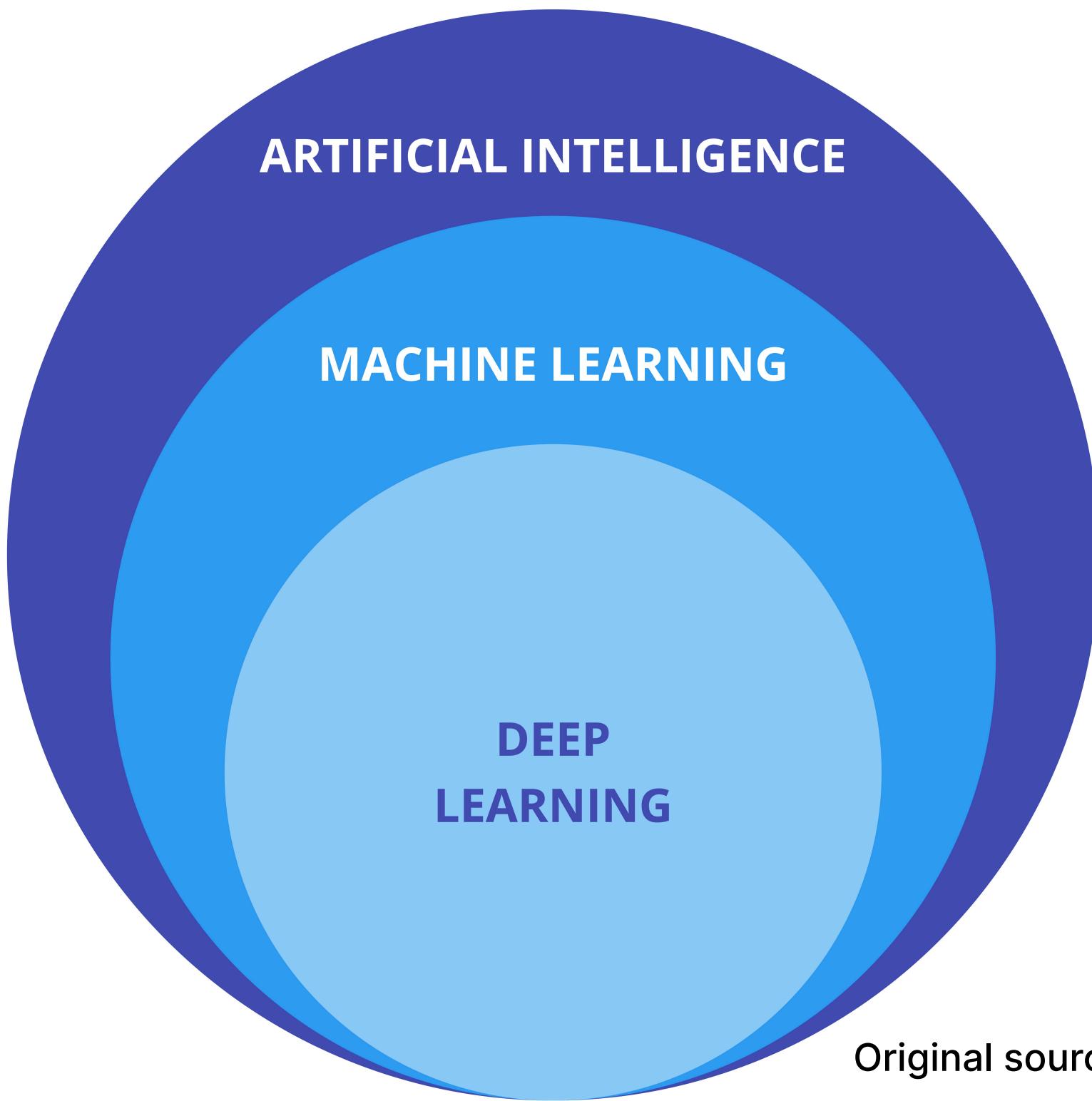




Original source: [ScienceDirect](#)



# What about terms like AI or Deep Learning?



Original source: [Intel](#)



[...] computational methods that use experience to [recognize complex patterns] and [create formulas for] **accurate predictions**

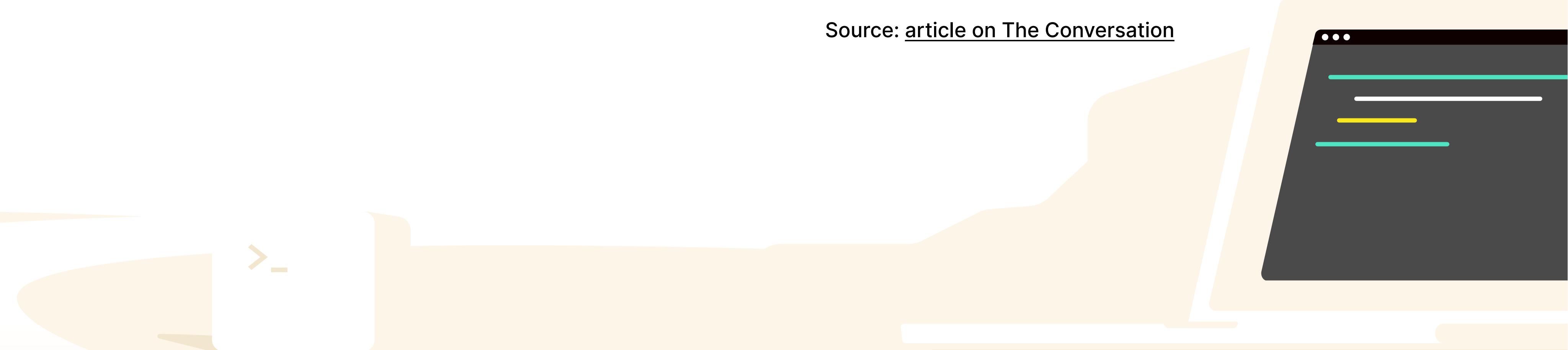
Source: [article on The Conversation](#)



**data**

[...] computational methods that use ~~experience~~ to [recognize complex patterns] and [create formulas for] **accurate predictions**

Source: [article on The Conversation](#)

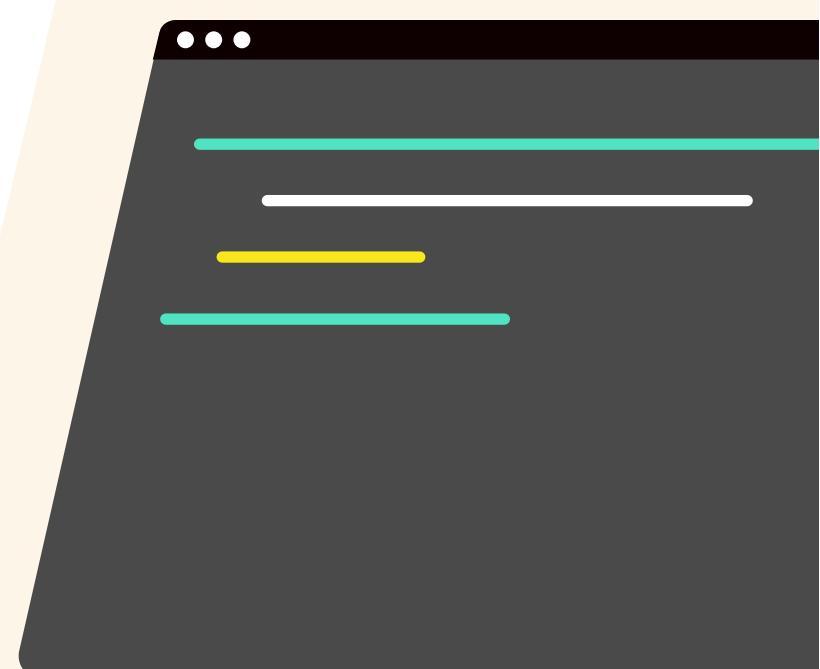


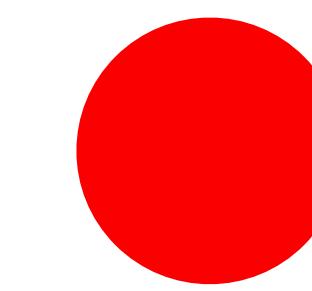
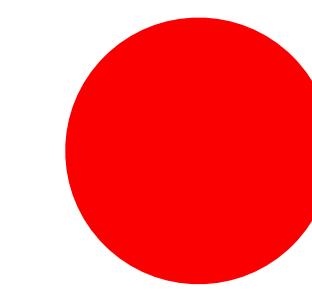
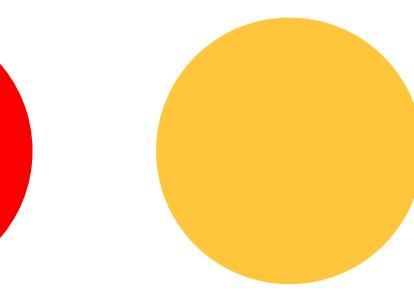
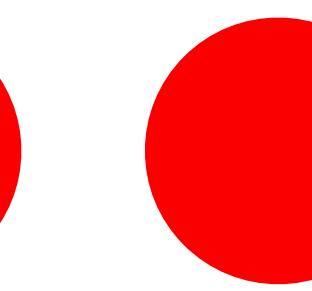
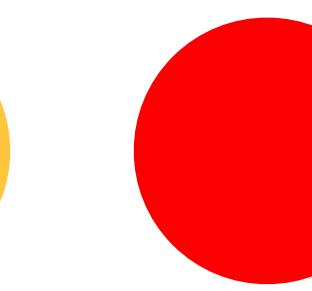
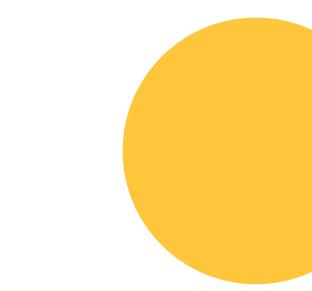
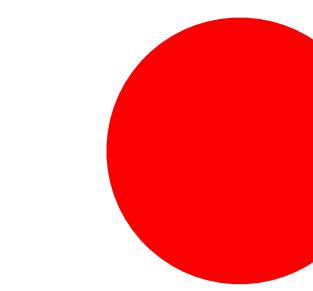
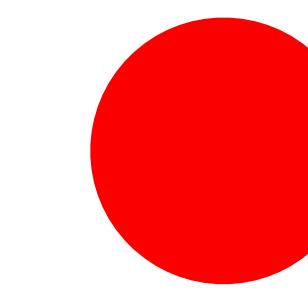
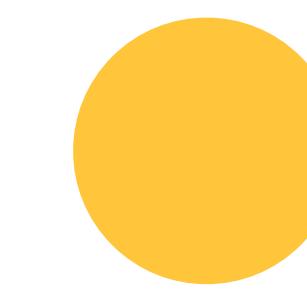
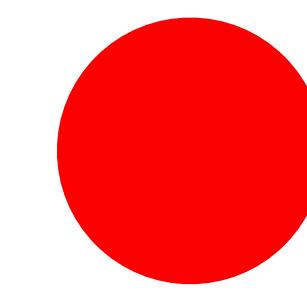
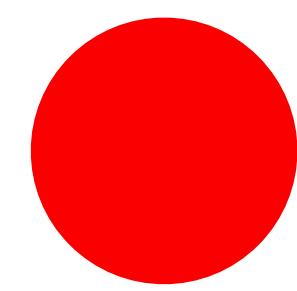
**data**

[...] computational methods that use ~~experience~~ to [recognize complex patterns] and [create ~~formulas~~ for] **accurate predictions**

**models**

Source: [article on The Conversation](#)

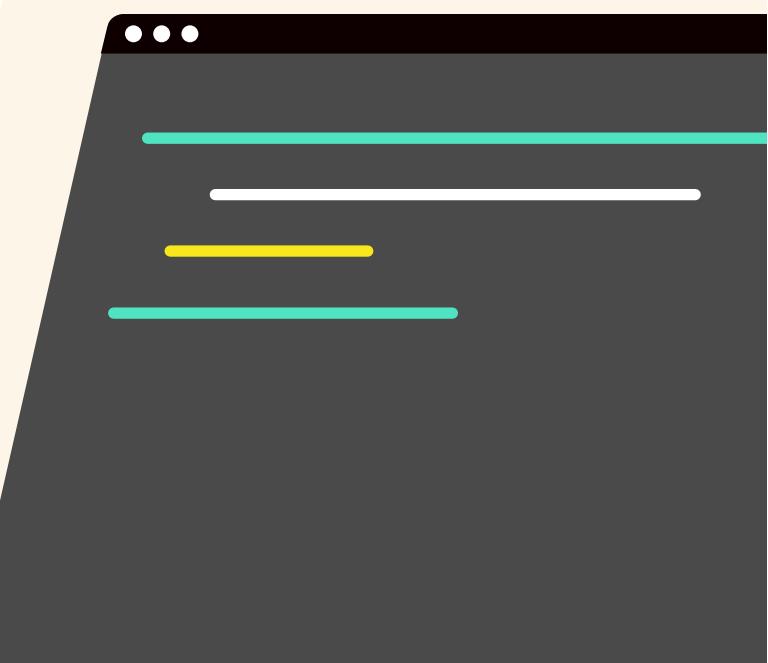




?



>





# How does Machine Learning work?



# Take basic observations

My Age/Height Ratio

Age	Height (cm)
5	111
6	115
7	119
8	122
9	126
10	130
11	133
12	137
13	145
14	147
15	153

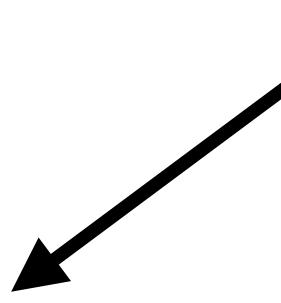
# Take basic observations

My Age/Height Ratio

Age	Height (cm)
-----	-------------

5	111
6	115
7	119
8	122
9	126
10	130
11	133
12	137
13	145
14	147
15	153

Input



# Take basic observations

My Age/Height Ratio	
Age	Height (cm)
5	111
6	115
7	119
8	122
9	126
10	130
11	133
12	137
13	145
14	147
15	153

Input

Output



# Take basic observations

My Age/Height Ratio		
	Age	Height (cm)
Feature	5	111
Input	6	115
	7	119
	8	122
	9	126
	10	130
	11	133
	12	137
	13	145
	14	147
Output	15	153



# Take basic observations

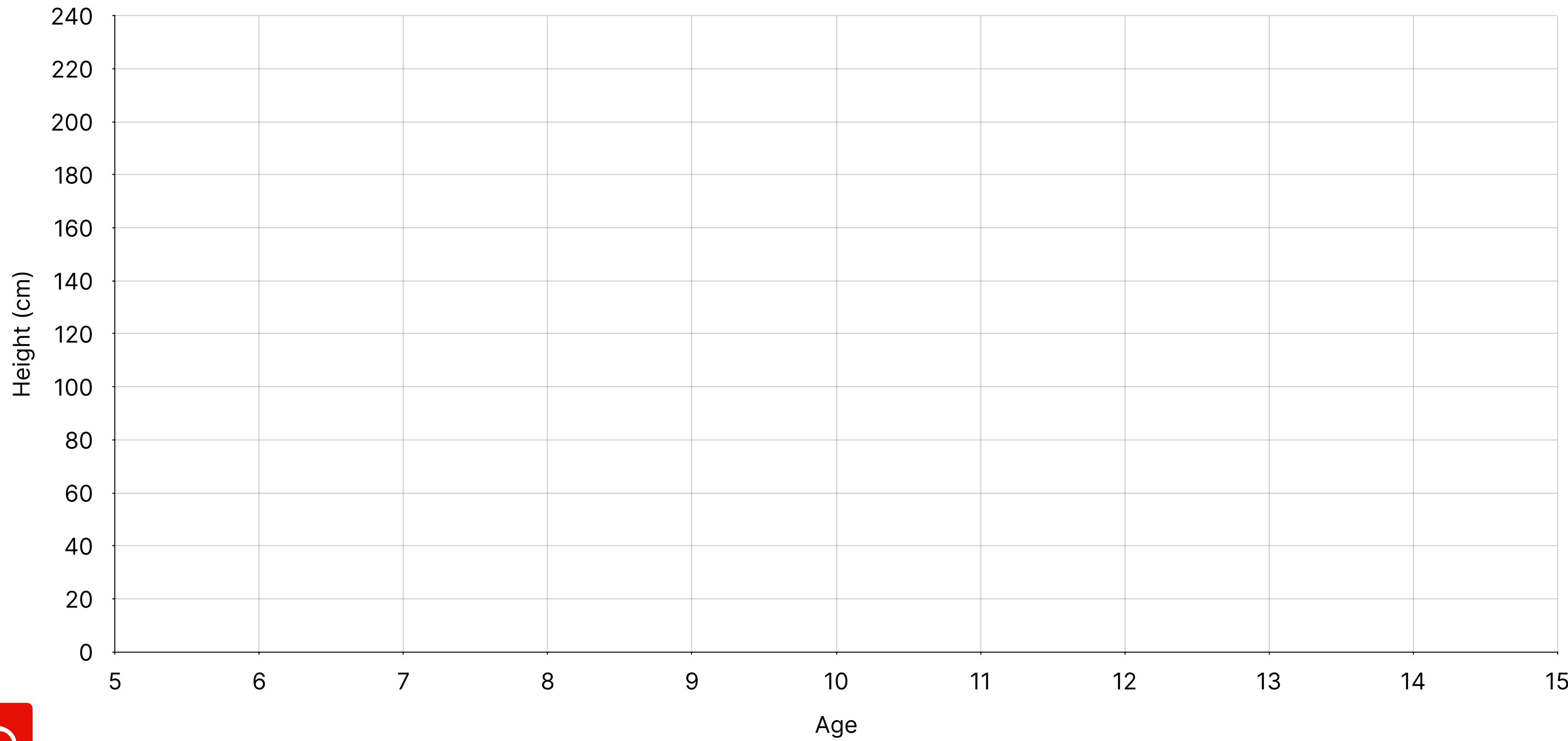
My Age/Height Ratio		
	Age	Height (cm)
Feature	5	111
Input	6	115
	7	119
	8	122
	9	126
	10	130
	11	133
	12	137
	13	145
	14	147
	15	153

Target

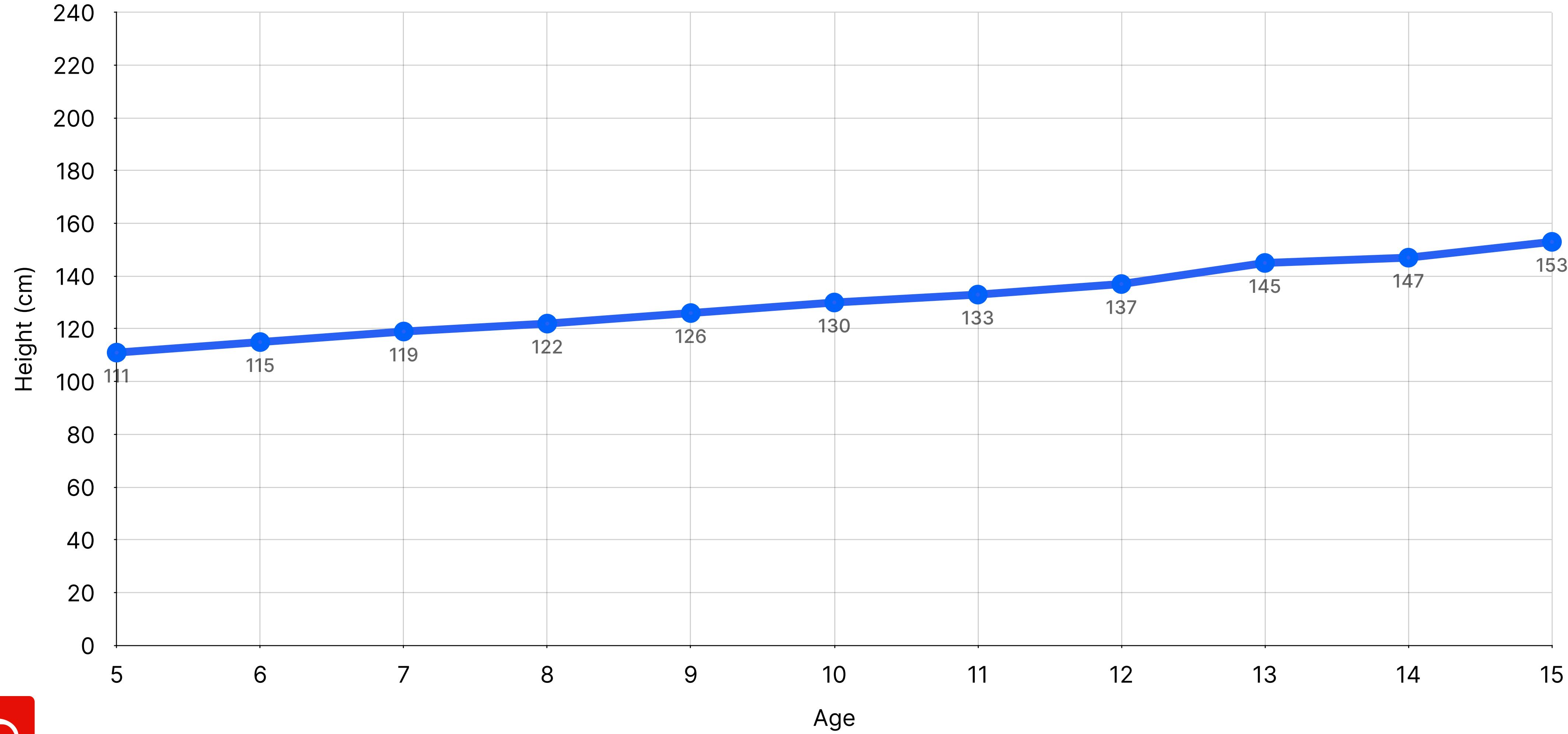
Output



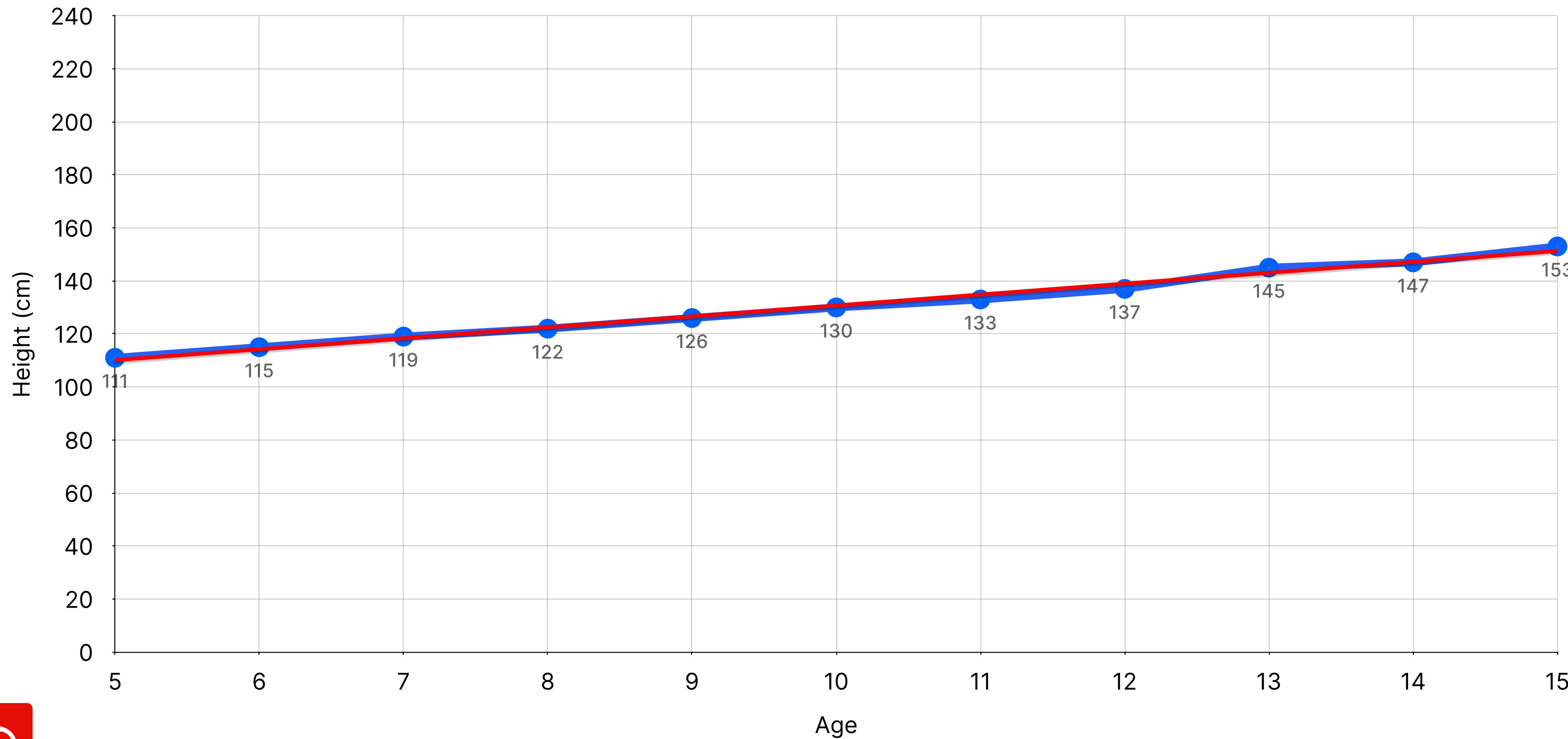
# Infer a mathematical function



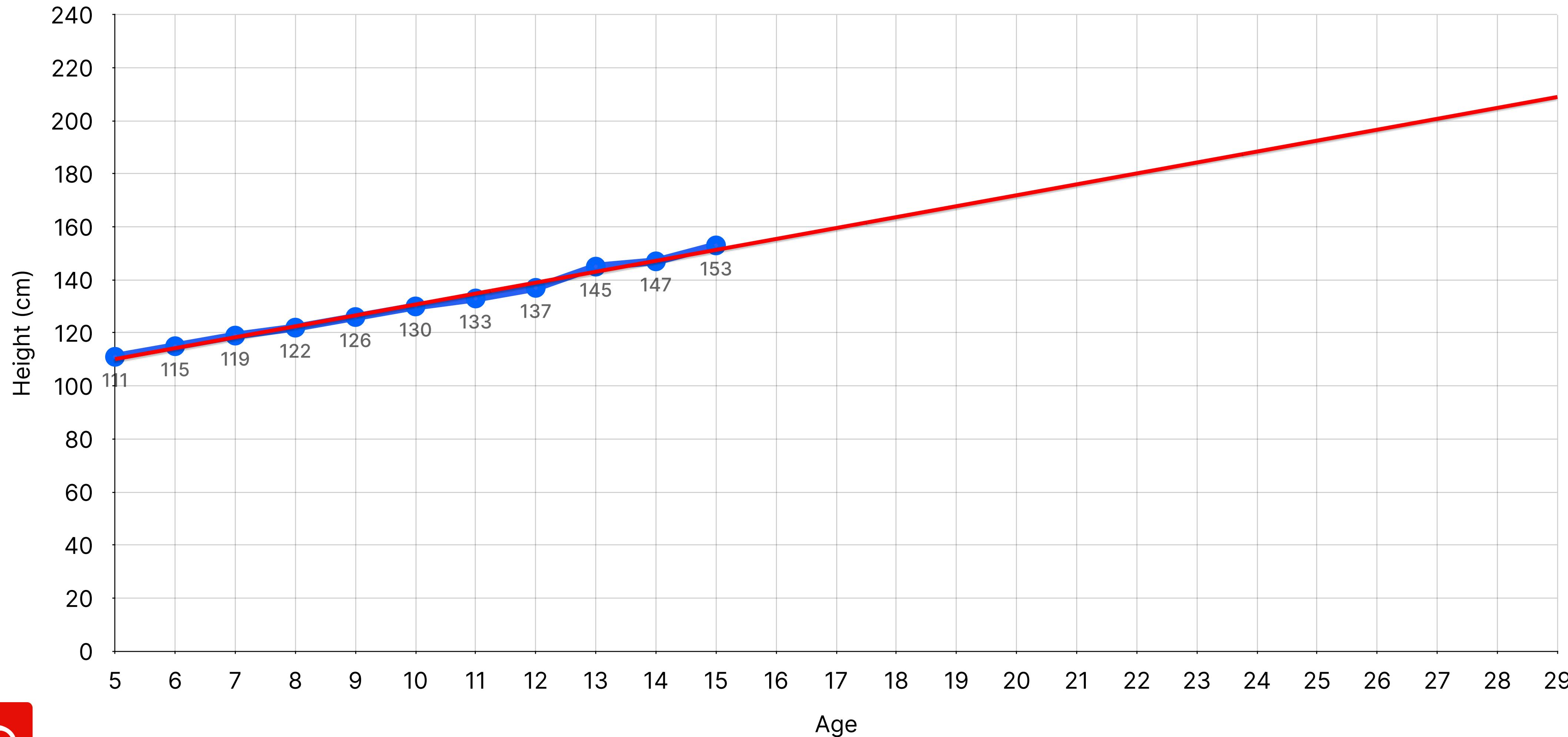
# Infer a mathematical function



# Infer a mathematical function



# Extrapolate new observations



**What if there were 150 different  
features and 1 Million observations  
for 2 Million different people?**

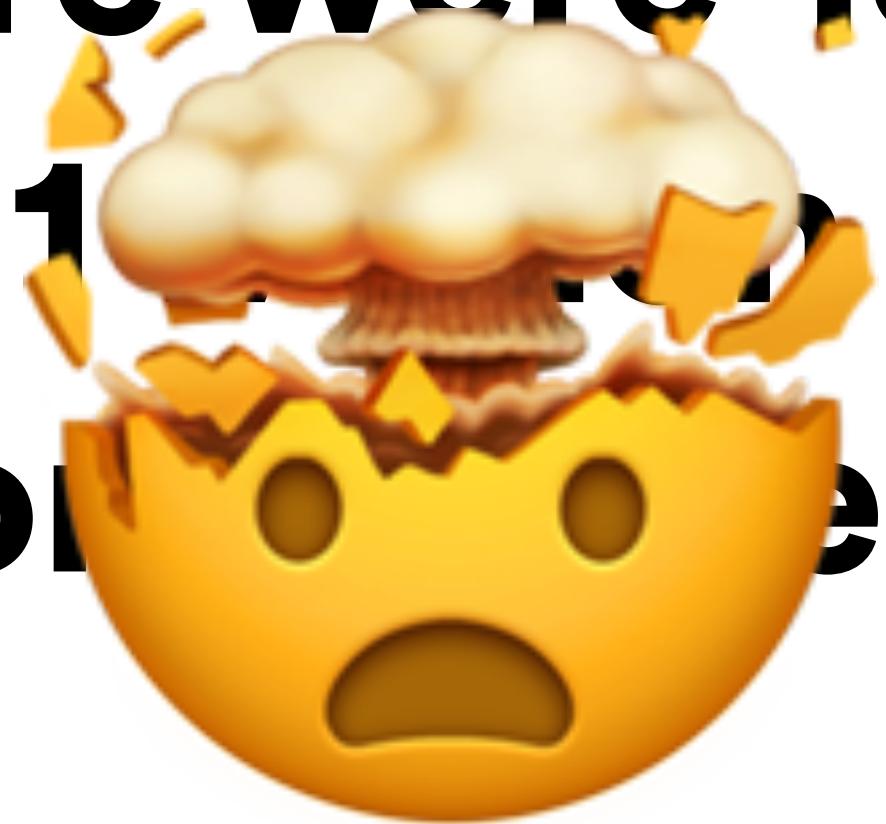


**What if there were 150 different  
features and 1 Million observations  
for 2 Million different people?**

**Every single day!**



**What if there were 150 different  
features and 1,000,000 observations  
for 2 Million different people?**



**Every single day!**



# Machine Learning is already all around us



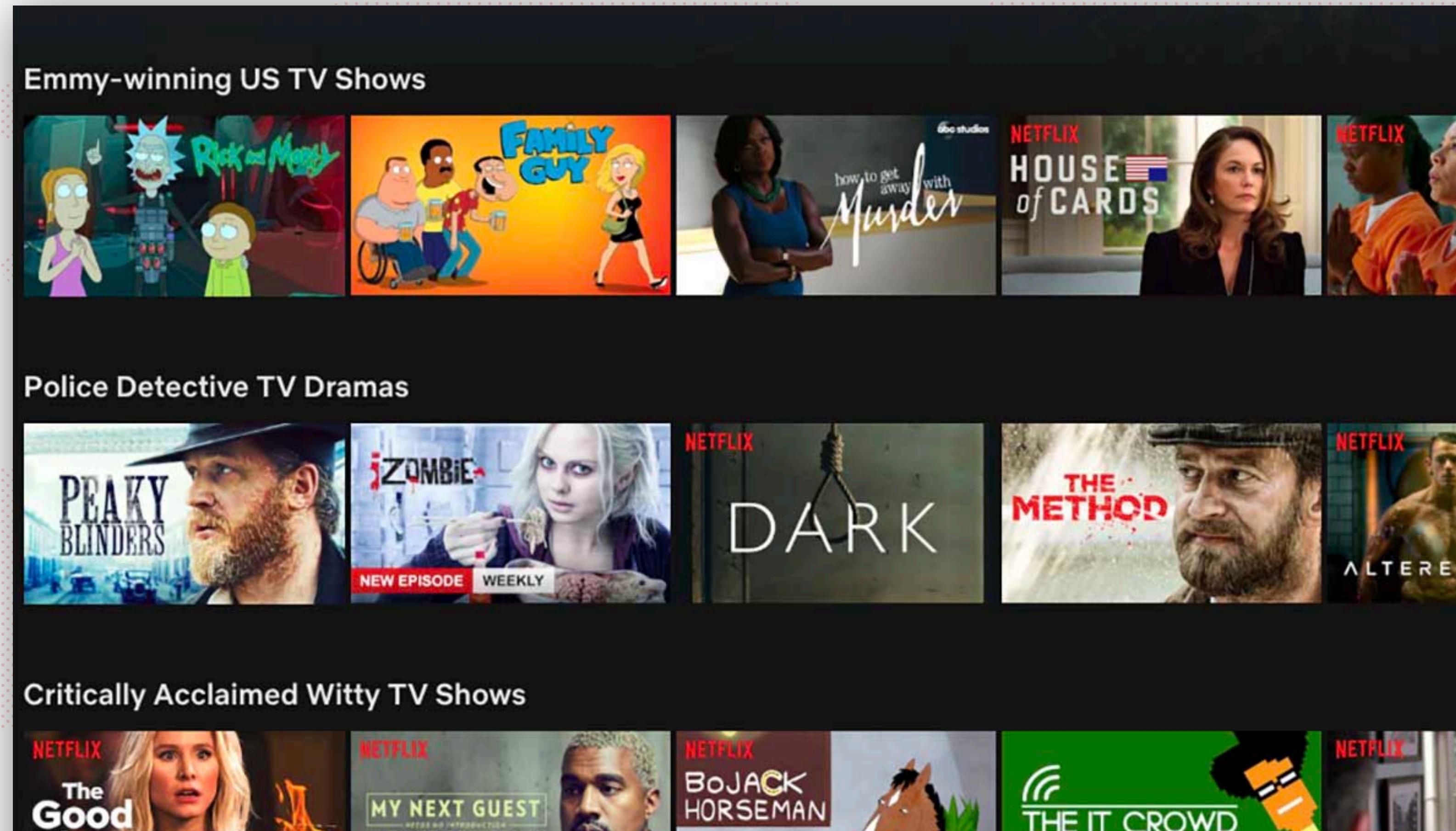
# Computer Vision

## Classifying and detecting visual data



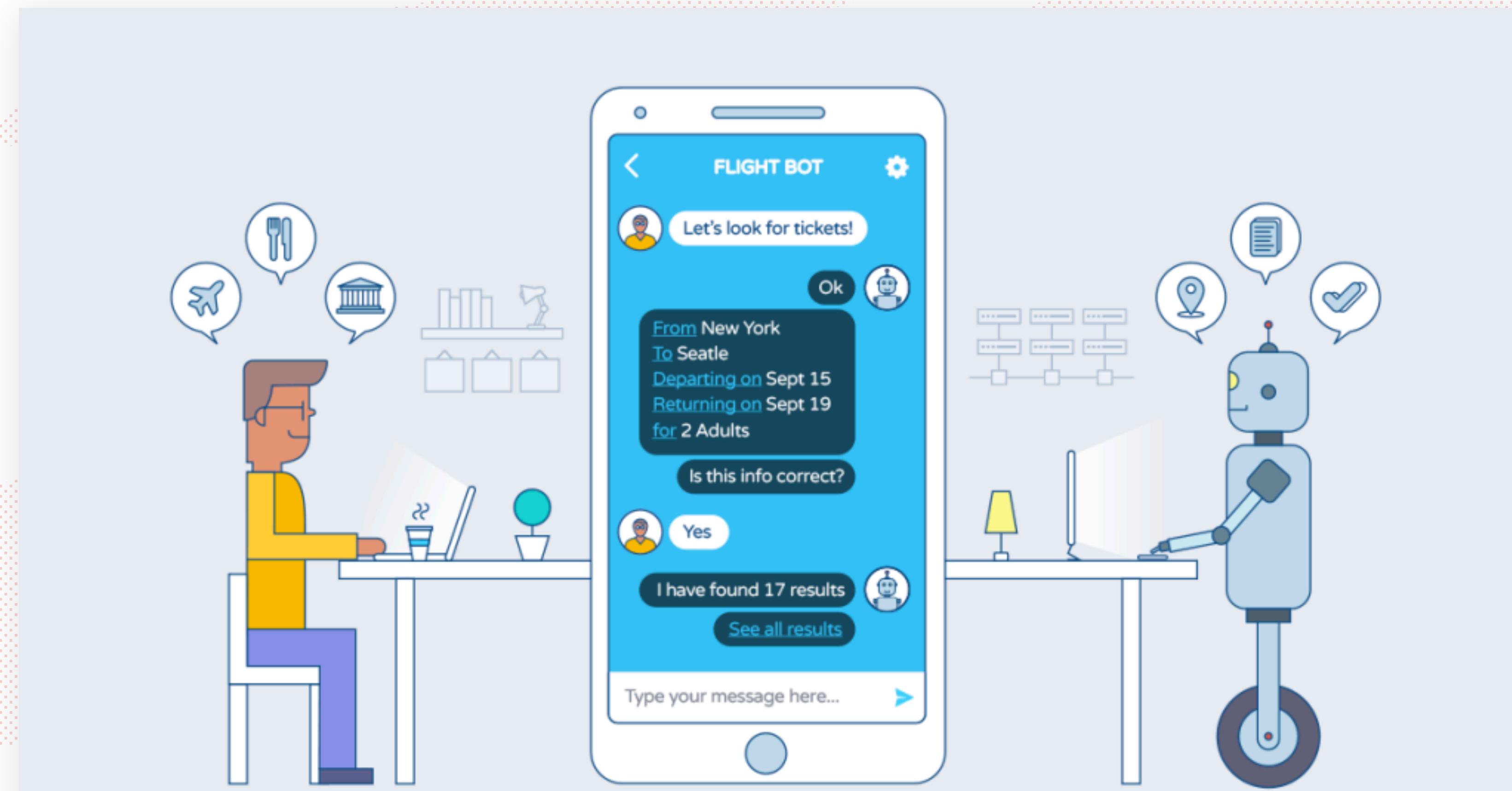
# Recommendation Engine

## Predicting your next action



# Natural Language Processing

## Finding meaning in text



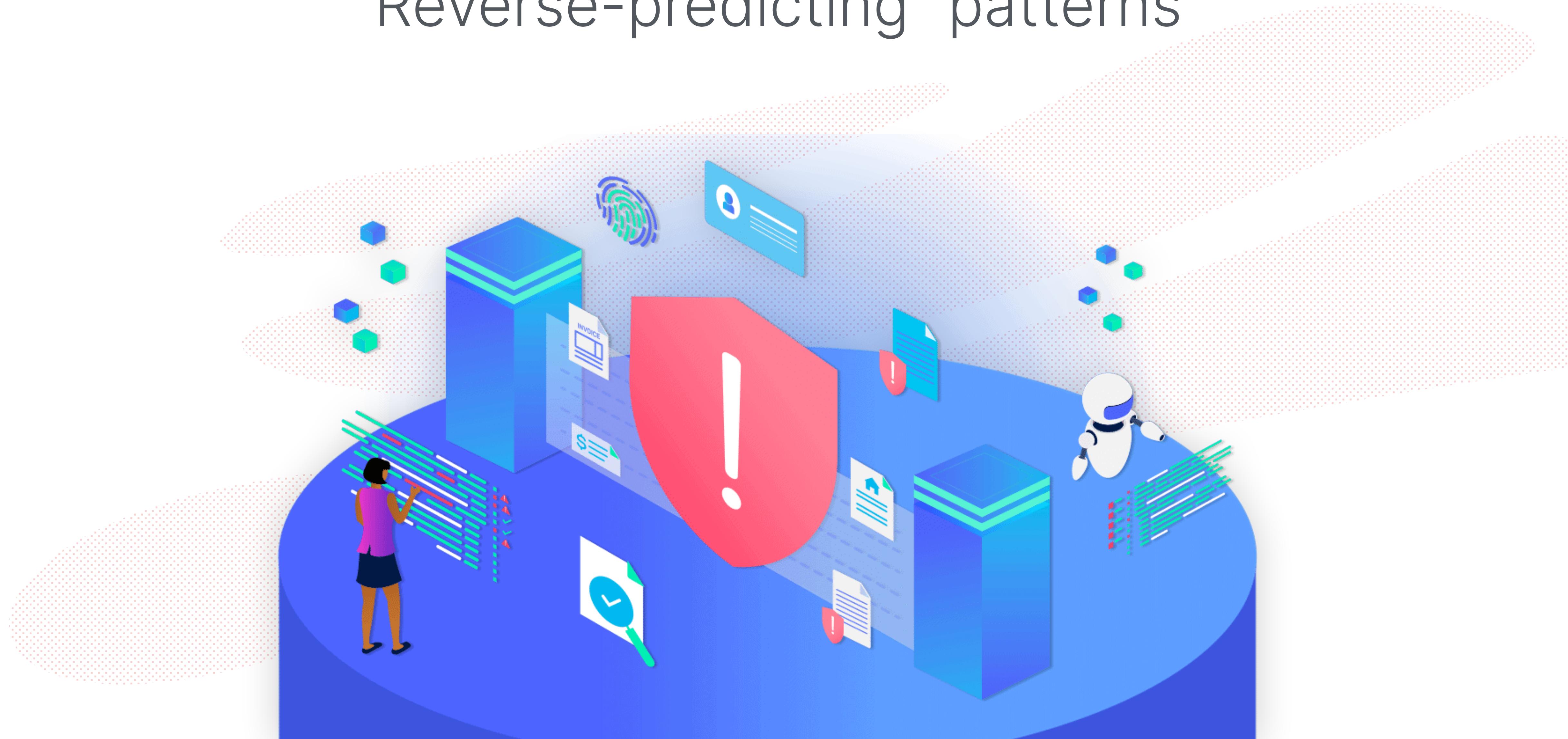
# Time Series Analysis

Can past changes predict future changes?

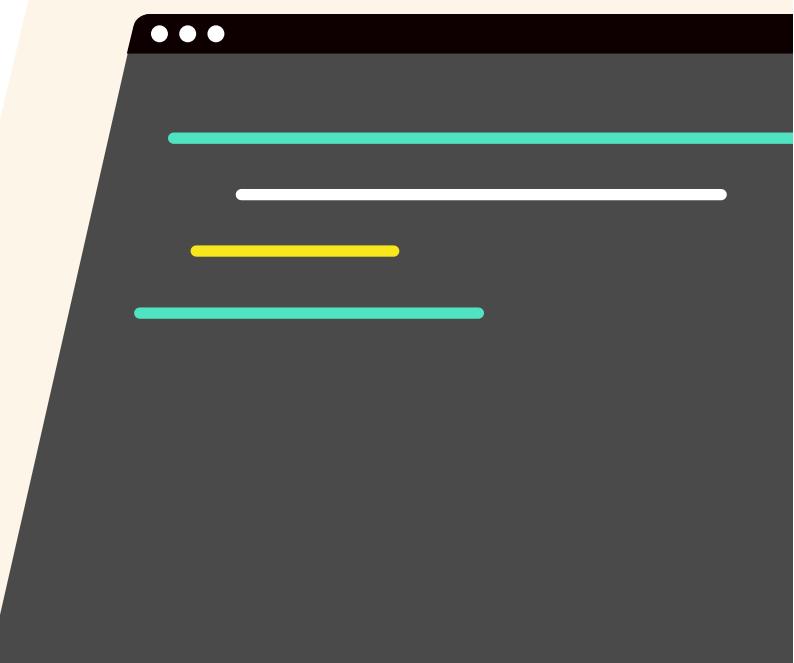


# Anomaly Detection

“Reverse-predicting” patterns



# When Machine Learning becomes just **hype**

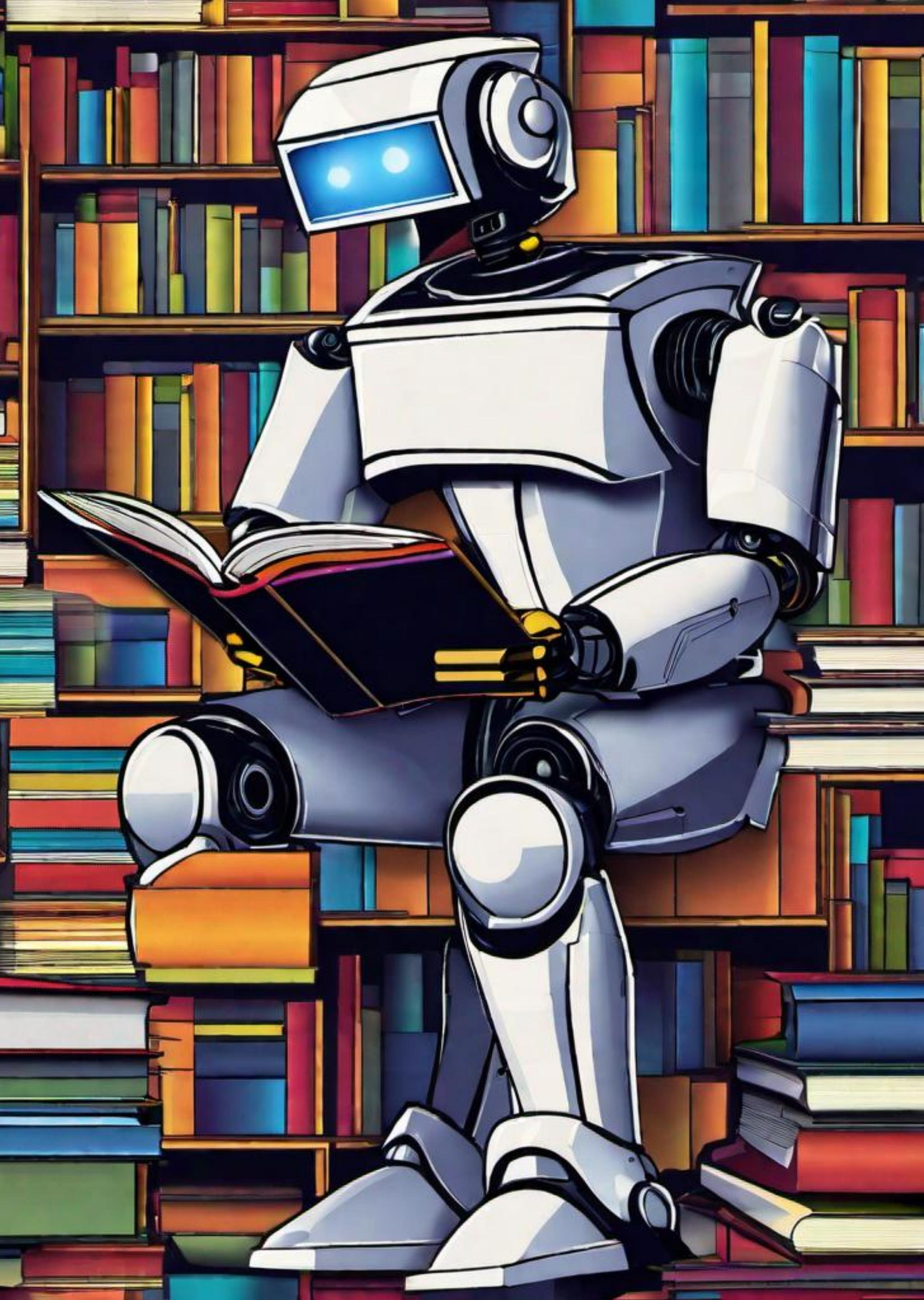


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

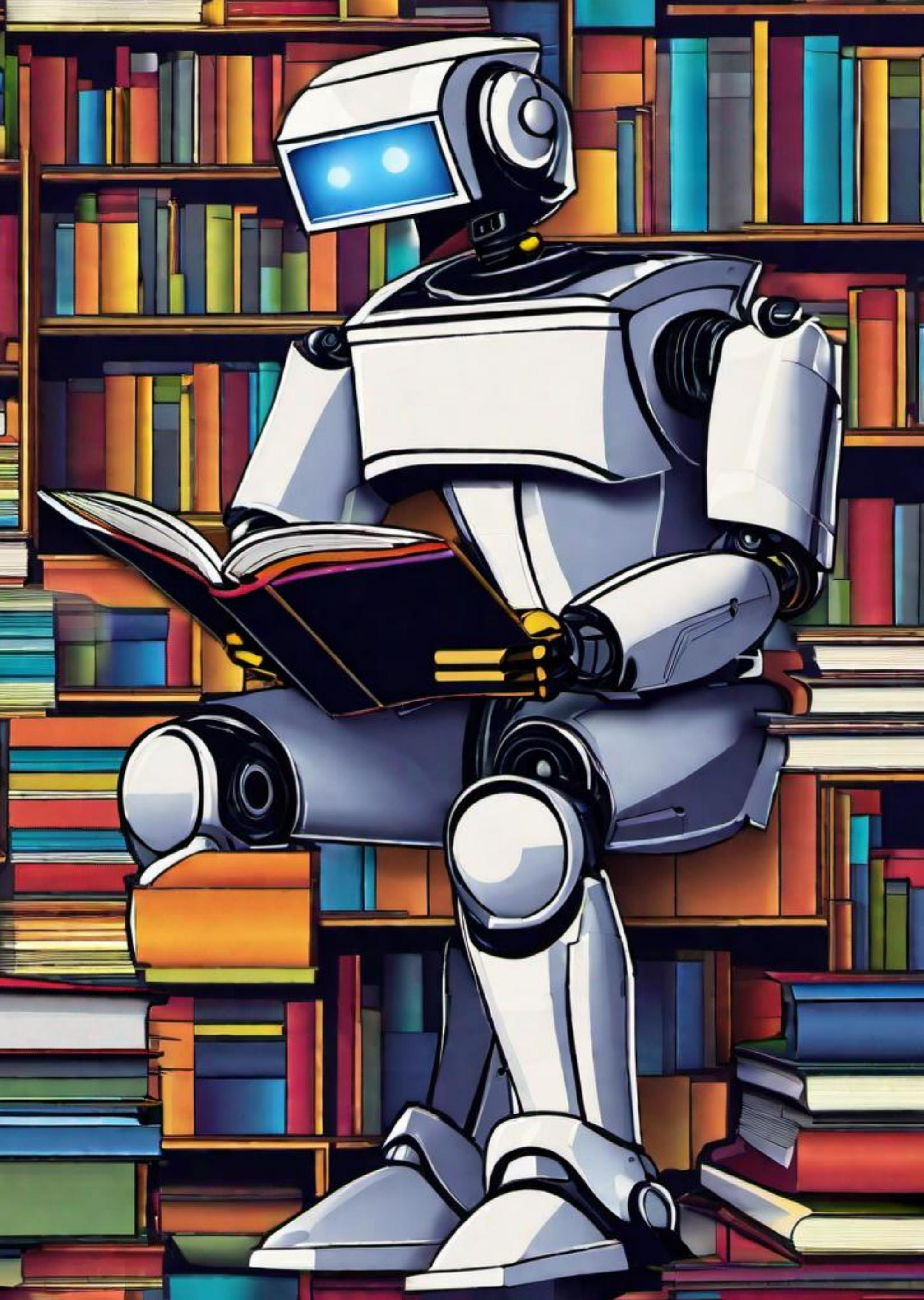


# Agenda

- ✓ What is Machine Learning
- ✓ What is **NOT** Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon



# **Analytics or Machine Learning?**



# **Analytics** or Machine Learning?

**Needs ML**

**Does **not** need ML**



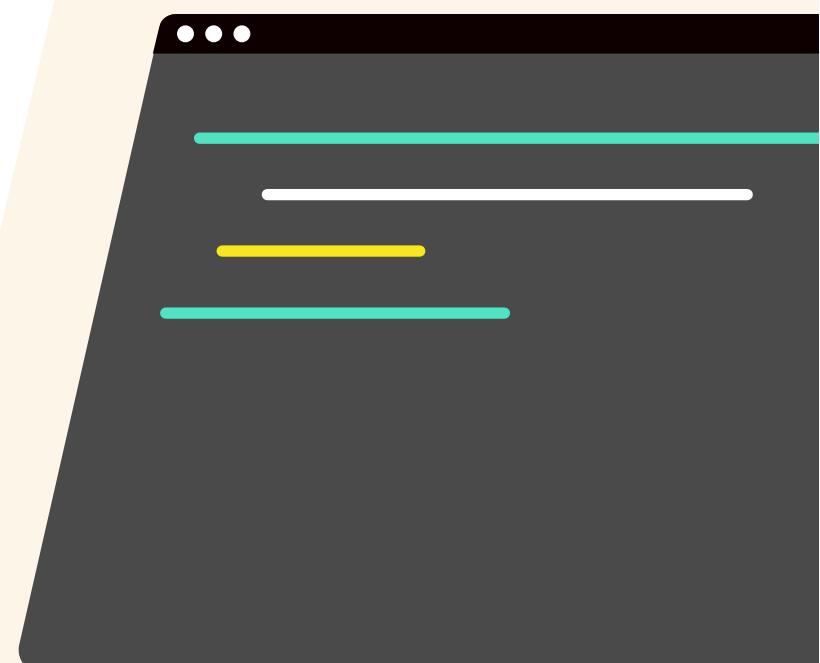
# Analytics or Machine Learning?

**Needs ML**

I want to predict the salaries of potential hires

**Does **not** need ML**

Do women and men earn the same in our company?



# Analytics or Machine Learning?

## Needs ML

I want to predict the salaries of potential hires

Why does this marketing channel bring the most clicks?

## Does **not** need ML

Do women and men earn the same in our company?

Which marketing channel brings the most clicks?



# Analytics or Machine Learning?

## Needs ML

I want to predict the salaries of potential hires

Why does this marketing channel bring the most clicks?

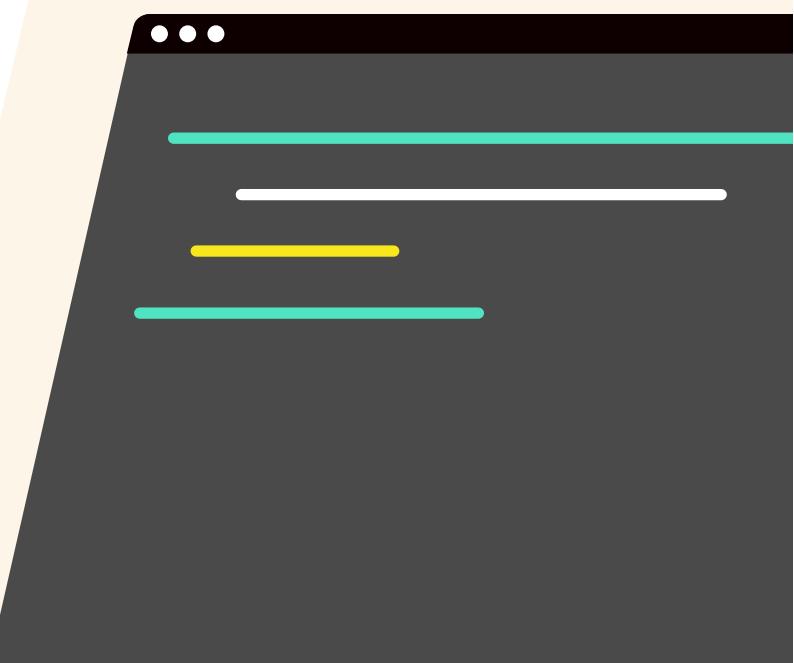
How will our sales look if we open an office in X country?

## Does **not** need ML

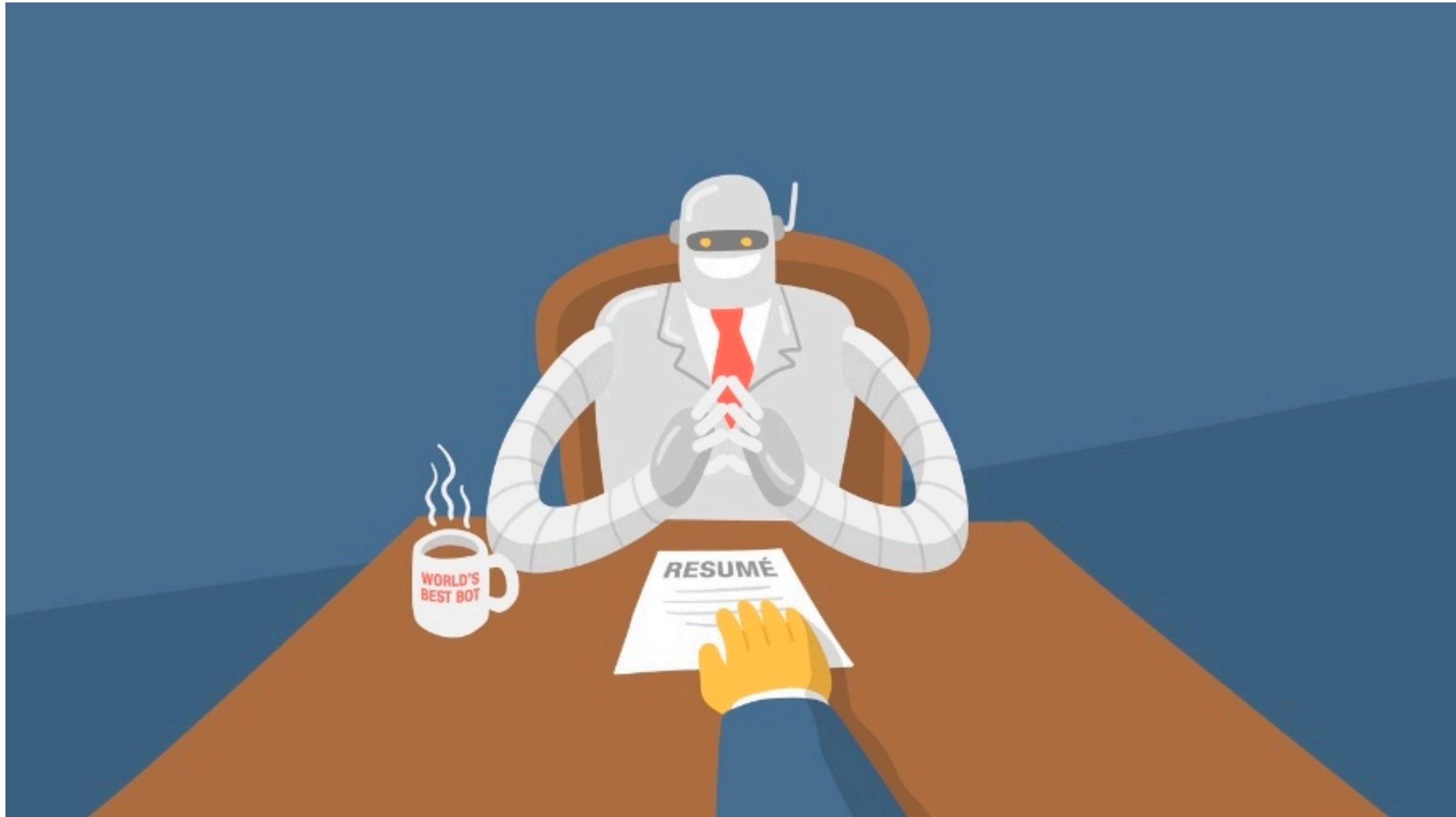
Do women and men earn the same in our company?

Which marketing channel brings the most clicks?

How will our sales look in the next quarter?



# Missing ground truth



Machine Learning for hiring the perfect candidate!



# Missing ground truth



# Missing ground truth



# Missing ground truth



vs



# Missing ground truth



vs



# Things have **changed**



# Things have changed

Let's take a drug dose analysis 💊



# Things have changed

Let's take a drug dose analysis 💊

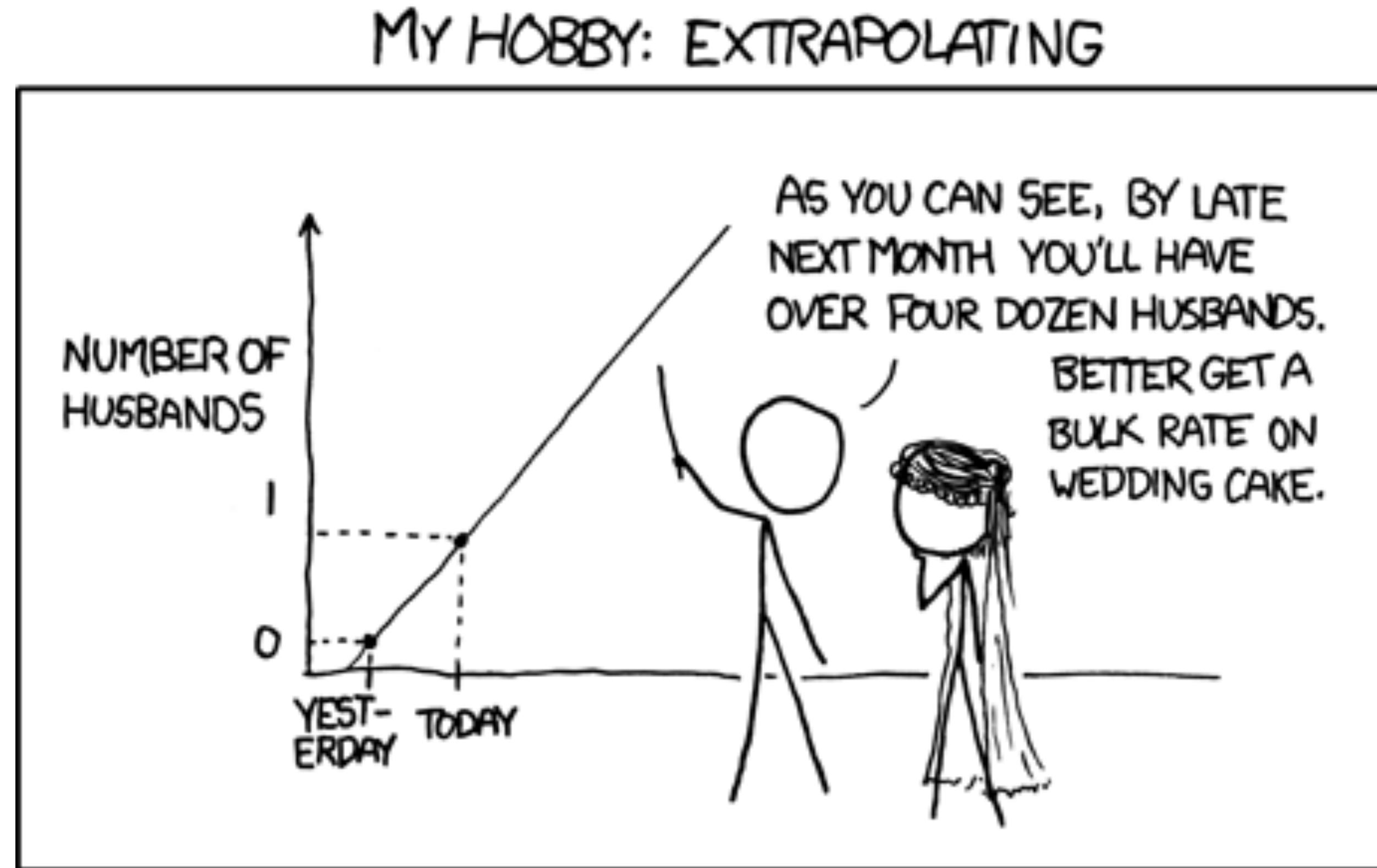
Drug Dosage per Day 💊

Day	Dosage (mg)
1	10
2	15
3	20
4	25
5	30
6	?



# Things have changed

Let's take a drug dose analysis 💊



Source: [xkcd](#)



# Things have **really changed**

TECH \ CORONAVIRUS

## The algorithms big companies use to manage their supply chains don't work during pandemics

*The data the algorithms use isn't reliable*

By Nicole Wetsman | Apr 27, 2020, 1:25pm EDT



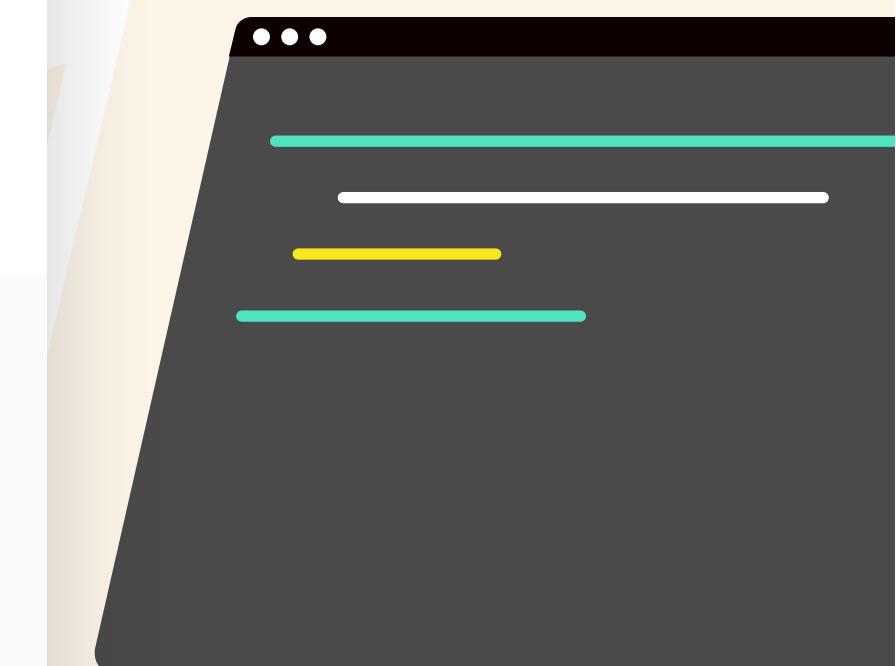
Listen to this article



SHARE



AD



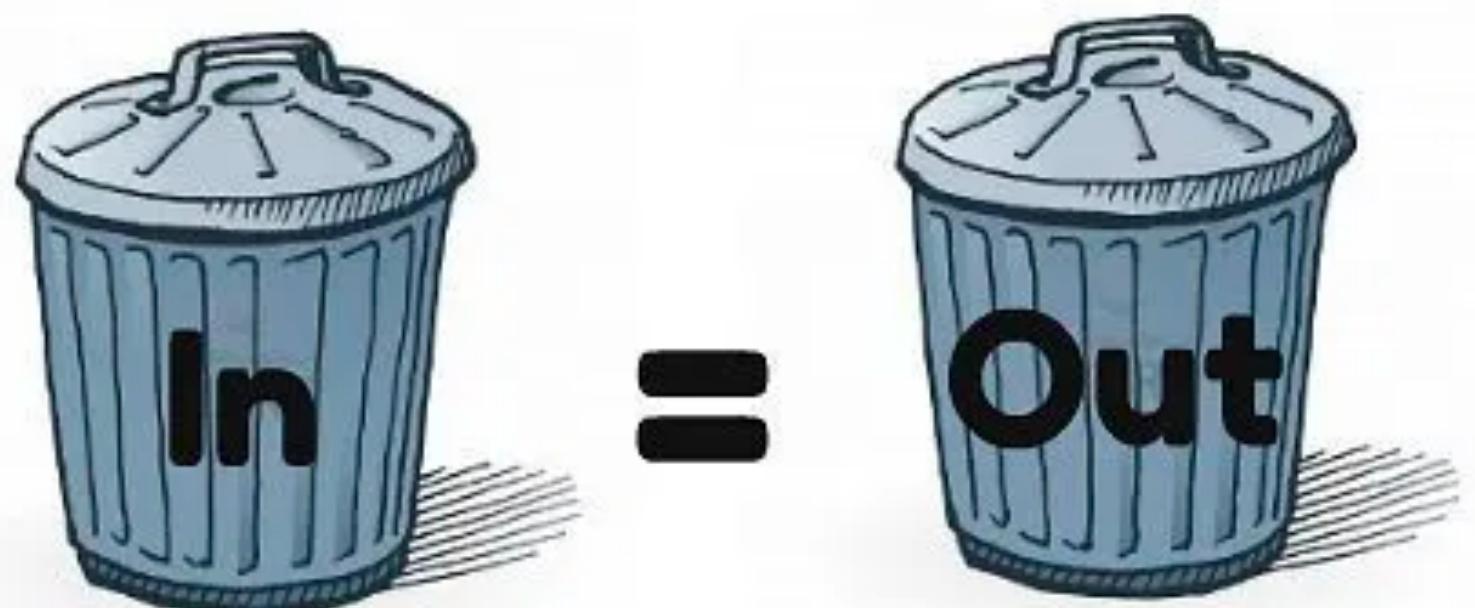
[...] Most firms that think they want advanced AI/ML really just need linear regression on **cleaned-up data.**

— *Tweet* by Robin Hanson (@robinhanson)



[...] Most firms that think they want advanced AI/ML really just need linear regression on **cleaned-up data.**

— *Tweet by Robin Hanson (@robinhanson)*

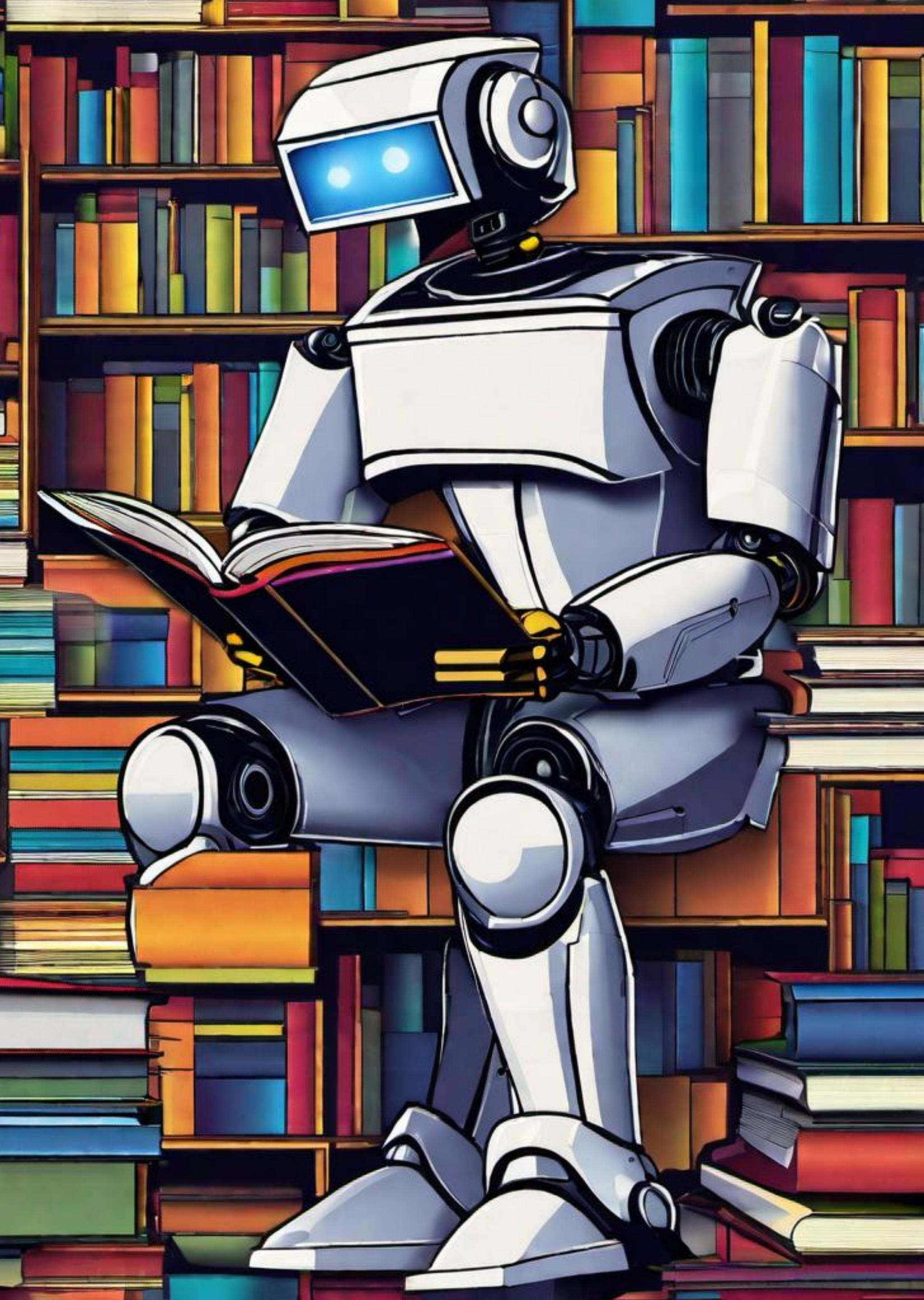


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

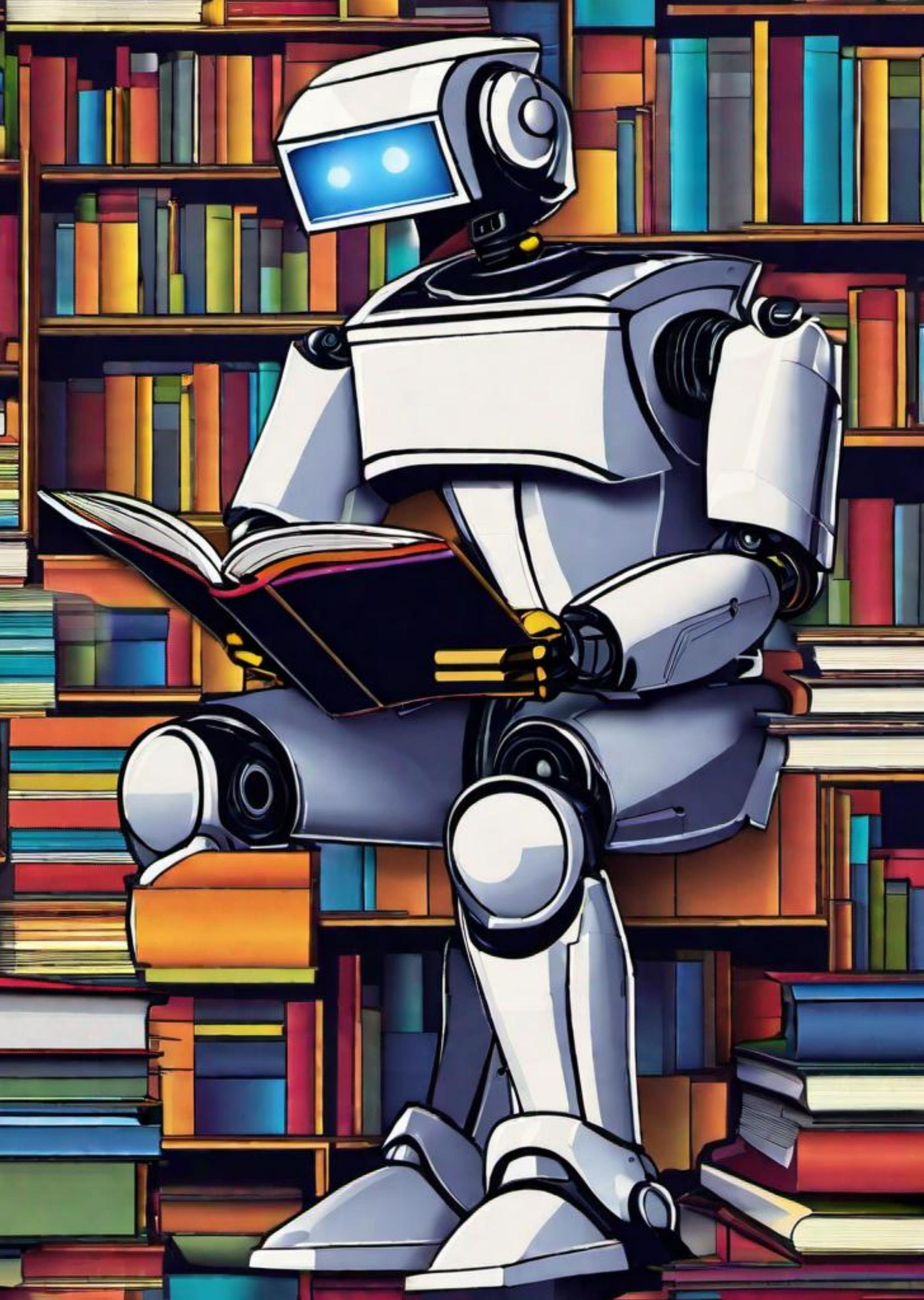


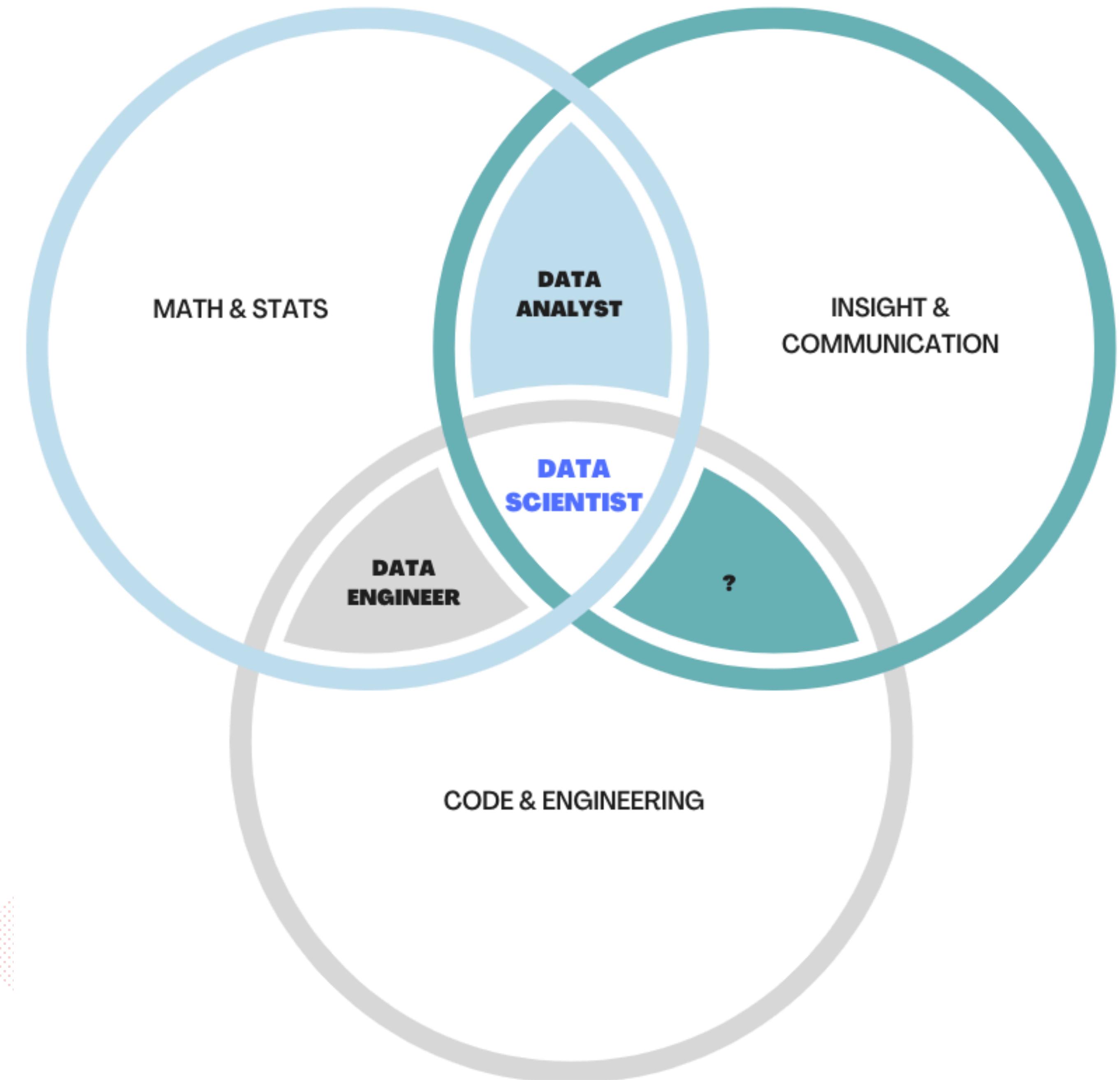
# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon





MATH & STATS

**DATA  
ANALYST**

INSIGHT &  
COMMUNICATION

**DATA  
ENGINEER**

**DATA  
SCIENTIST**

?

CODE & ENGINEERING



# Business Analyst

MATH & STATS

**DATA  
ANALYST**

INSIGHT &  
COMMUNICATION

**DATA  
ENGINEER**

**DATA  
SCIENTIST**

?

CODE & ENGINEERING



# Business Analyst

MATH & STATS

DATA  
ANALYST

INSIGHT &  
COMMUNICATION

DATA  
SCIENTIST

DATA  
ENGINEER

?

CODE & ENGINEERING

# Database Admin



# Business Analyst

MATH & STATS

DATA  
ANALYST

INSIGHT &  
COMMUNICATION

# Database Admin

DATA  
ENGINEER

DATA  
SCIENTIST

?

# Data Product Manager

CODE & ENGINEERING



Business Analyst

Digital Marketing

MATH & STATS

INSIGHT &  
COMMUNICATION

DATA  
ANALYST

DATA  
SCIENTIST

DATA  
ENGINEER

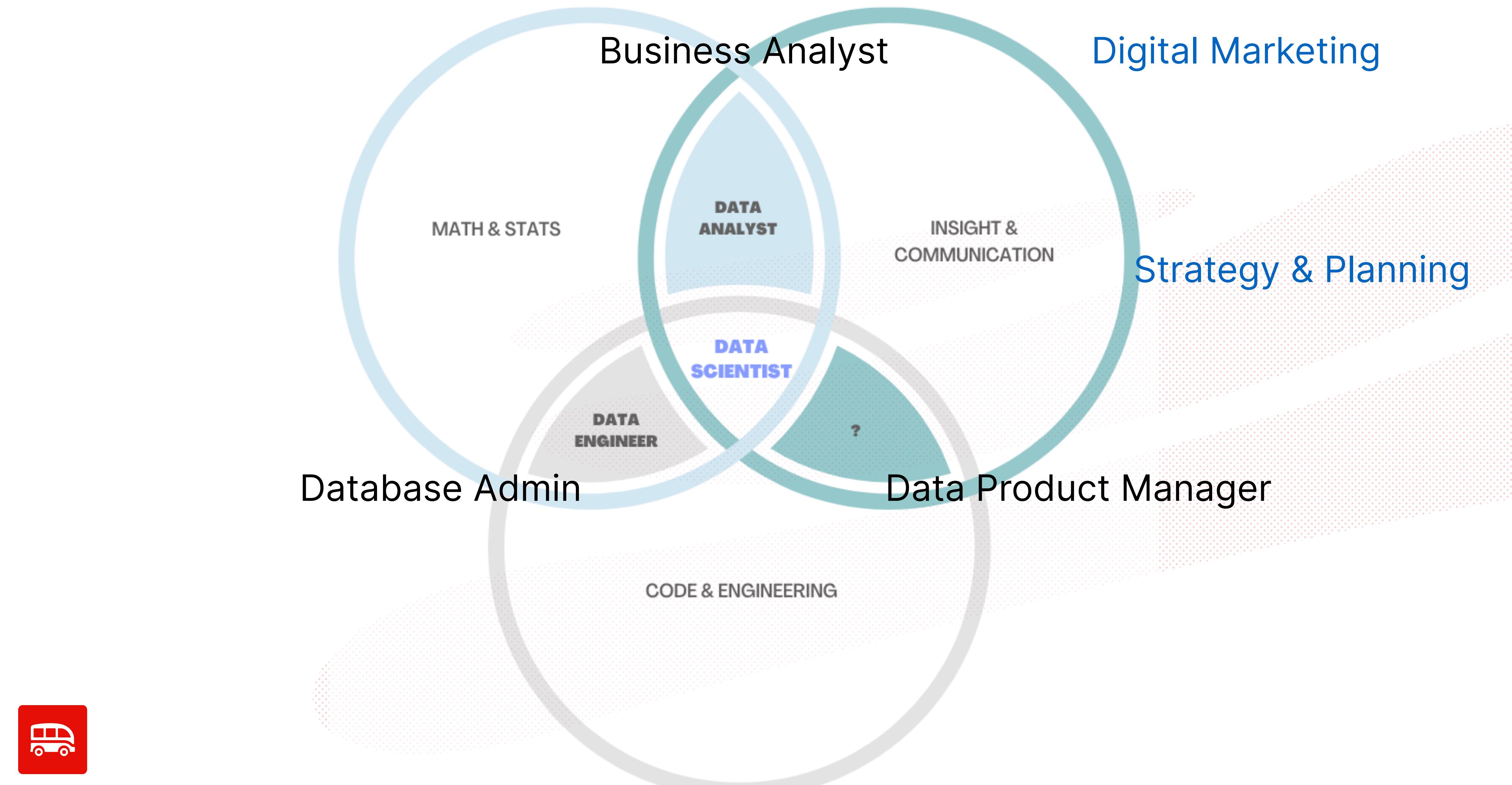
?

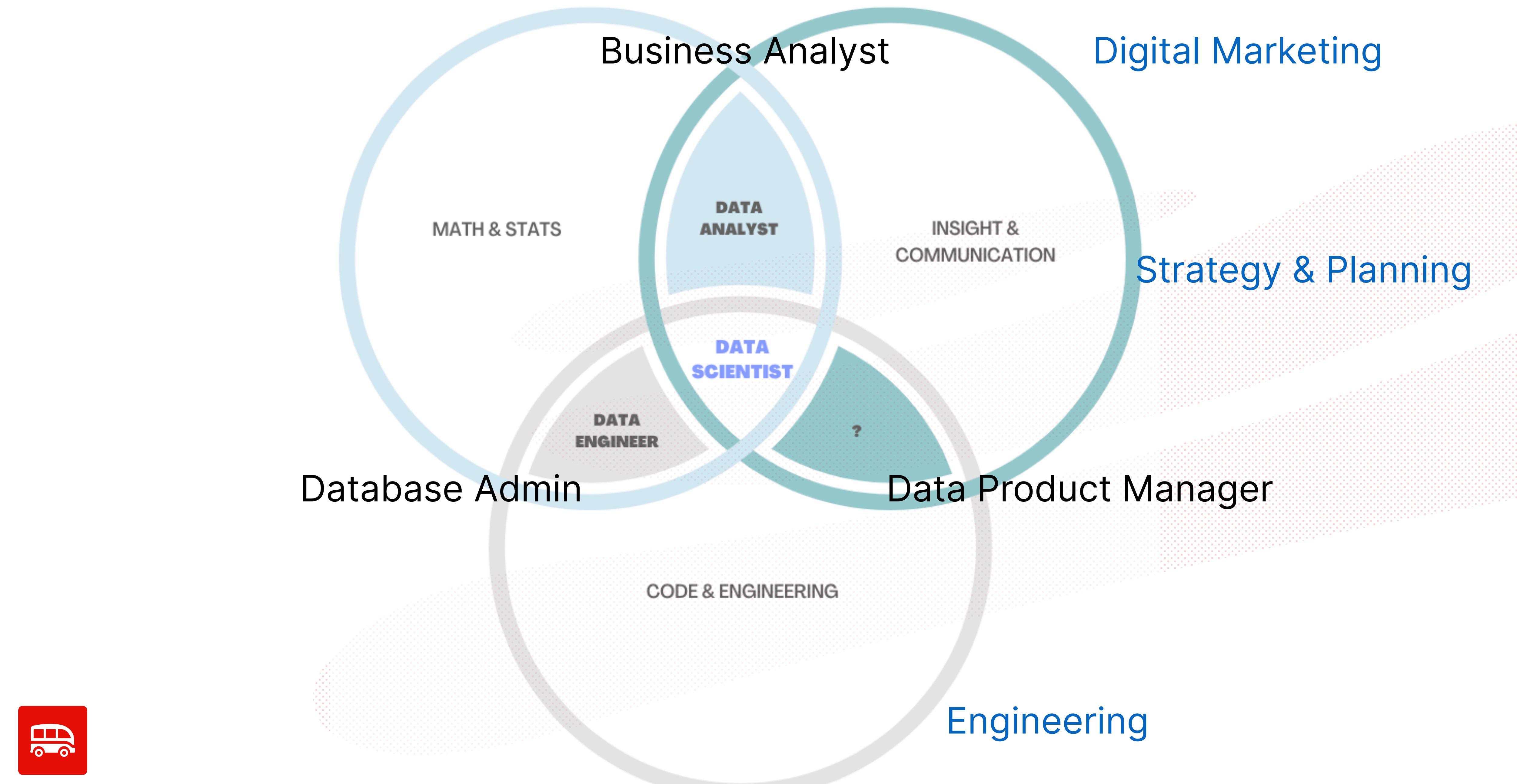
CODE & ENGINEERING

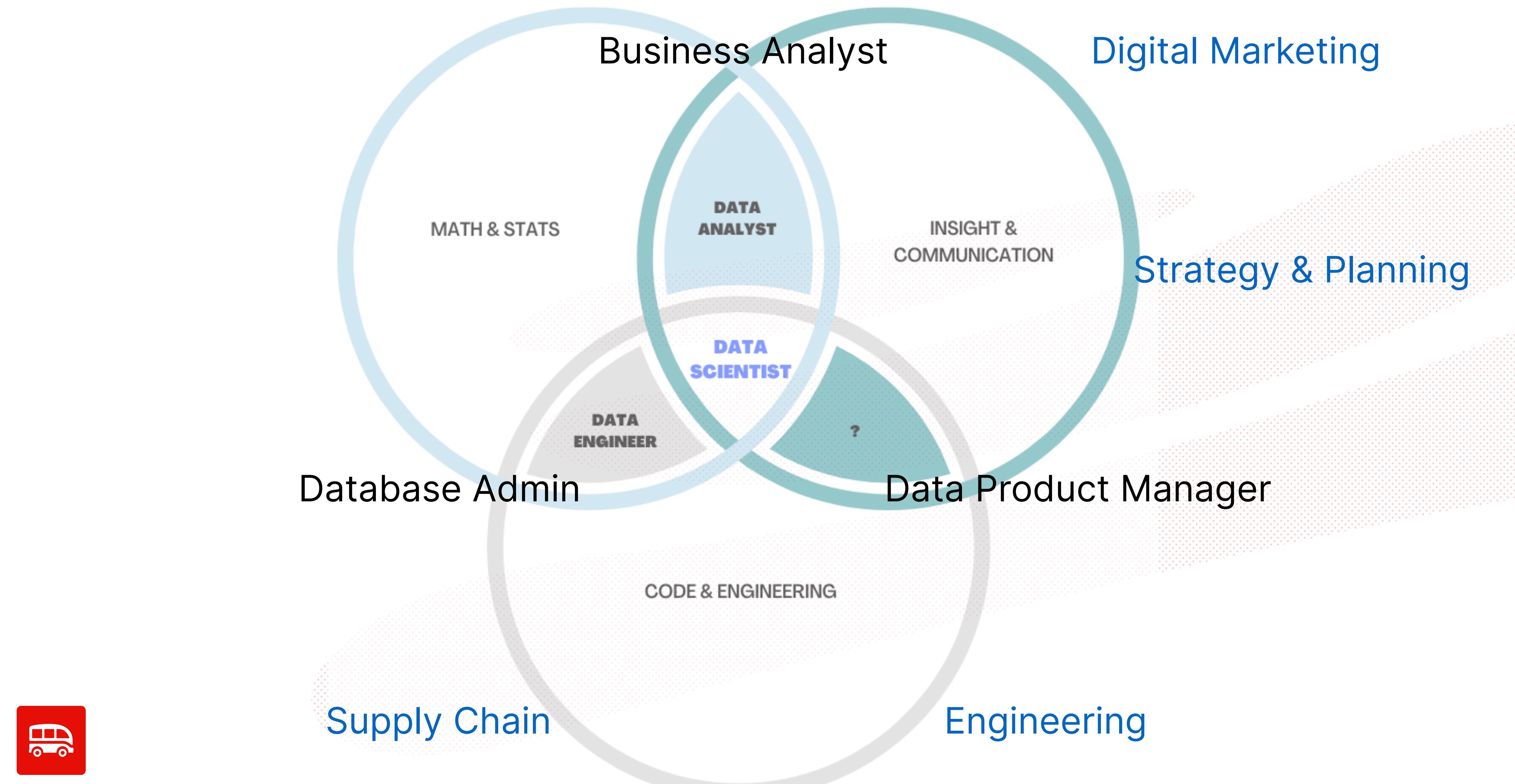
Database Admin

Data Product Manager









Operations

MATH & STATS

Business Analyst

Digital Marketing

DATA  
ANALYST

INSIGHT &  
COMMUNICATION

Strategy & Planning

DATA  
ENGINEER

DATA  
SCIENTIST

?

Data Product Manager

CODE & ENGINEERING

Engineering

Database Admin

Supply Chain



Finance & HR

Operations

Database Admin

Supply Chain

Business Analyst

Digital Marketing

INSIGHT &  
COMMUNICATION

Data Product Manager

Engineering

MATH & STATS

DATA  
ENGINEER

CODE & ENGINEERING

DATA  
SCIENTIST

DATA  
ANALYST

?



# Data Science **as a skill**



# Data Science **as a skill**



# Data Science **as a skill**



Data Journalism



Film Coloring



Bicycle Building



# Data Science **as a skill**



Data Journalism



Film Coloring



Bicycle Building



**Finance, Marketing, Strategy**

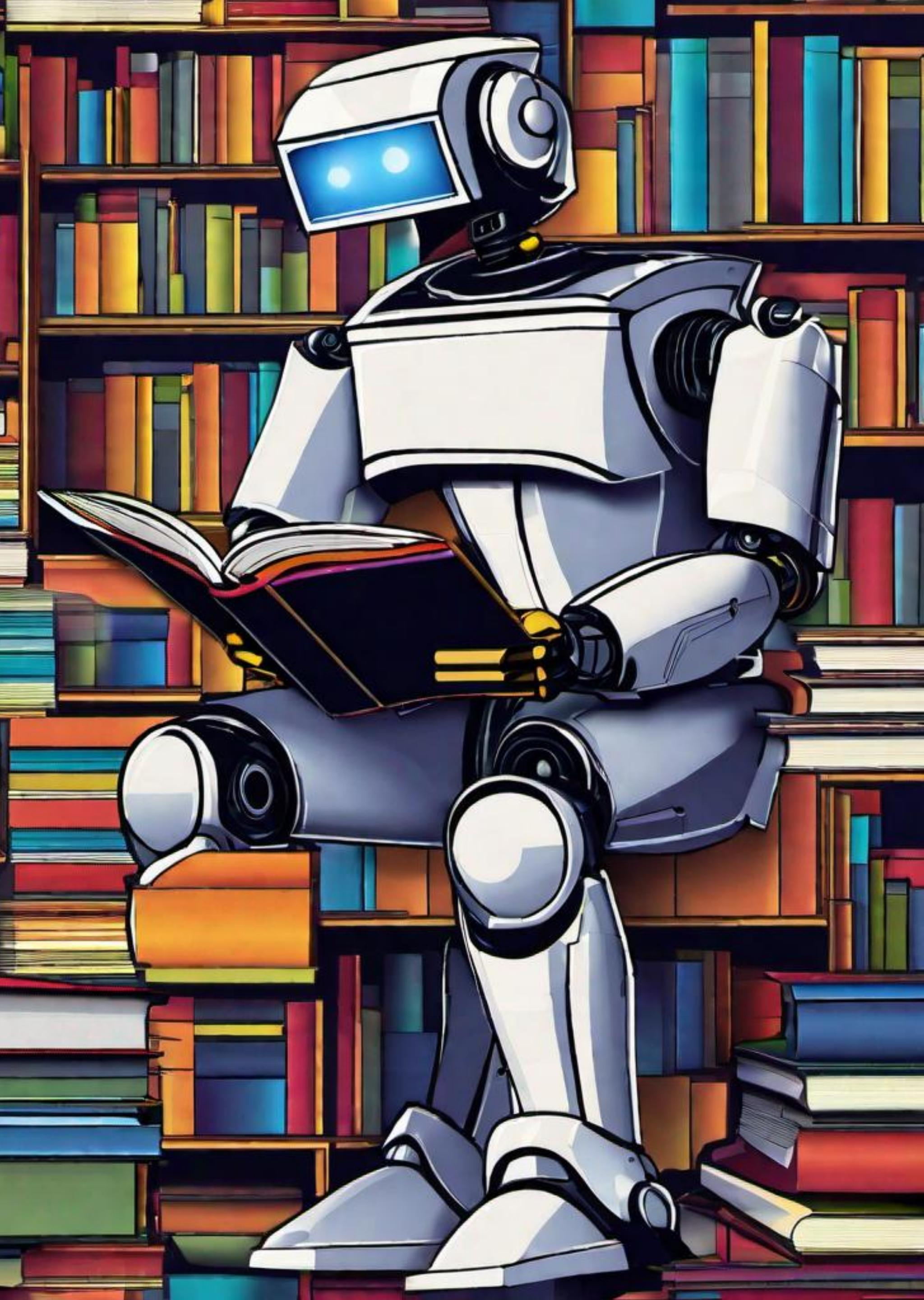


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

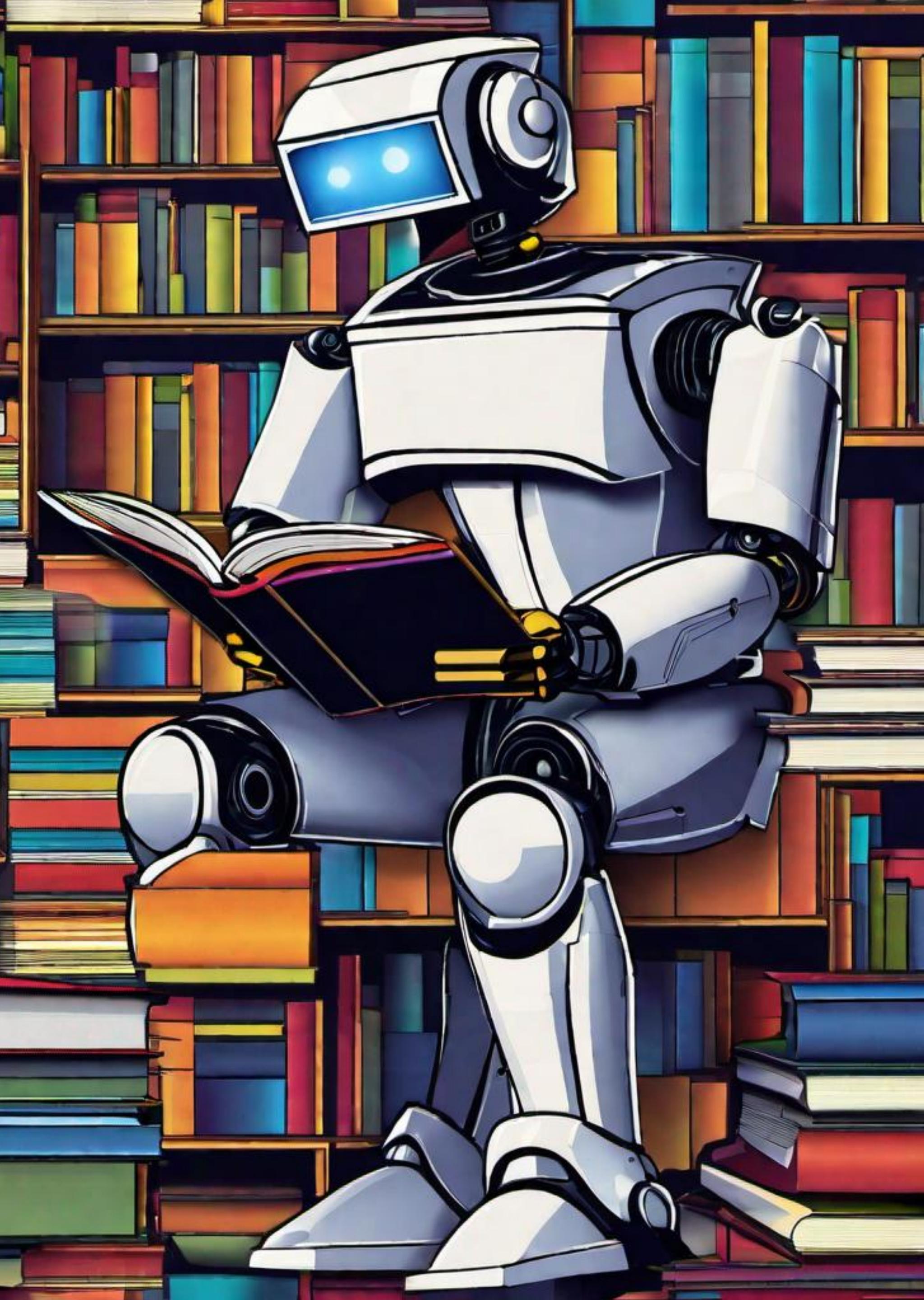


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon



# Ready to get **nerdy**? 😎

RStudio Connect

https://colorado.rstudio.com/rsc/jupyter-notebook-visualization/jupyter-static-visualization.html

## Python Visualization Libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

### Matplotlib

```
In [2]: np.random.seed(0)

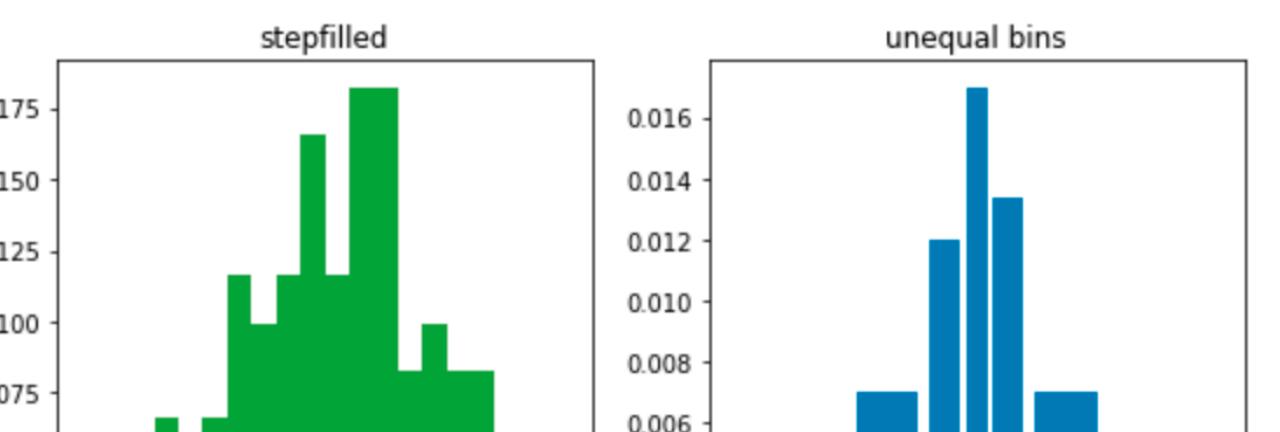
mu = 200
sigma = 25
x = np.random.normal(mu, sigma, size=100)

fig, (ax0, ax1) = plt.subplots(ncols=2, figsize=(8, 4))

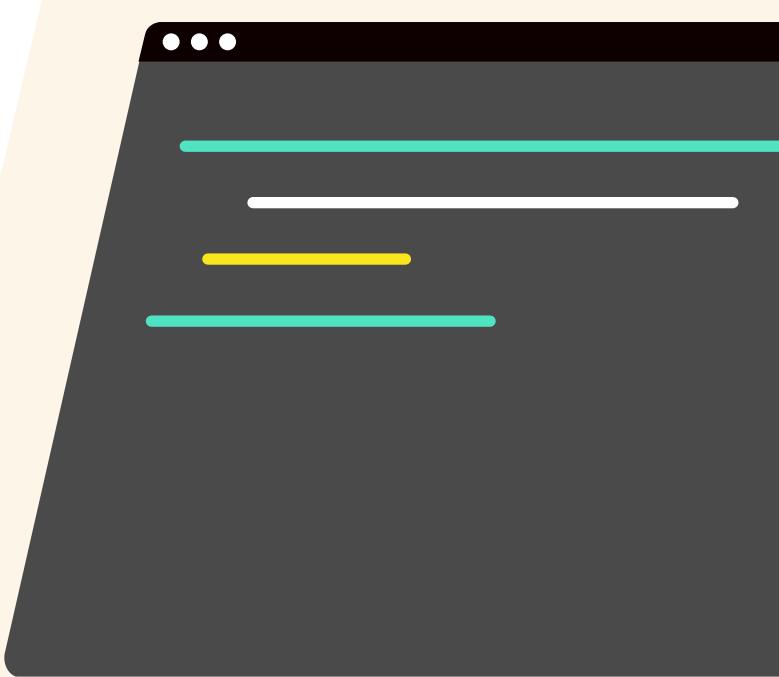
ax0.hist(x, 20, density=1, histtype='stepfilled', facecolor='g', alpha=0.75)
ax0.set_title('stepfilled')

# Create a histogram by providing the bin edges (unequally spaced).
bins = [100, 150, 180, 195, 205, 220, 250, 300]
ax1.hist(x, bins, density=1, histtype='bar', rwidth=0.8)
ax1.set_title('unequal bins')

fig.tight_layout()
plt.show()
```



# First, some analysis



# What's the Data?

Understand what we have

- ✓ How many rows and columns?
- ✓ What are the columns we have?
- ✓ What are the averages?
- ✓ What are the minima and maxima?

Methods to explore your data:

```
In [ ]: # to get the number of rows, columns  
salaries.shape
```

```
In [ ]: # to get the columns and their data types  
salaries.dtypes
```

```
In [ ]: # to get a readable summary of your data  
round(salaries.describe())
```

```
In [ ]: # to see only one column of the dataset  
salaries["Column Name"]
```

```
In [ ]: # to see multiple columns of the dataset  
salaries[["Column Name 1", "Column Name 2"]]
```



# Visualize the Data

Get a sense of what it tells you

- ✓ How many of each category do we have?
- ✓ Do any two columns relate to each other?
- ✓ Do some categories influence the output?
- ✓ First step of any Data Scientist!



# Visualize the Data

Get a sense of what it tells you

- ✓ How many of each category do we have?

```
In [ ]: # to count the distribution of values in a column  
sns.countplot(data=salaries, x='Column Name')
```

- ✓ Do any two columns relate to each other?

```
In [ ]: # to see the relation of one column to another  
sns.scatterplot(data=salaries, x='Input Column', y='Output Column')
```

- ✓ Do some categories influence the output?

```
In [ ]: # to add a category to the visualization, we use `hue`  
sns.scatterplot(data=salaries, x='Input Column', y='Output Column', hue='Category Column')
```



>\_

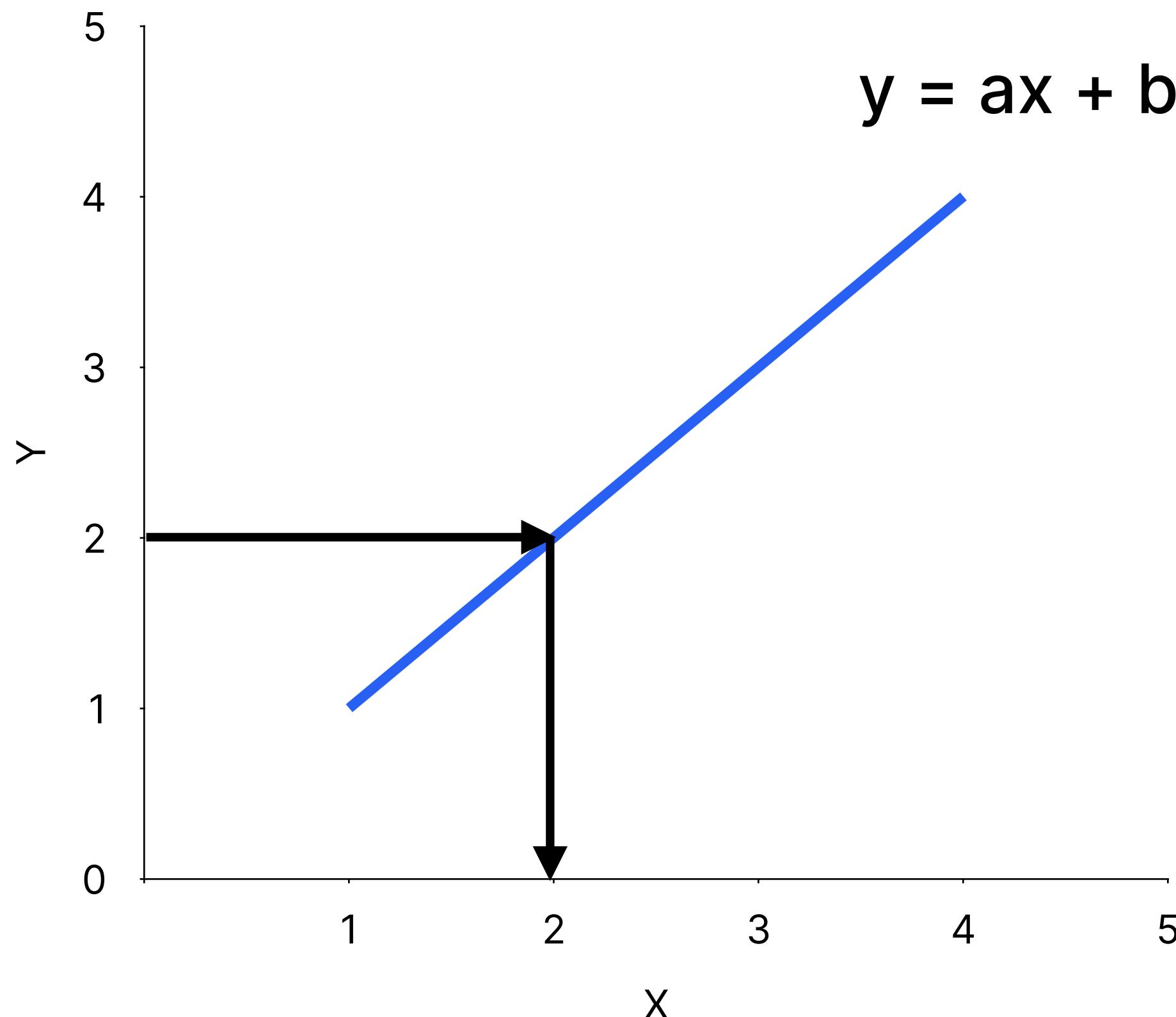


Challenge 1

# Predicting salaries 💰



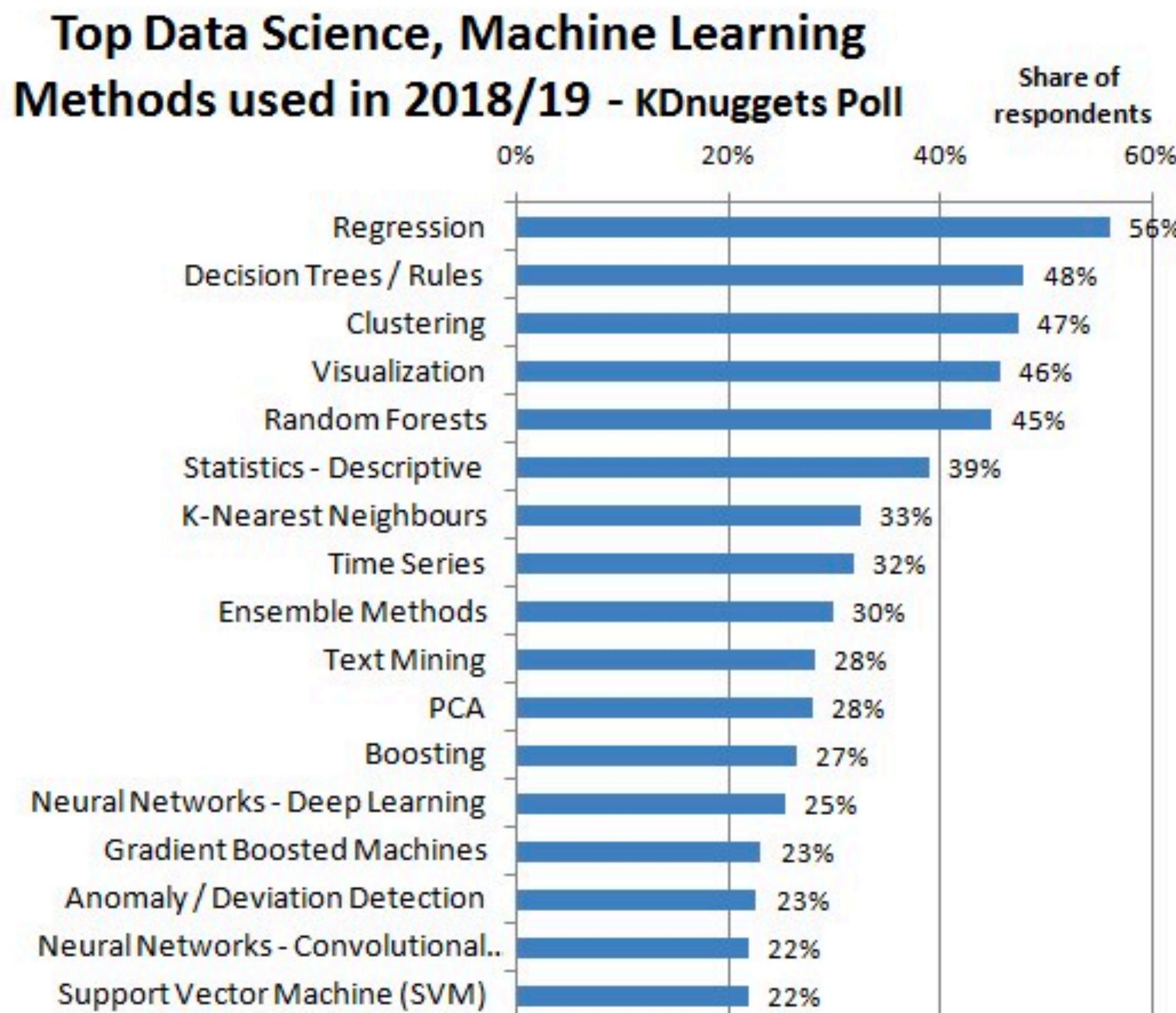
# Linear Regression



[...] Most firms that think they want advanced AI/ML really just need linear regression on **cleaned-up data.**

— *Tweet by Robin Hanson (@robinhanson)*

# Why Regression?



>\_

**Visual demo on setosa.io!**



# Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data



# Road to Machine Learning

## 1. Select the **features** and **targets**

Setting our features (inputs):

```
In [ ]: # we can select all needed columns...
features = salaries[["Gender", "Age", "Department_code", "Years_exp", "Tenure (months)"]]
```

```
In [ ]: # ...or we can simply drop the not needed!
features = salaries.drop(["Department", "Gross"], axis="columns")
```

Setting our target (output):

```
In [ ]: # we can simply select the column we need
target = salaries["Gross"]
```



# Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. Train the model
4. Score the model's performance
5. Predict with new data

Once we find the model we need, it's easy to import

```
In [ ]: # the syntax looks like this  
from sklearn.MODEL_TYPES import MODEL_YOU_NEED
```

```
In [ ]: # with Linear Regression we need this  
from sklearn.linear_model import LinearRegression
```

After importing, we need to **initialize** the model, like this:

```
In [ ]: model = LinearRegression()
```



# Road to Machine Learning

1. Select the **features** and **targets**
2. Import the model from Sklearn
3. Train the model
4. Score the model's performance
5. Predict with new data

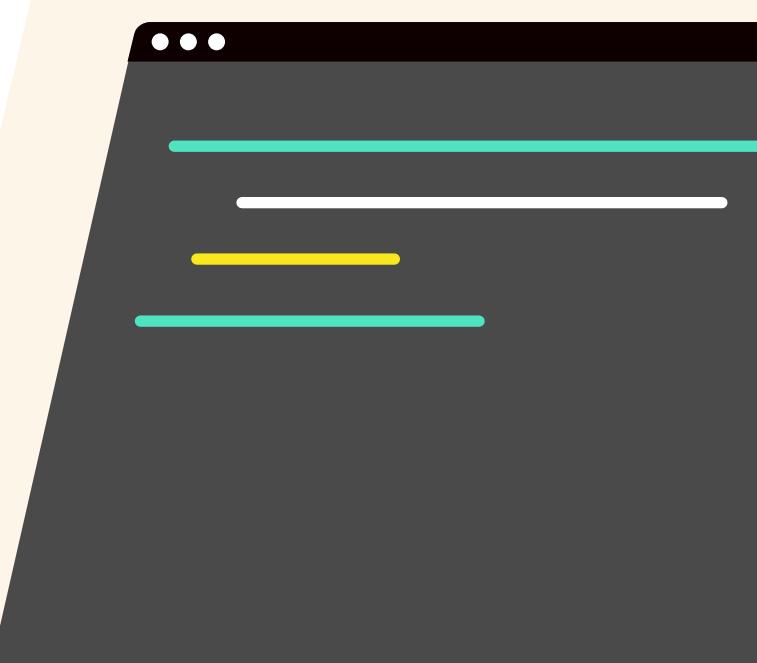
To train the model, we use the `.fit` method:

```
In [ ]: model.fit(features, target)
```

The `model` then finds the **best fitting** line between features and target



>\_



# Road to Machine Learning

1. Select the **features** and **targets**
2. Import the model from Sklearn
3. Train the model
4. Score the model's performance
5. Predict with new data

To score the model, we use the `.score` method:

```
In [ ]: model.score(features, targets)
```

We need to give it some **test data** to do the scoring



# Road to Machine Learning

1. Select the **features** and **targets**

2. Import the model from Sklearn

3. Train the model

4. Score the model's performance

5. Predict with new data

To predict -- you guessed it ;)

We use the `.predict` method:

In [ ]: `model.predict(new_data)`

`new_data` is the info we want our model to use to predict a new target (output).

In our case, it's a new `hire` !



>\_

# Road to Machine Learning

1. Select the **features** and **targets**
2. Import the model from Sklearn
3. Train the model
4. Score the model's performance
5. Predict with new data

The things that a Linear Regression model "learns" are **coefficients** and **intercept**.

The **coefficients** show how each feature influences the target:

```
In [ ]: model.coef_
```

The **intercept** shows what would the target be when all features are at zero:

```
In [ ]: model.intercept_
```



# Road to Machine Learning

1. Select the **features** and **targets**
2. Import the model from Sklearn
3. Train the model
4. Score the model's performance
5. Predict with new data
6. Explain the model

The things that a Linear Regression model "learns" are **coefficients** and **intercept**.

The **coefficients** show how each feature influences the target:

```
In [ ]: model.coef_
```

The **intercept** shows what would the target be when all features are at zero:

```
In [ ]: model.intercept_
```

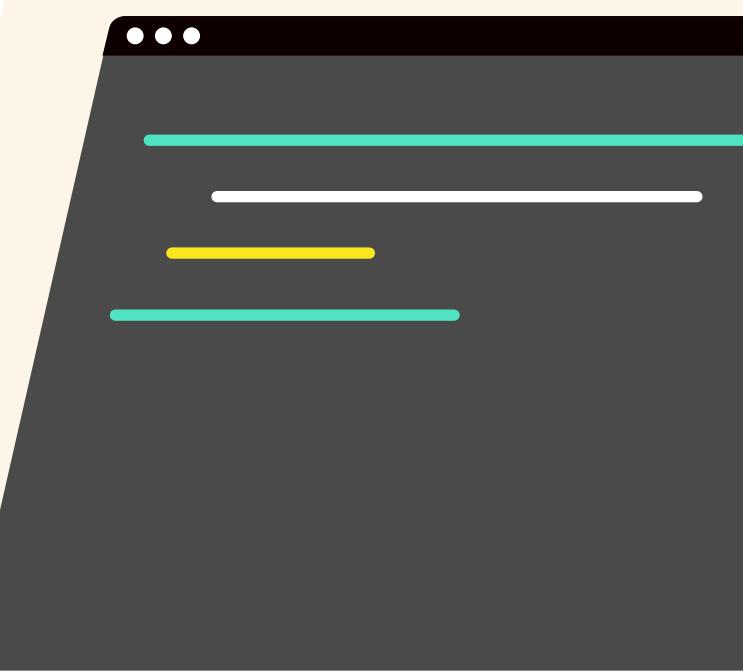


# That easy?! 😮

Yes...but no



>\_



# That easy?! 😱

Yes...but no

```
model.fit(features, target)
```

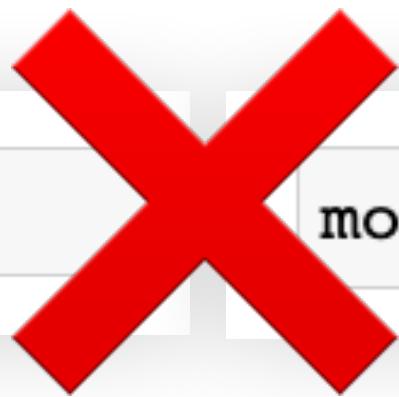
```
model.score(features, targets)
```



# That easy?! 😱

Yes...but no

```
model.fit(features, target)
```



```
model.score(features, targets)
```



# That easy?! 😱

More on this in challenge 2

```
model.fit(features, target)
```



```
model.score(features, targets)
```



```
101  
1101  
01101  
110101  
110001  
1001
```

Data Leakage



## Optional Challenge 2

**Customer churn**



# Regression VS Classification



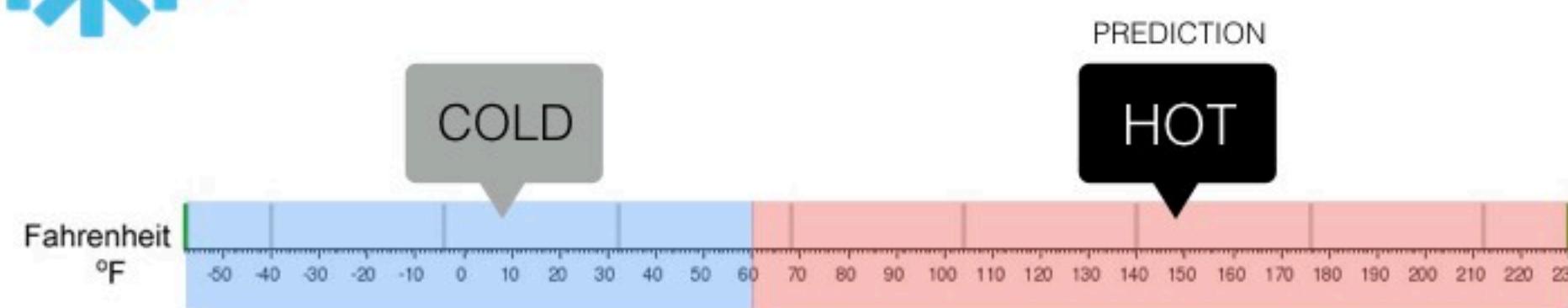
## Regression

What is the temperature going to be tomorrow?

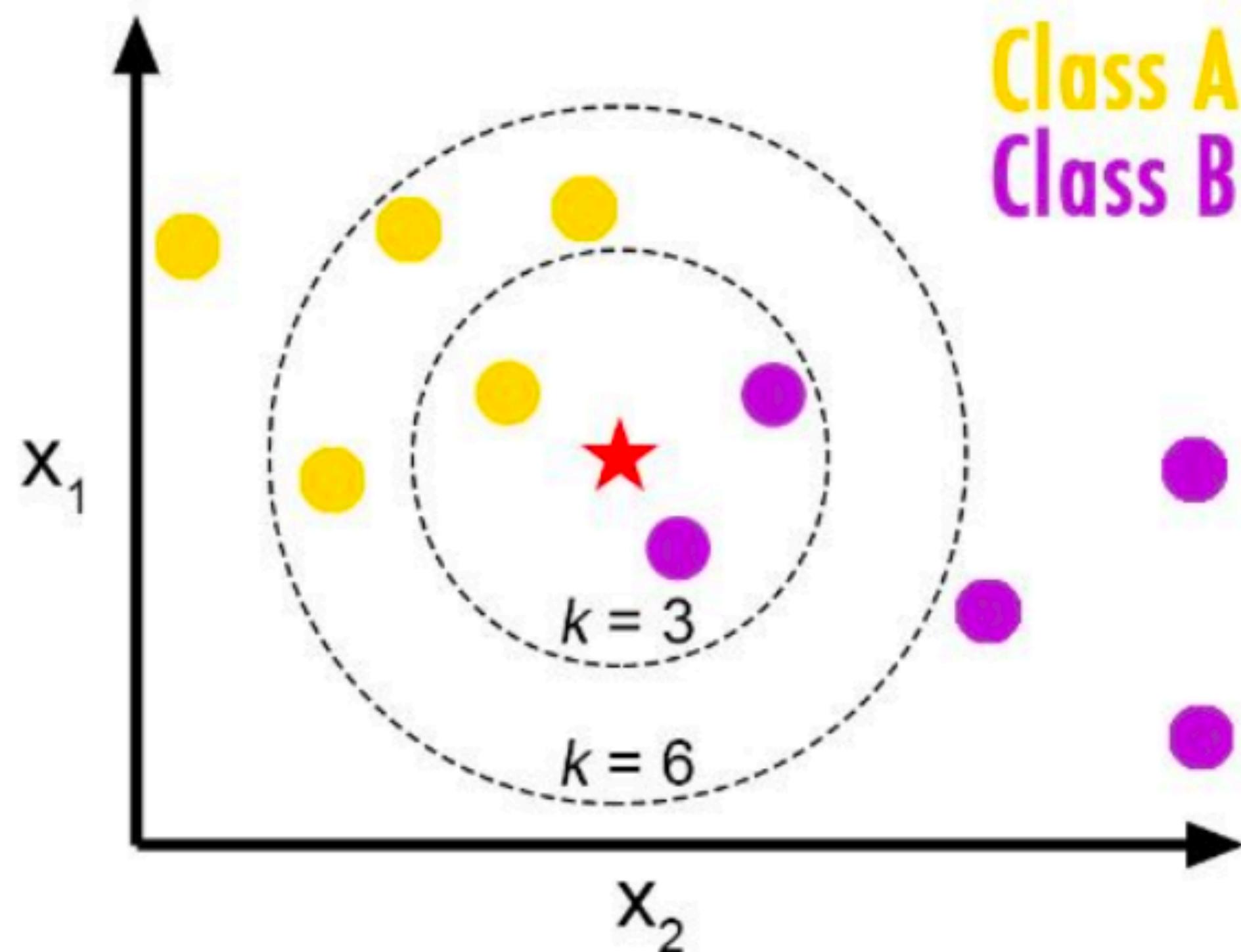


## Classification

Will it be Cold or Hot tomorrow?



# K-Nearest Neighbors



# Visual demo on ml-playground!



# We have a leak



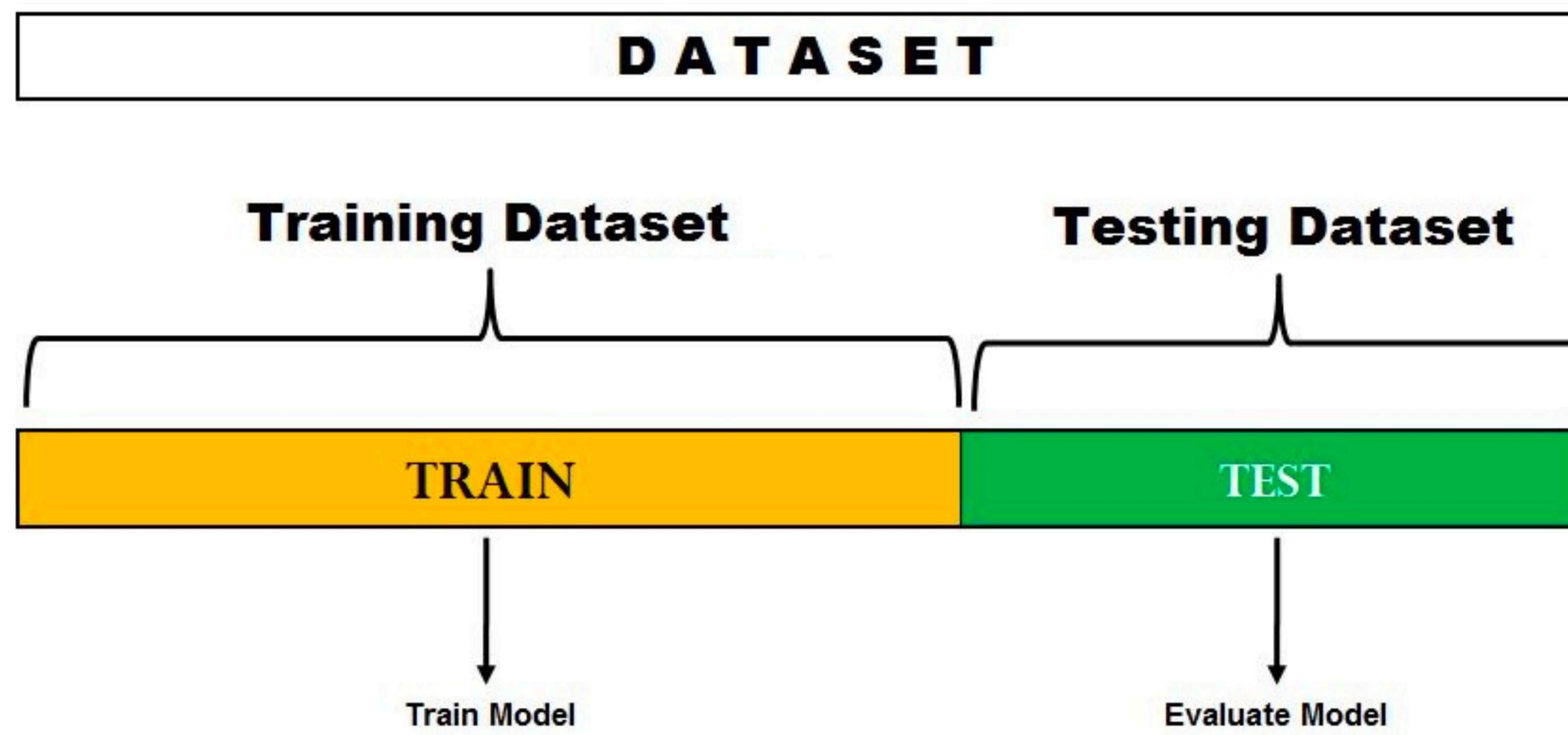
101  
1101  
01101  
110101  
110001  
1001

Data Leakage



>\_

# Holdout Method



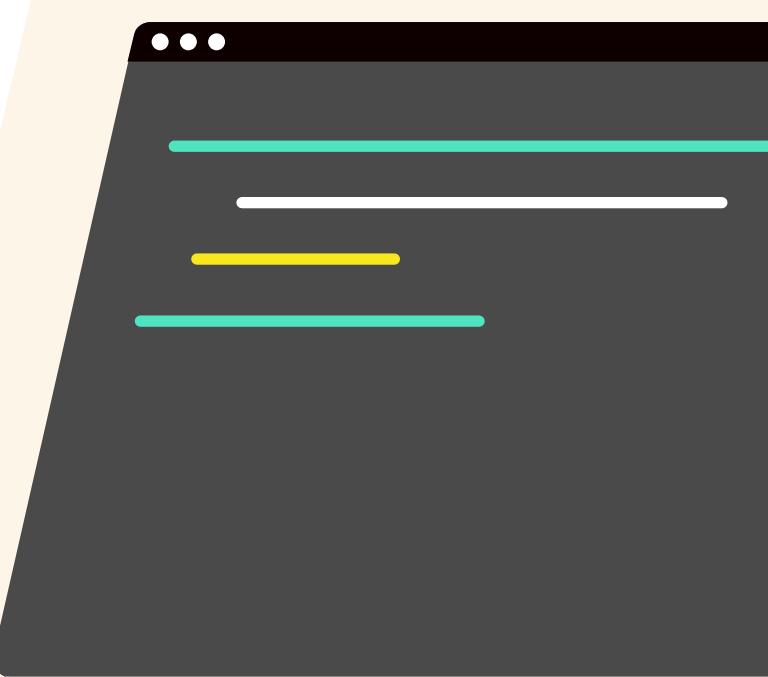
# Congrats! 🙌

Now you know



le wagon

>\_



# Congrats! 🙌

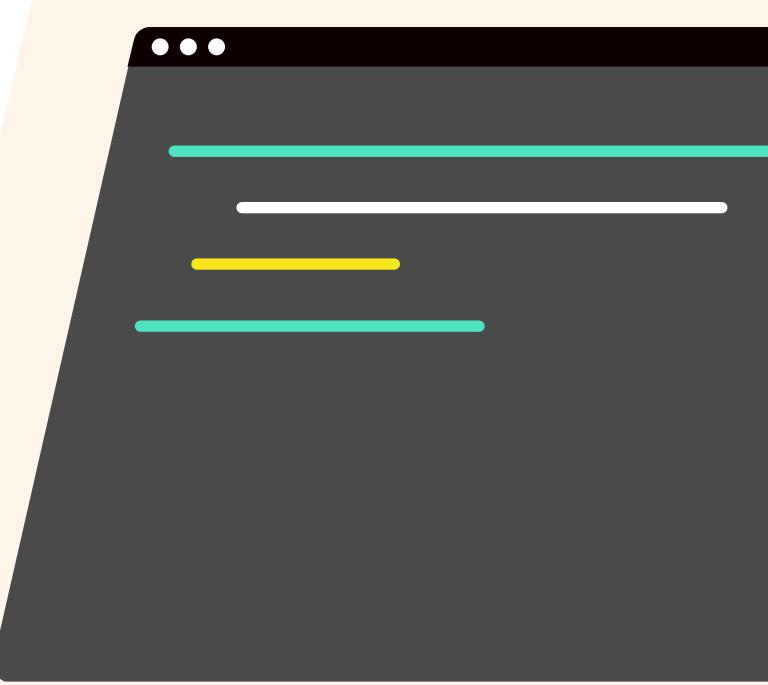
Now you know

- ✓ What the concept of Machine Learning is



le wagon

>\_



# Congrats! 🙌

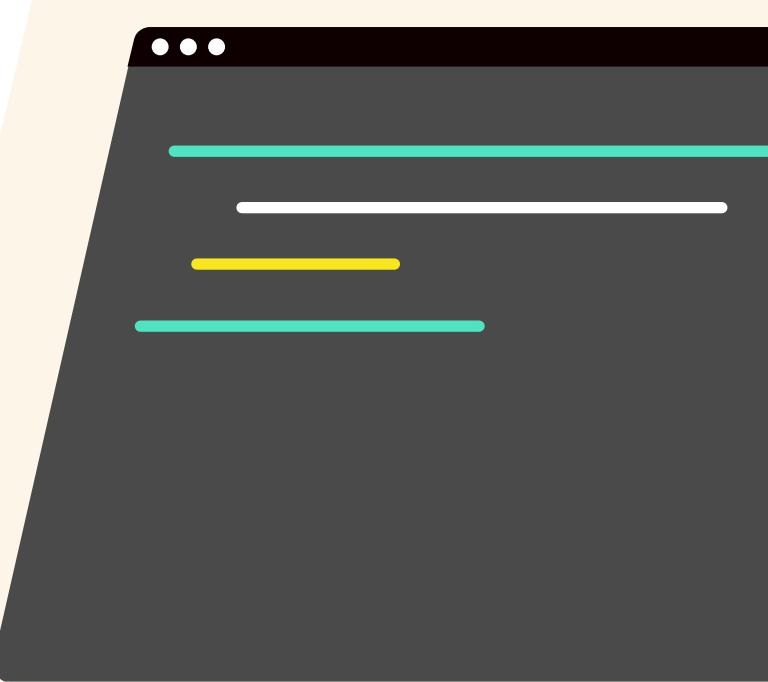
Now you know

- ✓ What the concept of Machine Learning is
- ✓ The basics of Jupyter Notebook - the #1 tool of any Data Scientist 💡



le wagon

>\_



# Congrats! 🙌

Now you know

- ✓ What the concept of Machine Learning is
- ✓ The basics of Jupyter Notebook - the #1 tool of any Data Scientist 💡
- ✓ How to use Python libraries - of which there are thousands 🎉



le wagon

# Congrats! 🙌

Now you know

- ✓ What the concept of Machine Learning is
- ✓ The basics of Jupyter Notebook - the #1 tool of any Data Scientist 💡
- ✓ How to use Python libraries - of which there are thousands 🎉
- ✓ How to import and visualize a CSV dataset 12  
34



le wagon

>\_



# Congrats! 🙌

Now you know

- ✓ What the concept of Machine Learning is
- ✓ The basics of Jupyter Notebook - the #1 tool of any Data Scientist 💡
- ✓ How to use Python libraries - of which there are thousands 🎉
- ✓ How to import and visualize a CSV dataset 12  
34
- ✓ And of course: **how to build your own ML models** 🚀

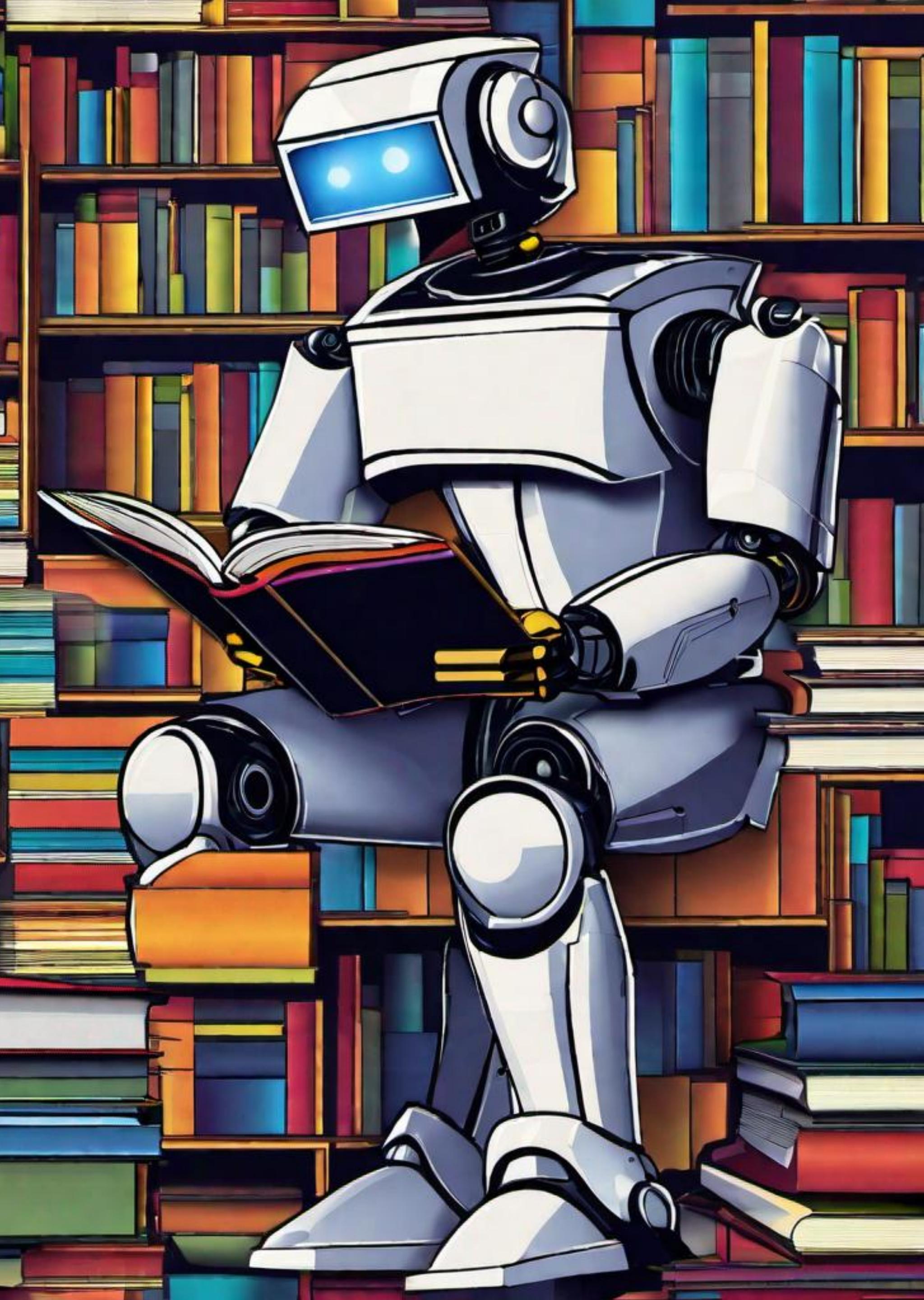


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

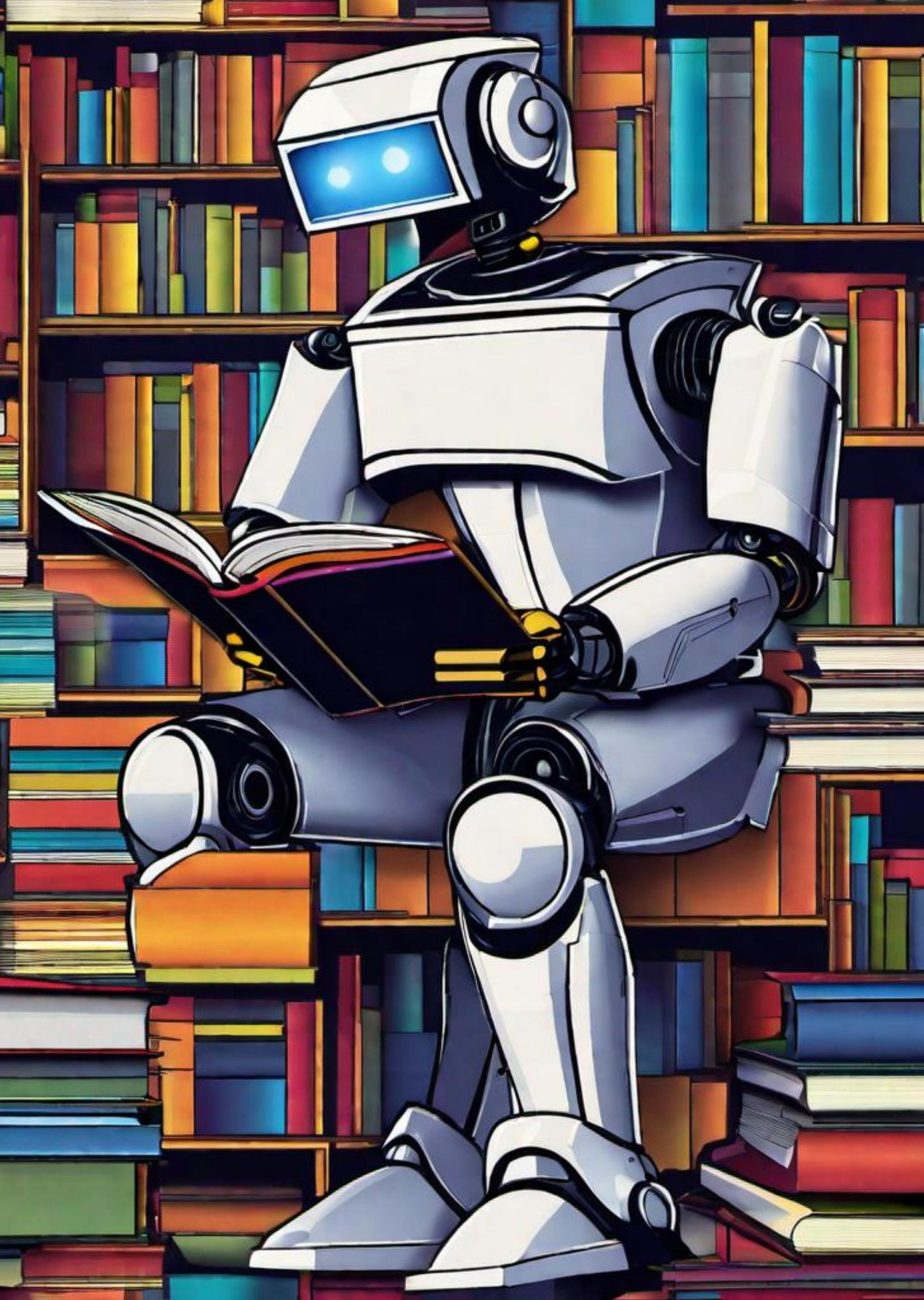


# Agenda

- ✓ What is Machine Learning
- ✓ What is NOT Machine Learning
- ✓ Who are the people building ML
- ✓ Let's code our own models! 🚀
- ✓ What we didn't cover



le wagon

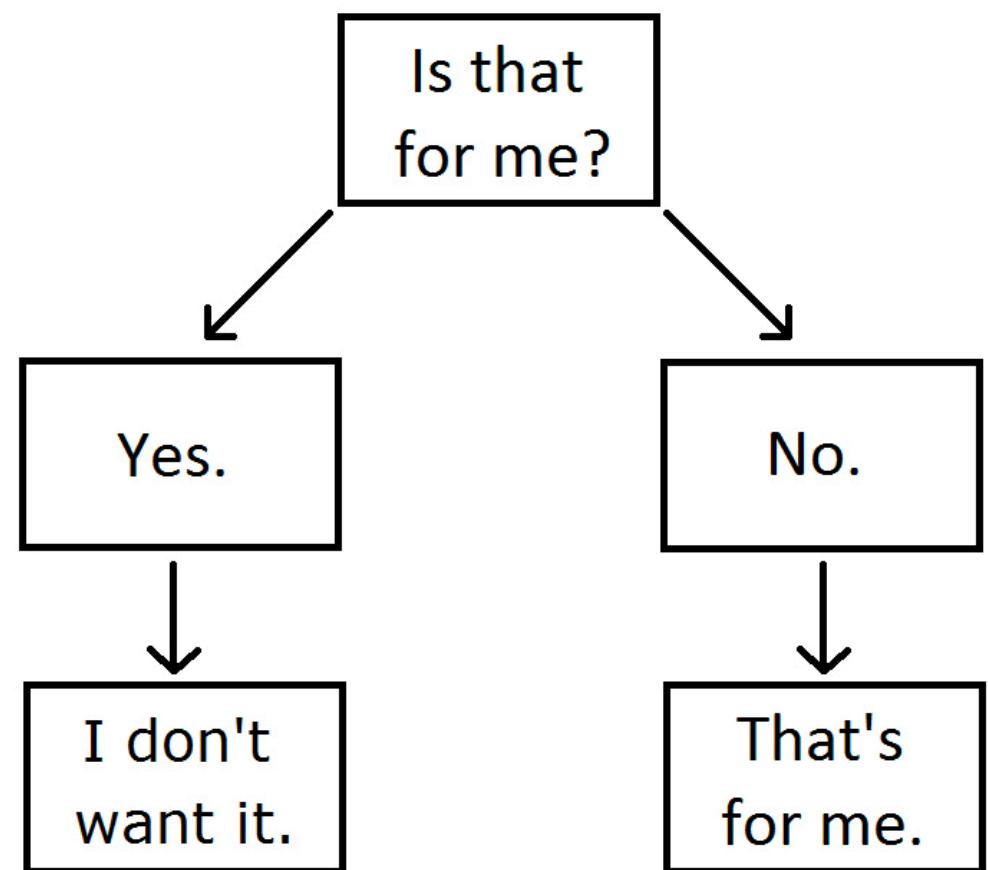


# Other model types



# Other model types

My Cat's Decision-Making Tree.

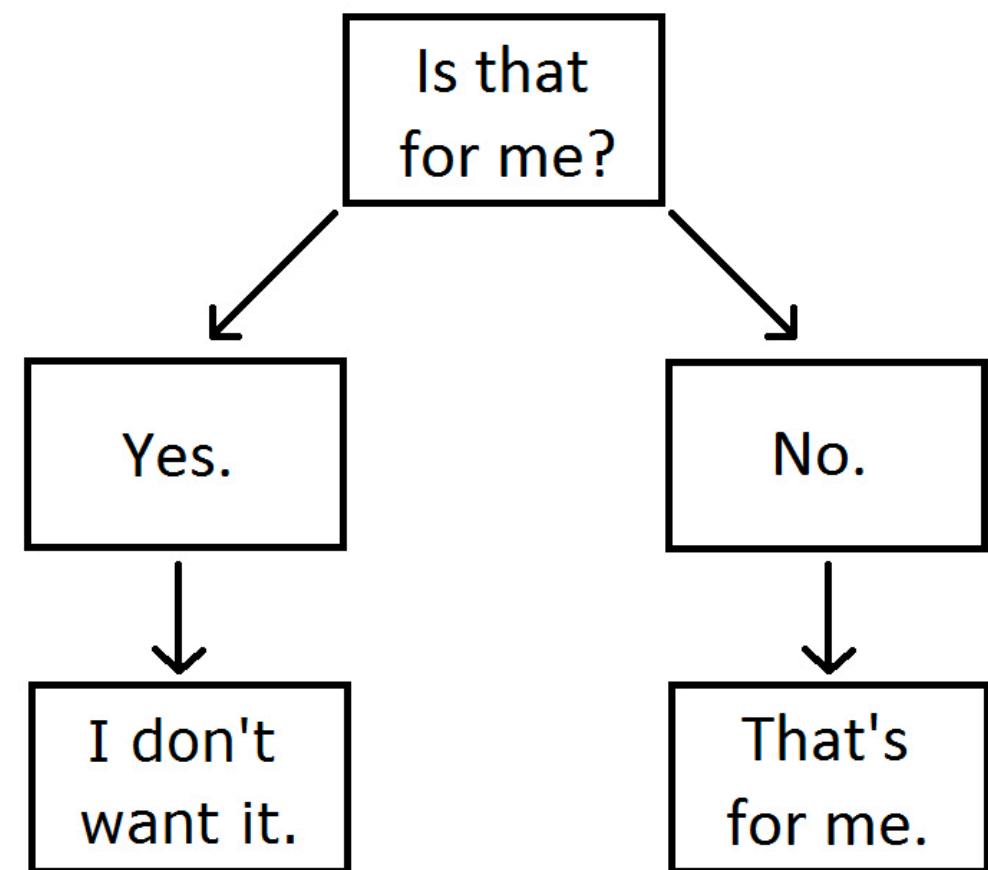


Decision tree based

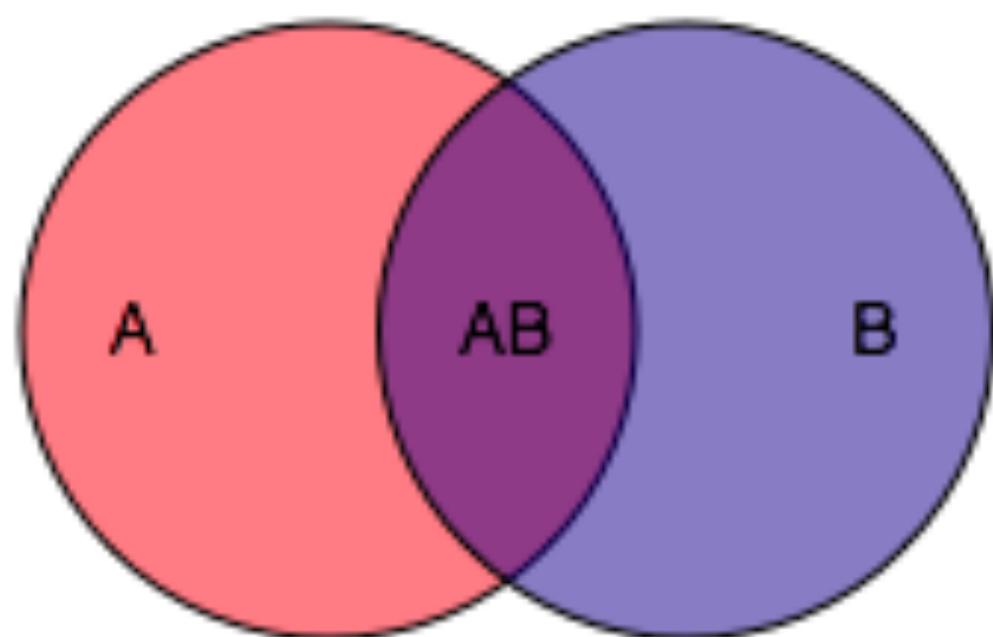


# Other model types

My Cat's Decision-Making Tree.



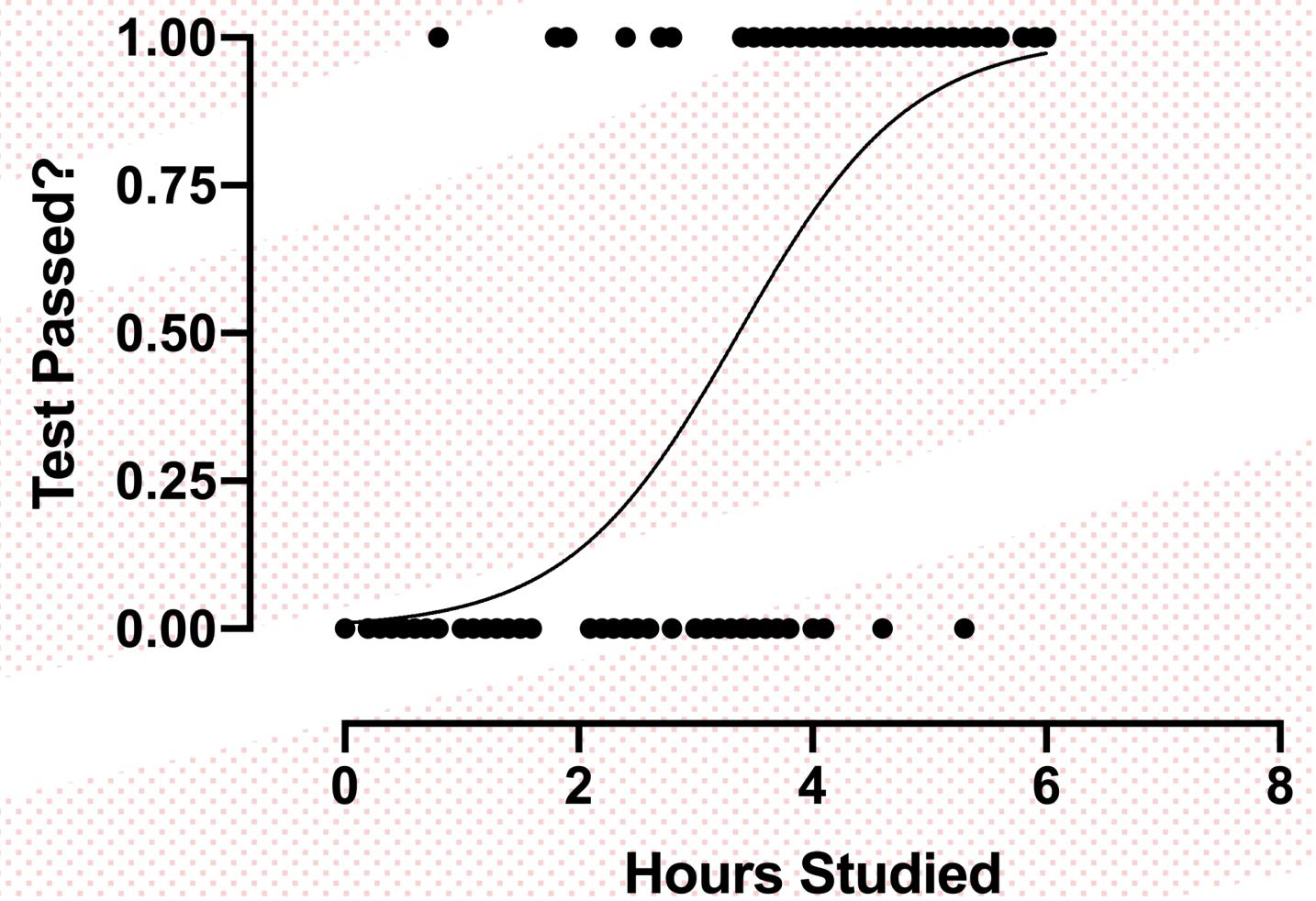
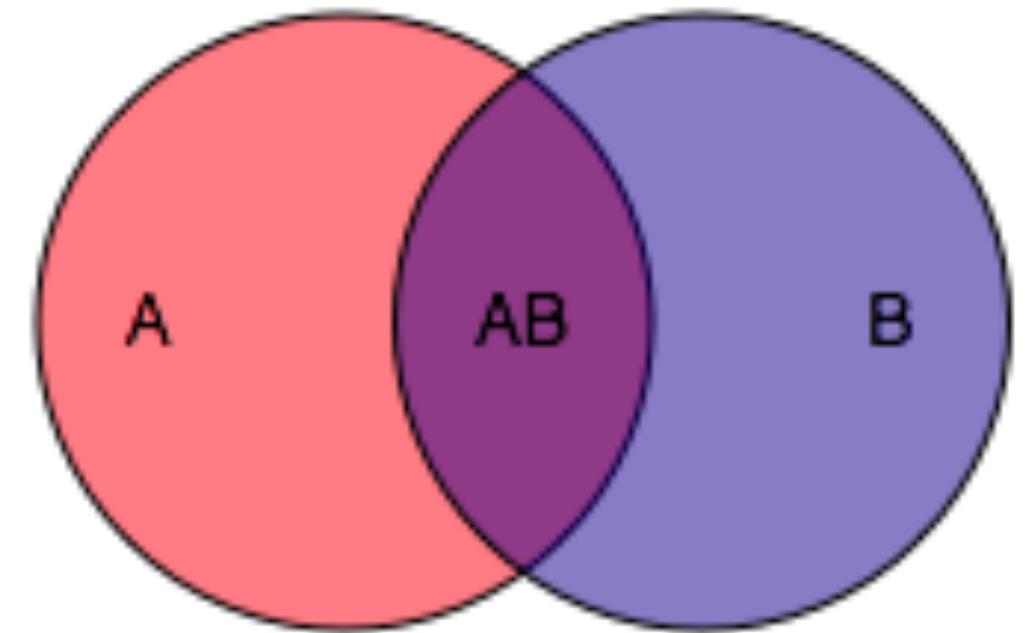
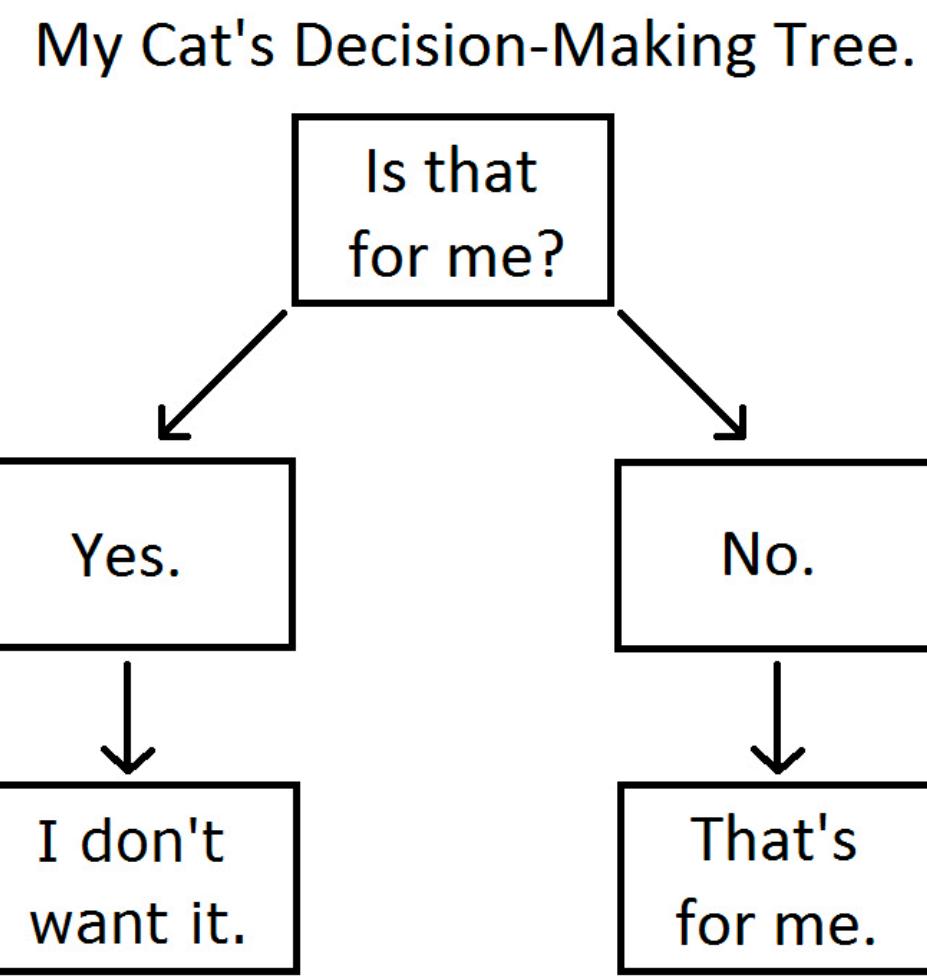
Decision tree based



Probabilistic



# Other model types



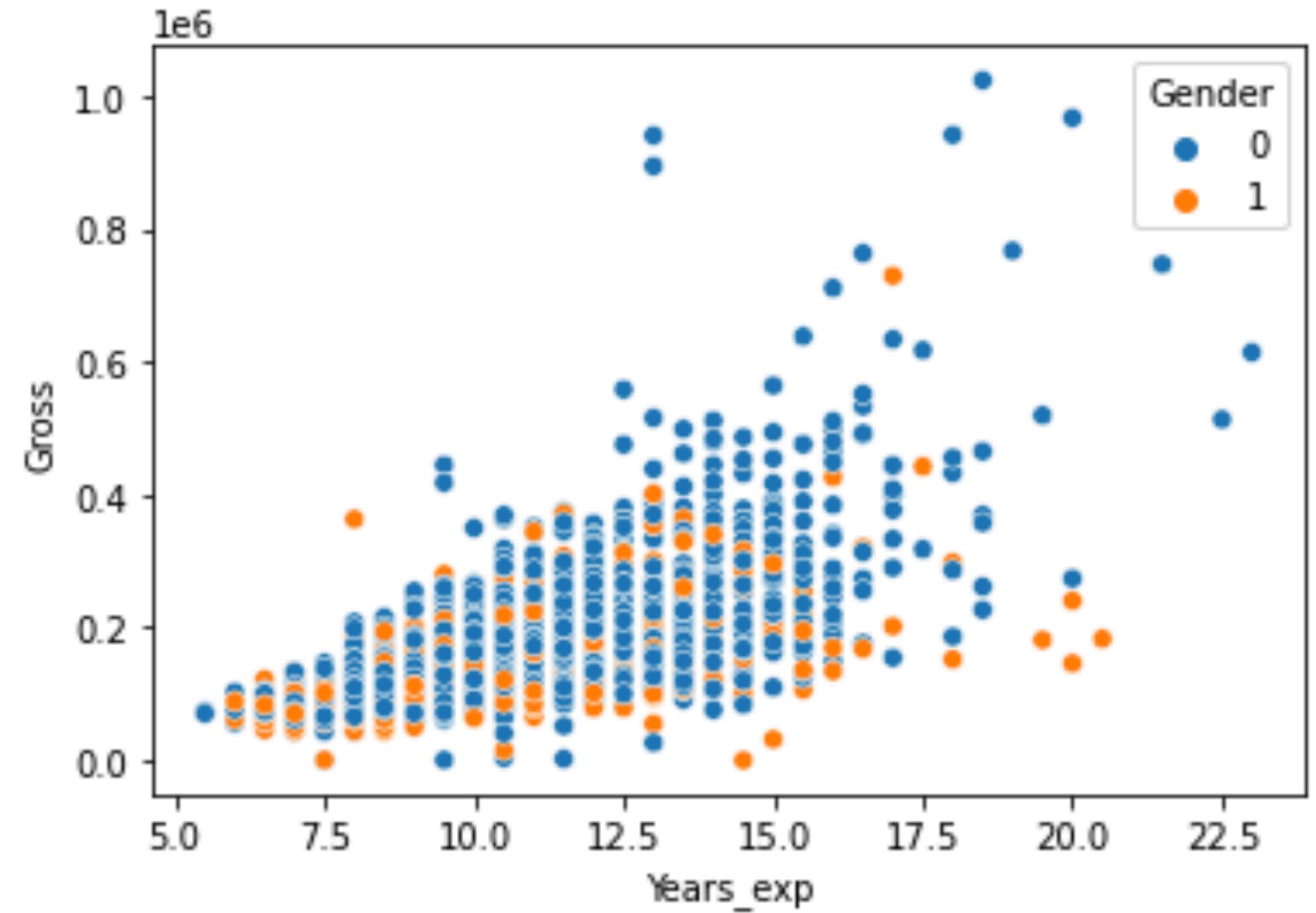
Decision tree based

Probabilistic

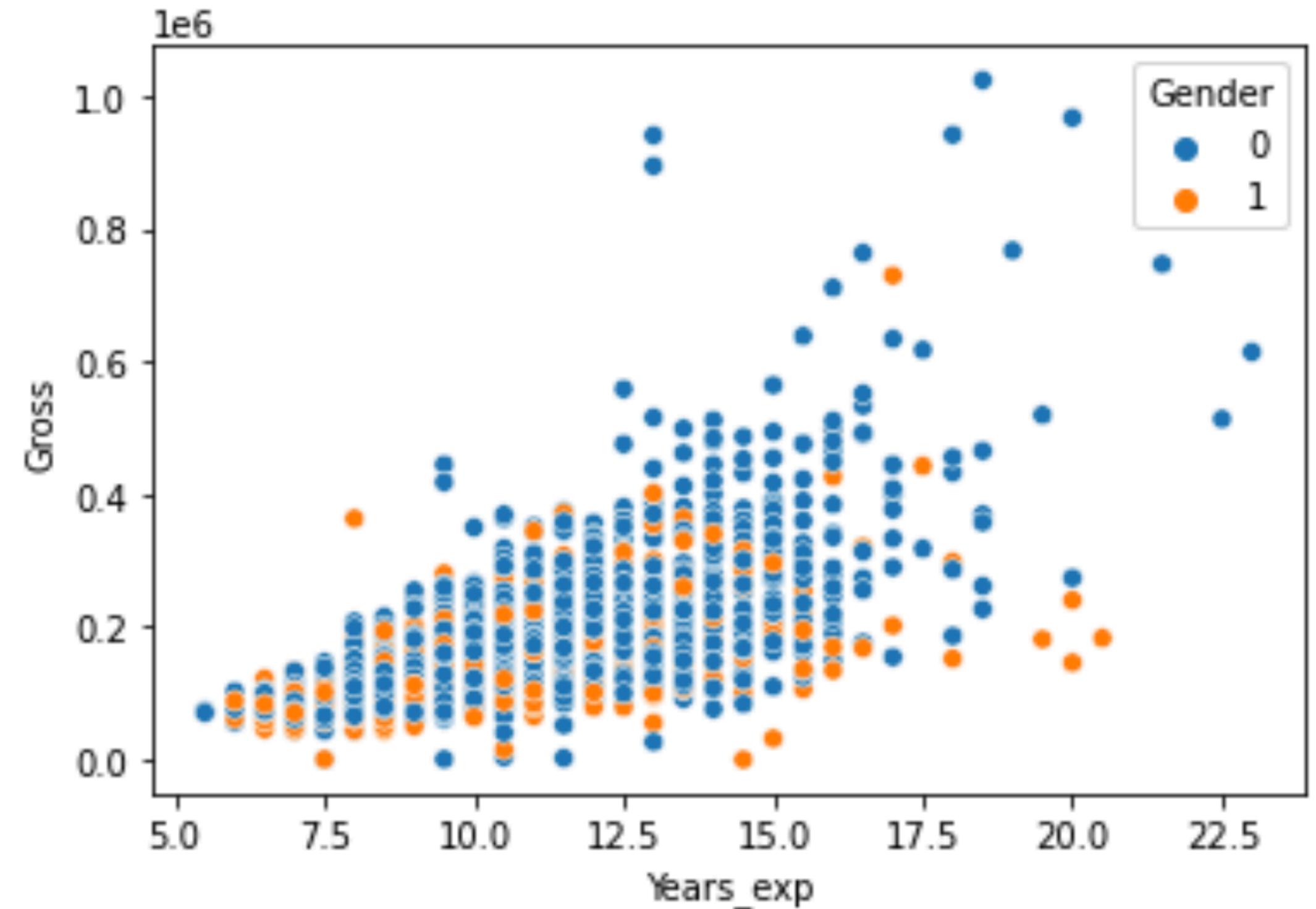
Logistic



# Balance and bias



# Balance and bias



Reality or poor data?



# Making a product



# Making a product



A screenshot of a Jupyter Notebook interface titled "Python Visualization Libraries". The notebook has two cells:

**In [1]:**

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

**In [2]:**

```
np.random.seed(0)
mu = 200
sigma = 25
x = np.random.normal(mu, sigma, size=100)

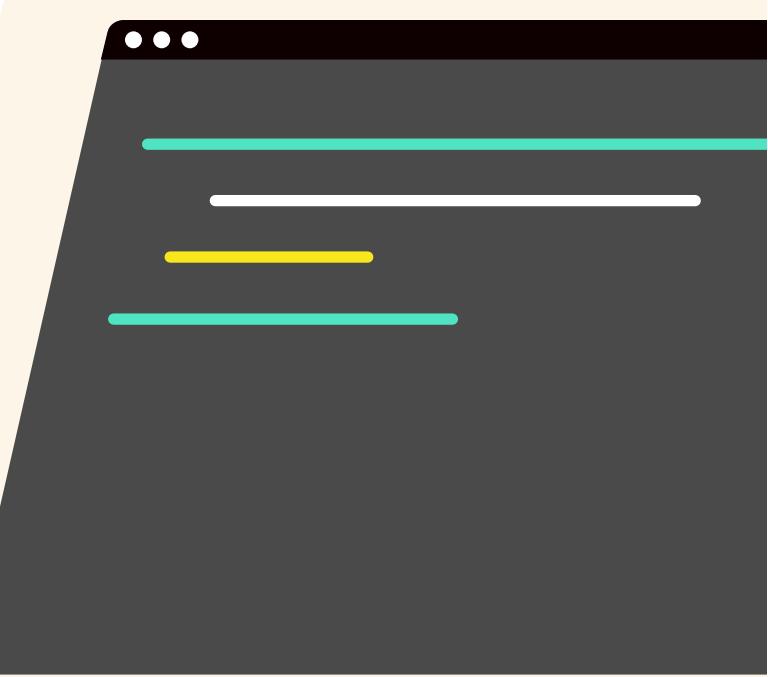
fig, (ax0, ax1) = plt.subplots(nrows=2, figsize=(8, 4))

ax0.hist(x, 20, density=1, histtype='stepfilled', facecolor='g', alpha=0.75)
ax0.set_title('stepfilled')

# Create a histogram by providing the bin edges (unequally spaced).
bins = [100, 150, 180, 195, 205, 220, 250, 300]
ax1.hist(x, bins, density=1, histtype='bar', rwidth=0.8)
ax1.set_title('unequal bins')

fig.tight_layout()
plt.show()
```

The notebook displays two histograms. The first histogram, titled "stepfilled", shows a normal distribution of data points with green bars. The second histogram, titled "unequal bins", shows the same data points with blue bars, using non-uniform bin widths.



# Making a product

RStudio Connect

https://colorado.rstudio.com/rsc/ipython-notebook-visualization/ipyper-static-visualization.html

Python Visualization Libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Matplotlib

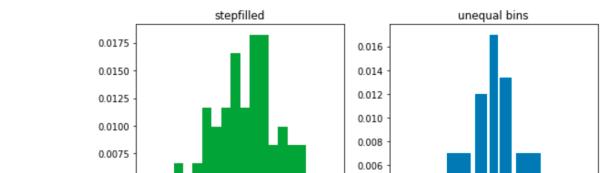
```
In [2]: np.random.seed(0)
mu = 200
sigma = 25
x = np.random.normal(mu, sigma, size=100)

fig, (ax0, ax1) = plt.subplots(nrows=2, figsize=(8, 4))

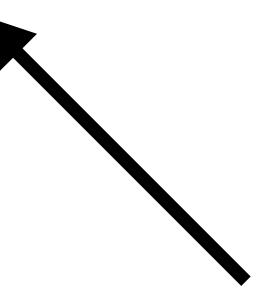
ax0.hist(x, 20, density=1, histtype='stepfilled', facecolor='g', alpha=0.75)
ax0.set_title('stepfilled')

# Create a histogram by providing the bin edges (unequally spaced).
bins = [100, 150, 180, 195, 205, 220, 250, 300]
ax1.hist(x, bins, density=1, histtype='bar', rwidth=0.8)
ax1.set_title('unequal bins')

fig.tight_layout()
plt.show()
```



# Making a product

A screenshot of an RStudio Connect browser window. The URL in the address bar is https://colorado.rstudio.com/rsc/jupyter-notebook-visualization/jupyter-static-visualization.html. The main content area displays a Jupyter notebook cell titled "Python Visualization Libraries". The code in the cell creates two histograms using Matplotlib. The first histogram is labeled "stepfilled" and the second is labeled "unequal bins".

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

In [2]: np.random.seed(0)
mu = 200
sigma = 25
x = np.random.normal(mu, sigma, size=100)

fig, (ax0, ax1) = plt.subplots(nrows=2, figsize=(8, 4))
ax0.hist(x, 20, density=1, histtype='stepfilled', facecolor='g', alpha=0.75)
ax0.set_title('stepfilled')

# Create a histogram by providing the bin edges (unequally spaced).
bins = [100, 150, 180, 195, 205, 220, 250, 300]
ax1.hist(x, bins, density=1, histtype='bar', rwidth=0.8)
ax1.set_title('unequal bins')

fig.tight_layout()
plt.show()
```



You now have the tools

**Your turn!** 



# Thank you!



Want to go further? Head over to

**Le Wagon!**



**le wagon**