Deep Learning Bootcamp

2026

GSA

IGP GSC

GRADUATE STUDENT CLUB

CCDS
GRADUATE STUDENT CLUB

GRADUATE STUDENT CLUB

SPMS
Graduate Student Club

NTU
EEE
GSC

Agenda

1. Introduction to Python
2. Basic numerical libraries for ML

Break 15:30

1. Linear Regression
2. Exploratory data analysis
3. Linear regression with PyTorch (deep learning library)

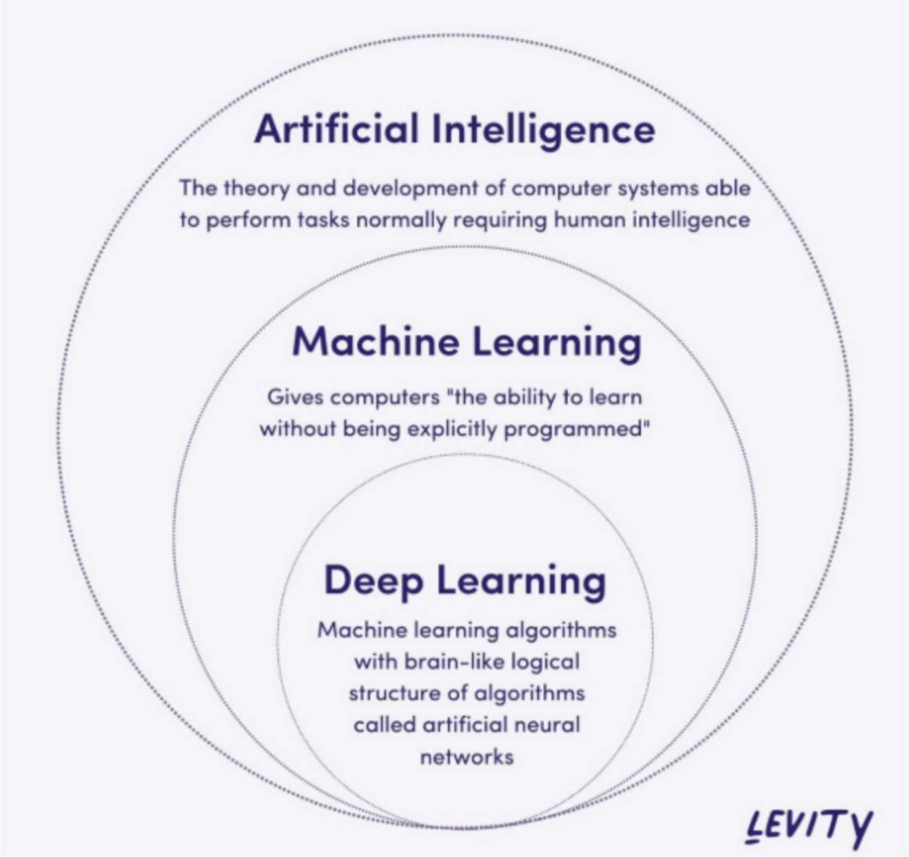
Schedule

Event No.	Details
Event 1 **24th Oct 2025**	*Deep Learning Essentials* Covers the basics of Python and necessary packages required for Deep Learning such as numpy, scipy, pandas etc.
Event 2 **10th Nov 2025**	*Deep Learning for Regression and Classification* Will cover the basics of PyTorch, as well as how to use PyTorch for performing regression and classification tasks.
Event 3 **17th Nov 2025**	*Deep Learning for Images* In this event, we will extend the classification using deep learning, specifically focusing on datasets involving images.
Event 4 **TBD – Sem 2 – 2026**	*Deep Learning for Sequence Data (text and time series)* In this event, we will focus on using Deep Learning models for datasets involving sequences or temporal relations. We plan to cover examples from both text and time-series datasets.
Event 5 **TBD – Sem 2 – 2026**	*Reinforcement Learning* This session will introduce Deep Reinforcement Learning techniques with some practical applications.

What is Deep Learning?

- Machine learning: learn from examples or data
- Deep Learning: use artificial neural networks: weight * variable + some non-linear layer

$$y = \sigma\left(W_L \ldots \sigma\left(W_2 \sigma(W_1 x)\right)\right)$$



Artificial Intelligence

The theory and development of computer systems able to perform tasks normally requiring human intelligence

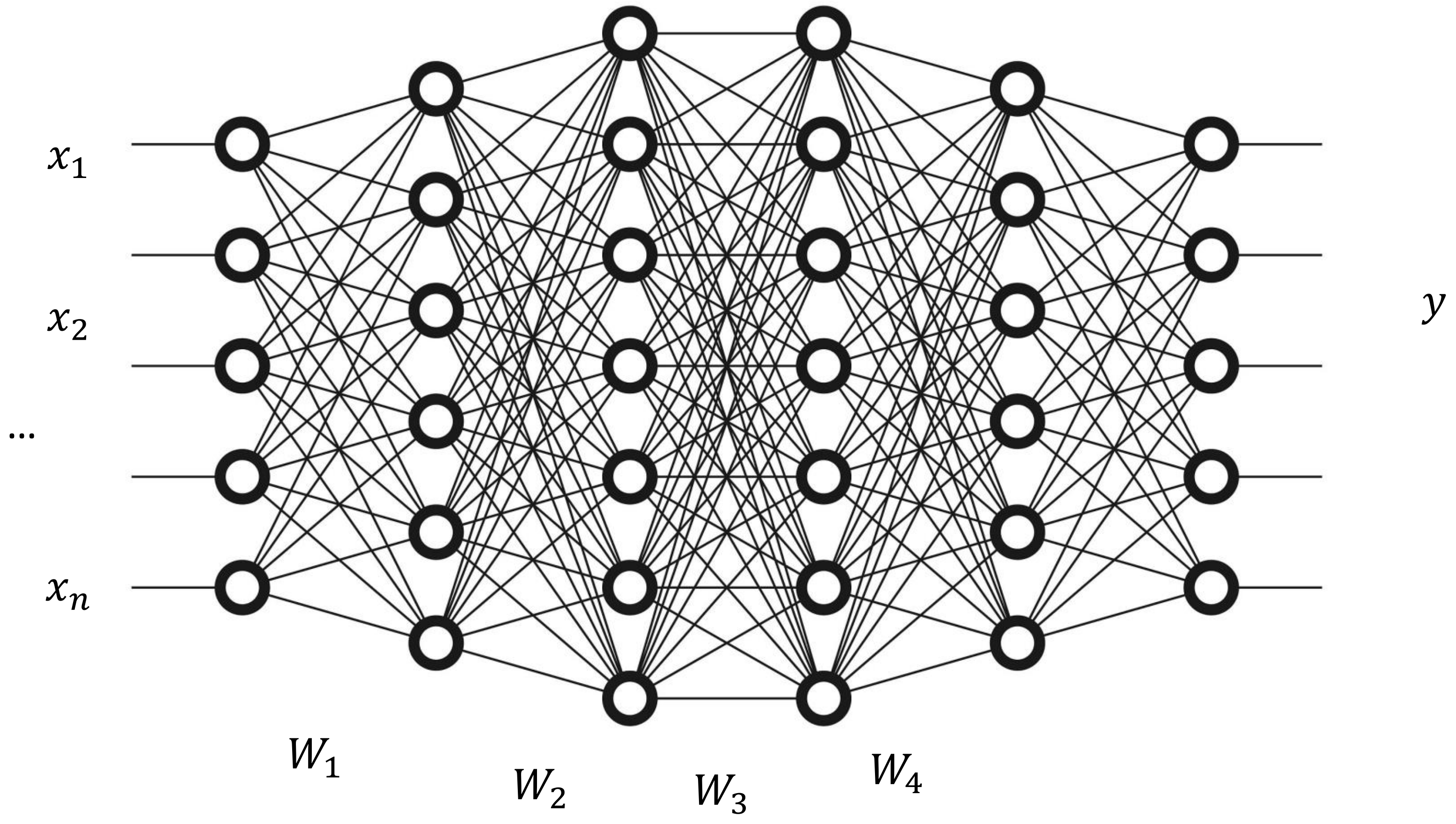
Machine Learning

Gives computers "the ability to learn without being explicitly programmed"

Deep Learning

Machine learning algorithms with brain-like logical structure of algorithms called artificial neural networks

Deep Learning is powered by deep neural networks



Deep as in depth

i.e., many layers

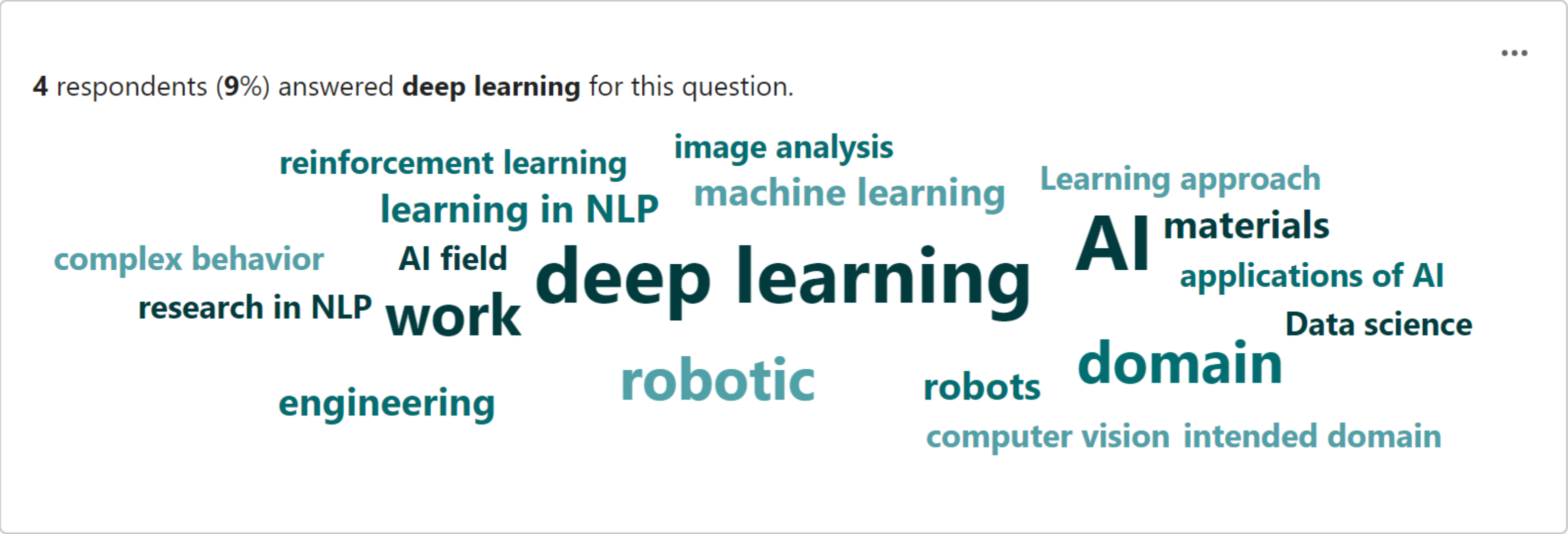
https://cdn-images-1.medium.com/max/2400/1*1mpE6fsq5LNxH31xeTWi5w.jpeg

Why Deep Learning?

- Find hidden patterns in a lot of data (high-dimensional, complex, etc.)
- Many applications, open models
  - Material properties prediction
  - Protein folding: see AlphaFold
  - Text generation/Natural Language Processing: ChatGPT
  - Image generation: DALLE, MidJourney, etc.
  - Computer Vision: ImageNet, image segmentation, detect face in phone, etc.
  - …

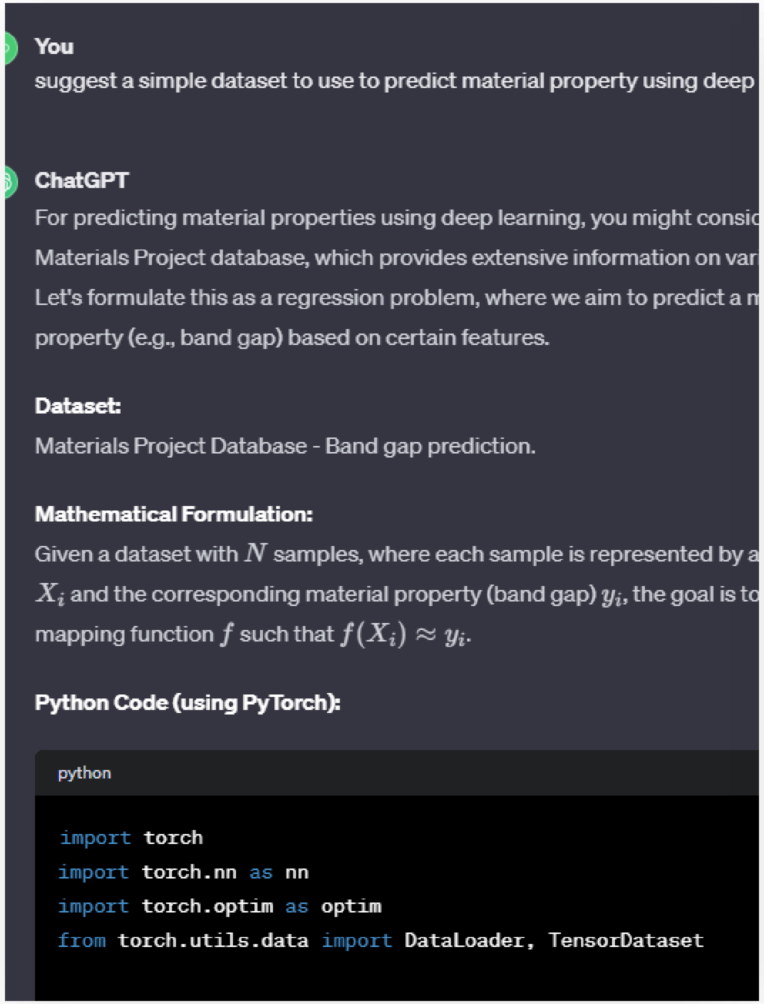
Deep Fakes…

Your interest



4 respondents (**9**%) answered **deep learning** for this question.

Examples


You

suggest a simple dataset to use to predict material property using deep

ChatGPT

For predicting material properties using deep learning, you might consi
Materials Project database, which provides extensive information on var
Let's formulate this as a regression problem, where we aim to predict a
property (e.g., band gap) based on certain features.

Dataset:

Materials Project Database - Band gap prediction.

Mathematical Formulation:

Given a dataset with N samples, where each sample is represented by a
X_i and the corresponding material property (band gap) y_i, the goal is to
mapping function f such that $f(X_i) \approx y_i$.

Python Code (using PyTorch):

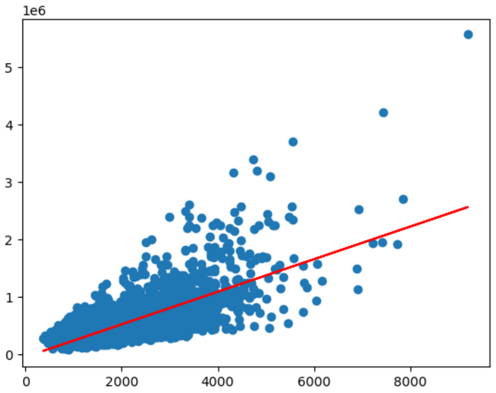
python

```python
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, TensorDataset
```

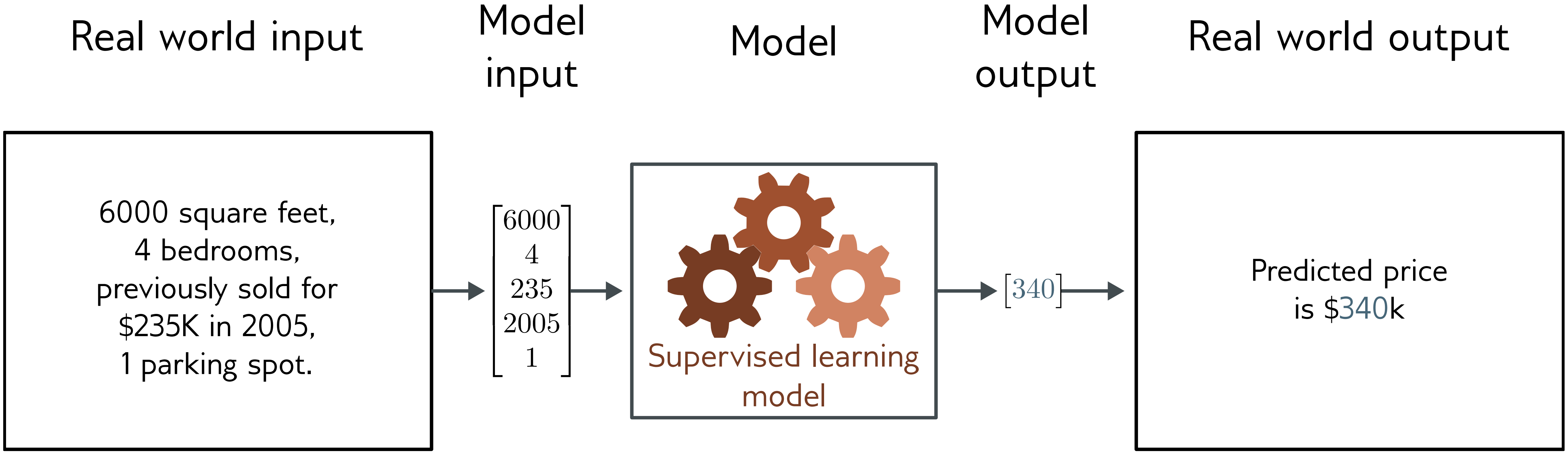
ML 101

- Approximate a function $y = f(x;\theta)$
  - x: image pixels; y: classes (dog, cat)
  - x: house size; y: house price
  - θ are the parameters

- Two typical problems
  - Classification: discrete categories
  - Regression: continuous categories



Regression



Real world input

Model input

Model

Model output

Real world output

6000 square feet,
4 bedrooms,
previously sold for
$235K in 2005,
1 parking spot.

$$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$$

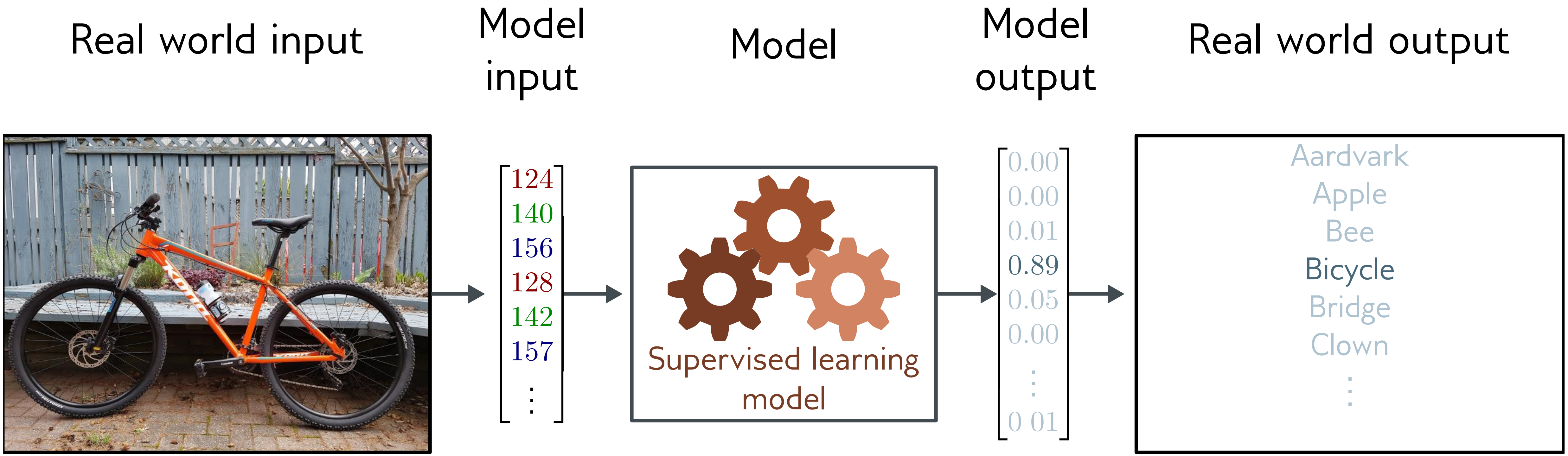
Supervised learning model

$[340]$

Predicted price
is $340k

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

Classification



Real world input

Model input

Model

Model output

Real world output

$$\begin{bmatrix} 124 \\ 140 \\ 156 \\ 128 \\ 142 \\ 157 \\ \vdots \end{bmatrix}$$

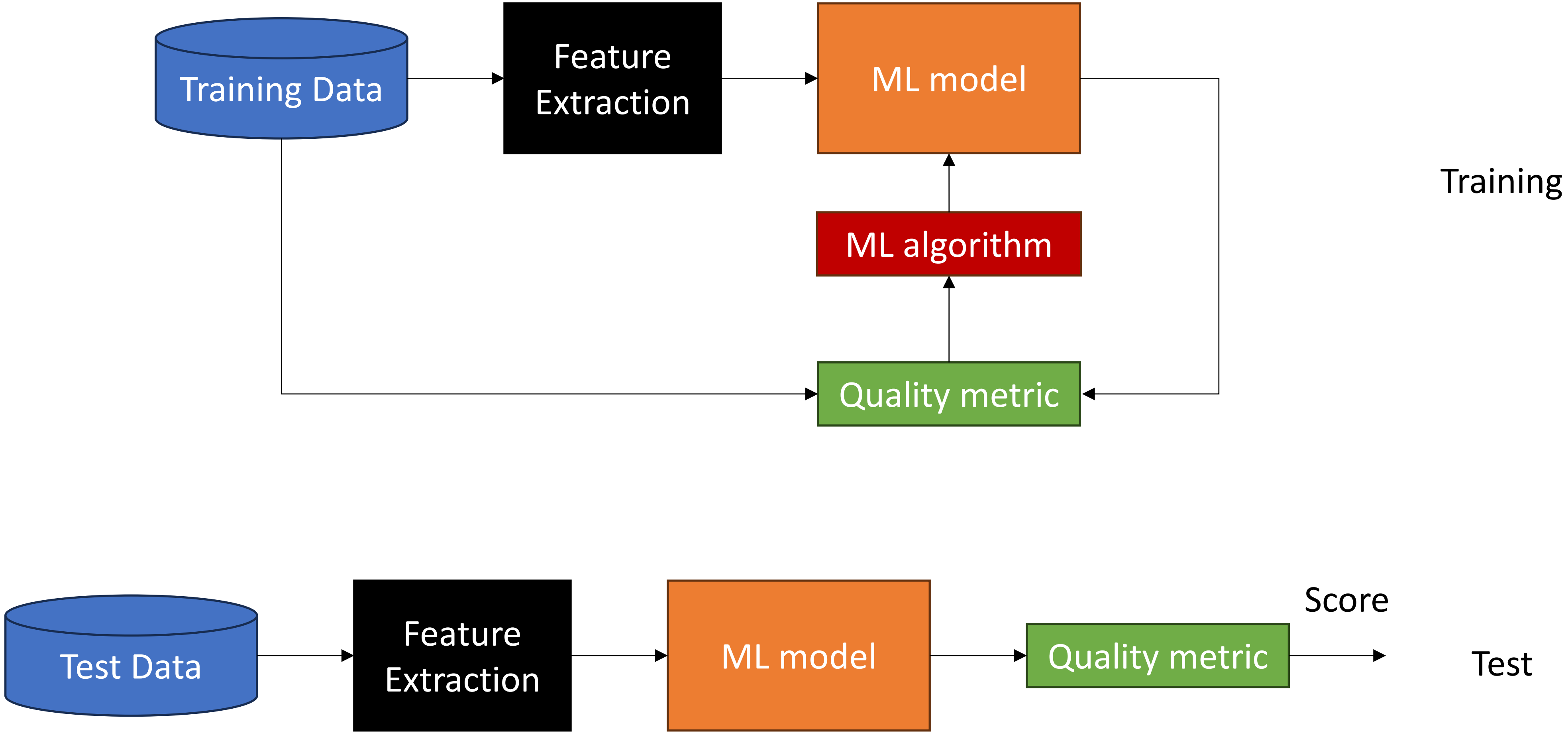
Supervised learning model

$$\begin{bmatrix} 0.00 \\ 0.00 \\ 0.01 \\ 0.89 \\ 0.05 \\ 0.00 \\ \vdots \\ 0.01 \end{bmatrix}$$

Aardvark
Apple
Bee
Bicycle
Bridge
Clown
⋮

- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

ML workflow



https://www.coursera.org/learn/ml-foundations?specialization=machine-learning

Feature Extraction 101

- Machine learning needs numerical data as input:

$$\mathrm{x} \in \mathbb{R}^n$$

- Data normalization: assumption of data with mean 0, stddev 1

$$x' = \frac{x - \mu}{\sigma}$$

```
$ x_std = StandardScaler().fit_transform(x)
```

- Typical representations

Data Type	**Common Representation**
Images	Pixel intensity arrays
Text	Word embeddings (Word2Vec, GloVe, BERT)
Audio	Spectrogram or MFCC features

Model Selection 101

ChatGPT for everything   XGBoost

- **Many ML models to choose from:**
    - **Linear models:** simple, interpretable, good baseline
    - **Decision Trees / XGBoost:** handle complex patterns, less linear assumptions
    - **Neural Networks:** powerful but need large data and tuning

- **Not every problem needs Deep Learning!**
    - NNs can overfit on small datasets
    - Many hyperparameters, hard to tune
    - Require more computation and time

- **Model selection = experimentation**
    - Try multiple models
    - Tune hyperparameters
    - Compare results using validation metrics (e.g., accuracy, RMSE, F1)
    - Pick the model that’s **accurate, simple, and practical**

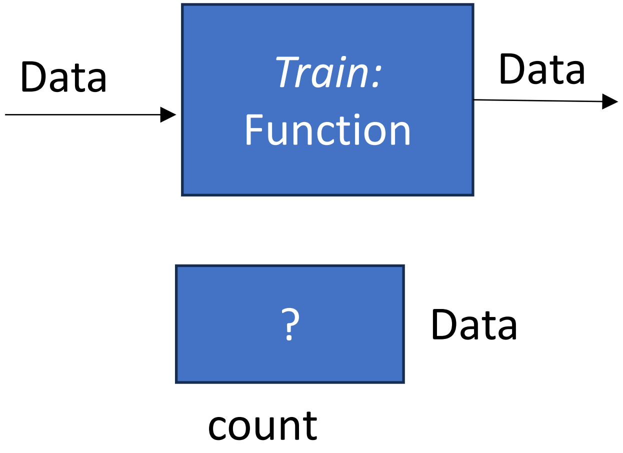
Pro & Cons of DL

Pro	Cons
• Can learn features & representations easily • Ability to process a lot of data • Flexible framework • Maps well into parallel hardware (GPU and others)	• Hard to understand and build intuition of why the model works: explainability and interpretability • Requires a lot of data and expensive hardware.

Python 101: Why Python?

- A lot of libraries for ML/DL: PyTorch, scikit-learn, pandas, numpy
- Easy to learn, simple syntax
- Interactive notebooks: Jupyter Notebook, Kaggle, Google Collab
- Free & open-source

Python 101: Basic features



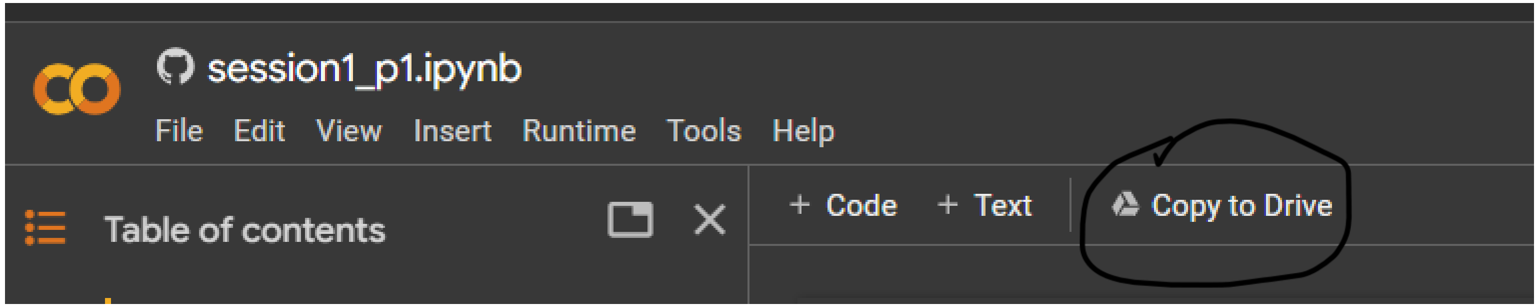
```
def vowel_count(word):
    vowels = ["a", "e", "i", "o", "u"]
    count = 0
    for char in word: # loops
        if char in vowels: # conditions
            count += 1
    return count

vowel_count("hello")
```

MyModel: Class

my_model: Data1
params: Data2
…

Train: Function1
Test: Function2
…

Google Colab: Our tool for today



Can you open the notebooks?

Work session 1.0

Handling numbers & data: numpy

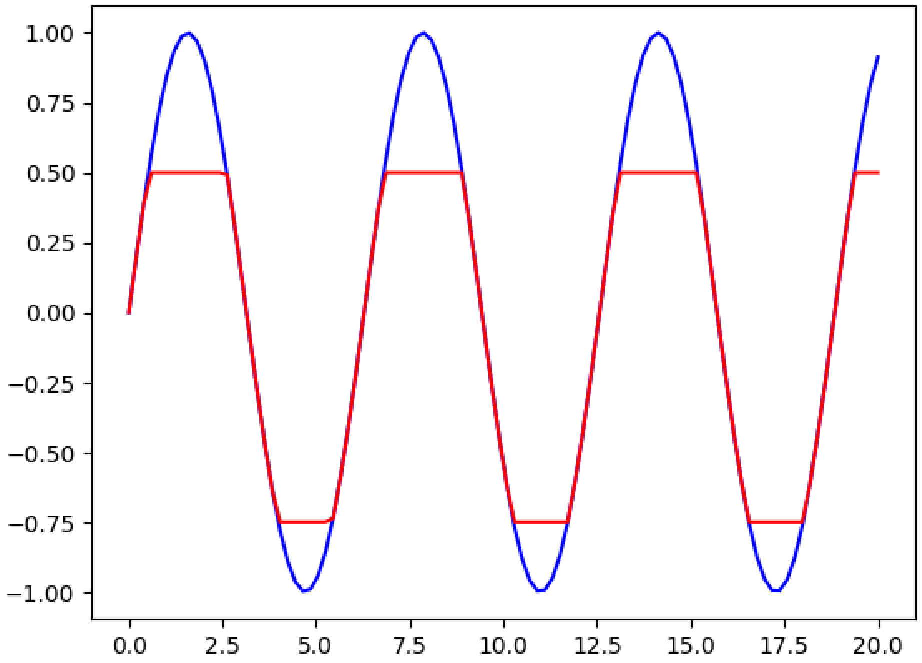
- Numpy is library for handling array, vector & matrices

- Provide all keys operations in arrays
  - Creation : *a = np.array([[1, 2, 3], [4, 5, 6]])*
  - Add/subtract/multiply: *a + b, a – b, a * b, …*
  - Dot product and matrix multiplication: *a @ b*
  - Slicing: *a[1]=> [4,5,6]*

Plotting: matplotlib

- Matplotlib is the Python library used to plot data

- You can create easily different types of plots (scatter plot, histogram, X-Y plot), add legend and different details.

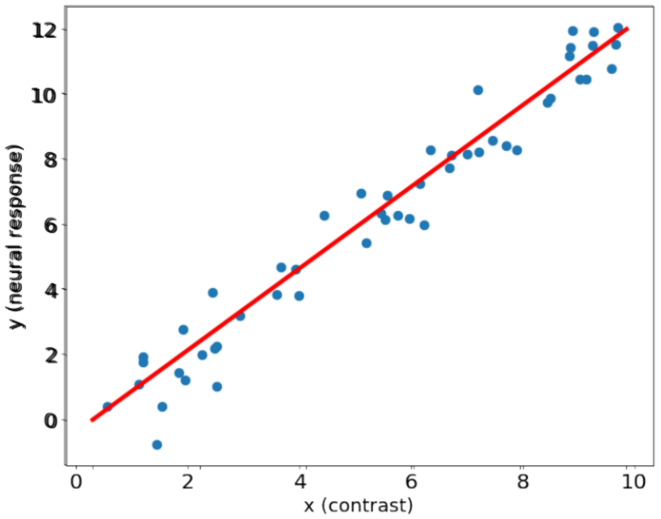


Work Session 1.1

Linear Regression

Linear regression makes predictions about the linear relationship between the input variable x (contrast) and the output variable y (neural response).

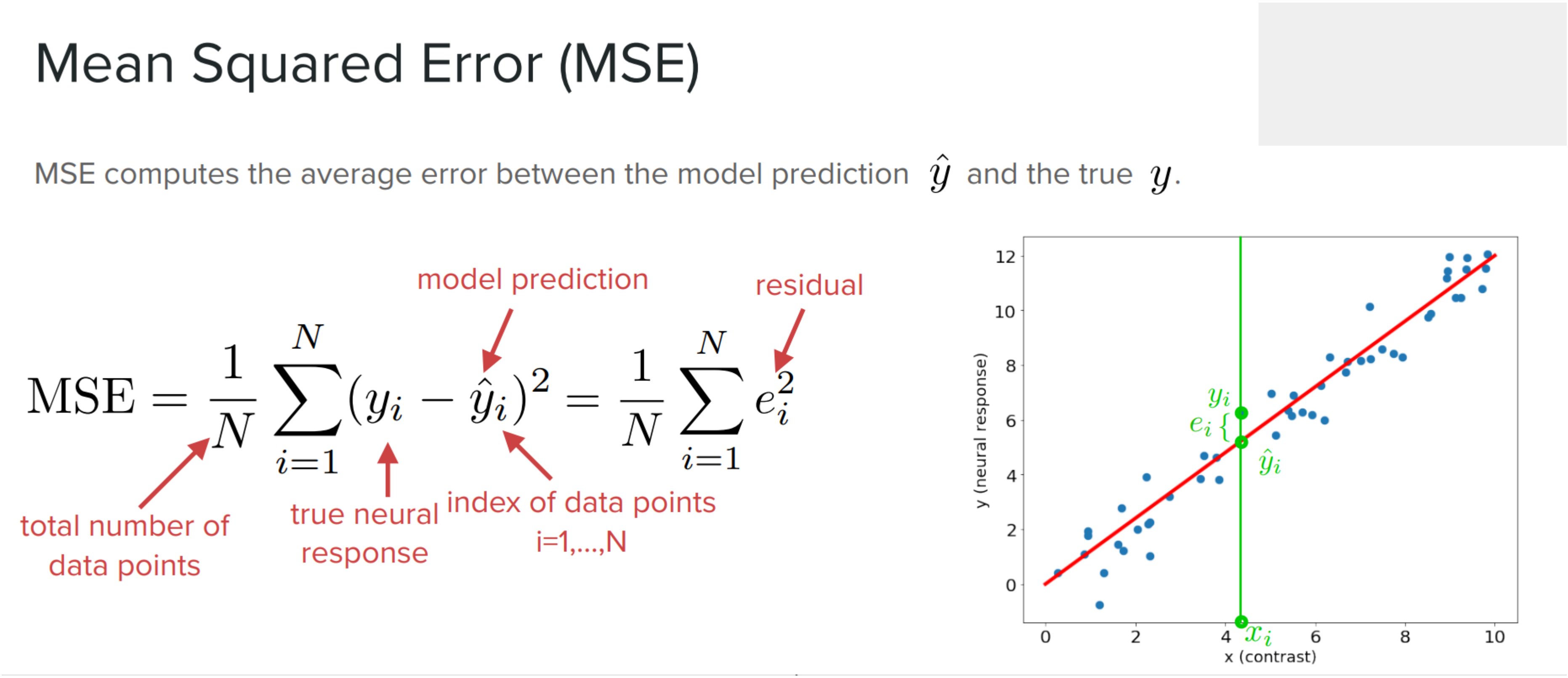
$$\underset{\text{neural response}}{\underset{\uparrow}{y}} = \underset{\text{linear weight}}{\underset{\uparrow}{\theta_1}} \times \underset{\text{contrast}}{\underset{\uparrow}{x}} + \underset{\text{Intercept}}{\underset{\uparrow}{\theta_0}}$$



We are not considering the intercept for simplicity, resulting in a one-parameter model.

Linear Regression: MSE

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (y_i - \theta x_i)^2$$



Mean Squared Error (MSE)

MSE computes the average error between the model prediction $\hat{y}$ and the true y.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \frac{1}{N} \sum_{i=1}^{N} e_i^2$$

model prediction

residual

total number of data points

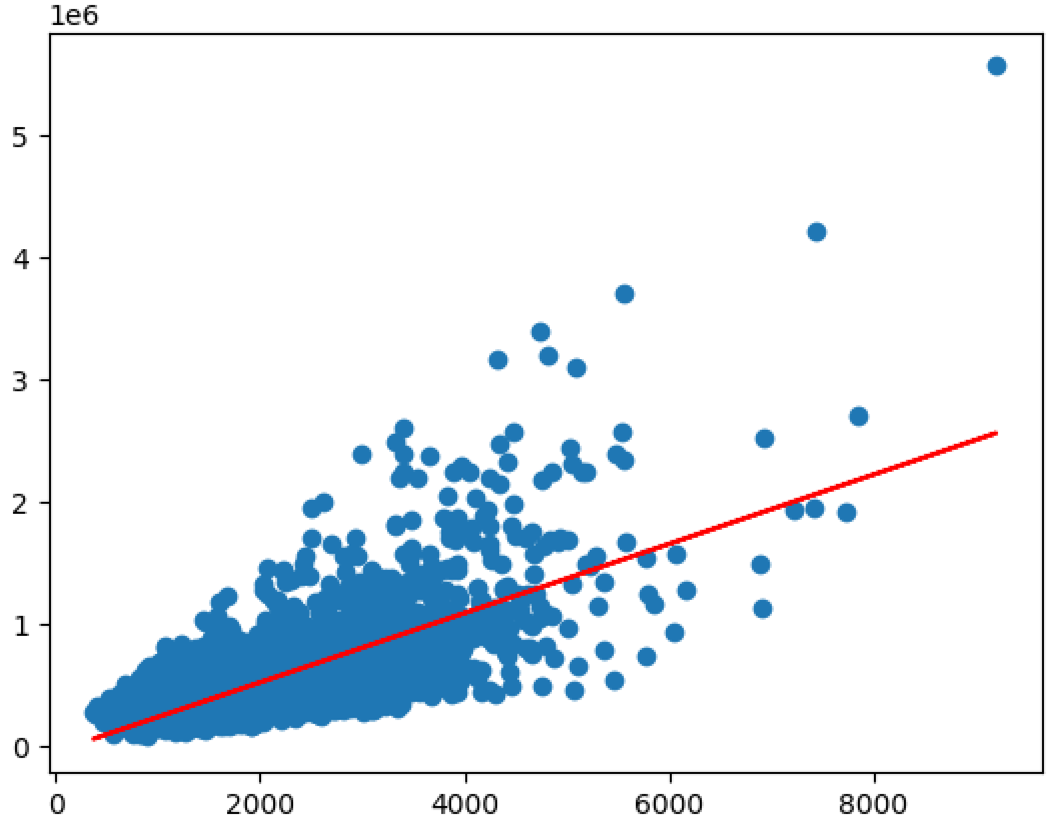
true neural response

index of data points i=1,...,N

Work session 2

Housing data exploration

- Let’s explore a housing data
- Relevant plots
- Linear regression model
  - Scikit-learn
  - PyTorch



Deep Learning & Pytorch

- Pytorch one of the most popular libraries for deep learning

- Components of a DL model:
    - The model itself: the structure of the model
    - A loss function (error metric)
    - An optimization algorithm
    - The training loop

Yasharth to add stuff model eval

- Mae / mse => regression

- Cross entropy => classification
  - Also things like FP / FN etc

- Confusion matrix

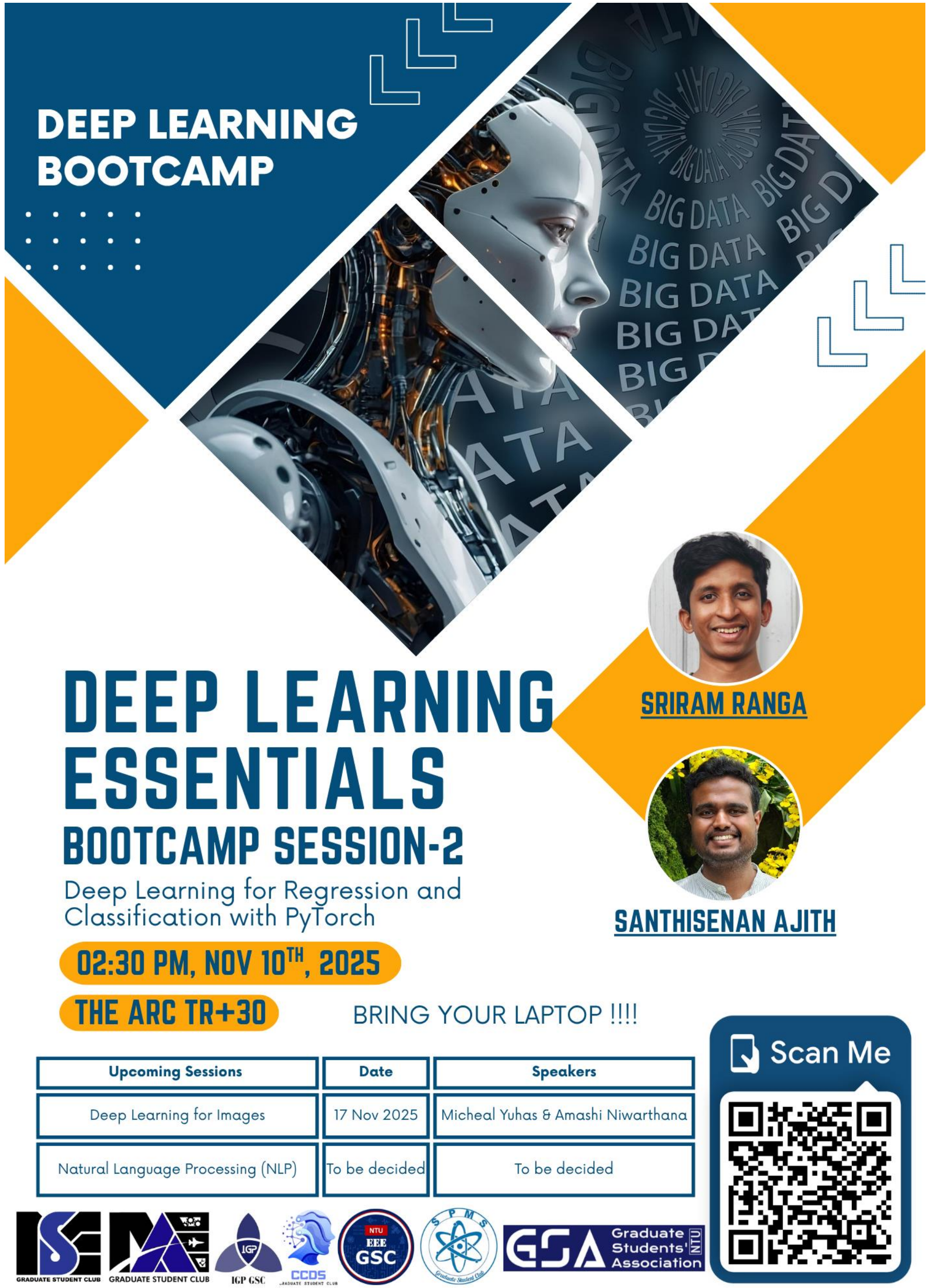
References / Reading

- Python introduction: https://swcarpentry.github.io/python-novice-inflammation/
- More on scientific python: https://lectures.scientific-python.org/
- https://deeplearning.neuromatch.io/
- NeuroMatch Academy: https://deeplearning.neuromatch.io/tutorials/W1D1_BasicsAndPytorch/chapter_title.html
- Exploratory computing w/ Python: https://mbakker7.github.io/exploratory_computing_with_python/
- https://udlbook.github.io/udlbook/
  - Reuse slides from there

Please provide your feedback!

3rd DL Bootcamp (2025-26) - Session 1 - Post Session Feedback

See you in the next session



DEEP LEARNING BOOTCAMP

DEEP LEARNING ESSENTIALS

BOOTCAMP SESSION-2

Deep Learning for Regression and Classification with PyTorch

SRIRAM RANGA

SANTHISENAN AJITH

02:30 PM, NOV 10TH, 2025

THE ARC TR+30

BRING YOUR LAPTOP !!!!

Upcoming Sessions	Date	Speakers
Deep Learning for Images	17 Nov 2025	Micheal Yuhas & Amashi Niwarthana
Natural Language Processing (NLP)	To be decided	To be decided

Scan Me