Foundations Of Neural Networks and Deep Learning

Intro Class

SME: Gagan P

Contact: gaganp000999@gmail.com

what is this course about?

- **AI** is the **buzz** word right now, every body wants to learn ML and DL fast, but directly jump into libraries like **TensorFlow/PyTorch**
- These libraries are easier to learn but they come with a **hidden cost** for beginners
- You can write a **Neural Network** in just **5 lines** of code but you dont even know what's happening behind the scenes
- As a **beginner** knowing the **fundamentals** and implementing everything from scratch is the key
- This course makes sure that you are fundamentally strong in all the concepts and math needed to understand almost all the major concepts in AI such as Computer Vision, LLM's, etc
- Once we implement everything from scratch we will then move onto **PyTorch**
- We will then finish the course by building **OUR OWN LIBRARY FOR IMAGE PROCESSING!!!!**

timeline of the course

1

Week 1

- Brush up Basic Mathematics and Python.
- Intro to Numpy and Matplotlib
- Intro to Basic ML algorithms
- Build projects on Linear and Logistic regression

2

Week 2

- Neural network intro
- PyTorch basics
- backpropagation training techniques.

3

Week 3

- FCNN from scratch implementation
- Advanced neural network regularization techniques

4

Week 4

- CNN introduction
- Advanced architectures, transfer learning
- Final project presentations.

Expected outcome from this course

- Extremely **strong fundamentals**
- First principles approach to build anything from scratch
- Strong hold on concepts and truly understand them
- Have a base knowledge upon which you can pivot to any field in AI be it **Computer Vision, Reinforcement Learning, Gen AI, etc**
- Have a **portfolio worthy project** in the end

what you should expect from me

- detailed explanation of all the concepts
- explain the complete code and implementation
- books, resources, etc will be shared
- all the **code** and **slides** will be uploaded to a **github** repo
- ask me any question in class, or always available on whatsapp/email
- have a doubt? ask me anytime
- any further **competitions/opportunites in research/ internships** will be shared in the community!

what I expect from you

- this is not a formal class, so **chill!**
- **active participation** in assignments and discussions
- have a doubt? it’s your responsibility to get it clarified(online/offline)
- be honest in your work, this will not affect your **GPA**
- **do not copy-paste** answers for assignments and projects, even if you use AI understand what is happening, it’s fine even if you don’t complete the assignment
- spend **atleast 1 hour** implementing everything by yourself after the classes and maintain a **separate individual github** repo containing all the codes(this will be checked at the end)
- try writing a blog about what you learnt(optional)

lets start today’s class!

what is AI?

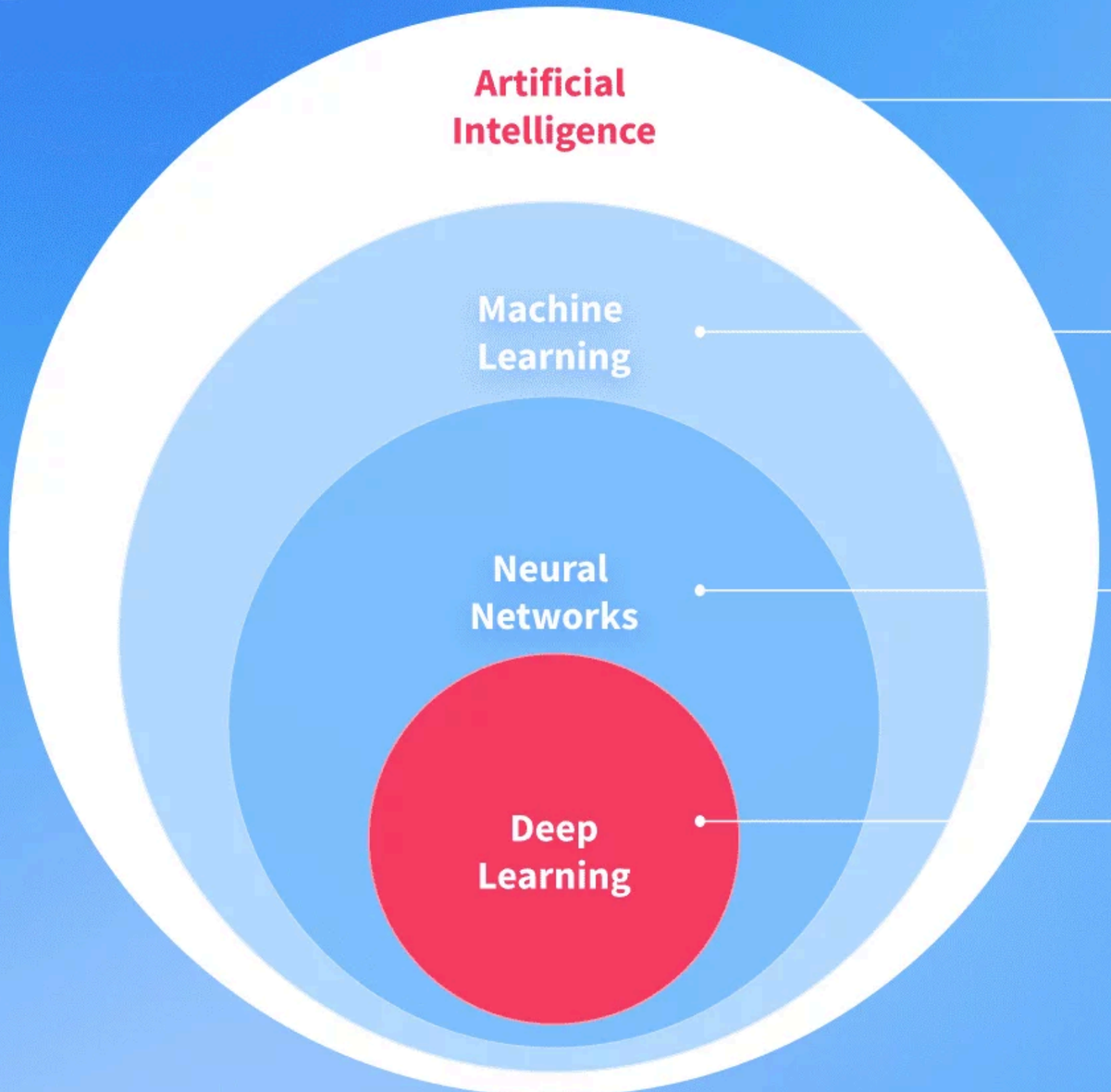
Is AI, Machine Learning, Deep Learning, etc. related?

Artificial Intelligence (AI)

Definition: Computer systems that can perform tasks normally requiring human intelligence
Goal: Simulate human-like thinking, reasoning, and decision-making
Examples:

- Chess-playing computers
- Voice assistants (Siri, Alexa)

How AI, ML, NNs, DL Are Related to Each Other



Allows a system to automatically learn and improve from experience

Allows a system to automatically learn and improve from experience

Allows programs to recognize patterns and solve common problems in AI, ML, and DL.

Allows to use complex algorithms and deep neural nets to train a model.

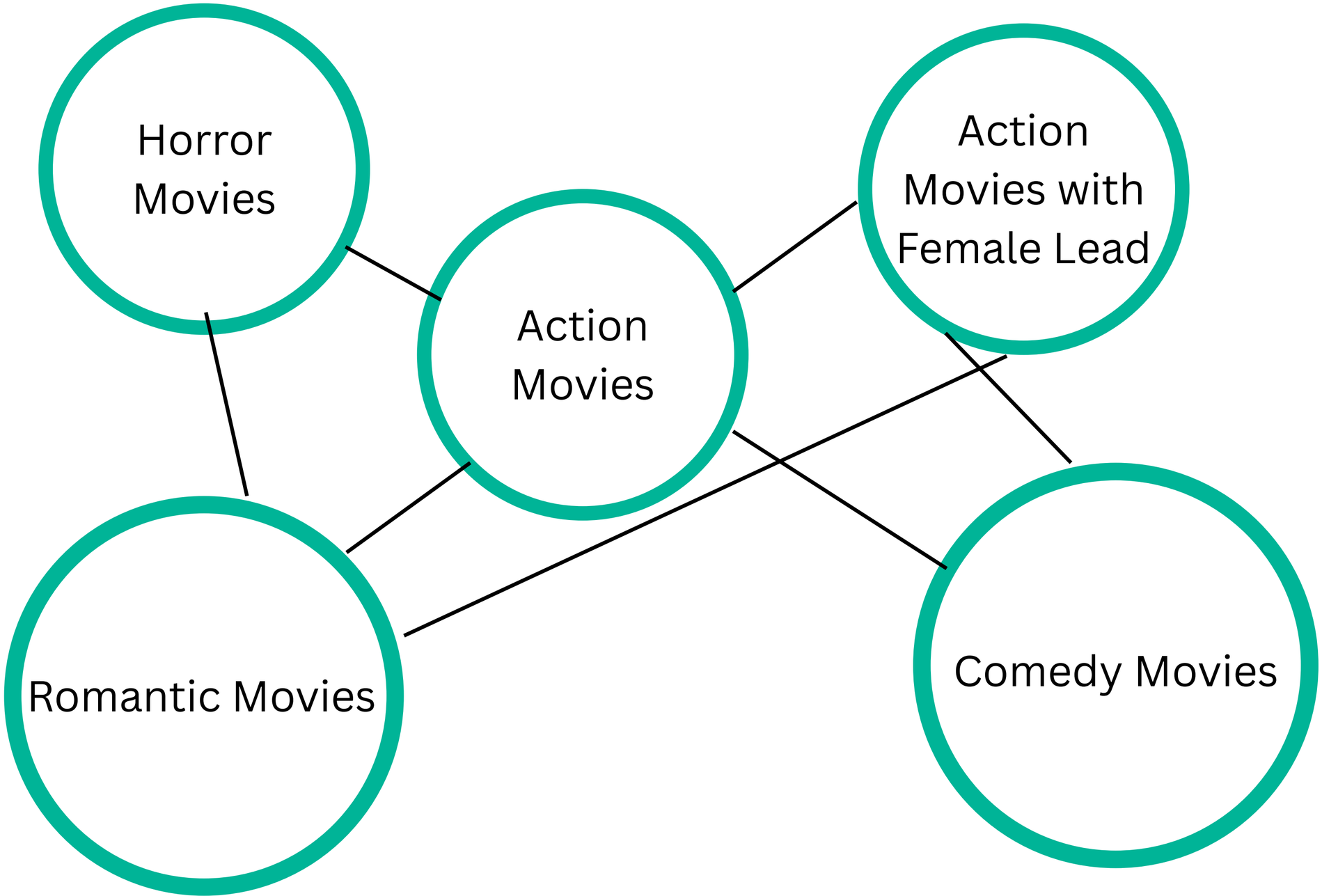
Machine Learning (ML)

Definition: A subset of AI that learns from data without being explicitly programmed for every scenario
Key Feature: Systems improve automatically through experience

Examples:

- Netflix recommendations
- Email spam filters
- Product suggestions
- Fraud detection

How Netflix Recommendation Works



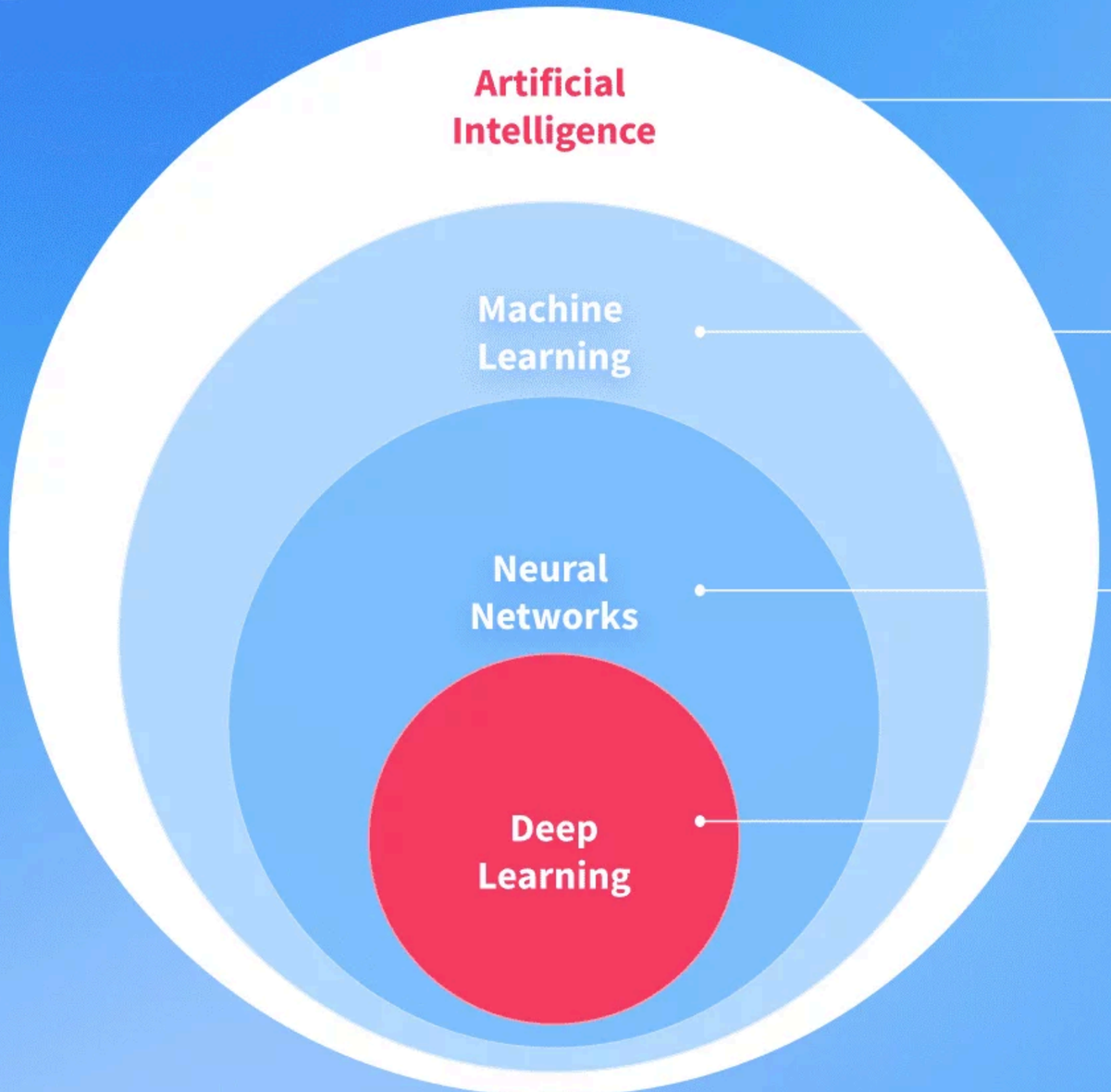
Deep Learning (DL)

Definition: A subset of ML using multi-layered neural networks inspired by the human brain
Key Feature: Processes data through many layers to handle complex patterns

Examples:

- Face recognition
- Self-driving cars
- Voice assistants
- Medical image analysis

How AI, ML, NNs, DL Are Related to Each Other



Allows a system to automatically learn and improve from experience

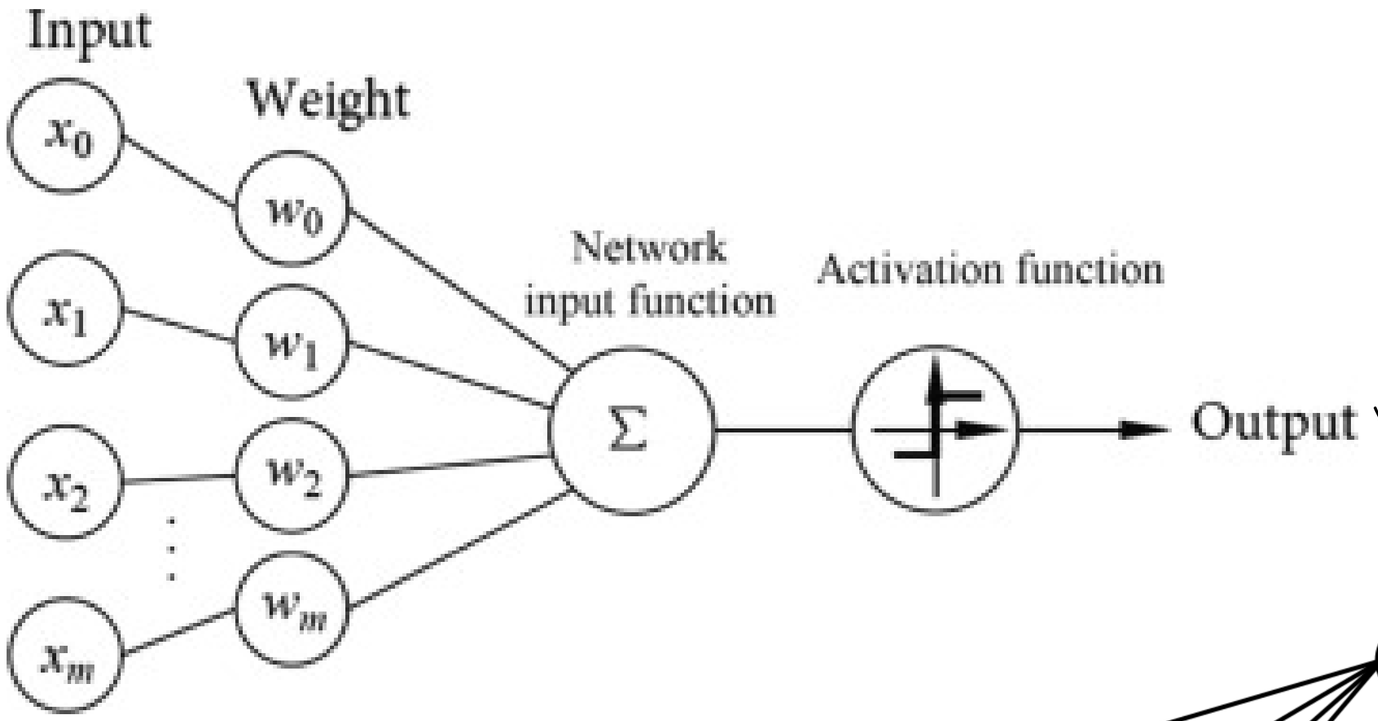
Allows a system to automatically learn and improve from experience

Allows programs to recognize patterns and solve common problems in AI, ML, and DL.

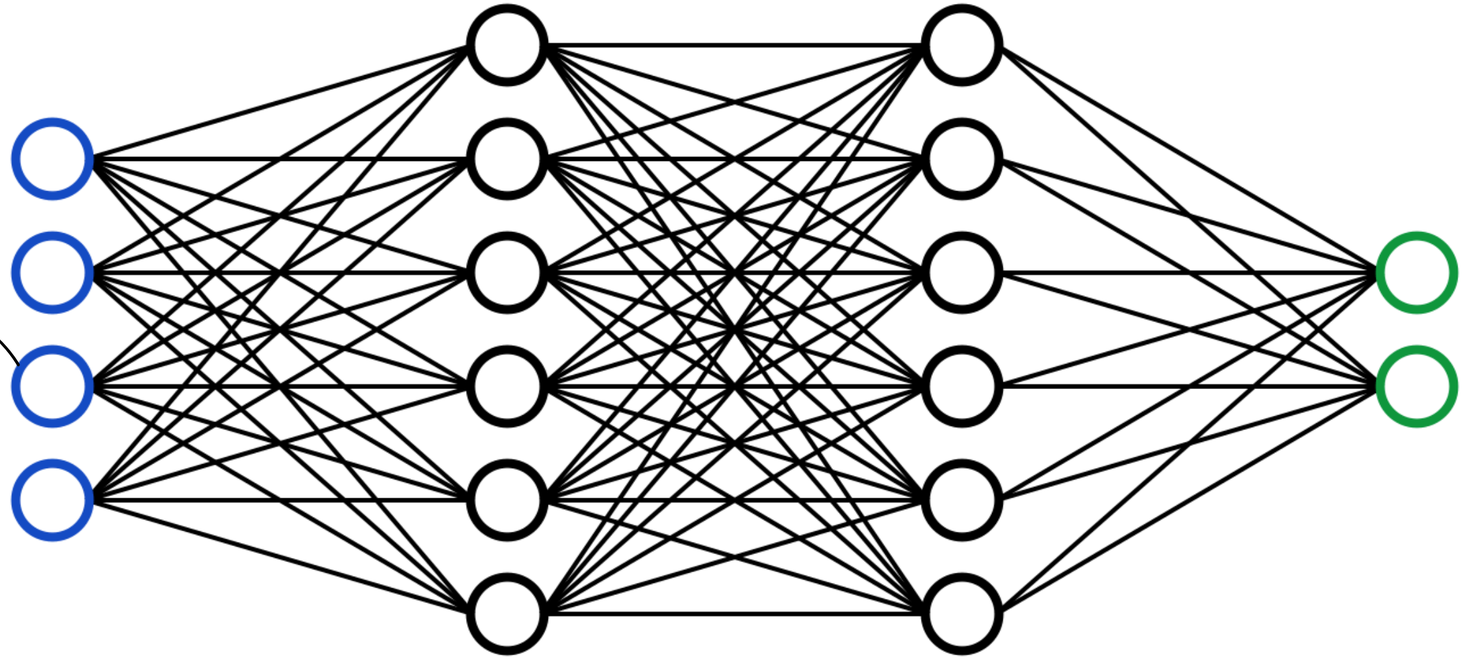
Allows to use complex algorithms and deep neural nets to train a model.

brief history of AI

- **Rule-Based Systems (1940s-50s):** Hard-coded if-then logic. Example: Early medical diagnosis systems.
- **First Learning (1950s):** Arthur Samuel's checkers program improved through play.
- **Perceptrons (1957):** Rosenblatt's trainable networks. Example: Basic character recognition.
- **AI Winter (1970s-80s):** Funding cuts, limited progress. Example: Neural research nearly stopped.
- **Backpropagation (1980s):** LeCun & others enabled multilayer training. Example: NetTalk learned pronunciation.
- **CNNs (1989-98):** Yann LeCun created LeNet for image recognition. Example: Bank check reading (10% of US checks).
- **Data-Driven (1990s):** Shift from knowledge to data-based learning. Example: Deep Blue beat Kasparov.
- **Deep Learning (2006-12):** Hinton's deep networks enabled breakthroughs. Example: ImageNet competition victories.
- **Transformers (2017):** **"Attention is All You Need"** revolutionized NLP. Example: Google Translate improvements.
- **LLMs (2018-Present):** Massive pre-trained models. Example: ChatGPT, GPT-5 conversational AI.



perceptron



Neural Network

so what is machine learning?

Definition

Machine learning is a subset of artificial intelligence that enables computers to learn and make decisions from data without being explicitly programmed for every specific task.

Core Concept

Instead of writing detailed instructions, we feed the computer examples and let it discover patterns to make predictions about new, unseen data.

data feed ⟶ model ⟶ train ⟶ make predictions on unseen data

is this machine learning?

example 1

a student develops a chat bot to assist you with cancelling an order on an online platform , here is the chat:

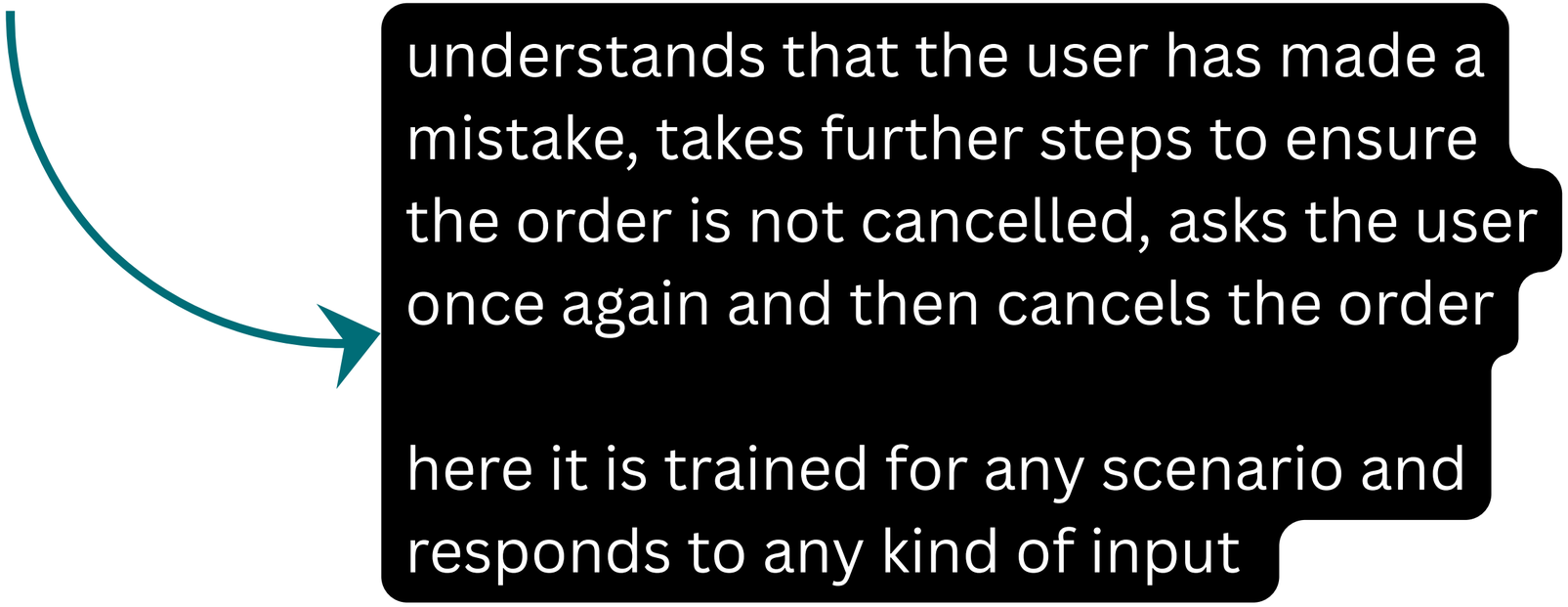
- **bot**: good morning, how may I help you, enter 1 for cancel, 2 for refreshing the chat
- **user**: hello
- **bot**: hello!
- **user**: 1
- **bot**: thank you! Your order has been cancelled successfully!
- **user**: nooo i didnt wanted to cancel my order, it was by mistake
- **bot**: Cannot address that message, please follow the instructions in the beginning
- **user**: idiot I didn’t wanted to cancel my order it was by mistake!!!
- **bot**: Cannot address that message, please follow the instructions in the beginning

this is not machine learning because it is rule-based and only responds to certain inputs

```python
def order_cancellation_bot():
    # Initialize conversation
    print("Good morning, how may I help you?")
    print("Enter 1 for cancel, 2 for refreshing the chat")

    # Main conversation loop
    while True:
        user_input = input("User: ").strip().lower()

        # Check for exact matches first
        if user_input == "hello" or user_input == "hi":
            print("Bot: Hello!")

        elif user_input == "1":
            print("Bot: Thank you! Your order has been cancelled successfully!")

        elif user_input == "2":
            print("Bot: Chat refreshed!")
            print("Enter 1 for cancel, 2 for refreshing the chat")

        elif user_input == "exit" or user_input == "quit":
            print("Bot: Goodbye!")
            break

        # Default response for unrecognized input
        else:
            print("Bot: Cannot address that message, please follow the instructior

# Run the chatbot
order_cancellation_bot()
```

what an intelligent model would do

User: "nooo I didn't want to cancel"
Bot: "I understand this was accidental. I can restore your order #12345 right away. Should I proceed with restoring it?"
User: "yes please"
Bot: "Done! Your order #12345 has been restored and will ship as originally scheduled."



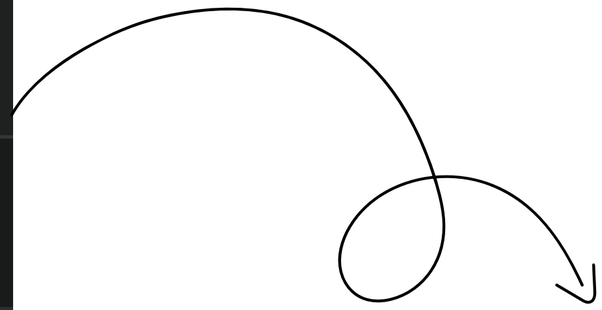
Machine learning algorithms:

Supervised learning
Unsupervised learning
Recommender systems
Reinforcement learning

Supervised learning

Learns from being given “right answers”

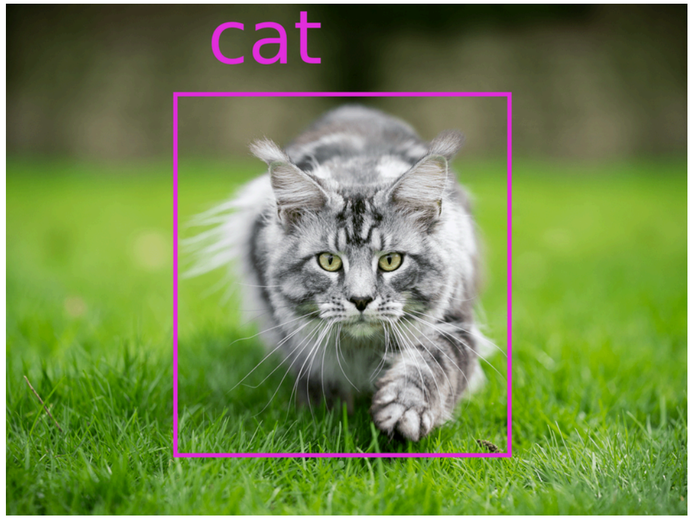
Area (m²)	Price (INR)
46.45	₹69,69,214
46.45	₹44,19,615
46.45	₹25,05,792
46.45	₹35,15,347
46.45	₹29,50,720
47.38	₹83,00,635

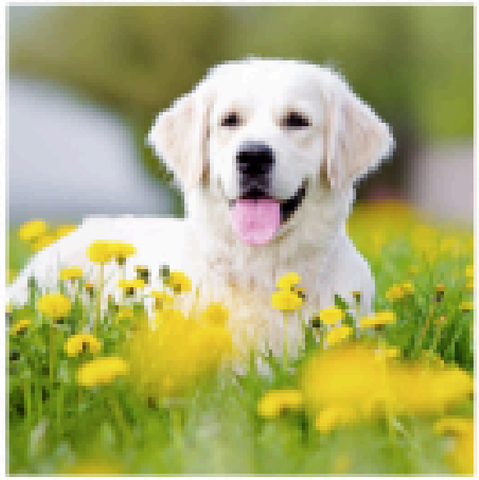


train the model on this data set

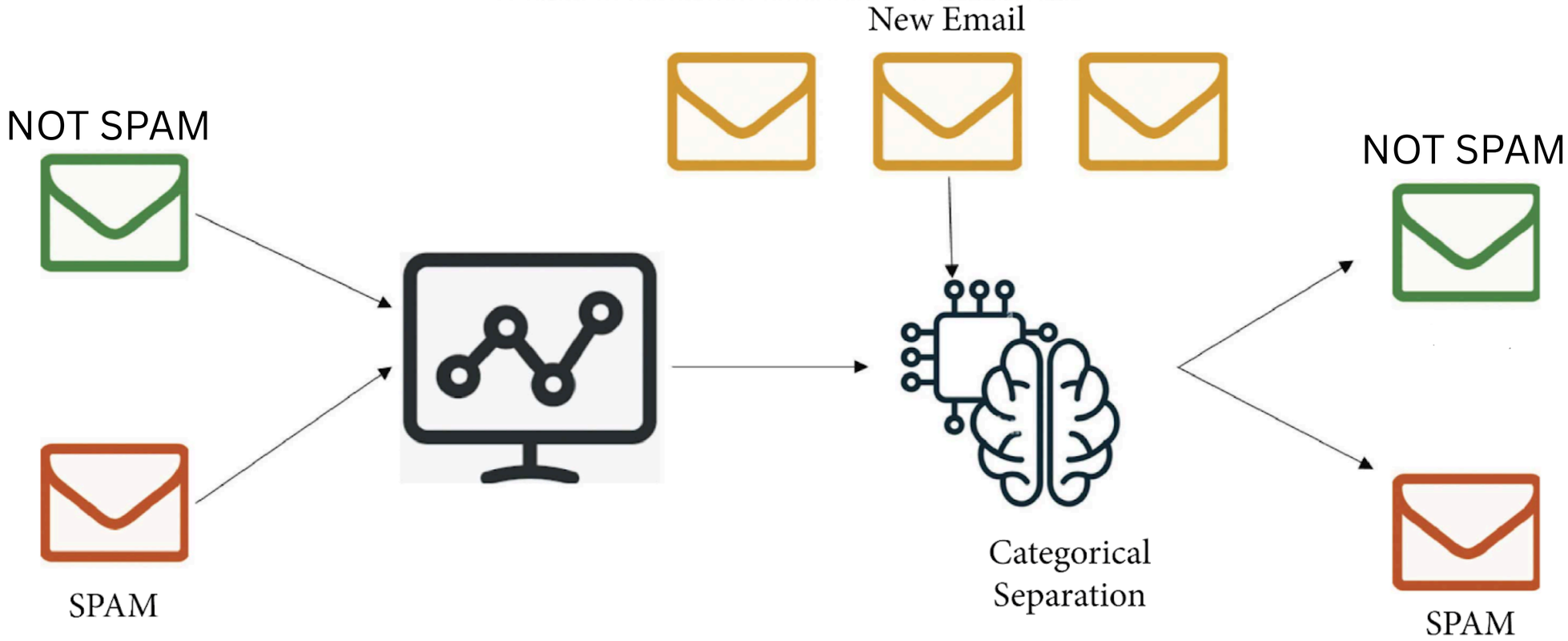
predict the price for a new area

Input (X)	Output (Y)	Application
email	spam? (0/1)	spam filtering
audio	text transcripts	speech recognition
English	Spanish	machine translation
ad, user info	click? (0/1)	online advertising
image, radar info	position of other cars	self-driving car
image of phone	defect? (0/1)	visual inspection


cat



dog

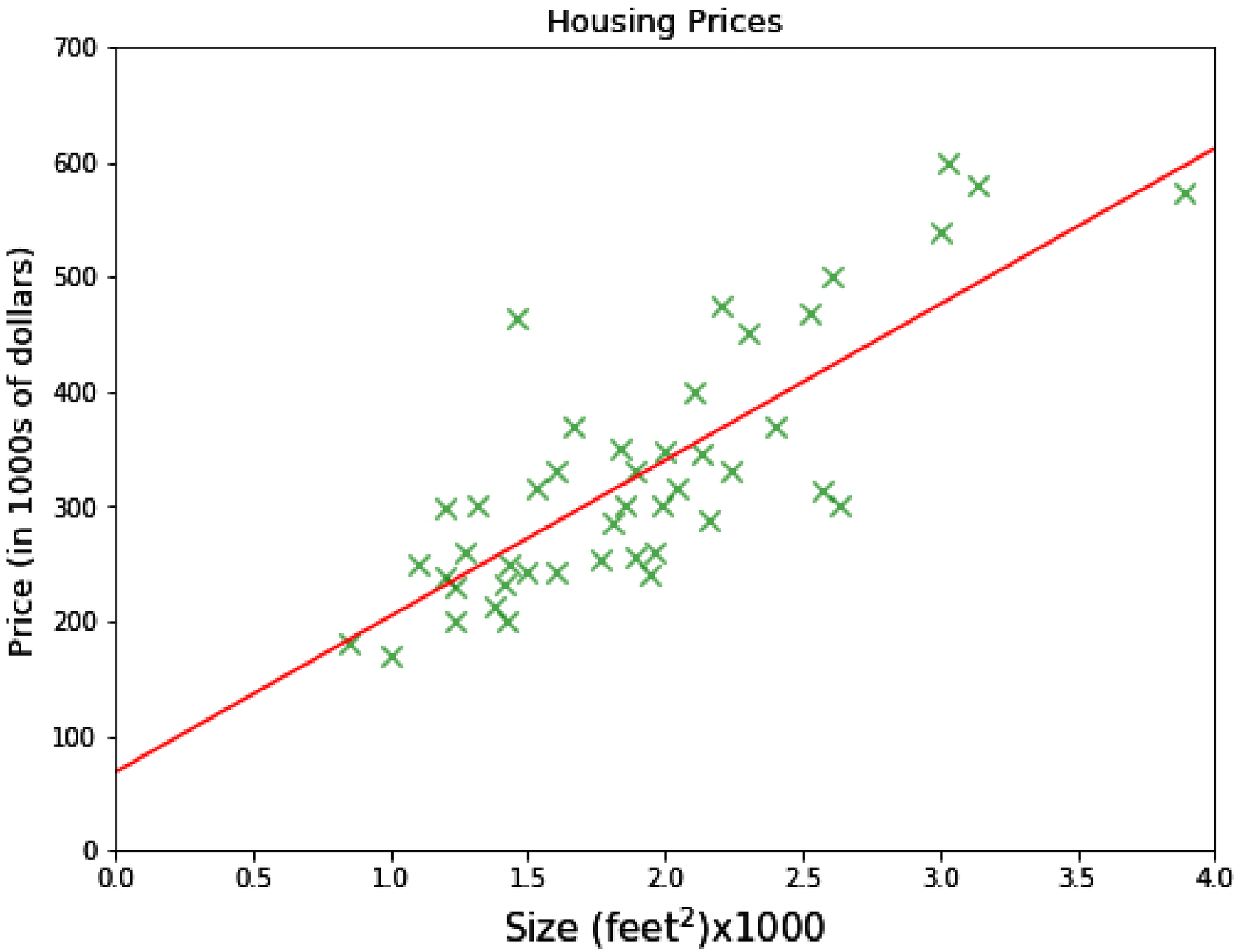

New Email
NOT SPAM
SPAM
Categorical
Separation
NOT SPAM
SPAM

Supervised Learning: Regression

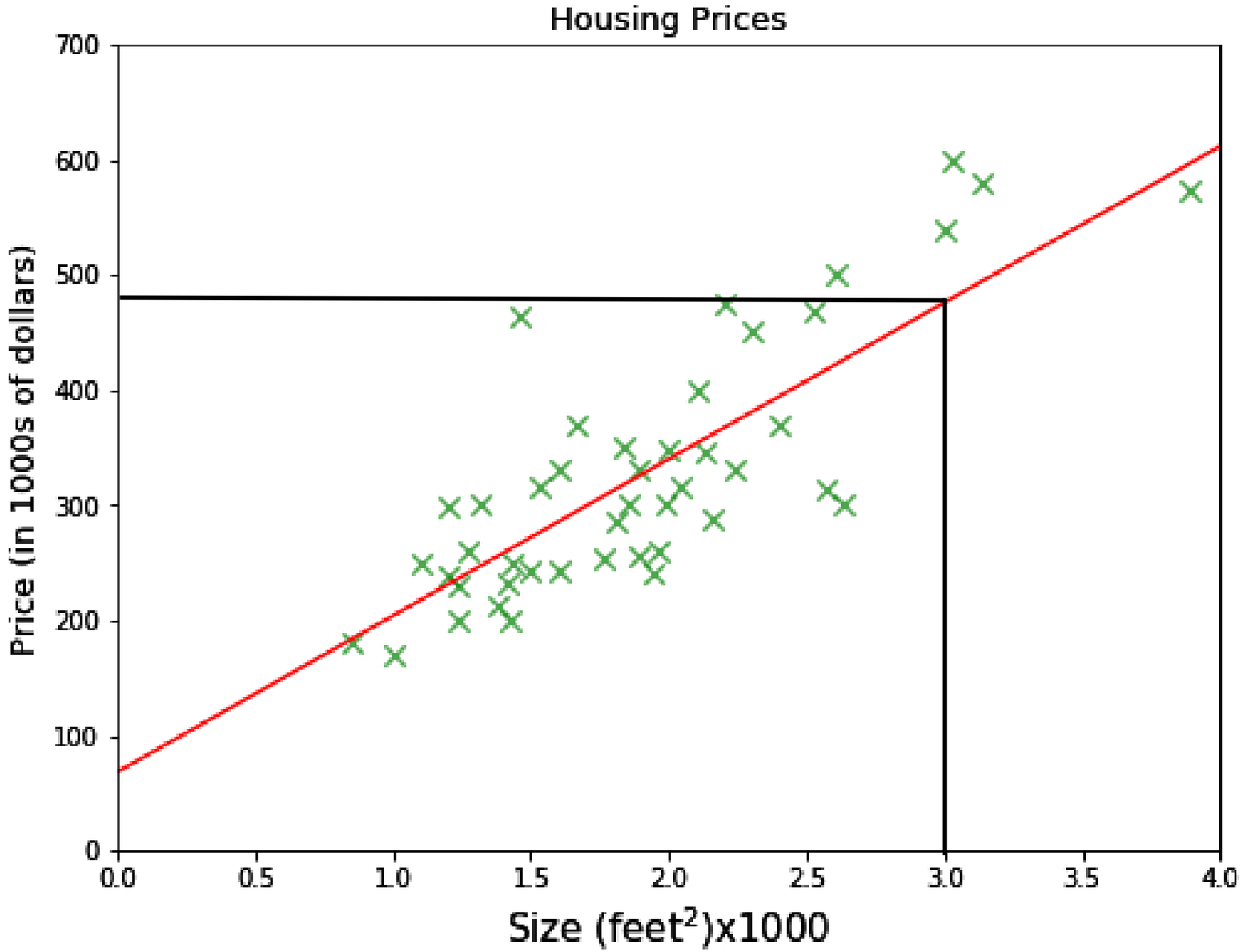
make a prediction out of the given data

example: predict the price of a house if the area and price is given for 10 samples

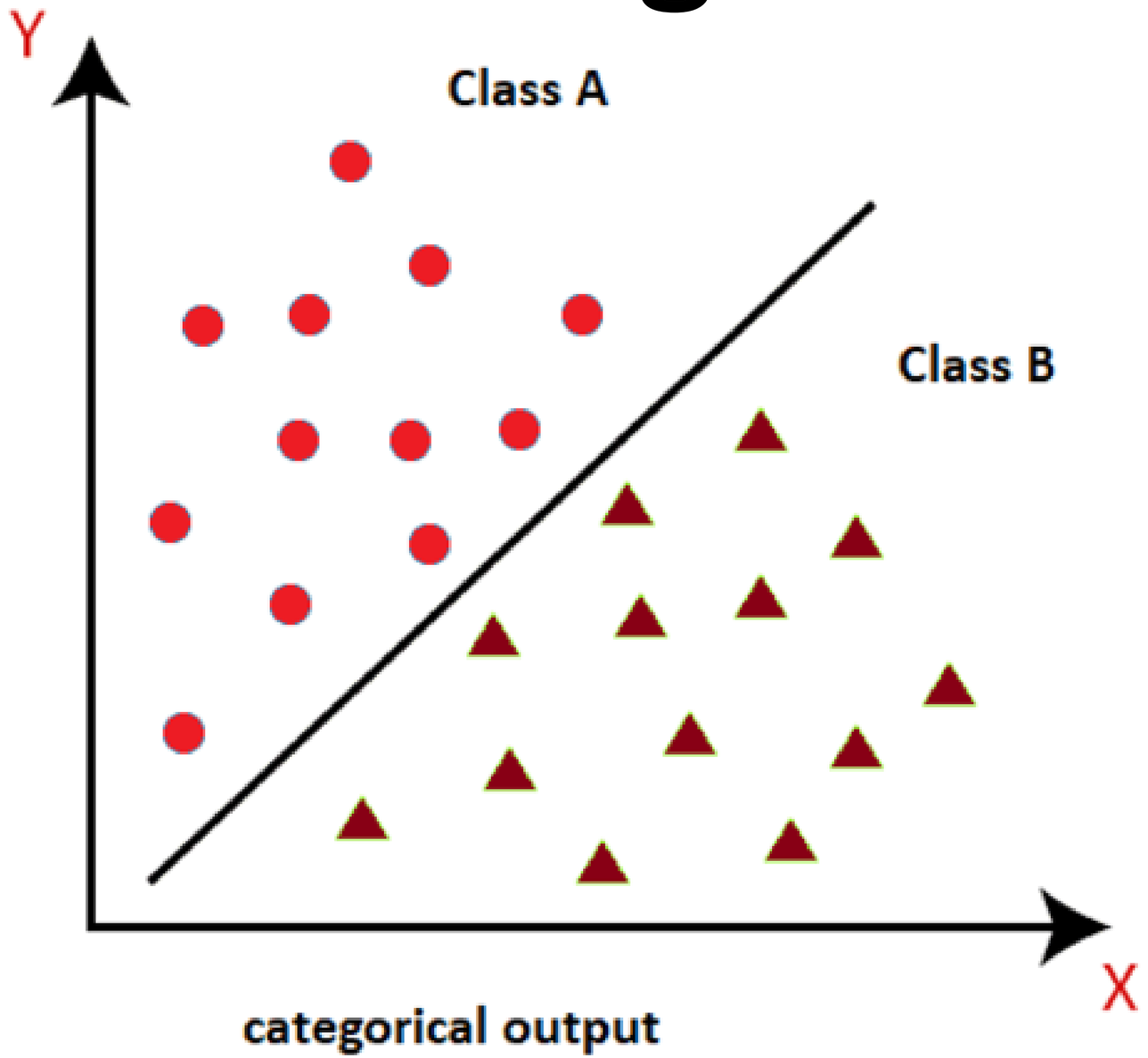
here input is area of the house and output is the price
we have to make a prediction for the price of a house whose area is given, this will be done based on the data of all the other samples


Housing Prices
Price (in 1000s of dollars)
Size (feet^2)x1000
0
100
200
300
400
500
600
700
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

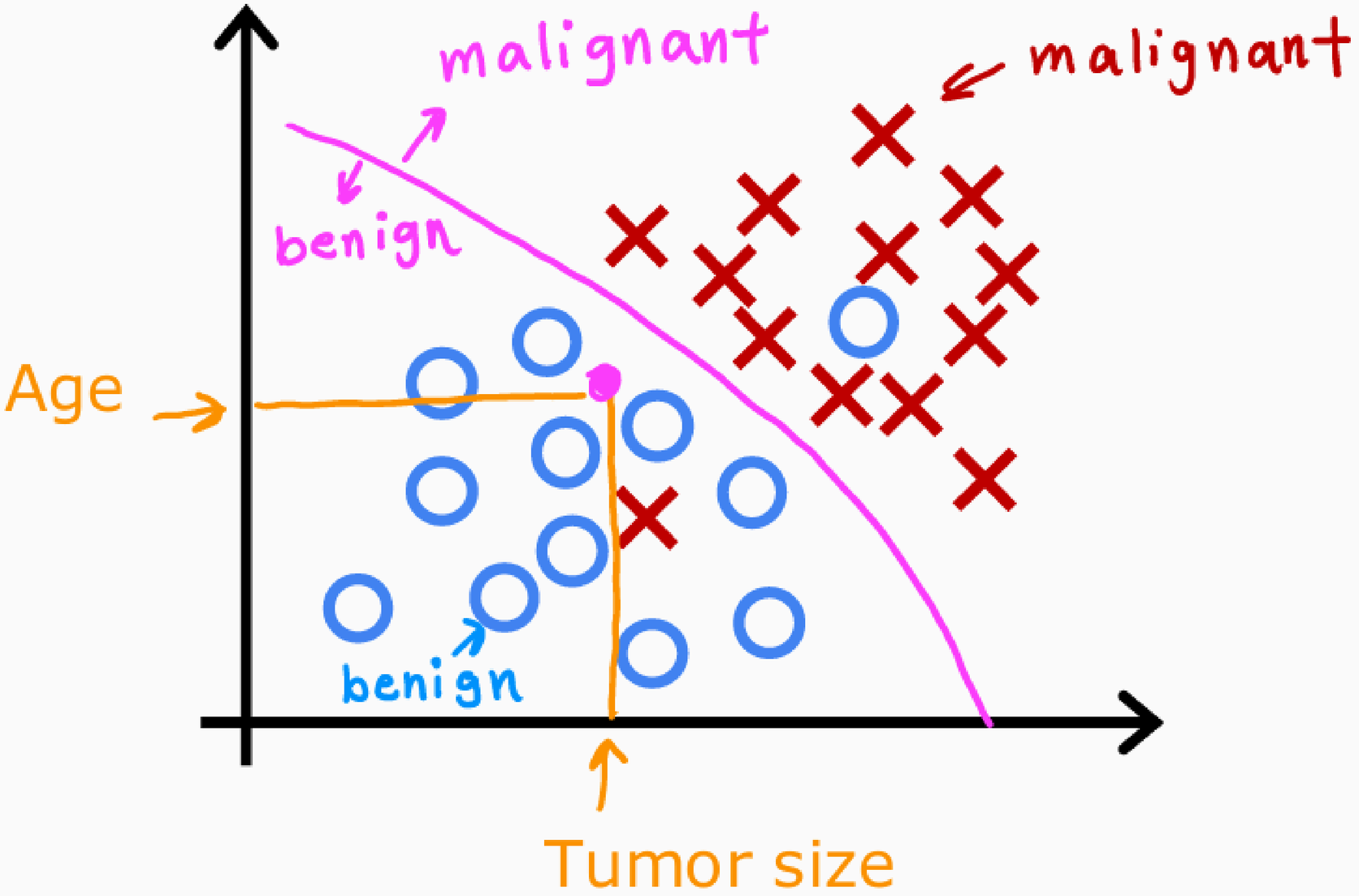
if size is given=3 prediction, approx $ 480k



Supervised Learning: Classifcation



Two or more inputs



other examples:

Image Classification

- Task: Predict the object in an image.

Medical Diagnosis

- Task: Classify whether a patient has a disease.
- Input: Symptoms, test results, imaging scans.
- Output: “Diabetes: Yes/No”, “Cancer: Benign/Malignant”.

Sentiment Analysis

- Task: Determine the sentiment of a text.
- Input: Movie review, tweet, or product review text.
- Output: “Positive”, “Negative”, “Neutral”

hands on session:

- create a github profile
- start a repository
- sign into google collab

1. all the codes should be implemented and pushed to this repository

2. go to google collab and start a new notebook

3. save the notebook in your github repo