CSDS503 / COMP552 – Advanced Machine Learning

Faizad Ullah

Supervised Learning

What does a classifier see?

What does a classifier see?

- Features

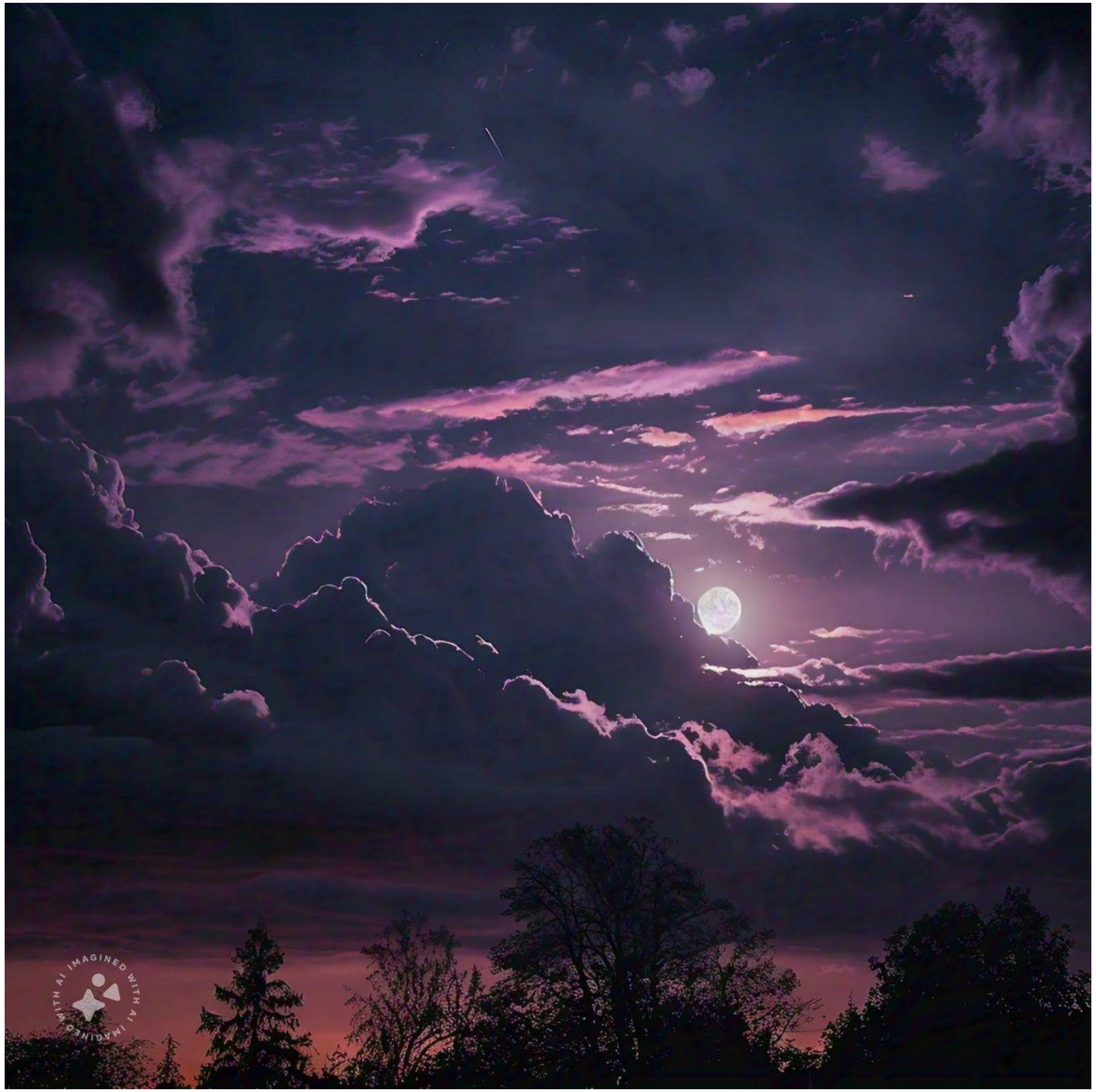
Morning:

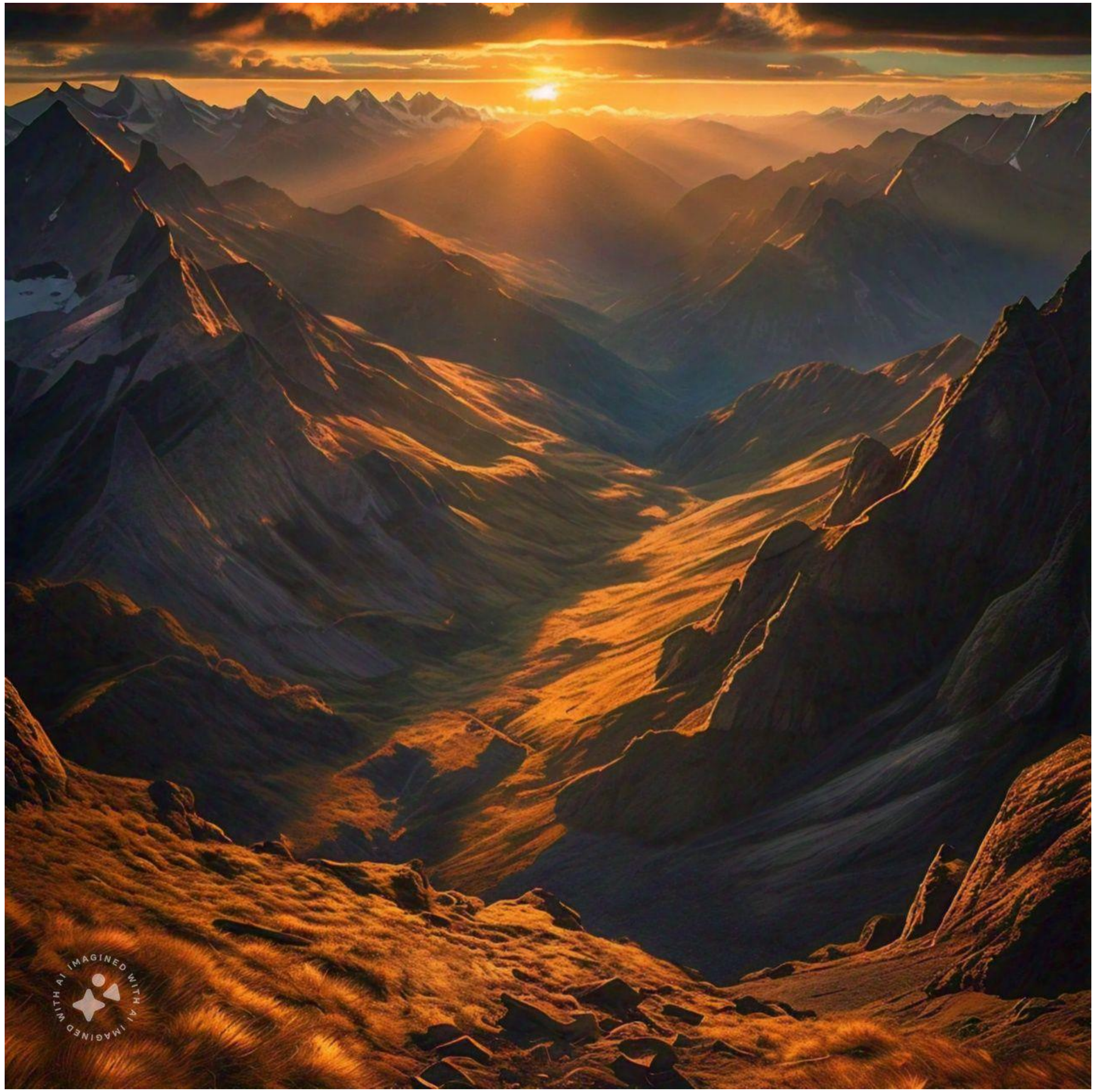
1.
2.
3.
4.
5.

Evening:

1.
2.
3.
4.
5.


IMAGINED WITH AI IMAGINED WITH AI

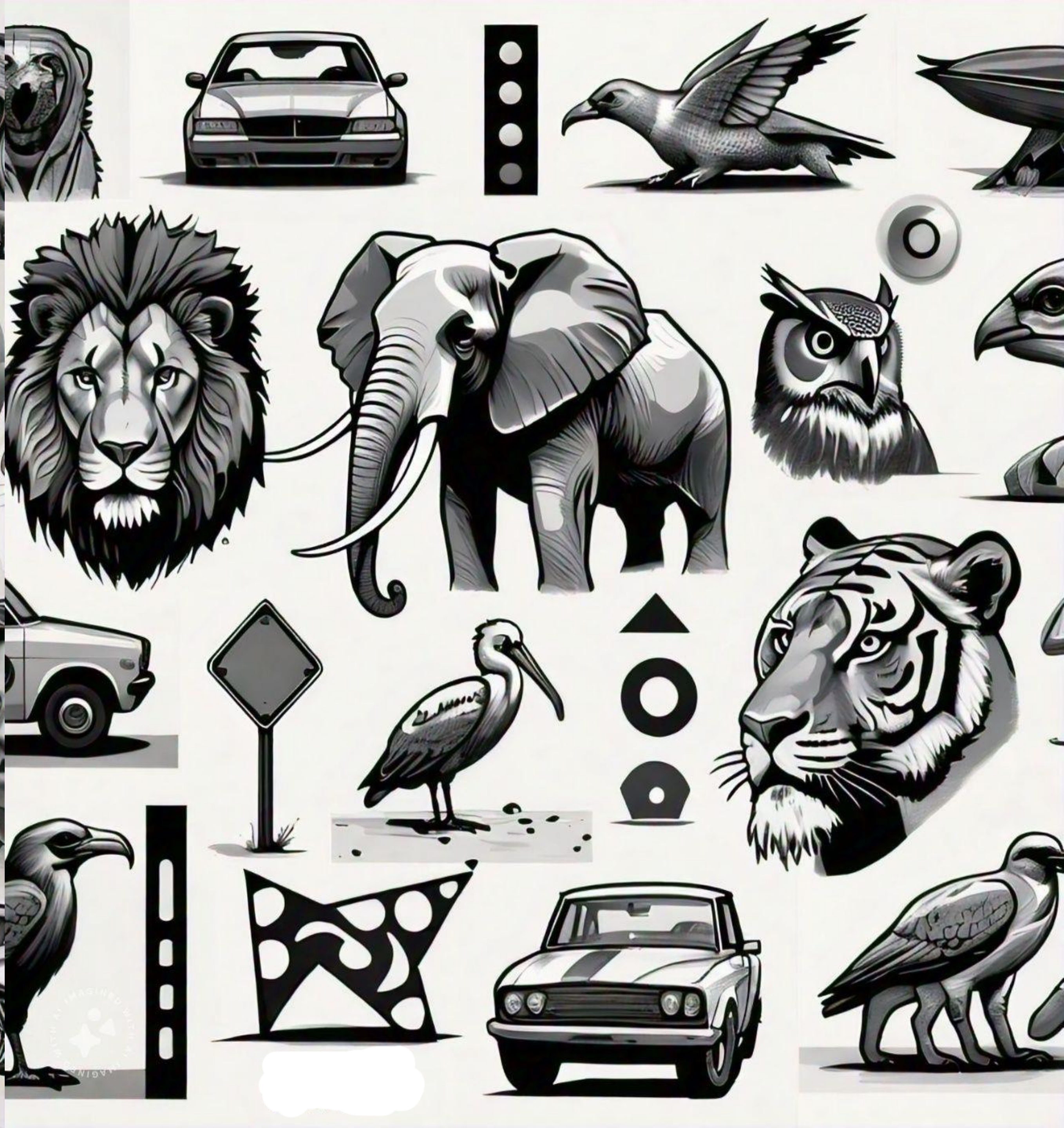

IMAGINED WITH AI


IMAGINED WITH AI


IMAGINED WITH AI IMAGINED WITH AI


IMAGINED WITH AI IMAGINED WITH AI

Unsupervised Learning


2
63


Supervised Learning Setup

Feature Space: Tabular Data

Features/Dimensions

Label/Class/Category

Dense Vectors

Height (inches)	Weight (kgs)	B.P.Sys	B.P.Dia	Heart disease
62	70	120	80	No
72	90	110	70	No
74	80	130	70	No
65	120	150	90	Yes
67	100	140	85	Yes
64	110	130	90	No
69	150	170	100	Yes
66	125	145	90	?
74	67	110	60	?

Record is 4-dimensional Feature Vector

Training Data/Training Split

Testing Data/Testing Split

As labels are discrete, this is a classification task.

Feature Space: Tabular Data

Features/Dimensions

Label

Dense Vectors

Height (inches)	Weight (kgs)	B.P.Sys	B.P.Dia	Cholesterol Level
62	70	120	80	150.50
72	90	110	70	165.70
74	80	130	70	135.45
65	120	150	90	210.09
67	100	140	85	195.00
64	110	130	90	125.56
69	150	170	100	250.80
66	125	145	90	?
74	67	110	60	?

A Record is 4-dimensional Feature Vector

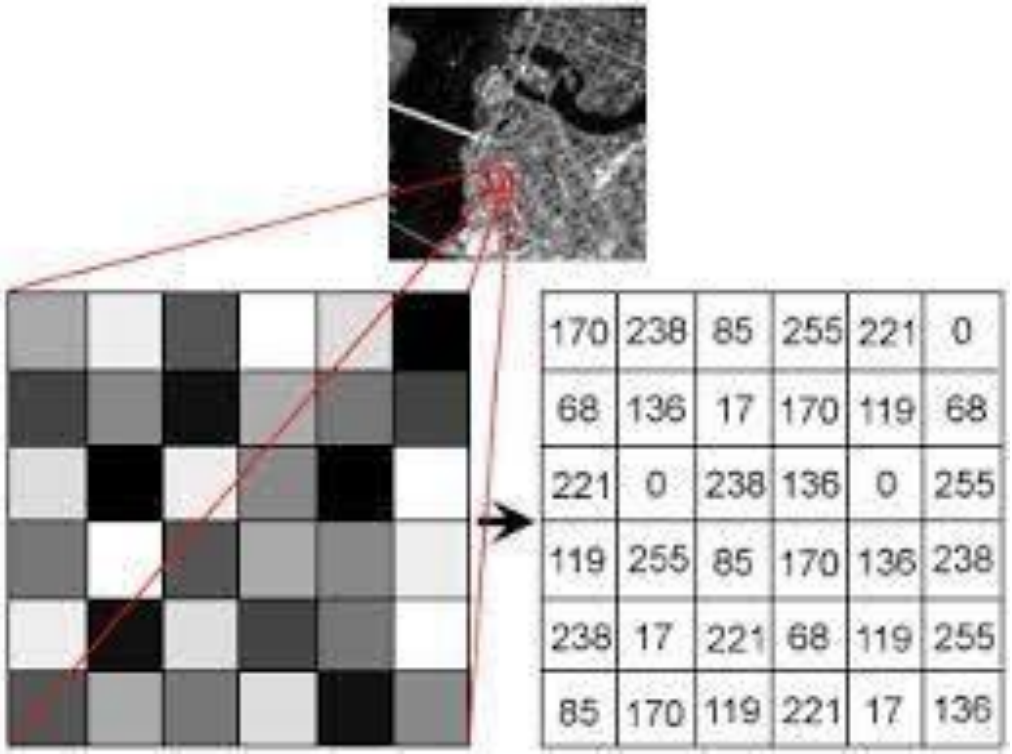
Training Data/Training Split

Testing Data/Testing Split

As labels are continuous, this is a regression task.

Feature Space: Image Data

- Images are nothing but a **2D/3D arrays** with values of color intensities, typically ranging $\mathbf{0} - \mathbf{255}$

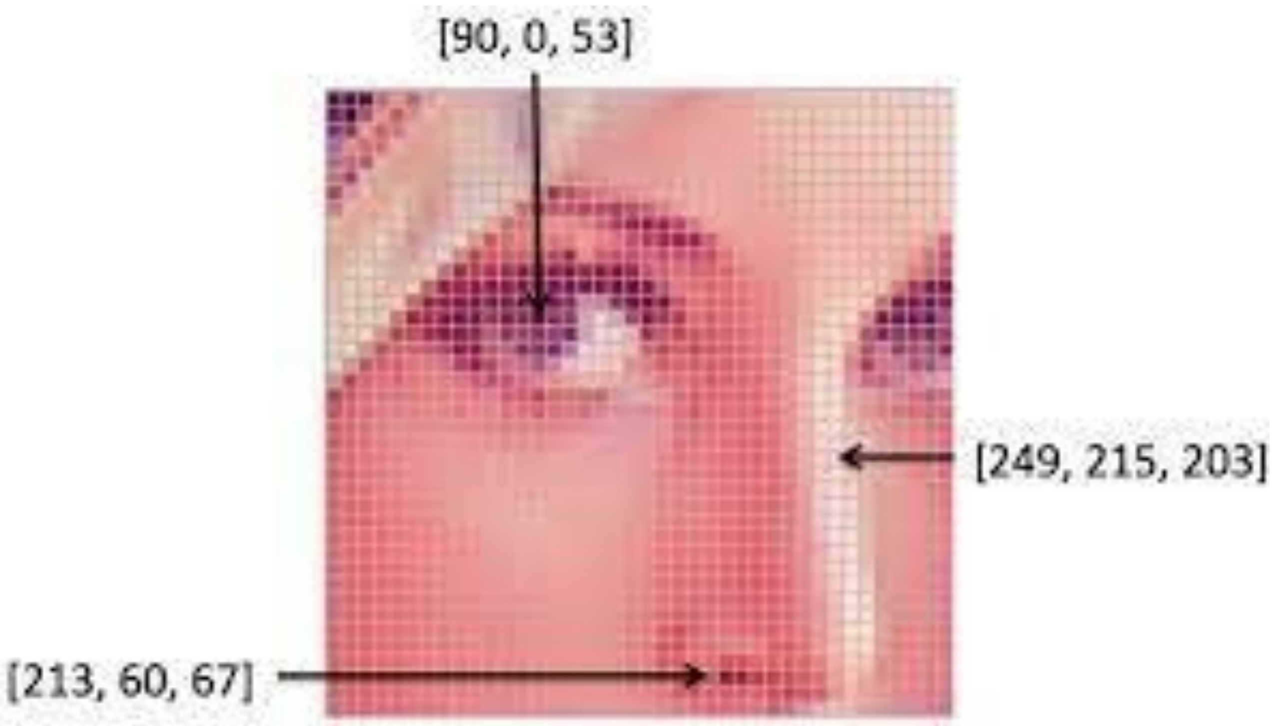


Dense Vectors

Feature Space: Image Data

- The color Image is 3D array $(Width \times Height \times Channels)$
- Color image has 3 channels while grayscale image has 1 channel.

Dense Vectors



Feature Space: Text Data

- Suppose you are given labeled textual data in excel sheet

	Document#	Text	Class
Training	1	the best movie best	Pos
	2	the best best ever	Pos
	3	the best film	Pos
	4	the worst cast ever	Neg
Testing	5	The Best best best worst ever	?

Sparse Vectors

the	best	movie	ever	film	worst	cast
1	1	1	0	0	0	0
1	1	0	1	0	0	0
1	1	0	0	1	0	0
1	0	0	1	0	1	1
These are called "Binary Occurrences" features.						
1	1	0	1	0	1	0

label
1
1
1
0

?

Distributions

- Suppose we want to predict whether someone is a child or an adult based on height and weight.
- If the feature makes sense, then there is a distribution $\boldsymbol{P}$ of the labels across the input features.
- Every child and adult gets sampled from that hidden distribution based on probabilities.

We cannot access $\boldsymbol{P}$

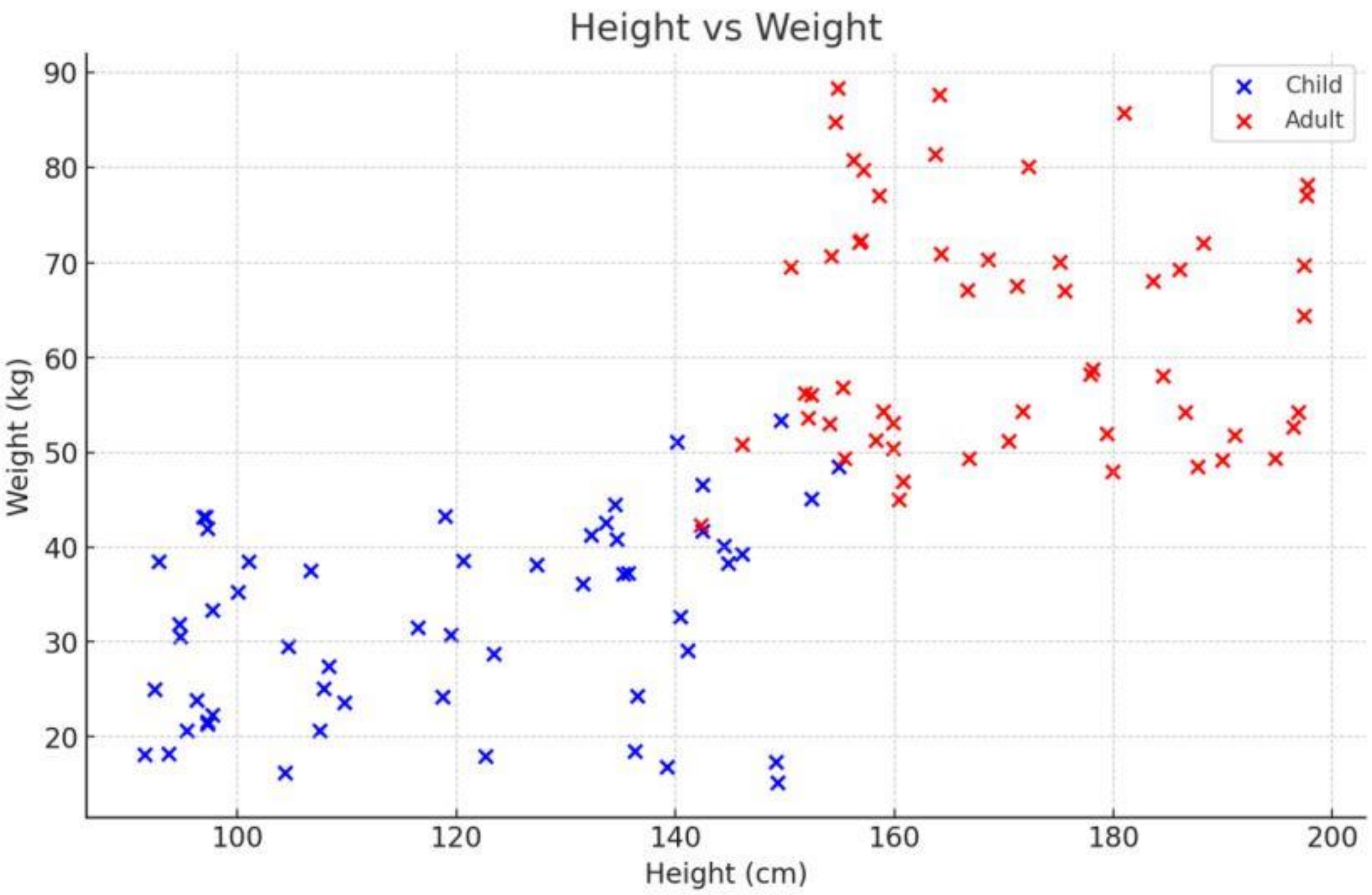
- But we can estimate (reverse-engineer) it by sampling from it

Estimating Distributions

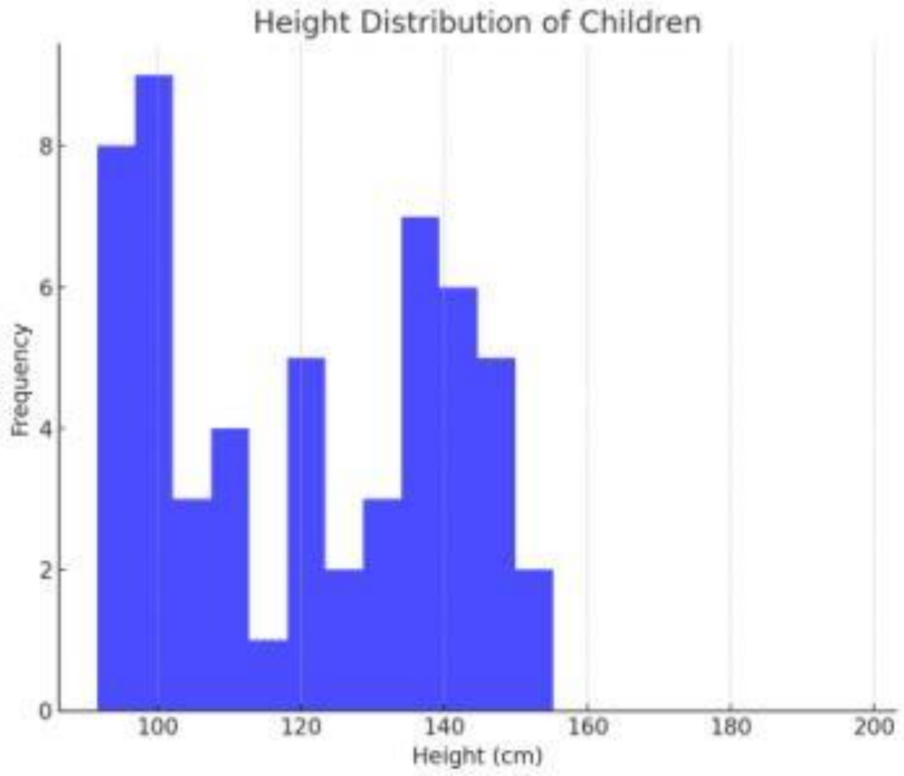
- We collect a dataset of 50 adults and 50 children

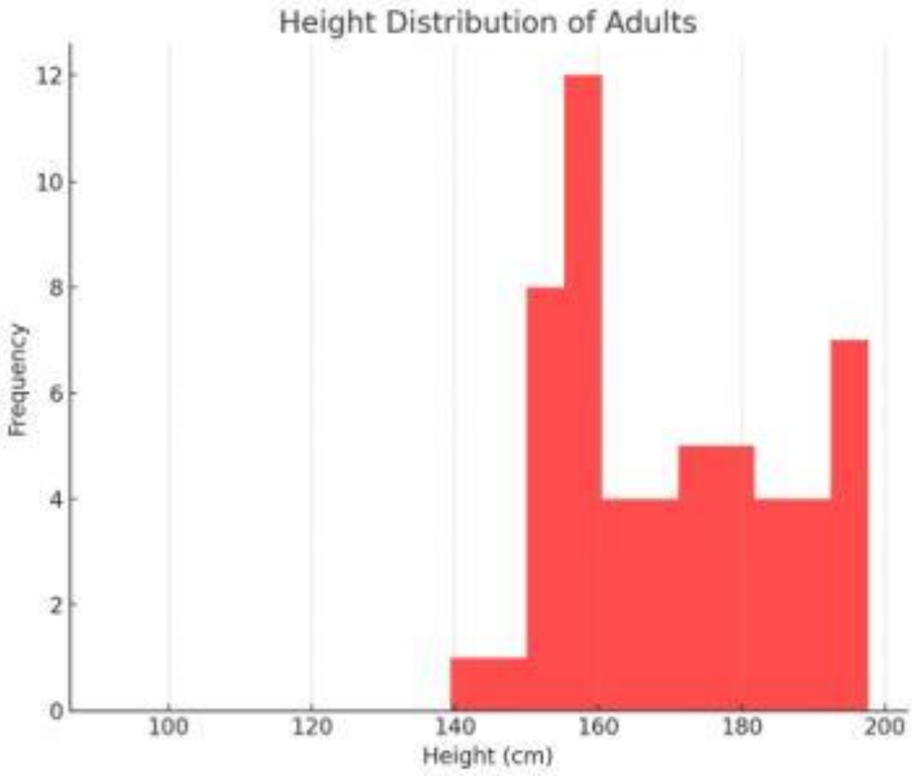
Height (cm)	Weight (kg)	Status
96.9	40.4	Child
131.6	30.9	Child
...	...	...
160.4	64.4	Adult
187.7	72.2	Adult
...	...	...

Estimating Distributions

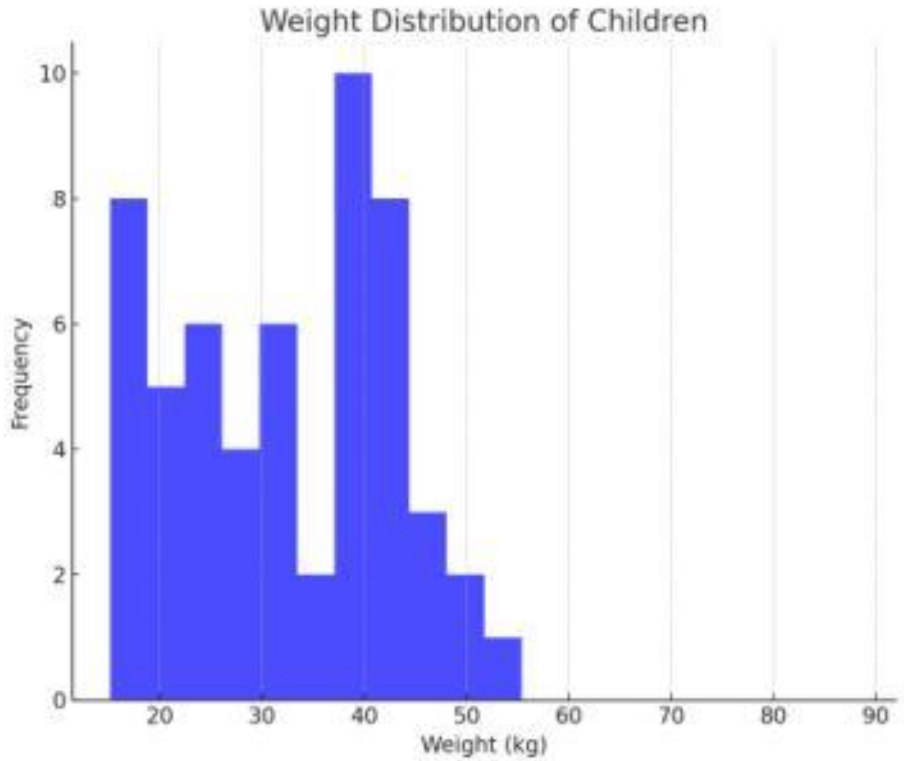


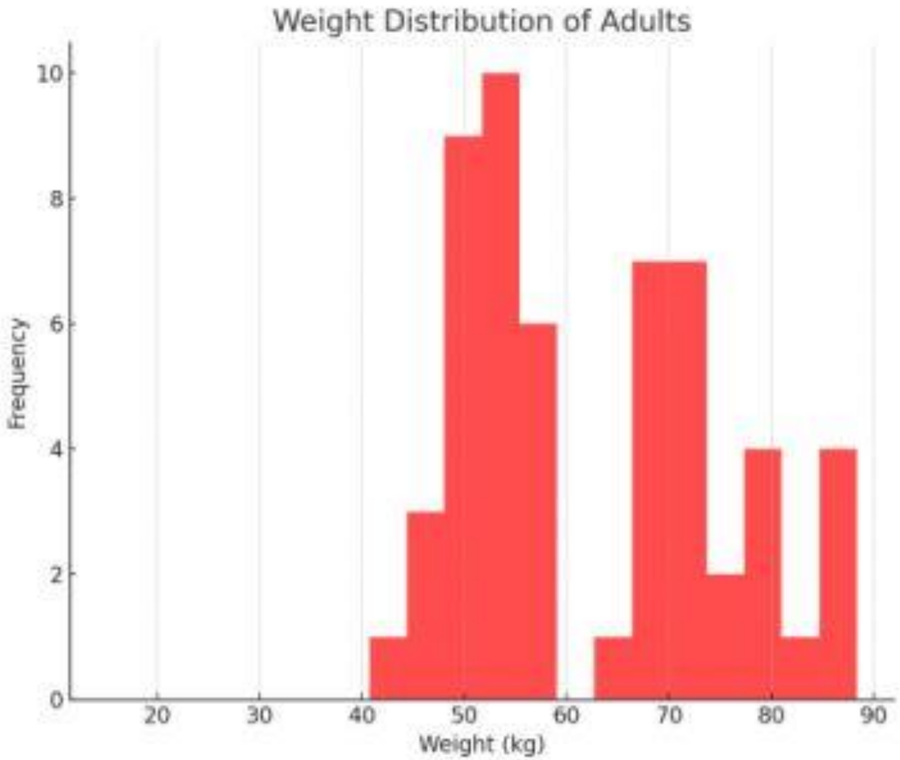
Estimating Distributions





Estimating Distributions





Rules vs. Learning

- Suppose we are working on classification of emails into “spam” and “ham” (not spam)
- **We can write a complicated set of rules**
  - Works well for a while
  - Cannot adapt well to new emails
  - Program could be reverse-engineered and circumvented
- **Learn the mapping between an email and its label using past labelled data**
  - Can be retrained on new emails
  - Not easy to reverse-engineer and circumvent in all cases
  - Easier to plug the leaks

Formalizing the Setup

$$\boldsymbol{D} = \{(\boldsymbol{x^1}, \boldsymbol{y_1}), (\boldsymbol{x^2}, \boldsymbol{y_2}), \dots, (\boldsymbol{x^n}, \boldsymbol{y_n}) \subseteq \boldsymbol{X} \times \boldsymbol{Y}$$

Feature vector

$$\boldsymbol{D} = \{(\overrightarrow{\boldsymbol{x_1}}, \boldsymbol{y_1}), (\overrightarrow{\boldsymbol{x_2}}, \boldsymbol{y_2}), \dots, (\overrightarrow{\boldsymbol{x_n}}, \boldsymbol{y_n}) \subseteq \boldsymbol{X} \times \boldsymbol{Y}$$

- Where,
  - D is the dataset
  - X is the d-dimensional feature space $(\mathbb{R}^d)$
  - $\overrightarrow{x_i}$ or x^i is the input vector of the ith sample/record/instance
  - Y is the label space

Any categorical attribute can be converted to numerical representation.

The data points are drawn from an **unknown** distribution P

$$(\overrightarrow{\boldsymbol{x_i}}, \boldsymbol{y_i}) \sim \boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y})$$

If we don't know the distribution, lets approximate that using samples we gathered!

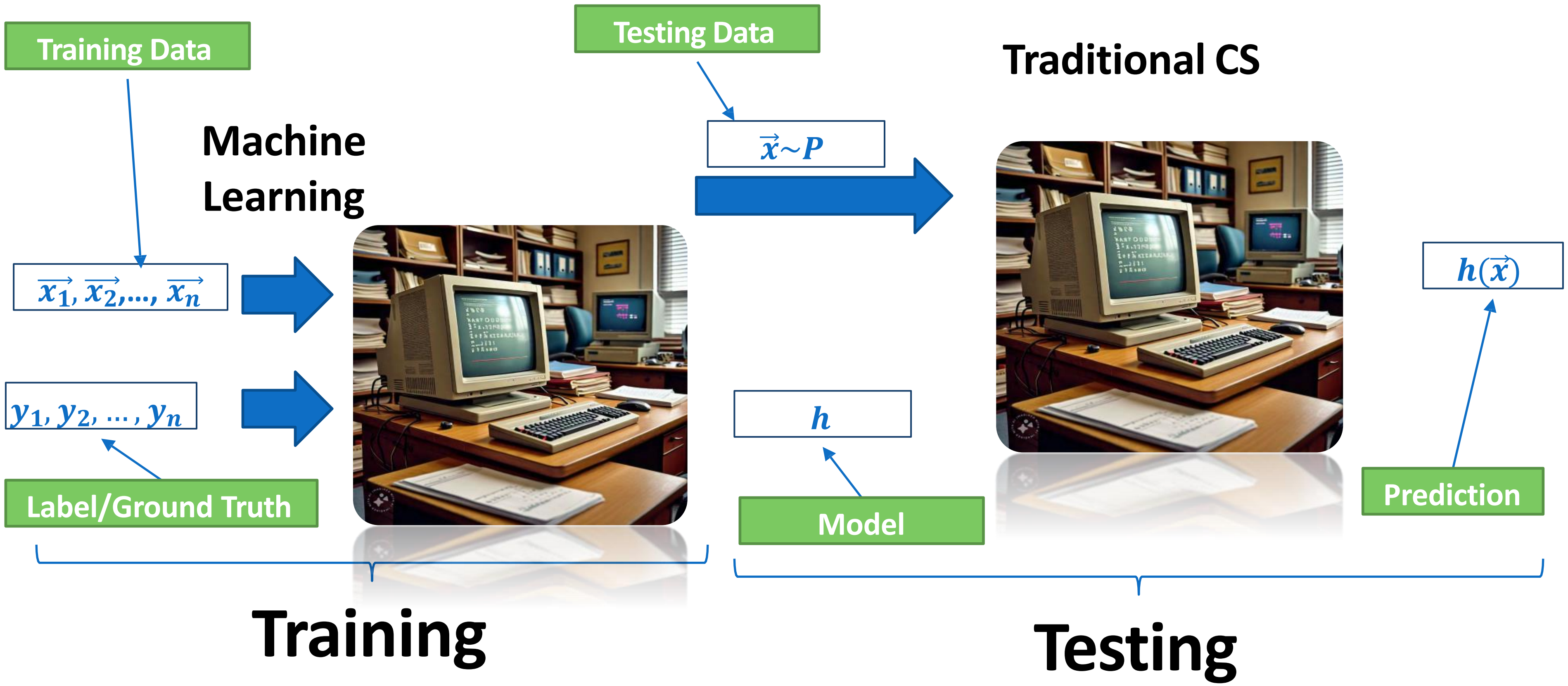
We want to learn a function $h \in H$, such that for a new instance $(\boldsymbol{x_1}, \boldsymbol{y}) \sim P$

$\boldsymbol{h}(\vec{\boldsymbol{x}}) = \boldsymbol{y}$ with a high probability or at least $\boldsymbol{h}(\vec{\boldsymbol{x}}) \approx \boldsymbol{y}$

This also have to be from the same distribution as $\overrightarrow{x_i}$

In plain words, don't train on dogs and ask prediction for cats.

Training and Testing: Formally



$h(\vec{x}) = y$ **(Ideal)**

$h(\vec{x}) \approx y$ **(Plausible)**

Label Space

- **Binary (Binary classification)**
  - Sentiment: positive / negative
  - Email: spam / ham
  - Online Transactions Fraud: Yes / No
  - Tumor: Malignant / Benign
  - $y \in \{0,1\}$
  - $y \in \{-1, 1\}$
- **Multi-class (multi-class classification)**
  - Sentiment: Positive / Negative / Neutral
  - Emotion: Happy / Sad / Surprised / Angry / …
  - Parts of Speech Tag: Noun / Verb / Adjective / Adver / …
  - $y \in \{0,1,2, \dots\}$
- **Real-valued (Regression)**
  - Temperature, height, age, length, weight, duration, price, …

Hypothesis Space

- The hypothesis h is sampled from a hypothesis space H

  $\boldsymbol{h \in H}$ $\boldsymbol{H \in \{H_D, H_R, H_{SVM}, H_{DL}, \dots\}}$

- H can be thought of to contain types of hypotheses, which share sets of assumptions like:
  - Support Vector Machines $\boldsymbol{H_{SVM} \in \{H_1, H_2, \dots\}}$
  - Decision Tree $\boldsymbol{H_D \in \{H_1, H_2, \dots\}}$
  - Perceptron $\boldsymbol{H_P \in \{H_1, H_2, \dots\}}$
  - Neural Networks $\boldsymbol{H_{NN} \in \{H_1, H_2, \dots\}}$
  - …

  $\boldsymbol{h \in H_D}$

  Selection done automatically.

- For example: $h \in H$ for H decision trees:

  Selection done manually.

  - Would be instance of decision trees of different height, arity, thresholds etc.

So, how do we choose our h?

- **Randomly?**
- **Exhaustively?**

How do we evaluate h?

How to choose h?

- **Randomly**
  - May not work well
  - Like using a random program to solve your sorting problem!
  - May work if H is constrained enough
- **Exhaustively**
  - Would be very slow!
  - The space H is usually very large (if not infinite)
- H is usually chosen by ML Engineers (You!) based on their experience
  - $h \in H$ is estimated efficiently using various optimization techniques

References

- Murphy Chapter 1
- Alpaydin Chapter 1
- TM Chapter 1
- Lectures of Andrew Ng., Dr. Ali Raza, and “Machine Learning for Intelligent Systems (CS4780/CS5780)”, Kilian Weinberger.
- This disclaimer should serve as adequate citation.