Week 05 | Lecture 06
Linear Regression and Classification in Machine Learning

Wan Fang
Southern University of Science and Technology

Robot is not limited to Physical Body

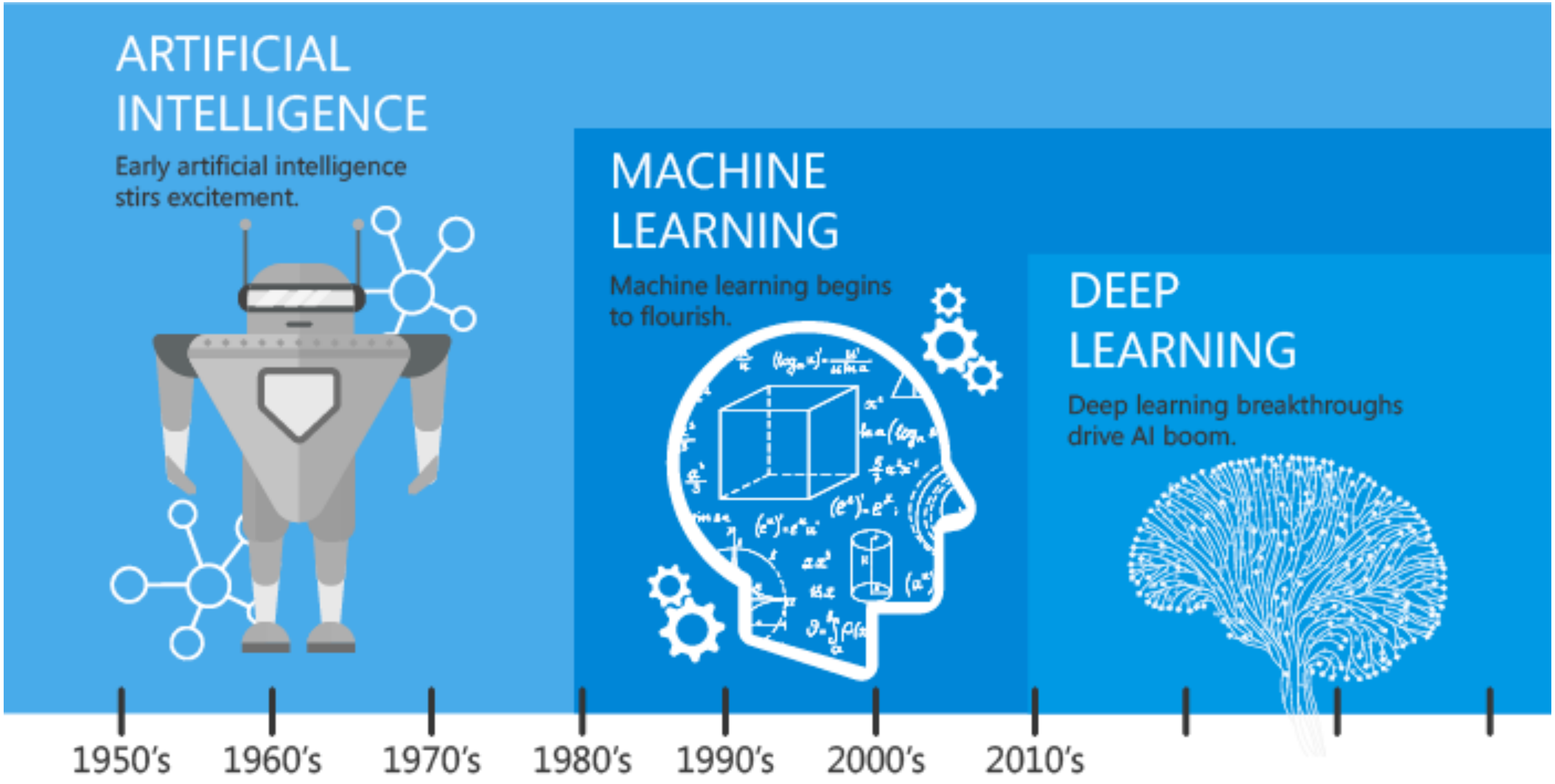
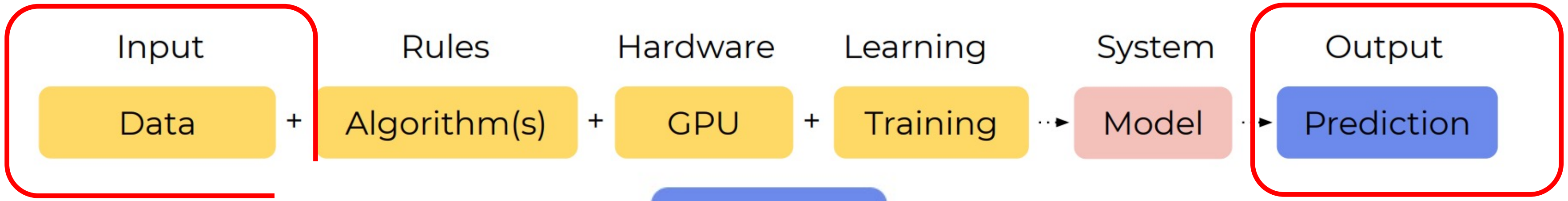


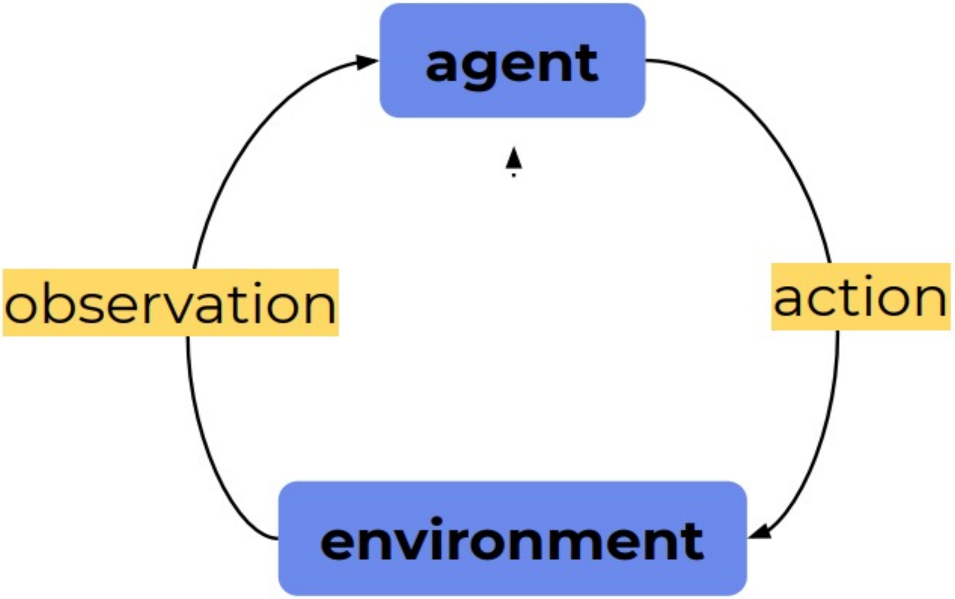
Image: Linked In | Machine Learning vs Deep learning

The ML process

To get acquainted with terms and understand how a model arrives at a prediction, it can be helpful to draw an analogy with a process we're familiar with: baking a cake.



Camera
Microphone
Keyboard
Screen
Wearable device
…



Light
Speaker
Vibrator
Screen
Wearable device
Robots
Car
…

Space, Time, Human, Objects…
Virtual world
Learn from virtual world and apply in real world

Today's Agenda

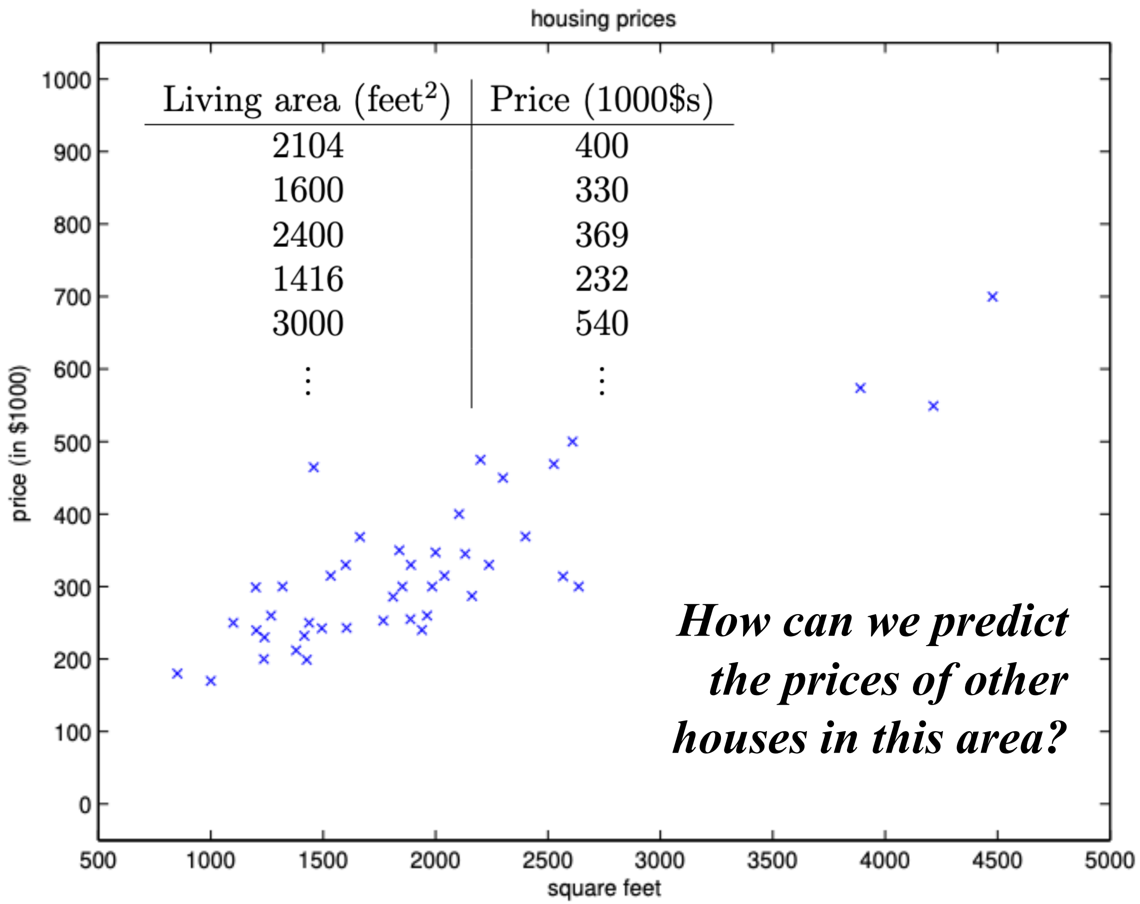
- **Machine Learning Basics**
  - Data-driven Learning Theory
  - Elements of Machine Learning
  - A Roadmap of Supervised Learning
- **Regression vs. Classification**
  - Linear Regression
  - Linear Classification
  - Perceptron with an Activation

Machine Learning Basics

(Supervised) Machine Learning

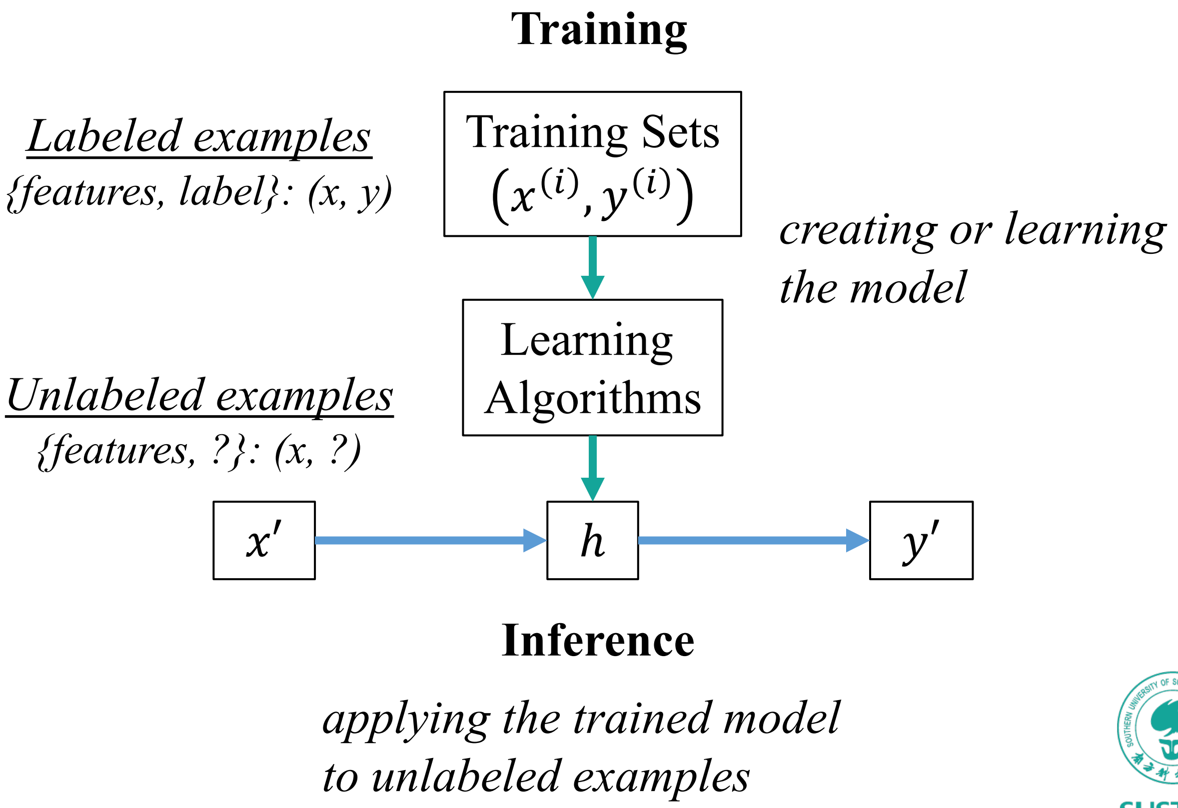
The ability to teach a computer without explicitly programming it

- Design a **Model** that defines the relationship between **Features** (input x_i) and **Labels** (output y)



Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

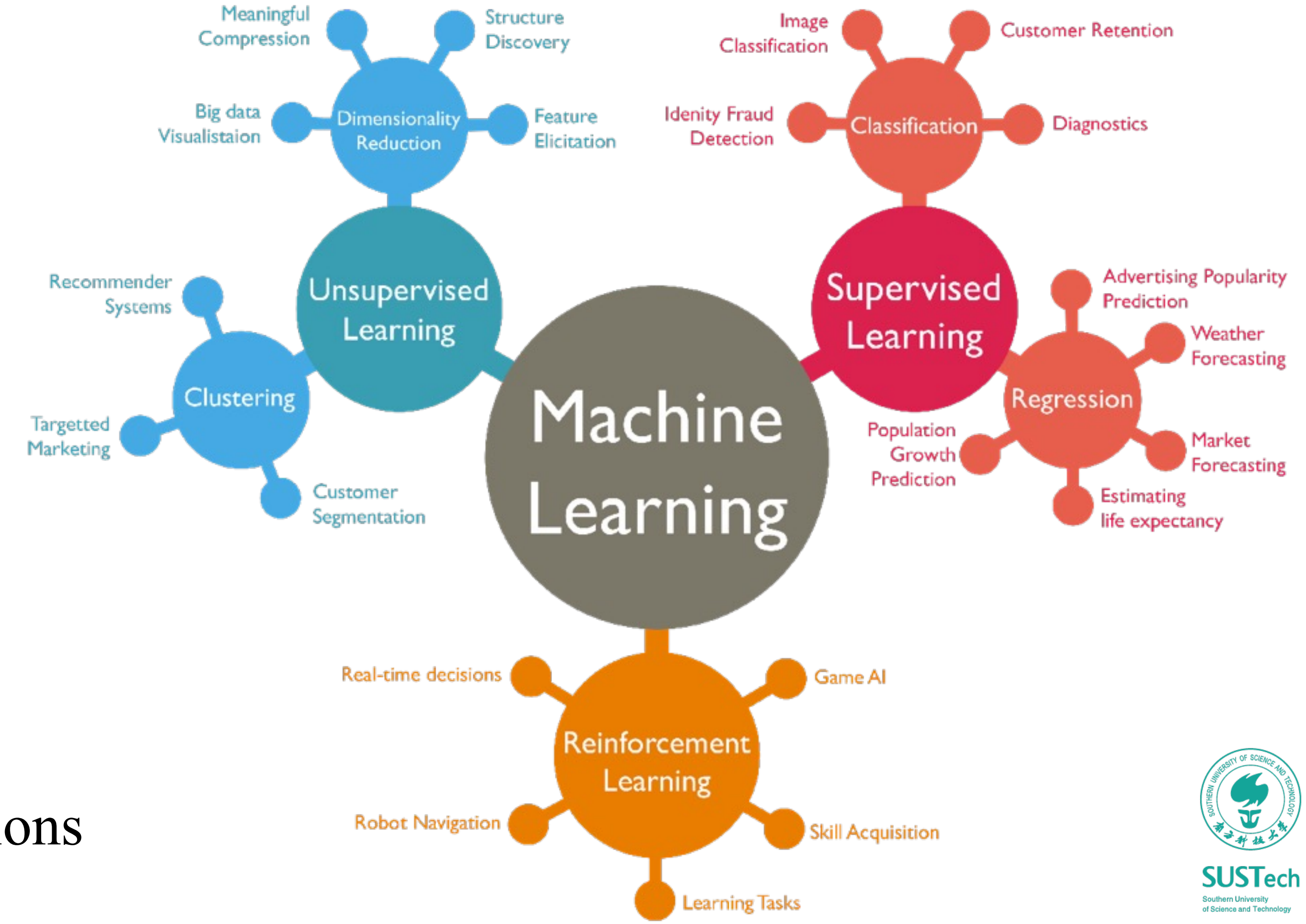
How can we predict the prices of other houses in this area?



The Landscape of Machine Learning

Differences between different learning problems

- **Supervised Learning**
  - Training data is labeled
  - Goal is correctly label new data
- **Reinforcement Learning**
  - Training data is unlabeled
  - Receives feedback for its actions
  - Goal is to perform better actions
- **Unsupervised Learning**
  - Training data is unlabeled
  - Goal is to categorize the observations



AncoraSIR.com

Features in Machine Learning

The observations (input variable x_i) that are used to form predictions

- **Image Classification**
  - Label images with appropriate categories
  - *The pixels are the features*
- **Autonomous Driving**
  - Enable cars to drive
  - *Data from the cameras, range sensors, and GPS are features*
- **Speech Recognition**
  - Convert voice snippets to text (e.g. Siri)
  - *The pitch and volume of the sound samples are the features*
- **Extracting relevant features is important for building a model**
  - *Time of day* is an irrelevant feature when classifying images
  - *Time of day* is relevant when classifying emails because SPAM often occurs at night
- **Common Types of Features in Robotics**
  - Pixels (RGB data)
  - Depth data (sonar, laser rangefinders)
  - Movement (encoder values)
  - Orientation or Acceleration (Gyroscope, Accelerometer, Compass)

AncoraSIR.com

Measuring Success for Classification

A confusion matrix that allows visualization of the performance of an algorithm

		y **Actual Value** (as confirmed by experiment)		
		True	False	
y' **Predictive Value** (predicted by the test)	Positive	**True Positive (TP)** *Correctly identified as relevant*	**False Positive (FP)** *Incorrectly labeled as relevant* Type I Error	Precision $\frac{TP}{(TP+FP)}$
	Negative	**True Negative (TN)** *Correctly identified as not relevant* Type II Error	**False Negative (FN)** *Incorrectly labeled as not relevant*	Negative Predictive Value $\frac{TN}{(TN+FN)}$
		Sensitivity $\frac{TP}{(TP+TN)}$	Precision $\frac{FP}{(FP+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Regression uses other measurements

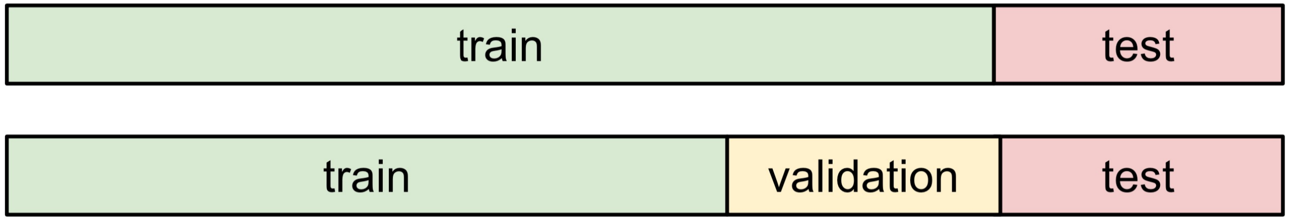
Can be used for both ***single-class*** and ***multi-class*** classification problems

AncoraSIR.com

Training and Test Data, Bias and Variance

Characteristics of Data

- **Training Data**
  - Data used to learn a model
- **Test Data**
  - Data used to assess the accuracy of model



Overfitting

- Model performs well on training data but poorly on test data

- **Bias**
  - Expected difference between model's prediction and truth
- **Variance**
  - How much the model differs among training sets

$$\begin{aligned} \mathrm{MSE} &= \mathbb{E}[(\hat{\theta}_m - \theta)^2] \\ &= \mathrm{Bias}(\hat{\theta}_m)^2 + \mathrm{Var}(\hat{\theta}_m) \end{aligned}$$

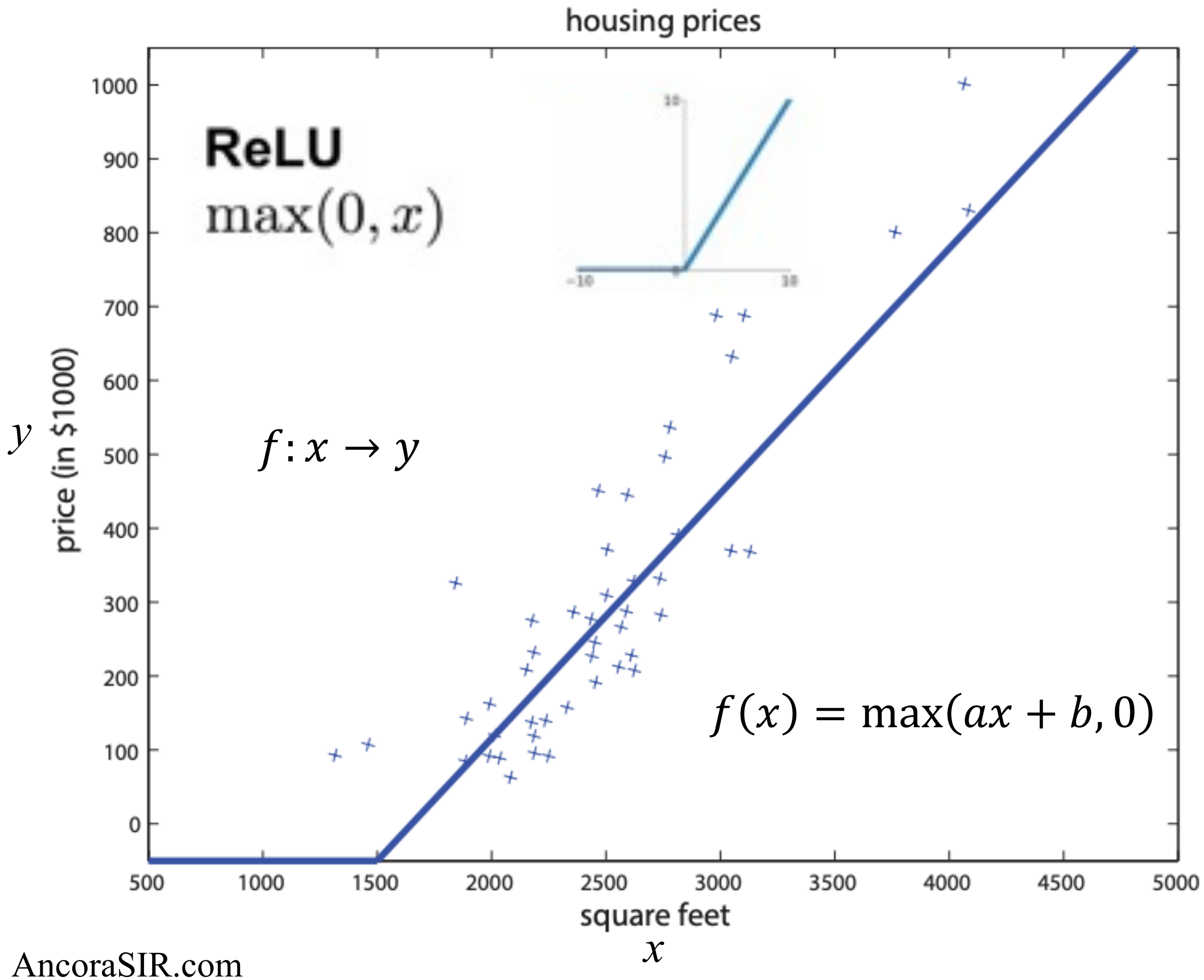
Model Scenarios

- *High Bias*: Model makes inaccurate predictions on training data
- *High Variance*: Model does not generalize to new datasets
- *Low Bias*: Model makes accurate predictions on training data
- *Low Variance*: Model generalizes to new datasets

AncoraSIR.com

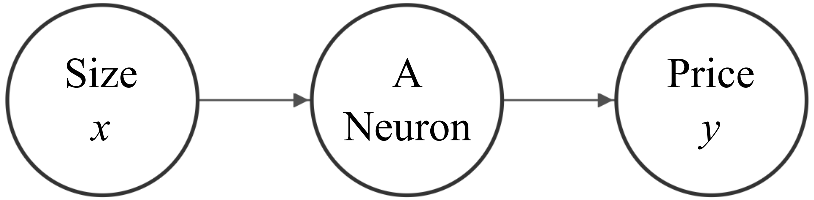
A Further Look into Housing Price Prediction

Building a neural network with Weighted-Sum Scores

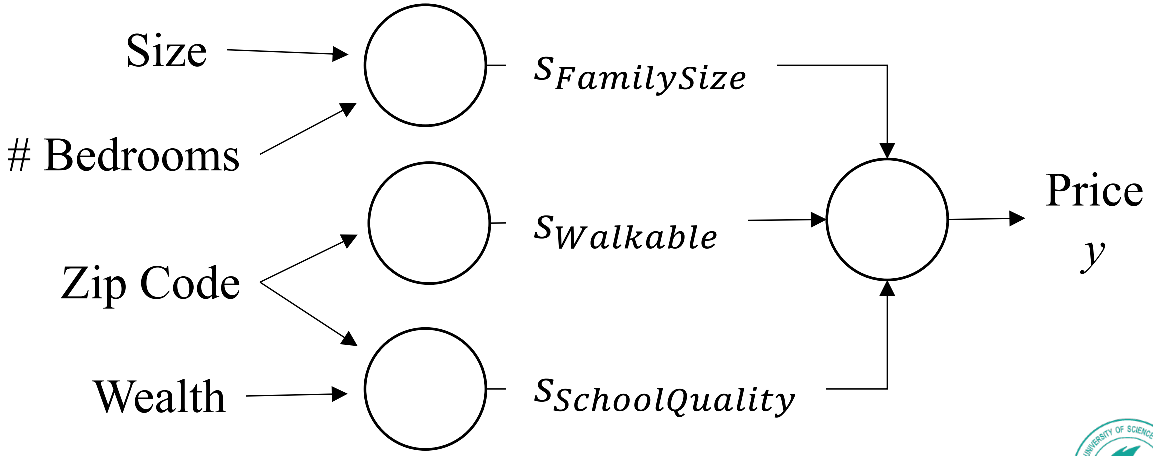


AncoraSIR.com

$$y = f_{WeightedSum}(x) = wx + b$$



*From **a single neuron** to **a network of neurons***

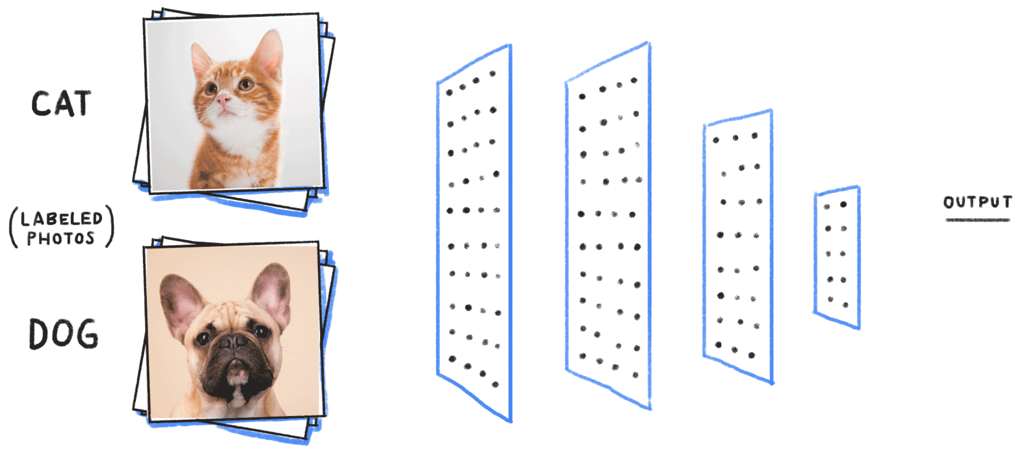


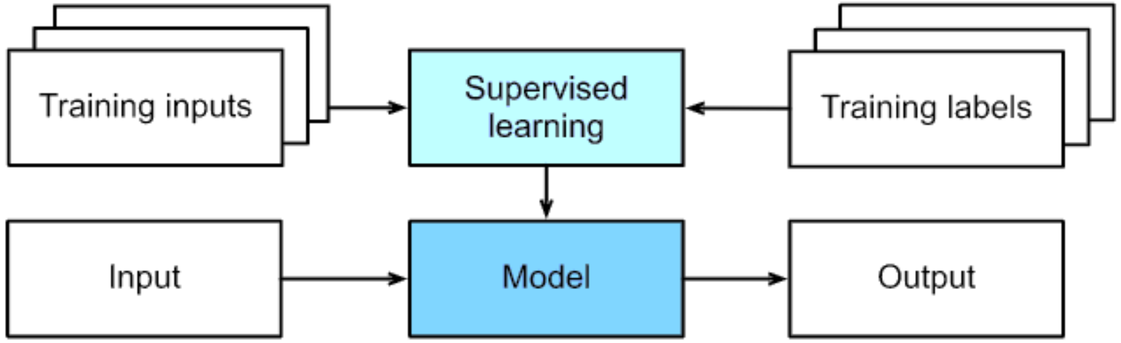
$$\boldsymbol{s} = f_{WeightedSum1}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$

$$y = f_{WeightedSum2}(\boldsymbol{s})$$

Supervised Learning with Neural Networks

Structured Data vs. Unstructured Data



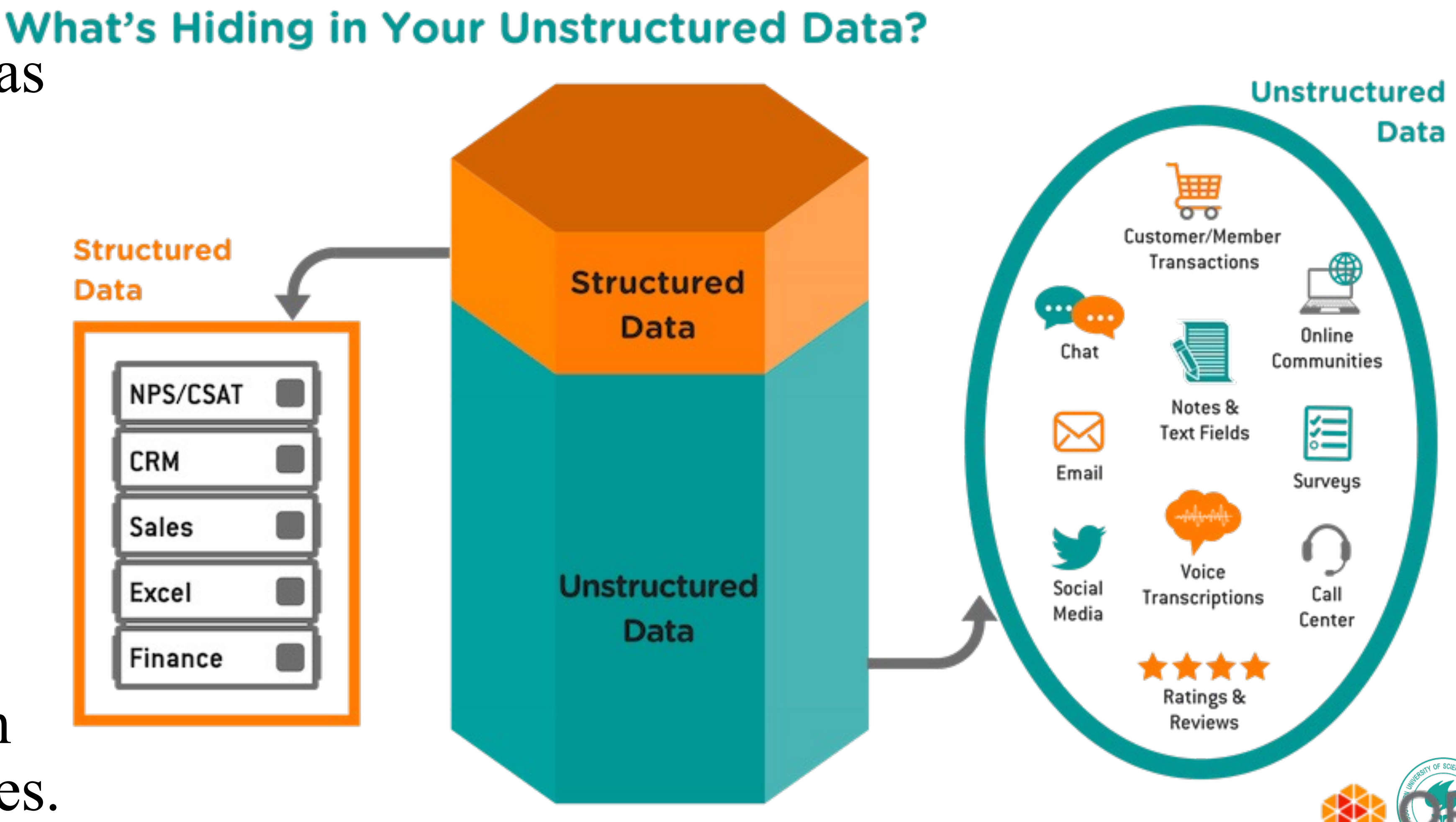


AncoraSIR.com

What is Data?

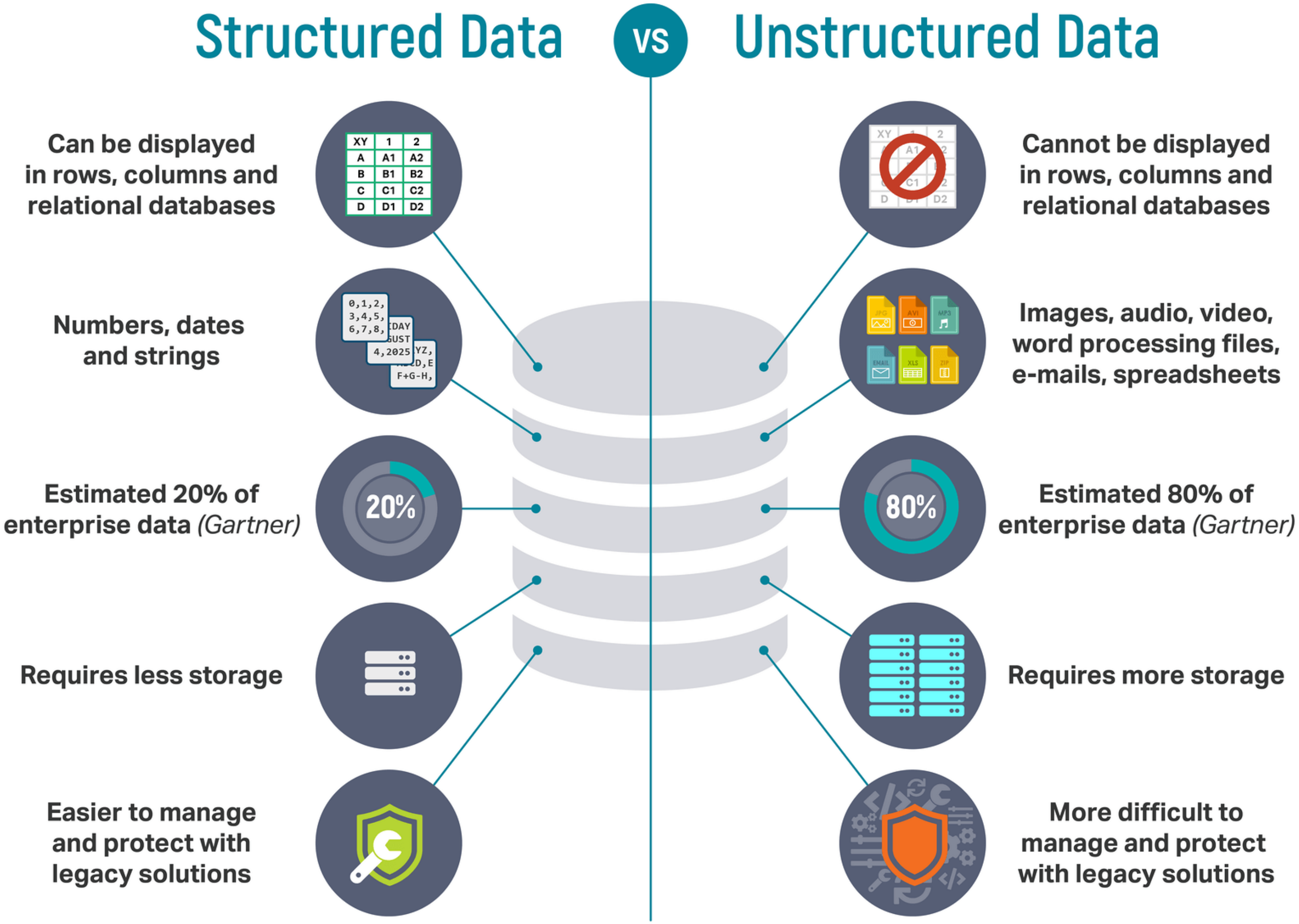
- Data can be defined as a representation of facts, concepts, or instructions in a formalized manner,

- Suitable for communication, interpretation, or processing by human or electronic machines.



Source: Graphic adapted from January 2018 CXPA Presentation "The Why Behind the What," Jim Kitterman

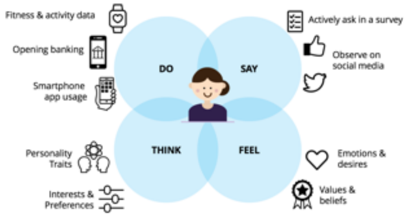
What is Data?

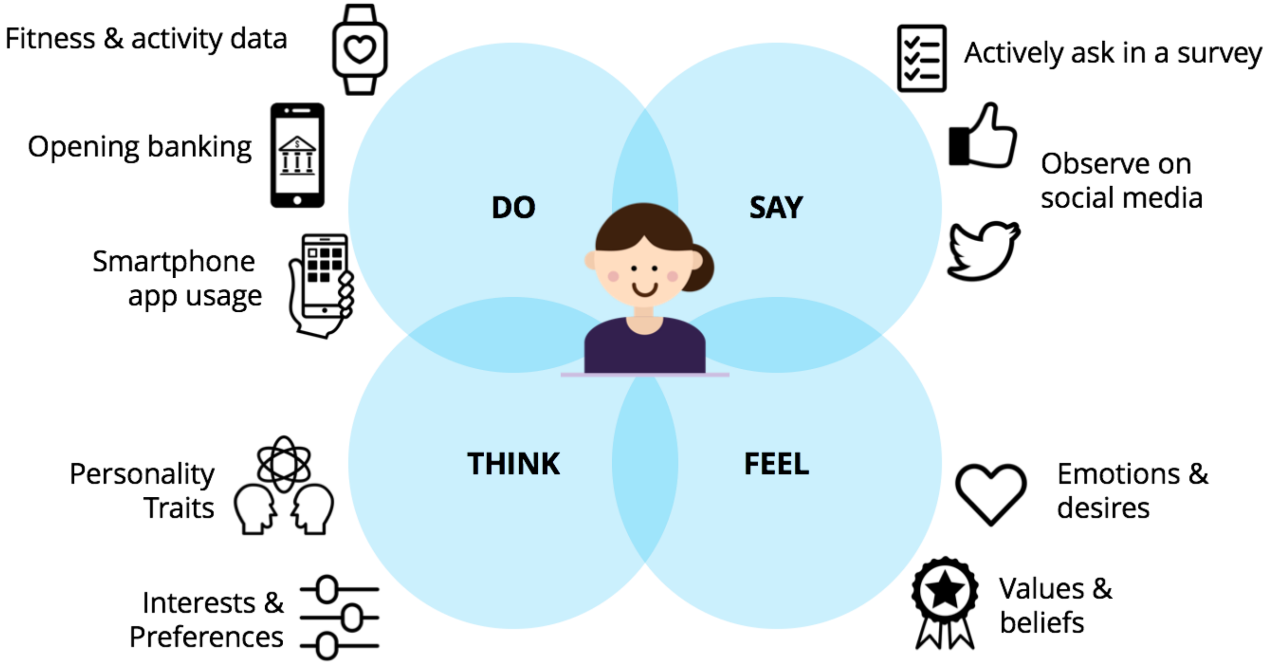


Structured Data
VS
Unstructured Data
Can be displayed in rows, columns and relational databases
Numbers, dates and strings
Estimated 20% of enterprise data (Gartner)
20%
Requires less storage
Easier to manage and protect with legacy solutions
Cannot be displayed in rows, columns and relational databases
Images, audio, video, word processing files, e-mails, spreadsheets
Estimated 80% of enterprise data (Gartner)
80%
Requires more storage
More difficult to manage and protect with legacy solutions

The Full Human Data Stack

Open Data	E.g. census data
Processed Algorithmic Data	E.g. your credit score
Shared Data	E.g. smart home data
Personal Data	

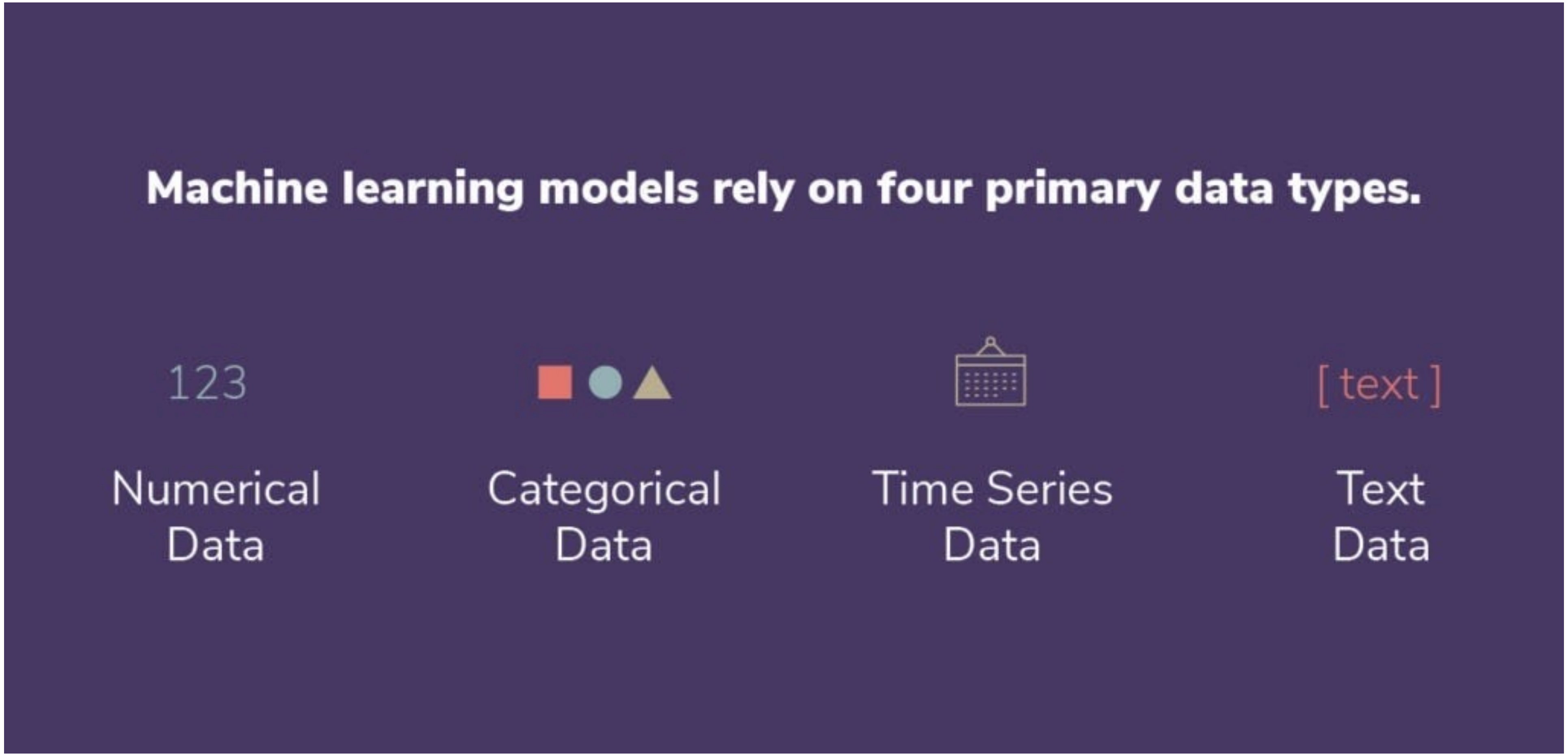




AncoraSIR.com

https://www.citizenme.com/human-data/

What type of data does machine learning need?



AncoraSIR.com

SUSTech
Southern University of Science and Technology

Source

Categorical Data

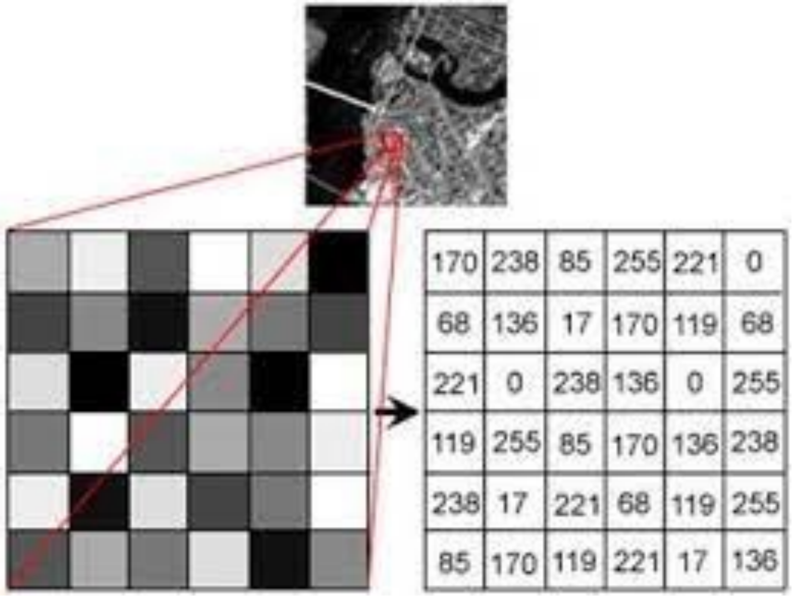
Color	Digitized
Red	0
Green	1
Blue	2

Hometowm	
Guangdong	0
Hunan	1
Fujian	2
…	…

Gender	
Female	0
Male	1
…	…

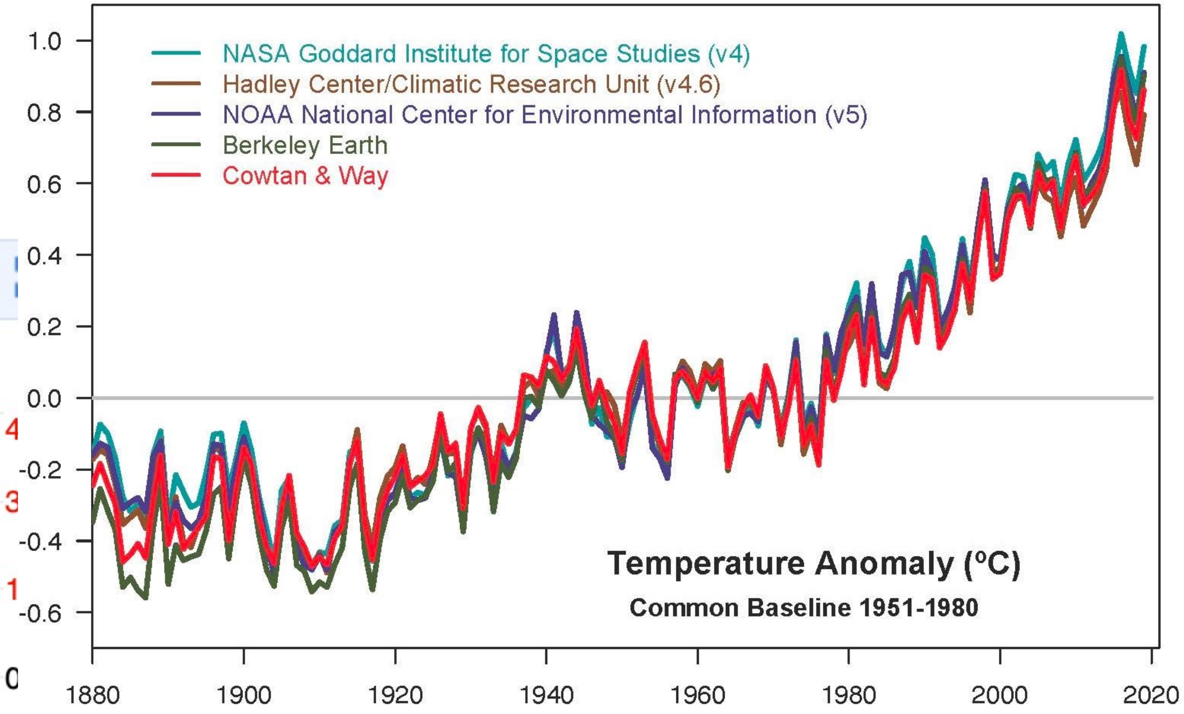
Numerical data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	DATA						RANK					TREND		
2	Sales Pers	May	June	July	Aug		May	June	July	Aug		June	July	Aug
3	1	84	138	72	45		35	2	42	60		↑	↓	↓
4	2	98	57	122	129		27	52	13	9		↓	↑	↑
5	3	83	108	107	107		36	19	20	20		↑	↓	→
6	4	91	135	120	56		30	3	14	54		↑	↓	↓
7	5	133	61	47	62		6	49	59	47		↓	↓	↑
8	6	73	80	86	113		41	37	33	18		↑	↑	↑
9	7	57	98	66	117		52	27	44	16		↑	↓	↑
10	8	86	52	134	132		33	56	5	8		↓	↑	↓
11	9	53	99	48	106		55	25	58	22		↑	↓	↑
12	10	96	80	59	69		29	37	50	43		↓	↓	↑
13	11	78	102	104	116		40	24	23	17		↑	↑	↑
14	12	133	119	90	89		6	15	31	32		↓	↓	↓
15	13	79	127	128	124		39	11	10	12		↑	↑	↓
16	14	49	66	64	62		57	44	46	47		↑	↓	↓
17	15	58	135	99	141		51	3	25	1		↑	↓	↑



SUSTech
Southern University of Science and Technology

Time series data



Text data


COLLECT TEXT DATA FROM ANY SOURCE INCLUDING NEWS, SOCIAL MEDIA, EMAILS, SURVEYS AND PRODUCT REVIEWS
NEWS
PDF


cultural
analysis
approaches
historical
texts
field
network
collections
Humanities
passages
literary
researcher
modeling
new
writing
sociological
large
information
interactions
expressive
scholars
tools
culture
work
social
literature
collection
data
text
vast
digital
domain
Europe
time
visualizations
visualization
textual
whole
virtual
interest
processes
produced
considerable
opportunities
approach
leading
forms
entities
unread
discover
statistical
STM
poets
challenges
understand
problems
movement
well-understood
research
bibliographic
specific
Mathematical
include
intellectual
dynamics
mathematics
modernism
machine
Rainey
great
modernist
topic
world
history
group
explore
accounts
across
individuals
program
patterns
corpus
influence
derive
many
wants
art
three
locations
sea
well
use

Where Do We Get Data for ML?

Five of the most popular ML dataset resources:

- Google ⟶ Google's Dataset Search
- Microsoft ⟶ Microsoft Research Open Data
- aws ⟶ Amazon Datasets
- UCI ⟶ UCI Machine Learning Repository
- DATA.GOV ⟶ Government Datasets

A Roadmap of Supervised Machine Learning

$$\hat{y} = g_{Activation}\left[f_{WeightedSum}(\mathbf{x})\right]$$

- Linear Regression
  - (Argurably) the simplest ML model
  - Basic concepts applicable to all ML problems
  - $\hat{y} = f_{WeightedSum}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
- Linear Classification
  - Vectorized weights for multiple classes
  - $\mathbf{s} = f_{WeightedSum}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
  - $\hat{\mathbf{y}} = g_{Activation}(\mathbf{s}) = ?$
- Single-neuron Perceptron
  - Binary LC using step activation
  - $s = f_{WeightedSum}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
  - $\hat{y} = g_{Activation}(s) = \text{step}(s, 0)$
- Logistic Regression
  - Binary LC using sigmoid activation
  - Binary output with a probability
  - $\hat{y} = g_{Activation}(s) = \text{sigmoid}(s)$
- Softmax Regressions
  - Multi-class LC using softmax activation
  - Multi-class output with a probability distribution
  - $\mathbf{s} = f_{WeightedSum}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
  - $\hat{\mathbf{y}} = g_{Activation}(\mathbf{s}) = \text{softmax}(\mathbf{s})$
- Multi-layer Perceptron
  - Neural network featuring hidden units
  - $\hat{\mathbf{y}}_N = g_{A_N}\left[f_{W_N}(\hat{\mathbf{y}}_{N-1})\right] \ldots \hat{\mathbf{y}}_1 = g_{A_1}\left[f_{W_1}(\mathbf{x})\right]$

AncoraSIR.com

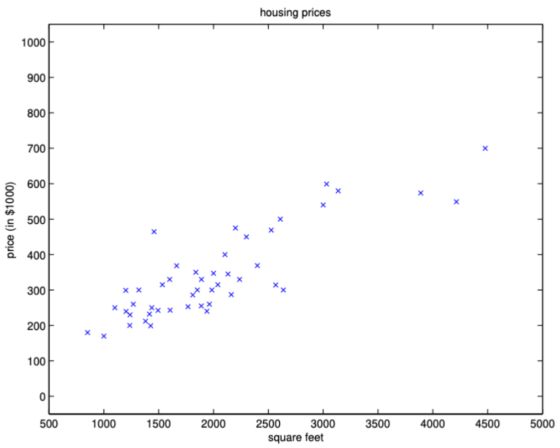
SUSTech Southern University of Science and Technology

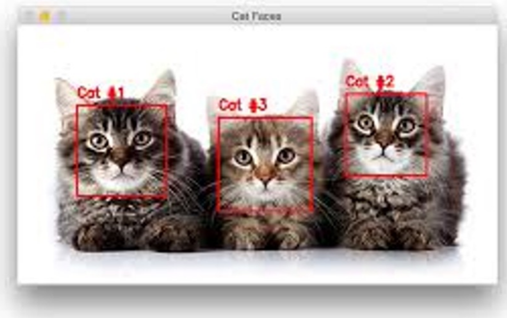
Regression vs. Classification

Classification vs. Regression

Continuous or Discrete Values

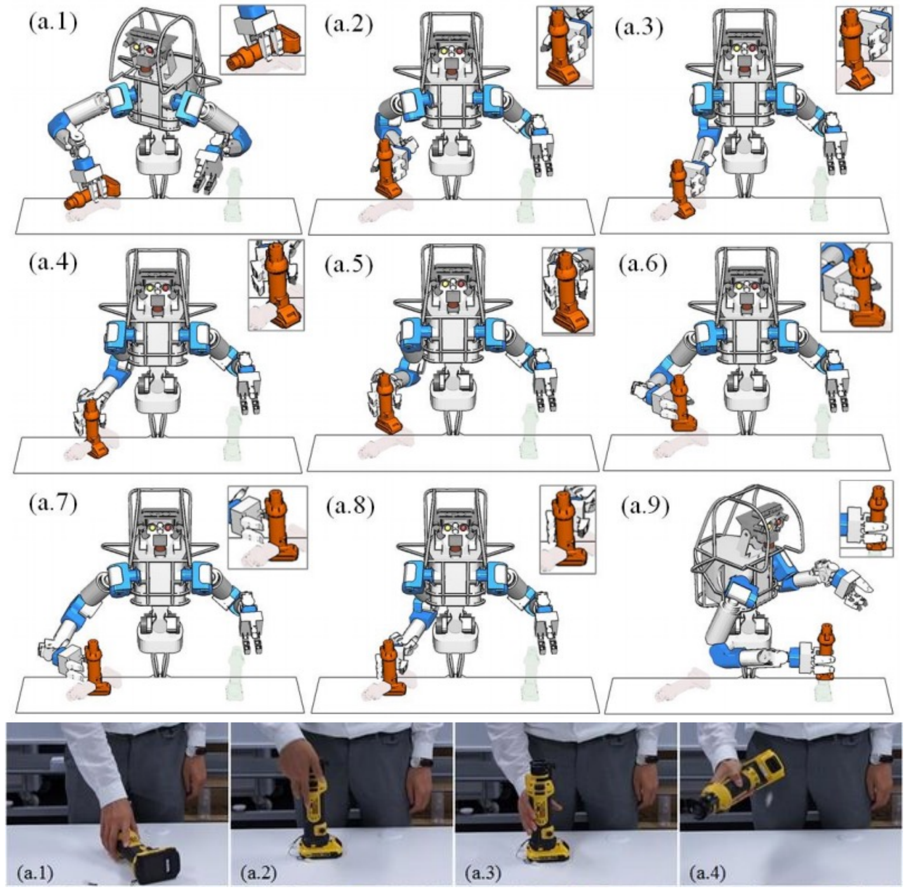
- Design a **Model** that defines the relationship between **Features** (input x_i) and **Labels** (output y)





- **Regression**: usually predicts continuous values.
    - *What is the value of a house in Shenzhen?*
    - *What is the probability that a user will click on this ad?*
- **Classification**: usually predicts discrete values.
    - *Is a given email spam or not spam?*
    - *Is this an image of a dog, a cat, or a hamster?*
- **Regression as classification**
    - *Scores higher than 60 gets a pass?*
    - *What's the probability of getting a pass?*
    - *How likely the robot's motion is similar to the human's motion?*

https://arxiv.org/pdf/1812.03274.pdf



AncoraSIR.com

Linear Regression

Arguably the simplest and most popular among the standard tools

- Linear Regression Assumption
  1. The relationship between the feature *x* and target *y* is linear

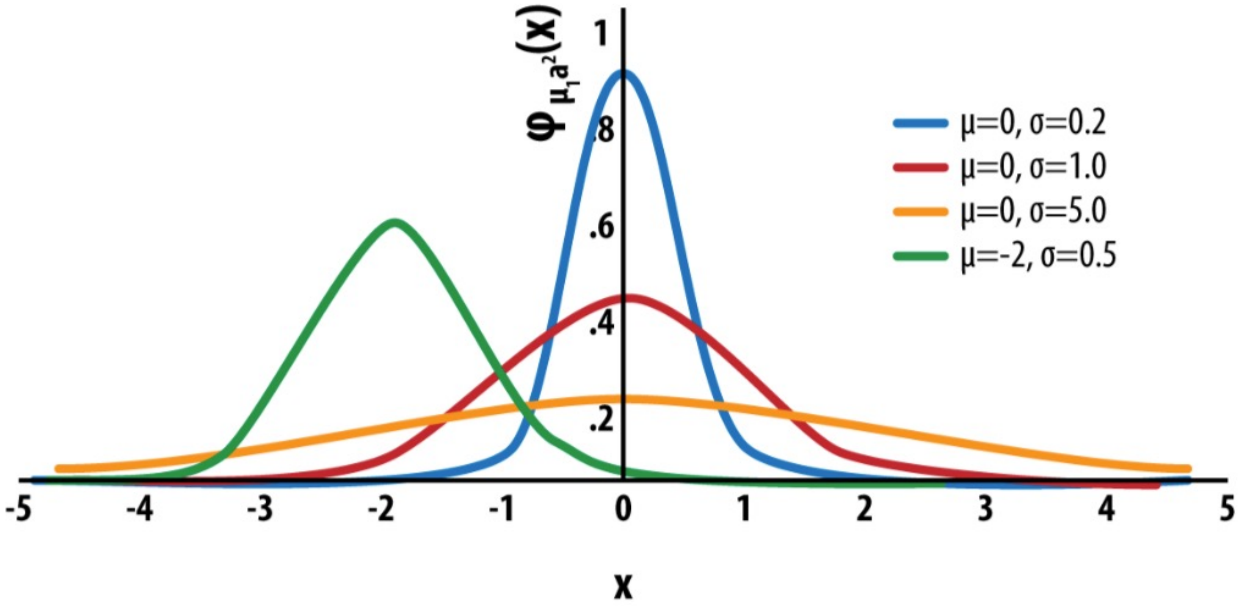
$$y = wx + b$$

learnable parameters that must be estimated from data

  2. Any noise is well-balanced, i.e. follows a Gaussian distribution

$$y = wx + b + N(0, \epsilon)$$

standard deviation of the noise term



$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\mu)^2\right)$$

AncoraSIR.com

Linear Model

The goal of linear regression

- **w**
  - The ***weight*** determines the influence of each feature on our prediction, usually a vector form with w_i
- b
  - The ***bias*** says what value the predicted price should take when all features take 0
- Given a dataset, **our goal** is
  - To choose the weights **w** and bias b such that on average, the predictions made based on our model best fit the true prices observed in the data.

$$\hat{y} = w_1 \cdot x_1 + \ldots + w_d \cdot x_d + b \longrightarrow \hat{y} = \mathbf{w}^T \mathbf{x} + b.$$

$$\hat{y}^i = w_1 x_1^i + w_2 x_2^i + \cdots + w_d x_d^i + b$$

index label data point

i y^i [x_1^i x_2^i $x_{\ldots}^i$ x_d^i]

City	Number of weekly riders	Price per week ($)	Population of city	Monthly income of riders ($)	Average parking rates per month ($)
1	192000	15	1800000	5800	50
2	190400	15	1790000	6200	50
3	191200	15	1780000	6400	60
4	177600	25	1778000	6500	60
5	176800	25	1750000	6550	60
6	178400	25	1740000	6580	70
7	180800	25	1725000	8200	75
8	175200	30	1725000	8600	75
9	174400	30	1720000	8800	75
10	173920	30	1705000	9200	80
11	172800	30	1710000	9630	80
12	163200	40	1700000	10570	80
13	161600	40	1695000	11330	85
14	161600	40	1695000	11600	100
15	160800	40	1690000	11800	105
16	159200	40	1630000	11830	105
17	148800	65	1640000	12650	105
18	115696	102	1635000	13000	110
19	147200	75	1630000	13224	125
20	150400	75	1620000	13766	130
21	152000	75	1615000	14010	150
22	136000	80	1605000	14468	155
23	126240	86	1590000	15000	165
24	123888	98	1595000	15200	175
25	126080	87	1590000	15600	175
26	151680	77	1600000	16000	190
27	152800	63	1610000	16200	200

Vectorization of a Linear Model

The goal of linear regression

$$\hat{y} = w_1 \cdot x_1 + \ldots + w_d \cdot x_d + b \rightarrow \hat{y} = \mathbf{w}^T\mathbf{x} + b \rightarrow \hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$$

- Vectorization
  - All features into a vector **x** for a single data point
  - All weights into a vector **w**
  - Our entire dataset as the *design matrix* **X,** including one row for every example and one column for every feature

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(i)} & \cdots & x_d^{(i)} \end{bmatrix}$$

one row for every example

one column for every feature

$$\hat{y}^i = w_1 x_1^i + w_2 x_2^i + \cdots + w_d x_d^i + b$$

index label data point

i y^i [x_1^i x_2^i $x_{\ldots}^i$ x_d^i]

City	Number of weekly riders	Price per week ($)	Population of city	Monthly income of riders ($)	Average parking rates per month ($)
1	192000	15	1800000	5800	50
2	190400	15	1790000	6200	50
3	191200	15	1780000	6400	60
4	177600	25	1778000	6500	60
5	176800	25	1750000	6550	60
6	178400	25	1740000	6580	70
7	180800	25	1725000	8200	75
8	175200	30	1725000	8600	75
9	174400	30	1720000	8800	75
10	173920	30	1705000	9200	80
11	172800	30	1710000	9630	80
12	163200	40	1700000	10570	80
13	161600	40	1695000	11330	85
14	161600	40	1695000	11600	100
15	160800	40	1690000	11800	105
16	159200	40	1630000	11830	105
17	148800	65	1640000	12650	105
18	115696	102	1635000	13000	110
19	147200	75	1630000	13224	125
20	150400	75	1620000	13766	130
21	152000	75	1615000	14010	150
22	136000	80	1605000	14468	155
23	126240	86	1590000	15000	165
24	123888	98	1595000	15200	175
25	126080	87	1590000	15600	175
26	151680	77	1600000	16000	190
27	152800	63	1610000	16200	200

AncoraSIR.com

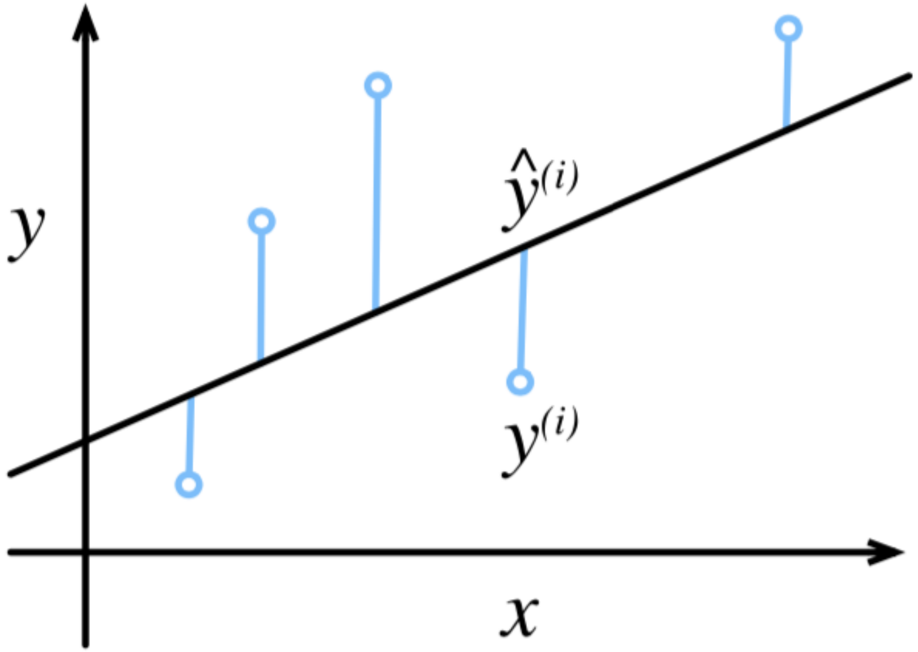
Loss Function

A quality measure for some given model

- To quantify the distance between the ***predicted*** and ***real*** value of the target.
  - usually be a non-negative number where smaller values are better
  - perfect predictions incur a loss of 0
- The Sum of Squared Errors $l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$
  - the empirical error is only
    a function of the model parameters
- Loss Function as an averaged SSE

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

$$\mathbf{w}^*, b^* = \underset{\mathbf{w}, b}{\operatorname{argmin}}\ L(\mathbf{w}, b)$$

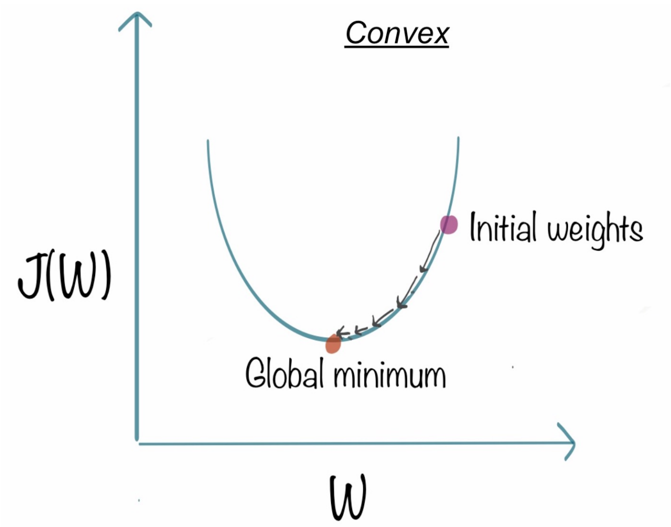


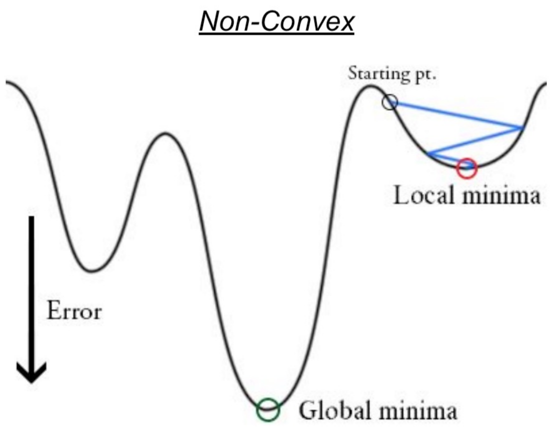
AncoraSIR.com

Gradient Descent

A procedure for updating the model parameters to improve its quality

- **Iteratively reducing** the error by updating the parameters in the direction that incrementally lowers the loss function, or Gradient Descent
  - On *convex* loss surfaces, it will eventually converge to a global minimum
  - For *nonconvex* surfaces, it will at least lead towards a (hopefully good) local minimum.





- The key technique for optimizing *nearly any* deep learning model

AncoraSIR.com

Stochastic Gradient Descent

a more efficient practice

- Sampling a random minibatch of examples every time we need to computer the update
  - Initialize model parameters at random;
  - Iteratively sample random batches to update the parameters in the direction of the negative gradient

learning rate

$$(\mathbf{w}, b) \leftarrow (\mathbf{w}, b) - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{(\mathbf{w}, b)} l^{(i)}(\mathbf{w}, b)$$

batch size: the number of examples in each minibatch

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b) = w - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{x}^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right),$$

$$b \leftarrow b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_b l^{(i)}(\mathbf{w}, b) = b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right).$$

- Hyperparameters
  - The values of the batch size and learning rate are manually pre-specified and not typically learned through model training.
  - Tunable but not updated in the training loop.

Maximum Likelihood Estimation

Assume that observations arise from normally distributed noisy observations

- The best values of *b* and *w* are those that maximize the likelihood of the entire dataset

$$y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$P(Y \mid X) = \prod_{i=1}^{n} p(y^{(i)} | \mathbf{x}^{(i)})$$

- Maximizing the product of many exponential functions is *difficult*

- The likelihood of seeing a particular *y* for a given *x*

$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

$$-\log p(\mathbf{y}|\mathbf{X}) = \sum_{i=1}^{n} \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)} - b\right)^2$$

Negative Log-Likelihood (NLL)

If a constant

SSE

*Why **minimizing squared error** is equivalent to **maximum likelihood estimation** of a linear model under the assumption of additive Gaussian noise?*

AncoraSIR.com

Linear Classification

How to scientifically calculate a decision

- Hypothesis
  - Acceptance depending on Test and Grade
- Data
  - i sets of example data $\left(x^{(i)}, y^{(i)}\right)$
- Input
  - $x_1^{(i)}$ as test scores and $x_2^{(i)}$ as test scores
- Output
  - $\hat{y}^{(i)}$ as a threshold decision of Accept or Reject
- Model
  - A linear boundary line to separate the data
    - $w_1 x_1 + w_2 x_2 + b = 0$
  - A threshold to activate a decision against the line
    - > 0: Accept; < 0: Reject
- Learning
  - Obtain a set of w_i and b with small enough $y^{(i)} - \hat{y}^{(i)}$

An example of acceptance at a University

A Linear Boundary Line of $2x_1 + x_2 - 18 = 0$ as a decision criteria from regression to classification

AncoraSIR.com

An Example of Linear Classification with Images

A data-driven approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

A General Problem Statement

- Given
  - A **score function** that maps the raw data to class scores,
  - A **loss function** that quantities the agreement between the predicted scores and the ground truth labels.
- Goal
  - As an **Optimization Problem** in which we will minimize the loss function with respect to the parameters of the score function.

AncoraSIR.com

An Example of Linear Classifier for Images

A data-driven approach for linear classification

- Data
  - i sets of labelled image data $\left(x^{(i)}, y^{(i)}\right)$
- Hypothesis
  - Image features provides the data for classification
- Input
  - $x^{(i)}$ of image pixels,
  - i.e., arrays of 32x32x3 numbers
- Output
  - $\hat{y}^{(i)}$ as predicted classification of the image
  - i.e., a 10x1 vector with scores for each entry
- Model
  - A score function of weighted-sum
    - $f(\boldsymbol{x}, \mathbf{W}) = \mathbf{W}\,\boldsymbol{x} + b$
- Learning
  - An optimization algorithm that updates the the weight **W** (10x3072) and bias b (10x1) by minimizing a loss function

Array of **32x32x3** numbers (3072 numbers total)

3072x1

$f(\boldsymbol{x}, \mathbf{W}) = \mathbf{W}\,\boldsymbol{x} + b$ 10x1

10x1 10x3072

→ $f(\boldsymbol{x}, \mathbf{W})$ → **10** numbers giving class scores

↑

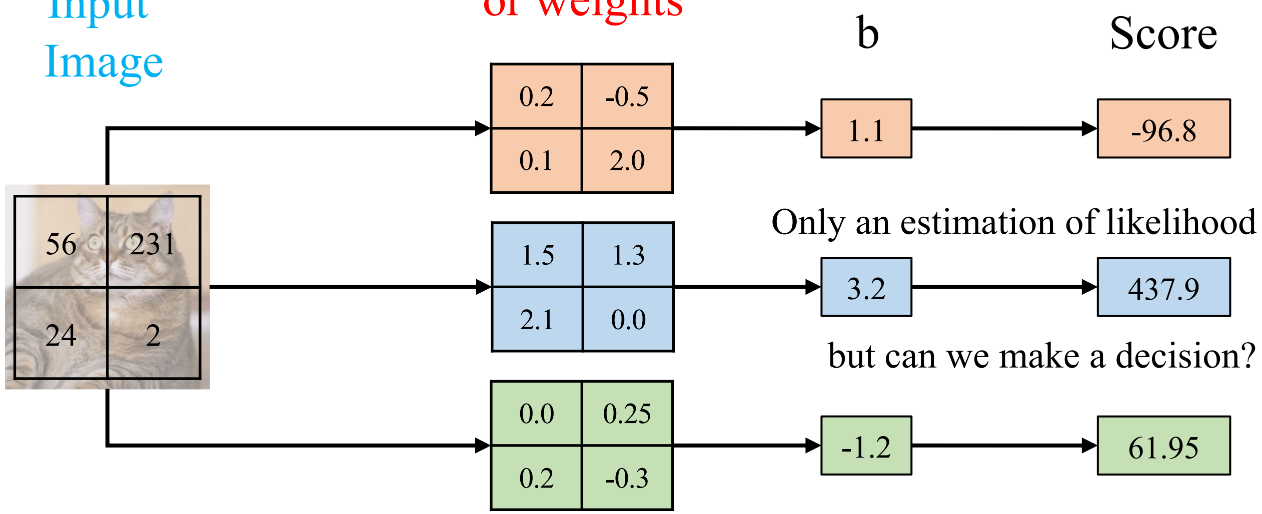
W

Parameters or weights

A (linear) score function

- *How can we tell whether this W is good or bad?*

Input Image



Only an estimation of likelihood

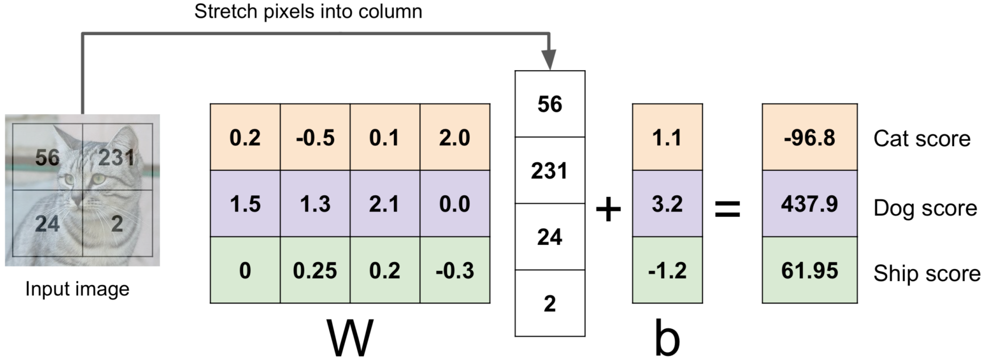
but can we make a decision?

AncoraSIR.com

Three Viewpoints of Image Classification

Strategies for making a decision based on weighted sum of the image features

Algebraic $f(\boldsymbol{x}, \mathbf{W}) = \mathbf{W}\,\boldsymbol{x} + b$



$$f(\boldsymbol{x}, \mathbf{W}) = \mathbf{W}\,\boldsymbol{x} + b$$

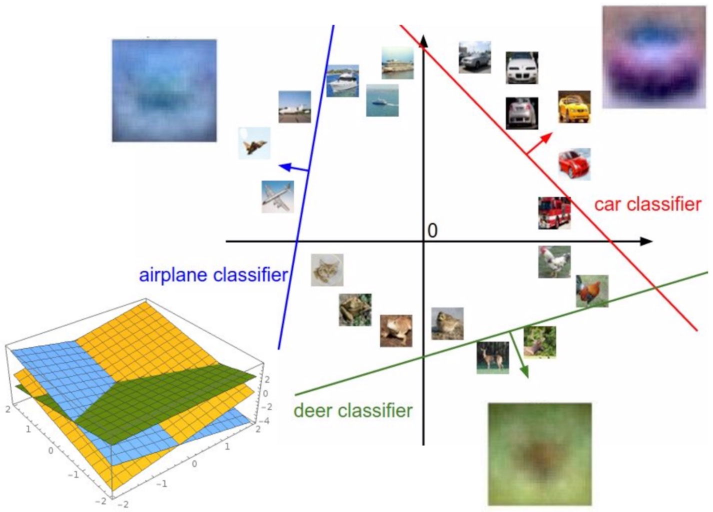
$f(\boldsymbol{x}, \mathbf{W})$: 3x1; $\mathbf{W}$: 3x2; $\boldsymbol{x}$: 2x1; b: 3x1

$$W = \begin{bmatrix} w_1^{(1)} & \cdots & w_d^{(1)} \\ \vdots & \ddots & \vdots \\ w_1^{(i)} & \cdots & w_d^{(i)} \end{bmatrix}$$

one row for each class

one column for every feature

Geometric



Hyperplanes cutting up space

Visual One template per class

AncoraSIR.com

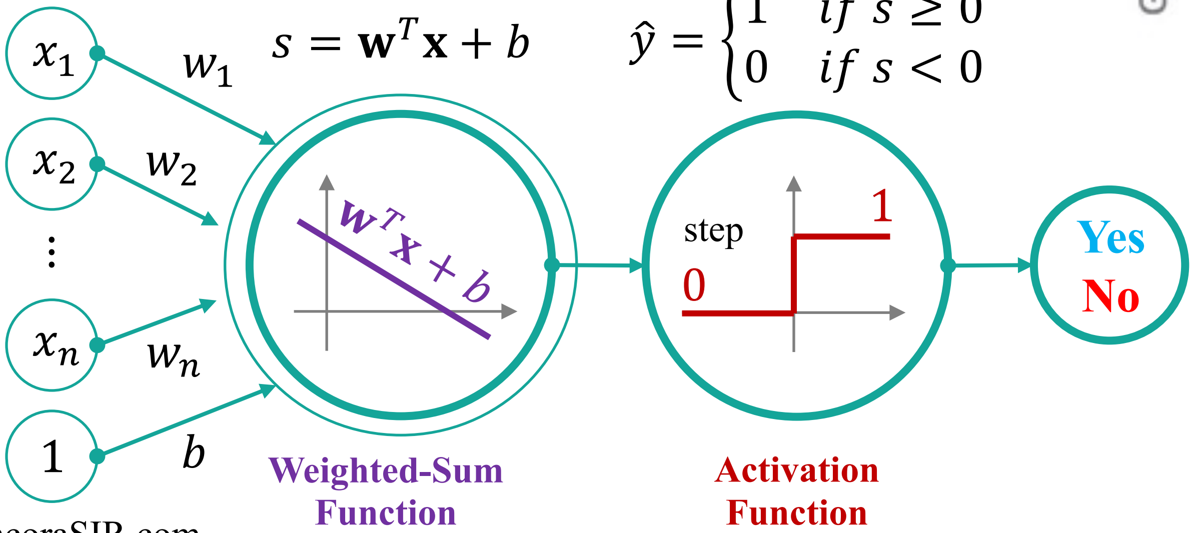
SUSTech Southern University of Science and Technology

Perceptron with an Activation Function

Linear extraction of information from the data to help making a decision

- An Artificial Neuron with two nodes
  - **Weighted-sum node**
    - Calculate a linear equation $s(x)$ with inputs on the weights plus bias
  - **Activation node**
    - Apply the step function to get the predicted result $\hat{y}(s)$

$$s = \mathbf{w}^T\mathbf{x} + b \qquad \hat{y} = \begin{cases} 1 & if\ s \geq 0 \\ 0 & if\ s < 0 \end{cases}$$



AncoraSIR.com

<u>*An example of acceptance at a University*</u>

$$\begin{aligned}\hat{y} &= g_{Activation}\left[f_{WeightedSum}(\mathbf{x})\right] \\ &= \text{step}(s, 0) \\ &= \text{step}(\mathbf{w}^T\mathbf{x} + b, 0)\end{aligned}$$

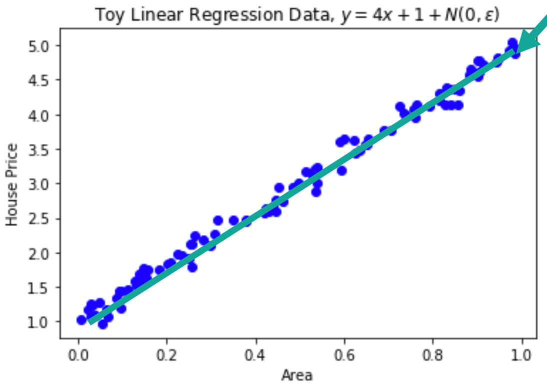
SUSTech Southern University of Science and Technology

Linear Regression and Classification

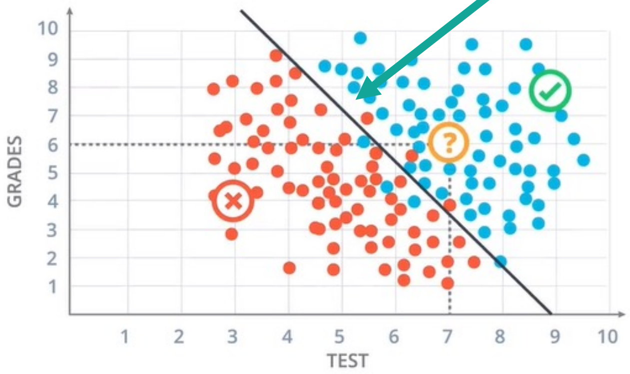
A summary

- Linear Regression
  - A basic linear model for line-fitting
  - $\hat{y} = f_{WeightedSum}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

Estimate a line
to describe a relationship



Predict University Acceptance
based on Test and Grades
(*only two categories*)

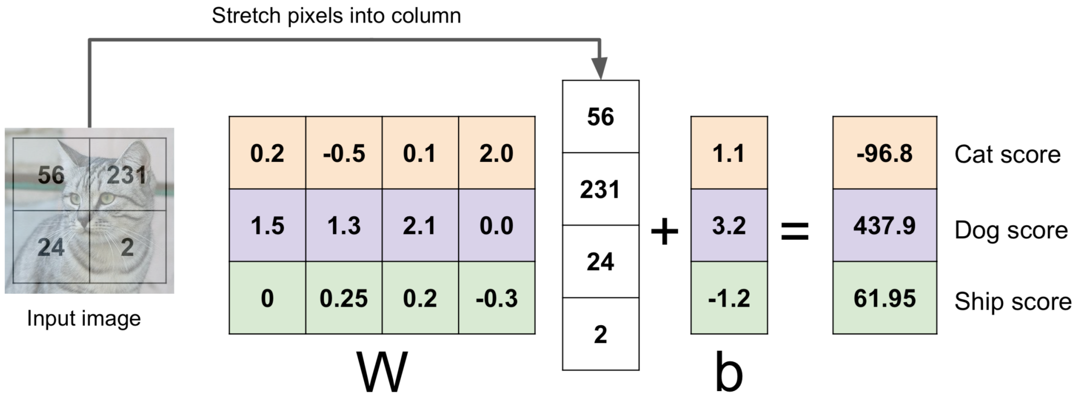


$$\hat{y} = g_{Activation}(s) = \begin{cases} 1 & if\ s \geq 0 \\ 0 & if\ s < 0 \end{cases}$$

- What if the problem becomes more complex?

- Linear Classification
  - Vectorized weights for two or multiple classes
  - $\mathbf{s} = f_{WeightedSum}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$

Is this picture a cat, a dog, or a ship?
(*Can we make a decision based on the results on the right?*)



- Information lost about the distance to the cutoff value
- Uncertain about the final decision

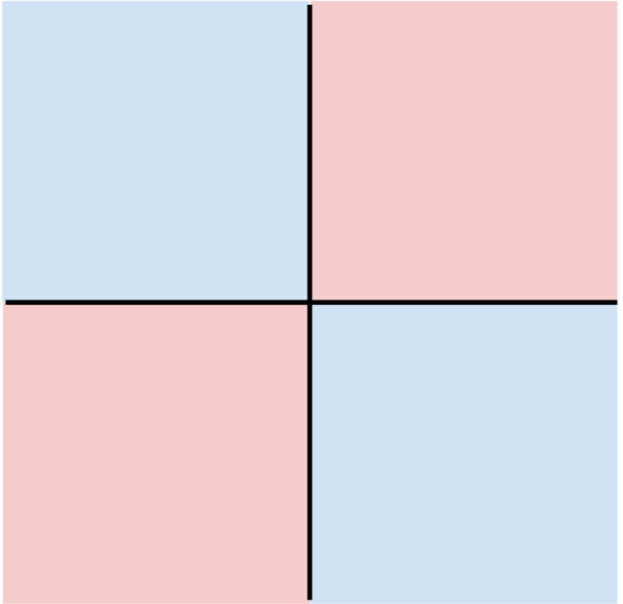
AncoraSIR.com

Hard Cases for a Linear Classifier

Simple linear classifiers are not enough to make a complex decision

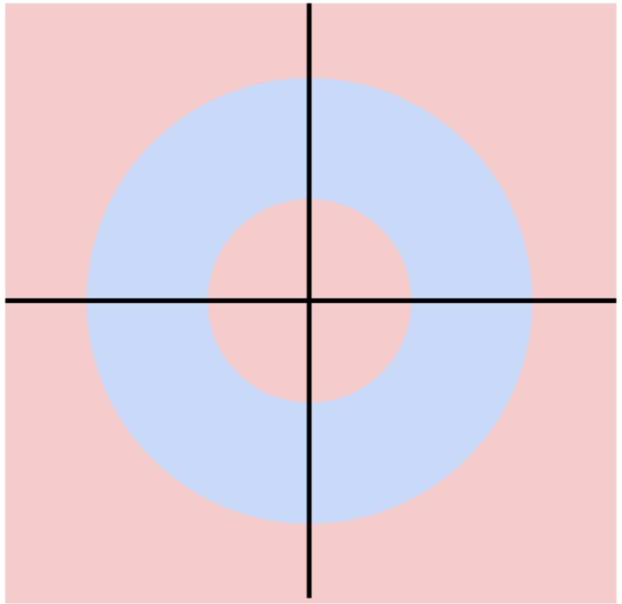
Class 1:
First and third quadrants

Class 2:
Second and fourth quadrants



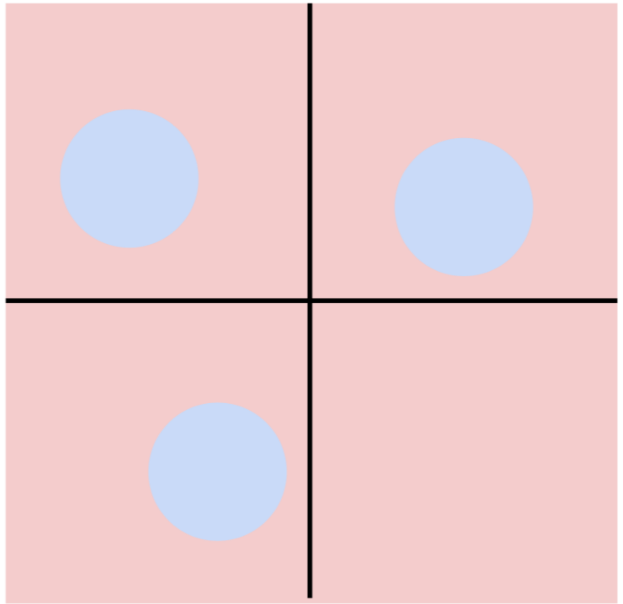
Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else



Class 1:
Three modes

Class 2:
Everything else



AncoraSIR.com

Week 05 | Lecture 06
Regression in Machine Learning

Thank you~

Wan Fang
Southern University of Science and Technology