6CS012-Artificial Intelligence and Machine Learning
Lecture-02

Learning→ Artificial Intelligence
Understanding the Components of Learning: A Classification Perspective.

Siman Giri

Learning outcomes for the Week!!!

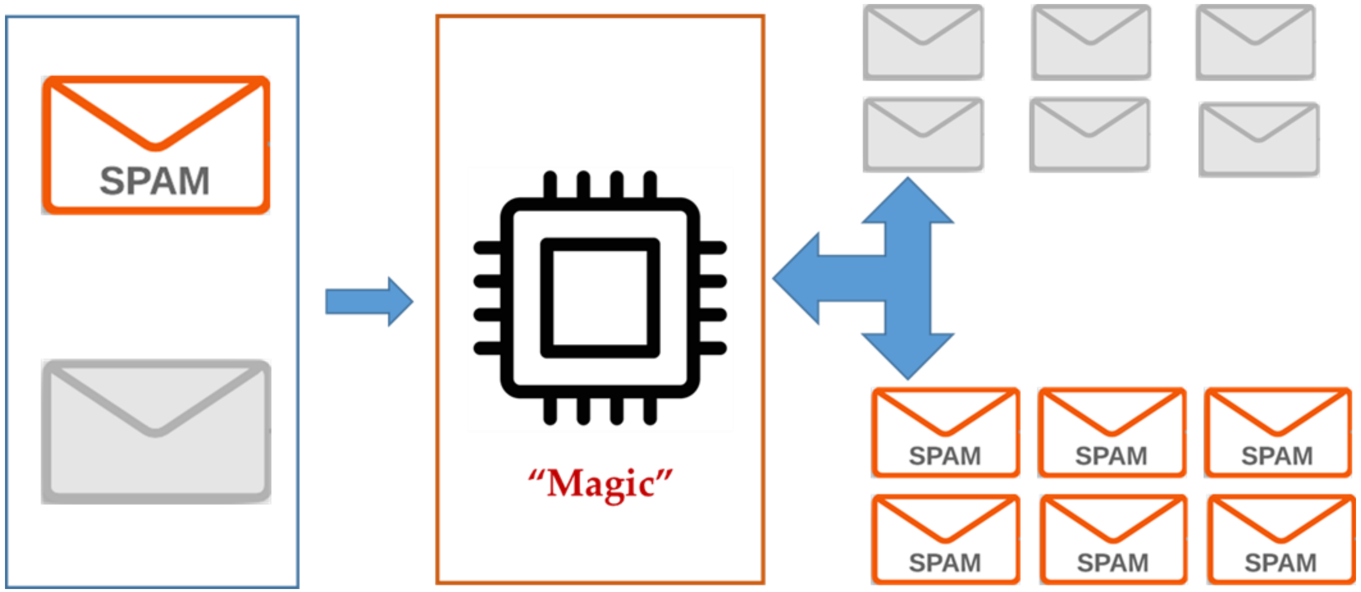
- To **revise** the various **components** of (Machine) **Learning** that we discuss **@5CS037**.
- To **review and re-familiarize** above mentioned **components** with the context of **Classification task →** "**Logistic Regression**".
- To able to **differentiate** between **Machine Learning and Deep Learning**.

A review on (Machine) Learning!!

1.What is Learning?

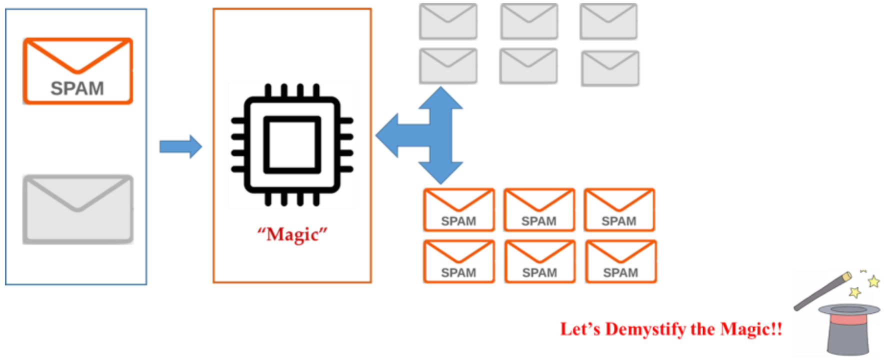
1.1 What is Learning? Intuition.

- Task-Example: Identify the spam emails!!!
- (Program a machine that learns how to filter spam emails.)



Let's Demystify the Magic!!

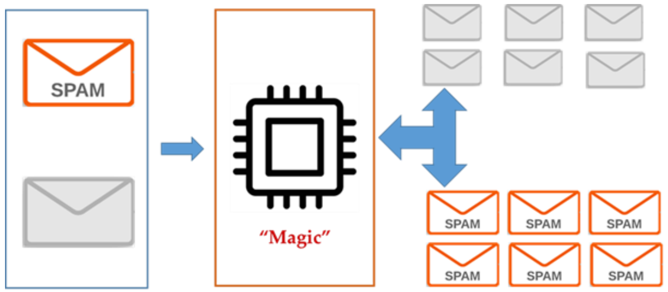
1.2 Demystifying Magic-1: Expert System.



- **Example: Identify the spam emails!!!**
- **(Program a machine that learns how to filter spam emails.)**
  - **Expert-System:**
  - In the early days of **"intelligent"** applications, many systems used **hand-coded rules** of **"if"** and **"else"** decisions to **process data** or **adjust to user input**.
  - A naïve solutions: machine can simply **make a array of all the words**, appearance of whose result in an **email being spam**, when a **new email arrives**, machine can check for those **blacklisted word from array**. If it matches one of them, it can be assigned as **spam** otherwise can be moved to **inbox**.
  - This would be an example of using an expert-designed rule system ("learning by memorization") to design an **"intelligent"** application.

1.2 Demystifying Magic-1: Expert System.

- **Expert-System ~ learning by memorizations:**
  - In our example - "**learning by memorization**" approach might work well but it lacks one important aspects of learning systems
    - – the ability to **label unseen email-messages** i.e**. email messages which may be spam but does not contain any of the word in the black-list(array) will be delivered to our inbox**.



- **Manually crafting decision rules** is **feasible for some application**, but has following **two disadvantages**:
  - The **logic** required to **make a decision** is specific to a **single domain** and **task**. **Changing the task** even slightly might required to **rewrite** of the **whole system**.
  - **Designing rules** requires a **deep understanding** of how a **decision should be made by a human expert.**
  - **{We did not learn from the data! instead we memorize a features of data.}**

1.3 When do we need Learning?

- A **successful learning system** must be able to **progress from individual examples** to **broader generalization**
  - – also referred as "**inductive reasoning**" or "**inductive inference**".
- **Example1: detect cat in an image.**

- **Challenges with Expert System**:
  - way in which **pixels** (~ which make up an image in a computer) are "**perceived**" by the **computer** is very different from how **humans perceive** a **face**.
  - This difference in representation makes it basically **impossible for a human** to come up with a **good set of rules to describe** what **constitutes a cat in a digital image**.
  - **Using machine to learn**:
    - however, simply presenting a program with a large collection of images of faces is enough for an algorithm to determine what characteristics are needed to identify a face.
  - **{learning from data ~ What does it means to learn from data?}**

1.4 (Machine/Deep) Learning: Definition.

- Machine/Deep learning is a sub-domain of artificial intelligence (AI) that utilizes **Statistics, Pattern recognition, knowledge discovery and data mining** to **automatically learn and improve with experiences** without **being explicitly programmed**.
- Disclaimer!!
  "In Machine/Deep Learning we do not write a program to solve a specific problem or task instead we write a code/program to facilitate machine to learn from the data."
- Almost any application that involves **understanding data or signals** that come from the real world can be best **addressed using machine learning**.
- Great examples are face detection and speech recognition and many kinds of language-processing tasks.

1.5 (Machine/Deep) Learning: Premises.

- **When and Why do we build Machine Learning System?**
  - There exists some **pattern/behavior** of interest:
    (Some Task to be solved)
  - The **pattern/behavior** is difficult to **describe:**
    (Encoding a rule to understand a behavior is difficult)
  - There is **data**
    (past experiences are in abundant)
  - Use data to **“learn”** the pattern

1.6 (Machine/Deep) Learning : Cautions!!

- Machine/Deep learning is a very general and useful framework, but it is not **"magic"** and **may not always work**.
  - In order to better understand when it will and when it will not work, it is useful to **formalize** the **learning problem** more.
- **Some challenges of Machine/Deep Learning:**
  - **Why do we think that previously seen data will help us predict/infer the future?**
  - **estimation:**
    - **When we have data that are noisy reflections of some underlying quantity of interest, we have to aggregate the data and make estimates or predictions about the quantity.**
      - **How do we deal with the fact that, for example, the same treatment may end up with different results on different trials?**
      - **How can we predict how well an estimate may compare to future results?**
  - **generalization:**
    - **How can we predict results of a situation or experiment that we have never encountered before in our data set?**

Components of Learning.

data + model $\xrightarrow{\text{compute}}$ inference

2. Data and Learning Paradigm.

2.1 Data – Basic Overview and Definitions.

- **"Data"** :a **collection of facts** about any **objects or phenomenon**.
    - Facts/Measurements can be of quantitative(numeric) or qualitative(descriptive) in nature.
    - **Variables** and **Measurements**
- Some similar definitions:
    - Factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation
    - Information in digital form that can be transmitted or processed
    - Information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful **Cautions!!!!**

Datum

A single piece of information, which can be treated as an observation

Data

The plural of datum; multiple observations

Dataset

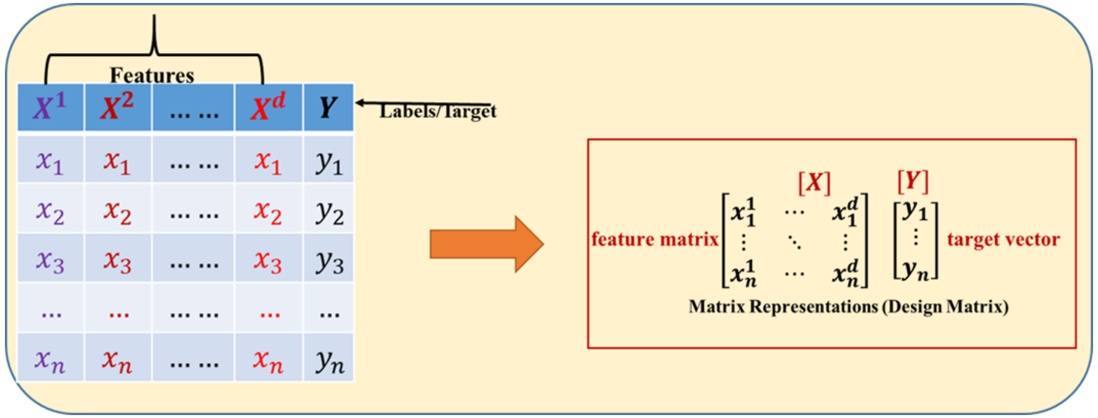
A homogenous collection of data (each datum must have the same focus)

Fig: Data(sets) Format.

2.2 Dataset Formats: In Practice.

- Some Terminology associated with dataset in practice:
- Variables:
    - **Target or output variables also referred as dependent variables.**
    - **Predictor, Feature or input variables also referred as independent variables**
- Notations:
    - Feature Variables: x or X.
    - Actual Target Variables: y or Y.
    - Predicted Target Variables: $\hat{y}$ or $\hat{Y}$.

Machine Learning



Deep Learning

2.3 Framing a Learning Problem

- Learning Problem(Tasks) in Machine Learning depends on type of the data we have:
- Datasets are made up of individual examples that contain features and a label.
  - Examples that contain both features and a label are called **labeled datasets**.
  - Examples that contain only features are called **unlabeled datasets**.

Features | Label

date	lat	long	temp	humidity	cloud_coverage	wind_direction	atmp_pressure	rainfall
2021-09-09	49.71N	82.16W	74	20	3	N	18.6	.01
2021-09-09	32.71N	117.16W	82	42	6	SW	29.94	.23

Example

Supervised Learning Techniques.

Features

date	lat	long	temp	humidity	cloud_coverage	wind_direction	atmp_pressure
2021-09-09	49.71N	82.16W	74	20	3	N	18.6
2021-09-09	32.71N	117.16W	82	42	6	SW	29.94

Example

Un-Supervised Learning Techniques.

data + model $\xrightarrow{\text{compute}}$ inference

2.4 Supervised Machine Learning.

- **Data in Supervised Learning:**
  - For Supervised Learning Setup, **training data** comes in pairs of inputs **(x, y):** where $X \in R^d$ is the input instance and Y its label, which can be written as:
    - $D = \{(x_1, y_1) \dots (x_n, y_n)\} \subseteq R^d * C$
    - Where:
      - R^d: **d-dimensional feature space.**
      - x_i: **input vector of the** i^{th} **sample.**
      - y_i: **label of the** i^{th} **sample.**
      - C: **label space.**
- **Tasks in Supervised Learning:**
  - There can be multiple scenario for the label space c.

Binary Classification	$c = \{0 \; or \; 1\}$	E.g.: An email is either spam or not a spam.
Multi Class Classification	$c = \{1,2,..k\}(k \geq 2)$	E.g.: Traffic sign Classification.
Regression	$c = \mathbb{R}$	E.g.: Height of the person.

2.4 Supervised Machine Learning: Examples.

Regression

- House Price Prediction:

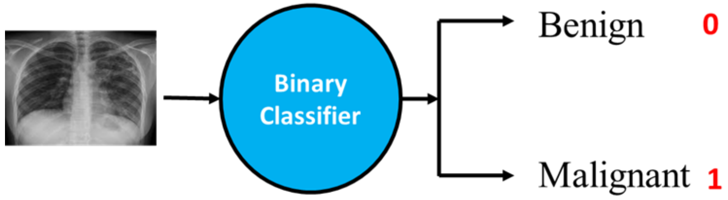
Street	OverallCond	YearBuilt	YearRemo	MasVnrArea	TotalBsmt	Heating	CentralAir	BsmtFullB	FullBath	HalfBath	Bedroom	SaleCondition	Price
Pave	6	1961	1961	0	882	GasA	Y	0	1	0	2	Normal	11622
Pave	6	1958	1958	108	1329	GasA	Y	0	1	1	3	Normal	14267
Pave	5	1997	1998	0	928	GasA	Y	0	2	1	3	Normal	13830
Pave	6	1998	1998	20	926	GasA	Y	0	2	1	3	Normal	9978
Pave	5	1992	1992	0	1280	GasA	Y	0	2	0	2	Normal	5005
Pave	5	1993	1994	0	763	GasA	Y	0	2	1	3	Normal	10000
Pave	7	1992	2007	0	1168	GasA	Y	1	2	0	3	Normal	7980
Pave	5	1998	1998	0	789	GasA	Y	0	2	1	3	Normal	8402
Pave	5	1990	1990	0	1300	GasA	Y	1	1	1	2	Normal	10176
Pave	5	1970	1970	0	882	GasA	Y	1	1	0	2	Normal	8400
Pave	5	1999	1999	0	1405	GasA	Y	1	2	0	2	Normal	5858
Pave	5	1971	1971	504	483	GasA	Y	0	1	1	2	Normal	1680
Pave	5	1971	1971	492	525	GasA	Y	0	1	1	3	Normal	1680
Pave	6	1975	1975	0	855	GasA	Y	0	2	1	3	Normal	2280
Pave	6	1975	1975	0	836	GasA	Y	0	1	0	2	Normal	2280
Pave	5	2009	2010	162	1590	GasA	Y	0	2	1	3	Partial	12858
Pave	5	2009	2010	256	1544	GasA	Y	0	2	0	3	Partial	12883
Pave	5	2005	2005	615	1698	GasA	Y	0	2	0	3	Normal	11520
Pave	5	2005	2006	240	1822	GasA	Y	0	2	0	3	Normal	14122
Pave	5	2003	2004	1095	2846	GasA	Y	1	2	1	3	Normal	14300
Pave	5	2002	2002	232	1671	GasA	Y	1	2	1	3	Normal	13650
Pave	5	2006	2006	178	1370	GasA	Y	0	2	0	2	Normal	7132
Pave	5	2005	2005	0	1324	GasA	Y	0	2	0	3	Normal	18494
Pave	5	2006	2006	14	1145	GasA	Y	0	2	0	2	Normal	3203
Pave	5	2004	2004	0	384	GasA	Y	1	2	1	3	Normal	13300

Classification

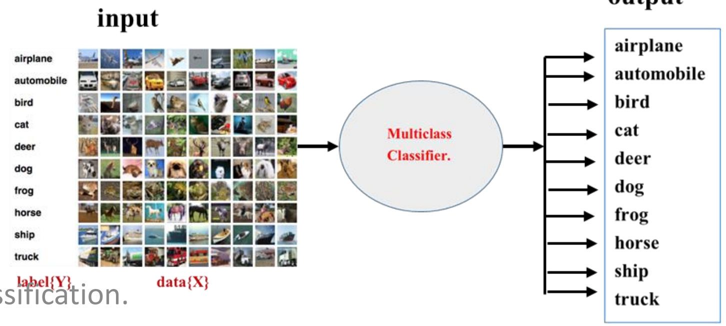
- Tasks in Supervised Learning-Classification Task:
  - **Binary Classification.**
  - **Multi-Class Classification.**

Input:

Output:

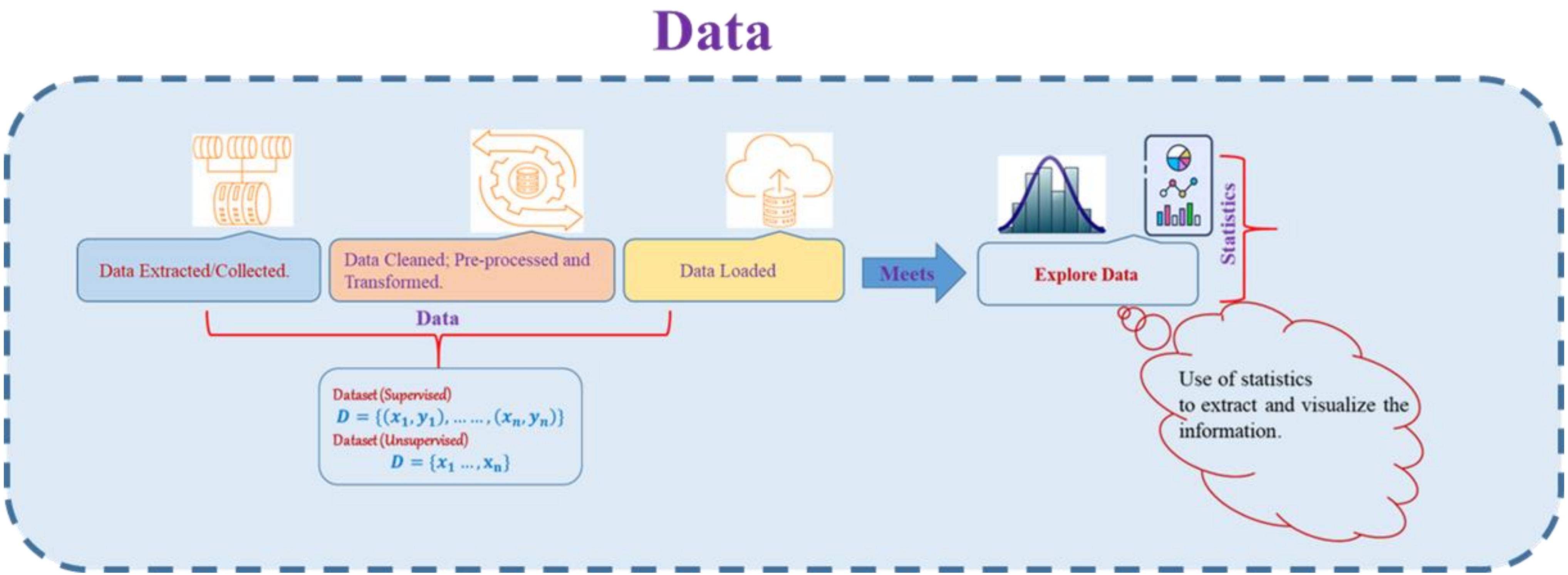


Binary Classification



Multiclass Classification

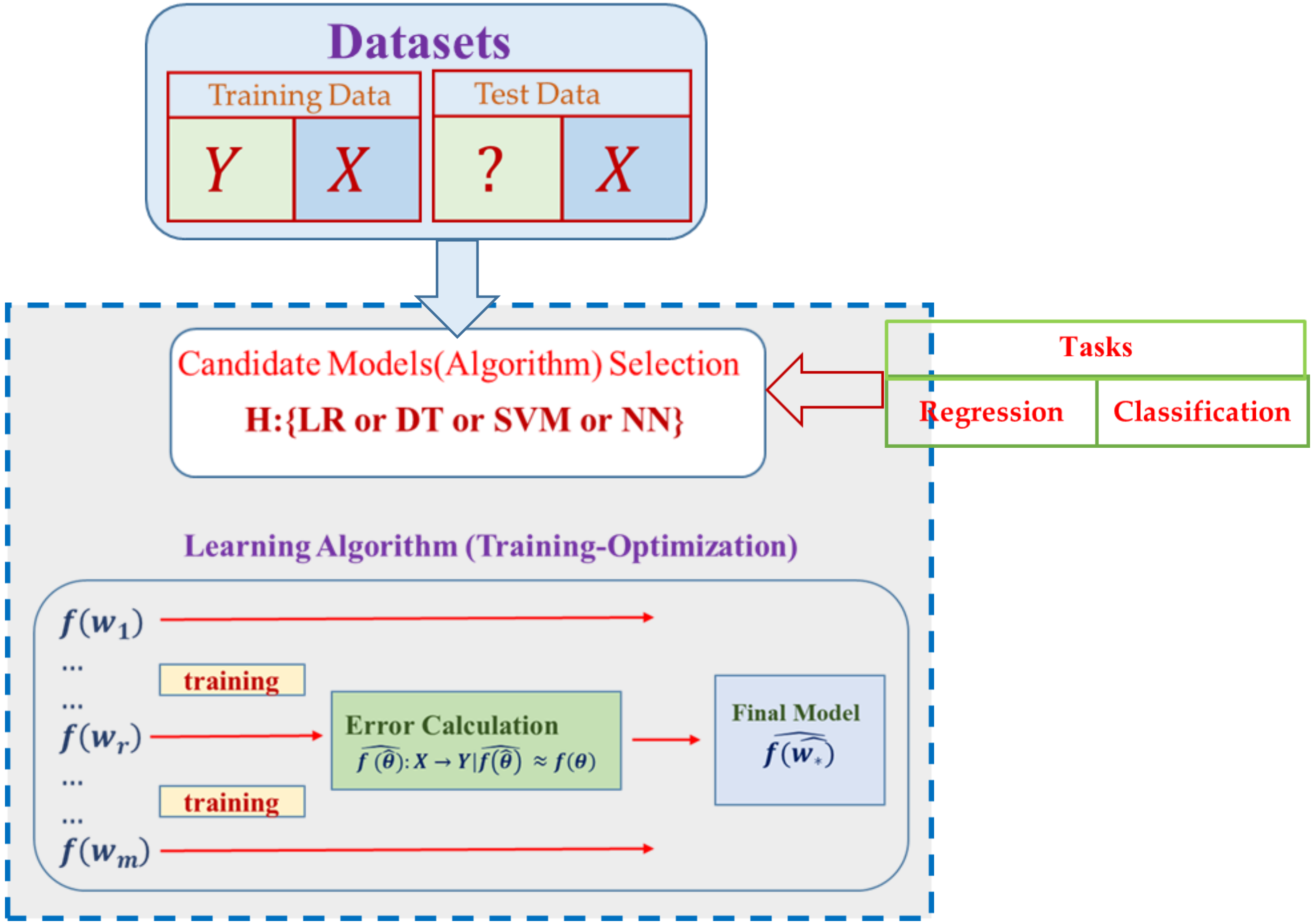
2. Summary: Dataset.



data + model compute inference

2.5 Elements of (Machine) Learning.

- **Dataset:**
  - Labelled vs. Unlabeled Dataset.
- (Machine) Learning:
  - **A Decision Process (Representation/Model):**
    - Machine learning algorithms(Models) are used to make inference or estimate of an output based on input data – labeled or unlabeled.
  - **An Error Function (Evaluation):**
    - A performance metric used to evaluate the estimate of a model.
    - Metrics depends on types of learning (supervised or unsupervised) and types of task (Classification or Regression)
  - **An model Optimization Process:**
    - An automated algorithm or process used to update parameters of machine learning models until threshold or accepted evaluation metric has been achieved



!!! Learning a Model → means finding the parameter of feature and target mapping function.

2.6 What after learning $f(W, b)$?

- **Prediction:**
  - **Learned model(hypothesis) h(.)** is used to predict the label Y for data without label, the predicted label is represented as $\hat{Y}$.
- **Inference:**
  - Understanding the association between **Y and X**.
    - **Which predictors are associated with the response?**
    - **What is the relationship between the response and predictor?**
    - **Can the relationship between Y and each predictor be adequately summarized using a linear equation?**

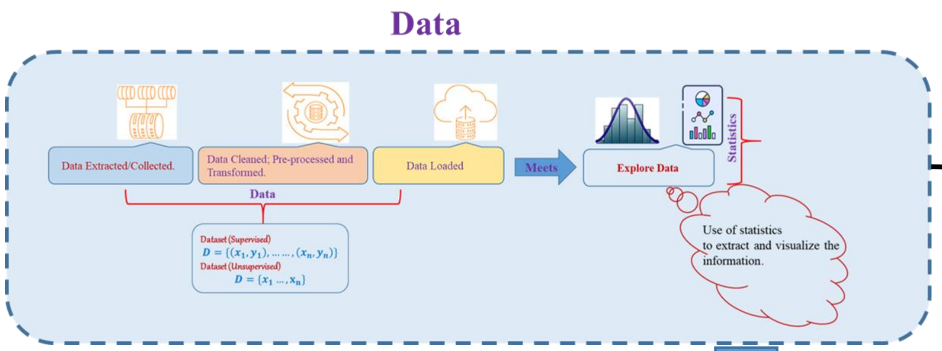
data + model $\xrightarrow{\text{compute}}$ inference

2.7 (Supervised) Machine Learning: Conclusion.

- It is an attempt to find the function *"f"* that minimizes the selected loss such that:
  - $f = argmin_{f \in H} \mathbb{L}(f)$
- A big part of machine learning focuses on the question, how to do this minimization efficiently?
  - **Optimization Techniques.**
- If you find a function *f(.)* with low loss on your data ***D***, how do you know whether it will still get examples right that are not in ***D***?
  - **Generalization!!!**

data + model $\xrightarrow{\text{compute}}$ inference

Workflow: Supervised Learning.



Given training data $\{(x_i, y_i): 1 \le i \le n\}$ i.i.d from the distribution D.

Find $y = f(x) \in H$ that minimizes $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$

such that the expected loss is small:

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, xy)]$$

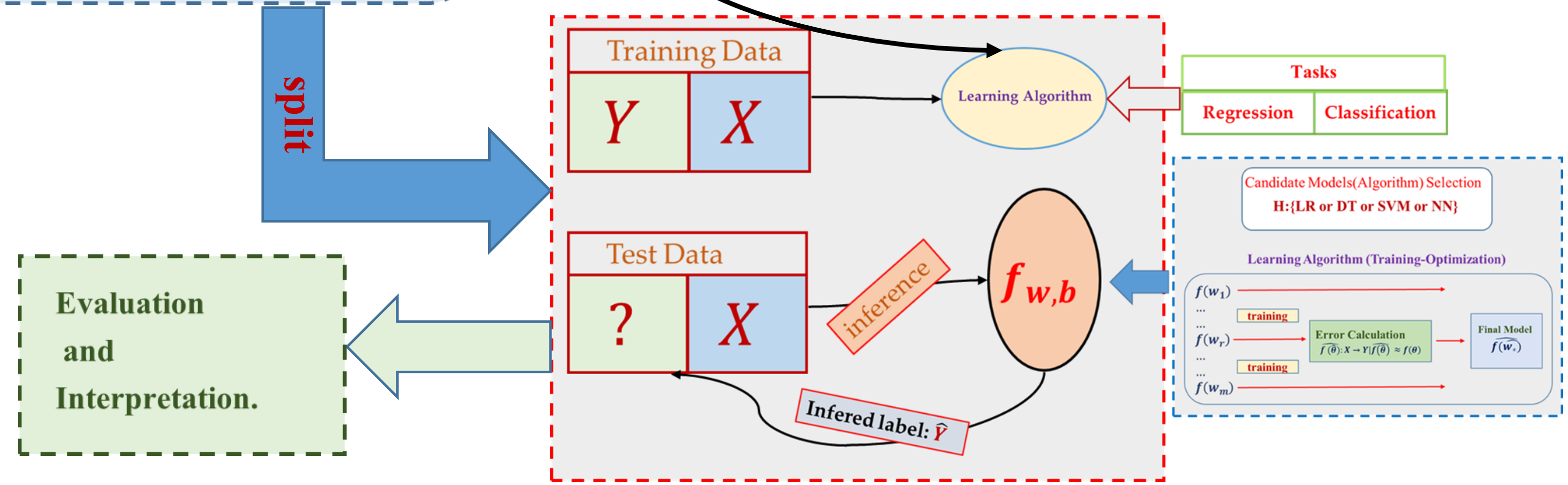


Fig: The General supervised approach to machine learning: a learning algorithm reads in training data and learns a parameters of chosen function. This function(parameters) are then used to perform an inference.

Classification with Logistic Regression.

3. Model ~ The Decision Process.

3.1 Classification: Introduction.

- **Objective of Classification Problem:**
  - We consider a **pattern classification** problem which is **formulated** in the following way.
    - There is a large, perhaps infinite, **set of objects** (observations, patterns, dataset etc.) which **can be classified** into **two classes** (**that is, assigned to two sets**).
    - We do not have an algorithm that does this classification, but we have a sample of objects with known class labels.
      - Using these classification examples, **we want to define an algorithm/model** that will **classify objects from the entire set with the minimum error**.
- **Training Set:**
  - A sample of objects with known class labels is called training set and is written as $(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \dots, (\boldsymbol{x_n}, \boldsymbol{y_n})$
    - where $\boldsymbol{y_i} \in \{-1, 1\}$ is the class label of vector $\boldsymbol{x_i}$ and $\boldsymbol{n}$ is the size of the training set.
- **A Decision Function:**
  - A classifier is represented by a decision function:

$$\boldsymbol{f(x)}: \boldsymbol{V} \to \{-\mathbf{1}, \mathbf{1}\}$$

such that $\boldsymbol{f(x)} = \mathbf{1}$ if the **classifier assigns** $\boldsymbol{x}$ to the **first class**,
and $\boldsymbol{f(x)} = -\mathbf{1}$, if the **classifier assigns** $\boldsymbol{x}$ to the **second class**.

3.3 Logistic Regression for Binary Classification.

- **Logistic Regression – The Model (Decision Process):**
  - Logistic Regression is a probabilistic, linear classifier parameterized by a weight matrix W and a bias vector b. Mathematically:
    - $\boldsymbol{P(y = 1|x, W, b) = \sigma(w.x + b) = \frac{1}{1+e^{-(w.x+b)}}}$
    - $\boldsymbol{P(y = 0|x, W, b) = 1 - \sigma(w.x + b) = 1 - \frac{1}{1+e^{-(w.x+b)}}}$
- **Decision Function:**
  - The sigmoid function from the prior section thus gives us a way to take an instance x and compute the probability $P(y = 1|x, W, b)$.
  - How do we make a decision about which class to apply to a instance example x?

3.3 Logistic Regression for Binary Classification.

- **Logistic Regression – The Model {Decision Process}:**
  - Logistic Regression is a probabilistic, linear classifier parameterized by a weight matrix W and a bias vector b. Mathematically:
    - $\boldsymbol{P(y = 1|x, W, b) = \sigma(w.x + b) = \frac{1}{1+e^{-(w.x+b)}}}$
    - $\boldsymbol{P(y = 0|x, W, b) = 1 - \sigma(w.x + b) = 1 - \frac{1}{1+e^{-(w.x+b)}}}$
- **Decision Function:**
  - The sigmoid function from the prior section thus gives us a way to take an instance x and compute the probability $P(y = 1|x, W, b)$.
  - How do we make a decision about which class to apply to a instance example x?
  - We design a decision boundary i.e.:
    - **decision(x) =** $\boldsymbol{\begin{cases} 1 \; if \; P(y = 1|x, W, b) > 0.5 \\ 0 \; Otherwise \end{cases}}$

Sigmoid Function.

- **Logistic/Sigmoid function:**
  - The logistic function $\boldsymbol{\sigma}$ is a function from the real line to the **unit interval (0,1)**
    - $\boldsymbol{\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \quad -\infty < x < \infty}$
  - The function maps any real value $\{x \in (-\infty, +\infty)\}$ into another value between 0 and 1.

- **Properties:**
  - Range: $\boldsymbol{0 < \sigma(x) < 1}$.
  - **Inverse: x= $\boldsymbol{\sigma^{-1}(p) = \ln\left(\frac{p}{1-p}\right)}$: logit function.**
  - Derivative: $\boldsymbol{\frac{d}{dt}\sigma(x) = \sigma(x)\left(1 - \sigma(x)\right) = \sigma(x)\sigma(-x)}$
  - In machine learning, we use sigmoid to map **predictions to probabilities.**

Data/Feature:

$\boldsymbol{X} \in \mathbb{R}$ **and** $\boldsymbol{Y} \in [\mathbf{0}, \mathbf{1}]$

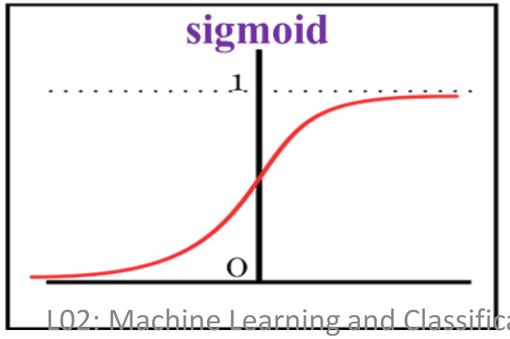
Here: $\boldsymbol{X}$**: is a feature matrix i.e.**

$$\boldsymbol{X} := \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$$

And: $\boldsymbol{Y}$**: is a binary label space.**

Input:

$\boldsymbol{X} = [x_{11} \quad \cdots \quad x_{1n}] \in \mathbb{R}$



Output:

$\mathbf{P}(\boldsymbol{Y} = \mathbf{1} | \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b})$: $\{\boldsymbol{Y} \in [\mathbf{0}, \mathbf{1}]\}$

Decision Function

decision:= $\begin{cases} \mathbf{1}\ \boldsymbol{if}\ \boldsymbol{P} \geq \mathbf{0.5} \\ \mathbf{0}\ \boldsymbol{otherwise} \end{cases}$

3.3 Logistic Regression for Multi-class Classification.

- **aka ~ Multinomial Logistic Regression ~ The Model {Decision Process}:**
  - Mathematically, the probability that an input vector $\boldsymbol{x}$ is a member of a class $\boldsymbol{i}$, a value of a stochastic variable Y, can be written as:
  - $\boldsymbol{P(Y = i|x, W, b) = softmax\,(Wx + b) = \frac{e^{w_i x_i + b_i}}{\sum_j e^{w_j x + b_j}}}$
- **Decision Function:**
  - The model's prediction $\boldsymbol{y_{pred}}$ is the class whose **probability is maximal** i.e.:
  - $\boldsymbol{y_{pred} = argmax_i P(Y = i|x, W, b)}$

3.3 Logistic Regression for Multi-class Classification.

- **aka ~ Multinomial Logistic Regression ~ The Model {Decision Process}:**
  - Mathematically, the probability that an input vector $\boldsymbol{x}$ is a member of a class $\boldsymbol{i}$, a value of a stochastic variable Y, can be written as:
  - $\boldsymbol{P(Y = i|x, W, b) = softmax\,(Wx + b) = \frac{e^{w_i x_i + b_i}}{\sum_j e^{w_j x + b_j}}}$
- **Decision Function:**
  - The model's prediction $\boldsymbol{y_{pred}}$ is the class whose **probability is maximal** i.e.:
  - $\boldsymbol{y_{pred} = argmax_i P(Y = i|x, W, b)}$

Confused!!! – Let's look into example but first redefine sigmoid and softmax

Softmax Function

- **Let's ask "chatgpt" what is softmax function**:
  - Softmax is a **mathematical function** that is often used in machine learning and **deep learning** for various purposes, but most commonly for **multiclass classification problems**.
    - It is used to **transform a vector of raw scores or logits** (real numbers) into a **probability distribution** over **multiple classes**.
  - The **softmax function** takes an **input vector (commonly denoted as "z")** of length "**N**" and **computes a new vector of the same length**, where **each element in the new vector represents the probability** of the **corresponding class.**
  - Represented by:
    - $softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$
    - Here:
      - **e: Euler's number.**
      - **Zi : is the raw score or "logit" for class "i".**
      - **Denominator: sum of the exponentials of all the raw scores, ensuring output probabilities to sum 1.**

- **"chatgpt"**

Softmax: Example

Data/Feature:

$X \in \mathbb{R}$ and $Y \in [0, 1, 2]$

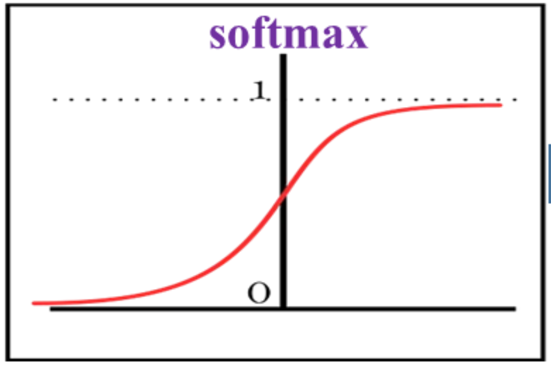
Here: X: is a feature matrix i.e.

$$X := \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$$

And: Y: multiclass label space {3 class}.

Input:

$$X = [x_{11} \quad \cdots \quad x_{1n}] \in \mathbb{R}$$



Output:

$$P(Y = 0 | X, W, b) = p_0$$
$$P(Y = 1 | X, W, b) = p_1$$
$$P(Y = 2 | X, W, b) = p_2$$

Decision Function

$$\text{decision} := \text{argmax} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix}$$

Fig: Workflow of Multinomial Logistic Regression with Softmax (Softmax Regression)

Classification with Logistic Regression.

4. The Error/Loss Function.

4.1The Loss Function: Introduction.

- Any function that express how close the model/classifier's output $\{\hat{y}\}$ is to the correct output $\{y\}$. These function are called as loss function i.e.
  - $L(\hat{y}, y)$ = **How much $\hat{y}$ differs from y.**
- The loss function we use for classification task are in general called as cross entropy loss and in general written as:

The formula for **cross entropy loss** is:

$$L_{CE}(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Where:

- y is the **true label** from provided set of data.
- $\hat{y}$ is the **predicted label** by the classifier.
- C is the **number of classes** in dataset.

- For Binary classification where C = 2, the cross entropy loss is defined as:

$$\boxed{L_{BCE}(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]}$$

4.2 Loss Vs. Cost Function.

- **Loss function** are calculated for **each pair** of input and target variable whereas **Cost function** is an **average** of loss function. Example:

Input$\{X\}$	**Target**$\{Y\}$	**Predicted**$\{\hat{Y}\}$	**Loss**$\{Y - \hat{Y}\}$
x^1	y_1	$\widehat{y_1}$	$(y_1 - \widehat{y_1})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x^n	y_n	$\widehat{y_n}$	$(y_n - \hat{y}_n)$
		Cost:=	$\frac{\sum_i^n (y_i - \hat{y})}{n}$

pairwise

average

- The Objective of any Machine/Deep Learning algorithms are to find best mapping function :
  - $f(W, b): X \rightarrow Y$
  - One way to achieve such is to find the parameters i.e. **weight and biases**$\{f(W, b)\}$ which minimizes our loss/cost function.

Classification with Logistic Regression.

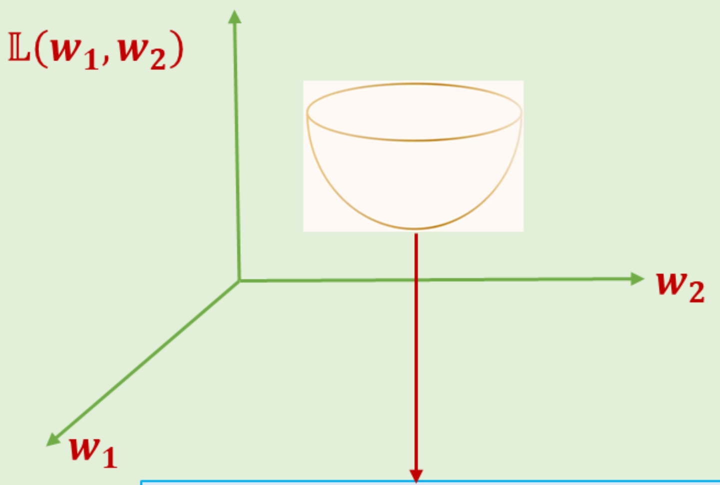
5. The optimization ~ Learning/Training the Model.

5.1 Learning/Training The Model

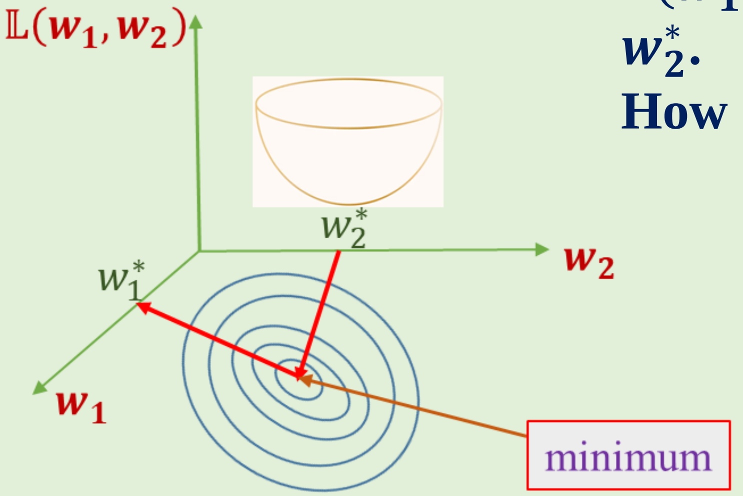
- The Objective of any Machine/Deep Learning algorithms are to find best mapping function :
  - $\boldsymbol{f(W, b): X \rightarrow Y}$
  - One way to achieve such is to find the parameters i.e. **weight and biases**$\{\boldsymbol{f(W, b)}\}$ which minimizes our loss/cost function.
- How do we minimize a (loss) function?
- To answer above question first let's find out what kind of function we are dealing by plotting the loss value Vs. Weights

5.2 Error/loss Surface

At which value of w_1 and w_2 :$\mathbb{L}(w_1, w_2)$ is minimum?



Now minimizing the error becomes the problem of finding minimum of the error surface governed by error function.



$\mathbb{L}(w_1, w_2)$ is minimum at w_1^* and w_2^*.
How do we find such value?

Options-1: Brute Force:

A way to estimate $argmin_{w_1, w_2} \mathbb{L}$ is to :
Calculate the loss function for every possible w_2 and w_1.
Then select w_1 and w_2 where the loss function is minimum.
Is it a optimal Method?

5.3 What is Gradient?

- The **gradient** is a fancy word for derivative, or the rate of change of a function. It's a vector (a direction to move) that
  - Points in the direction of greatest increase of a function.
  - Is zero at a local maximum or local minimum (because there is no single direction of increase).
- The term "gradient" is typically used for functions with several inputs{X} and a single output{Y}.
- **Derivative:**
  - The regular, plain-old derivative gives **us the rate of change of a single variable**, usually x.
    - For example, $\frac{dF}{dx}$ tells us how much the function **F** changes for a change in **x**.
  - But if a function takes multiple variables, such as x and y and z, it will have multiple derivatives:
    - We can represent these multiple rates of change in a vector, with one component for each derivative. Thus, a function that takes 3 variables will have a gradient with 3 components:
      - $F(x, y, z)$ has three variables and three derivatives: $\frac{dF}{dx}, : \frac{dF}{dy}: , : \frac{dF}{dz}$ {Partial Derivative}
- The gradient of a multi-variable function has a component for each direction.

5.4 Gradient Descent Algorithm!!!

- **Idea:**
  - It is an iterative methods used to compute minimum.
  - The gradient ∇L at any point is the direction of the steepest increase. The negative gradient is the direction of steepest decrease.
  - By following the –ve gradient, we can eventually find the lowest point.
  - This method is called Gradient Descent.
- **Algorithm:**
  - For some cost/loss functions: $\mathbb{L}(w_0, \dots, w_d)$.
  - Start off with some guesses for $w_0, \dots, w_d$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until Convergence:{

$$w_{new} := w_{old} - \underbrace{\alpha}_{\text{Learing Rate}} \frac{\partial\, \mathbb{L}(w_0, \dots, w_d)}{\partial w}$$

}

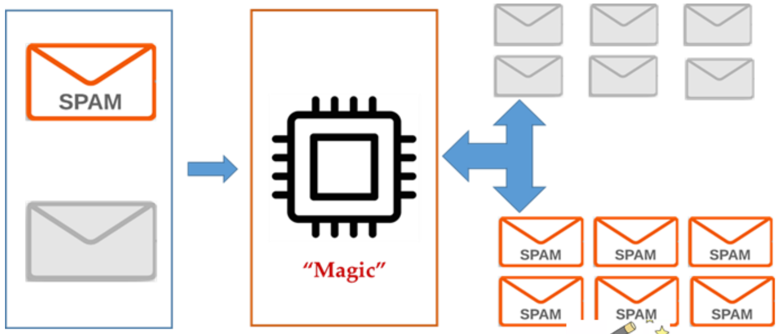
{Convergence:=Until pre-defined number of iterations.}

How Good is Your Model/Classifier?

6. The Evaluation Metrics ~ For Binary Classification.

{Idea can be extended to Multi-class}

6.1 Example: Remember!!!



We replaced the magic with Logistic Regression and Got the following Result

Let's Demystify the Magic!!

	Email	Actual	Predicted
1	I need help please wire me $1000 right now	1 - Spam	1 - Spam
2	Hot new investment, don't miss this!	1 - Spam	0 - Not Spam
3	Please help me?	1 - Spam	0 - Not Spam
4	Your parcel will be delivered today	0 - Not Spam	0 - Not Spam
5	Review changes to our Terms and Conditions	0 - Not Spam	0 - Not Spam
6	Weekly sync notes	0 - Not Spam	0 - Not Spam
7	Re: Follow up from today's meeting	0 - Not Spam	0 - Not Spam
8	Pre-read for tomorrow	0 - Not Spam	0 - Not Spam
9	Brief results from new UX study	0 - Not Spam	0 - Not Spam
10	Meeting notes from today	0 - Not Spam	0 - Not Spam
11	A reminder about your next appointment	0 - Not Spam	1 - Spam
12	An invitation to a conference call tomorrow	0 - Not Spam	0 - Not Spam
13	We have some news about your application	0 - Not Spam	1 - Spam
14	Final briefing notes for your meeting	0 - Not Spam	0 - Not Spam
15	Attached are the updated guidelines for this project	0 - Not Spam	0 - Not Spam
16	New feedback on your project from last week's meeting	0 - Not Spam	0 - Not Spam
17	Thank you for participating in our survey	0 - Not Spam	0 - Not Spam
18	Your account has been upgraded to Premium status	0 - Not Spam	1 - Spam
19	Confirming the dates for the annual board meeting	0 - Not Spam	0 - Not Spam
20	Please review these updated guidelines	0 - Not Spam	0 - Not Spam

How good of a job we did?

How good is my Model?

6.2 Accuracy!!

- The most straightforward way to measure a classifier's performance is using the Accuracy metric.
- Here, we compare the actual and predicted class of each data point, i.e. for total predictions how many were correctly predicted.
- Accuracy is given as:
  - accuracy = $\frac{\text{Number of correct predictions}}{\text{Total number of predictions.}}$
- For our Example:

	Email	Actual	Predicted	
1	I need help please wire me $1000 right now	1 - Spam	1 - Spam	✓
2	Hot new investment, don't miss this!	1 - Spam	0 - Not Spam	⊗
3	Please help me?	1 - Spam	0 - Not Spam	⊗
4	Your parcel will be delivered today	0 - Not Spam	0 - Not Spam	✓
5	Review changes to our Terms and Conditions	0 - Not Spam	0 - Not Spam	✓
6	Weekly sync notes	0 - Not Spam	0 - Not Spam	✓
7	Re: Follow up from today's meeting	0 - Not Spam	0 - Not Spam	✓
8	Pre-read for tomorrow	0 - Not Spam	0 - Not Spam	✓
9	Brief results from new UX study	0 - Not Spam	0 - Not Spam	✓
10	Meeting notes from today	0 - Not Spam	0 - Not Spam	✓
11	A reminder about your next appointment	0 - Not Spam	1 - Spam	⊗
12	An invitation to a conference call tomorrow	0 - Not Spam	0 - Not Spam	✓
13	We have some news about your application	0 - Not Spam	1 - Spam	⊗
14	Final briefing notes for your meeting	0 - Not Spam	0 - Not Spam	✓
15	Attached are the updated guidelines for this project	0 - Not Spam	0 - Not Spam	✓
16	New feedback on your project from last week's meeting	0 - Not Spam	0 - Not Spam	✓
17	Thank you for participating in our survey	0 - Not Spam	0 - Not Spam	✓
18	Your account has been upgraded to Premium status	0 - Not Spam	1 - Spam	⊗
19	Confirming the dates for the annual board meeting	0 - Not Spam	0 - Not Spam	✓
20	Please review these updated guidelines	0 - Not Spam	0 - Not Spam	✓

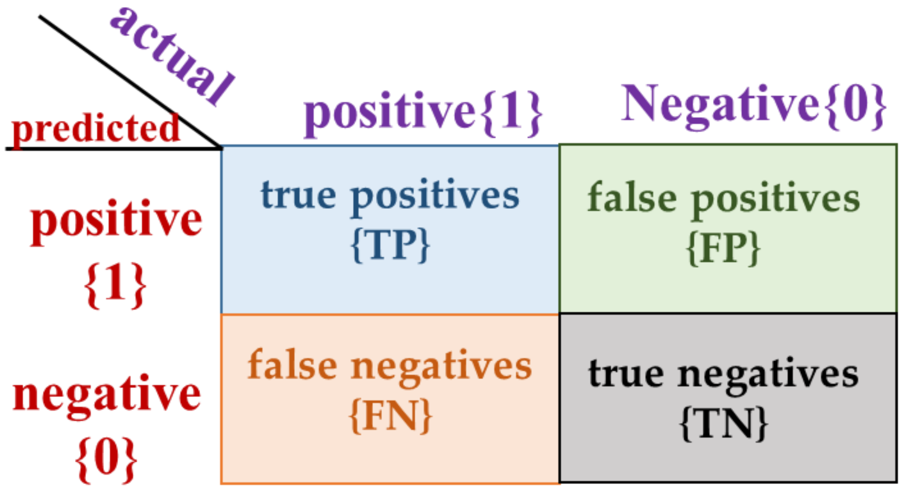
$$\text{accuracy} = \frac{15}{20} \times 100\% = 75\%$$

6.3 Is accuracy an adequate evaluation metrics?

- Accuracy is often used as the measure of classification performance because it is simple to compute and easy to interpret.
- However, it can turn out to be misleading in some cases. For instances:
    - **class imbalance**: scenario where certain classes contain way more data points than the others.
        - In our example: 17 out of 20 datapoints are of not-spam class, and only 3 are from spam class.
            - What if our model has predicted all datapoints to be not-spam, accuracy would have been:= $\frac{17}{20} \times 100\% = 85\%$.
    - **differential misclassification costs** – getting a positive wrong costs more than getting a negative wrong.
        - For example: we built a model to identify Tumor from the given datasets and overall accuracy more than 90% {out of 10 times 9 times we predicted tumor correctly}.
            - What happens for onetime we predicted person to have Tumor and the person do not have tumor?

6.4 Other Accuracy Metrics: Confusion Matrix

- A confusion matrix, is a technique for summarizing the performance of classification algorithm.
- The Confusion Matrix takes the classification results and groups them into four categories:
- For our email example we assign:
    - spam a 1 label{class of interest}
    - not-spam a 0 label



confusion matrix

6.5 Confusion Matrix: Example.

- Let's populate the confusion matrix with email classification example.
- For our email example we assign:
    - spam a **1** label{class of interest}
    - **not-spam** a **0** label
- Thus:
    - TP:→ actual **spam {1}** predicted **spam {1}** := 1
    - FP:→ actual **spam{1}** predicted **not-spam{0}**:= 2
    - TN:→ actual **not-spam{0}** predicted **not-spam{0}**:= 14
    - FN:→ actual **not-spam{0}** predicted **spam{1}**:= 3

predicted \ actual	positive{1}	Negative{0}
positive {1}	true positives {TP=1}	false positives {FP=2}
negative {0}	false negatives {FN=3}	true negatives {TN=14}

	Email	Actual	Predicted
1	I need help please wire me $1000 right now	1 - Spam	1 - Spam
2	Hot new investment, don't miss this!	1 - Spam	0 - Not Spam
3	Please help me?	1 - Spam	0 - Not Spam
4	Your parcel will be delivered today	0 - Not Spam	0 - Not Spam
5	Review changes to our Terms and Conditions	0 - Not Spam	0 - Not Spam
6	Weekly sync notes	0 - Not Spam	0 - Not Spam
7	Re: Follow up from today's meeting	0 - Not Spam	0 - Not Spam
8	Pre-read for tomorrow	0 - Not Spam	0 - Not Spam
9	Brief results from new UX study	0 - Not Spam	0 - Not Spam
10	Meeting notes from today	0 - Not Spam	0 - Not Spam
11	A reminder about your next appointment	0 - Not Spam	1 - Spam
12	An invitation to a conference call tomorrow	0 - Not Spam	0 - Not Spam
13	We have some news about your application	0 - Not Spam	1 - Spam
14	Final briefing notes for your meeting	0 - Not Spam	0 - Not Spam
15	Attached are the updated guidelines for this project	0 - Not Spam	0 - Not Spam
16	New feedback on your project from last week's meeting	0 - Not Spam	0 - Not Spam
17	Thank you for participating in our survey	0 - Not Spam	0 - Not Spam
18	Your account has been upgraded to Premium status	0 - Not Spam	1 - Spam
19	Confirming the dates for the annual board meeting	0 - Not Spam	0 - Not Spam
20	Please review these updated guidelines	0 - Not Spam	0 - Not Spam

6.6 Evaluation Metrics

- Confusion metrics are then utilize to generate various other metrics including accuracy.
- **Accuracy:**
  - simply a ratio of correctly predicted observation to the total observations.
    - $\boldsymbol{accuracy = \frac{TP+TN}{TP+FP+FN+TN}}$
- **Precision:**
  - Only looks at positive class/label.
  - Precision is the ratio of **correctly predicted positive observations** to the **total predicted positive observations**.
    - {Out of all **predicted positive class** how many **are correct**}
    - $\boldsymbol{precison = \frac{TP}{TP+FP}}$

6.6 Evaluation Metrics

- Confusion metrics are then utilize to generate various other metrics including accuracy.
- Recall {aka sensitivity aka true positive rate}:
  - Recall is the ratio of correctly predicted positive observations to the all observations in actual class – yes
  - Out of all the positive classes, how many instances were identified correctly.
  - i.e. Sensitivity describes how good a model at predicting positive classes.
  - higher the sensitivity value means your model is good in predicting positive classes
  - {Among total positive classes in dataset: How many were correctly classified}

$$\boldsymbol{recall = \frac{TP}{TP + FN}}$$

6.6 Evaluation Metrics

- If you are not sure about which metrics is better for you task in hand, there is always F1-Score:
- F1 score is the weighted average of Precision and Recall.

$$F1 - Score = \frac{Precision * Recall}{Precision + Recall}$$

6.7 Confusion Matrix : Multi-class Problem

- Error in classification problem can be broadly of two kind i.e.
  - True:
    - label $\{y\}$ is 0 predicted $\{\hat{y}\}$ is 0.
    - label $\{y\}$ is 1 predicted $\{\hat{y}\}$ is 1.
  - False:;
    - label $\{y\}$ is 0 predicted $\{\hat{y}\}$ is 1.
    - label $\{y\}$ is 1 predicted $\{\hat{y}\}$ is 0.
- We can extend this idea to build confusion Matrix for multi class problem.

predicted \ actual	0	1	2
0	True {T}	False {F}	False {F}
1	False {F}	True {T}	False {F}
2	False {F}	False {F}	True {T}

Confusion matrix with 3 class

6.8 Extending to: Precision and Recall

- In our example of email classification we only have two class **spam and not a spam**.
- Now let's imagine there are three different kind of email tags namely:
  - **urgent, normal and spam**
- We built a Multinomial Logistic Regression or Softmax Regression we can determine precision and recall as:

	urgent	normal	spam
urgent	8	10	1
normal	5	60	50
spam	3	30	200

$$\textbf{precision}_u = \frac{8}{8+10+1}$$

$$\textbf{precision}_n = \frac{60}{5+60+50}$$

$$\textbf{precision}_s = \frac{200}{3+30+200}$$

$$\textbf{recall}_u = \frac{8}{8+5+3} \qquad \textbf{recall}_n = \frac{60}{10+60+30} \qquad \textbf{recall}_s = \frac{200}{1+50+200}$$

In Tutorial:

- We will build a Logistic Regression for Multiclass Classification Problem with image dataset.
- Come with laptops.

Thank You and Questions.