中国地质大学
CHINA UNIVERSITY OF GEOSCIENCES
武汉 · WUHAN
七十周年校庆
THE 70TH ANNIVERSARY
1952-2022

“国际学生学术能力提升工程”系列讲座

丝路博士论坛

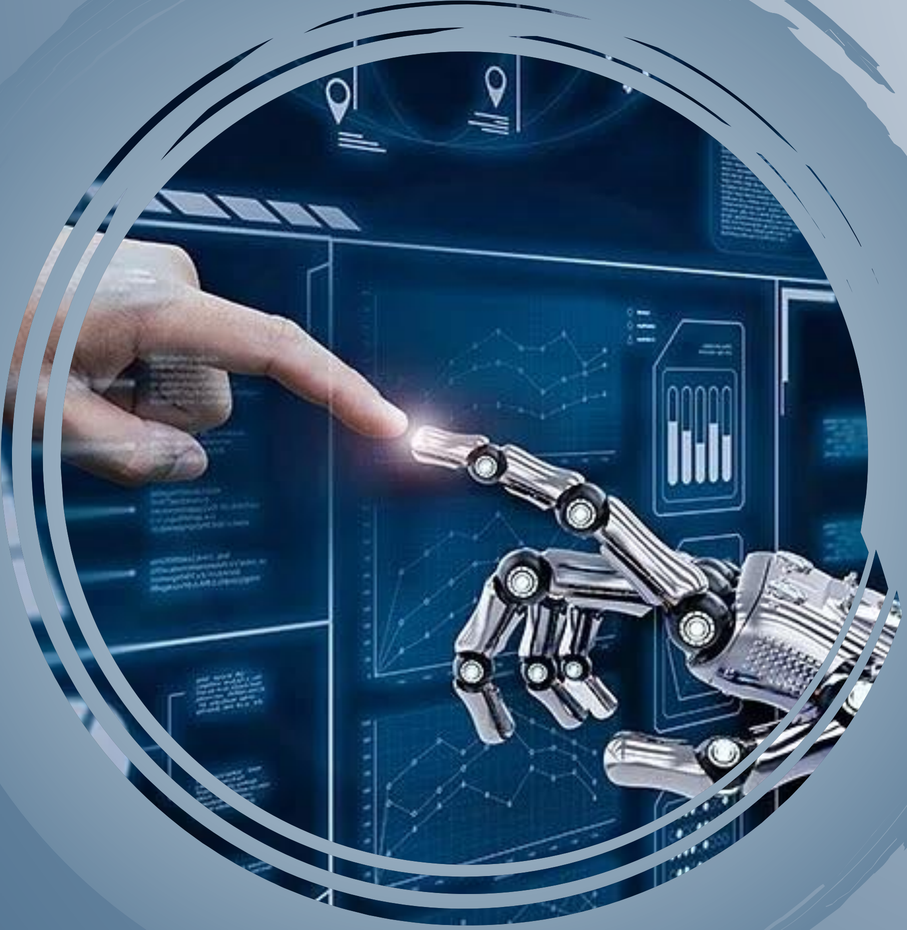
SILK ROAD DOCTOR FORUM

(2022–12)

丝绸之路学院
2022年10月24日

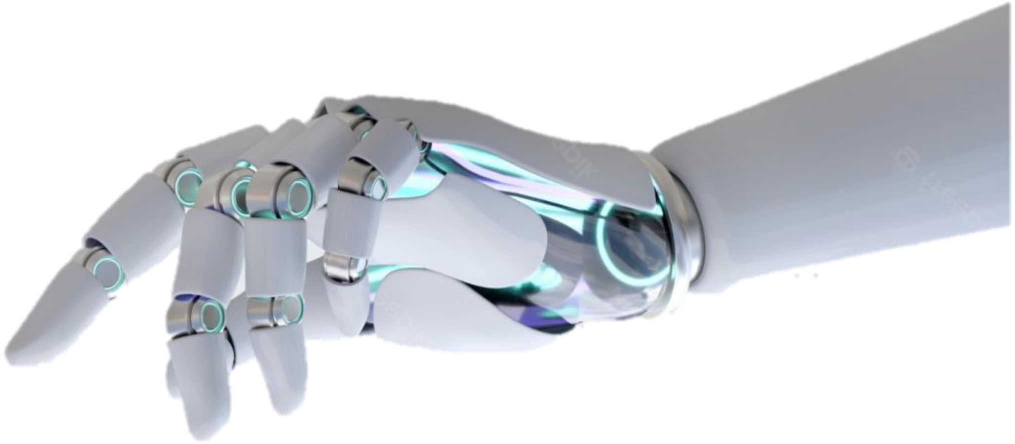
Fundamentals of Deep Learning and the Enhancing Factors

Dalal Mohammed AL-Alimi
Prof. Cai Zhihua

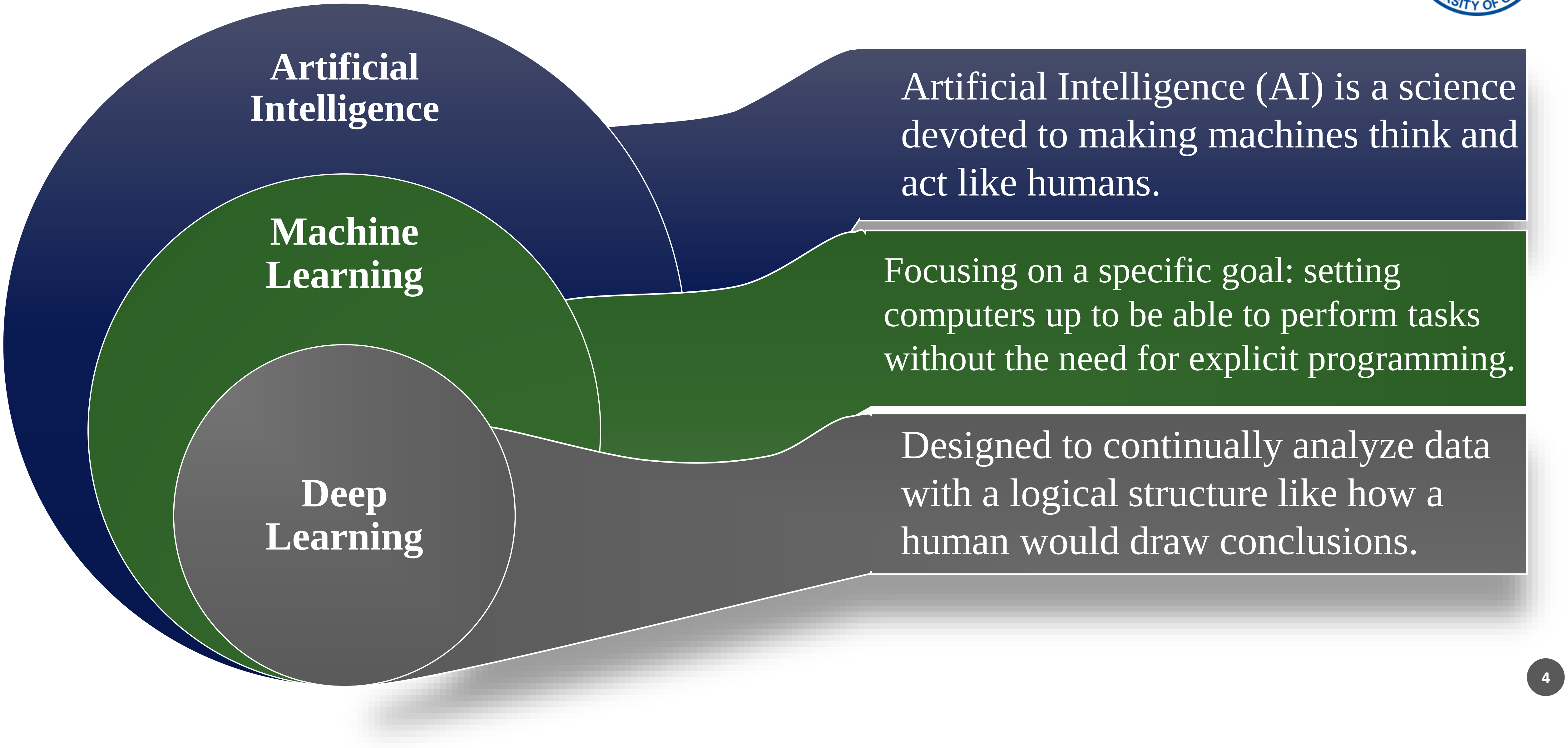


Outlines

- ✓ The definition and differences of Deep Learning and Machine Learning.
- ✓ Understanding the Data.
- ✓ Cleaning Data.
- ✓ Scaling.
- ✓ Deep Learning Concepts
- ✓ Choosing or Creating the Deep Learning Models.



The definition of Artificial Intelligence (AI)



Artificial Intelligence

Machine Learning

Deep Learning

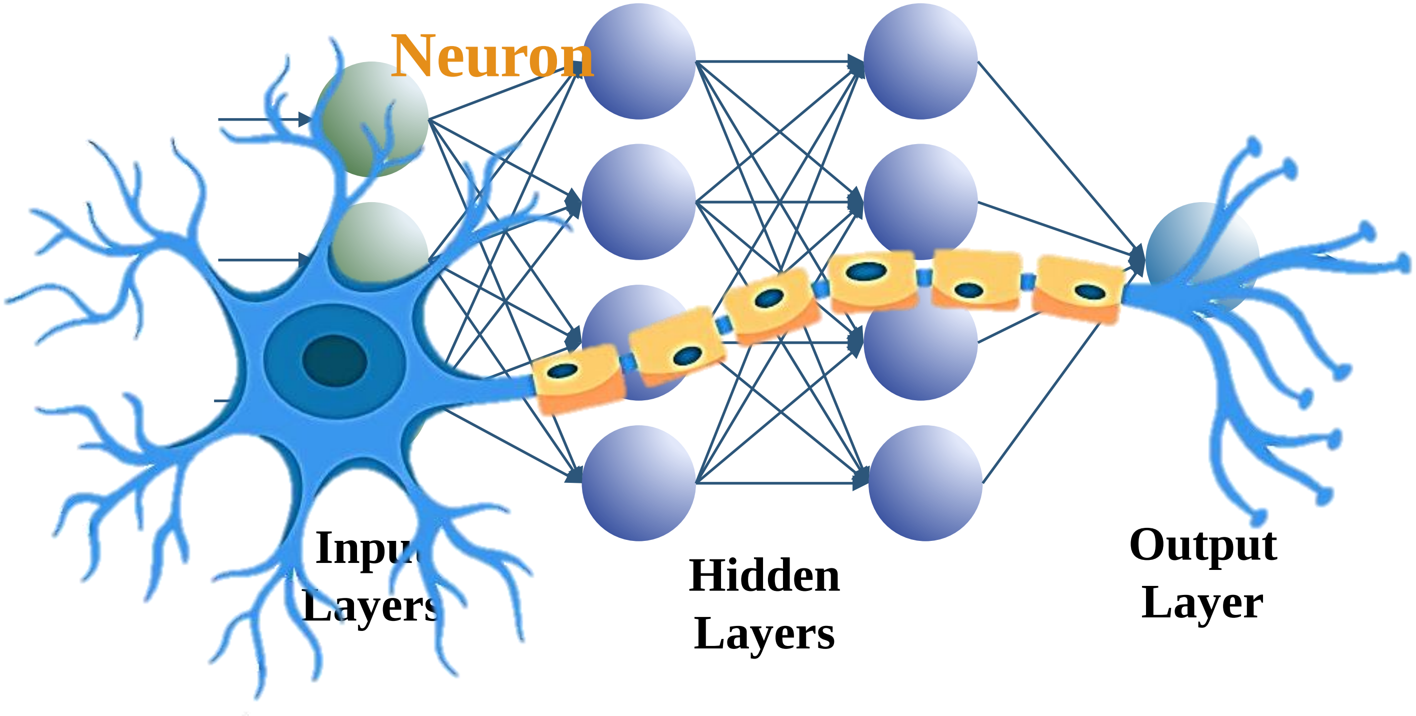
Artificial Intelligence (AI) is a science devoted to making machines think and act like humans.

Focusing on a specific goal: setting computers up to be able to perform tasks without the need for explicit programming.

Designed to continually analyze data with a logical structure like how a human would draw conclusions.

Deep Learning Methods

Deep learning models introduce an extremely sophisticated approach to machine learning and are set to tackle these challenges because they've been specifically modeled after the human brain. Complex, multi-layered “deep neural networks” are built to allow data to be passed between nodes (like neurons) in highly connected ways. The result is a non-linear transformation of the data that is increasingly abstract.



Key Differences Between Machine Learning And Deep Learning

Human Intervention

ML more than DL

Resources

DL more than ML

Approach

DL employs neural networks and ML uses traditional algorithms

Hardware

DL more than ML

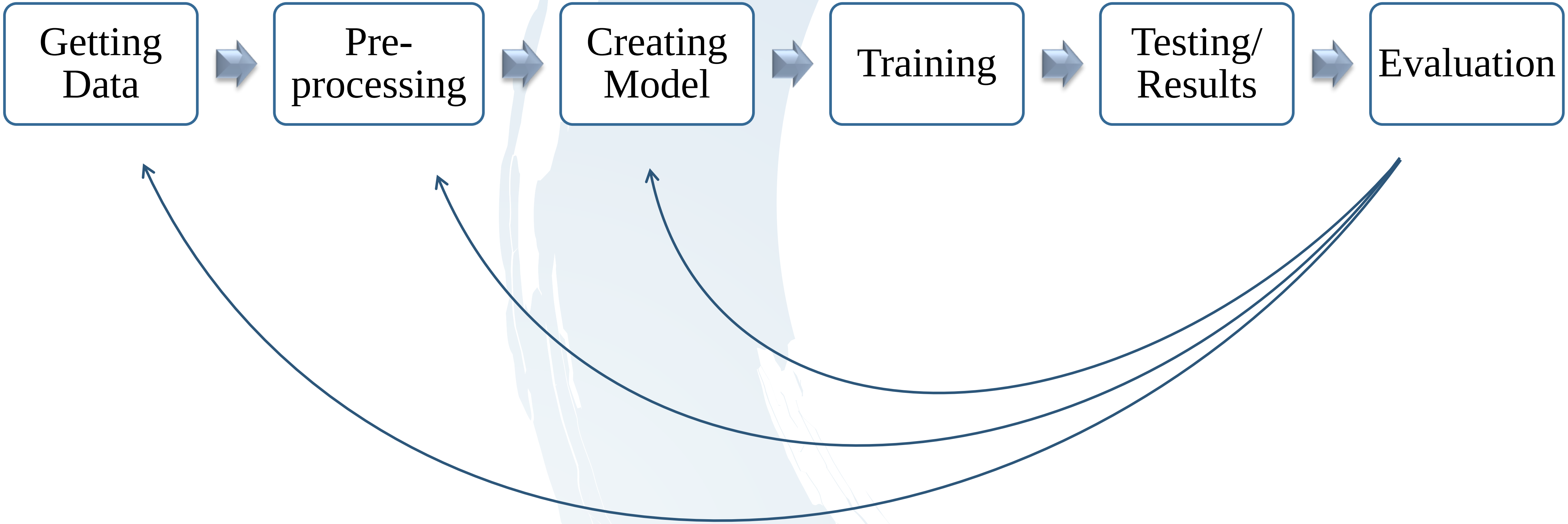
Time

DL more than ML

Applications

DL is more complex and less used than ML

The Work Stages in the AI Field



Getting Data

Pre-processing

Creating Model

Training

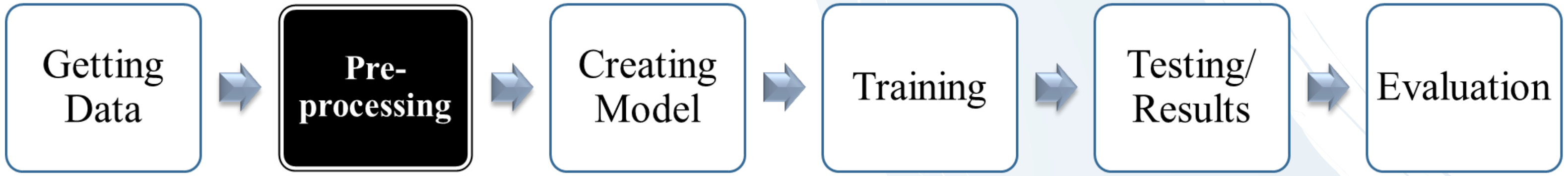
Testing/ Results

Evaluation

Understanding the Data

> *It has been stated that up to **80% of data analysis** is spent on the process of **cleaning and preparing data**. However, being a prerequisite to the rest of the data analysis workflow (visualization, modeling, reporting), it's essential that you become fluent and efficient in data wrangling techniques.*

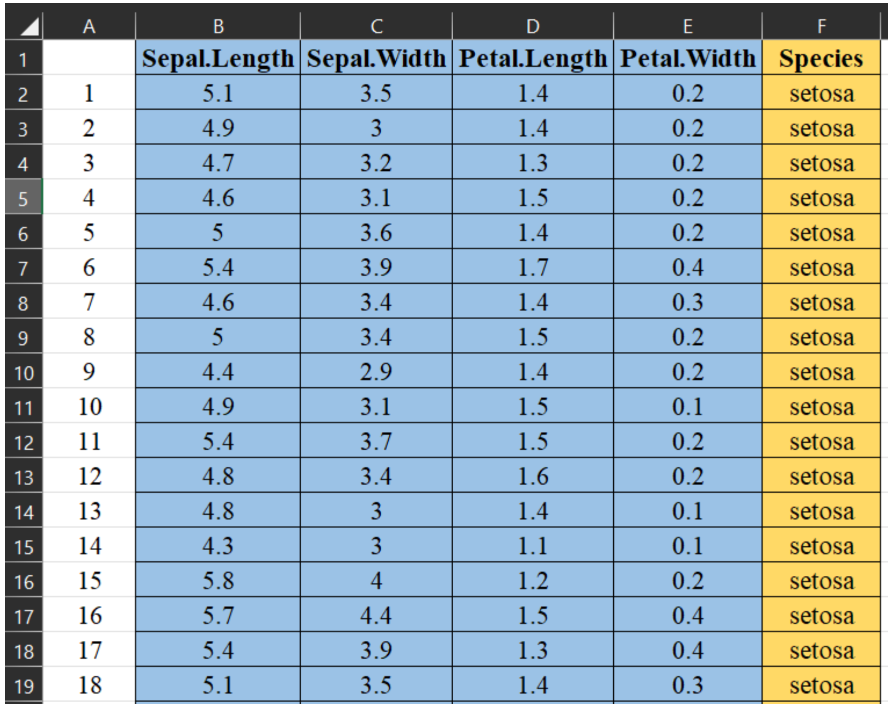
Page v, Data Wrangling with R, 2016.



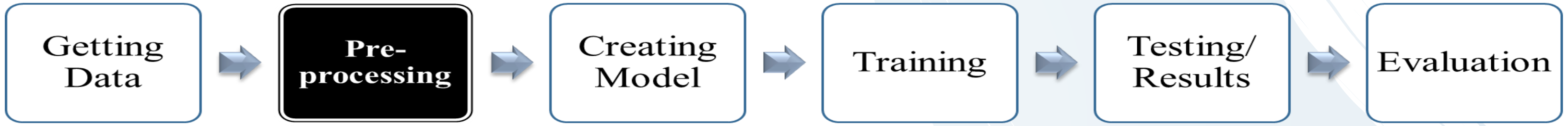
Understanding the Data

- **Row.** A single example from the domain is often called an *instance*, *example*, *case*, or *sample* in machine learning.
- **Column.** A single property recorded for each example is often called a *variable*, *predictor*, *attribute*, or *feature* in machine learning.

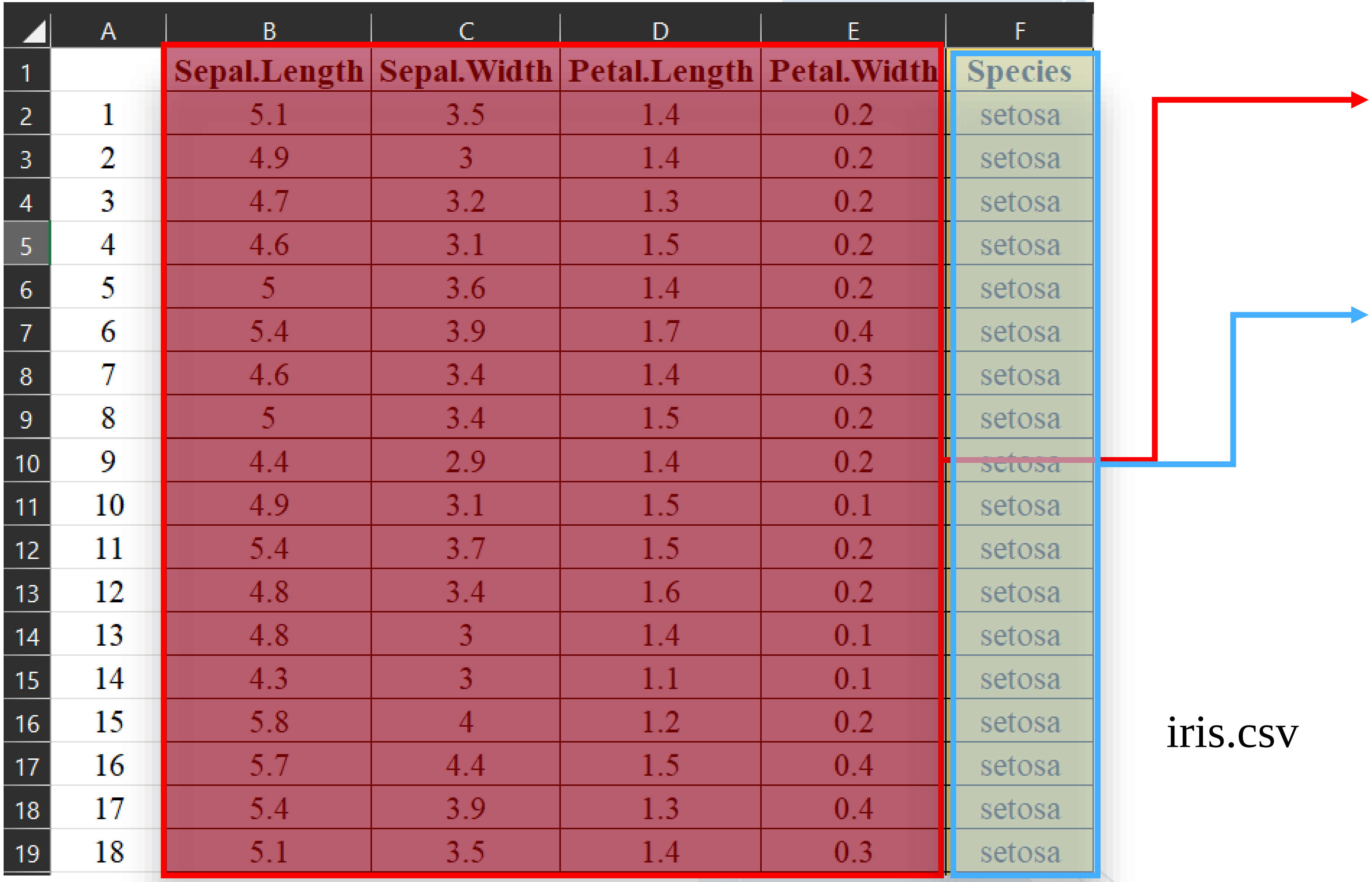
- **Input Variables:** Columns in the dataset provided to a model in order to make a prediction.
- **Output Variable:** Column in the dataset to be predicted by a model.



	A	B	C	D	E	F
1		Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	1	5.1	3.5	1.4	0.2	setosa
3	2	4.9	3	1.4	0.2	setosa
4	3	4.7	3.2	1.3	0.2	setosa
5	4	4.6	3.1	1.5	0.2	setosa
6	5	5	3.6	1.4	0.2	setosa
7	6	5.4	3.9	1.7	0.4	setosa
8	7	4.6	3.4	1.4	0.3	setosa
9	8	5	3.4	1.5	0.2	setosa
10	9	4.4	2.9	1.4	0.2	setosa
11	10	4.9	3.1	1.5	0.1	setosa
12	11	5.4	3.7	1.5	0.2	setosa
13	12	4.8	3.4	1.6	0.2	setosa
14	13	4.8	3	1.4	0.1	setosa
15	14	4.3	3	1.1	0.1	setosa
16	15	5.8	4	1.2	0.2	setosa
17	16	5.7	4.4	1.5	0.4	setosa
18	17	5.4	3.9	1.3	0.4	setosa
19	18	5.1	3.5	1.4	0.3	setosa



Understanding the Data



	A	B	C	D	E	F
1		Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	1	5.1	3.5	1.4	0.2	setosa
3	2	4.9	3	1.4	0.2	setosa
4	3	4.7	3.2	1.3	0.2	setosa
5	4	4.6	3.1	1.5	0.2	setosa
6	5	5	3.6	1.4	0.2	setosa
7	6	5.4	3.9	1.7	0.4	setosa
8	7	4.6	3.4	1.4	0.3	setosa
9	8	5	3.4	1.5	0.2	setosa
10	9	4.4	2.9	1.4	0.2	setosa
11	10	4.9	3.1	1.5	0.1	setosa
12	11	5.4	3.7	1.5	0.2	setosa
13	12	4.8	3.4	1.6	0.2	setosa
14	13	4.8	3	1.4	0.1	setosa
15	14	4.3	3	1.1	0.1	setosa
16	15	5.8	4	1.2	0.2	setosa
17	16	5.7	4.4	1.5	0.4	setosa
18	17	5.4	3.9	1.3	0.4	setosa
19	18	5.1	3.5	1.4	0.3	setosa

Features /x

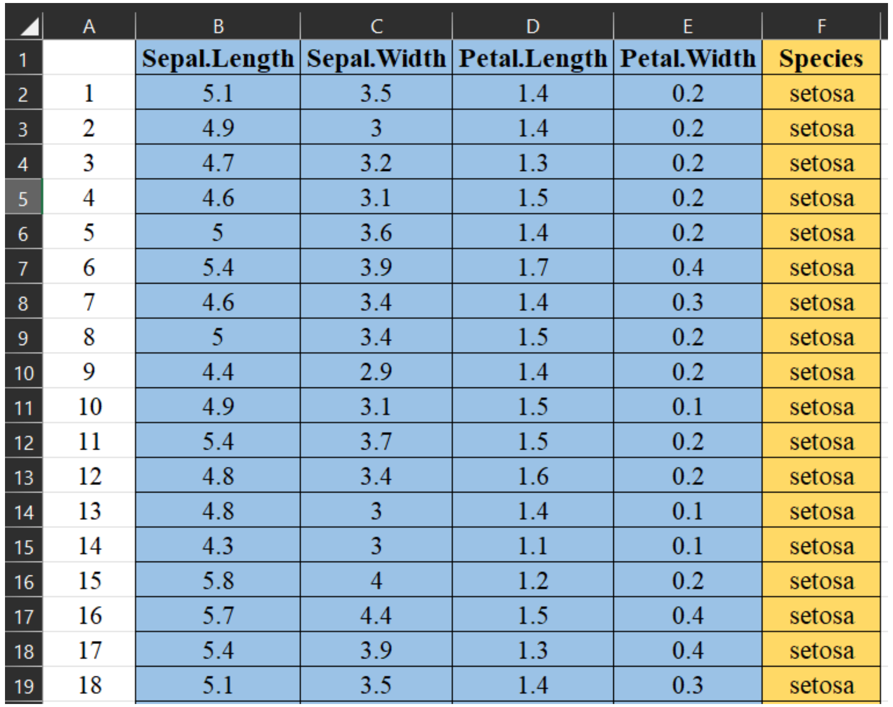
Labels /y

iris.csv

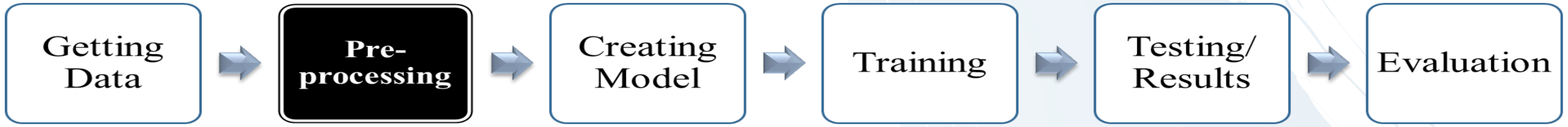
Understanding the Data

Mostly the collected data need:

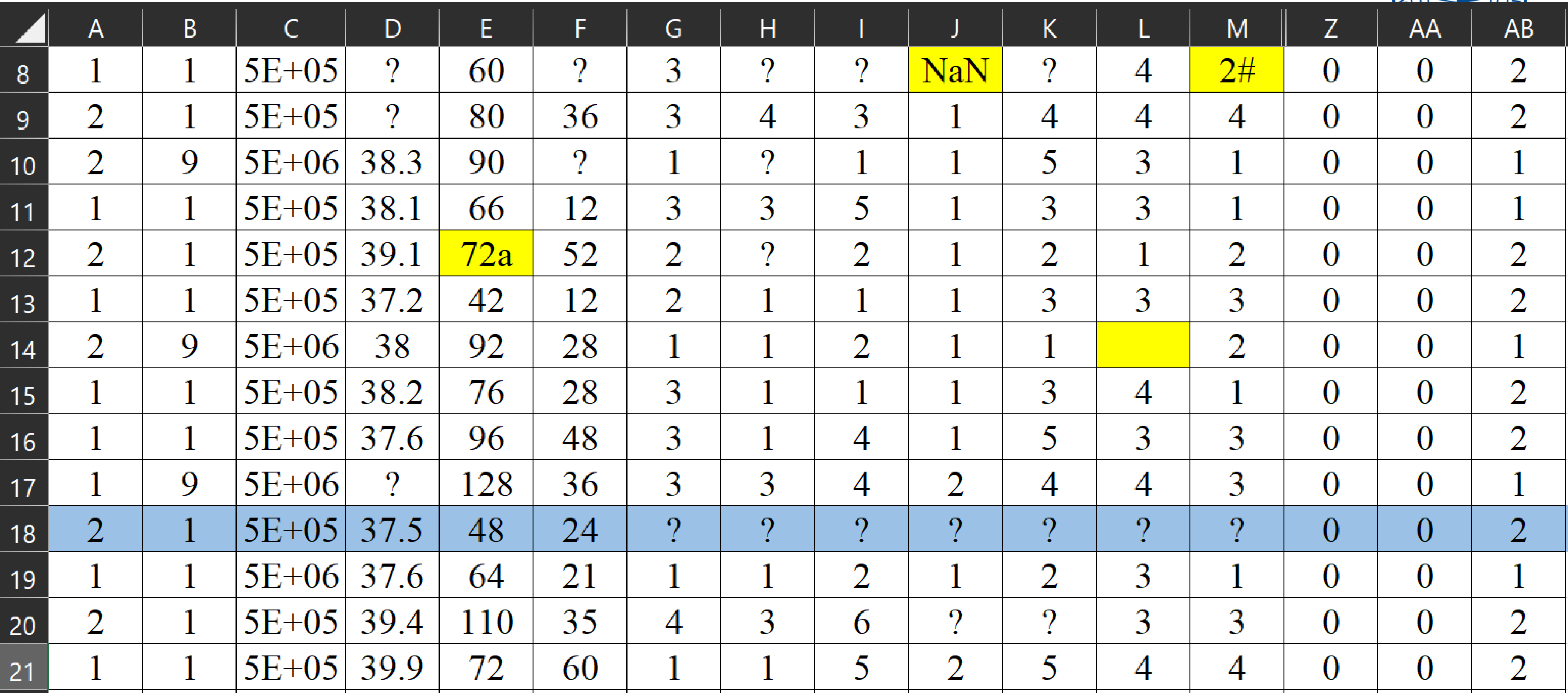
1. Cleaning,
2. Enhance the data range,
3. Dimensionality reduction methods, or
4. Padding



	A	B	C	D	E	F
1		**Sepal.Length**	**Sepal.Width**	**Petal.Length**	**Petal.Width**	**Species**
2	1	5.1	3.5	1.4	0.2	setosa
3	2	4.9	3	1.4	0.2	setosa
4	3	4.7	3.2	1.3	0.2	setosa
5	4	4.6	3.1	1.5	0.2	setosa
6	5	5	3.6	1.4	0.2	setosa
7	6	5.4	3.9	1.7	0.4	setosa
8	7	4.6	3.4	1.4	0.3	setosa
9	8	5	3.4	1.5	0.2	setosa
10	9	4.4	2.9	1.4	0.2	setosa
11	10	4.9	3.1	1.5	0.1	setosa
12	11	5.4	3.7	1.5	0.2	setosa
13	12	4.8	3.4	1.6	0.2	setosa
14	13	4.8	3	1.4	0.1	setosa
15	14	4.3	3	1.1	0.1	setosa
16	15	5.8	4	1.2	0.2	setosa
17	16	5.7	4.4	1.5	0.4	setosa
18	17	5.4	3.9	1.3	0.4	setosa
19	18	5.1	3.5	1.4	0.3	setosa



Getting Data → Pre-processing → Creating Model → Training → Testing/ Results → Evaluation

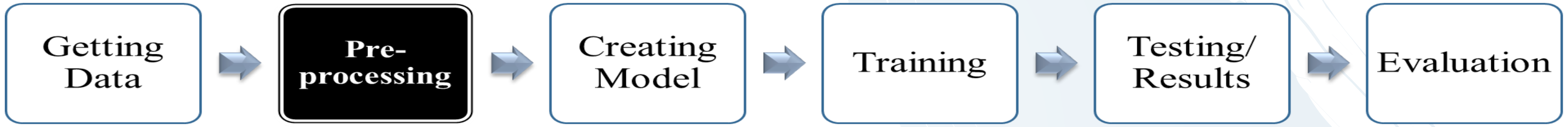


	A	B	C	D	E	F	G	H	I	J	K	L	M	Z	AA	AB
8	1	1	5E+05	?	60	?	3	?	?	NaN	?	4	2#	0	0	2
9	2	1	5E+05	?	80	36	3	4	3	1	4	4	4	0	0	2
10	2	9	5E+06	38.3	90	?	1	?	1	1	5	3	1	0	0	1
11	1	1	5E+05	38.1	66	12	3	3	5	1	3	3	1	0	0	1
12	2	1	5E+05	39.1	72a	52	2	?	2	1	2	1	2	0	0	2
13	1	1	5E+05	37.2	42	12	2	1	1	1	3	3	3	0	0	2
14	2	9	5E+06	38	92	28	1	1	2	1	1		2	0	0	1
15	1	1	5E+05	38.2	76	28	3	1	1	1	3	4	1	0	0	2
16	1	1	5E+05	37.6	96	48	3	1	4	1	5	3	3	0	0	2
17	1	9	5E+06	?	128	36	3	3	4	2	4	4	3	0	0	1
18	2	1	5E+05	37.5	48	24	?	?	?	?	?	?	?	0	0	2
19	1	1	5E+06	37.6	64	21	1	1	2	1	2	3	1	0	0	1
20	2	1	5E+05	39.4	110	35	4	3	6	?	?	3	3	0	0	2
21	1	1	5E+05	39.9	72	60	1	1	5	2	5	4	4	0	0	2

Understanding the Data

Existing Outlier, Skewed, and Noise values may happen because of many reasons, such as:

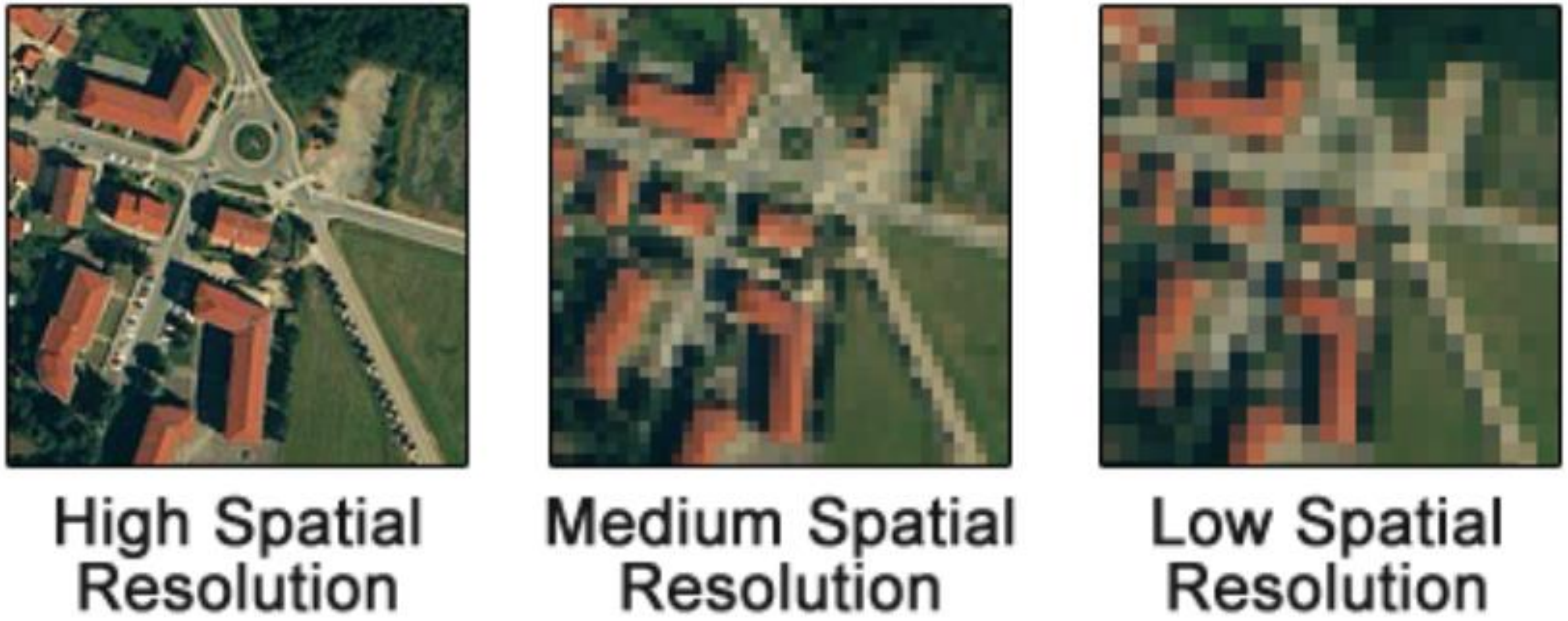
- Measurement or input errors.
- Because of image resolution, or weather situation
- Data corruption.
- Outlier observation

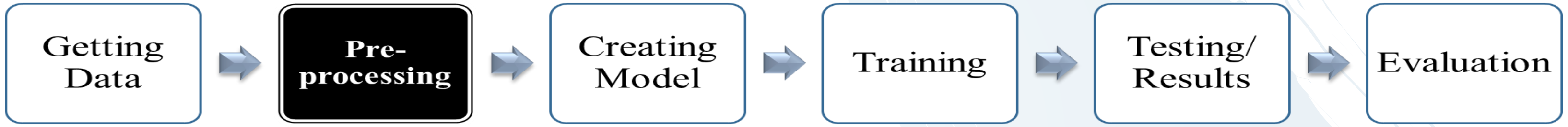


Getting Data → **Pre-processing** → Creating Model → Training → Testing/ Results → Evaluation

Understanding the Data

Spatial resolution describes the quality of an image and how detailed objects are in an image. If the grid cells are smaller, this means the spatial resolution has more detail with more pixels.





Getting Data

Pre-processing

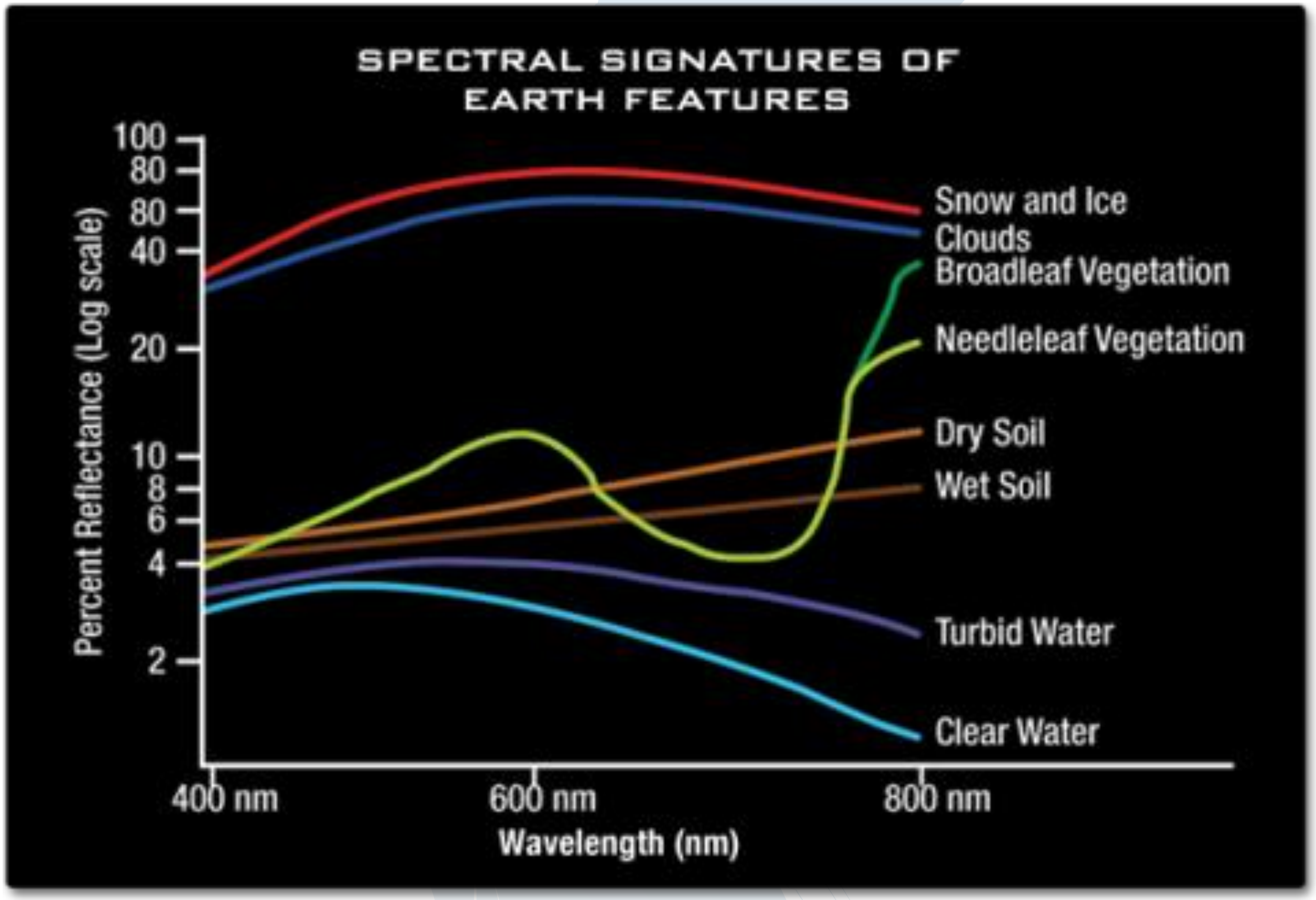
Creating Model

Training

Testing/ Results

Evaluation

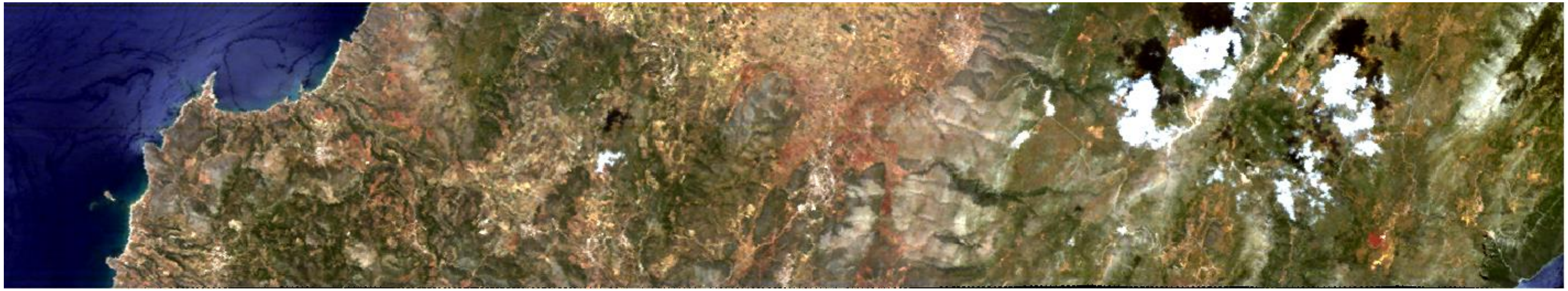
Understanding the Data



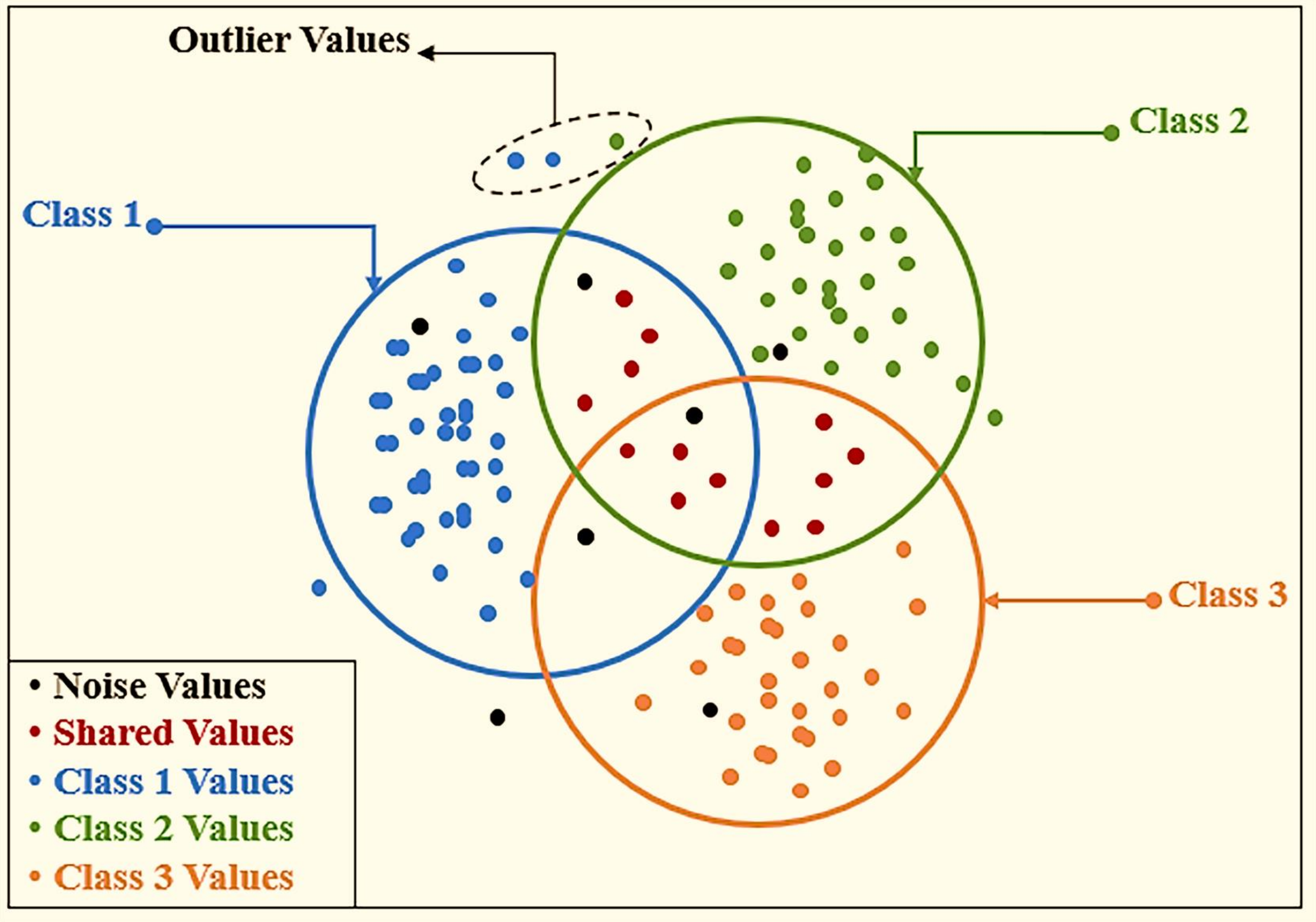
Getting Data ➡ **Pre-processing** ➡ Creating Model ➡ Training ➡ Testing/ Results ➡ Evaluation

Understanding the Data

Dataset	Sensor	Band Numbers	Spatial Dimensions	Spatial Resolution	Classes Number
Dioni	Hyperion sensor (EO-1, USGS)	176	250 × 1376m	30m	12



The Dioni Dataset (HIS)


Outlier Values
Class 1
Class 2
Class 3
• Noise Values
• Shared Values
• Class 1 Values
• Class 2 Values
• Class 3 Values

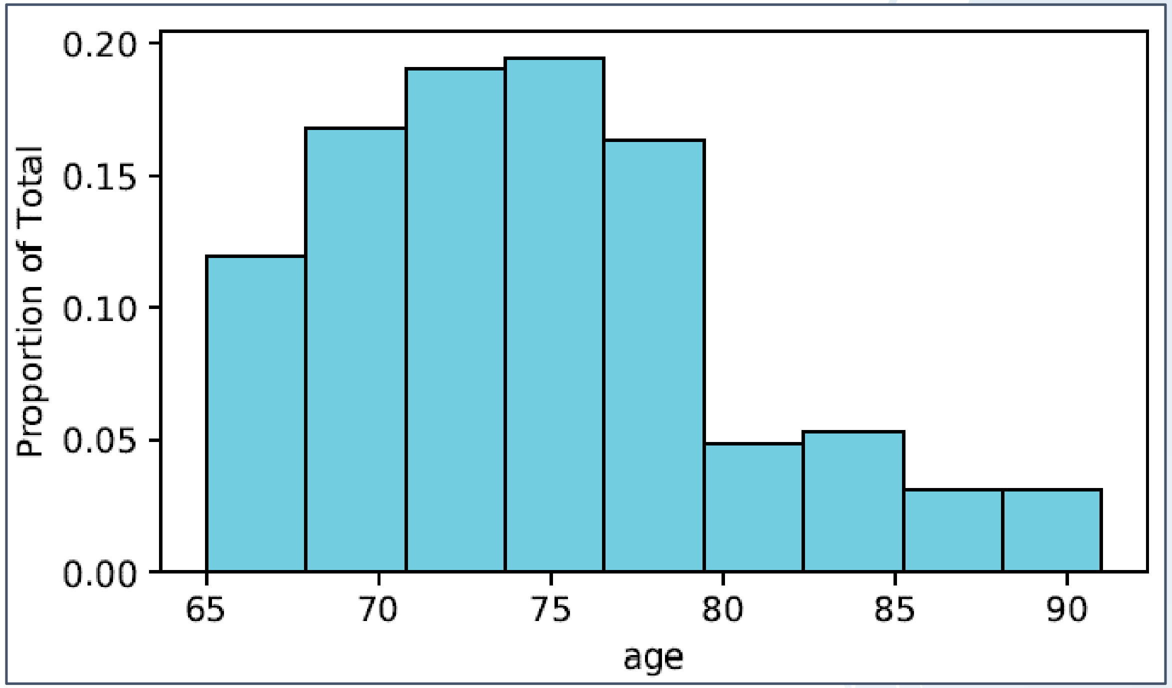
HSI

HOW DO I KNOW THE SHAPE OF MY DATA?!

The content and the distribution

Histogram

What Is a Histogram?



Key Takeaways

- A histogram is a bar graph-like representation of data that buckets a range of classes into columns along the horizontal **x-axis**.
- The vertical **y-axis** represents the number count or percentage of occurrences in the data for each column (frequencies).

When to Use Histogram?

The histogram graph is used under certain conditions. They are:

- The data should be numerical.
- A histogram is used to check the shape of the data distribution.
- Used to check whether the process changes from one period to another.

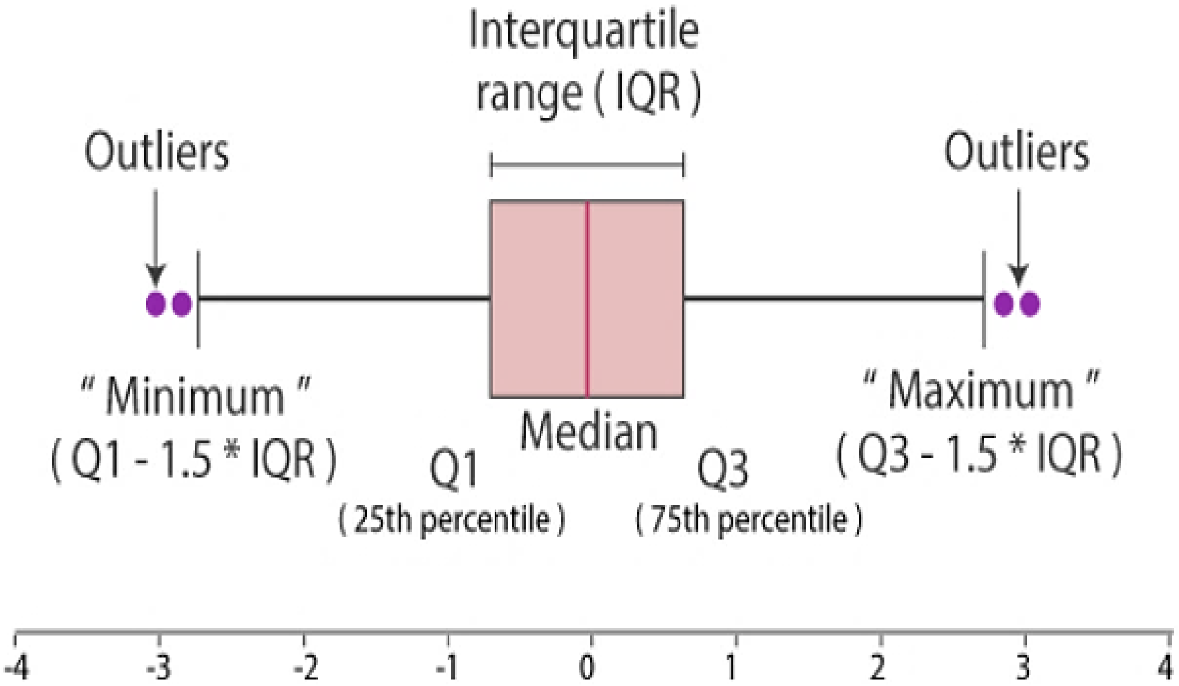
The Box and Whisker Pot

- A boxplot can be viewed as a graphical representation of the five-number summary of the data consisting of the minimum, maximum, and the first, second, and third quartiles.
- The box and whisker plot displays how the data is spread out.
- It does not display the distribution as accurately as a stem and leaf plot or histogram does. But it is principally used to show whether a distribution is ***skewed*** or not and if there are potential unusual observations present in the data set, which are also called ***outliers***.
- Boxplots are also very useful when huge numbers of data collections are involved or compared.

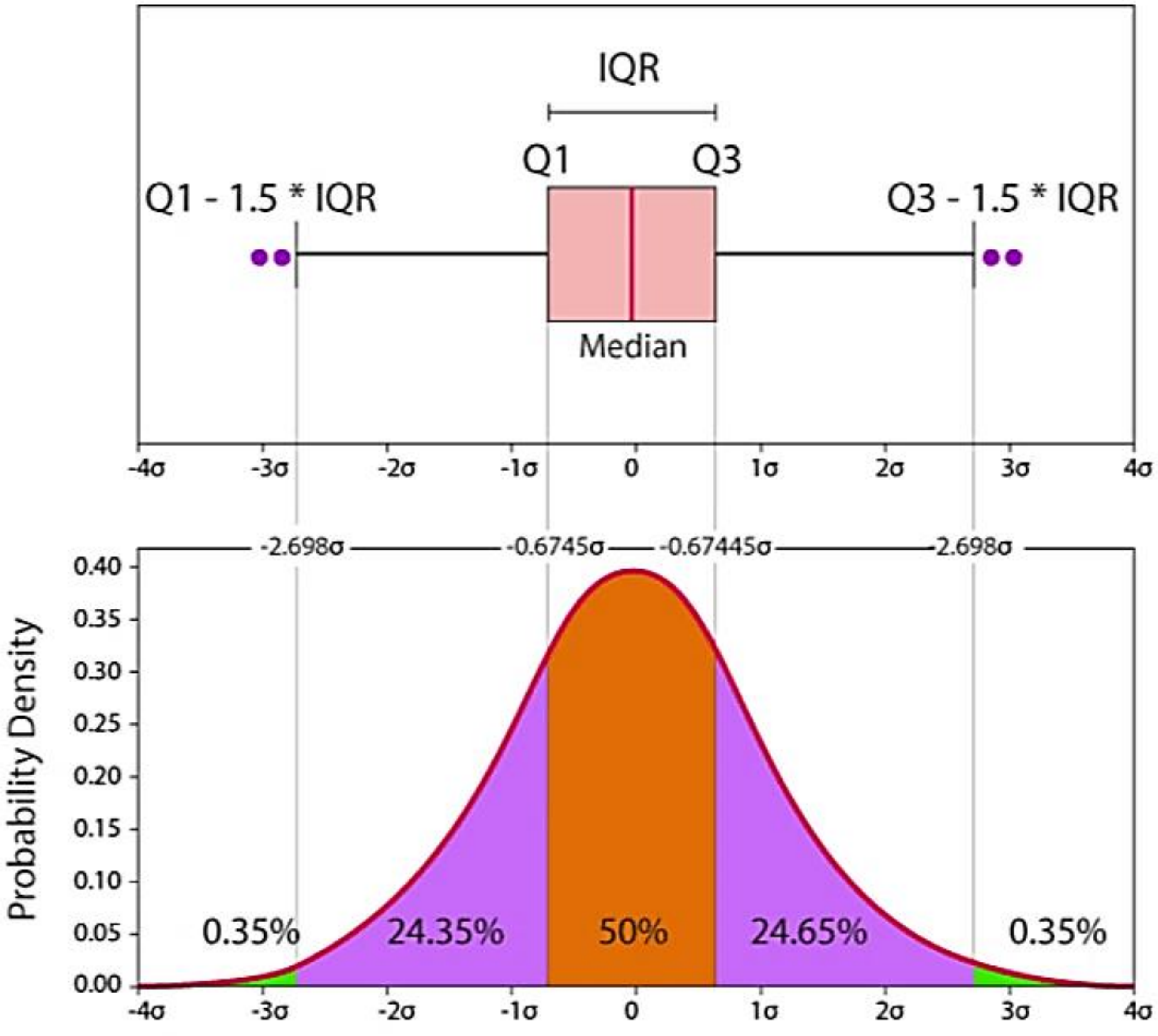
The Box and Whisker Pot

Elements of a Box and Whisker Plot

1. **Minimum value** (Q_0 or 0th percentile)
2. **First quartile** (Q_1 or 25th percentile)
3. **Median** (Q_2 or 50th percentile)
4. **Third quartile** (Q_3 or 75th percentile)
5. **Maximum value** (Q_4 or 100th percentile)



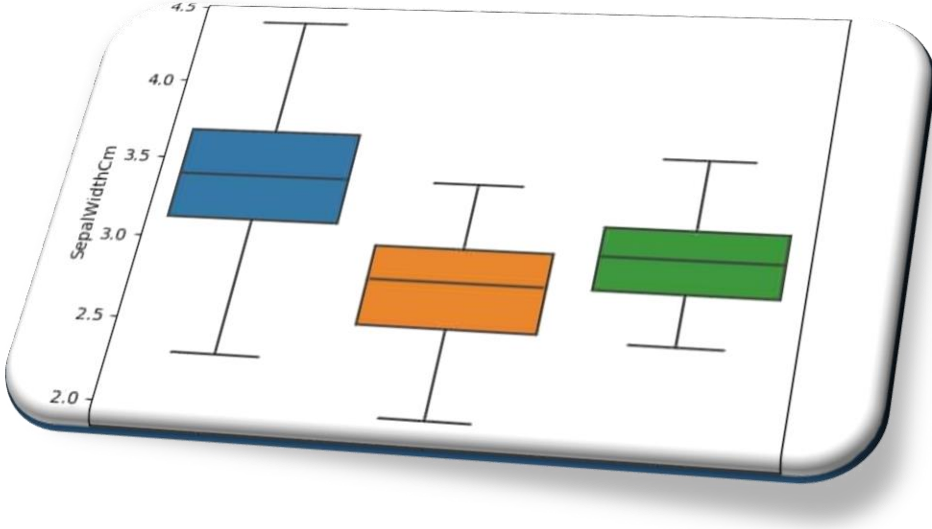
The Box and Whisker Pot



Boxplot on a normal distribution

When to Use the Box Pot?

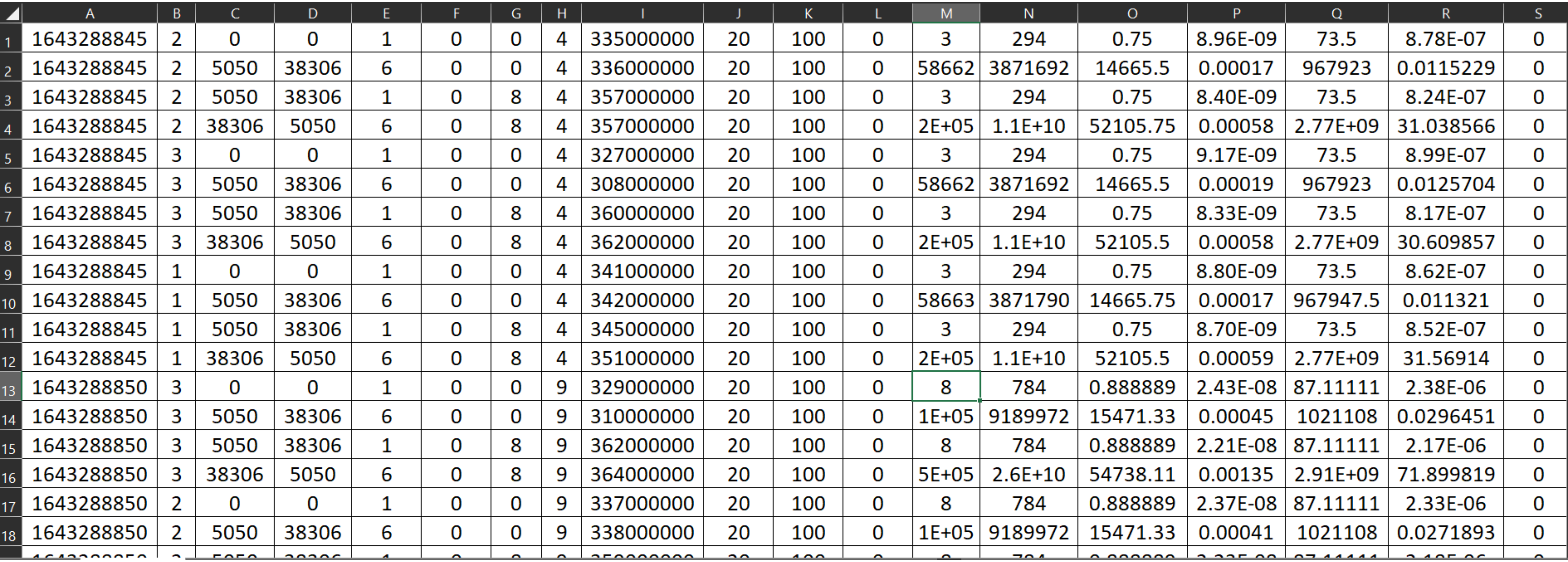
- To check the shape of the data distribution.
- It is useful when having a huge dataset.
- To show whether a distribution is skewed or have outliers.



Solve Some of Data Challenges

The scaling and the distribution

What do you see



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	1643288845	2	0	0	1	0	0	4	335000000	20	100	0	3	294	0.75	8.96E-09	73.5	8.78E-07	0
2	1643288845	2	5050	38306	6	0	0	4	336000000	20	100	0	58662	3871692	14665.5	0.00017	967923	0.0115229	0
3	1643288845	2	5050	38306	1	0	8	4	357000000	20	100	0	3	294	0.75	8.40E-09	73.5	8.24E-07	0
4	1643288845	2	38306	5050	6	0	8	4	357000000	20	100	0	2E+05	1.1E+10	52105.75	0.00058	2.77E+09	31.038566	0
5	1643288845	3	0	0	1	0	0	4	327000000	20	100	0	3	294	0.75	9.17E-09	73.5	8.99E-07	0
6	1643288845	3	5050	38306	6	0	0	4	308000000	20	100	0	58662	3871692	14665.5	0.00019	967923	0.0125704	0
7	1643288845	3	5050	38306	1	0	8	4	360000000	20	100	0	3	294	0.75	8.33E-09	73.5	8.17E-07	0
8	1643288845	3	38306	5050	6	0	8	4	362000000	20	100	0	2E+05	1.1E+10	52105.5	0.00058	2.77E+09	30.609857	0
9	1643288845	1	0	0	1	0	0	4	341000000	20	100	0	3	294	0.75	8.80E-09	73.5	8.62E-07	0
10	1643288845	1	5050	38306	6	0	0	4	342000000	20	100	0	58663	3871790	14665.75	0.00017	967947.5	0.011321	0
11	1643288845	1	5050	38306	1	0	8	4	345000000	20	100	0	3	294	0.75	8.70E-09	73.5	8.52E-07	0
12	1643288845	1	38306	5050	6	0	8	4	351000000	20	100	0	2E+05	1.1E+10	52105.5	0.00059	2.77E+09	31.56914	0
13	1643288850	3	0	0	1	0	0	9	329000000	20	100	0	8	784	0.888889	2.43E-08	87.11111	2.38E-06	0
14	1643288850	3	5050	38306	6	0	0	9	310000000	20	100	0	1E+05	9189972	15471.33	0.00045	1021108	0.0296451	0
15	1643288850	3	5050	38306	1	0	8	9	362000000	20	100	0	8	784	0.888889	2.21E-08	87.11111	2.17E-06	0
16	1643288850	3	38306	5050	6	0	8	9	364000000	20	100	0	5E+05	2.6E+10	54738.11	0.00135	2.91E+09	71.899819	0
17	1643288850	2	0	0	1	0	0	9	337000000	20	100	0	8	784	0.888889	2.37E-08	87.11111	2.33E-06	0
18	1643288850	2	5050	38306	6	0	0	9	338000000	20	100	0	1E+05	9189972	15471.33	0.00041	1021108	0.0271893	0

Data Scaling Methods

- It is common to scale data before feeding it into a machine learning model. This is because data often consists of many different input variables or features (columns), and each may have a different range of values or units of measure, such as **feet, miles, kilograms, dollars**, etc.
- If input variables have very large values relative to the other input variables, these large values can dominate or skew some machine learning algorithms.
- ***The result is that the algorithms pay most of their attention to the large values and ignore the variables with smaller values.***

Complex the process → **Reduce the accuracy** → **Increasing the time**

Scaling Features to a Range

- An alternative standardization is scaling features to lie between a given minimum and maximum value, often between zero and one, or so that the maximum absolute value of each feature is scaled to unit size. Like:
- **MinMaxScaler:**

$$y = \frac{x - min(x)}{max(x) - min(x)}$$

  - *sklearn.preprocessing.MinMaxScaler()*

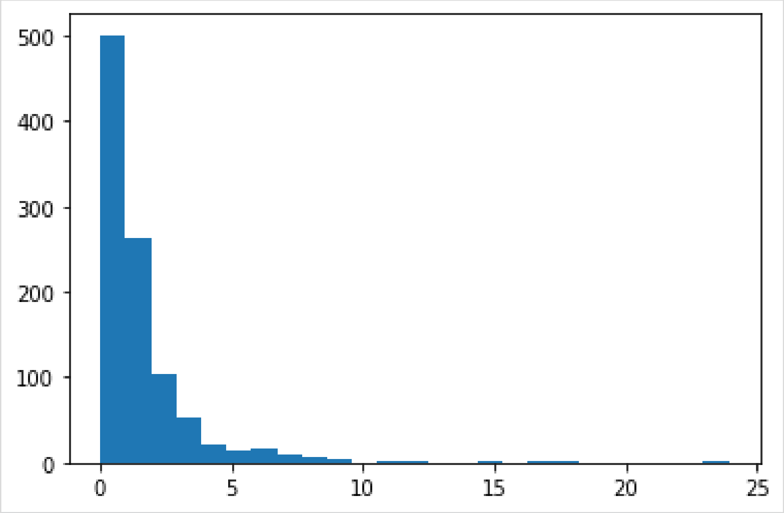
- **StandardScaler:**

$$y = \frac{x - mean(x)}{std(x)}$$

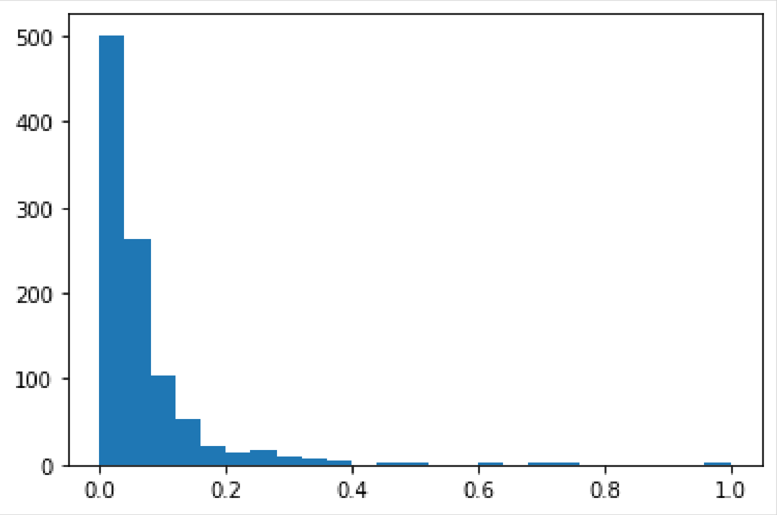
  - *sklearn.preprocessing.StandardScaler()*

Data Standardization

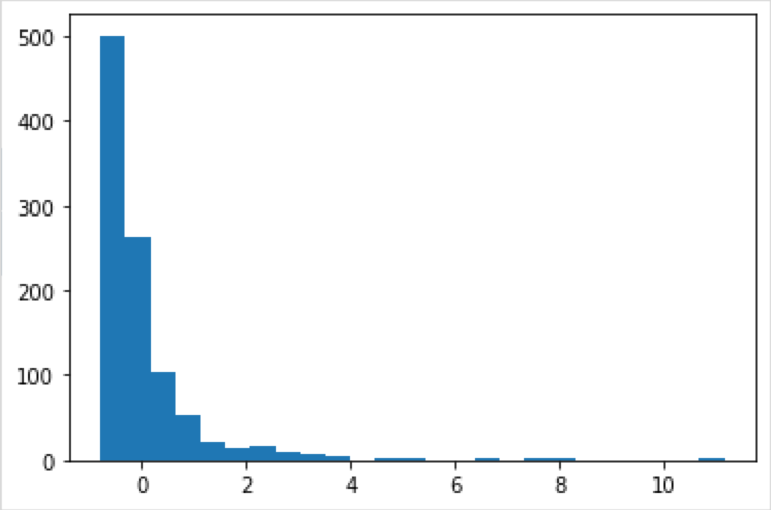
Input Data



MinMaxScaler



StandardScaler



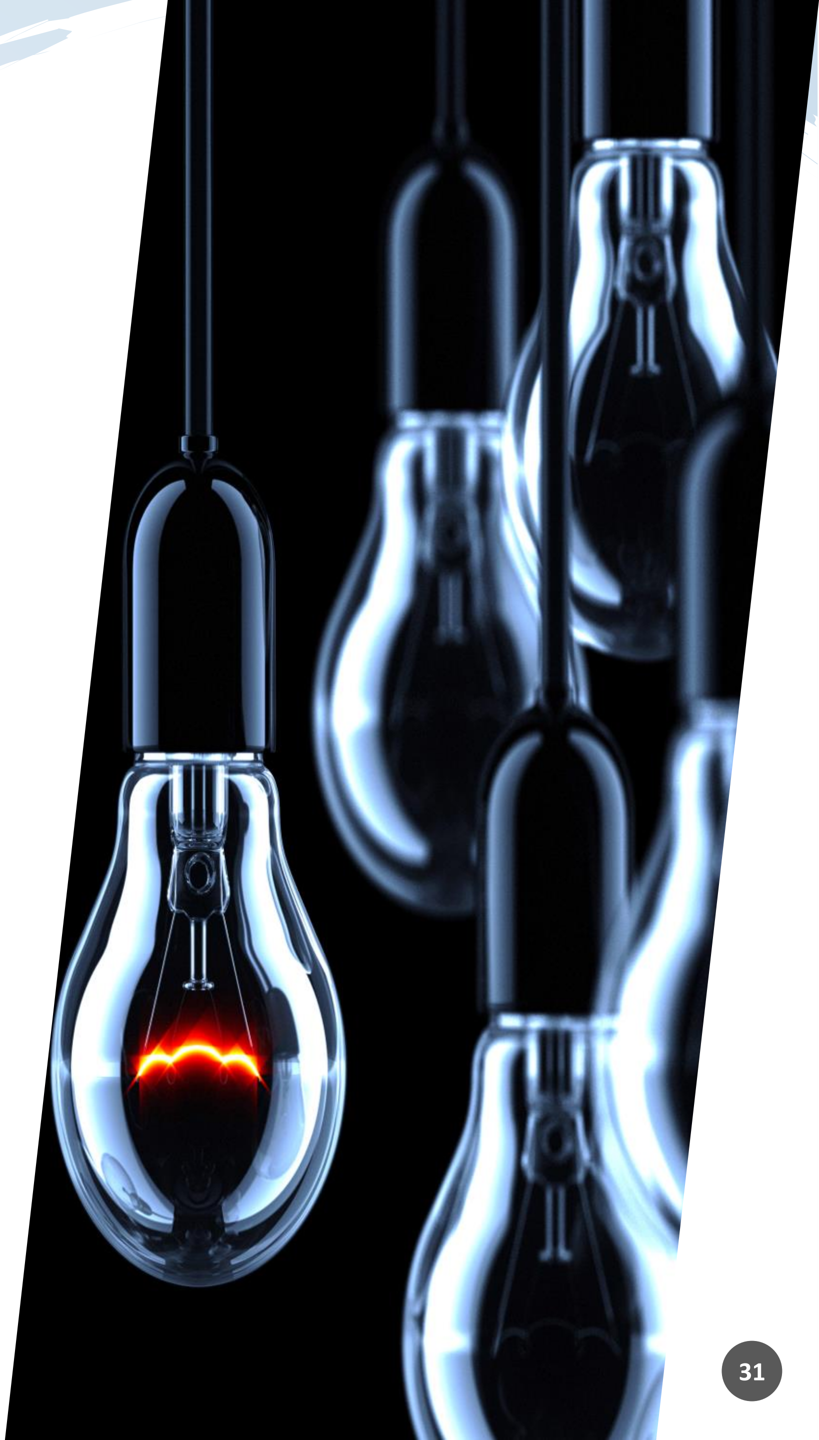
What do you see, and what the results ?

Data Standardization

What kind of problems can we solve using data standardization or scaling methods?

Abnormal Distribution

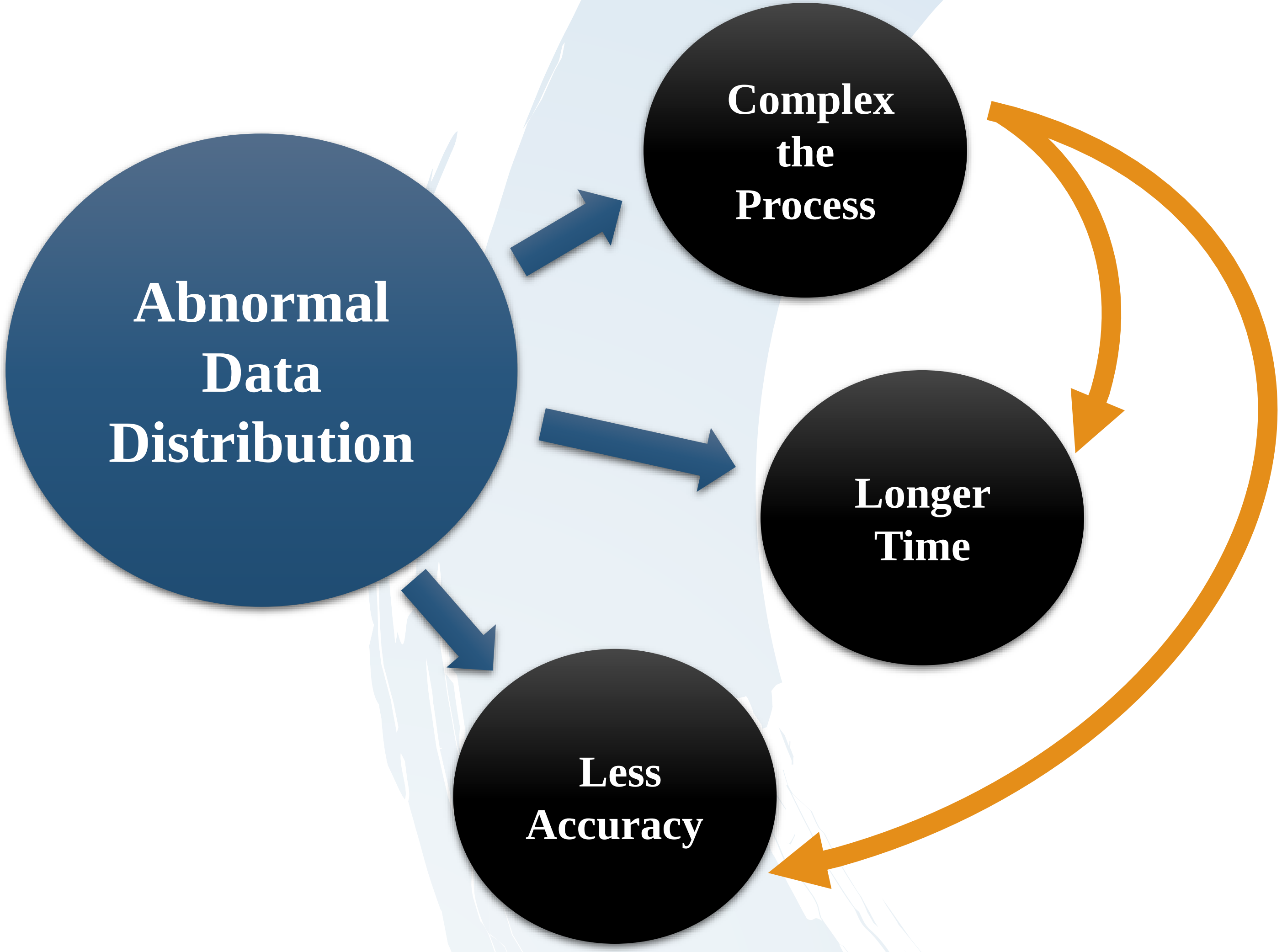
- Machine learning algorithms like Linear Regression and Gaussian Naive Bayes assume the numerical variables have a Gaussian probability distribution.
- Your data may not have a Gaussian distribution.
- Instead, they may have a Gaussian-like distribution (e.g., nearly Gaussian but with outliers or a skew) or
- a totally different distribution (e.g., exponential).



Abnormal Distribution

What do you think, which kind of difficulties you can face if your data has abnormal distribution?

Abnormal Distribution



Abnormal Data Distribution

Complex the Process

Longer Time

Less Accuracy

Gaussian/Uniform/Normal Distribution

- Some input variables may have a highly skewed distribution, such as an exponential distribution where the most common observations are bunched together. Some input variables may have outliers that cause the distribution to be highly spread.

1. The quantile function ranks or smooths out the relationship between observations and can be mapped onto other distributions, such as the ***uniform*** or ***normal*** distribution.
    - This quantile transform is available in the scikit-learn Python machine learning library via the ***QuantileTransformer*** class.
    - The class has an “output_distribution” argument that can be set to “***uniform***” or “***normal***” and defaults to “*uniform*“.

Gaussian/Uniform/Normal Distribution

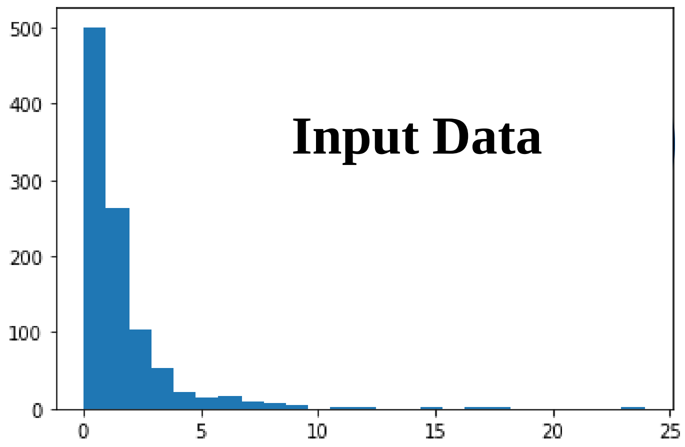
- **Make Data More Gaussian (normal distribution)**
  - ✓ The Gaussian is a common distribution with the familiar bell shape.
  - ✓ These transforms are most effective when the data distribution is nearly-Gaussian, to begin with, and is affected by skew or outliers.

2. Power Transforms

A power transform will make the probability distribution of a variable more Gaussian. This is often described as removing a skew in the distribution. There are two popular approaches (method) for such automatic power transforms:

1. Box-Cox Transform
2. Yeo-Johnson Transform

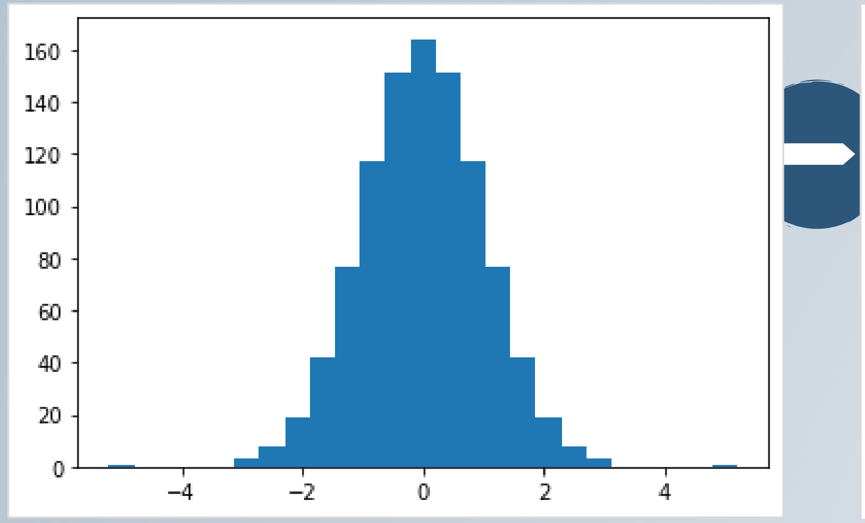
Gaussian/Uniform Distribution

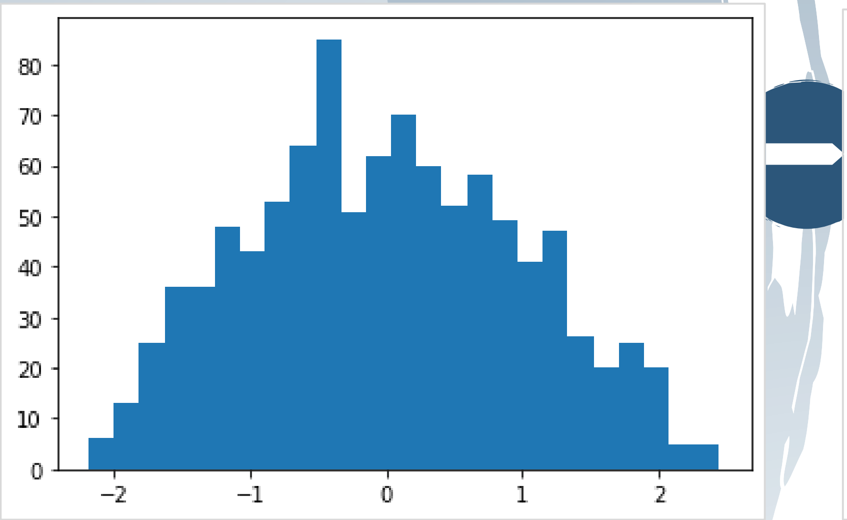

Input Data

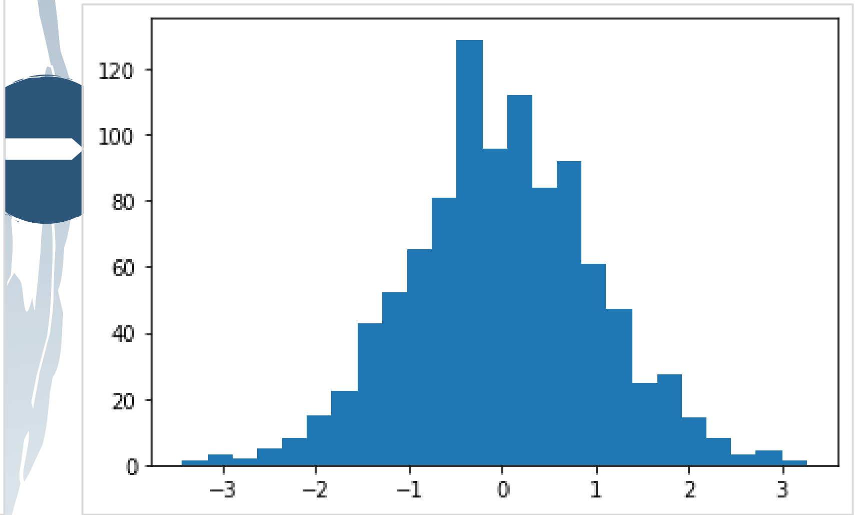
QuantileTransformer →'normal'

PowerTransformer →'yeo-johnson'

PowerTransformer →'box-cox'

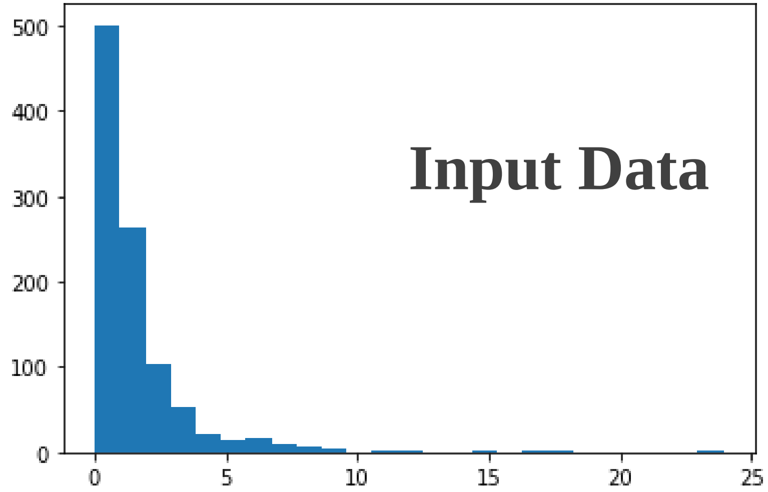


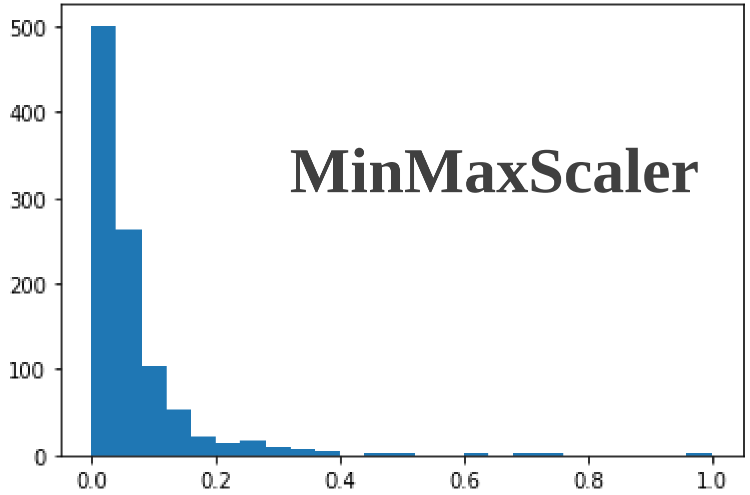


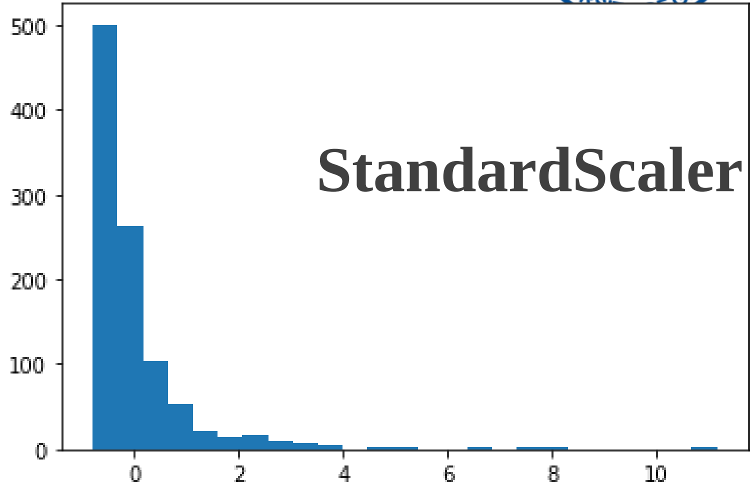


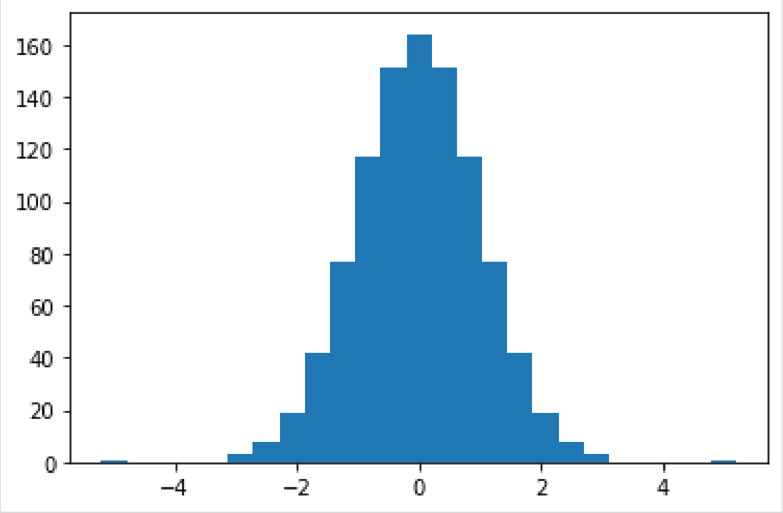
What do you see

Data Distribution

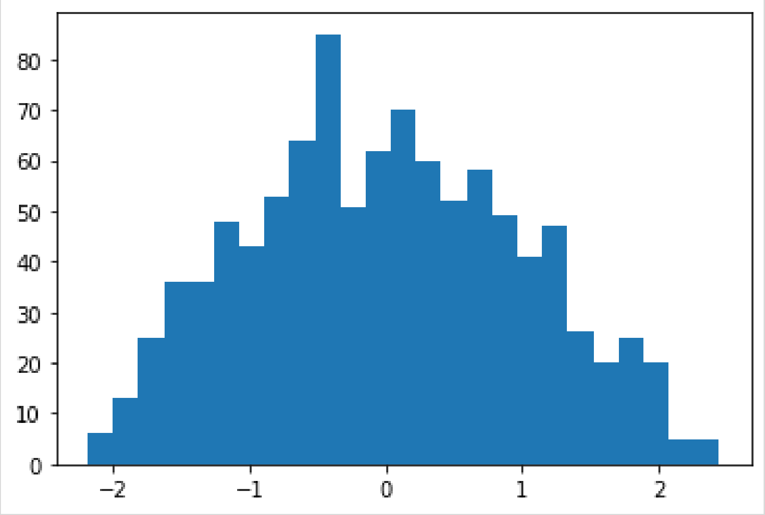




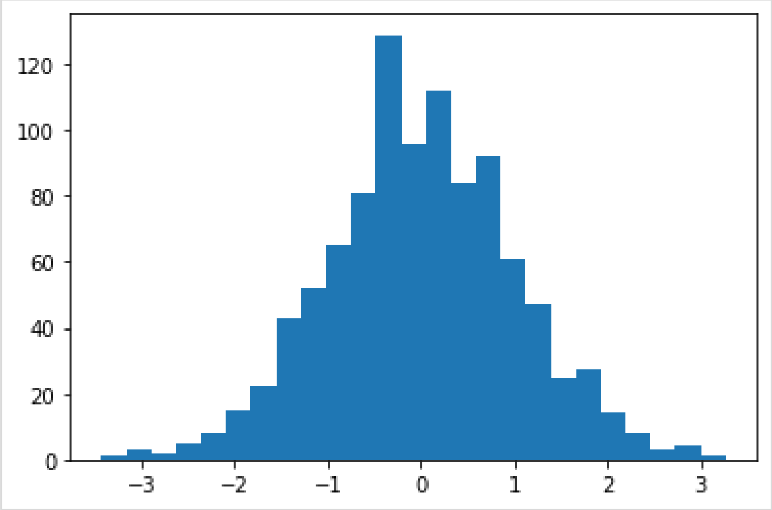




QuantileTransformer
→'normal'



PowerTransformer
→'yeo-johnson'



PowerTransformer
→'box-cox'

Dimensionality Reduction Methods

The enhancing and reducing

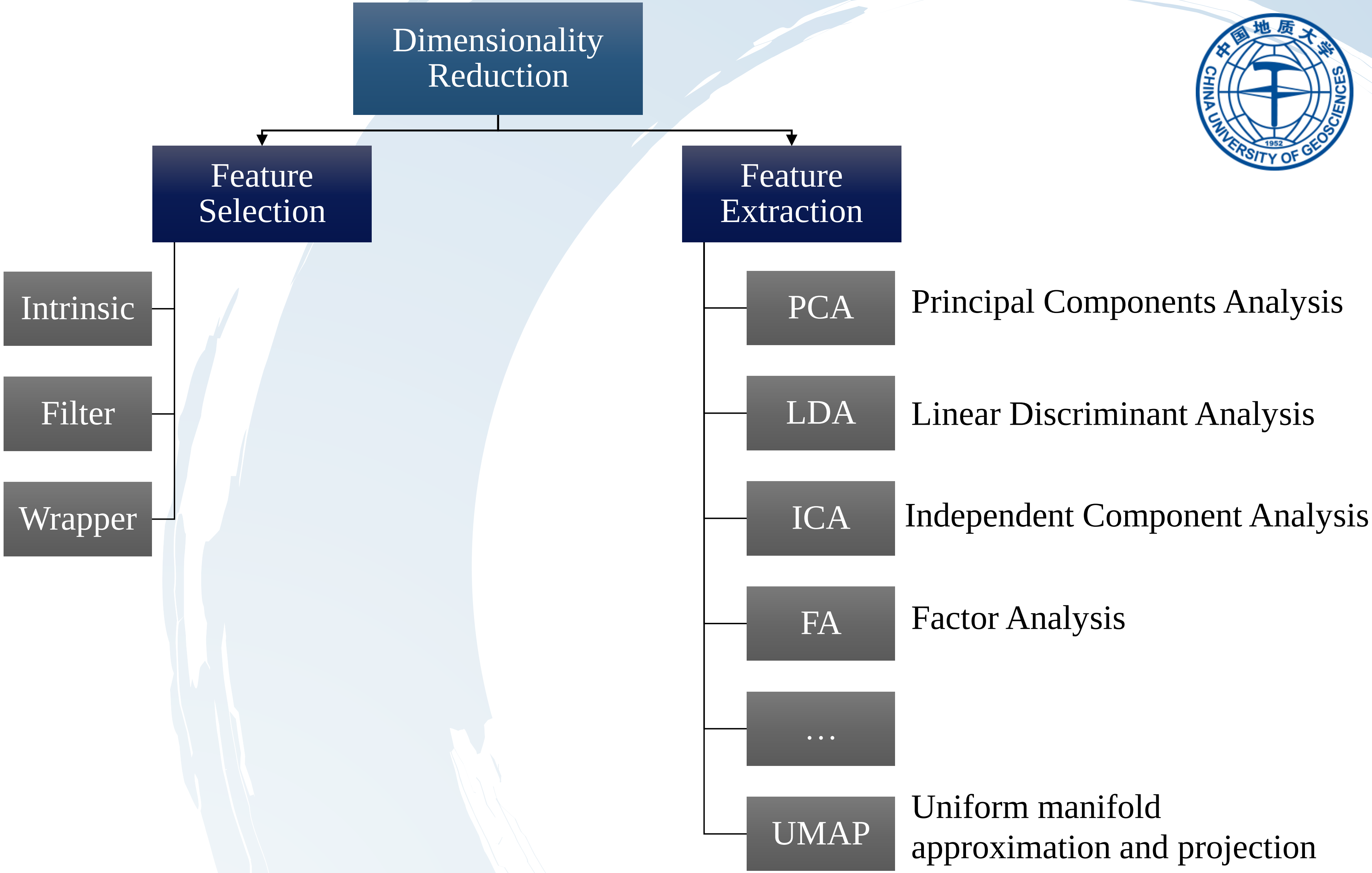
Dimensionality Reduction

- The number of input variables or features for a dataset is referred to as its dimensionality.
- Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.
- More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.

- ✓ ***Difficult visualization,***
- ✓ ***Complex method,***
- ✓ ***Longer time,***
- ✓ ***Large numbers of input features can cause poor performance for AI algorithms.,***
- ✓ ***Overfitting***
- ✓ ***More resources***

Dimensionality Reduction

- Dimensionality reduction can be used for
  - ✓ ***noise reduction,***
  - ✓ ***data visualization,***
  - ✓ ***cluster analysis,***
  - ✓ ***as an intermediate step to facilitate other analyses,***
  - ✓ ***speed up the process, and***
  - ✓ ***resource reduction***



Dimensionality Reduction

Feature Selection

- Intrinsic
- Filter
- Wrapper

Feature Extraction

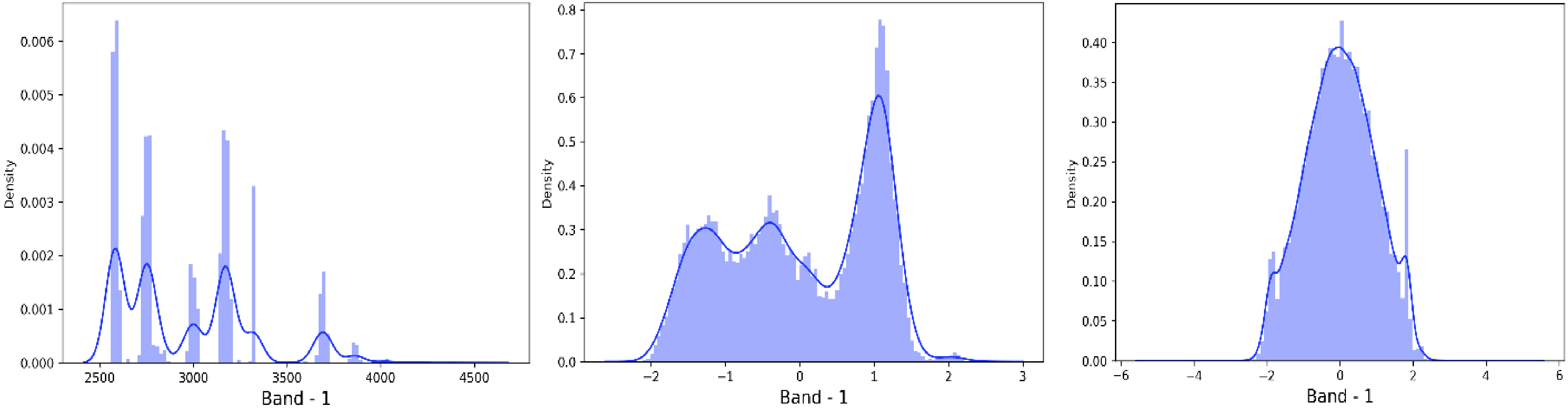
- PCA Principal Components Analysis
- LDA Linear Discriminant Analysis
- ICA Independent Component Analysis
- FA Factor Analysis
- …
- UMAP Uniform manifold approximation and projection

Dimensionality Reduction Methods

Dimensionality Reduction

- Compression and Reinforced Variation (CRV)
- Improving Distribution Analysis (IDA)
- Enhancing Transformation Reduction (ETR)
- https://github.com/DalalAL-Alimi

QPCA

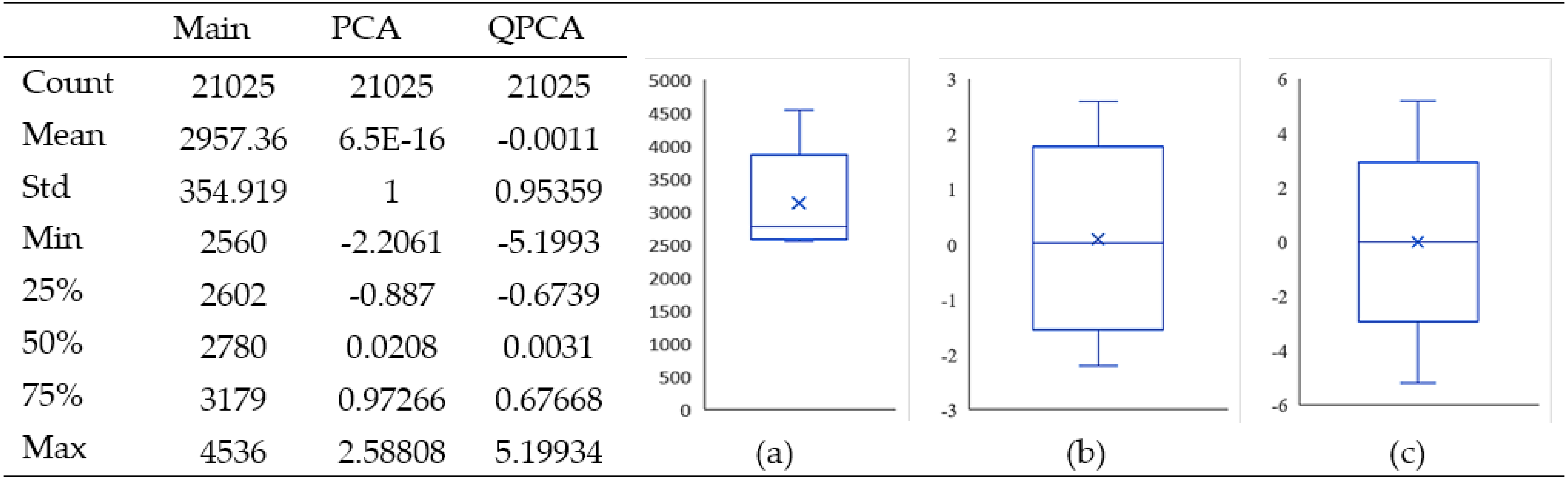


(a) (b) (c)

(a) is the original data, (b) is the change of data distribution after using principal components analysis (PCA), and (c) is the data distribution after using quantile transformation principal components analysis (QPCA).

https://www.mdpi.com/2072-4292/14/4/1038/htm

QPCA



	Main	PCA	QPCA
Count	21025	21025	21025
Mean	2957.36	6.5E-16	-0.0011
Std	354.919	1	0.95359
Min	2560	-2.2061	-5.1993
25%	2602	-0.887	-0.6739
50%	2780	0.0208	0.0031
75%	3179	0.97266	0.67668
Max	4536	2.58808	5.19934

(a) (b) (c)

(a) is the distribution of the original data, (b) is for the data after using PCA, and (c) is the shape of the data after using QPCA.

https://www.mdpi.com/2072-4292/14/4/1038/htm

QPCA

#	PCA-SVM	QPCA-SVM	PCA-CNN1d	QPCA-CNN1d	PCA-VGG-16	QPCA-VGG-16	PCA-Hybrid CNN	QPCA-Hybrid CNN	PCA-MLHM	QPCA-MLHM
KA (%)	80	73.91	80.39	78.33	83.77	94.68	98.96	99.25	99.39	99.41
OA (%)	82.50	77.16	82.84	81.03	85.87	95.34	99.09	99.34	99.47	99.47
AA (%)	80.28	70.63	79.48	78.24	66.58	81.89	98.17	98.71	99.01	99.36
Training Time (s)	0.41	0.67	21.87	36.04	53.44	50.95	65.60	60.40	41.67	25.47
Test Time (s)	2.40	2.45	0.25	0.27	2.20	2.24	1.93	1.83	4.53	3.64

https://www.mdpi.com/2072-4292/14/4/1038/htm

Considerations for Training the DL Model

Methods and Parameters

Considerations for the Training Model

- During the training time, we use many concepts and techniques to improve output and the accuracy of results.
- Some of these methods or techniques:

1. **Weight Initialization**
   - Using deep or many layers of Neural Networks (NN) cause exploding or vanishing gradients thus affect badly in backwards operation rather than improve the accuracy by adding more layers.
   - Due to this reason, weight initialization is used to prevent the outputs of activation layer from exploding or vanishing during the track of a forward pass through a deep neural network,
   - also using weight initialization speeds up the learning process and to keep the variance of the activations are the same in all network layers.

Considerations for the Training Model

- Some methods of the weights initializing:
  - ✓ random_normal_initializer
  - ✓ random_uniform
  - ✓ variance_scaling_initializer
  - ✓ zeros_initializer
  - ✓ keras.layers.BatchNormalization

2. Batch Normalization

- Batch Normalization (BN) is widely used to improve performance of models. Due to using deeper NN, two main problems appear:

1. **Exploding Gradient.**
2. **Vanishing Gradient.**

Considerations for the Training Model

- The main aim behind BN is to limit the covariate shift by normalizing each layer's activations.
- It, hopefully, helps each layer to learn with more stable input distribution, thus improving the network's training. Batch normalization firstly works to calculate the mean and variance inputs of layers.
- Then, to improve a neural network's stability, the output of a previous activation layer is normalized by subtracting the batch mean and dividing it by the standard deviation.
- The advantages of BN as the following:

1. **Faster and higher learning rate. → to get the target accuracy**
2. **Reducing the Covariate Shift.**
3. **Reducing the overfitting (Reducing the need for dropout).**
4. **Avoiding Vanishing Gradient.**

Considerations for the Training Model

3. **Cost Function**

- Binary for two classes
- with more than two classes, we need to use cross-entropy loss and Softmax classifiers functions.

~~Optimization Techniques~~

- Many optimization techniques are used to improve the learning ability of neural networks to solve many complex issues. For example, in machine learning, there are two types of parameters that can be used in any mode to optimize it.

  1. **Model parameters**

  - they are machine learning algorithms and a part of the learned model. Their values are not constant and depend on input and hyperparameters values. This means their values are updated and changed many times during the training time until getting the best weight or accuracy. Weight (w) and bias (b) are examples of model parameters.

Optimization Techniques

2. Hyperparameters

- These are parameters with constant values that must be tuned and edited by trial and error in order to obtain the model with optimal performance.
- Also, the potential values of these parameters are not changed or affected throughout the training.
- The intention of hyperparameters is to modify the weight and the bias and additionally to enhance the performance and the velocity of the training.
- The examples of hyperparameters include:

✓ **Learning rate (α)**

✓ **Iteration number (epoch)**

✓ **Number of hidden layers**

✓ **Number of hidden units**

✓ **Momentum**

✓ **Minibatch size**

✓ **Regularization parameter (λ)**

Activation Function

The Differences and Work

What is Activation Function?

- It is a function used to modify (control) and get the output of the node. It is also known as Transfer Function.

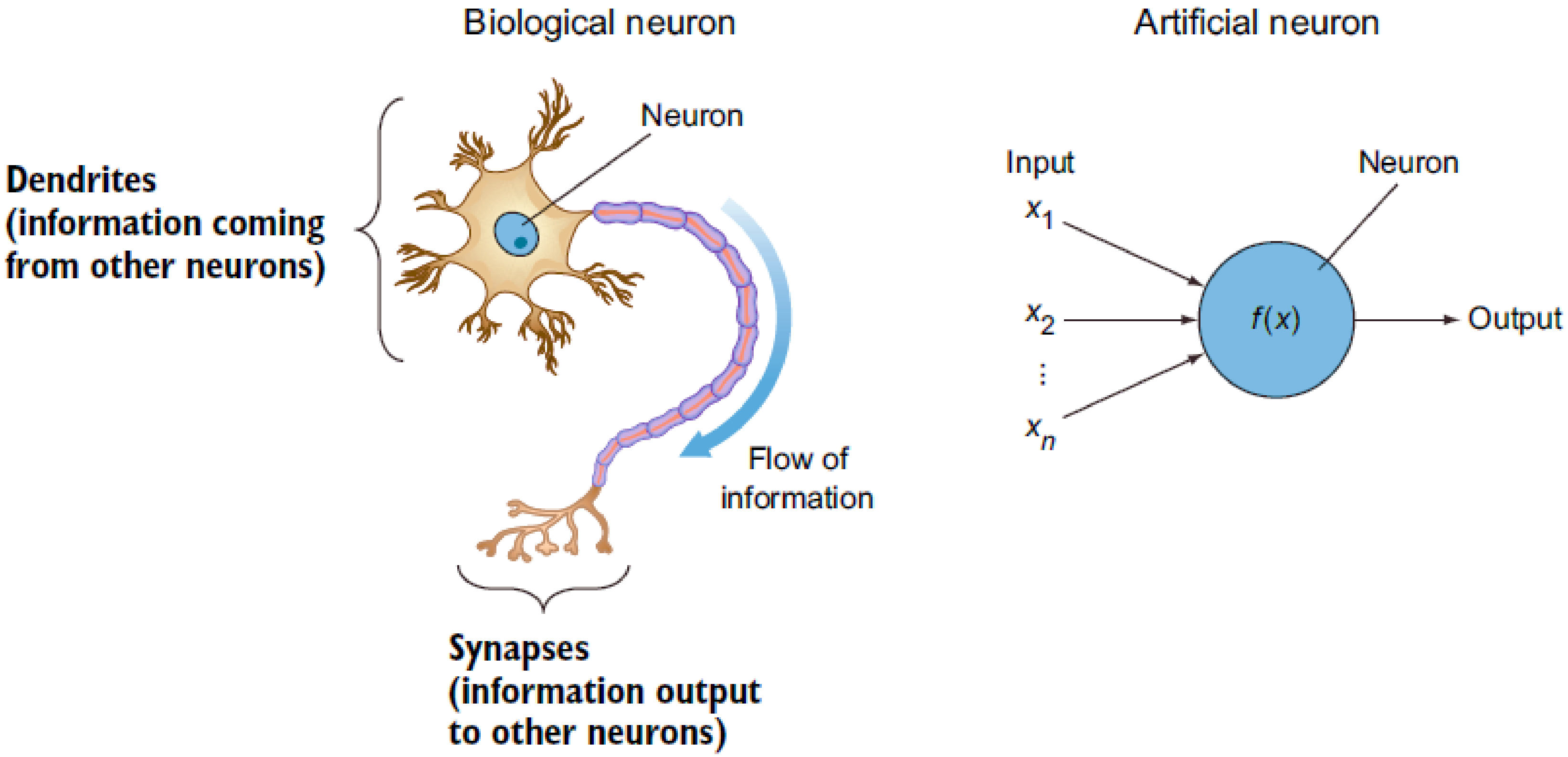
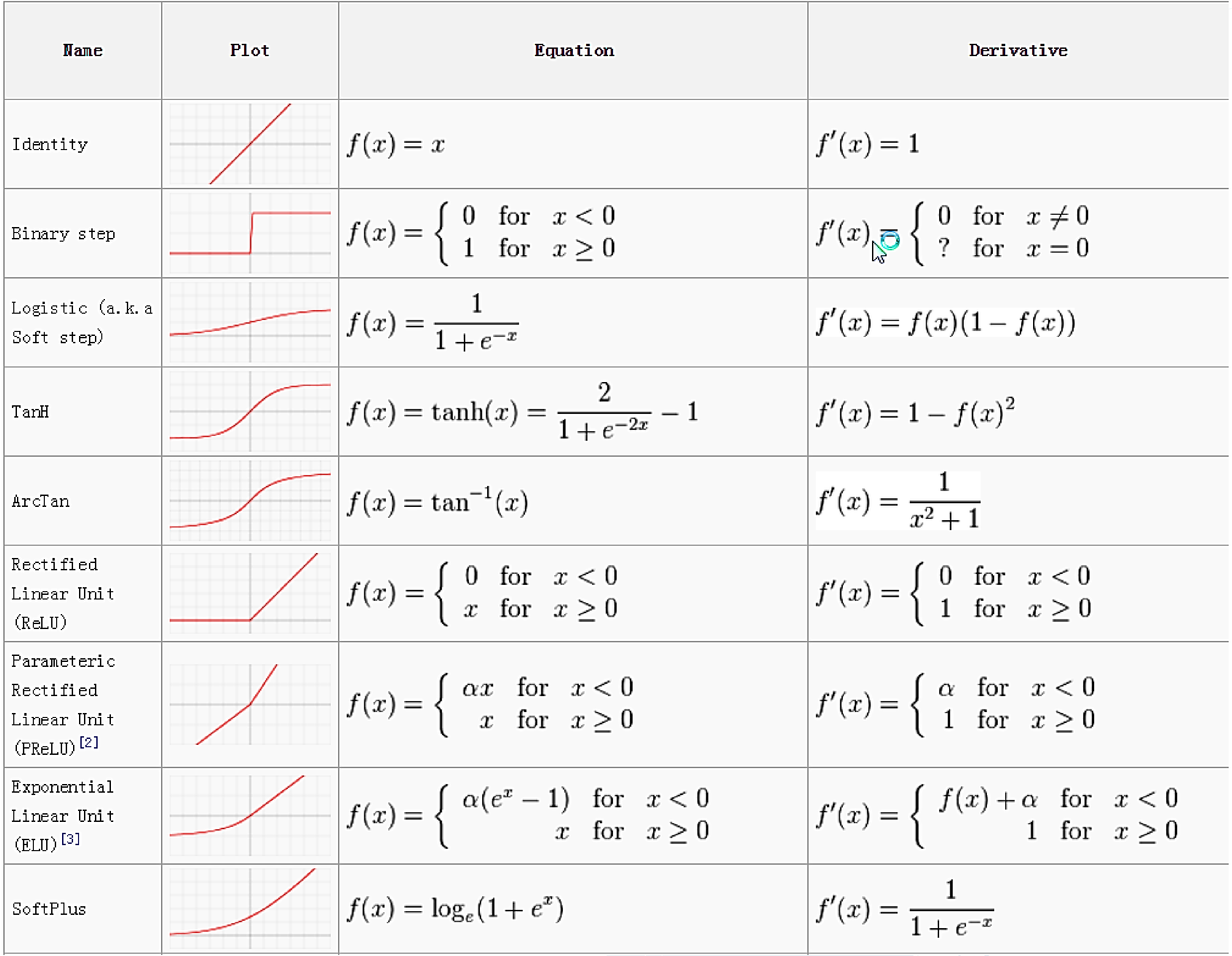


Figure 1.3 The similarities between biological neurons and artificial systems



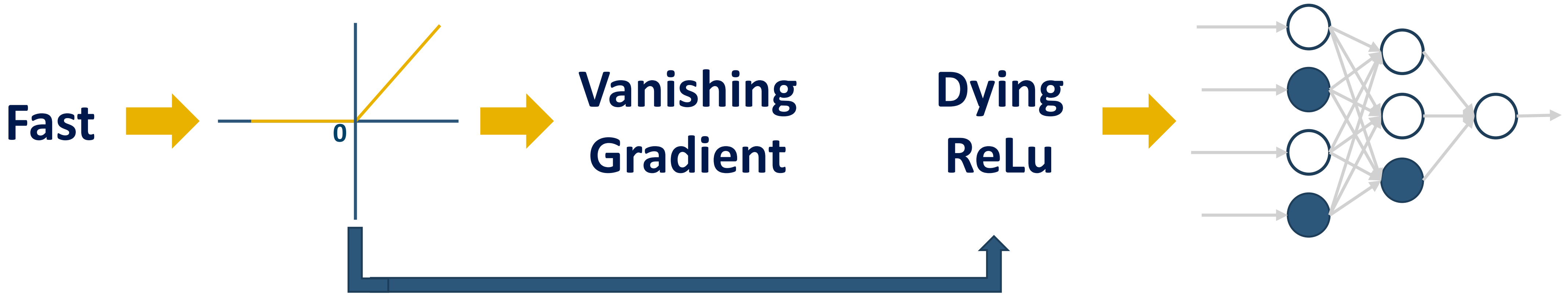
Name	Plot	Equation	Derivative
Identity		$f(x)=x$	$f'(x)=1$
Binary step		$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x\neq 0\\ ? & \text{for } x=0\end{cases}$
Logistic (a.k.a Soft step)		$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$
TanH		$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$
ArcTan		$f(x)=\tan^{-1}(x)$	$f'(x)=\frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x)=\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}f(x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
SoftPlus		$f(x)=\log_e(1+e^x)$	$f'(x)=\frac{1}{1+e^{-x}}$

Relu VS ELU

Its Differences and When It is Used

ReLu VS ELU Activation Functions (AF)

- ReLu is an activation function used in many kind of NN.

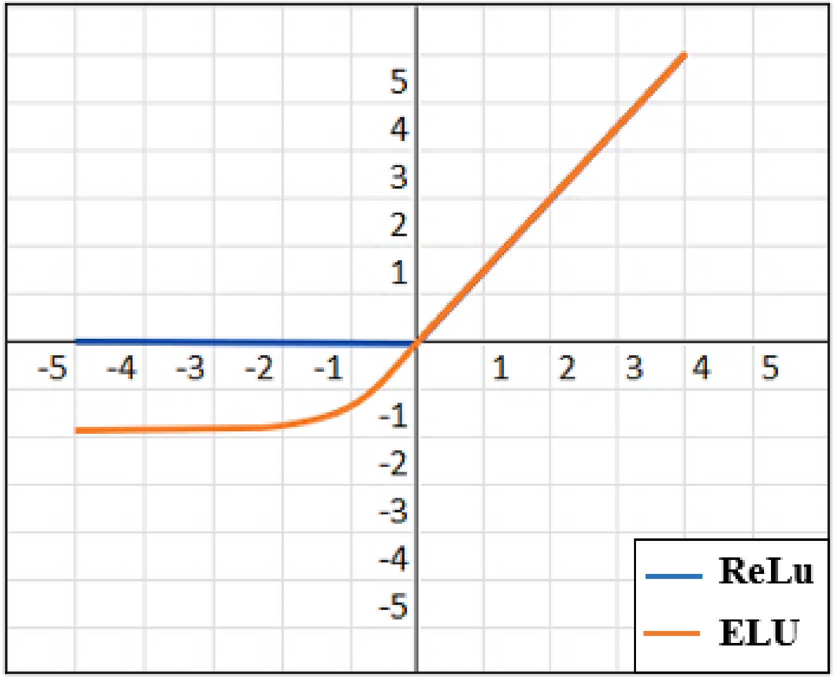


Fast

Vanishing Gradient

Dying ReLu

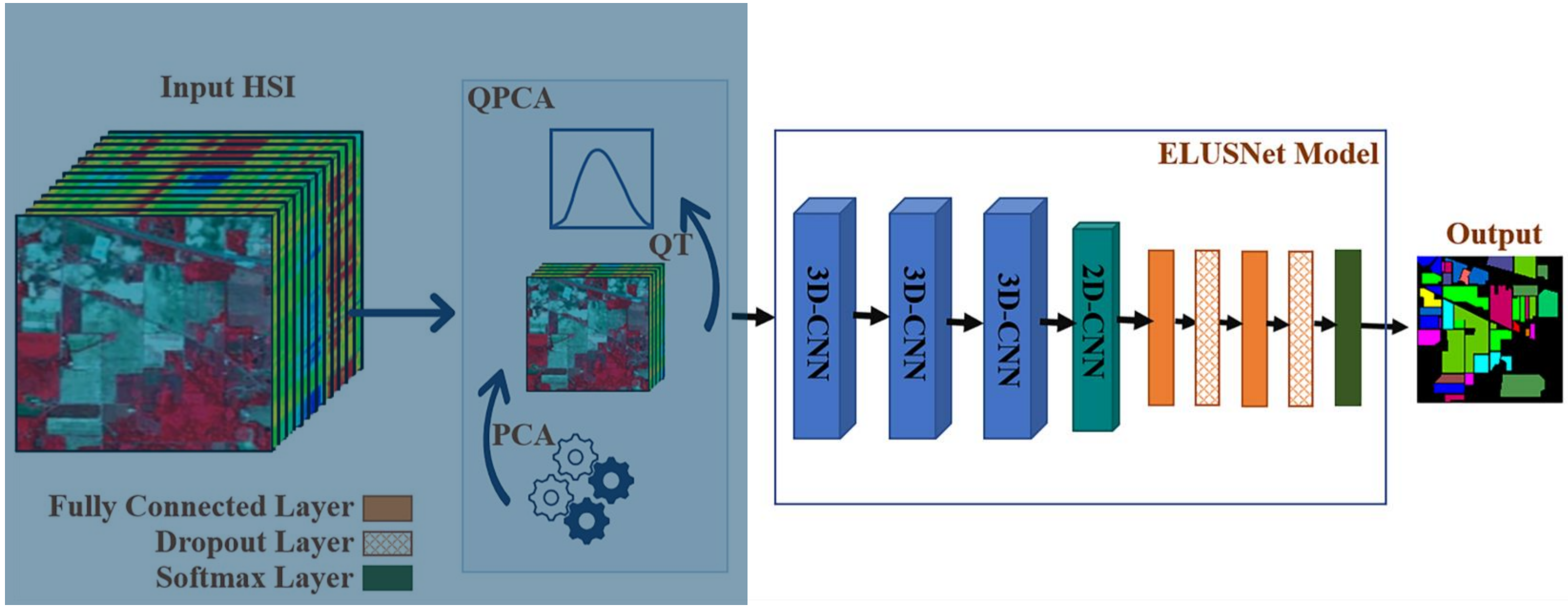
ReLu VS ELU Activation Functions (AF)

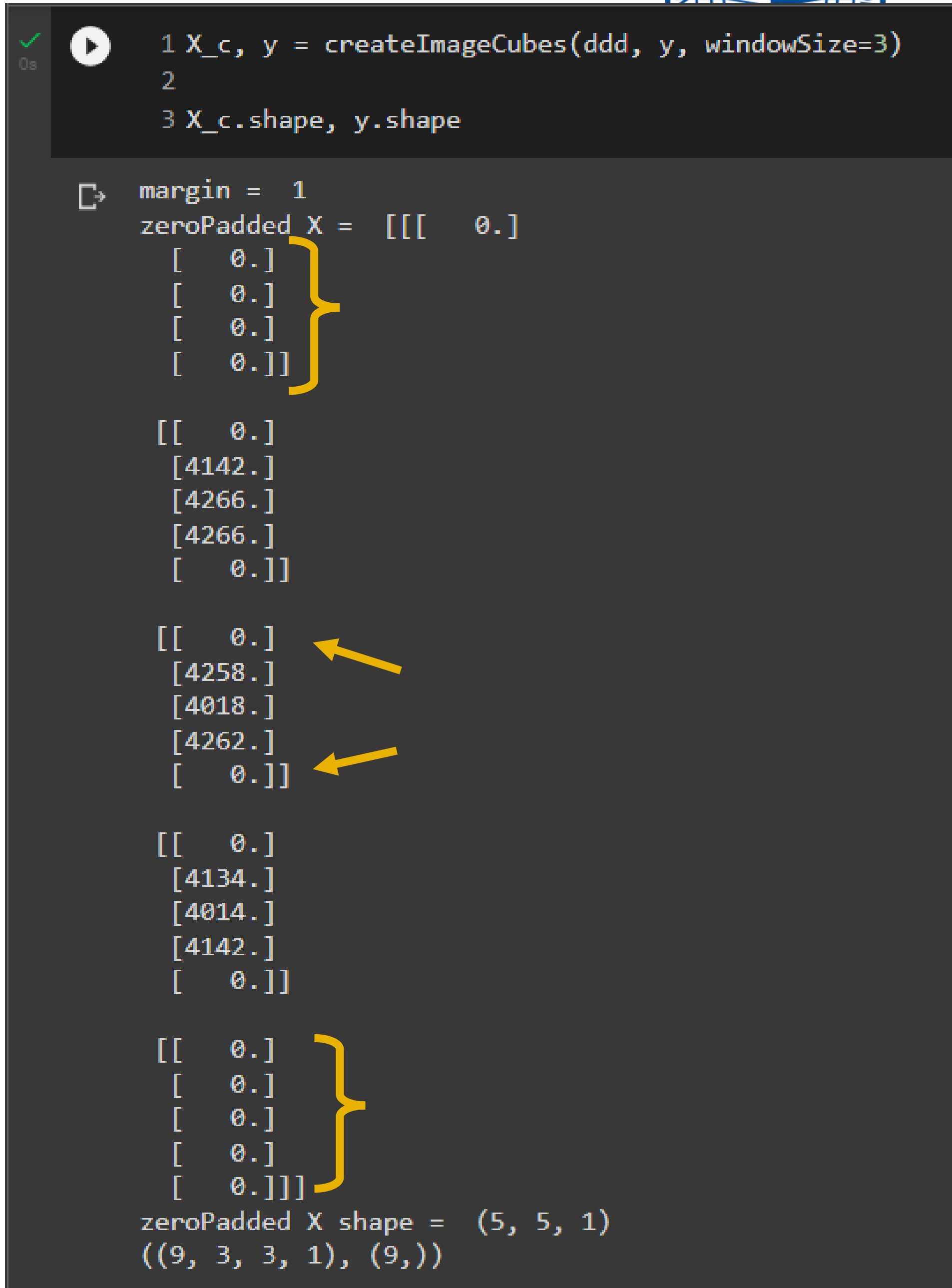


$$relu(x) = \begin{cases} 0 & if\ x \leq 0 \\ x & if\ x > 0 \end{cases} = max(0, x)$$

$$elu(x) = \begin{cases} \exp(x) - 1 & if\ x \leq 1 \\ x & if\ x > 1 \end{cases}$$

ReLu VS ELU AF

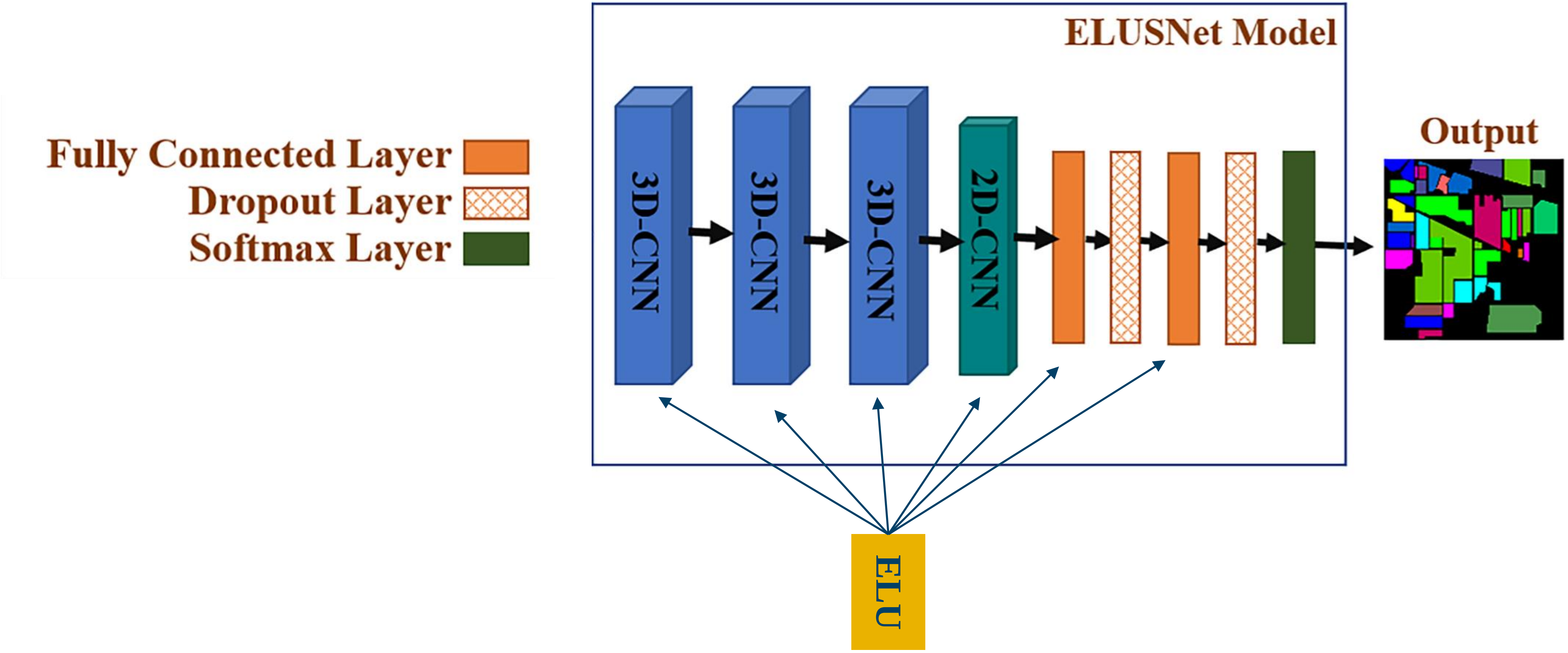




```
X_c, y = createImageCubes(ddd, y, windowSize=3)

X_c.shape, y.shape
```

```
margin =  1
zeroPadded X =  [[[    0.]
  [    0.]
  [    0.]
  [    0.]
  [    0.]]

 [[    0.]
  [4142.]
  [4266.]
  [4266.]
  [    0.]]

 [[    0.]
  [4258.]
  [4018.]
  [4262.]
  [    0.]]

 [[    0.]
  [4134.]
  [4014.]
  [4142.]
  [    0.]]

 [[    0.]
  [    0.]
  [    0.]
  [    0.]
  [    0.]]]
zeroPadded X shape =  (5, 5, 1)
((9, 3, 3, 1), (9,))
```

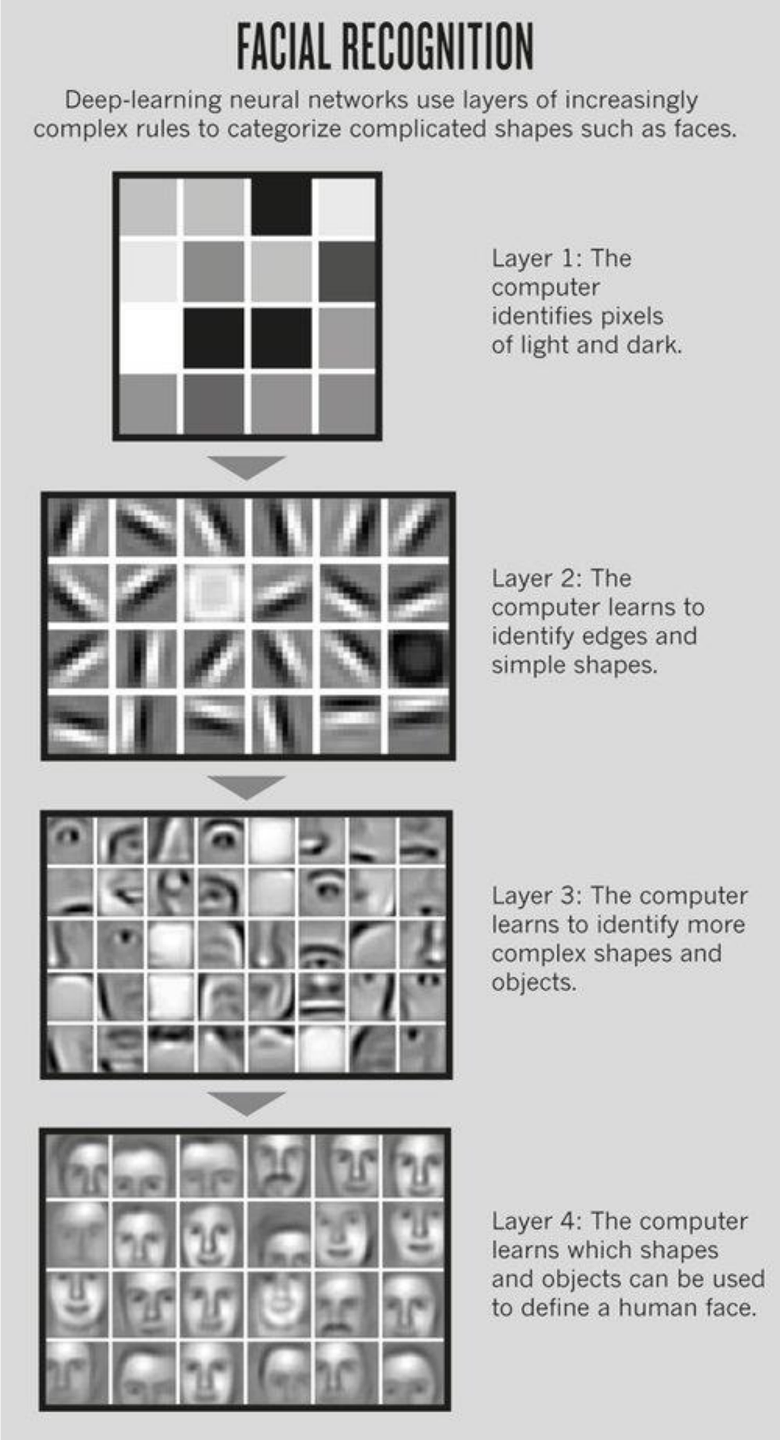
ReLu VS ELU AF



ReLu VS ELU Activation Functions (AF)

ELU

1. Avoiding dying neurons.
2. Normalizing the training.
3. Avoiding the vanishing gradient.
4. Speeding up the performance.



FACIAL RECOGNITION

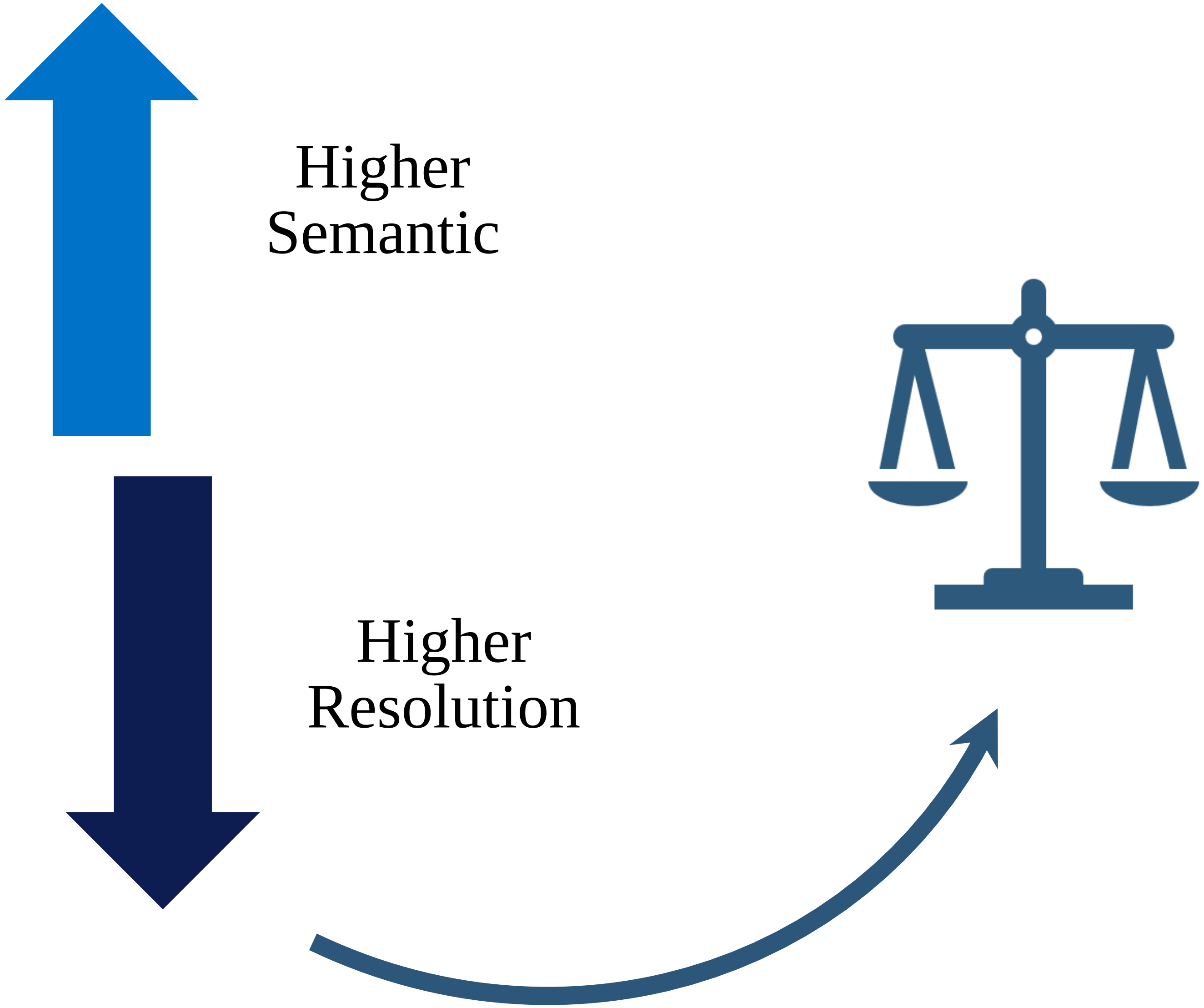
Deep-learning neural networks use layers of increasingly complex rules to categorize complicated shapes such as faces.

Layer 1: The computer identifies pixels of light and dark.

Layer 2: The computer learns to identify edges and simple shapes.

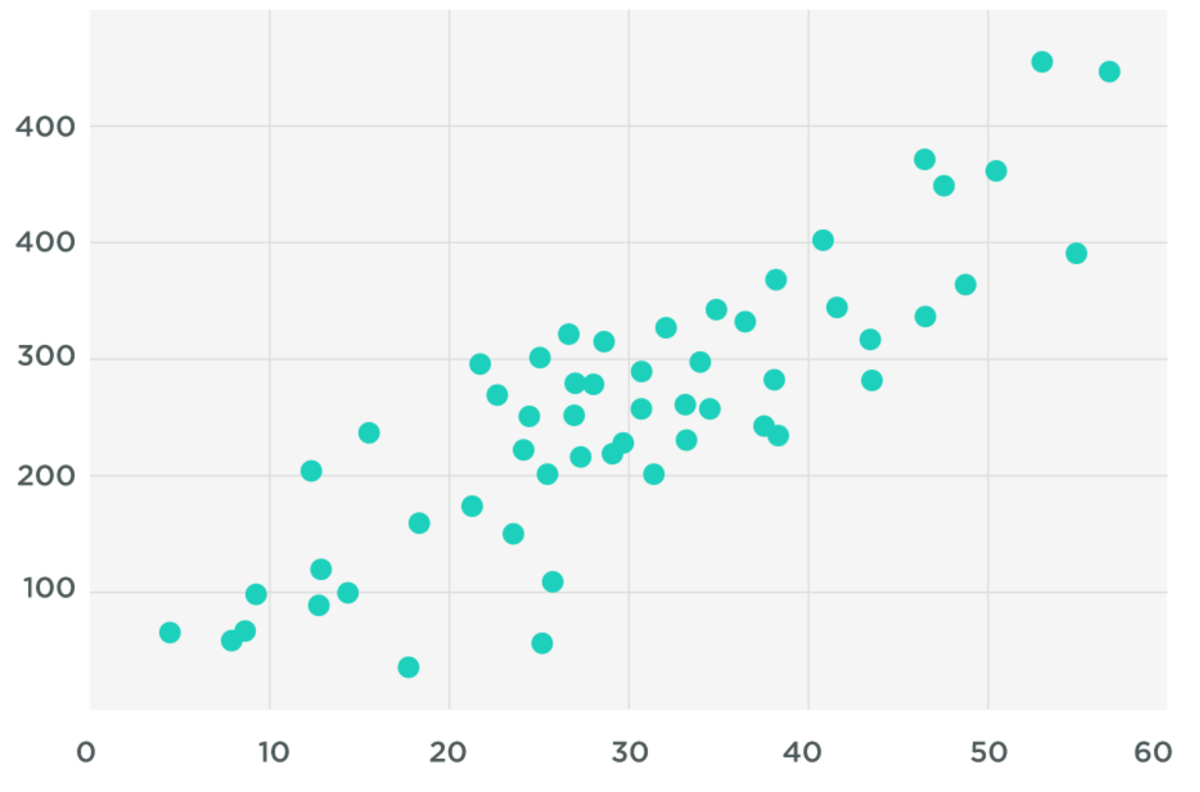
Layer 3: The computer learns to identify more complex shapes and objects.

Layer 4: The computer learns which shapes and objects can be used to define a human face.



Higher Semantic

Higher Resolution


400
400
300
200
100
0
10
20
30
40
50
60

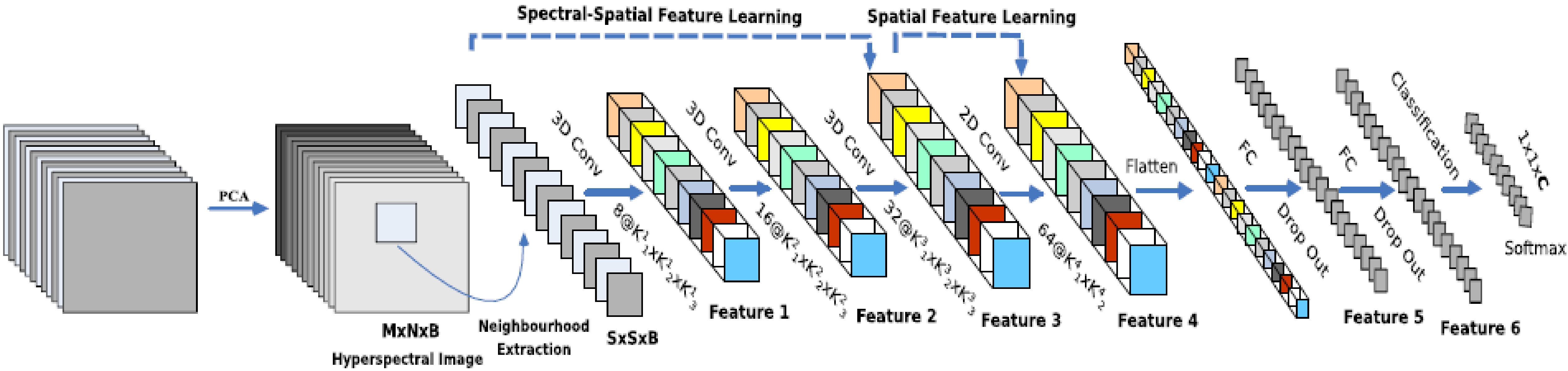
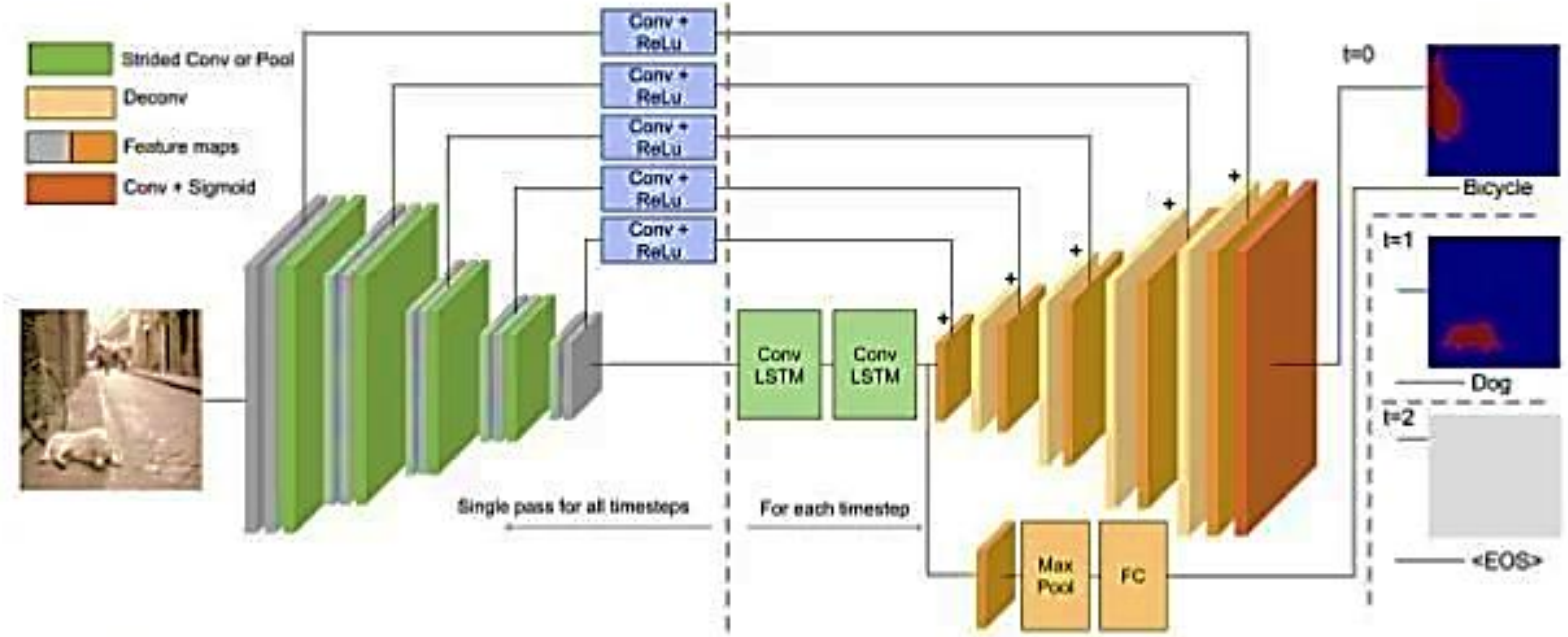
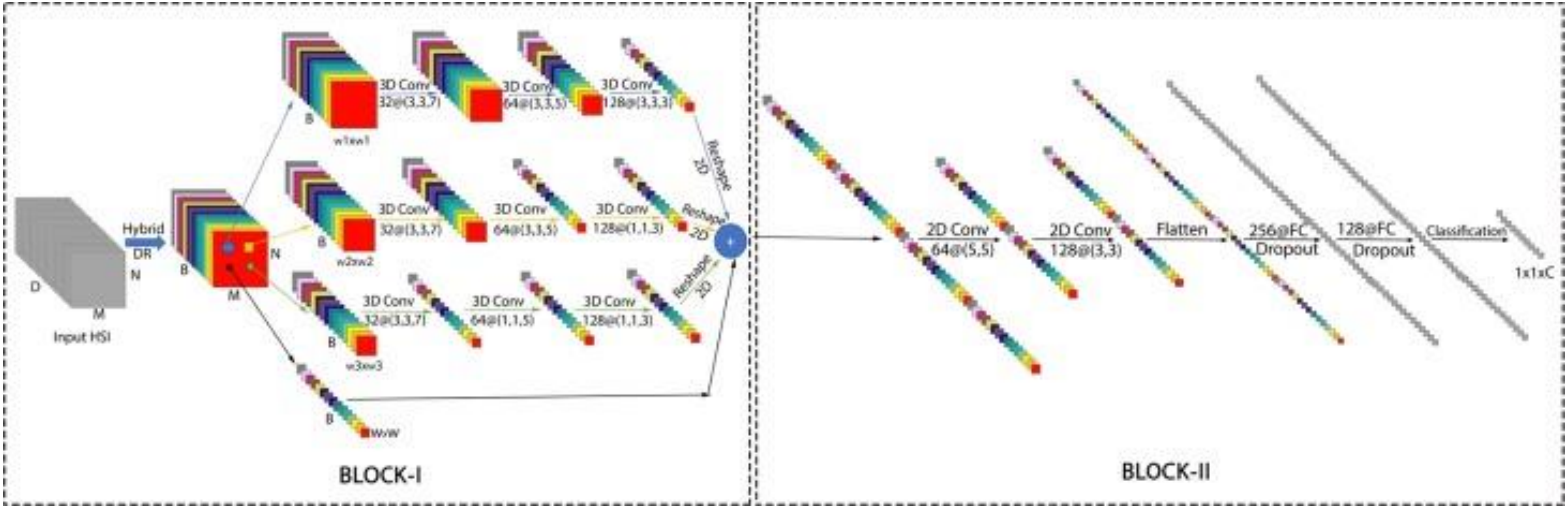


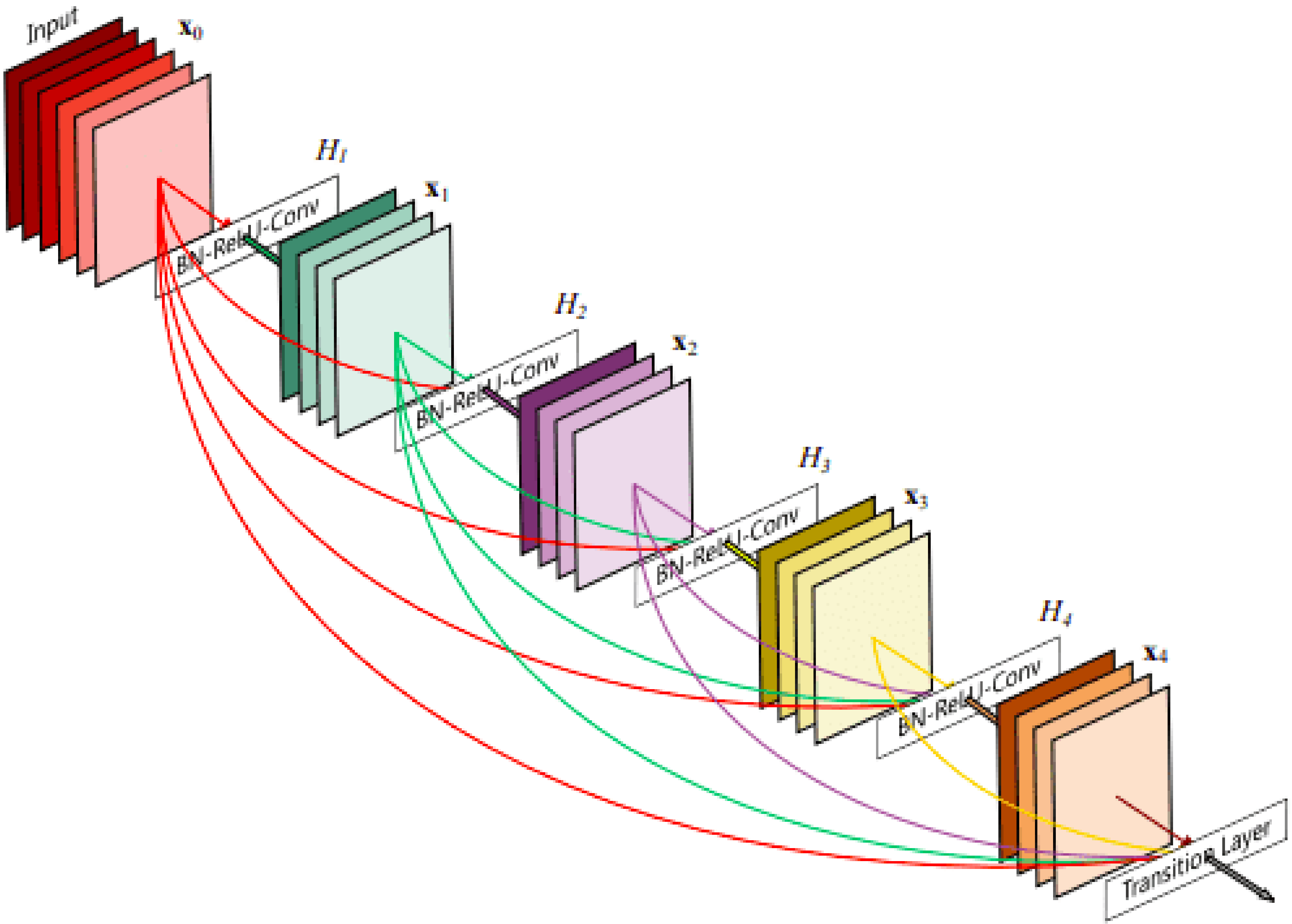
Fig. 1. Proposed HybridSpectralNet (HybridSN) model that integrates 3-D and 2-D convolutions for HSI classification.

Skip connections



Manel Baradad, Amaia Salvador, Xavier Giró-i-Nieto, Ferran Marqués (work under progress)


Input HSI
D
N
M
Hybrid
DR
B
N
M
B
w1xw1
3D Conv
32@(3,3,7)
3D Conv
64@(3,3,5)
3D Conv
128@(3,3,3)
Reshape
2D
B
w2xw2
3D Conv
32@(3,3,7)
3D Conv
64@(3,3,5)
3D Conv
128@(1,1,3)
Reshape
2D
B
w3xw3
3D Conv
32@(3,3,7)
3D Conv
64@(1,1,5)
3D Conv
128@(1,1,3)
Reshape
2D
B
WxW
+
BLOCK-I
2D Conv
64@(5,5)
2D Conv
128@(3,3)
Flatten
256@FC
Dropout
128@FC
Dropout
Classification
1x1xC
BLOCK-II

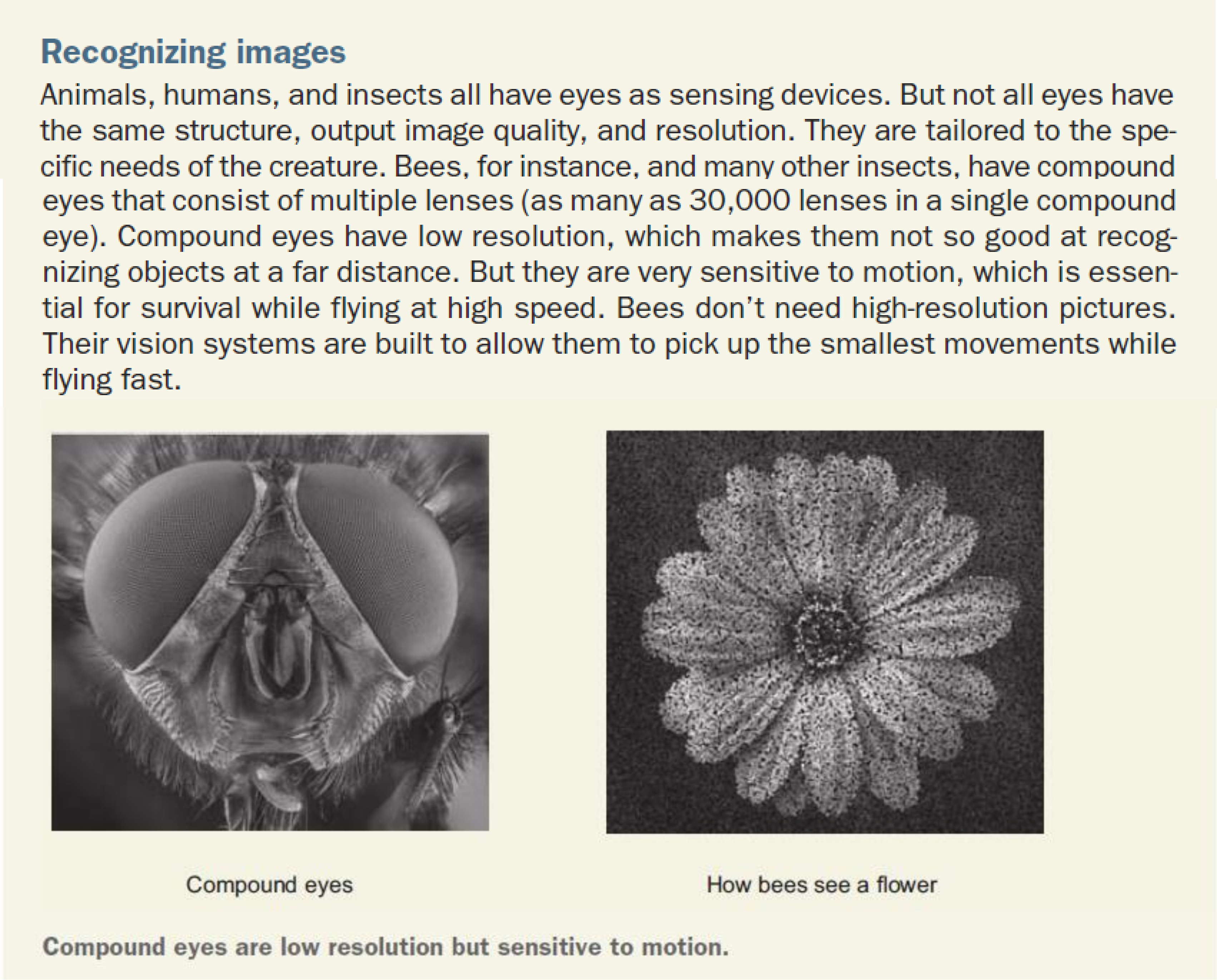

Input
$\mathbf{x}_0$
H_1
BN-ReLU-Conv
$\mathbf{x}_1$
H_2
BN-ReLU-Conv
$\mathbf{x}_2$
H_3
BN-ReLU-Conv
$\mathbf{x}_3$
H_4
BN-ReLU-Conv
$\mathbf{x}_4$
Transition Layer

QPCA

#	PCA-SVM	QPCA-SVM	PCA-CNN1d	QPCA-CNN1d	PCA-VGG-16	QPCA-VGG-16	PCA-Hybrid CNN	QPCA-Hybrid CNN	PCA-MLHM	QPCA-MLHM
KA (%)	80	73.91	80.39	78.33	83.77	94.68	98.96	99.25	99.39	99.41
OA (%)	82.50	77.16	82.84	81.03	85.87	95.34	99.09	99.34	99.47	99.47
AA (%)	80.28	70.63	79.48	78.24	66.58	81.89	98.17	98.71	99.01	99.36
Training Time (s)	0.41	0.67	21.87	36.04	53.44	50.95	65.60	60.40	41.67	25.47
Test Time (s)	2.40	2.45	0.25	0.27	2.20	2.24	1.93	1.83	4.53	3.64

https://www.mdpi.com/2072-4292/14/4/1038/htm

Time or accuracy



Recognizing images

Animals, humans, and insects all have eyes as sensing devices. But not all eyes have the same structure, output image quality, and resolution. They are tailored to the specific needs of the creature. Bees, for instance, and many other insects, have compound eyes that consist of multiple lenses (as many as 30,000 lenses in a single compound eye). Compound eyes have low resolution, which makes them not so good at recognizing objects at a far distance. But they are very sensitive to motion, which is essential for survival while flying at high speed. Bees don't need high-resolution pictures. Their vision systems are built to allow them to pick up the smallest movements while flying fast.

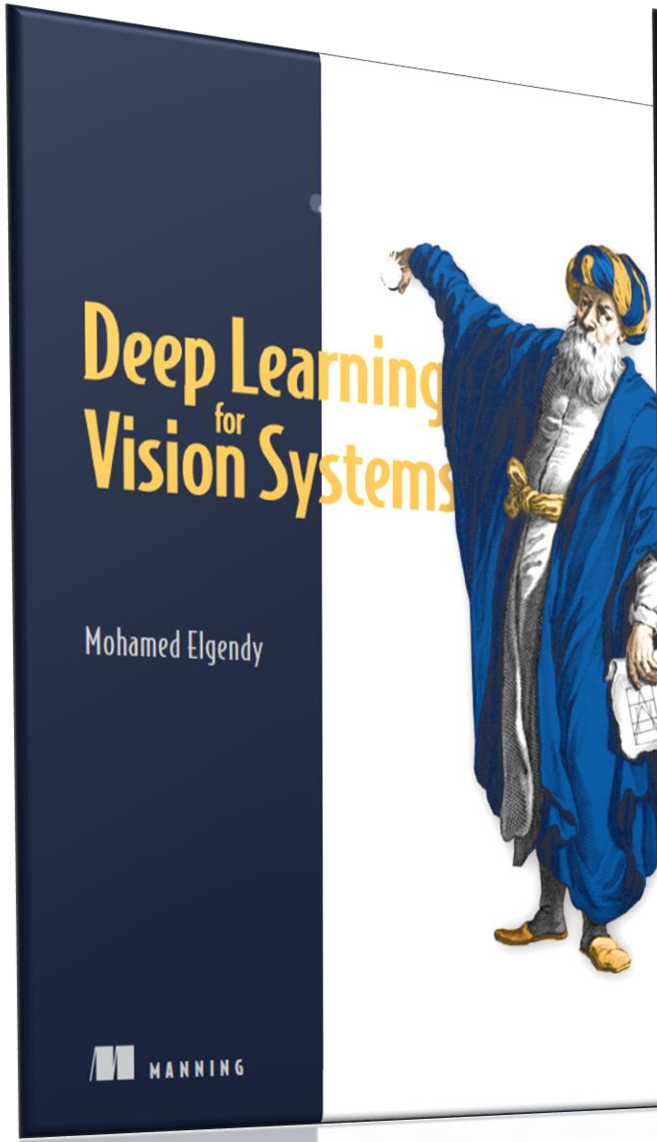
Compound eyes

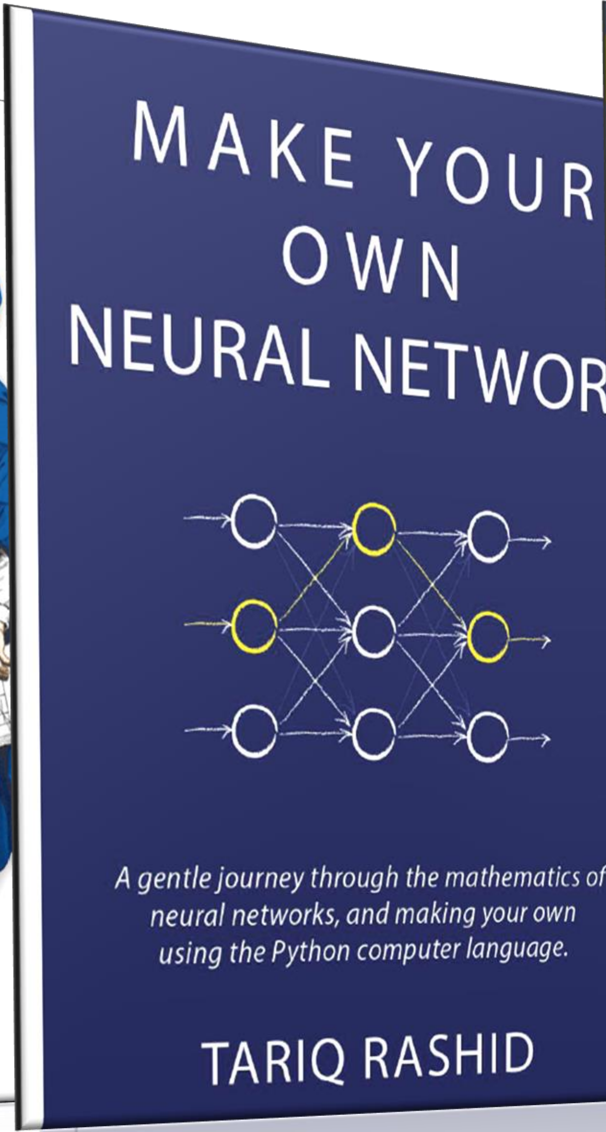
How bees see a flower

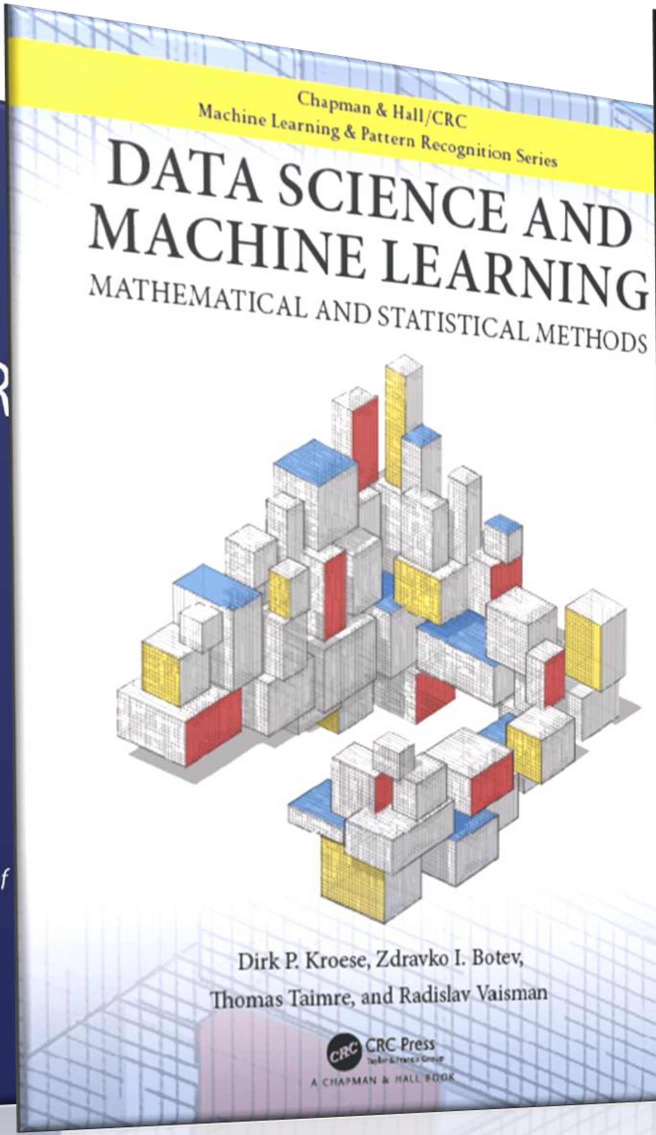
Compound eyes are low resolution but sensitive to motion.

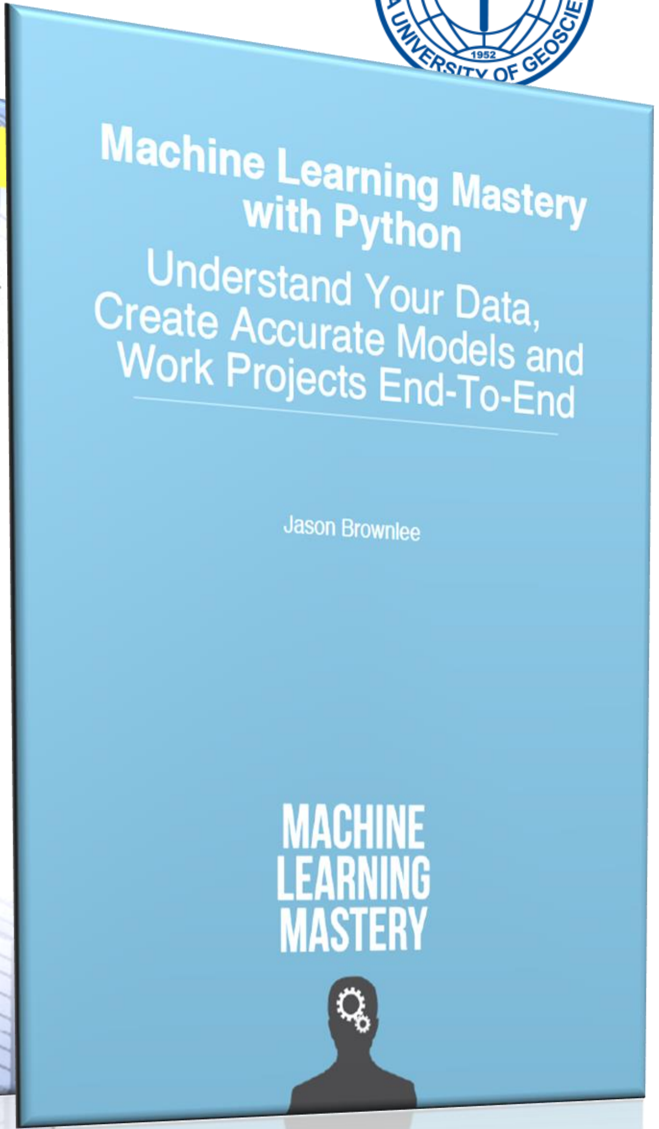
Useful Books

https://b-ok.cc/




MAKE YOUR OWN NEURAL NETWOR
A gentle journey through the mathematics of neural networks, and making your own using the Python computer language.
TARIQ RASHID


Chapman & Hall/CRC Machine Learning & Pattern Recognition Series
DATA SCIENCE AND MACHINE LEARNING
MATHEMATICAL AND STATISTICAL METHODS
Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, and Radislav Vaisman
CRC Press
A CHAPMAN & HALL BOOK


Machine Learning Mastery with Python
Understand Your Data, Create Accurate Models and Work Projects End-To-End
Jason Brownlee
MACHINE LEARNING MASTERY

Dalal AL-Alimi

https://www.researchgate.net/profile/Dalal-Al-Alimi

https://github.com/DalalAL-Alimi