

INTRODUCTION TO MACHINE LEARNING

**NATTHARIYA LAOPRACHA
COMPUTER SCIENCE
MAHASARAKHAM UNIVERSITY**

WHIS IS MACHINE LEARNING?

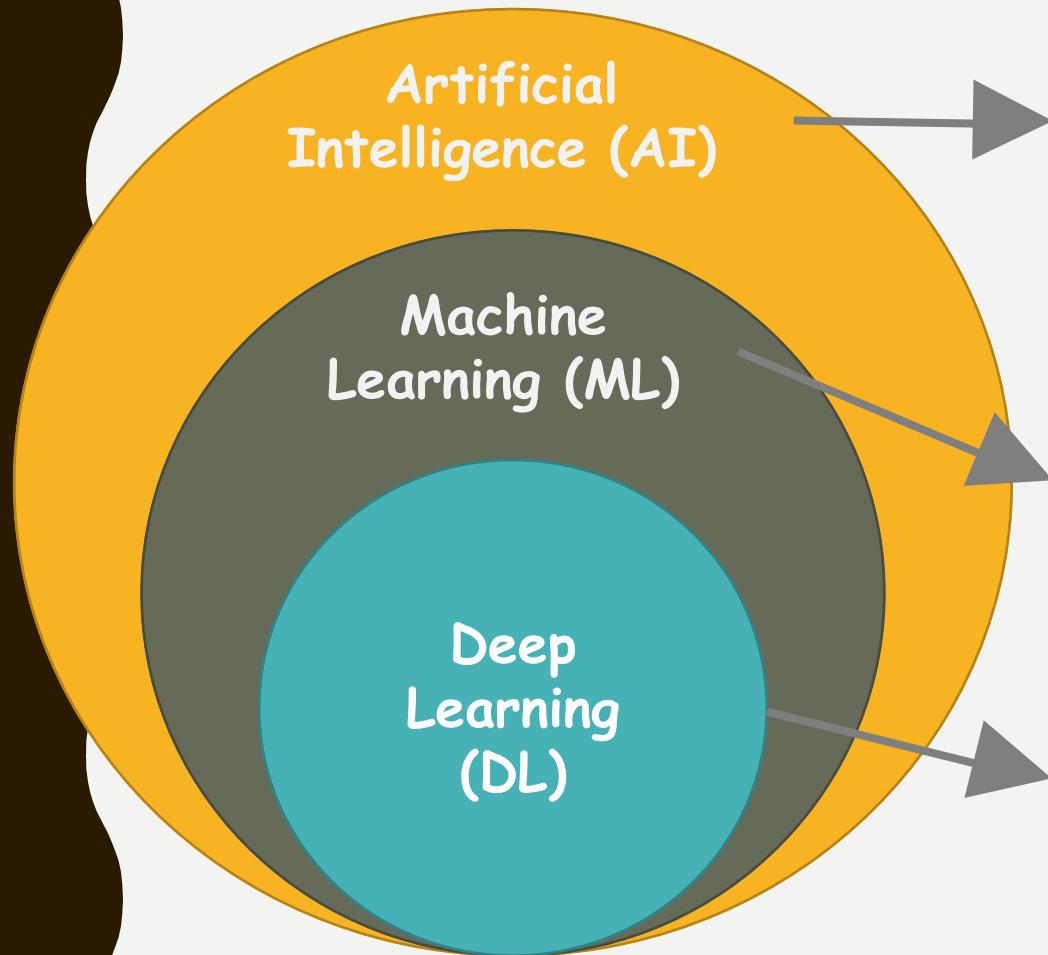
- Machine learning is a type of artificial intelligence (AI) that enables computers to learn and make decisions without explicitly being programmed.
- Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

THINK AND LEARN LIKE A BABY



Source. <https://www.slideshare.net/slideshow/introduction-to-machine-learning-259372973/259372973#2>

AI VS ML VS DL



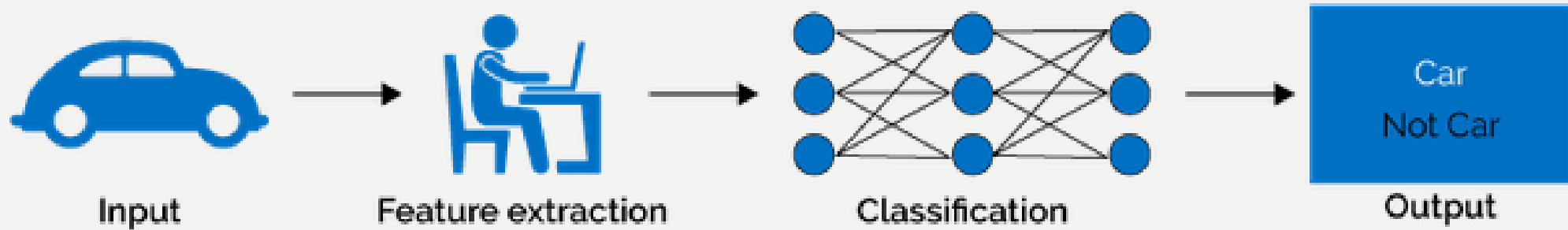
Artificial Intelligence is basically the study/process that enables machines to mimic human behavior through a particular algorithm

Machine Learning is the study that uses statistical methods to enable machines to improve with experience (automatically learning from data).

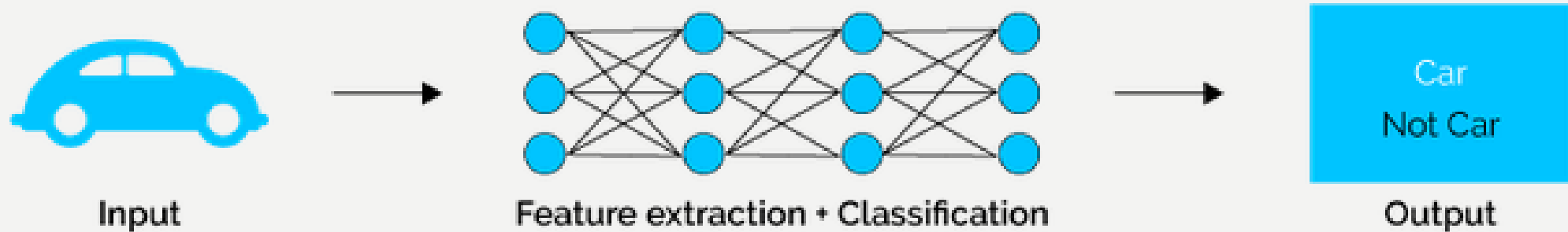
Deep Learning is the study that makes use of Neural Networks to imitate functionality just like a human brain.

ML VS DL

Machine Learning



Deep Learning

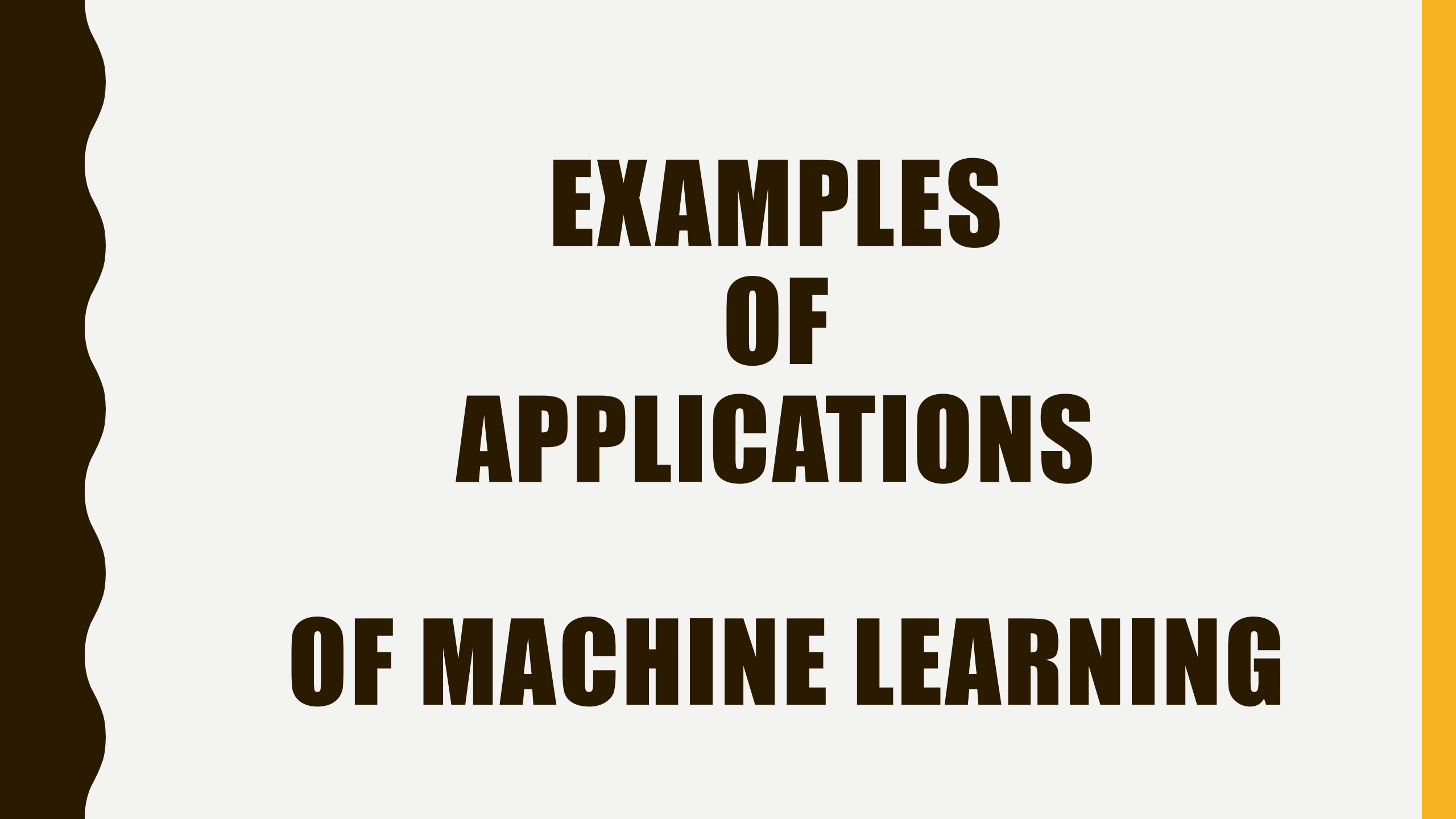


ADVANTAGES & DISADVANTAGES OF ML

Advantages



Source: [Advantages and Disadvantages of Machine Learning Language - DataFlair \(data-flair.training\)](https://data-flair.training)



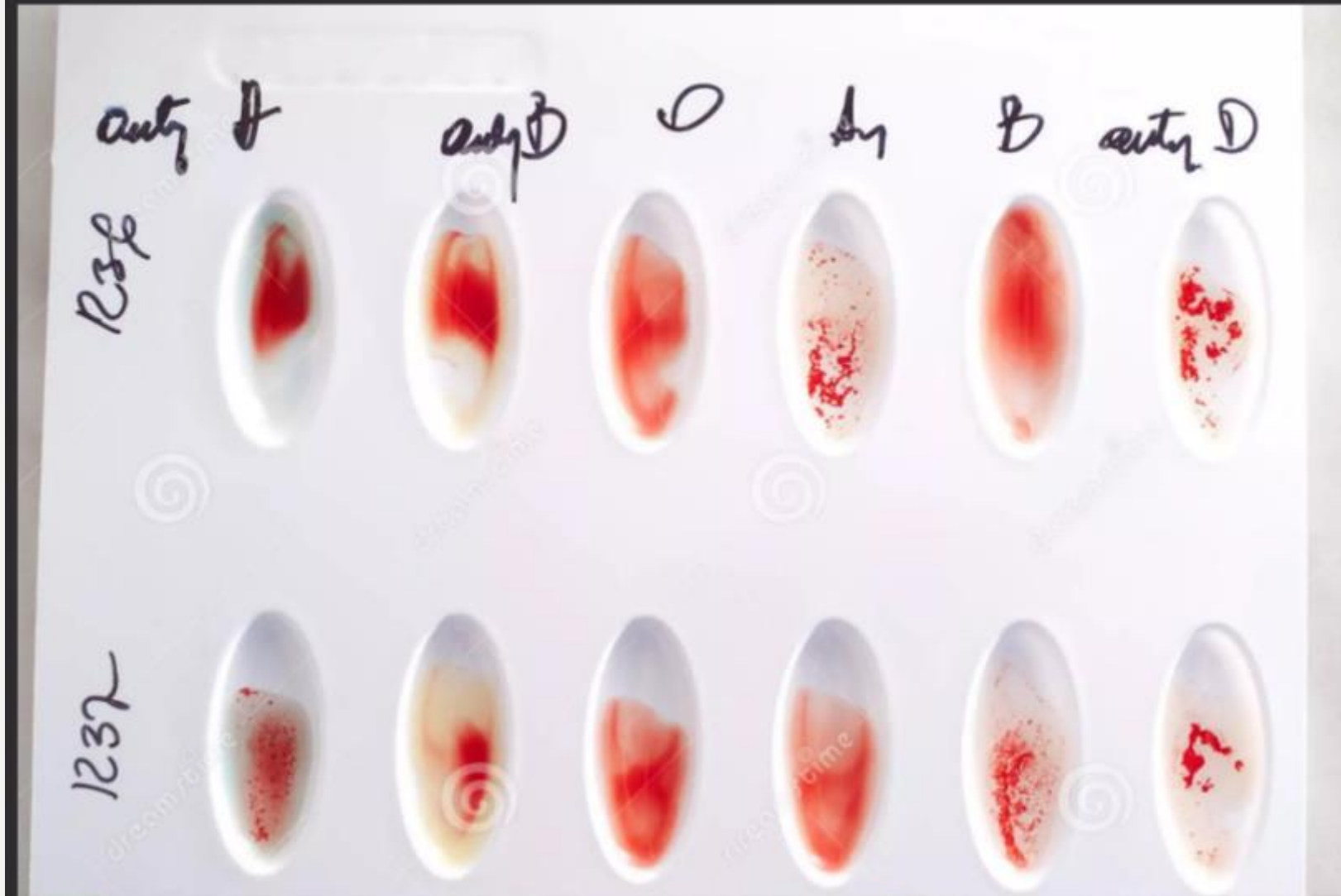
EXAMPLES OF APPLICATIONS OF MACHINE LEARNING

Handwriting Recognition / OCR



Source: <https://www.slideshare.net/slideshow/introduction-to-machine-learning-classifiers/65470558#5>

Blood Type Identification



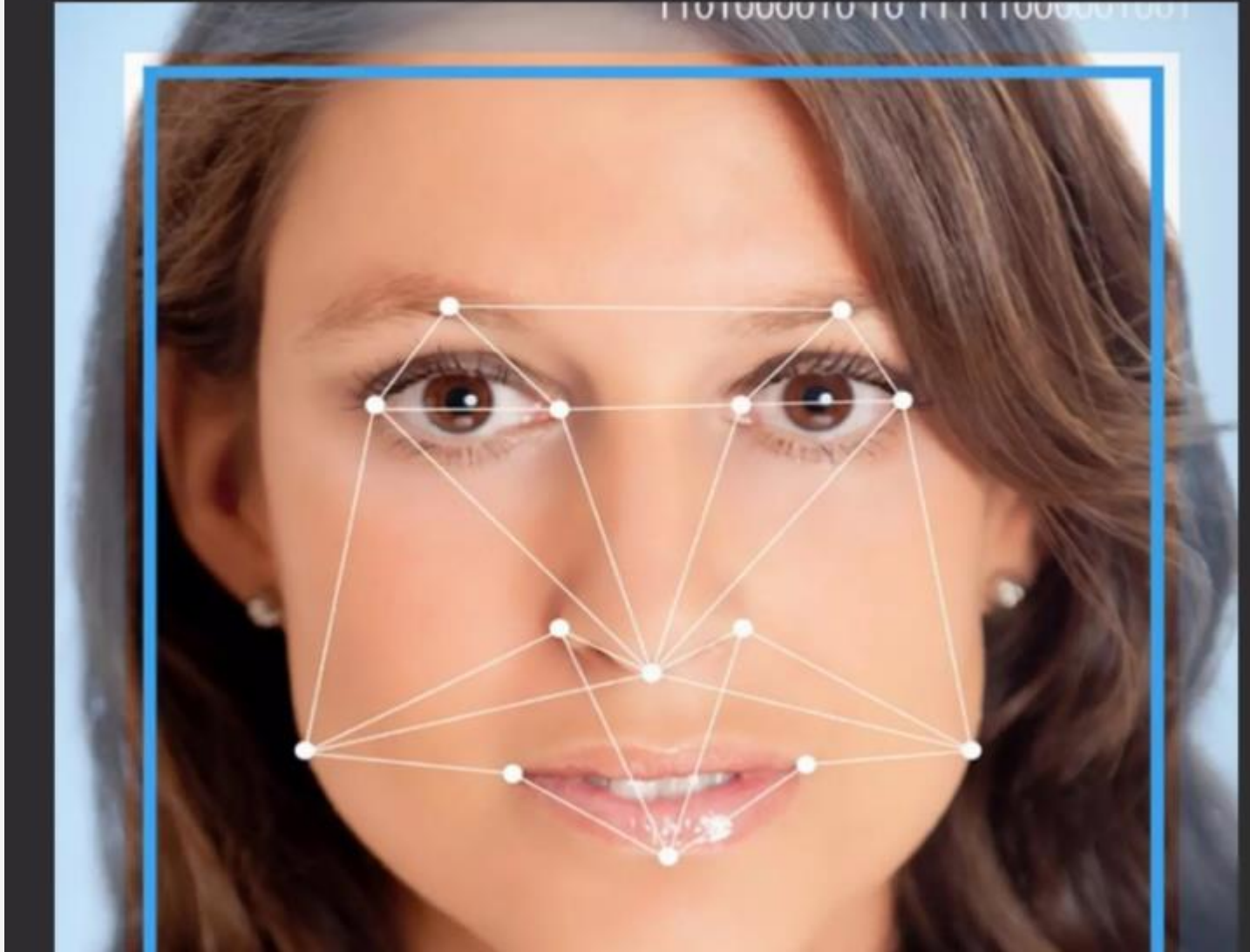
Source: <https://www.slideshare.net/slideshow/introduction-to-machine-learning-classifiers/65470558#5>

Automatic Document Classification



Source: <https://www.slideshare.net/slideshow/introduction-to-machine-learning-classifiers/65470558#5>

Face Recognition



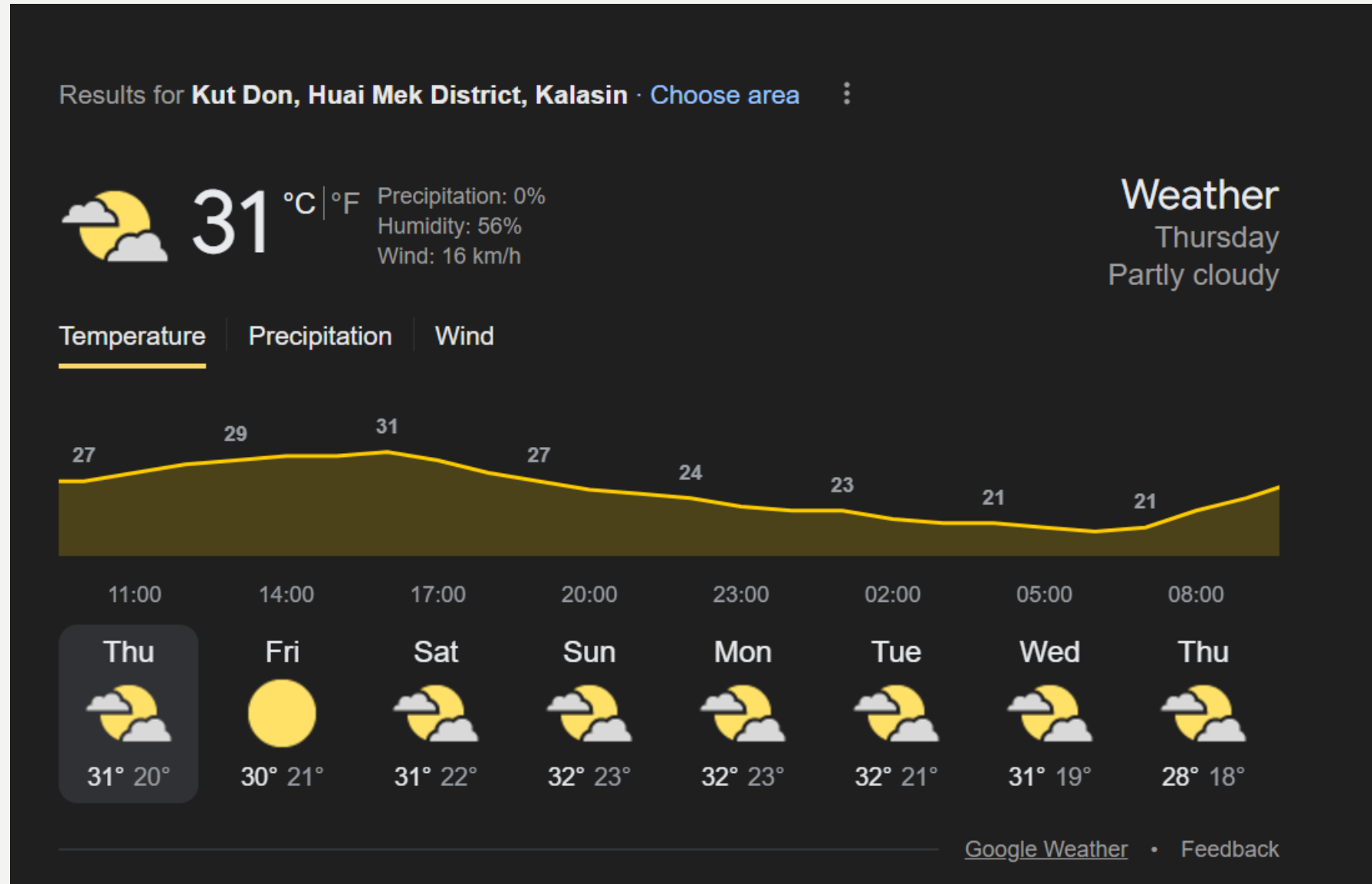
Source: <https://www.slideshare.net/slideshow/introduction-to-machine-learning-classifiers/65470558#5>

SHAZAM!!

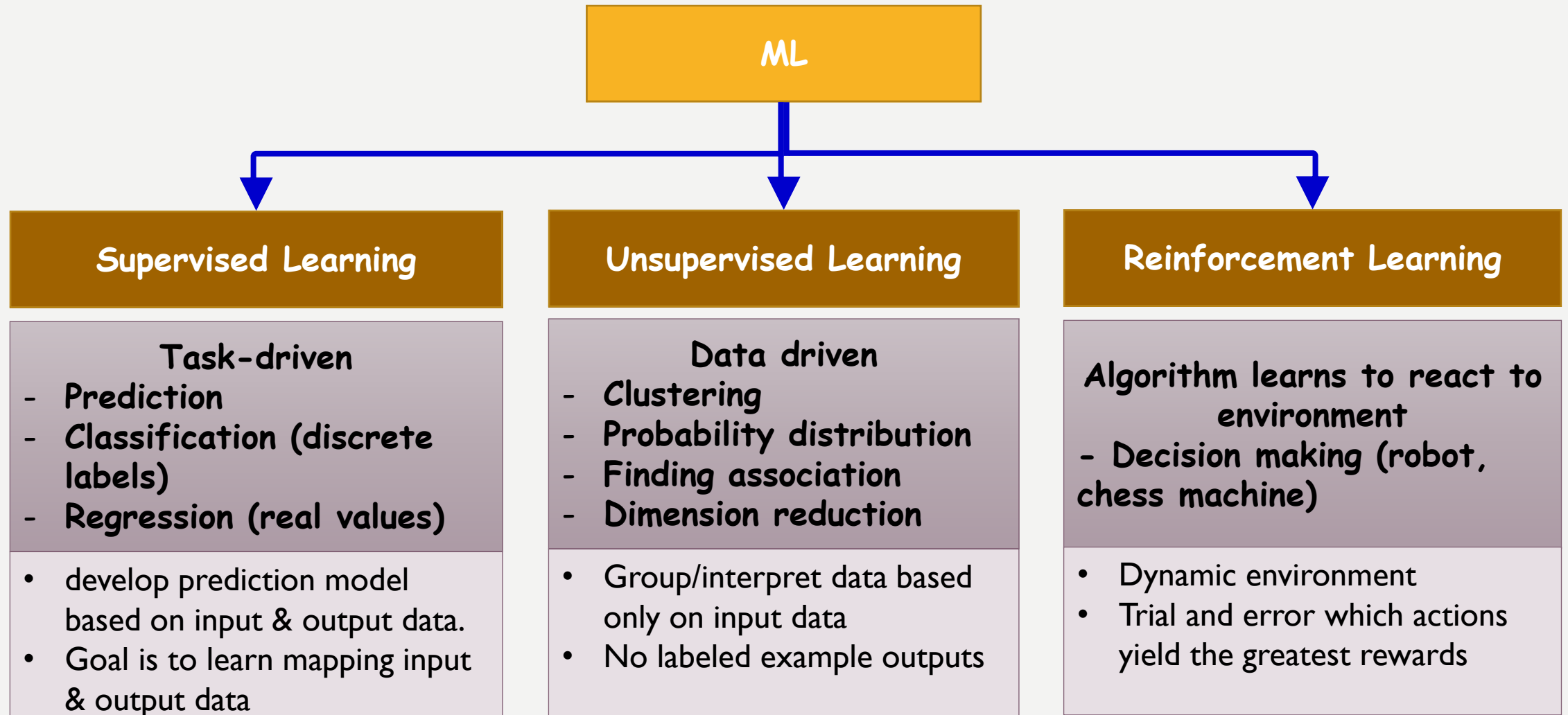


Source: <https://www.slideshare.net/slideshow/introduction-to-machine-learning-classifiers/65470558#5>

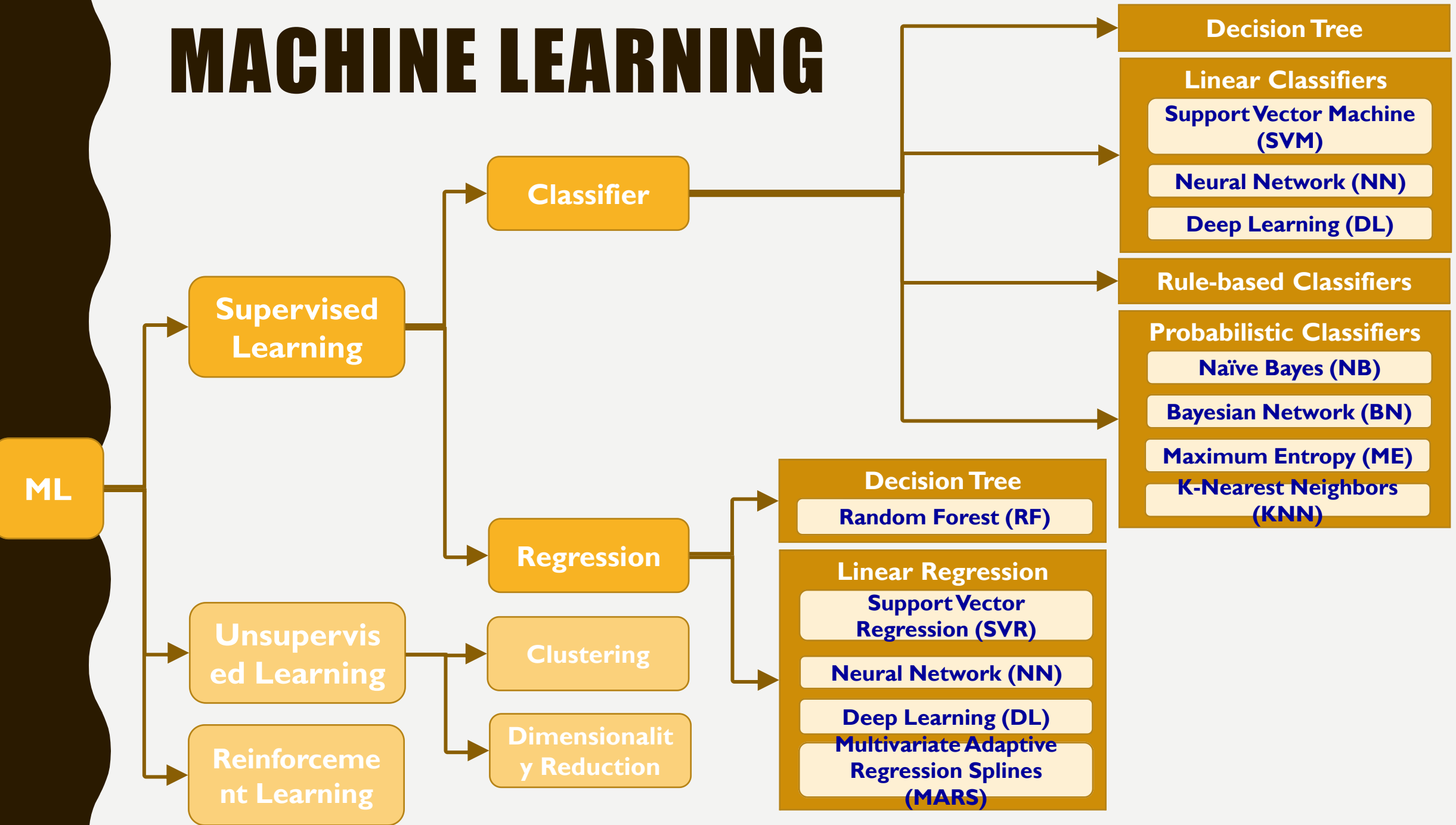
RAINFALL PREDICTION



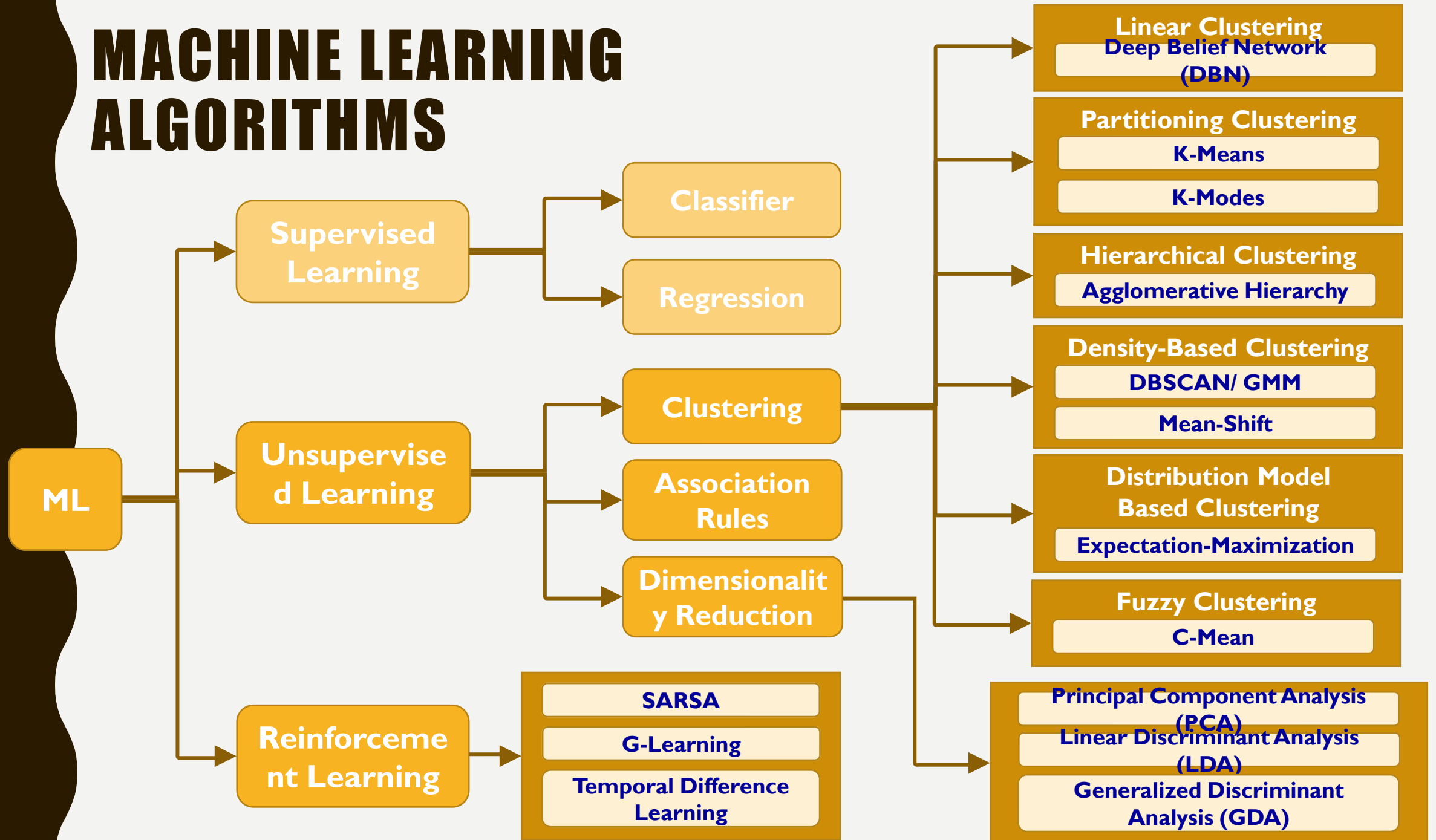
THE TYPES OF MACHINE LEARNING ALGORITHMS



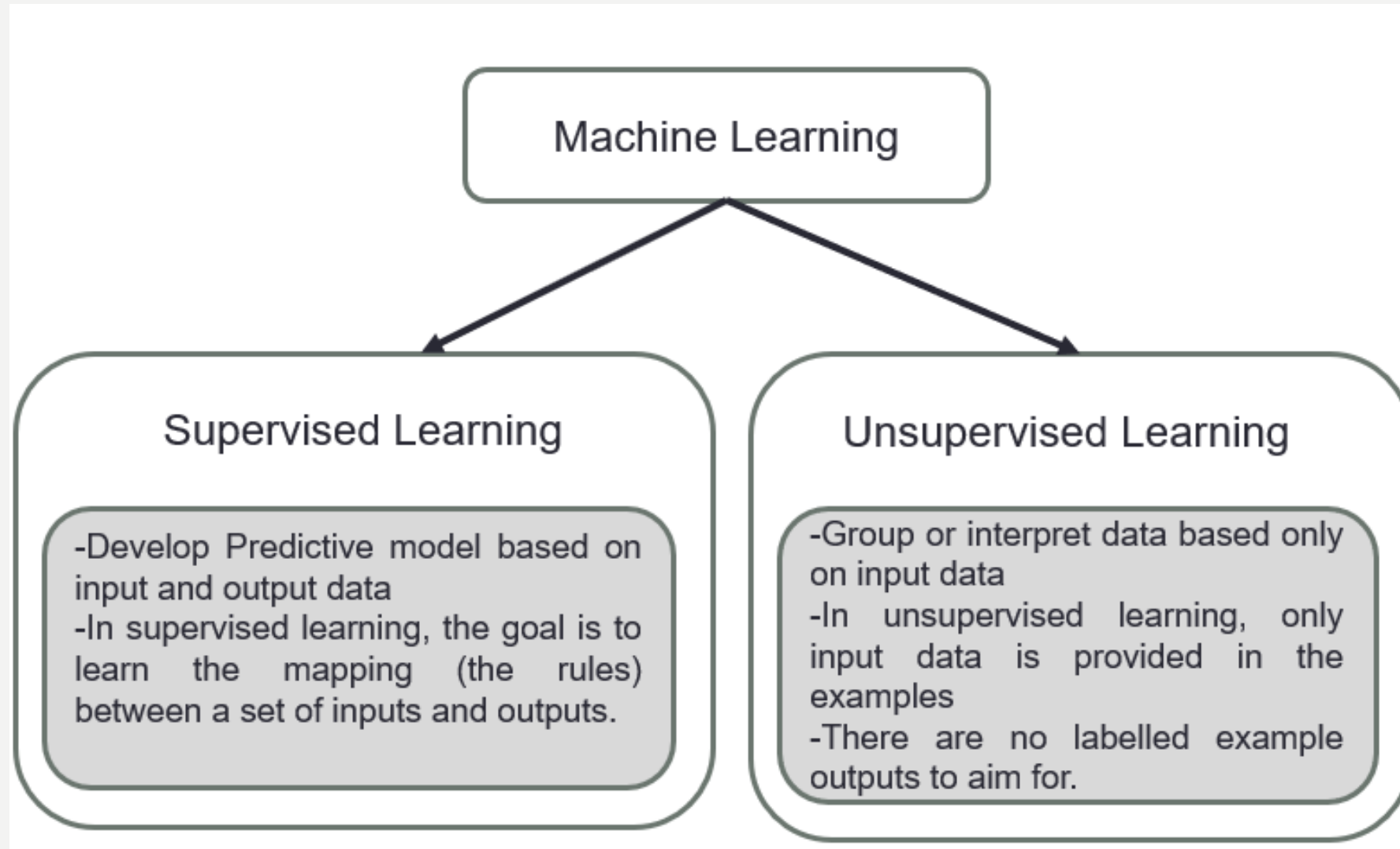
MACHINE LEARNING



MACHINE LEARNING ALGORITHMS



SUPERVISED VS UNSUPERVISED ML

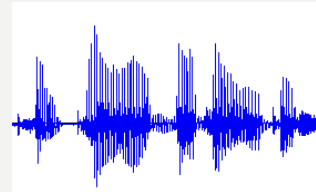


EXAMPLE (SUPERVISED LEARNING) TASKS

1. Use in a marketing company is trying to find out which customers are likely to churn.
2. Use it to predict the likelihood of the occurrence of perils like earthquakes, tornadoes, weather, etc.
3. Determine the Total Insurance Value.



Image recognition / classification



Speech recognition



Machine translation



Conversational agent / chatbot

EXAMPLE (UNSUPERVISED LEARNING) TASKS

- Association/ Recommendation
 - when a retailer wishes to find out what is the combination of products, customers tend to buy more frequently, such as in Amazon, Shopee, Lazada
 - in the pharmaceutical industry, unsupervised learning may be used to predict which diseases are likely to occur along with diabetes
 - Amazon recommends product
- Clustering
 - Grouping the documents or grouping customers
 - Image Segmentation

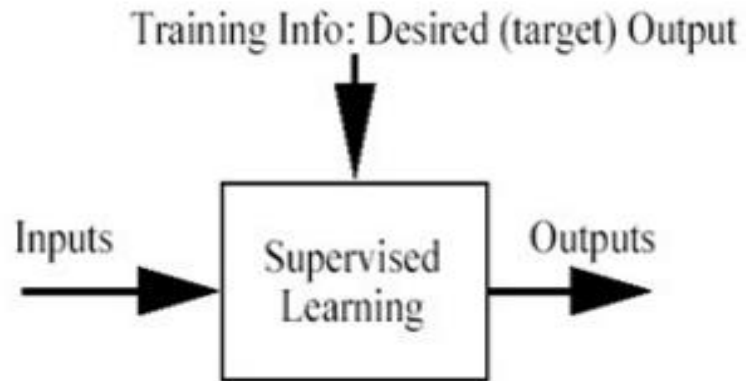


REINFORCEMENT LEARNING

- Algorithm discovers through **trial and error** which **actions yield the greatest rewards**.
- Three primary components:
 - the agent (the learner or decision maker),
 - the environment (everything the agent interacts with)
 - actions (what the agent can do).
- Objective: the agent chooses actions that maximize the expected reward over a given amount of time.

SUPERVISED VS REINFORCEMENT LEARNING

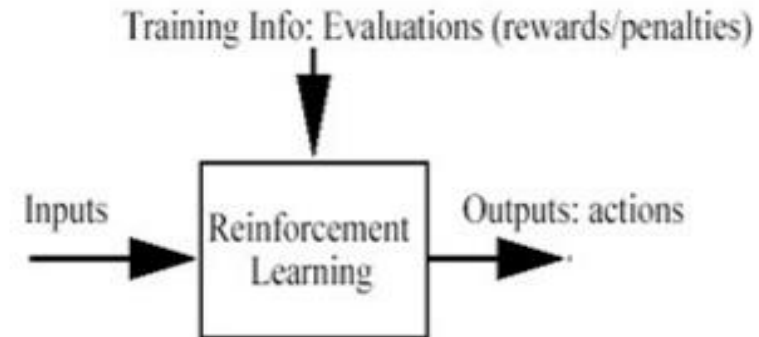
Supervised Learning



Error = (target output - actual output)

Input is an instance, output is a classification of the instance

Reinforcement Learning



Objective: Get as much reward as possible

Input is some "goal" output
Is a sequence of actions to meet the goal

EXAMPLE (REINFORCEMENT LEARNING) TASKS



Online Poker



Source:
RoboCup web site

May 11th, 1997
Computer won world champion of chess
(Deep Blue) (Garry Kasparov)

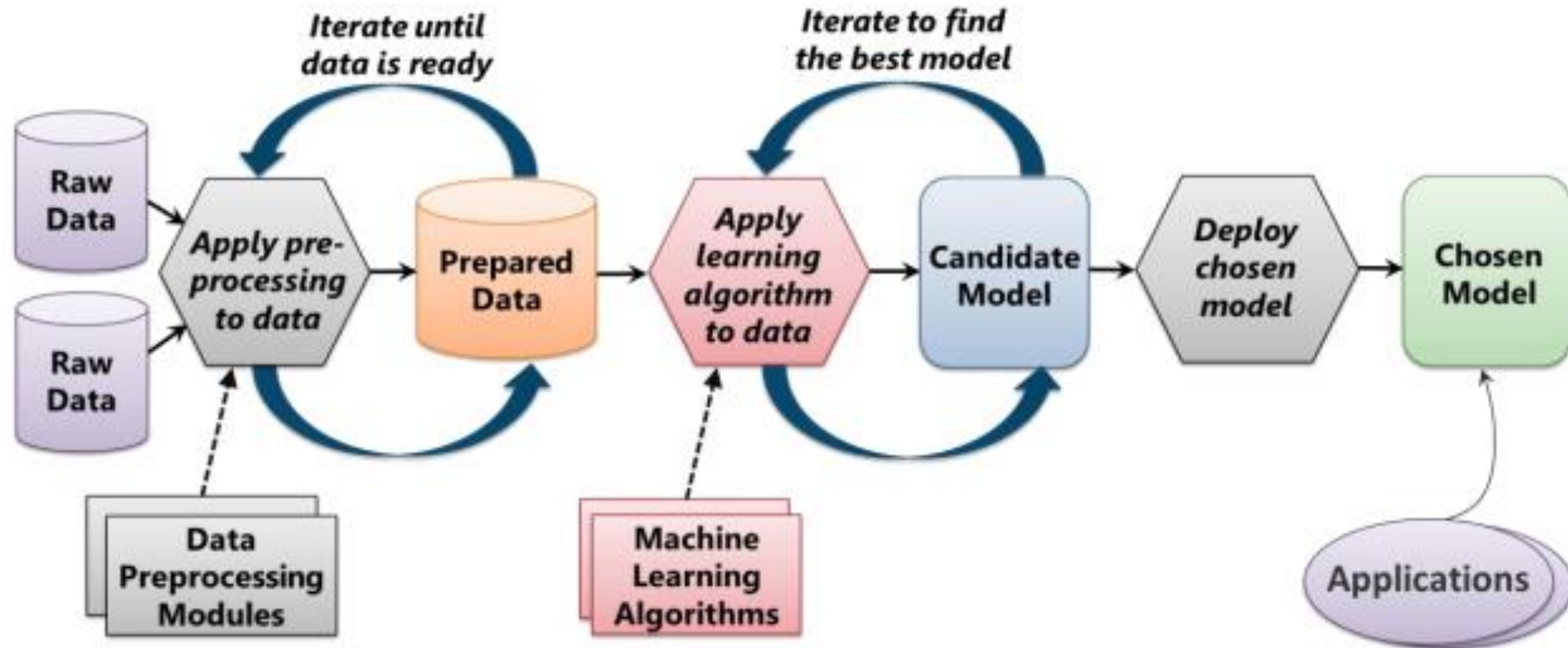


(Reuters = Kyodo News)

COMPARING TYPES OF ML

Supervised learning	Unsupervised learning	Reinforcement learning
Input/Output pairs	Input only	Input & Critic
Learning phase v.s. acting phase	Learning phase v.s. acting phase	Learning and acting simultaneously – online learning
		Explicitly explore the world and learn by trial and error

TYPICAL MACHINE LEARNING PROCESS



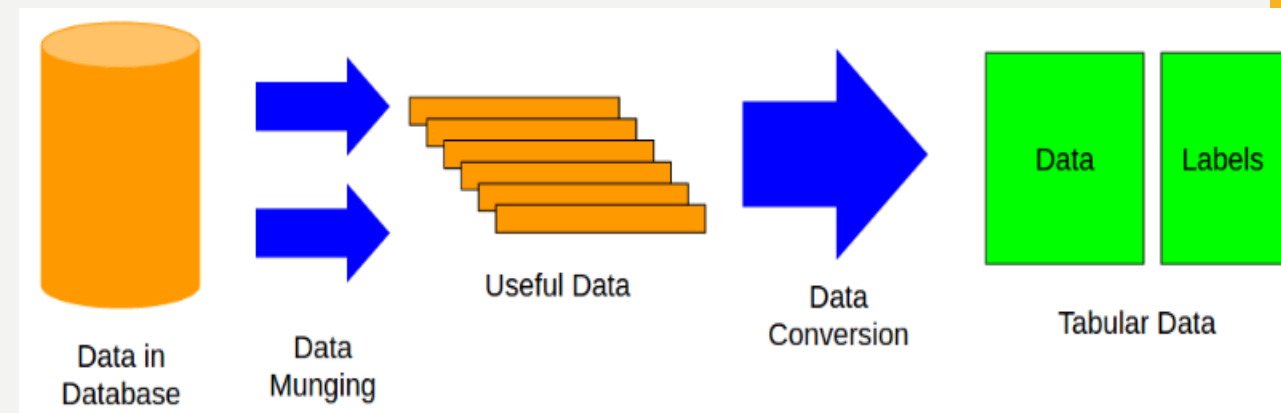
Typical Machine Learning Process (cont.)

- Data collection:

- Be it the raw data from Excel, access, text files, etc.
- This step (gathering past data) forms the foundation of future learning. The better the variety, density, and volume of relevant data, the better the learning prospects for the machine become.

Data preprocessing:

- Any analytical process thrives on the quality of the data used.
 - Selecting data
 - Cleaning data such as missing values or outliers
 - Reducing data



Typical Machine Learning Process (cont.)

- Training a model:

- Choosing the appropriate algorithm and representation of data in the form of the model.
- The cleaned data is split into two parts - train and test (proportion depending on the prerequisites);
 - the first part (training data) is used for developing the model.
 - The second part (test data), is used as a reference.

Evaluating the model:

- To test the accuracy, the second part of the data (holdout/ test data) is used.
- This step determines the precision in the choice of the algorithm based on the outcome.
- A better test to check the accuracy of a model is to see its performance on data that was not used at all during the model build.

Improving the performance:

- This step might involve choosing a different model altogether or introducing more variables to augment the efficiency.
- That's why a significant amount of time needs to be spent on data collection and preparation.

How to Separate Train/Test Dataset

- Hold-out method

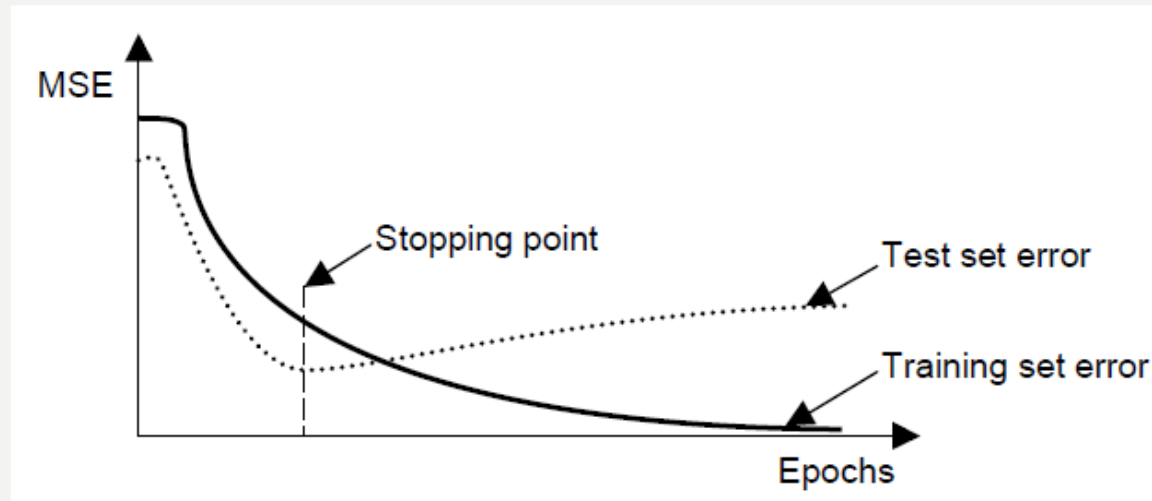
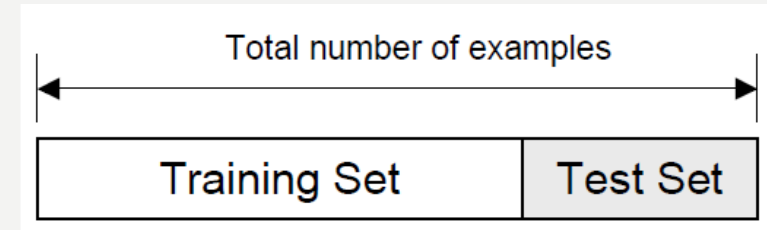
Cross validation

- Random Subsampling
- K-Fold Cross-Validation
- Leave-one-out Cross-Validation

Hold-out Method

- Split dataset into two groups

- Training set: used to train the classifier
- Test set: used to estimate the error rate of the trained classifier
- A typical application the holdout method is determining a stopping point for the back propagation error



The Holdout Method

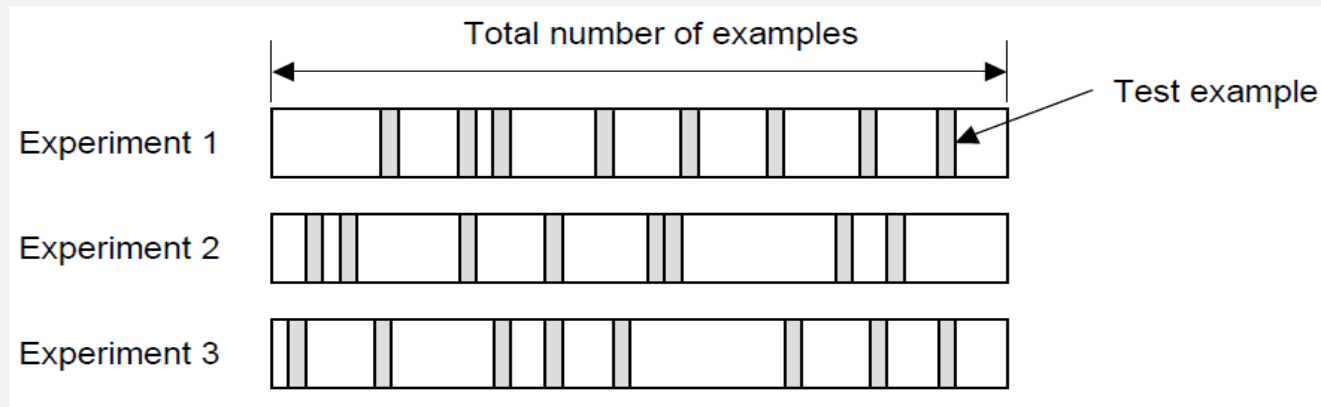
- Drawbacks

- In problems where we have a sparse dataset we may not be able to afford the “luxury” of setting aside a portion of the dataset for testing
- Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “unfortunate” split

The limitations of the hold-out can be overcome with a family of resampling methods at the expense of more computations

Random Subsampling

- Random Subsampling performs K data splits of the dataset
 - Each split randomly selects a (fixed) no. examples without replacement
 - For each data split we retrain the classifier from scratch with the training examples and estimate E_i with the test examples

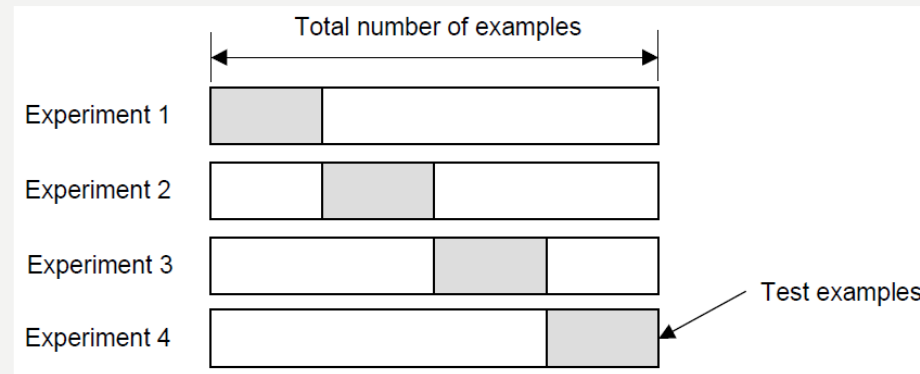


$$E = \frac{1}{K} \sum_{i=1}^K E_i$$

- The true error estimate is obtained as the average of the separate estimates E_i
 - This estimate is significantly better than the holdout estimate

K-Fold Cross-validation

- Create a K-fold partition of the dataset
 - For each of K experiments, use K-1 folds for training and the remaining one for testing



- K-Fold Cross validation is similar to Random Subsampling
 - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- As before, the true error is estimated as the average error rate

$$E = \frac{1}{K} \sum_{i=1}^K E_i$$

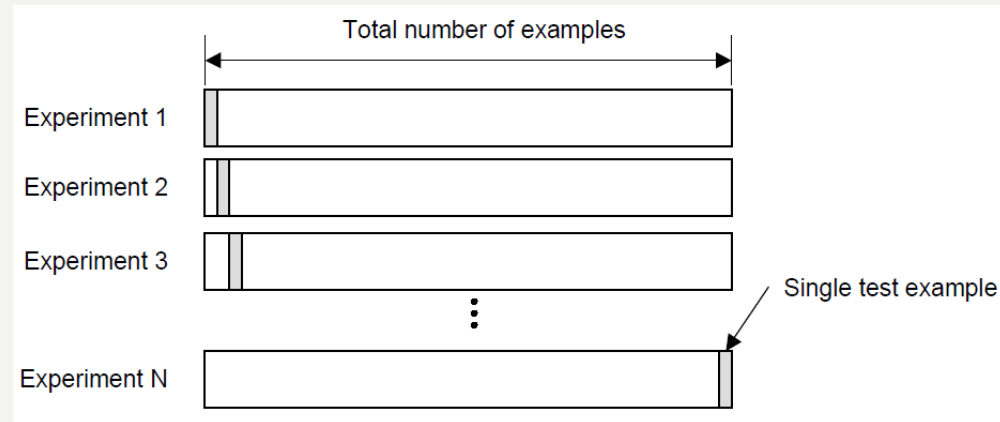
How many folds are needed?

- With a large number of folds
 - The bias of the true error rate estimator will be small (very accurate)
 - The variance of the true error rate estimator will be large
 - The computational time will be very large as well (many experiments)
- With a small number of folds
 - The computation time are reduced because of a few of experiments
 - The variance of the estimator will be small
 - The bias of the estimator will be large (higher than the true error rate)
- In practice, the choice of the number of folds depends on the size of the dataset
 - For large datasets, even 3-Fold Cross Validation will be quite accurate
 - For very sparse datasets, we may have to use leave-one-out in order to train on as many examples as possible

A common choice for K-Fold Cross Validation is $K=10$

Leave-one-out Cross Validation

- Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples
 - For a dataset with N examples, perform N experiments
 - For each experiment use N-1 examples for training and the remaining example for testing



As usual, the true error is estimated as the average error rate on test examples

$$E = \frac{1}{N} \sum_{i=1}^N E_i$$

Suit for small size of sample

DEVELOPING MODEL

- What is the problem being solved?
- What is the goal of the model?
 - Minimize error on the “training” data
 - Training data is the data used to train the model (all of it but the part we removed)

DEVELOPING MODEL



Evaluation Machine learning

1. Confusion matrix

2. Accuracy

3. Precision

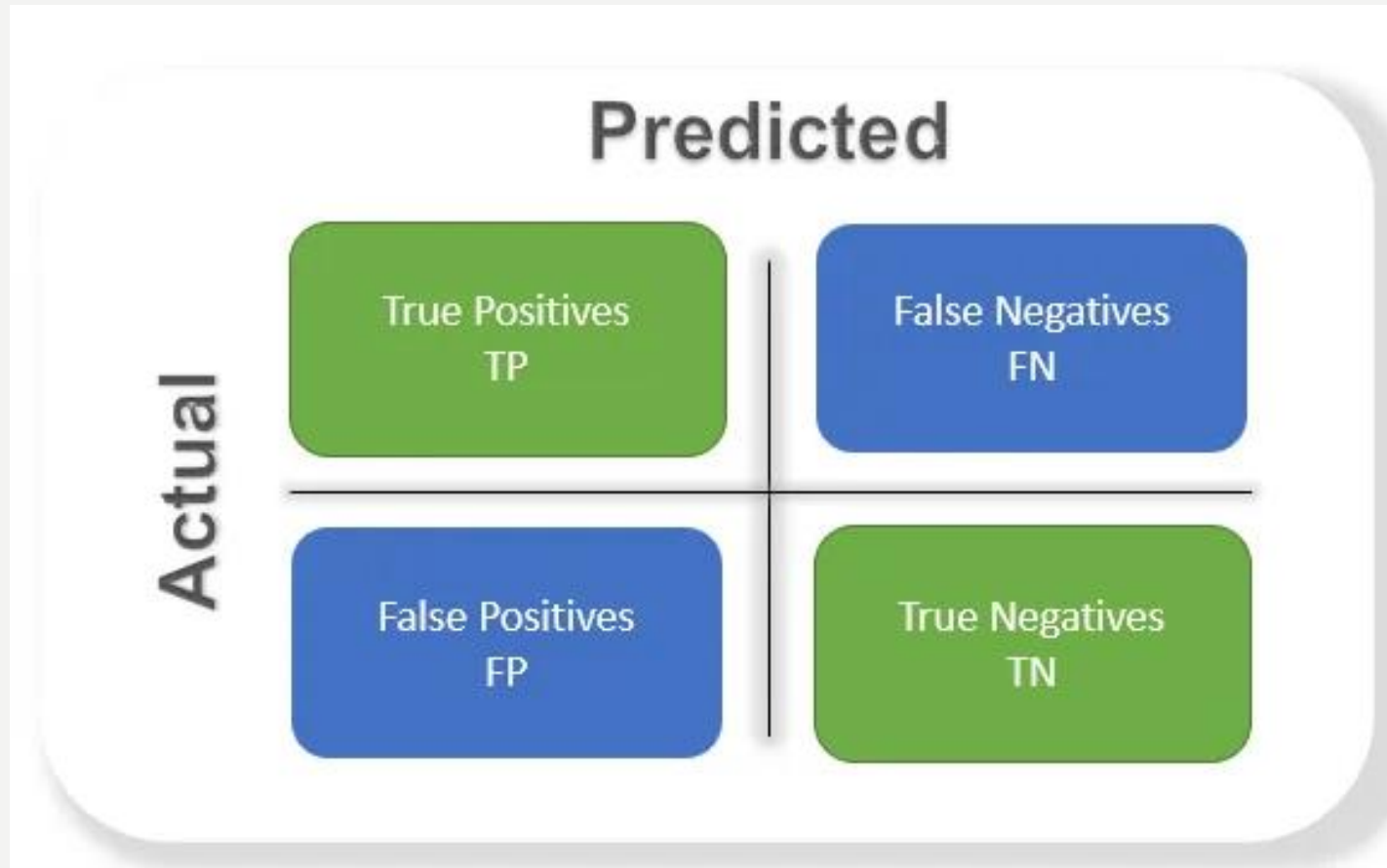
4. Recall

5. Specificity

6. F1 score

7. ROC (Receiver Operating Characteristics) curve

Confusion matrix



Accuracy

- The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Precision

- Percentage of positive instances out of the ***total predicted positive*** instances. Here denominator is the model prediction done as positive from the whole given dataset. Take it as to find out '*how much the model is right when it says it is right*'.

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate

- Percentage of positive instances out of the ***total actual positive*** instances. Therefore denominator ($TP + FN$) here is the *actual* number of positive instances present in the dataset. Take it as to find out '*how much extra right ones, the model missed when it showed the right ones*'.

$$\frac{TP}{TP + FN}$$

Specificity

- Percentage of negative instances out of the ***total actual negative*** instances. Therefore denominator ($TN + FP$) here is the *actual* number of negative instances present in the dataset. It is similar to recall but the shift is on the negative instances. *Like finding out how many healthy patients were not having cancer and were told they don't have cancer.* Kind of a measure to see how separate the classes are.

$$\frac{TN}{TN + FP}$$

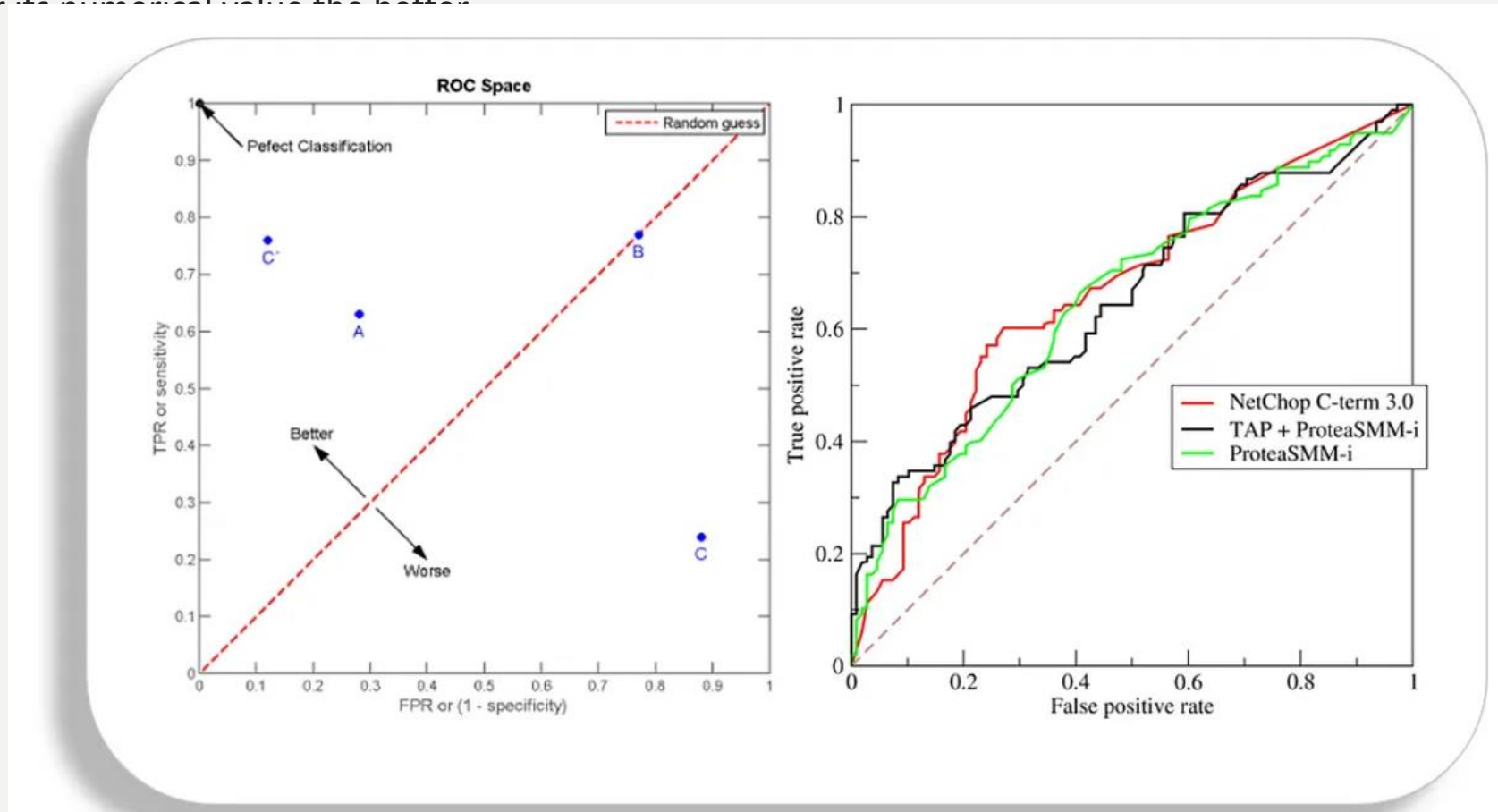
F1 score

- It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better. See that due to the product in the numerator if one goes low, the final F1 score goes down significantly. So a model does well in F1 score if the positive predicted are actually positives (precision) and doesn't miss out on positives and predicts them negative (recall).

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

ROC curve

- ROC stands for receiver operating characteristic and the graph is plotted against TPR and FPR for various threshold values. As TPR increases FPR also increases. As you can see in the first figure, we have four categories and we want the threshold value that leads us closer to the top left corner. Comparing different predictors (here 3) on a given dataset also becomes easy as you can see in figure 2, one can choose the threshold according to the application at hand. ROC AUC is just the area under the curve, the higher its numerical value the better.





Python in Google Colab

Import data

```
1 import pandas as pd  
2 import numpy as np  
3 data=pd.read_csv('/content/Iris.csv')
```

Preparing Data by Holdout Method

```
1 import numpy as np
2 from sklearn.model_selection import train_test_split
3 import pandas as pd
4 data=pd.read_csv('/content/Iris.csv')
5 Xdata=np.array(data)
6 X=Xdata[:,1:5]
7 y=Xdata[:,5]
8 #Holdout method 60:40
9 X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.6, test_size=0.4)
10 print(X_train)
11 print(y_train)
```

Preparing Data K-Fold Cross-Validation

```
1 from sklearn.model_selection import KFold
2 import pandas as pd
3 import numpy as np
4 data=pd.read_csv('/content/Iris.csv')
5 Xdata=np.array(data)
6 X=Xdata[:,1:5]
7 y=Xdata[:,5]
8 kf = KFold(n_splits=10, shuffle=True, random_state=150)
9
10 for train_index, test_index in kf.split(X,y):
11     print (train_index, test_index)
12     print(Xdata[train_index])
13     print(Xdata[test_index])
```