# DATA SCIENCE

SIKKIM UNIVERSITY
MCA 3$^{RD}$ SEMESTER

-PRATIKSHYA SHARMA

# CONTENT

✓DEFINITION-CLASSIFICATION

✓MACHINE LEARNING MODEL

✓CLASSIFICATION & PREDICTION

✓DATA SEPARABILITY

✓DECISION BOUNDARY

✓VALIDATION METHODS

✓CROSS VALIDATION

✓BOOTSTRAPPING

✓ASSESSMENT METRICS

✓BAYESIAN CLASSIFICATION – NAÏVE BAYES CLASSIFICATION

# CLASSIFICATION

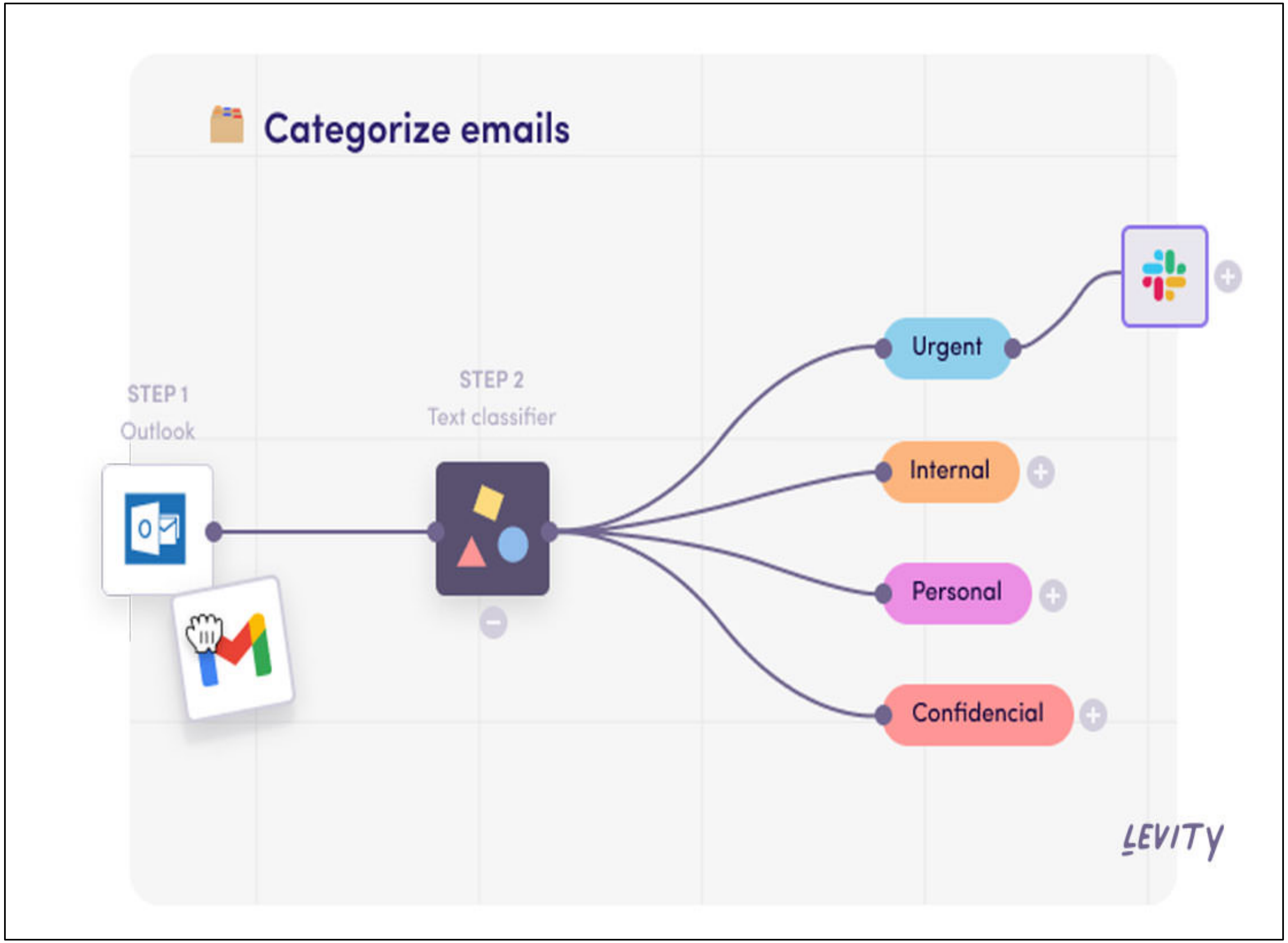
Definition

- Classification is a technique in data science used by data scientists to categorize data into a given number of classes.
- This technique can be performed on structured or unstructured data and its main goal is to identify the category or class to which a new data will fall under.
- Classes are sometimes called as targets/ labels or categories.
- In Classification, a model learns from the given dataset or observations and then classifies new observation into a number of classes or groups.
- Such as,Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc

**Example:**

Gmail classifies mails in more than one class like social, promotions, updates, and forums.

# CLASSIFICATION EXAMPLE



# CLASSIFIER

- The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

- **Binary Classifier:** If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
  **Examples:** YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

- **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
  **Example:** Classifications of types of crops, Classification of types of music.

# MACHINE LEARNING MODEL

# MACHINE LEARNING MODEL

- Machine Learning models can be understood as a program that has been trained to find patterns within new data and make predictions.
- These models are represented as a mathematical function that takes requests in the form of input data, makes predictions on input data, and then provides an output in response.
- First, these models are trained over a set of data, and then they are provided an algorithm to reason over data, extract the pattern from feed data and learn from those data.
- Once these models get trained, they can be used to predict the unseen dataset.
- In image recognition, a machine learning model can be taught to recognize objects - such as cars or dogs.
- A machine learning model can perform such tasks by having it 'trained' with a large dataset.
- During training, the machine learning algorithm is optimized to find certain patterns or outputs from the dataset, depending on the task.
- The output of this process - often a computer program with specific rules and data structures - is called a machine learning model.

# MACHINE LEARNING MODEL

Based on different business goals and data sets, there are three learning models for algorithms. Each machine learning algorithm settles into one of the three models:

1. **Supervised Learning**

*It* involves feedback to indicate when a prediction is right or wrong,

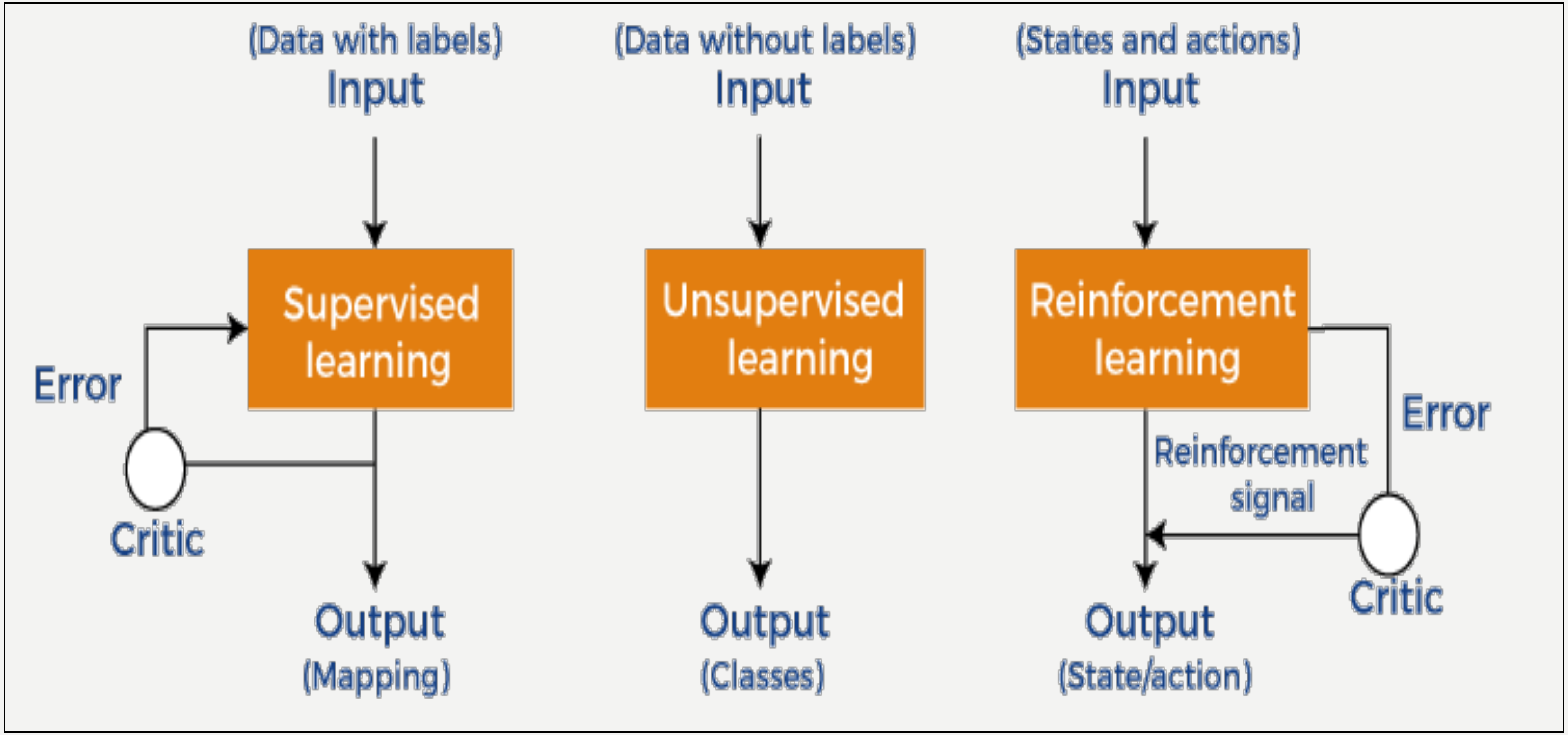
2. **Unsupervised Learning**

- *It* involves no response, the algorithm simply tries to categorize data based on its hidden structure

3. **Reinforcement Learning**

- It is similar to supervised learning in that it receives feedback, but it's not necessarily for each input or state.

# MACHINE LEARNING MODEL



# SUPERVISED LEARNING

- In supervised learning, a data set includes its desired outputs (or *labels*) such that a function can calculate an error for a given prediction.
- The supervision comes when a prediction is made and an error produced (actual vs. desired) to alter the function and learn the mapping.
- Supervised learning is the simplest of the learning models to understand.
- Learning in the supervised model entails creating a function that can be trained by using a training data set, then applied to unseen data to meet some predictive performance.
- The goal is to build the function so that it generalizes well over data it has never seen.

# SUPERVISED LEARNING

- We build and test a mapping function with supervised learning in two phases.
- In the **first phase,** we segment a data set into two types of samples: training data and test data.
- Both training and test data contain a test vector (the inputs) and one or more known desired output values.
- We train the mapping function with the training data set until it meets some level of performance (a metric for how accurately the mapping function maps the training data to the associated desired output).
- In the context of supervised learning, this occurs with each training sample, where you use the error (actual vs. desired output) to alter the mapping function.
- In the **next phase**, you test the trained mapping function against the test data.
- The test data represents data that has not been used for training and provides a good measure for how well the mapping function generalizes to unseen data.

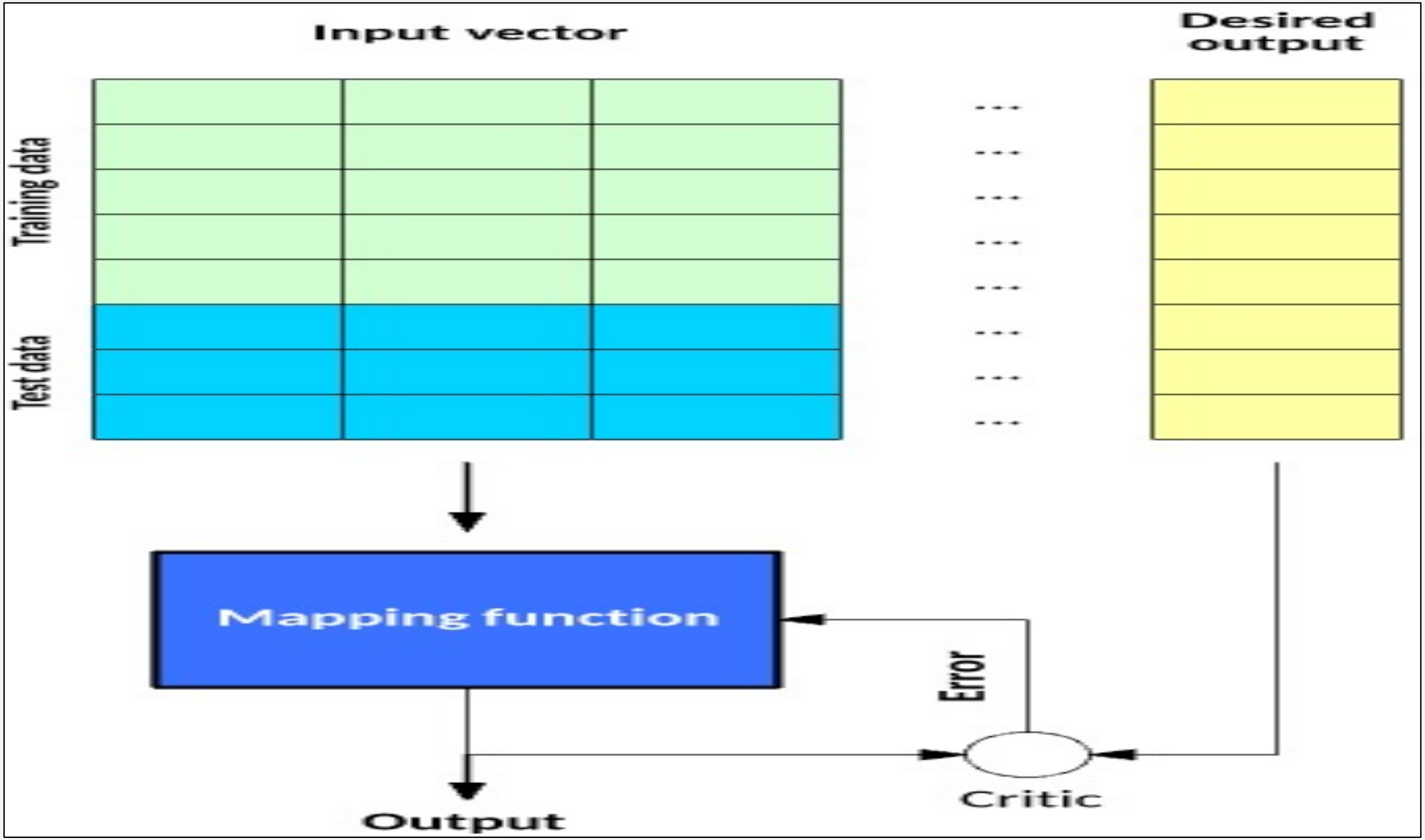


**Figure: The two phases of building and testing a mapping function with supervised learning**

# SUPERVISED LEARNING

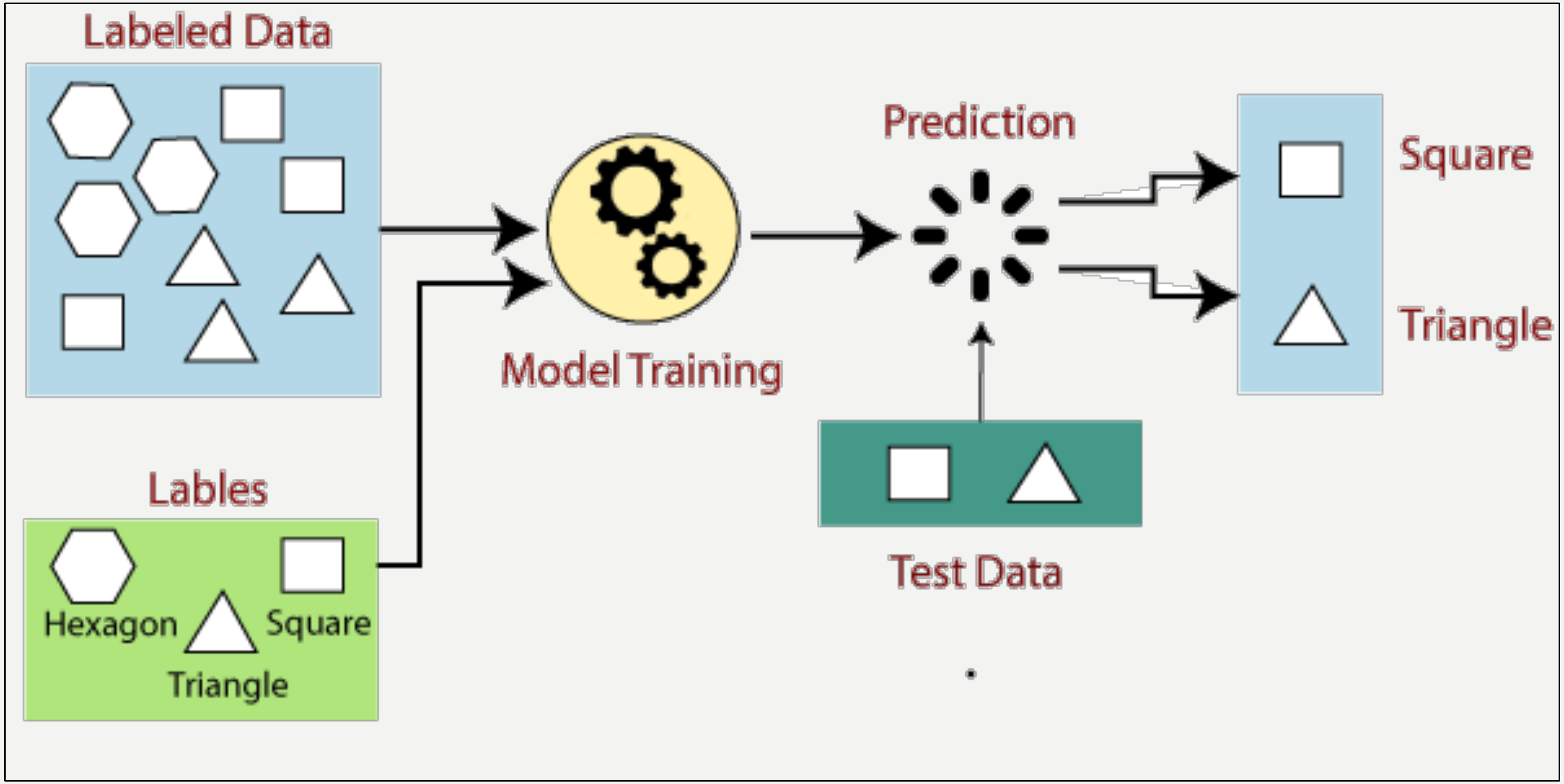
- Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output.
- The labelled data means some input data is already tagged with the correct output.
- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly.
- It applies the same concept as a student learns in the supervision of the teacher.
- Supervised learning is a process of providing input data as well as correct output data to the machine learning model.
- The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.
- In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

# HOW SUPERVISED LEARNING WORKS?

- In supervised learning, models are trained using labelled dataset, where the model learns about each type of data.

- Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

# SUPERVISED LEARNING -EXAMPLE



# SUPERVISED LEARNING

- Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.
- If the given shape has four sides, and all the sides are equal, then it will be labelled as a **Square**.
- If the given shape has three sides, then it will be labelled as a **triangle**.
- If the given shape has six equal sides then it will be labelled as **hexagon**.
- Now, after training, we test our model using the test set, and the task of the model is to identify the shape.
- The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

# SUPERVISED LEARNING

**Steps Involved in Supervised Learning:**

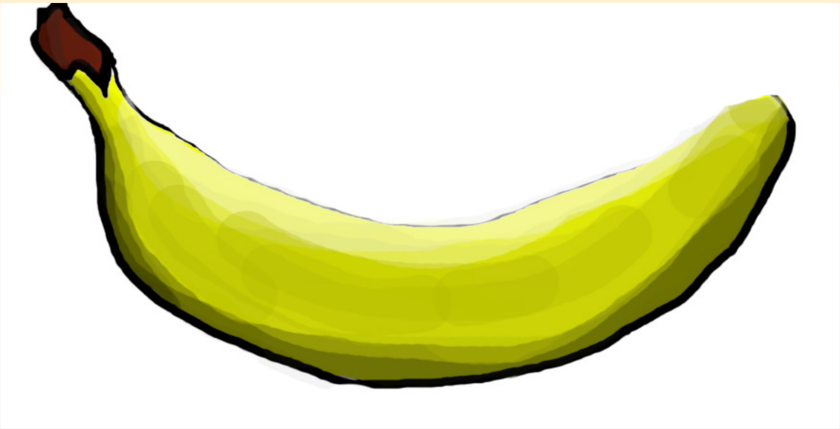
- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

# EXAMPLE

- **For instance**, suppose you are given a basket filled with different kinds of fruits.
- Now the first step is to train the machine with all the different fruits one by one.
- If the shape of the object is rounded and has a depression at the top, is red in color, then it will be labeled as –**Apple**.
- If the shape of the object is a long curving cylinder having Green-Yellow color, then it will be labeled as –**Banana**.

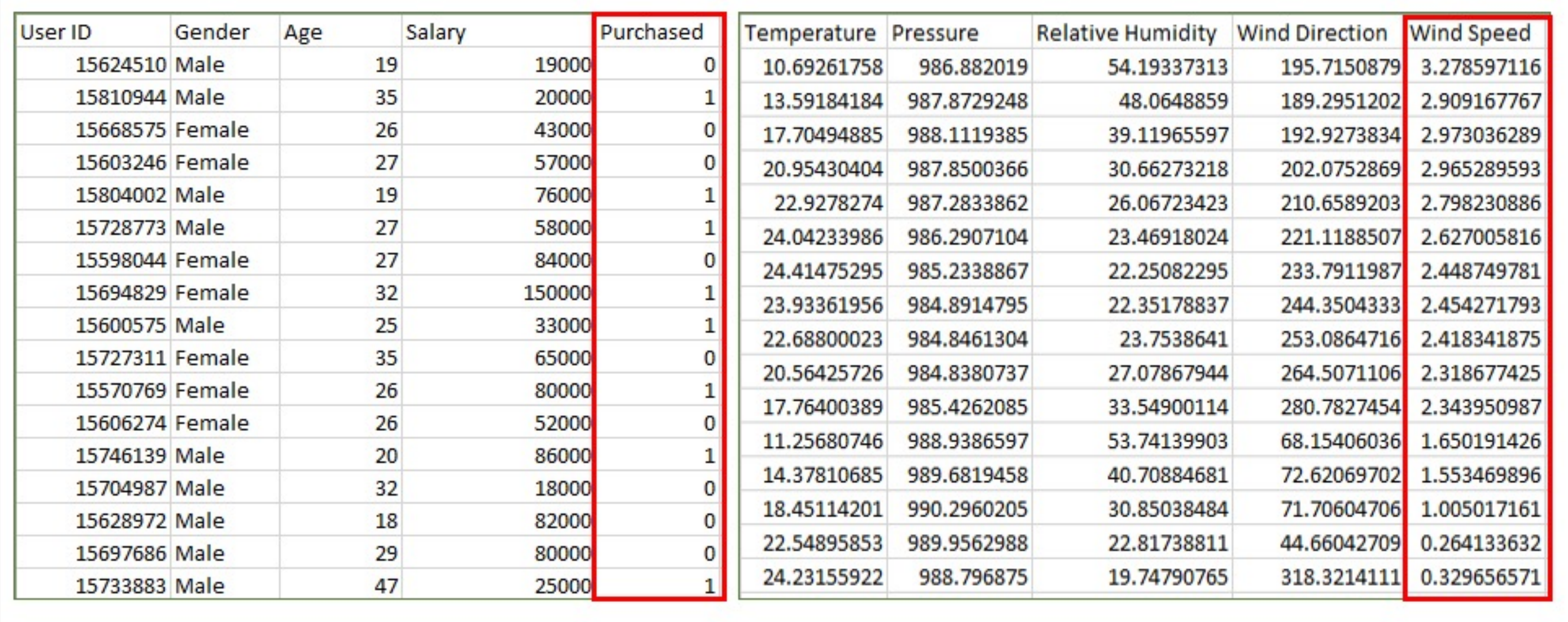
# EXAMPLE

- Now suppose after training the data, you have given a new separate fruit, say Banana from the basket, and asked to identify it.
- Since the machine has already learned the things from previous data and this time has to use it wisely.
- It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in the Banana category.
- Thus, the machine learns the things from training data(basket containing fruits) and then applies the knowledge to test data(new fruit).



## Example

**Supervised learning** is when the model is getting trained on a labelled dataset. A **labelled** dataset is one that has both input and output parameters. In this type of learning both training and validation, datasets are labelled as shown in the figures below.

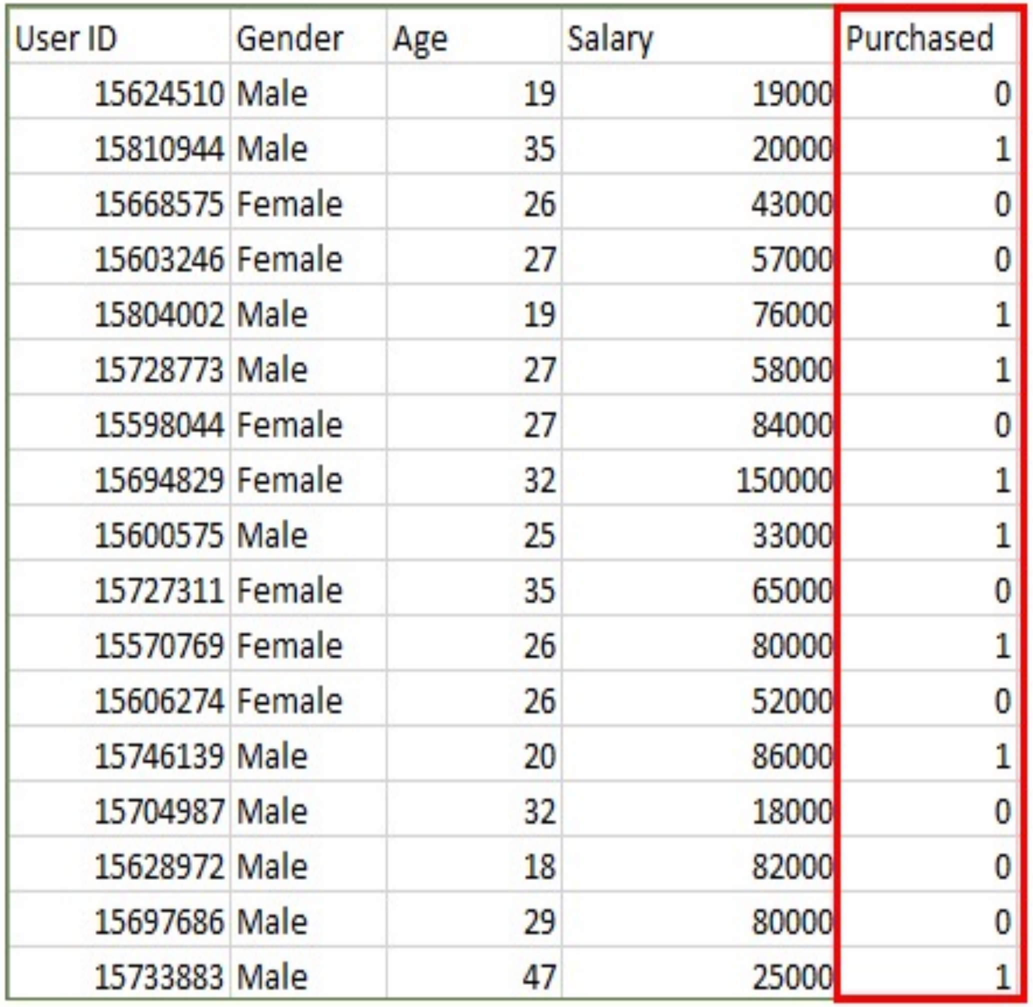


| User ID | Gender | Age | Salary | Purchased |
|---|---|---|---|---|
| 15624510 | Male | 19 | 19000 | 0 |
| 15810944 | Male | 35 | 20000 | 1 |
| 15668575 | Female | 26 | 43000 | 0 |
| 15603246 | Female | 27 | 57000 | 0 |
| 15804002 | Male | 19 | 76000 | 1 |
| 15728773 | Male | 27 | 58000 | 1 |
| 15598044 | Female | 27 | 84000 | 0 |
| 15694829 | Female | 32 | 150000 | 1 |
| 15600575 | Male | 25 | 33000 | 1 |
| 15727311 | Female | 35 | 65000 | 0 |
| 15570769 | Female | 26 | 80000 | 1 |
| 15606274 | Female | 26 | 52000 | 0 |
| 15746139 | Male | 20 | 86000 | 1 |
| 15704987 | Male | 32 | 18000 | 0 |
| 15628972 | Male | 18 | 82000 | 0 |
| 15697686 | Male | 29 | 80000 | 0 |
| 15733883 | Male | 47 | 25000 | 1 |

Figure A: CLASSIFICATION

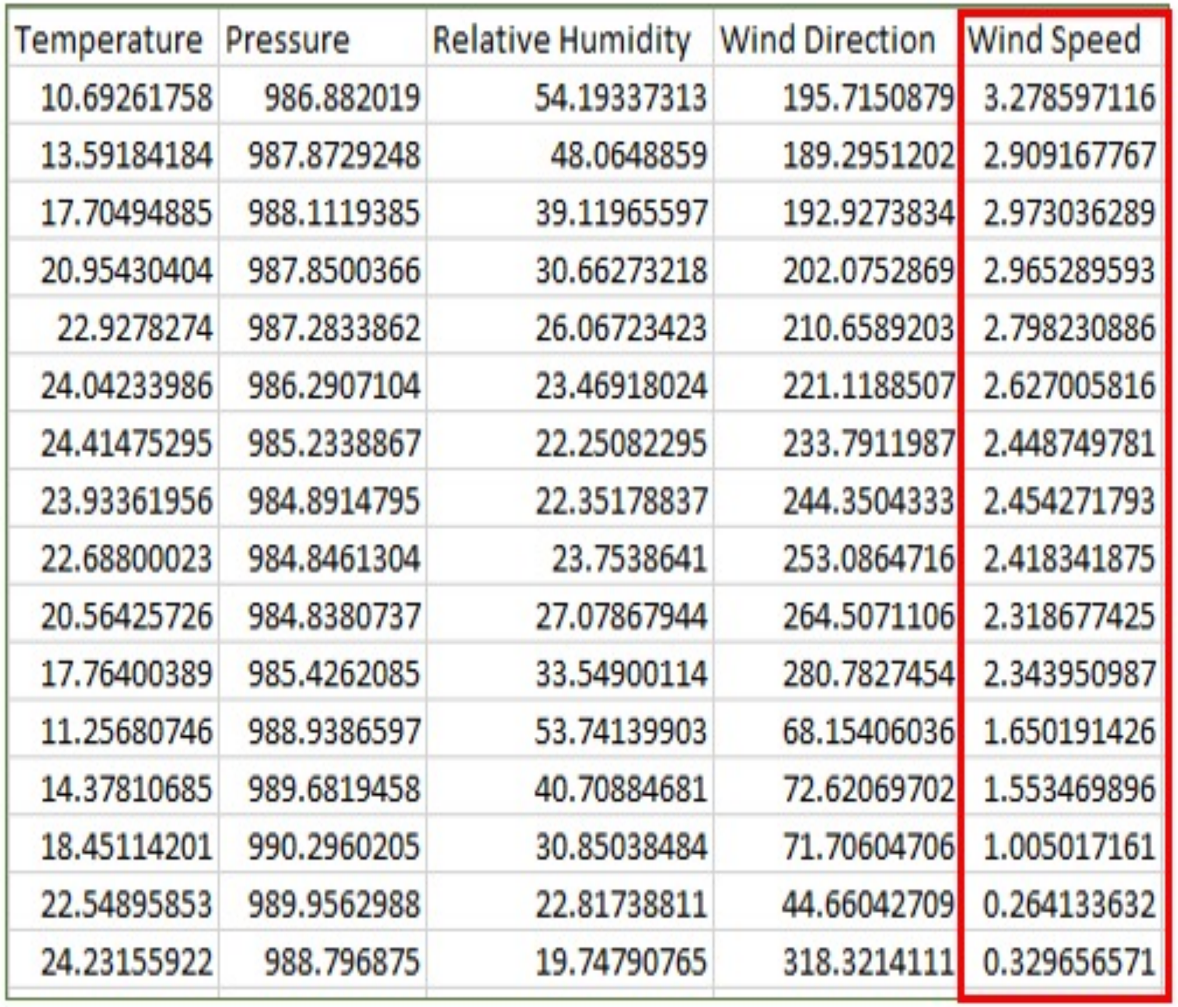
| Temperature | Pressure | Relative Humidity | Wind Direction | Wind Speed |
|---|---|---|---|---|
| 10.69261758 | 986.882019 | 54.19337313 | 195.7150879 | 3.278597116 |
| 13.59184184 | 987.8729248 | 48.0648859 | 189.2951202 | 2.909167767 |
| 17.70494885 | 988.1119385 | 39.11965597 | 192.9273834 | 2.973036289 |
| 20.95430404 | 987.8500366 | 30.66273218 | 202.0752869 | 2.965289593 |
| 22.9278274 | 987.2833862 | 26.06723423 | 210.6589203 | 2.798230886 |
| 24.04233986 | 986.2907104 | 23.46918024 | 221.1188507 | 2.627005816 |
| 24.41475295 | 985.2338867 | 22.25082295 | 233.7911987 | 2.448749781 |
| 23.93361956 | 984.8914795 | 22.35178837 | 244.3504333 | 2.454271793 |
| 22.68800023 | 984.8461304 | 23.7538641 | 253.0864716 | 2.418341875 |
| 20.56425726 | 984.8380737 | 27.07867944 | 264.5071106 | 2.318677425 |
| 17.76400389 | 985.4262085 | 33.54900114 | 280.7827454 | 2.343950987 |
| 11.25680746 | 988.9386597 | 53.74139903 | 68.15406036 | 1.650191426 |
| 14.37810685 | 989.6819458 | 40.70884681 | 72.62069702 | 1.553469896 |
| 18.45114201 | 990.2960205 | 30.85038484 | 71.70604706 | 1.005017161 |
| 22.54895853 | 989.9562988 | 22.81738811 | 44.66042709 | 0.264133632 |
| 24.23155922 | 988.796875 | 19.74790765 | 318.3214111 | 0.329656571 |

Figure B: REGRESSION



| User ID | Gender | Age | Salary | Purchased |
|---|---|---|---|---|
| 15624510 | Male | 19 | 19000 | 0 |
| 15810944 | Male | 35 | 20000 | 1 |
| 15668575 | Female | 26 | 43000 | 0 |
| 15603246 | Female | 27 | 57000 | 0 |
| 15804002 | Male | 19 | 76000 | 1 |
| 15728773 | Male | 27 | 58000 | 1 |
| 15598044 | Female | 27 | 84000 | 0 |
| 15694829 | Female | 32 | 150000 | 1 |
| 15600575 | Male | 25 | 33000 | 1 |
| 15727311 | Female | 35 | 65000 | 0 |
| 15570769 | Female | 26 | 80000 | 1 |
| 15606274 | Female | 26 | 52000 | 0 |
| 15746139 | Male | 20 | 86000 | 1 |
| 15704987 | Male | 32 | 18000 | 0 |
| 15628972 | Male | 18 | 82000 | 0 |
| 15697686 | Male | 29 | 80000 | 0 |
| 15733883 | Male | 47 | 25000 | 1 |

**Figure A:** CLASSIFICATION



| Temperature | Pressure | Relative Humidity | Wind Direction | Wind Speed |
|---|---|---|---|---|
| 10.69261758 | 986.882019 | 54.19337313 | 195.7150879 | 3.278597116 |
| 13.59184184 | 987.8729248 | 48.0648859 | 189.2951202 | 2.909167767 |
| 17.70494885 | 988.1119385 | 39.11965597 | 192.9273834 | 2.973036289 |
| 20.95430404 | 987.8500366 | 30.66273218 | 202.0752869 | 2.965289593 |
| 22.9278274 | 987.2833862 | 26.06723423 | 210.6589203 | 2.798230886 |
| 24.04233986 | 986.2907104 | 23.46918024 | 221.1188507 | 2.627005816 |
| 24.41475295 | 985.2338867 | 22.25082295 | 233.7911987 | 2.448749781 |
| 23.93361956 | 984.8914795 | 22.35178837 | 244.3504333 | 2.454271793 |
| 22.68800023 | 984.8461304 | 23.7538641 | 253.0864716 | 2.418341875 |
| 20.56425726 | 984.8380737 | 27.07867944 | 264.5071106 | 2.318677425 |
| 17.76400389 | 985.4262085 | 33.54900114 | 280.7827454 | 2.343950987 |
| 11.25680746 | 988.9386597 | 53.74139903 | 68.15406036 | 1.650191426 |
| 14.37810685 | 989.6819458 | 40.70884681 | 72.62069702 | 1.553469896 |
| 18.45114201 | 990.2960205 | 30.85038484 | 71.70604706 | 1.005017161 |
| 22.54895853 | 989.9562988 | 22.81738811 | 44.66042709 | 0.264133632 |
| 24.23155922 | 988.796875 | 19.74790765 | 318.3214111 | 0.329656571 |

**Figure B:** REGRESSION

**Example**

Both the figures have labelled data set as follows:

- **Figure A:** It is a dataset of a shopping store that is useful in predicting whether a customer will purchase a particular product under consideration or not based on his/ her gender, age, and salary.

**Input:** Gender, Age, Salary

**Output:** Purchased i.e. 0 or 1; 1 means yes the customer will purchase and 0 means that the customer won’t purchase it.

- **Figure B:** It is a Meteorological dataset that serves the purpose of predicting wind speed based on different parameters.

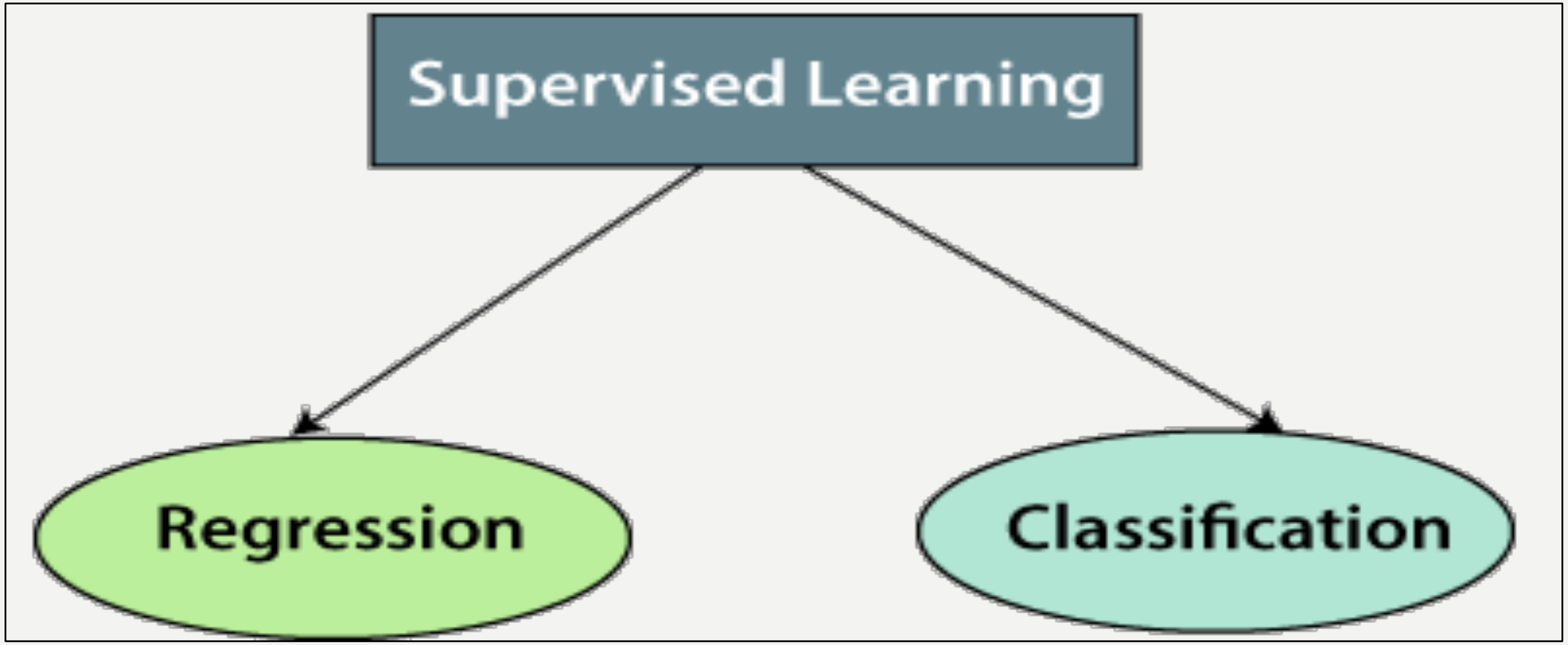
**Input:** Dew Point, Temperature, Pressure, Relative Humidity, Wind Direction

**Output:** Wind Speed

# TYPES OF SUPERVISED MACHINE LEARNING ALGORITHMS

•**Classification**: A classification problem is when the output variable is a category, such as "Red" or "blue" ,"disease" or "no disease".

•**Regression**: A regression problem is when the output variable is a real value, such as "dollars" or "weight".



# Supervised learning

## Regression

- Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:
- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

## Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, Spam Filtering, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

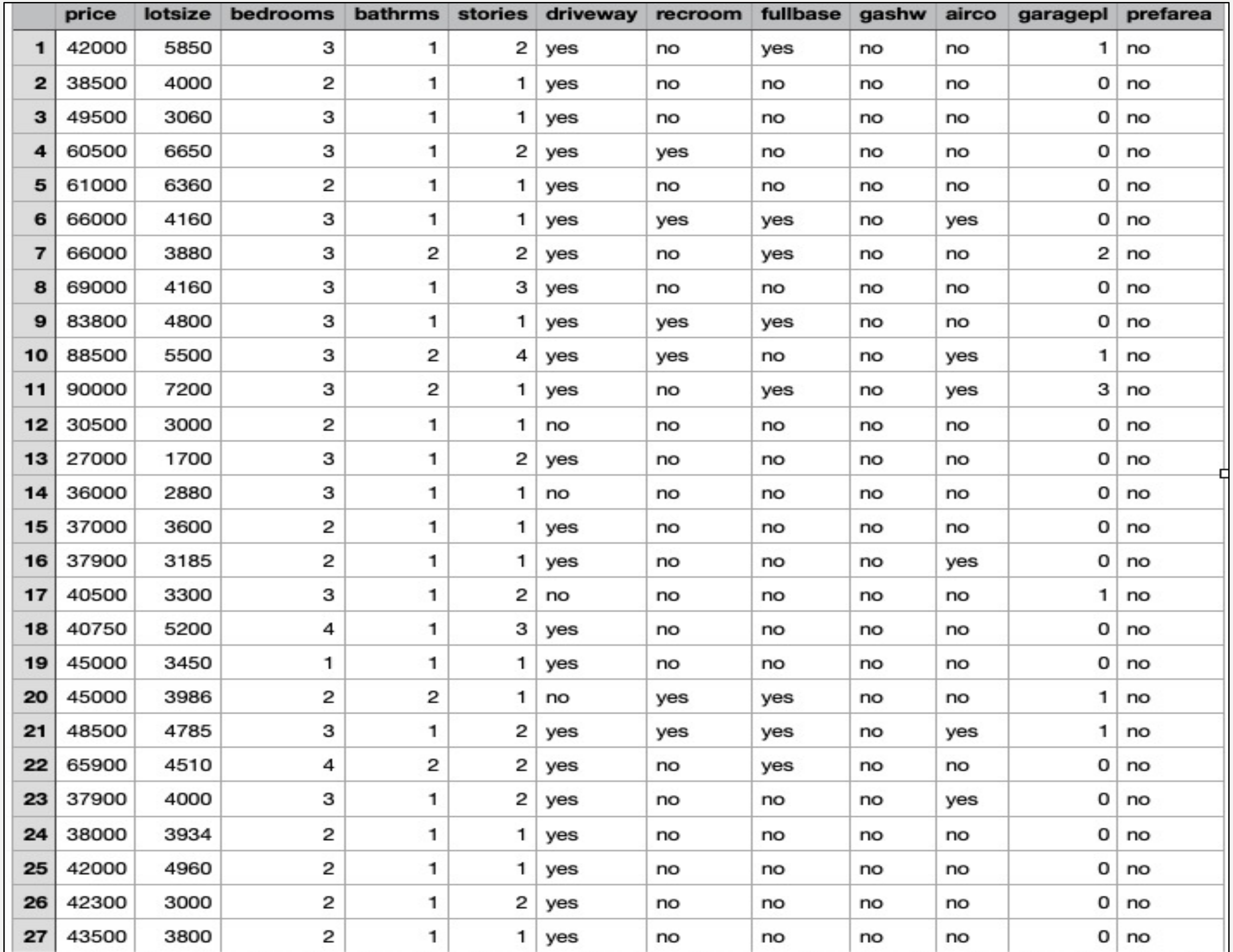
# WHICH OF THE FOLLOWING IS A REGRESSION TASK?

- Predicting age of a person ✓
- Predicting nationality of a person ✗
- Predicting whether stock price of a company will increase tomorrow ✗
- Predicting whether a document is related to sighting of UFOs? ✗

**Solution :**

Predicting age of a person (because it is a real value, predicting nationality is categorical, whether stock price will increase is discrete-yes/no answer, predicting whether a document is related to UFO is again discrete- a yes/no answer). Let’s take an example of linear regression.

Data set example

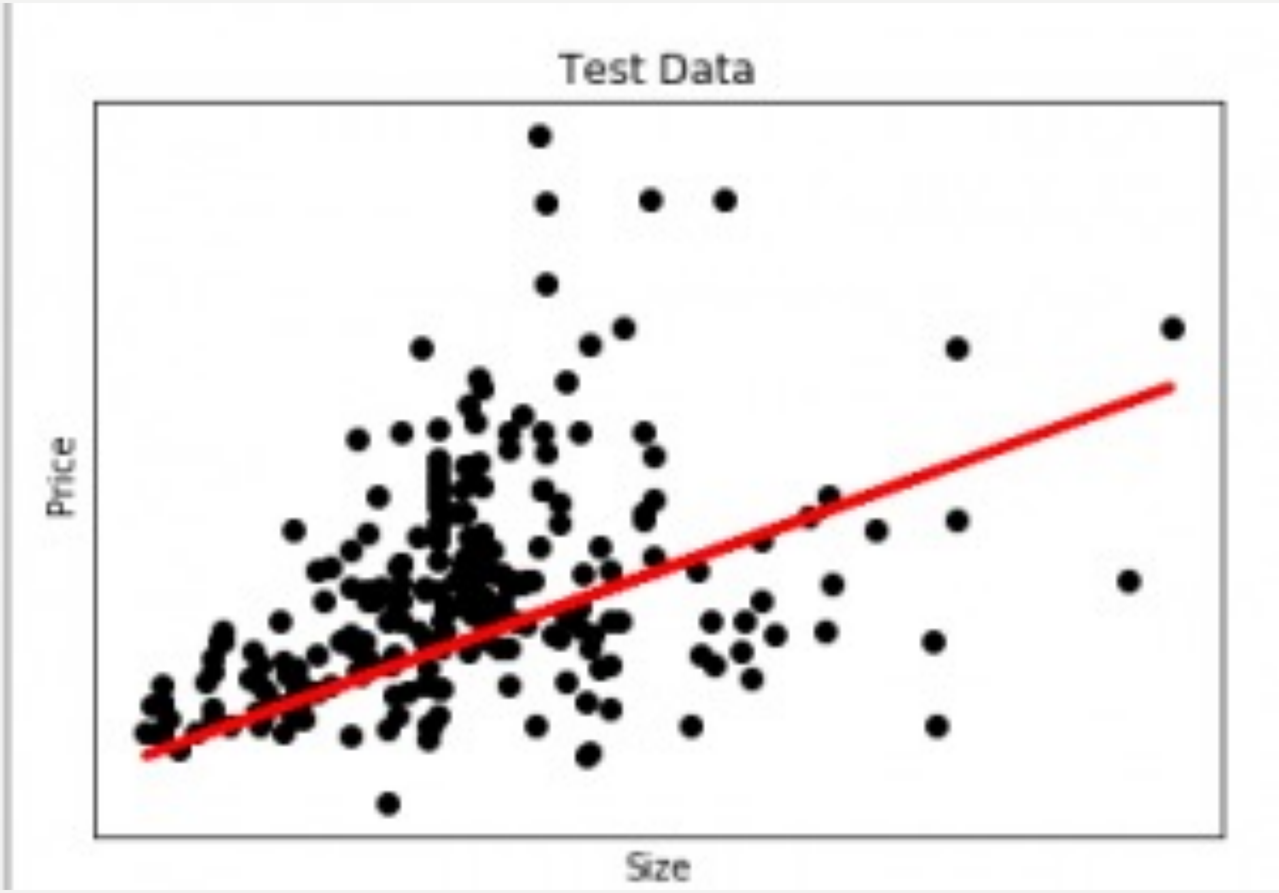


| | price | lotsize | bedrooms | bathrms | stories | driveway | recroom | fullbase | gashw | airco | garagepl | prefarea |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 42000 | 5850 | 3 | 1 | 2 | yes | no | yes | no | no | 1 | no |
| 2 | 38500 | 4000 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 3 | 49500 | 3060 | 3 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 4 | 60500 | 6650 | 3 | 1 | 2 | yes | yes | no | no | no | 0 | no |
| 5 | 61000 | 6360 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 6 | 66000 | 4160 | 3 | 1 | 1 | yes | yes | yes | no | yes | 0 | no |
| 7 | 66000 | 3880 | 3 | 2 | 2 | yes | no | yes | no | no | 2 | no |
| 8 | 69000 | 4160 | 3 | 1 | 3 | yes | no | no | no | no | 0 | no |
| 9 | 83800 | 4800 | 3 | 1 | 1 | yes | yes | yes | no | no | 0 | no |
| 10 | 88500 | 5500 | 3 | 2 | 4 | yes | yes | no | no | yes | 1 | no |
| 11 | 90000 | 7200 | 3 | 2 | 1 | yes | no | yes | no | yes | 3 | no |
| 12 | 30500 | 3000 | 2 | 1 | 1 | no | no | no | no | no | 0 | no |
| 13 | 27000 | 1700 | 3 | 1 | 2 | yes | no | no | no | no | 0 | no |
| 14 | 36000 | 2880 | 3 | 1 | 1 | no | no | no | no | no | 0 | no |
| 15 | 37000 | 3600 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 16 | 37900 | 3185 | 2 | 1 | 1 | yes | no | no | no | yes | 0 | no |
| 17 | 40500 | 3300 | 3 | 1 | 2 | no | no | no | no | no | 1 | no |
| 18 | 40750 | 5200 | 4 | 1 | 3 | yes | no | no | no | no | 0 | no |
| 19 | 45000 | 3450 | 1 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 20 | 45000 | 3986 | 2 | 2 | 1 | no | yes | yes | no | no | 1 | no |
| 21 | 48500 | 4785 | 3 | 1 | 2 | yes | yes | yes | no | yes | 1 | no |
| 22 | 65900 | 4510 | 4 | 2 | 2 | yes | no | yes | no | no | 0 | no |
| 23 | 37900 | 4000 | 3 | 1 | 2 | yes | no | no | no | yes | 0 | no |
| 24 | 38000 | 3934 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 25 | 42000 | 4960 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |
| 26 | 42300 | 3000 | 2 | 1 | 2 | yes | no | no | no | no | 0 | no |
| 27 | 43500 | 3800 | 2 | 1 | 1 | yes | no | no | no | no | 0 | no |

```
# Python code to illustrate
# regression using data set
import matplotlib
matplotlib.use('GTKAgg')

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
import pandas as pd

# Load CSV and columns
df = pd.read_csv("Housing.csv")

Y = df['price']
X = df['lotsize']

X=X.values.reshape(len(X),1)
Y=Y.values.reshape(len(Y),1)

# Split the data into training/testing sets
X_train = X[:-250]
X_test = X[-250:]
```

```
# Split the data into training/testing sets
X_train = X[:-250]
X_test = X[-250:]
# Split the targets into training/testing sets
Y_train = Y[:-250]
Y_test = Y[-250:]
# Plot outputs
plt.scatter(X_test,Y_test, color='black')
plt.title('Test Data')
plt.xlabel('Size')
plt.ylabel('Price')
plt.xticks(())
plt.yticks(())
# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train,Y_train)

# Plot outputs
plt.plot(X_test, regr.predict(X_test), color='red',linewidth=3)
plt.show()
```

The output of the code will be:



Here in this graph, we plot the test data. The red line indicates the best fit line for predicting the price.

# WHICH OF THE FOLLOWING IS/ARE CLASSIFICATION PROBLEM(S)?

- Predicting the gender of a person by his/her handwriting style. ✓
- Predicting house price based on area. ✗
- Predicting whether monsoon will be normal next year. ✓
- Predict the number of copies a music album will be sold next month. ✗

Solution :
Predicting the gender of a person.

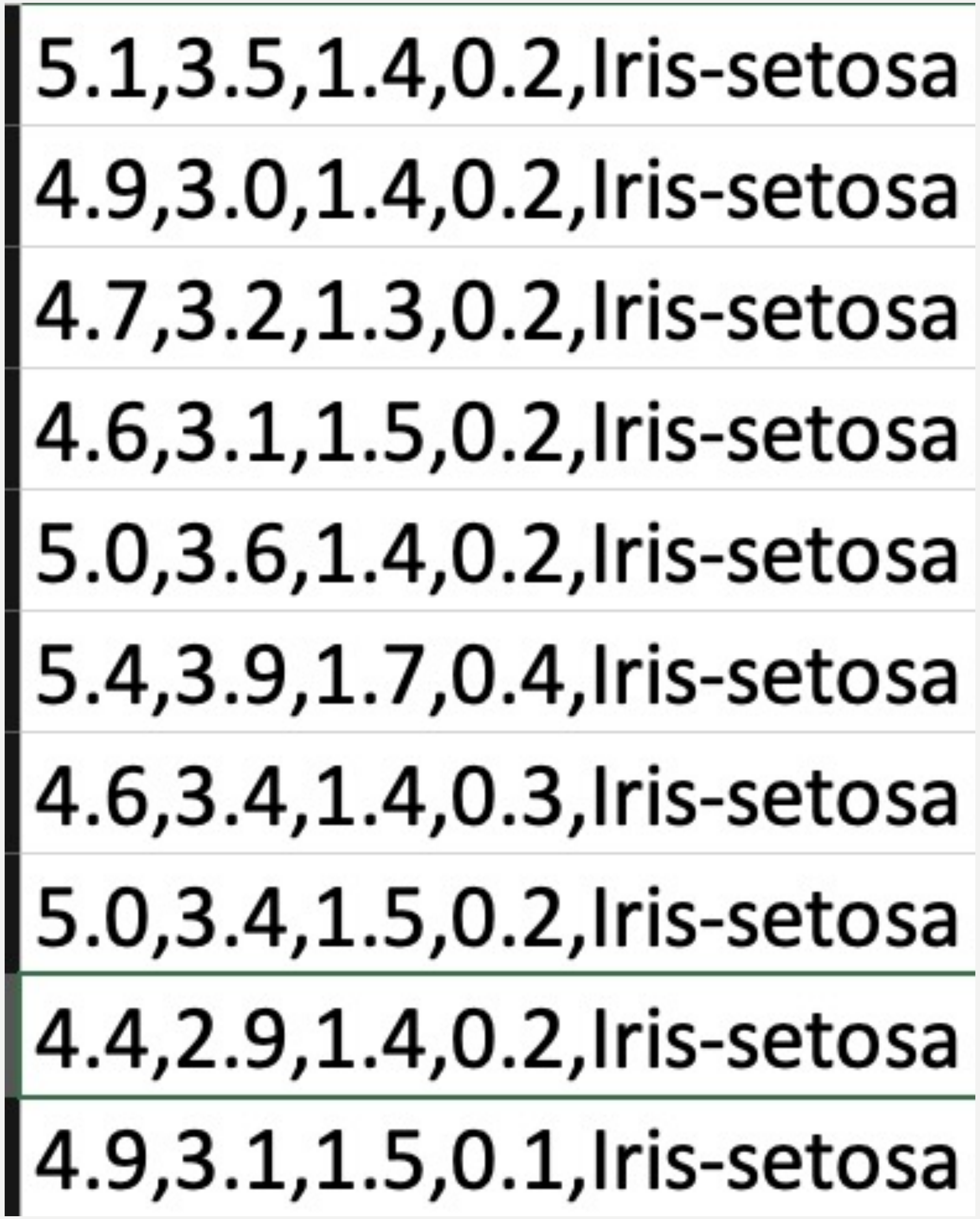
Predicting whether monsoon will be normal next year.

The other two are regression.

Title:
Iris Plants Database Attribute
Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
   - -- Iris Setosa
   - -- Iris Versicolour
   - -- Iris Virginica

Missing Attribute Values: None
Class Distribution: 33.3% for each of 3 classes

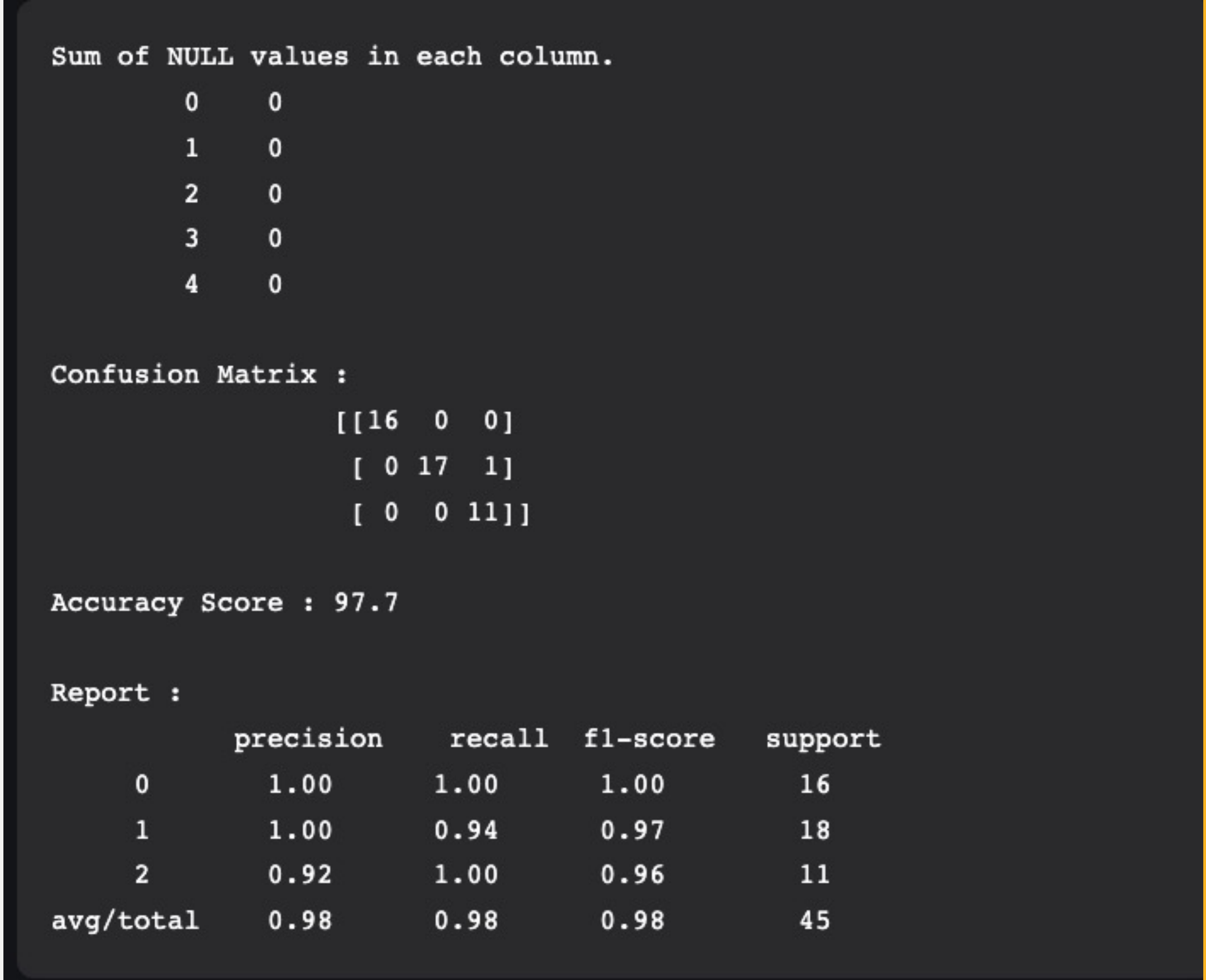


```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

```
# Python code to illustrate
# classification using data set
#Importing the required library
import pandas as pd
from sklearn.model_selection import
train_test_split
from sklearn.ensemble import
RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

#Importing the dataset
dataset = pd.read_csv(
            'https://archive.ics.uci.edu/ml/machine-
learning-'+
            'databases/iris/iris.data',sep= ',', header=
None)
data = dataset.iloc[:, :]

#checking for null values
print("Sum of NULL values in each column. ")
print(data.isnull().sum())
```

```
#separating the predicting column from the whole
dataset
X = data.iloc[:, :-1].values
y = dataset.iloc[:, 4].values

#Encoding the predicting variable
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)

#Splitting the data into test and train dataset
X_train, X_test, y_train, y_test = train_test_split(
                  X, y, test_size = 0.3, random_state = 0)
#Using the random forest classifier for the
prediction
classifier=RandomForestClassifier()
classifier=classifier.fit(X_train,y_train)
predicted=classifier.predict(X_test)

#printing the results
print ('Confusion Matrix :')
print(confusion_matrix(y_test, predicted))
print ('Accuracy Score :',accuracy_score(y_test, predicted))
print ('Report : ')
print (classification_report(y_test, predicted))
```

# OUTPUT



```
Sum of NULL values in each column.
        0    0
        1    0
        2    0
        3    0
        4    0

Confusion Matrix :
                 [[16  0  0]
                  [ 0 17  1]
                  [ 0  0 11]]

Accuracy Score : 97.7

Report :
           precision    recall  f1-score   support
     0       1.00       1.00      1.00        16
     1       1.00       0.94      0.97        18
     2       0.92       1.00      0.96        11
avg/total    0.98       0.98      0.98        45
```

# SUPERVISED LEARNING

## Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

## Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

# UNSUPERVISED LEARNING

- Unsupervised learning is also a relatively simple learning model, but as the name suggests, it lacks a critic and has no way to measure its performance.
- The goal is to build a mapping function that categorizes the data into classes based on features hidden within the data.
- As with supervised learning, you use unsupervised learning in two phases.
- In the **first phase**, the mapping function segments a data set into classes. Each input vector becomes part of a class, but the algorithm cannot apply labels to those classes.
- The segmentation of the data into classes may be the result (from which you can then draw conclusions about the resulting classes), but you can use these classes further depending on the application.
- One such application is a recommendation system, where the input vector may represent the characteristics or purchases of a user, and users within a class represent those with similar interests who can then be used for marketing or product recommendations.

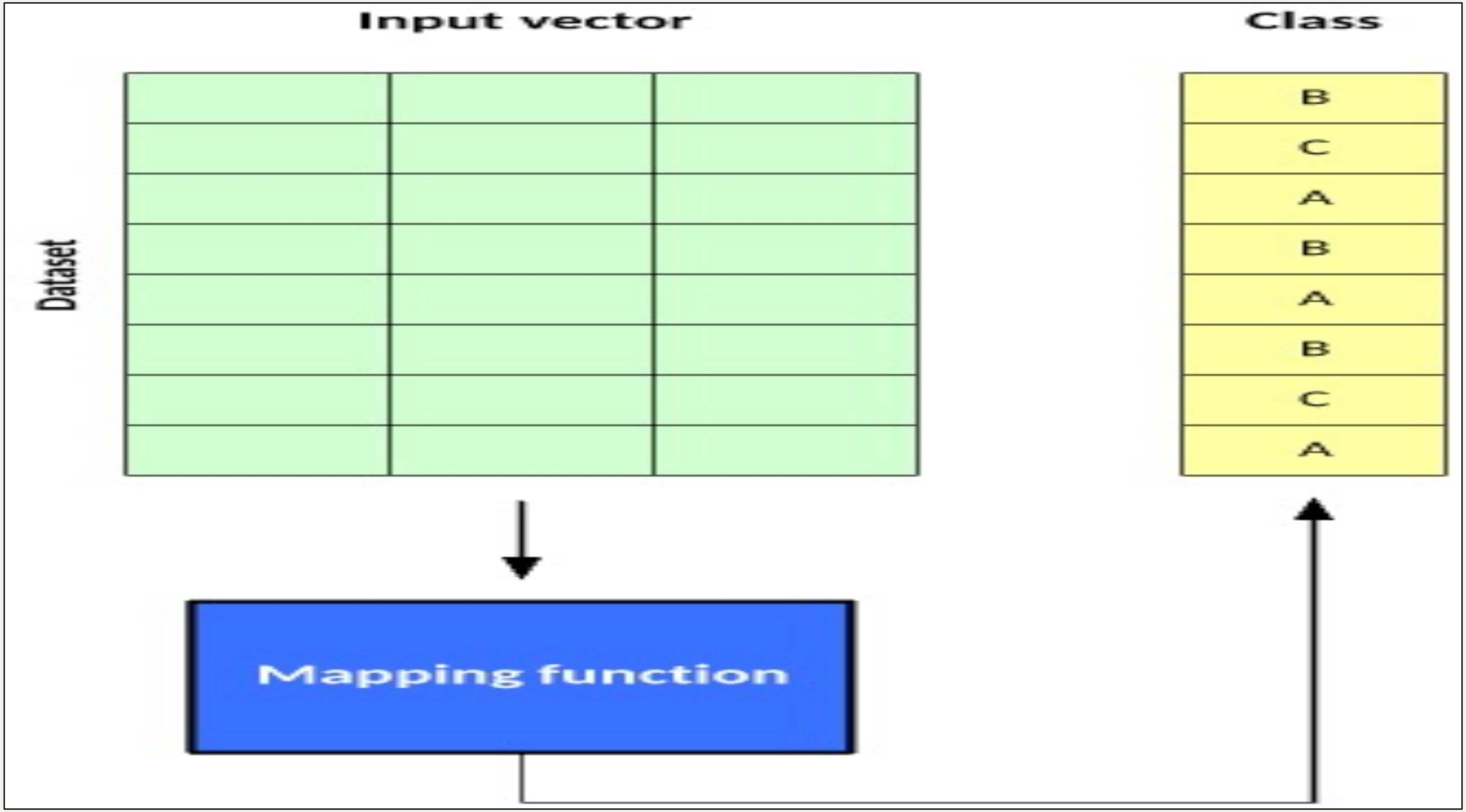


**Figure . The two phases of using unsupervised learning**

# UNSUPERVISED LEARNING

- As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data.
- It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

*"Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision."*

- Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.
- The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

# UNSUPERVISED LEARNING

**Example:**

- Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs.
- The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset.
- The task of the unsupervised learning algorithm is to identify the image features on their own.
- Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

# UNSUPERVISED LEARNING

**Example:**

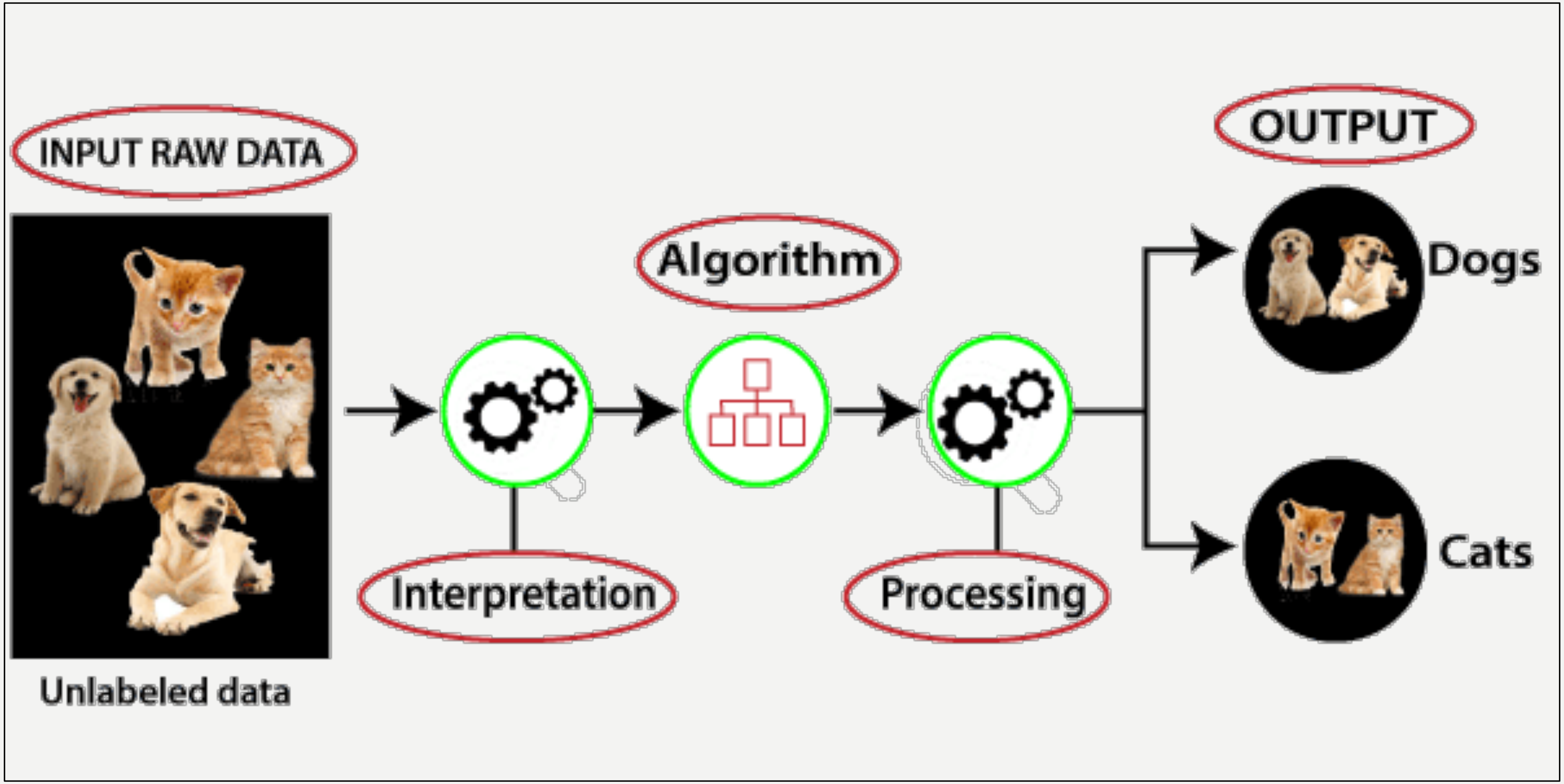
- Thus, the machine has no idea about the features of dogs and cats so we can't categorize it as 'dogs and cats '.
- But it can categorize them according to their similarities, patterns, and differences, i.e., we can easily categorize the above picture into two parts.
- The first may contain all pics having **dogs** in them and the second part may contain all pics having **cats** in them.
- Here you didn't learn anything before, which means no training data or examples.

# UNSUPERVISED LEARNING

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabelled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

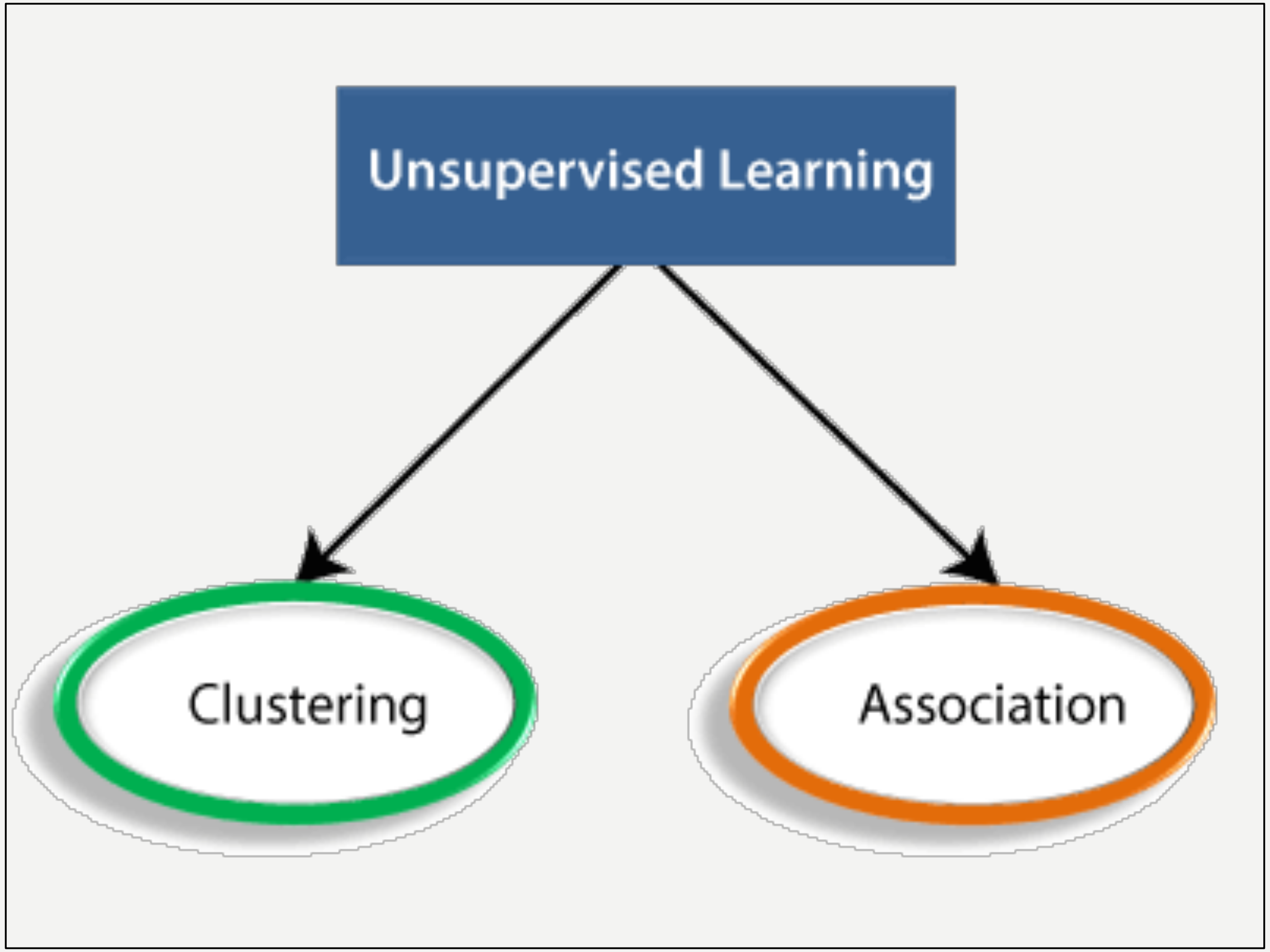
# WORKING OF UNSUPERVISED LEARNING



# WORKING OF UNSUPERVISED LEARNING

- Here, we have taken an unlabelled input data, which means it is not categorized and corresponding outputs are also not given.
- Now, this unlabelled input data is fed to the machine learning model in order to train it.
- Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.
- Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

# TYPES OF UNSUPERVISED LEARNING ALGORITHM



# TYPES OF UNSUPERVISED LEARNING ALGORITHM

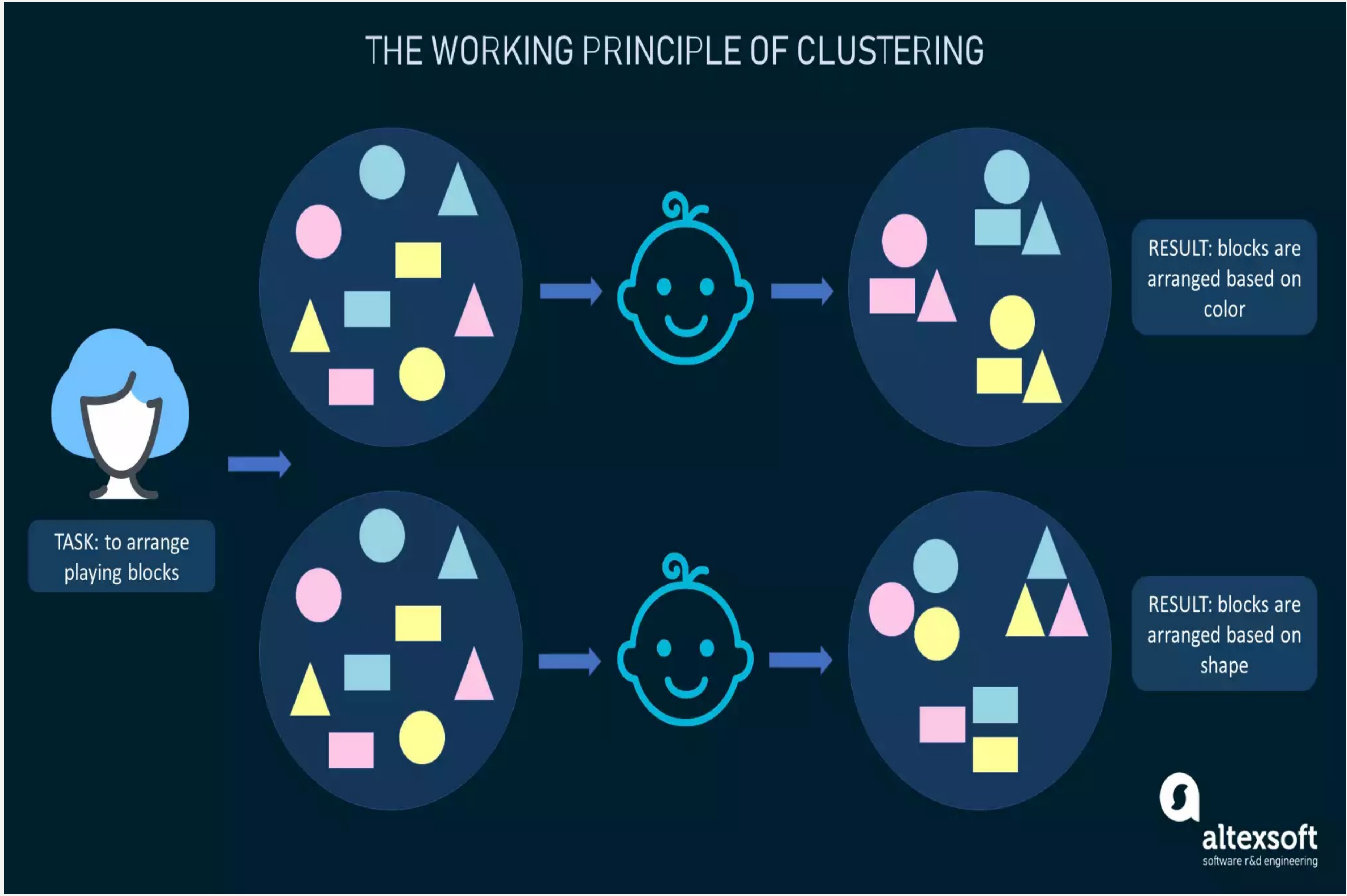
Clustering:

- Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group.
- Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

Association:

- An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset.
- Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item.

# CLUSTERING



THE WORKING PRINCIPLE OF CLUSTERING

TASK: to arrange playing blocks

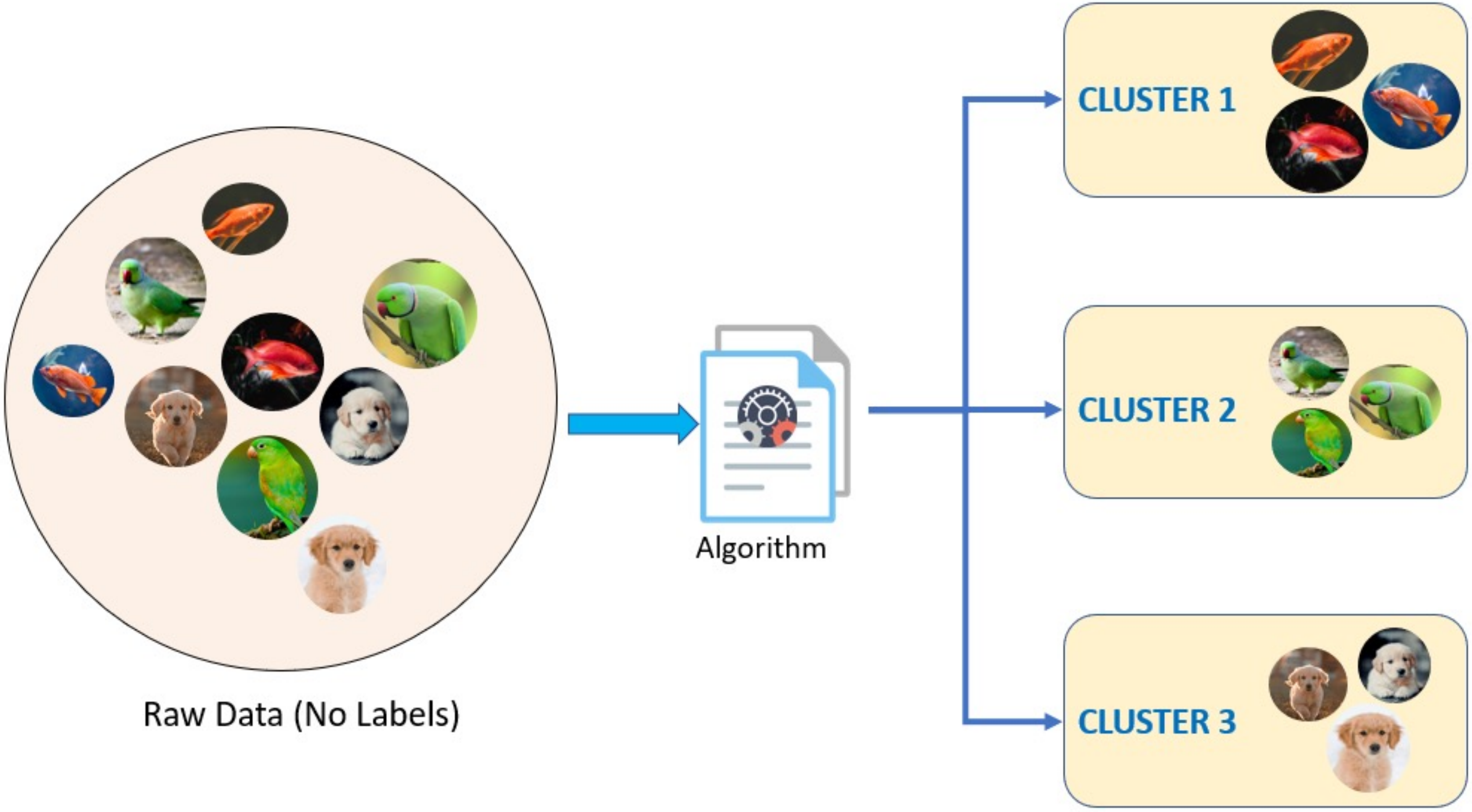
RESULT: blocks are arranged based on color

RESULT: blocks are arranged based on shape

# CLUSTERING

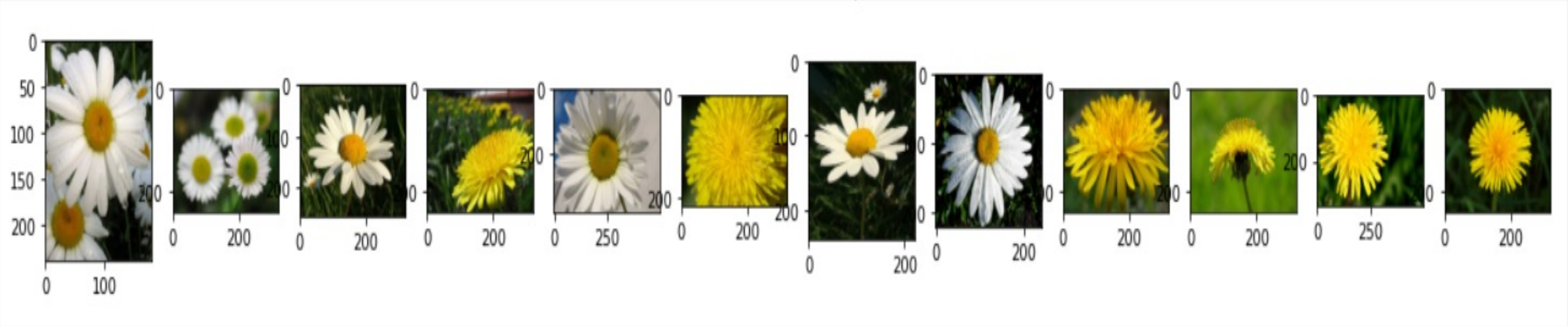
- To explain the clustering approach, here's a simple analogy.
- In a kindergarten, a teacher asks children to arrange blocks of different shapes and colors.
- Suppose each child gets a set containing rectangular, triangular, and round blocks in yellow, blue, and pink.
- The thing is a teacher hasn't given the criteria on which the arrangement should be done so different children came up with different groupings.
- Some kids put all blocks into three clusters based on the color – yellow, blue, and pink.
- Others categorized the same blocks based on their shape – rectangular, triangular, and round.
- There is no right or wrong way to perform grouping as there was no task set in advance.
- That's the whole beauty of clustering: It helps unfold various business insights you never knew were there.

# CLUSTERING EXAMPLE



Unsupervised Machine Learning

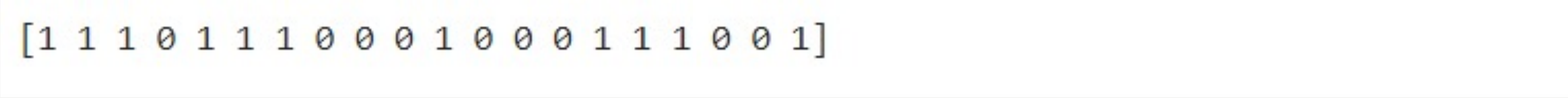
DATA - 15 images of two categories - daisy and dandelion



Using one of the pre-trained image classification models for transfer learning mentioned earlier, we can extract the features of these images. Once the features are extracted, we will apply the K-Means algorithm using the following code –

```
#Creating Clusters
kmeans = KMeans(n_clusters=2, init='k-means++', random_state=0)
# predict a label for each image based on clusters
Y = kmeans.fit_predict(img_features)
print(Y)
```

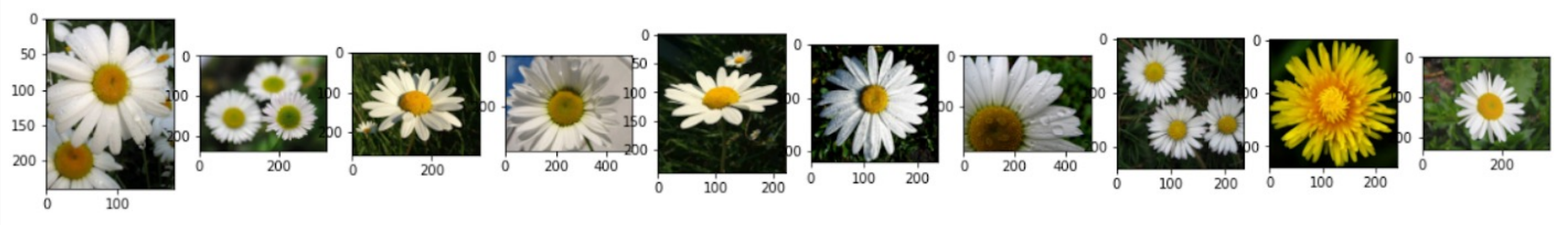
This is the output of the model which indicates the category of all the input images.

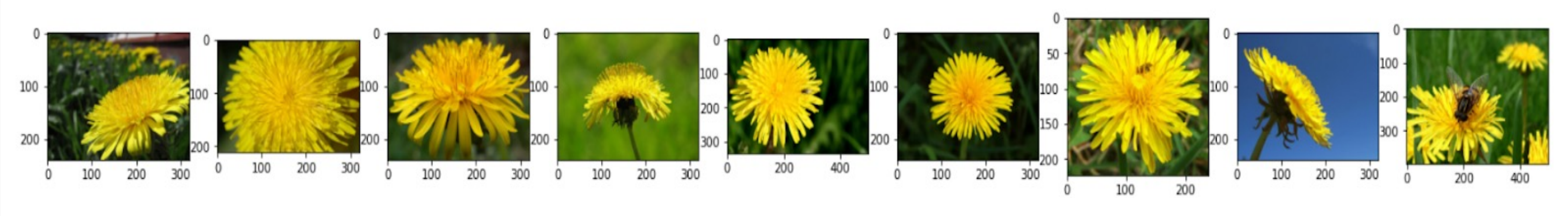


```
[1 1 1 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1]
```

- Since we have only two categories of flowers, we are using the number of clusters =2. However, for larger datasets with unknown labels, it is essential to experiment with the number of cluster values for acceptable results.
- Here are a few images from each cluster -

CLUSTER1





CLUSTER2

NOTE : It is clear that the model did a good job at grouping the images into two categories. All images except one image were correctly clustered.

# UNSUPERVISED LEARNING ALGORITHMS

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

# UNSUPERVISED LEARNING ALGORITHMS

**Advantages of Unsupervised Learning**

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

**Disadvantages of Unsupervised Learning**

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

# DIFFERENCE BETWEEN SUPERVISED AND UNSUPERVISED LEARNING

| Supervised Learning | Unsupervised Learning |
|---|---|
| Supervised learning algorithms are trained using labeled data. | Unsupervised learning algorithms are trained using unlabeled data. |
| Supervised learning model takes direct feedback to check if it is predicting correct output or not. | Unsupervised learning model does not take any feedback. |
| Supervised learning model predicts the output. | Unsupervised learning model finds the hidden patterns in data. |
| In supervised learning, input data is provided to the model along with the output. | In unsupervised learning, only input data is provided to the model. |
| The goal of supervised learning is to train the model so that it can predict the output when it is given new data. | The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset. |
| Supervised learning needs supervision to train the model. | Unsupervised learning does not need any supervision to train the model. |

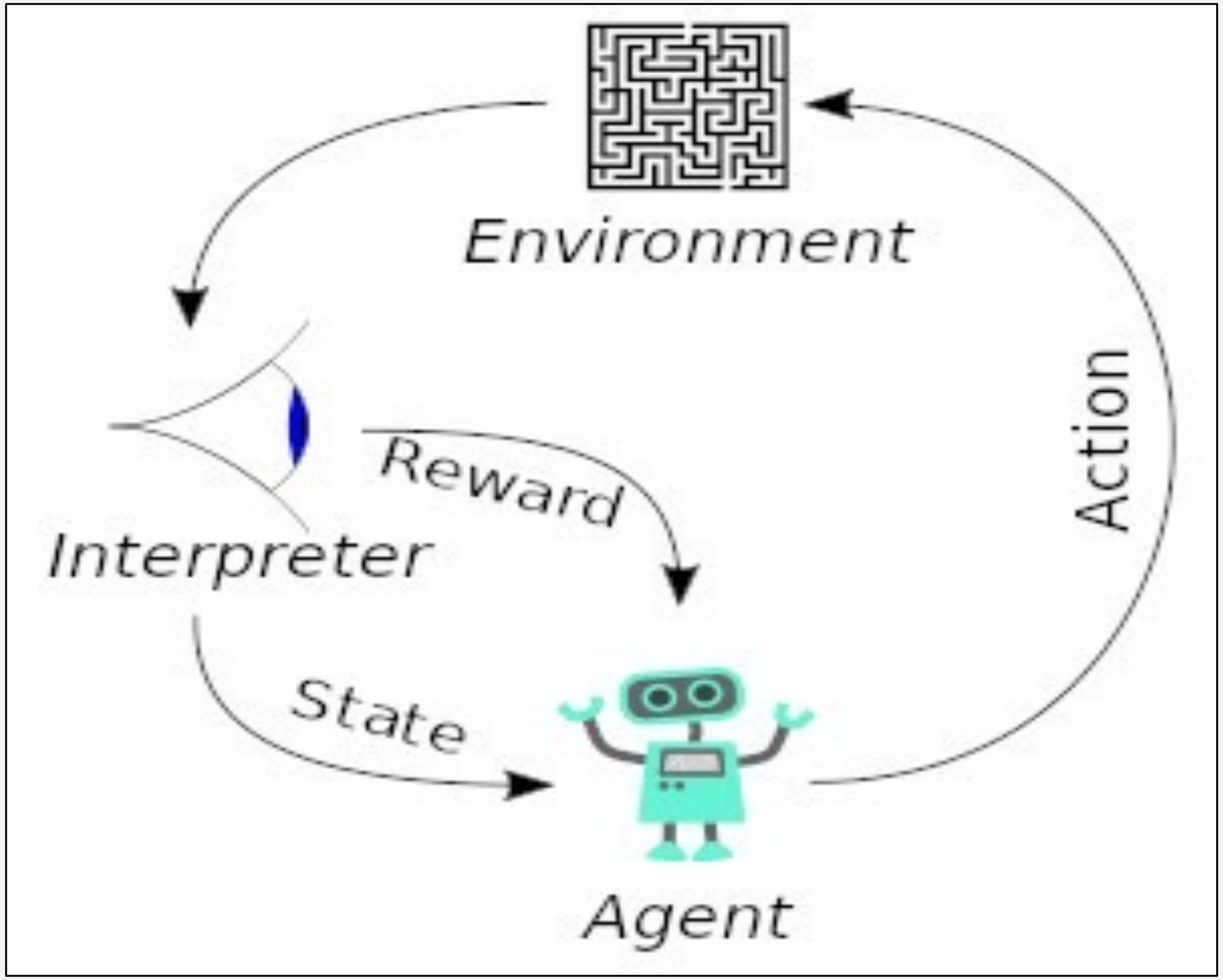
# DIFFERENCE BETWEEN SUPERVISED AND UNSUPERVISED LEARNING

| Supervised Learning | Unsupervised Learning |
|---|---|
| Supervised learning can be categorized in **Classification** and **Regression** problems. | Unsupervised Learning can be classified in **Clustering** and **Associations** problems. |
| Supervised learning can be used for those cases where we know the input as well as corresponding outputs. | Unsupervised learning can be used for those cases where we have only input data and no corresponding output data. |
| Supervised learning model produces an accurate result. | Unsupervised learning model may give less accurate result as compared to supervised learning. |
| Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output. | Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences. |
| It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc. | It includes various algorithms such as Clustering, KNN, and Apriori algorithm. |

# REINFORCEMENT LEARNING

- Reinforcement learning is a learning model, with the ability not just to learn how to map an input to an output but to map a series of inputs to outputs with dependencies (Markov decision processes, for example).
- Reinforcement learning exists in the context of states in an environment and the actions possible at a given state.
- During the learning process, the algorithm randomly explores the state–action pairs within some environment (to build a state–action pair table), then in practice of the learned information exploits the state–action pair rewards to choose the best action for a given state that lead to some goal state.

# REINFORCEMENT LEARNING



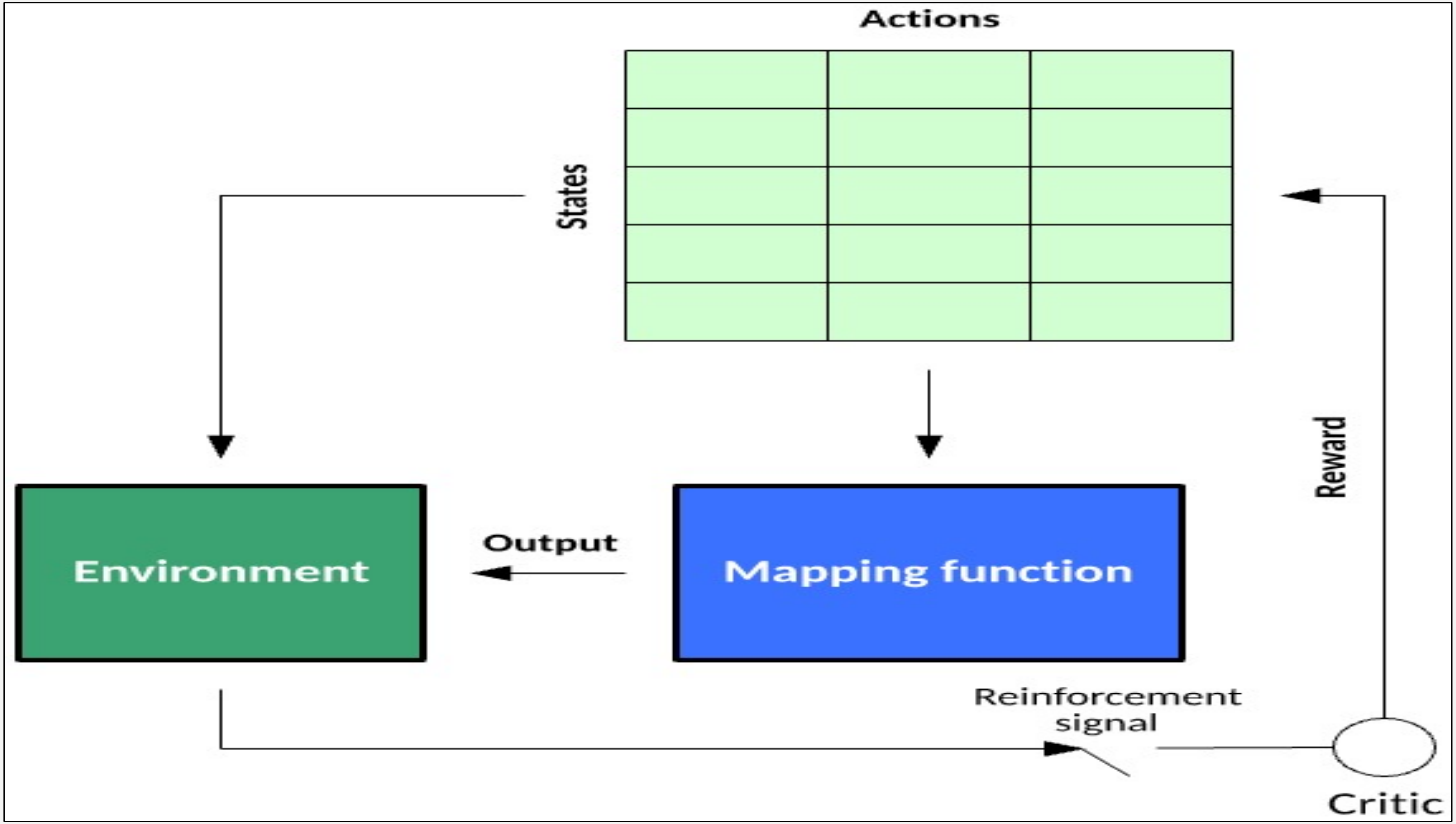


Figure: The reinforcement learning model

# EXAMPLE

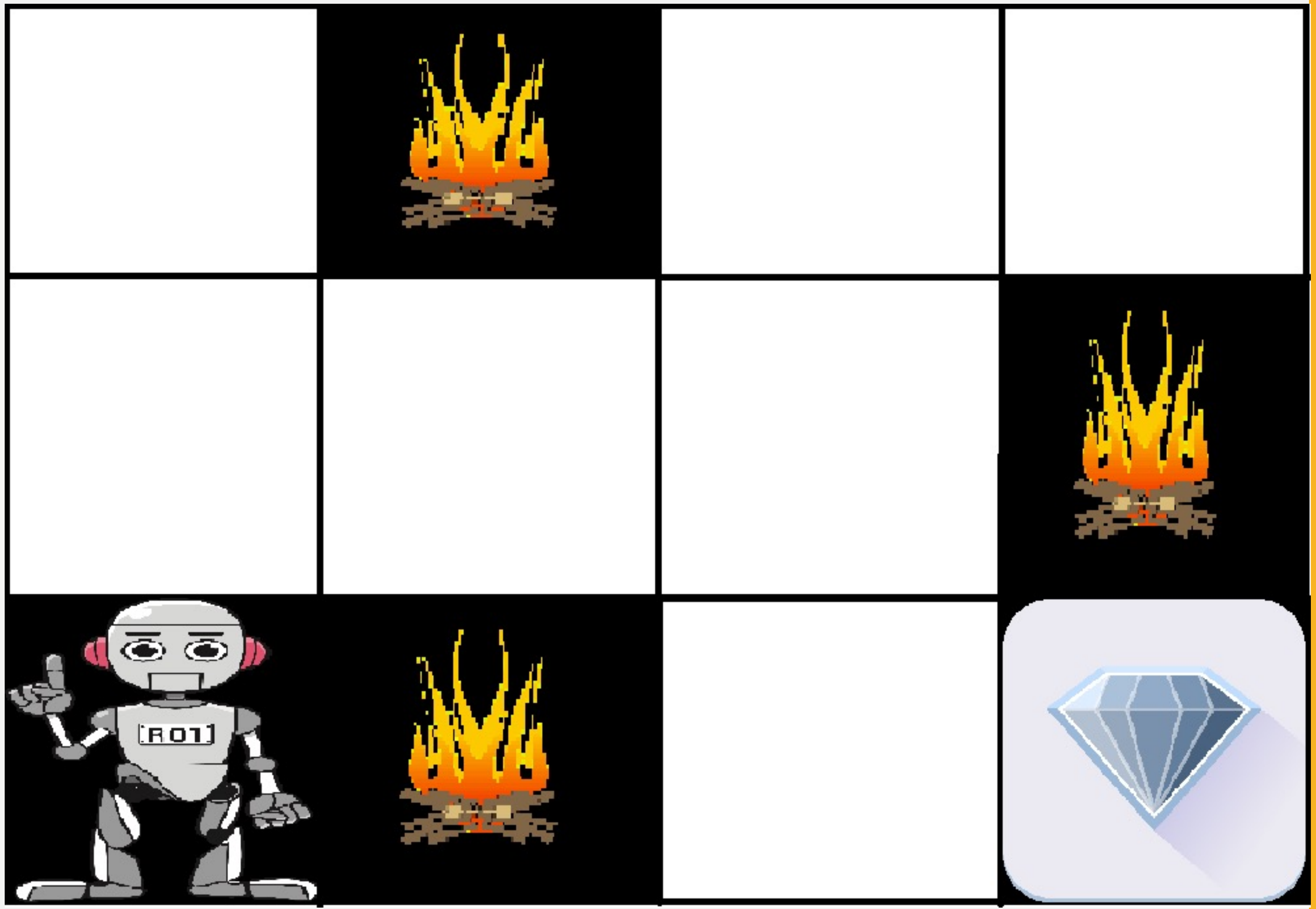
- Consider a simple agent that plays blackjack.
- The states represent the sum of the cards for the player.
- The actions represent what a blackjack-playing agent may do — in this case, hit or stand.
- Training an agent to play blackjack would involve many hands of poker, where reward for a given state–action nexus is given for winning or losing.
- For example, the value for a state of 10 would be 1.0 for hit and 0.0 for stand (indicating that hit is the optimal choice).
- For state 20, the learned reward would likely be 0.0 for hit and 1.0 for stand.

# EXAMPLE

- For a less-straightforward hand, a state of 17 may have action values of 0.95 stand and 0.05 hit.
- This agent would then probabilistically stand 95 percent of the time and hit 5 percent of the time.
- These rewards would be leaned over many hands of poker, indicating the best choice for a given state (or hand).
- Unlike supervised learning, where a critic grades each example, in reinforcement learning, that critic may only provide a grade when the goal state is met (having a hand with the state of 21).

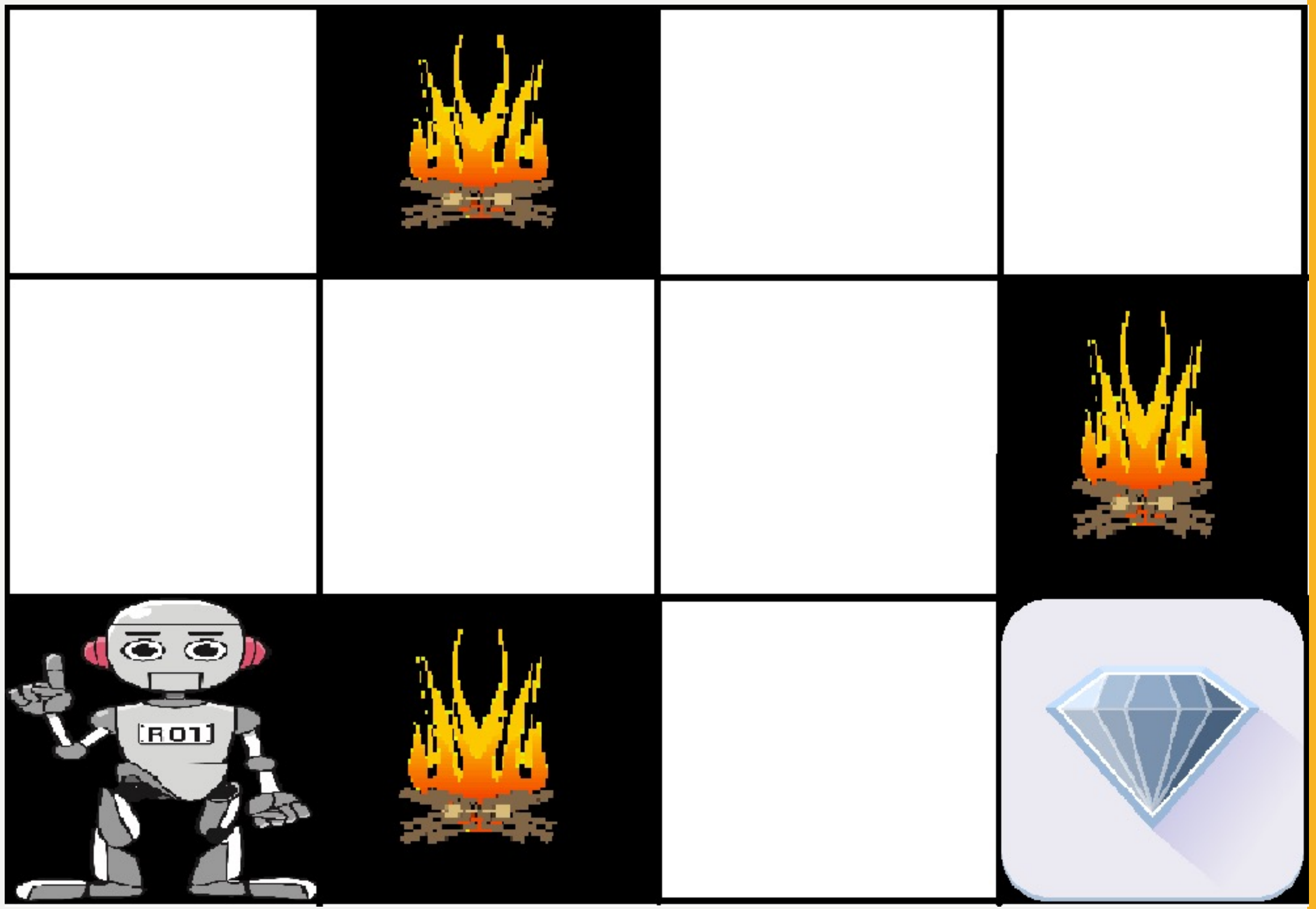
# EXAMPLE

- We have an agent and a reward, with many hurdles in between.
- The agent is supposed to find the best possible path to reach the reward.
- The following problem explains the problem more easily.
- The above image shows the robot, diamond, and fire.
- The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired.



# EXAMPLE

- The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles.
- Each right step will give the robot a reward and each wrong step will subtract the reward of the robot.
- The total reward will be calculated when it reaches the final reward that is the diamond.



# REINFORCEMENT LEARNING

**Main points in Reinforcement learning –**

- Input:

The input should be an initial state from which the model will start

- Output:

There are many possible outputs as there are a variety of solutions to a particular problem

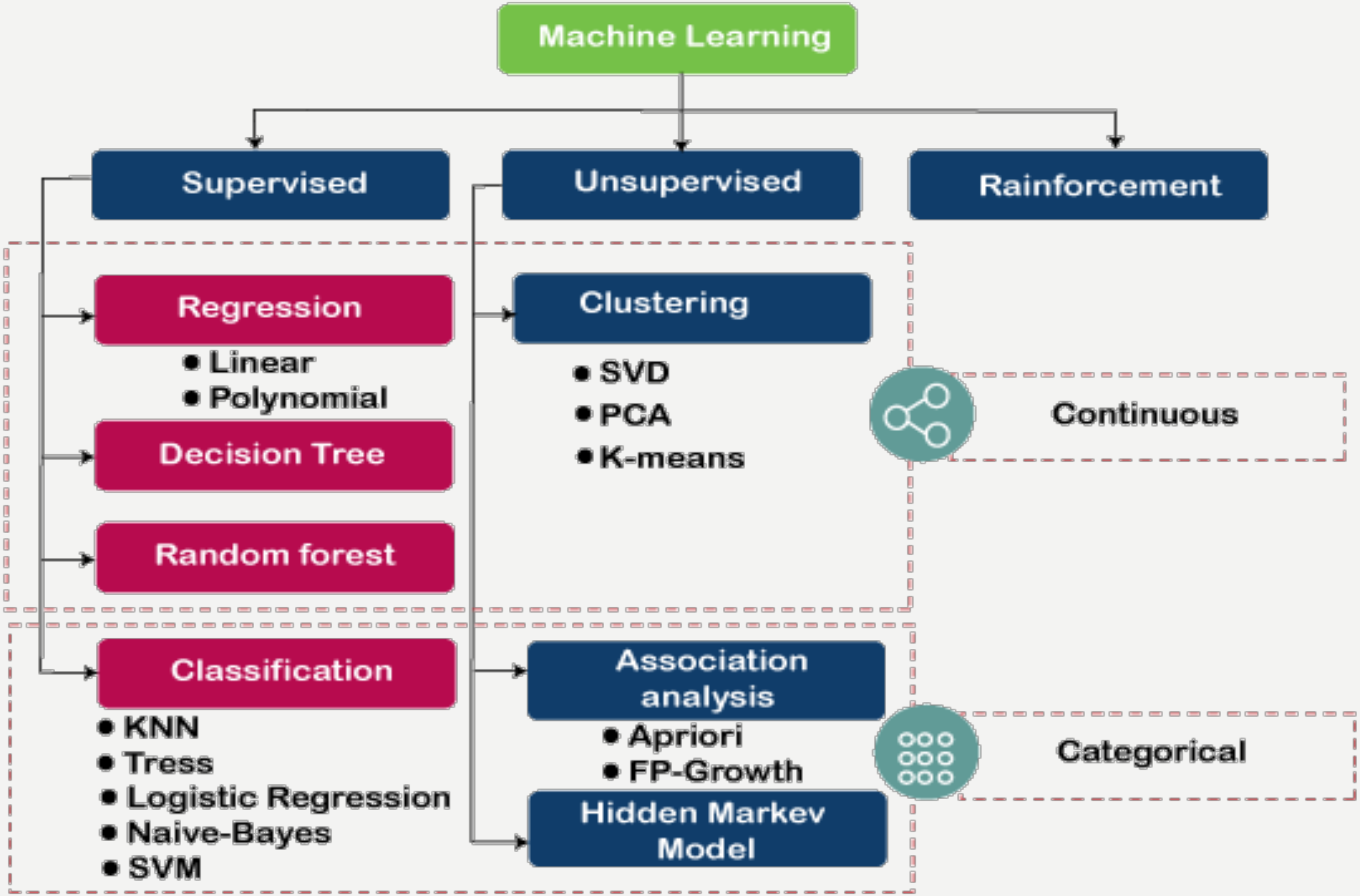
- Training:

The training is based upon the input,

The model will return a state and the user will decide to reward or punish the model based on its output.

The model keeps continues to learn.

The best solution is decided based on the maximum reward.



Machine Learning

Supervised

Unsupervised

Rainforcement

Regression

- Linear
- Polynomial

Decision Tree

Random forest

Classification

- KNN
- Tress
- Logistic Regression
- Naive-Bayes
- SVM

Clustering

- SVD
- PCA
- K-means

Association analysis

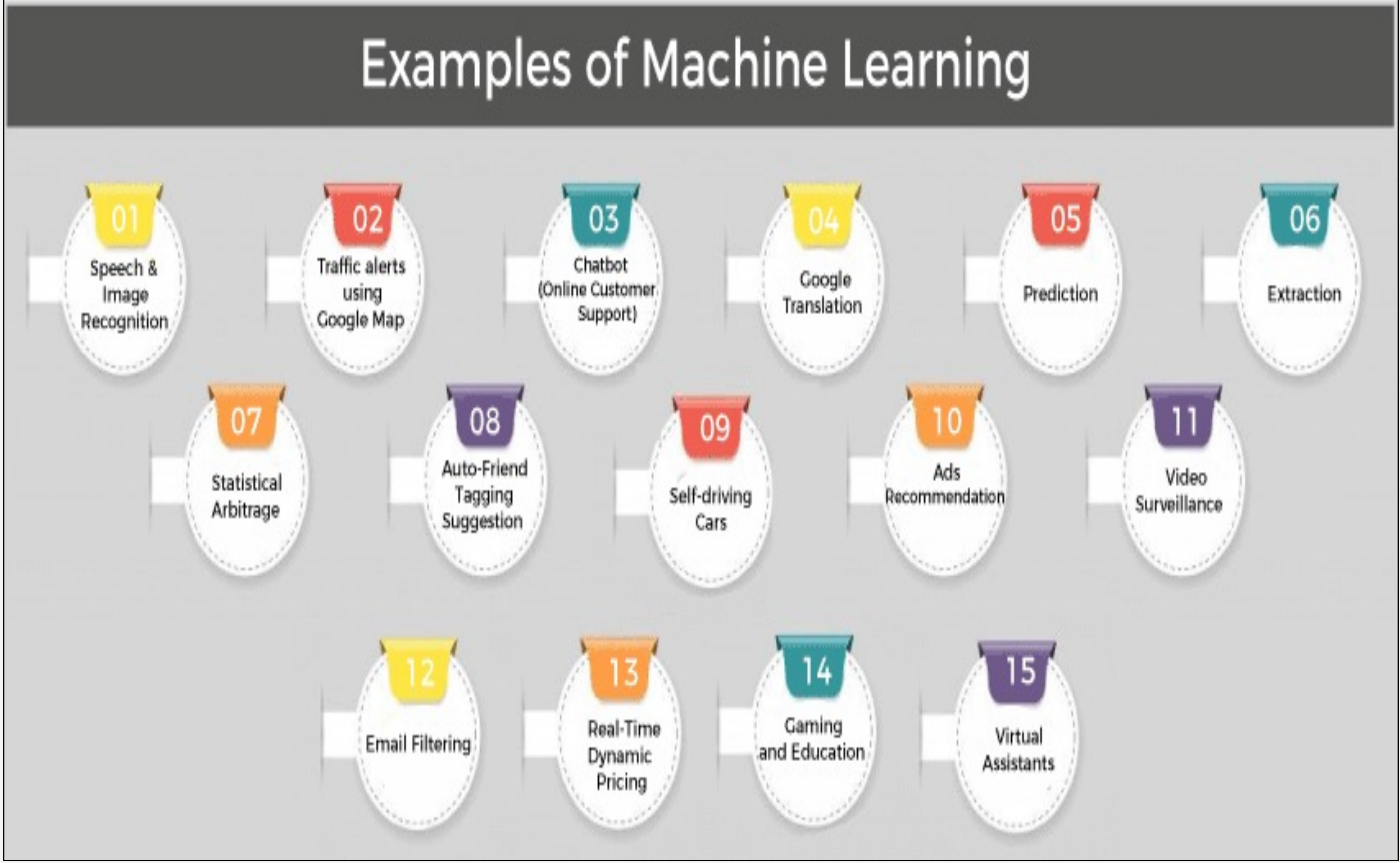
- Apriori
- FP-Growth

Hidden Markev Model

Continuous

Categorical

# Examples of Machine Learning



01 Speech & Image Recognition

02 Traffic alerts using Google Map

03 Chatbot (Online Customer Support)

04 Google Translation

05 Prediction

06 Extraction

07 Statistical Arbitrage

08 Auto-Friend Tagging Suggestion

09 Self-driving Cars

10 Ads Recommendation

11 Video Surveillance

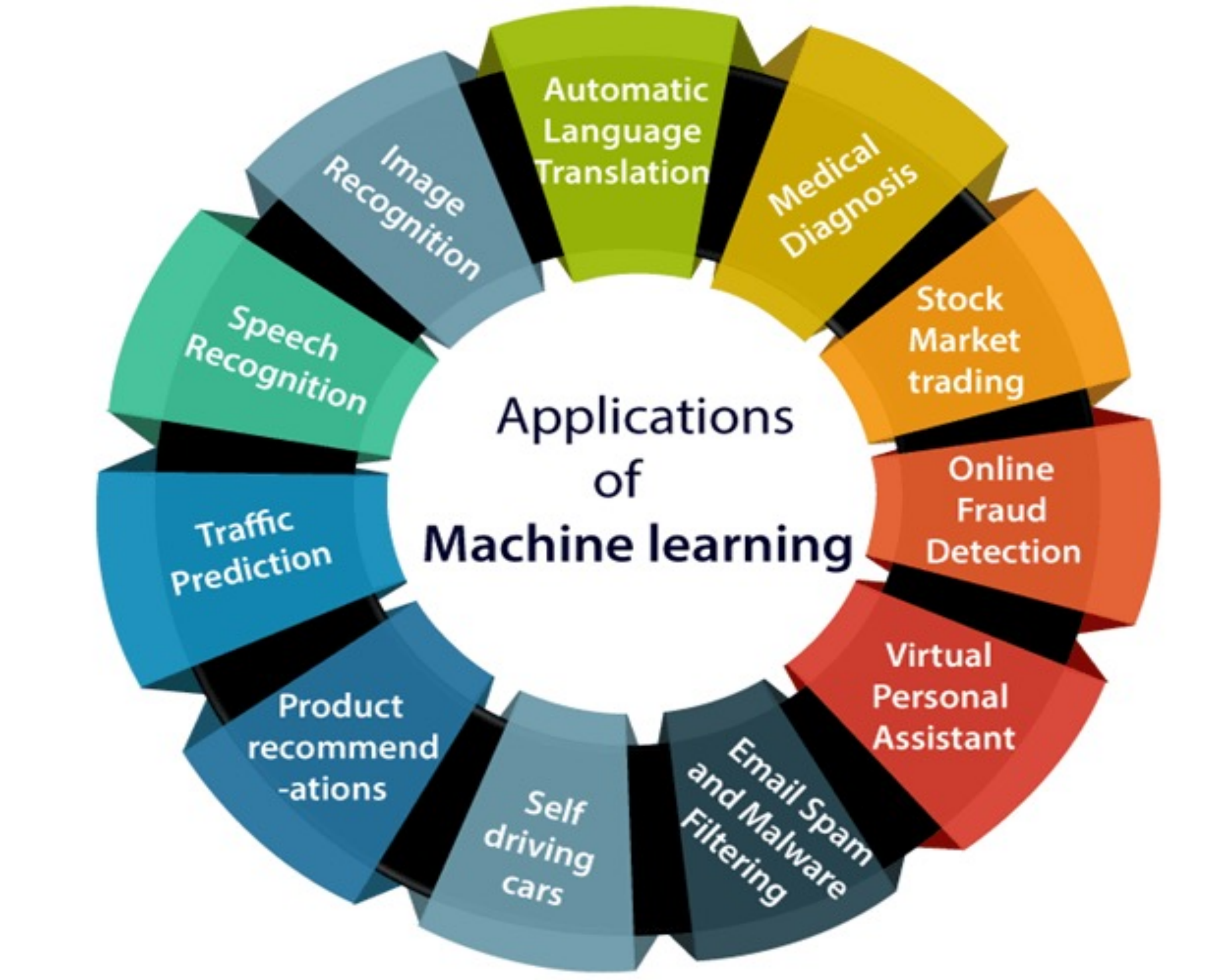
12 Email Filtering

13 Real-Time Dynamic Pricing

14 Gaming and Education

15 Virtual Assistants

# APPLICATIONS OF MACHINE LEARNING



# PREDICTION AND CLASSIFICATION

# PREDICTION

- Prediction is like something that may go to happen in the future.
- And just like that in prediction, we identify or predict the missing or unavailable data for a new observation based on the previous data that we have and based on the future assumptions.
- In prediction, the output is a continuous value.

the journal of
PREDICTION
MARKETS

# PREDICTION AND CLASSIFICATION

| Sr.No. | Prediction | Classification |
|---|---|---|
| 1. | Prediction is about predicting a missing/unknown element(continuous value) of a dataset. | Classification is about determining a (categorial) class (or label) for an element in a dataset. |
| 2. | Eg. We can think of prediction as predicting the correct treatment for a particular disease for an individual person. | Eg. Whereas the grouping of patients based on their medical records can be considered classification. |
| 3. | The model used to predict the unknown value is called a predictor. | The model used to classify the unknown value is called a classifier. |
| 4. | The predictor is constructed from a training set and its accuracy refers to how well it can estimate the value of new data. | A classifier is also constructed from a training set composed of the records of databases and their corresponding class names. |

# COMPARISON OF CLASSIFICATION AND PREDICTION METHODS

Few criteria used for comparing the methods of Classification and Prediction:

- **Accuracy:** Accuracy of the classifier can be referred to as the ability of the classifier to predicts the class label correctly**,** and the accuracy of the predictor can be referred to as how well a given predictor can estimate the unknown value.
- **Speed:** The speed of the method depends on the computational cost of generating and using the classifier/predictor.
- **Robustness:** Robustness is the ability to make correct predictions or classifications, in the context of data mining robustness is the ability of the classifier or predictor to make correct predictions from incoming unknown data.
- **Scalability:** Scalability is referring to an increase or decrease in performance of the classifier or predictor based on the given data.
- **Interpretability:** Interpretability can be referred to as how readily we can understand the reasoning behind predictions or classification made by the predictor or classifier.

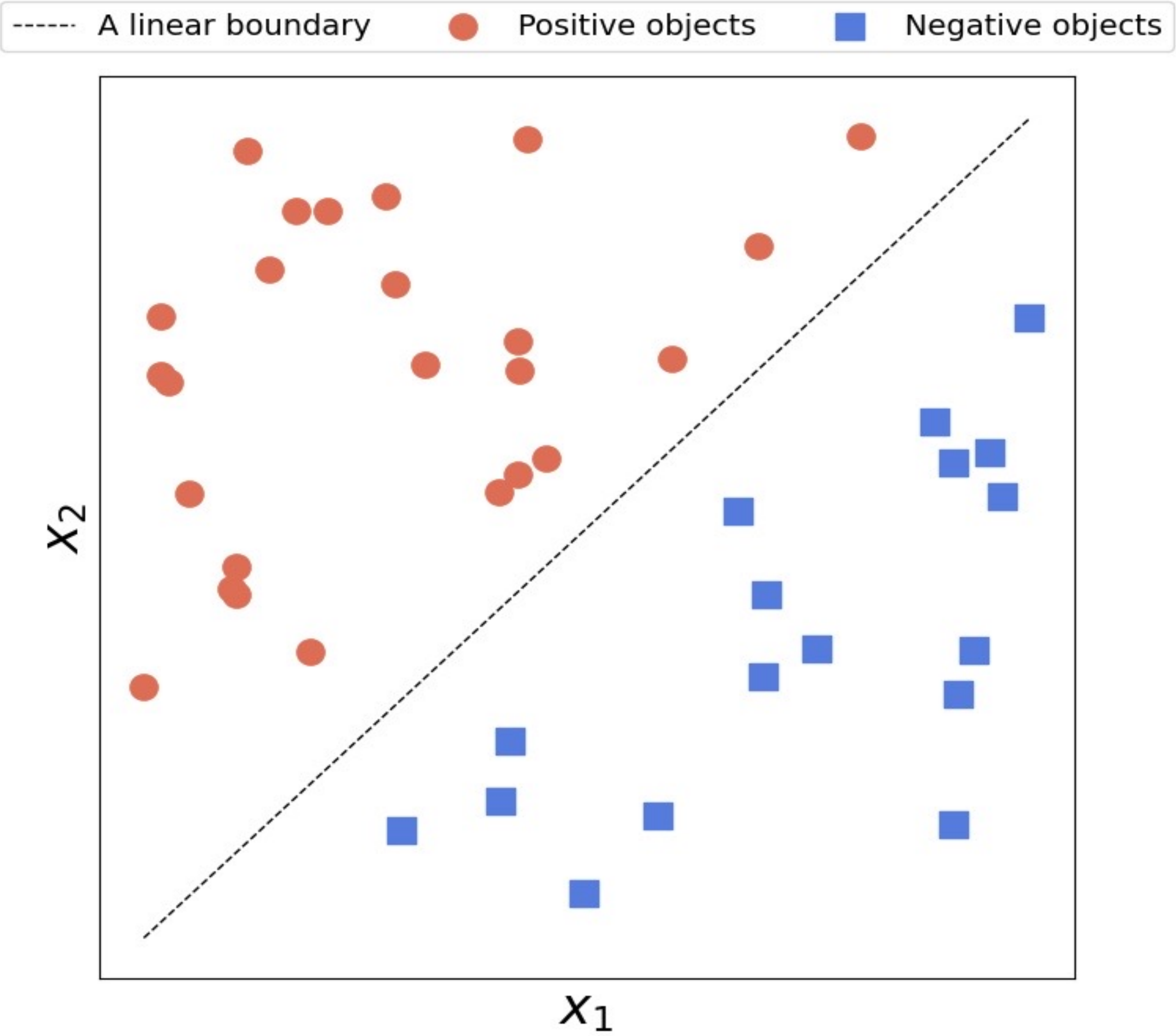
# DATA SEPARABILITY

# DATA SEPARABILITY

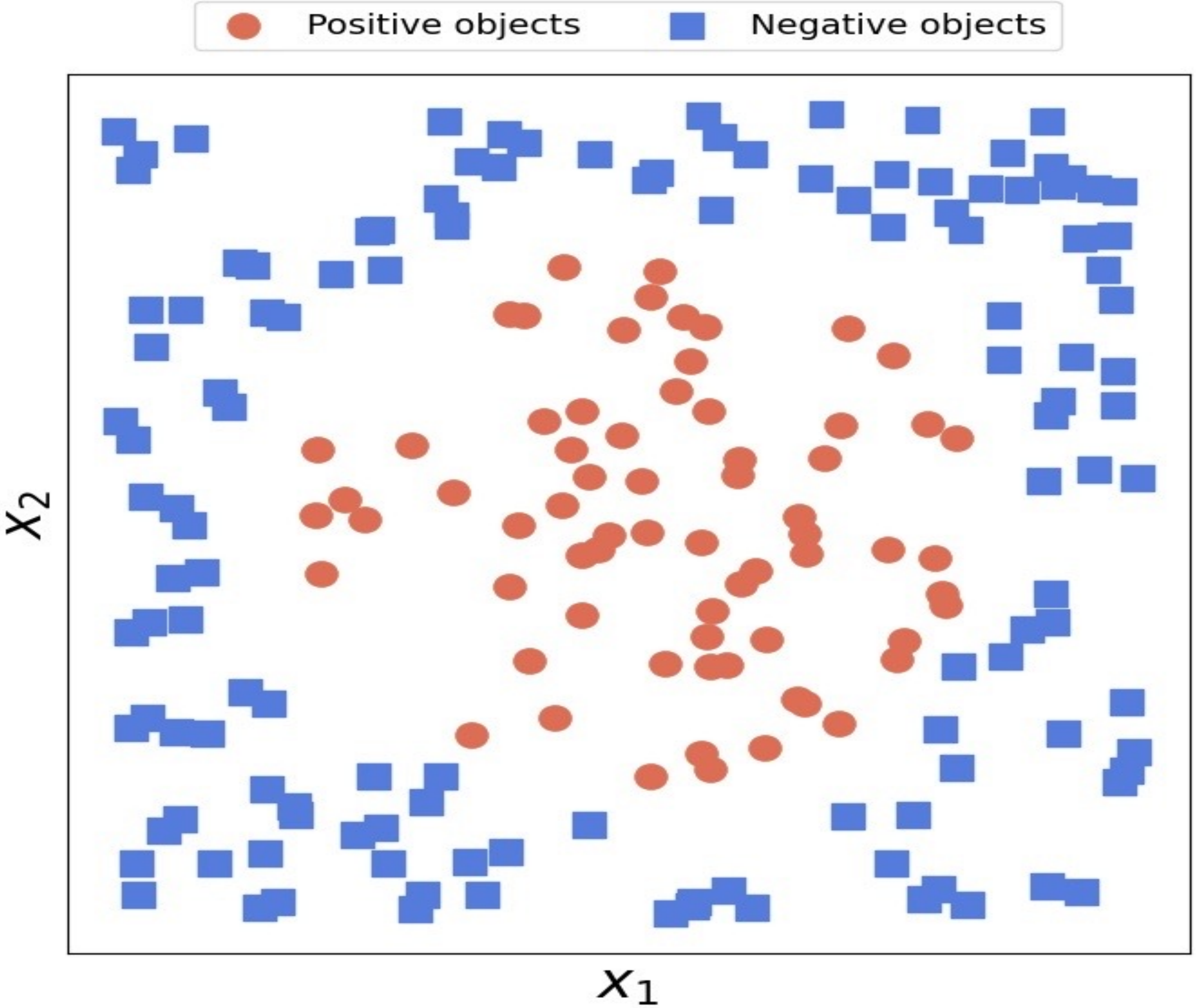
- The concept of separability applies to binary classification problems.
- In them, we have two classes: one positive and the other negative.
- We say they’re separable if there’s a classifier whose decision boundary separates the positive objects from the negative ones.
- If such a decision boundary is a linear function of the features, we say that the classes are linearly separable.
- Since we deal with labelled data, the objects in a dataset will be linearly separable if the classes in the feature space are too.
- We say a two-dimensional dataset is linearly separable if we can separate the positive from the negative objects with a straight line.
- It doesn’t matter if more than one such line exists. For linear separability, it’s sufficient to find only one

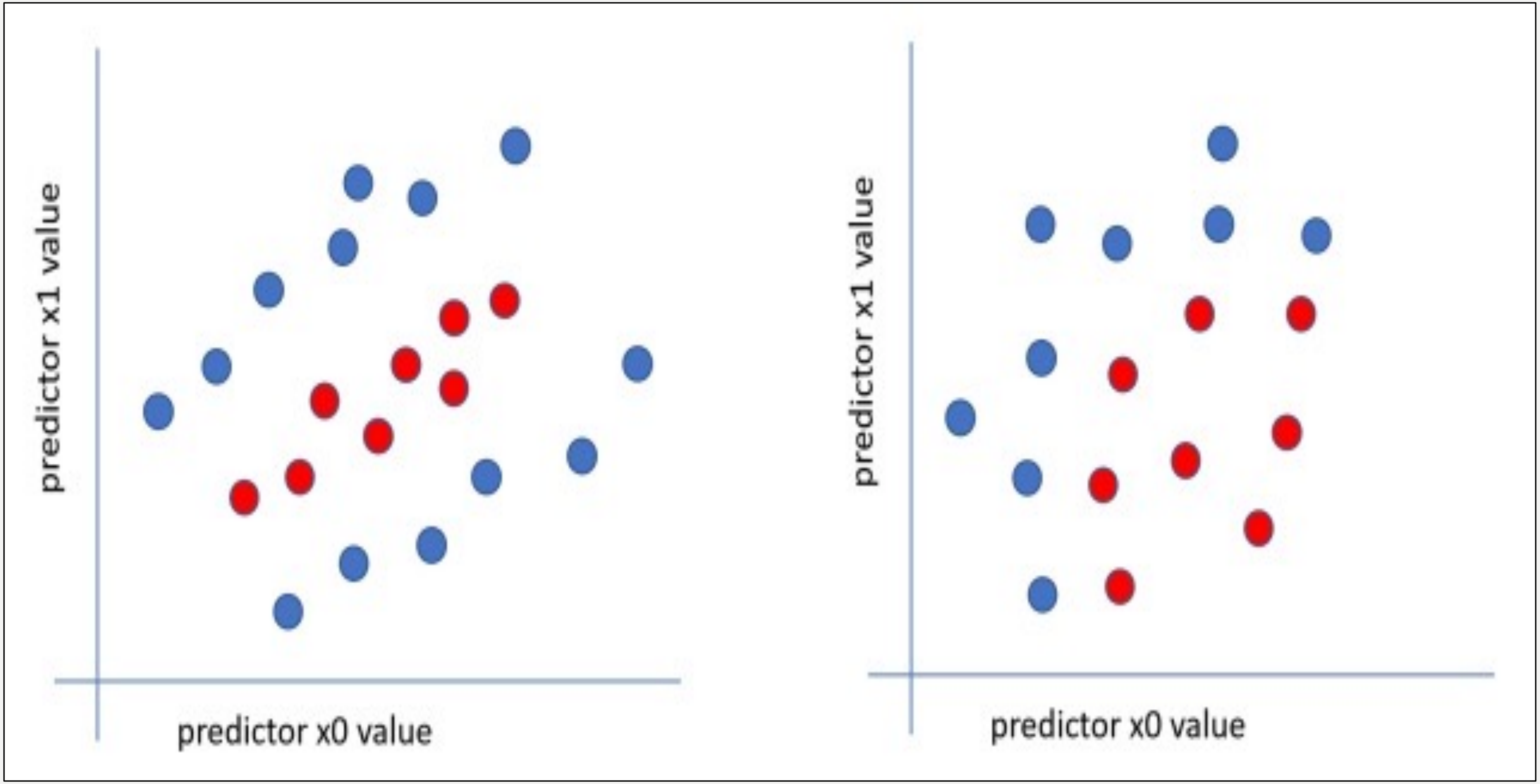
Linearly separable data is data that if graphed in two dimensions, can be separated by a straight line.

This data is linearly separable because there is a straight line from lower left to upper right that separates the red and blue data.



Conversely, no line can separate linearly inseparable 2D data





Neither of these two datasets is linearly separable. The data on the left needs two straight lines. The data on the right needs a curved line.

# DECISION BOUNDARY

# DECISION BOUNDARY

- A decision boundary is a line (in the case of two features), where all (or most) samples of one class are on one side of that line, and all samples of the other class are on the opposite side of the line.
- The line *separates* one class from the other.
- If you have more than two features, the decision boundary is not a line, but a (hyper)-plane in the dimension of your feature space.
- While training a classifier on a dataset, using a specific classification algorithm, it is required to define a set of hyper-planes, called Decision Boundary, that separates the data points into specific classes, where the algorithm switches from one class to another.
- On one side a decision boundary, a datapoints is more likely to be called as class A — on the other side of the boundary, it’s more likely to be called as class B.

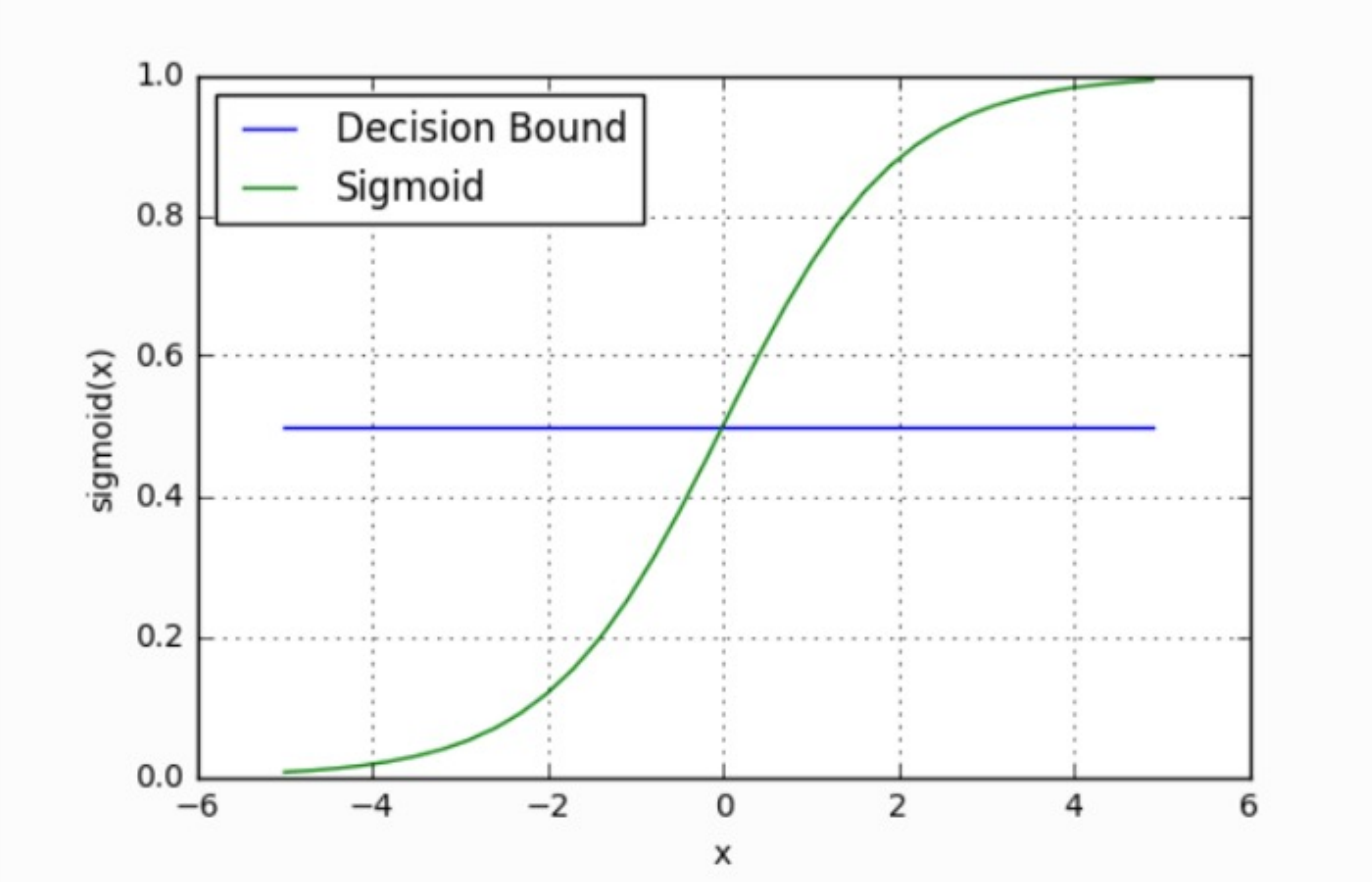
# DECISION BOUNDARY

- A decision boundary, is a surface that separates data points belonging to different class lables.
- Decision Boundaries are not only confined to just the data points that we have provided, but also they span through the entire feature space we trained on.
- The model can predict a value for any possible combination of inputs in our feature space.
- If the data we train on is not 'diverse', the overall topology of the model will *generalize poorly to new instances*.
- So, it is important to analyse all the models which can be best suitable for 'diverse' dataset, before using the model into production.
- Examining decision boundaries is a great way to learn how the training data we select affects performance and the ability for our model to generalize.
- Visualization of decision boundaries can illustrate how sensitive models are to each dataset, which is a great way to understand how specific algorithms work, and their limitations for specific datasets.

# EXAMPLE

- The goal  is to figure out some way to split the datapoints to have an accurate prediction of a given observation's class using the information present in the features.
- Let's suppose we define a line that describes the decision boundary. So, all of the points on one side of the boundary shall have all the datapoints belong to class A and all of the points on one side of the boundary shall have all the datapoints belong to class B.
- Our current prediction function returns a probability score between 0 and 1.

- In order to map this to a discrete class (A/B), we select a threshold value or tipping point above which we will classify values into class A and below which we classify values into class B.
- **p>=0.5,class=A**
- **p<=0.5,class=B**
- If our threshold was .5 and our prediction function returned .7, we would classify this observation belongs to class A. If our prediction was .2 we would classify the observation belongs to class B.
- ***So, line with 0.5 is called the decision boundary.***


Decision Bound
Sigmoid
1.0
0.8
0.6
0.4
0.2
0.0
sigmoid(x)
−6
−4
−2
0
2
4
6
x

# VALIDATION METHODS

# VALIDATION

- Validation techniques in machine learning are used to get the error rate of the ML model and evaluate  its accuracy and efficiency.
- The right validation techniques help to estimate unbiased generalized model performance and give a better understanding of how the model was trained.
- It helps to make sure that the machine learning model is accurately trained and that it outputs the right data and that the machine learning model's prediction is accurate when it is deployed to real-world scenarios.
- Models properly validated are robust enough to adapt to new scenarios in the real world.
- Validation catches problems before they become big problems and is a critical step in the implementation of any machine learning model.

# MODEL VALIDATION TECHNIQUES

- Train and Test Split or Holdout
- Resubstition
- Cross-Validation

  - K-Fold –Cross Validation

  - Leave-One-Out Cross-Validation (LOOCV)
- Random Subsampling
- Bootstrapping

# TRAIN AND TEST SPLIT OR HOLDOUT

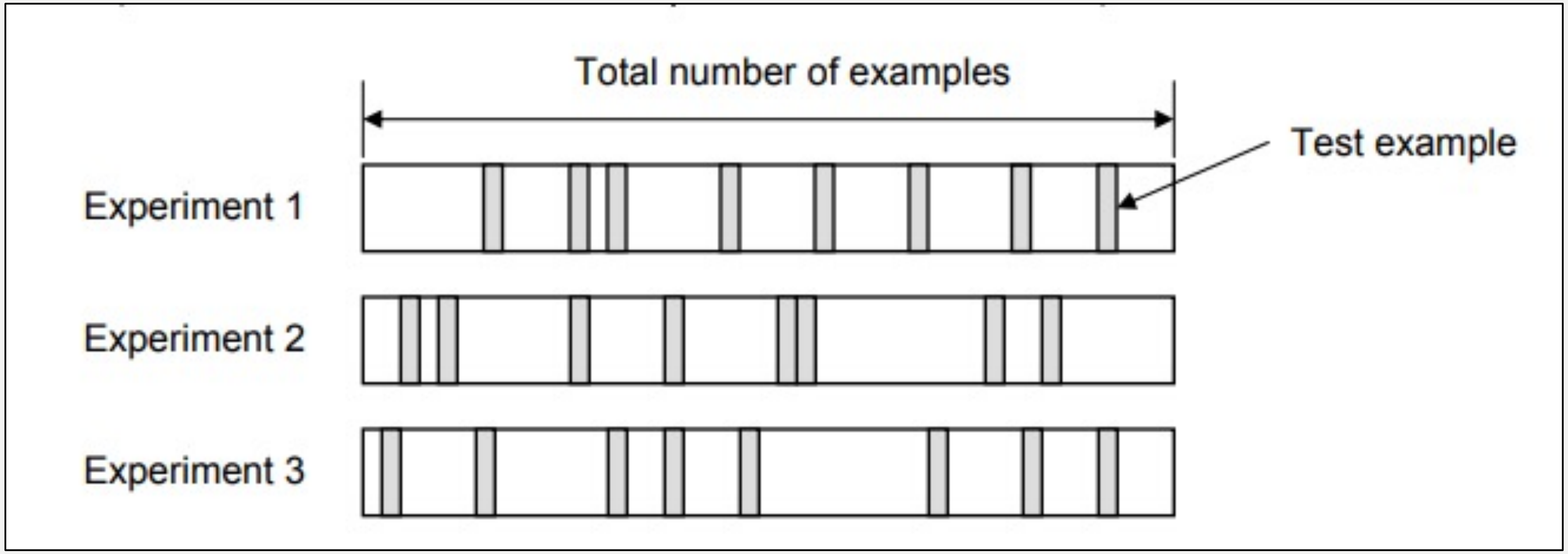
- This is the most common method to evaluate a classifier.
- In this method, the given data set is divided into two parts as a test and train set 20% and 80% respectively.
- The train set is used to train the data and the unseen test set is used to test its predictive power.
- To avoid the resubstitution error, the data is split into two different datasets labeled as a training and a testing dataset.
- This can be a 60/40 or 70/30 or 80/20 split.
- This technique is called the hold-out validation technique.
- In this case, there is a likelihood that uneven distribution of different classes of data is found in training and test dataset.
- To fix this, the training and test dataset is created with equal distribution of different classes of data. This process is called stratification.

# RESUBSTITUTION

- If all the data is used for training the model and the error rate is evaluated based on outcome vs. actual value from the same training data set, this error is called the *resubstitution error*.
- This technique is called the resubstitution validation technique.

# RANDOM SUBSAMPLING

- In this technique, multiple sets of data are randomly chosen from the dataset and combined to form a test dataset.
- The remaining data forms the training dataset.
- The following diagram represents the random subsampling validation technique.
- The error rate of the model is the average of the error rate of each iteration.



# CROSS VALIDATION

- There is always the need to test the stability of the model.
- It means based only on the training dataset; we can't fit our model on the training dataset.
- For this purpose, we reserve a particular sample of the dataset, which was not part of the training dataset.
- After that, we test our model on that sample before deployment, and this complete process comes under cross-validation.
- This is something different from the general train-test split.
- Hence the basic steps of cross-validations are:

➢ Reserve a subset of the dataset as a validation set.

➢ Provide the training to the model using the training dataset.

➢ Now, evaluate model performance using the validation set. If the model performs well with the validation set, perform the further step, else check for the issues.

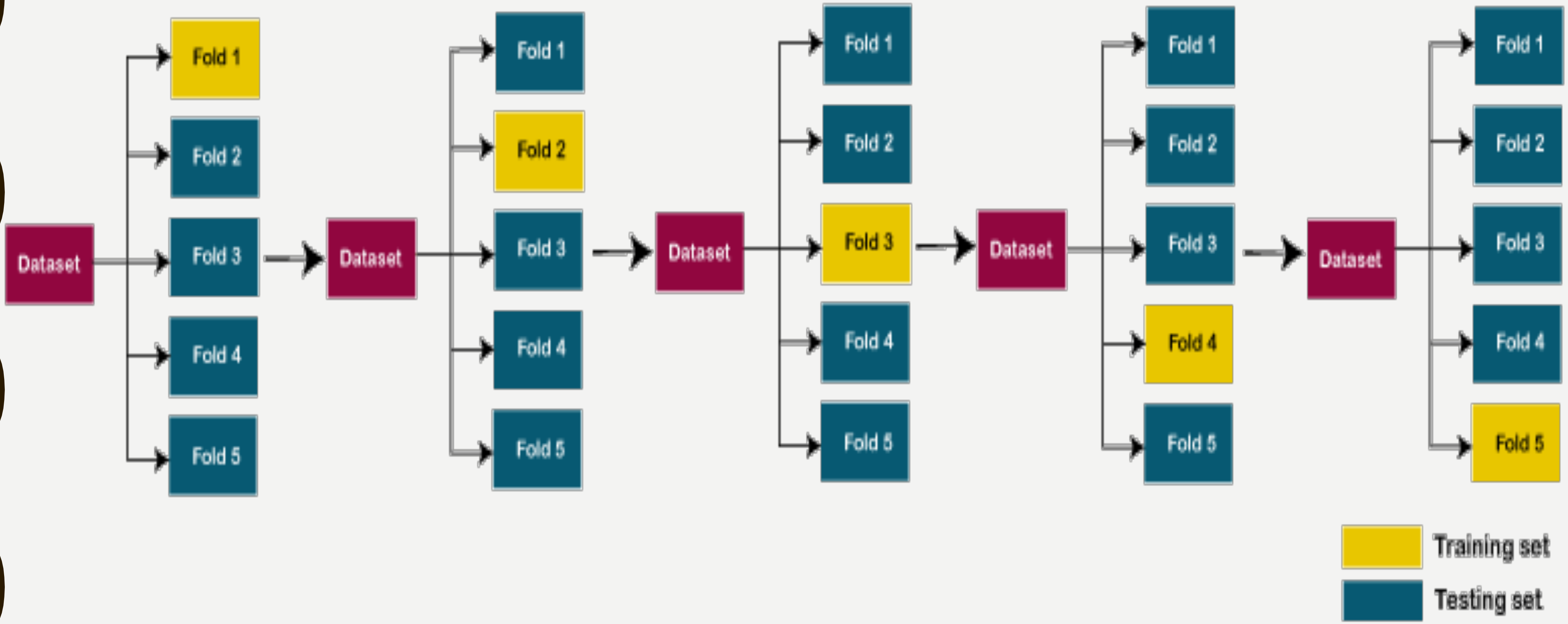
# K-FOLD CROSS-VALIDATION

- In this technique, k-1 folds are used for training and the remaining one is used for testing as shown in the picture given below.
- The advantage is that entire data is used for training and testing.
- The error rate of the model is average of the error rate of each iteration.
- This technique can also be called a form the repeated hold-out method.
- The error rate could be improved by using stratification technique.
- Over-fitting is the most common problem prevalent in most of the machine learning models.
- K-fold cross-validation can be conducted to verify if the model is over-fitted at all.
- In this method, the data set is randomly partitioned into **k mutually exclusive** subsets, each of which is of the same size.
- Out of these, one is kept for testing and others are used to train the model.
- The same process takes place for all k folds.

# K-FOLD CROSS-VALIDATION

- In this technique, k-1 folds are used for training and the remaining one is used for testing as shown in the picture given below.
- The advantage is that entire data is used for training and testing.
- The error rate of the model is average of the error rate of each iteration.
- This technique can also be called a form the repeated hold-out method. The error rate could be improved by using stratification technique.
- Over-fitting is the most common problem prevalent in most of the machine learning models. K-fold cross-validation can be conducted to verify if the model is over-fitted at all.

# K-FOLD CROSS-VALIDATION



Dataset

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

Dataset

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

Dataset

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

Dataset

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

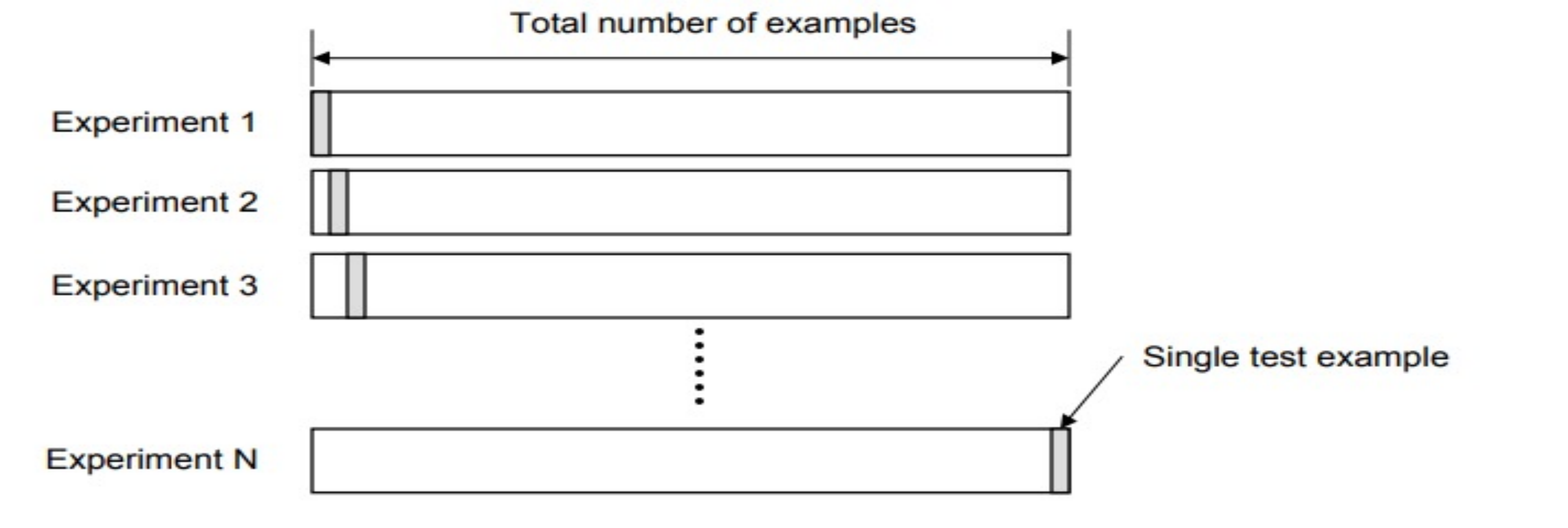
Dataset

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

🟨 Training set
🟦 Testing set

# LEAVE-ONE-OUT CROSS-VALIDATION (LOOCV)

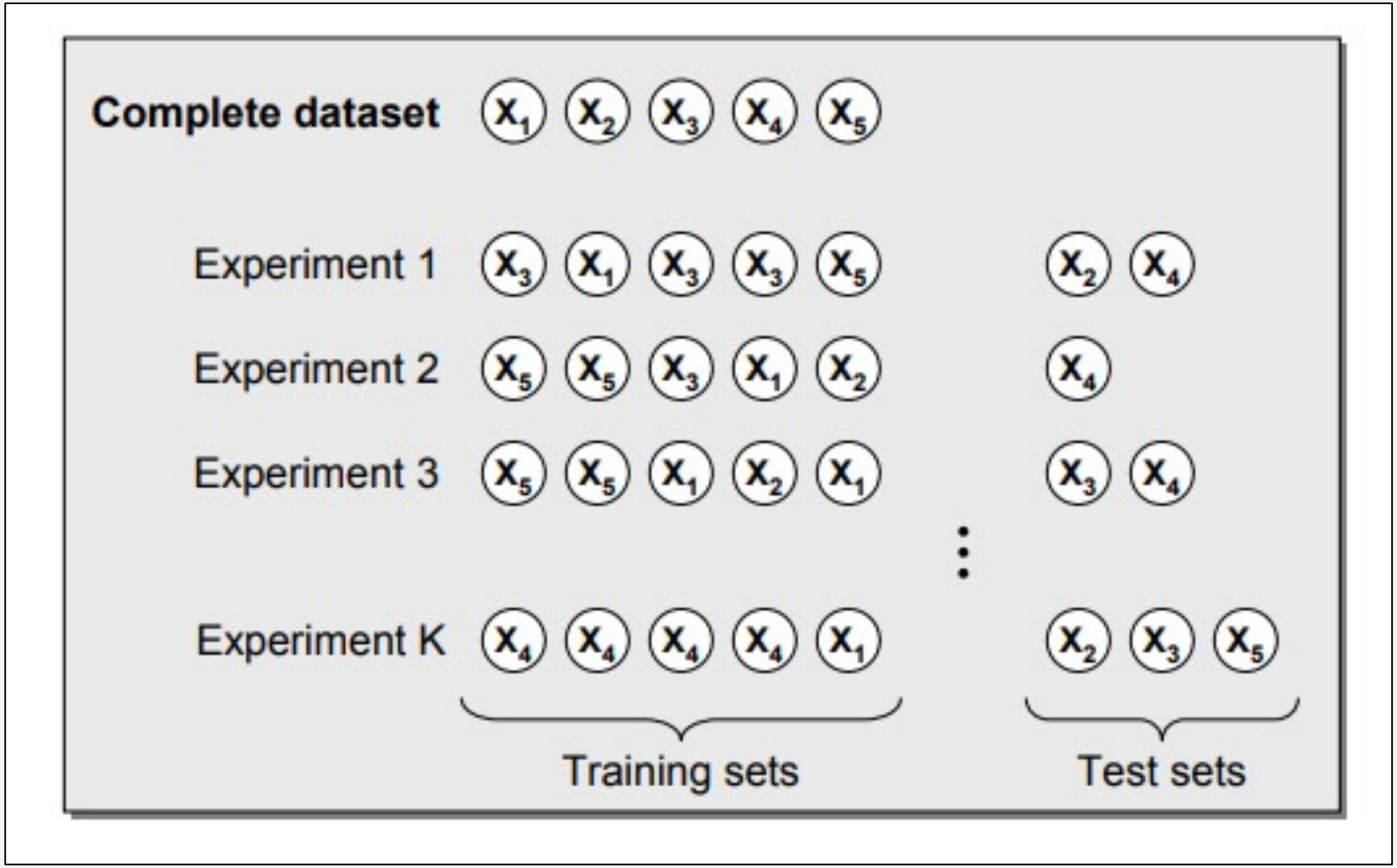
- In this technique, all of the data except one record is used for training and one record is used for testing.
- This process is repeated for N times if there are N records.
- The advantage is that entire data is used for training and testing.
- The error rate of the model is average of the error rate of each iteration.
- The following diagram represents the LOOCV validation technique.



# BOOTSTRAPPING

- In this technique, the training dataset is randomly selected with replacement.
- The remaining examples that were not selected for training are used for testing.
- Unlike K-fold cross-validation, the value is likely to change from fold-to-fold.
- The error rate of the model is average of the error rate of each iteration.
- The diagram in next slide represents the same.

# BOOTSTRAPPING



Complete dataset $X_1$ $X_2$ $X_3$ $X_4$ $X_5$

Experiment 1 $X_3$ $X_1$ $X_3$ $X_3$ $X_5$ $X_2$ $X_4$

Experiment 2 $X_5$ $X_5$ $X_3$ $X_1$ $X_2$ $X_4$

Experiment 3 $X_5$ $X_5$ $X_1$ $X_2$ $X_1$ $X_3$ $X_4$

⋮

Experiment K $X_4$ $X_4$ $X_4$ $X_4$ $X_1$ $X_2$ $X_3$ $X_5$

Training sets

Test sets

# ASSESSMENT METRICS

# ASSESSMENT METRICS

- Confusion matrix
- Sensitivity
- Specificity
- Accuracy

# CONFUSION MATRIX

- The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data.
- It can only be determined if the true values for test data are known.
- The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**.
- Some features of Confusion matrix are given below:

➢ For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.

➢ The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.

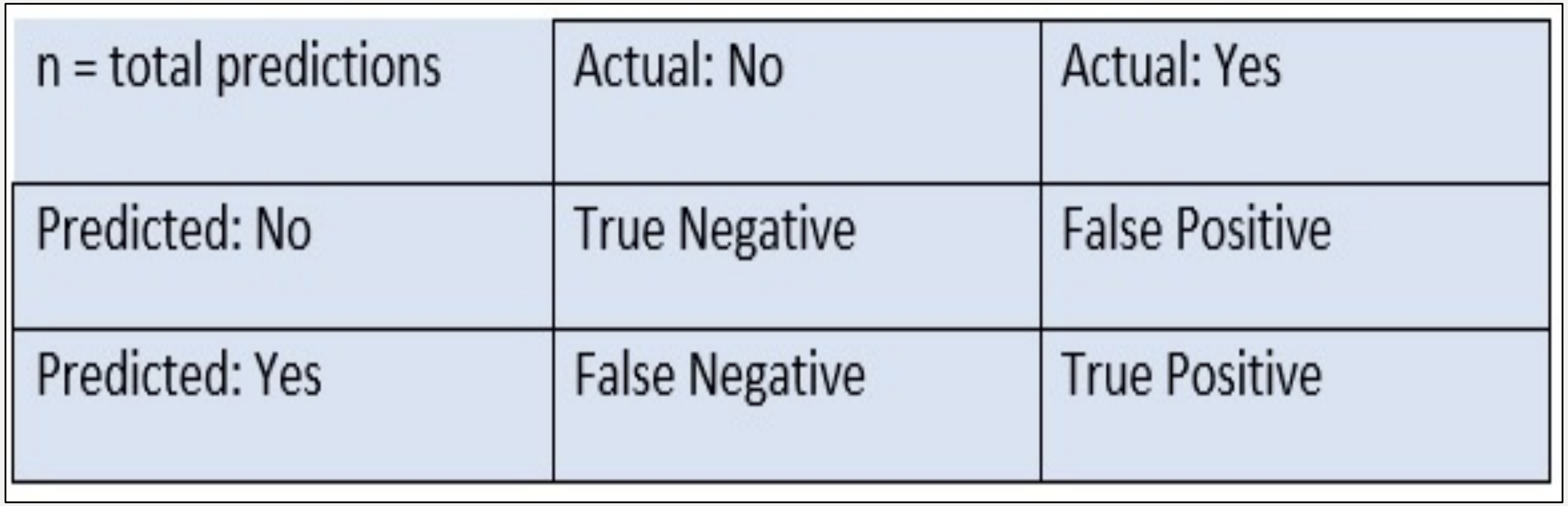
➢ Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

•

# CONFUSION MATRIX

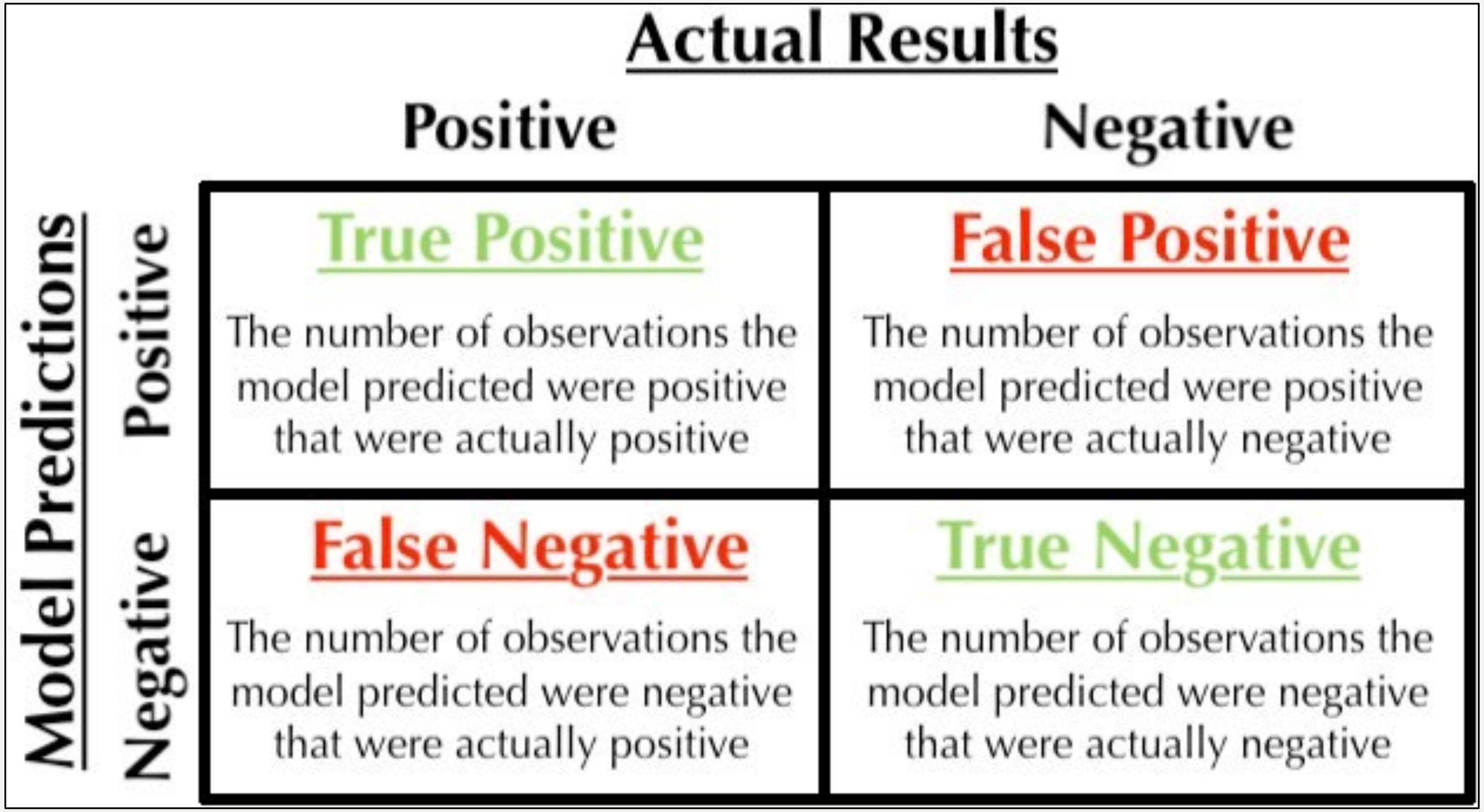
The table has the following cases:

- **True Negative:** Model has given prediction No, and the real or actual value was also No.
- **True Positive:** The model has predicted yes, and the actual value was also true.
- **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
- **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**



| n = total predictions | Actual: No | Actual: Yes |
|---|---|---|
| Predicted: No | True Negative | False Positive |
| Predicted: Yes | False Negative | True Positive |

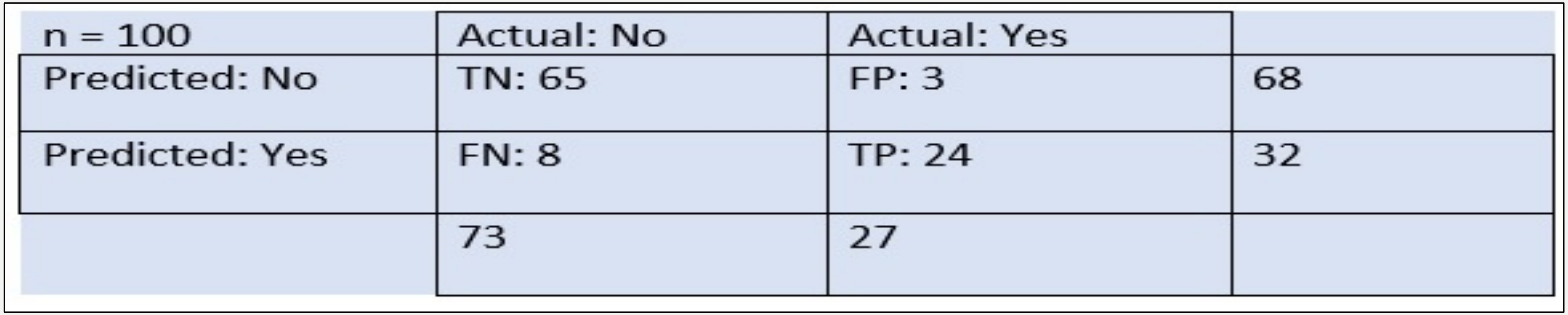
# CONFUSION MATRIX



| | | Actual Results | |
|---|---|---|---|
| | | Positive | Negative |
| Model Predictions | Positive | True Positive<br>The number of observations the model predicted were positive that were actually positive | False Positive<br>The number of observations the model predicted were positive that were actually negative |
| | Negative | False Negative<br>The number of observations the model predicted were negative that were actually positive | True Negative<br>The number of observations the model predicted were negative that were actually negative |

# EXAMPLE

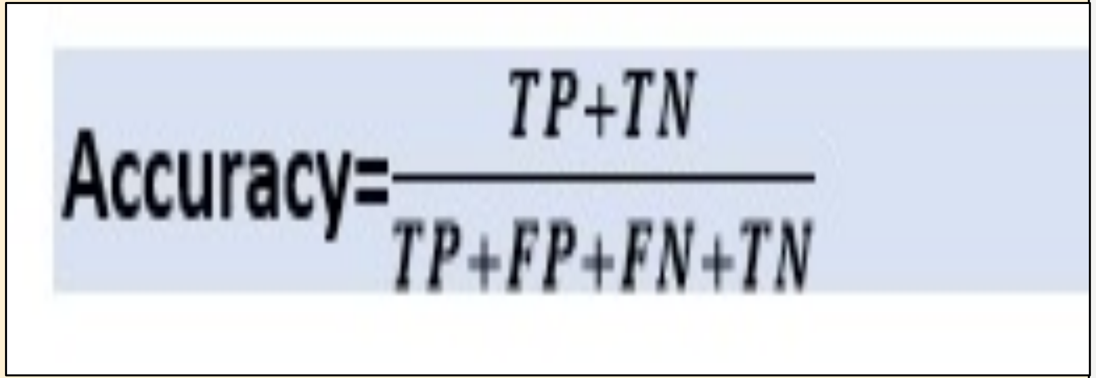
- We can understand the confusion matrix using an example.
- Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. So, the confusion matrix for this is given as in the table.
- From the above example, we can conclude that:
- The table is given for the two-class classifier, which has two predictions "Yes" and "NO." Here, Yes defines that patient has the disease, and No defines that patient does not has that disease.
- The classifier has made a total of **100 predictions**. Out of 100 predictions, **89 are true predictions**, and **11 are incorrect predictions**.
- The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.



| n = 100 | Actual: No | Actual: Yes | |
|---|---|---|---|
| Predicted: No | TN: 65 | FP: 3 | 68 |
| Predicted: Yes | FN: 8 | TP: 24 | 32 |
| | 73 | 27 | |

# CALCULATIONS USING CONFUSION MATRIX

- We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are as follows:
- Accuracy
- Precision
- Recall



$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN}$$

**Classification Accuracy**

- It is one of the important parameters to determine the accuracy of the classification problems.
- It defines how often the model predicts the correct output
- It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers.
  - Accuracy is a ratio of correctly predicted observation to the total observations
  - True Positive:The number of correct predictions that the occurrence is positive.
  - True Negative: Number of correct predictions that the occurrence is negative.

# CALCULATIONS USING CONFUSION MATRIX

**Precision**

precision = (TP) / (TP+FP)

TP is the number of true positives, and FP is the number of false positives.

- A trivial way to have perfect precision is to make one single positive prediction and ensure it is correct (precision = 1/1 = 100%).
- This would not be very useful since the classifier would ignore all but one positive instance.
- The above equation can be explained by saying, from all the classes we have predicted as positive, how many are actually positive.
- Precision should be high as possible.

**Recall**

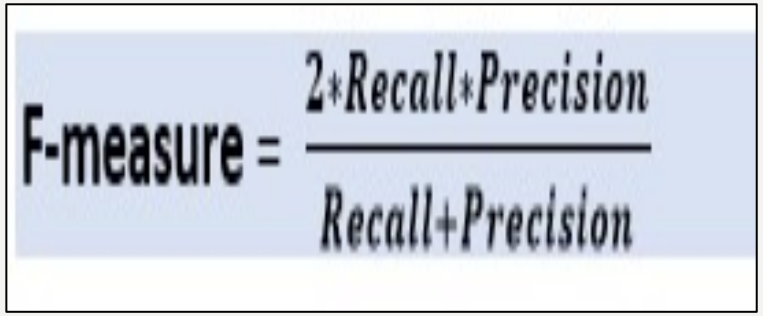
recall = (TP) / (TP+FN)

- The above equation can be explained by saying, from all the positive classes, how many we predicted correctly.
- Recall should be high as possible.

# CALCULATIONS USING CONFUSION MATRIX

<u>F-measure</u>

- If two models have low precision and high recall or vice versa, it is difficult to compare these models.
- So, for this purpose, we can use F-score.
- This score helps us to evaluate the recall and precision at the same time.
- The F-score is maximum if the recall is equal to the precision.
- It can be calculated using the below formula:



$$\text{F-measure} = \frac{2 * Recall * Precision}{Recall + Precision}$$

# CONFUSION MATRIX

- EXAMPLE(SLIDE NO 30)

# SENSITIVITY

- Sensitivity is a measure of how well a machine learning model can detect positive instances.
- It is also known as the **true positive rate (TPR) or recall.**
- Sensitivity is used to evaluate model performance because it allows us to see how many positive instances the model was able to correctly identify.
- A model with high sensitivity will have few false negatives, which means that it is missing a few of the positive instances.
- In other words, sensitivity measures the ability of a model to correctly identify positive examples.
- This is important because we want our models to be able to find all of the positive instances in order to make accurate predictions.
- **The sum of sensitivity (true positive rate) and false negative rate would be 1.**
- The higher the true positive rate, the better the model is in identifying the positive cases in the correct manner.

# EXAMPLE

- Let’s try and understand this with the model used for predicting whether a person is suffering from the disease.
- **Sensitivity** or **true positive rate** is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease.
- In other words, the person who is unhealthy (positive) actually got predicted as unhealthy.
- Mathematically, **sensitivity** or **true positive rate** can be calculated as the following:

**Sensitivity = (True Positive)/(True Positive + False Negative)**

- A high sensitivity means that the model is correctly identifying most of the positive results, while a low sensitivity means that the model is missing a lot of positive results.

- The following are the details in relation to True Positive and False Negative used in the sensitivity equation.
- **True Positive:** Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.
- **False Negative:** Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false-negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.
- The higher value of sensitivity would mean a higher value of the true positive and a lower value of false negative. The lower value of sensitivity would mean a lower value of the true positive and a higher value of false negative. For the healthcare and financial domain, models with high sensitivity will be desired.

# SPECIFICITY

- When sensitivity is used to evaluate model performance, it is often compared to specificity.
- Specificity measures the proportion of true negatives that are correctly identified by the model.
- This implies that there will be another proportion of actual negative which got predicted as positive and could be termed as **false positives**.
- This proportion could also be called a **True Negative Rate (TNR)**. **The sum of specificity (true negative rate) and false positive rate would always be 1.**
- High specificity means that the model is correctly identifying most of the negative results, while a low specificity means that the model is mislabeling a lot of negative results as positive.

# EXAMPLE

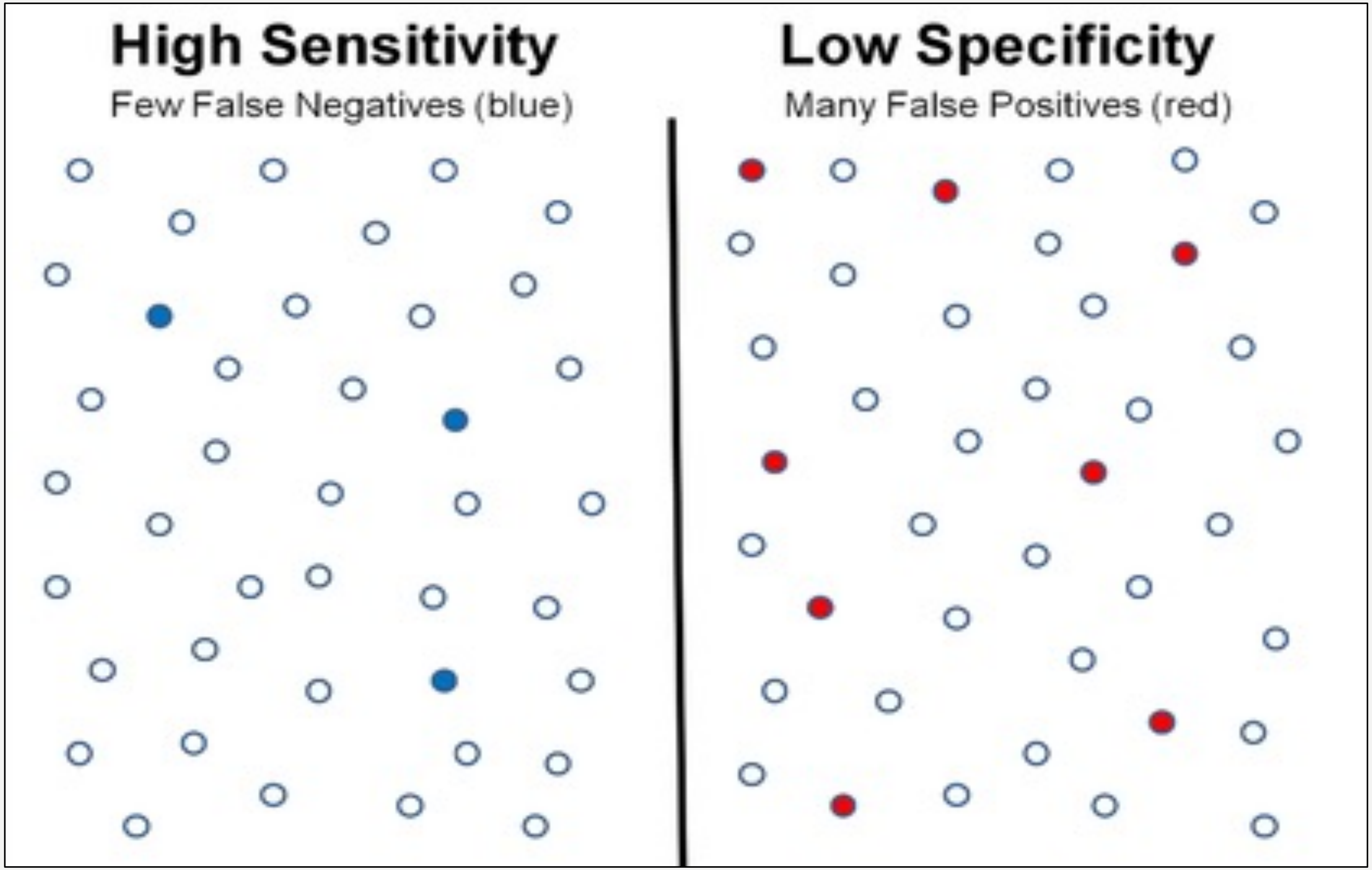
- Let’s try and understand this with the model used for predicting whether a person is suffering from the disease.
- Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease.
- In other words, the proportion of person who is healthy actually got predicted as healthy is specificity.
- Mathematically, specificity can be calculated as the following:

**Specificity = (True Negative)/(True Negative + False Positive)**

# EXAMPLE

- The following are the details in relation to True Negative and False Positive used in the specificity equation.
- **True Negative**: Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true negative represents the number of persons who are healthy and are predicted as healthy.
- **False Positive**: Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.
- Ideally, the model would be expected to have a very high specificity or true negative rate. The higher value of specificity would mean a higher value of true negative and a lower false-positive rate. The lower value of specificity would mean a lower value of the true negative and a higher value of false positive.

The diagram below represents a scenario of high sensitivity (low false negatives) and low specificity (high false positives).



# FORMULA

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

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

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

# NEXT CLASS

- NAÏVE BAYES CLASSIFICATION