MACHINE LEARNING

INTRODUCTION

MACHINE LEARNING - BASIC

- Machine Learning (ML) is basically that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do.
- In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method.
- The key focus of ML is to allow computer systems to learn from experience without being explicitly programmed or human intervention.

MACHINE LEARNING – AGE OF DATA

- We are living in the 'age of data' that is enriched with better computational power and more storage resources,.
- This data or information is increasing day by day, but the real challenge is to make sense of all the data.
- Businesses & organizations are trying to deal with it by building intelligent systems using the concepts and methodologies from Data science, Data Mining and Machine learning.
- Among them, machine learning is the most exciting field of computer science.
- Machine learning is the application and science of algorithms that provides sense to the data.

WHAT IS MACHINE LEARNING ? RECAP

- Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do.

- In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method.
- The main focus of ML is to allow computer systems learn from experience without being explicitly programmed or human intervention.

WHY & WHEN TO MAKE MACHINES LEARN ?

1) Lack of human expertise
   - The very first scenario in which we want a machine to learn and take data-driven decisions, can be the domain where there is a lack of human expertise.
   - The examples can be navigations in unknown territories or spatial planets.

WHY & WHEN TO MAKE MACHINES LEARN ?

2) Dynamic scenarios
   - There are some scenarios which are dynamic in nature i.e. they keep changing over time.
   - In case of these scenarios and behaviors, we want a machine to learn and take data-driven decisions.
   - Some of the examples can be network connectivity and availability of infrastructure in an organization.

WHY & WHEN TO MAKE MACHINES LEARN ?

3) Difficulty in translating expertise into computational tasks
   - There can be various domains in which humans have their expertise,; however, they are unable to translate this expertise into computational tasks. In such circumstances we want machine learning.
   - The examples can be the domains of speech recognition, cognitive tasks etc.

MACHINE LEARNING MODEL - DEFINITION

- We must need to understand the following formal definition of ML given by professor Mitchell −

- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, *if its performance at tasks in T, as measured by P, improves with experience E.*"

- The above definition is basically focusing on three parameters, also the main components of any learning algorithm, namely Task(T), Performance(P) and experience (E).

MACHINE LEARNING MODEL – SIMPLIFIED DEFINITION

- ML is a field of AI consisting of learning algorithms that −
  - Improve their performance (P)
  - At executing some task (T)
  - Over time with experience (E)

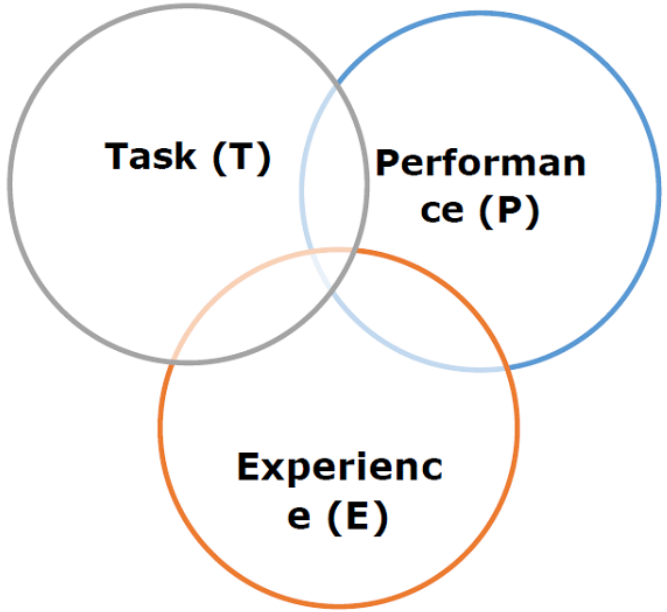


Diagram represents a Machine Learning Model

MACHINE LEARNING MODEL

1) Task(T)
   - From the perspective of problem, we may define the task T as the real-world problem to be solved.
   - The problem can be anything like finding best house price in a specific location or to find best marketing strategy etc.
   - On the other hand, machine learning, the definition of task is different because it is difficult to solve ML based tasks by conventional programming approach.
   - A task T is said to be a ML based task when it is based on the process and the system must follow for operating on data points.
   - The examples of ML based tasks are Classification, Regression, Structured annotation, Clustering, Transcription etc.

MACHINE LEARNING MODEL

2) Experience (E)
   - It is the knowledge gained from data points provided to the algorithm or model.
   - Once provided with the dataset, the model will run iteratively and will learn some inherent pattern.
   - **The learning thus acquired is called experience(E).**
   - Making an analogy with human learning, we can think of this situation as in which a human being is learning or gaining some experience from various attributes like situation, relationships etc.
   - Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience.
   - The experience gained by out ML model or algorithm will be used to solve the task T.

MACHINE LEARNING MODEL

- Performance (P)
  - An ML algorithm is supposed to perform task and gain experience with the passage of time.
  - The measure which tells whether ML algorithm is performing as per expectation or not is its performance (P).
  - P is basically a quantitative metric that tells how a model is performing the task, T, using its experience, E.
  - There are many metrics that help to understand the ML performance, such as accuracy score, F1 score, confusion matrix, precision, recall, sensitivity etc.

CHALLENGES IN MACHINES LEARNING

- The challenges that ML is facing currently are −

1. Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.
2. Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
3. Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
4. No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
5. Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.
6. Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
7. Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

APPLICATIONS OF MACHINES LEARNING

- Following are some real-world applications of ML −
  - Emotion analysis
  - Sentiment analysis
  - Error detection and prevention
  - Weather forecasting and prediction
  - Stock market analysis and forecasting
  - Speech synthesis
  - Speech recognition
  - Customer segmentation
  - Object recognition
  - Fraud detection
  - Fraud prevention
  - Recommendation of products to customer in online shopping.

MACHINE LEARNING – METHODS - BASED ON HUMAN SUPERVISION

1. **Supervised Learning**
   - Supervised learning algorithms or methods are the most commonly used ML algorithms.
   - This method or learning algorithm take the data sample i.e. the training data and its associated output i.e. labels or responses with each data samples during the training process.
   - The main objective of supervised learning algorithms is to learn an association between input data samples and corresponding outputs after performing multiple training data instances.

     For example, we have

     x: Input variables and

     Y: Output variable

     Now, apply an algorithm to learn the mapping function from the input to output as follows −

     $Y=f(x)$

MACHINE LEARNING – METHODS - BASED ON HUMAN SUPERVISION

- Now, the main objective would be to approximate the mapping function so well that even when we have new input data (x), we can easily predict the output variable (Y) for that new input data.
- It is called supervised because the whole process of learning can be thought as it is being supervised by a teacher or supervisor. Examples of supervised machine learning algorithms includes **Decision tree, Random Forest, KNN, Logistic Regression** etc.
  - Based on the ML tasks, supervised learning algorithms can be divided into following two broad classes −
    1. Classification
    2. Regression

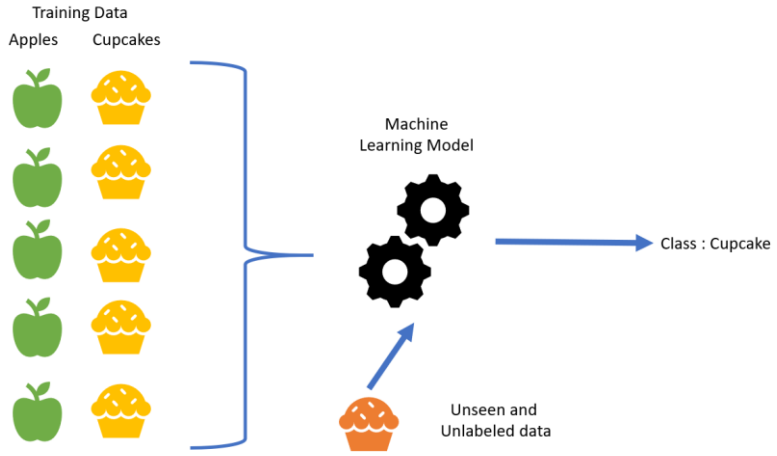
MACHINE LEARNING – METHODS - SUPERVISED LEARNING ALGORITHMS

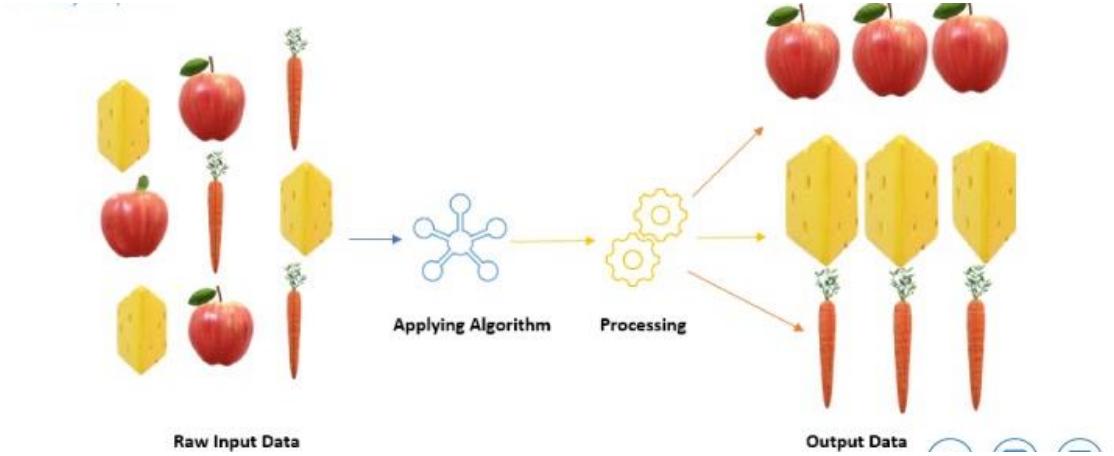
Classification

- The key objective of classification-based tasks is to predict categorical output labels or responses for the given input data.
- The output will be based on what the model has learned in training phase.
- As we know that the categorical output responses means unordered and discrete values, hence each output response will belong to a specific class or category.

MACHINE LEARNING – METHODS - - SUPERVISED LEARNING ALGORITHMS - EXAMPLES

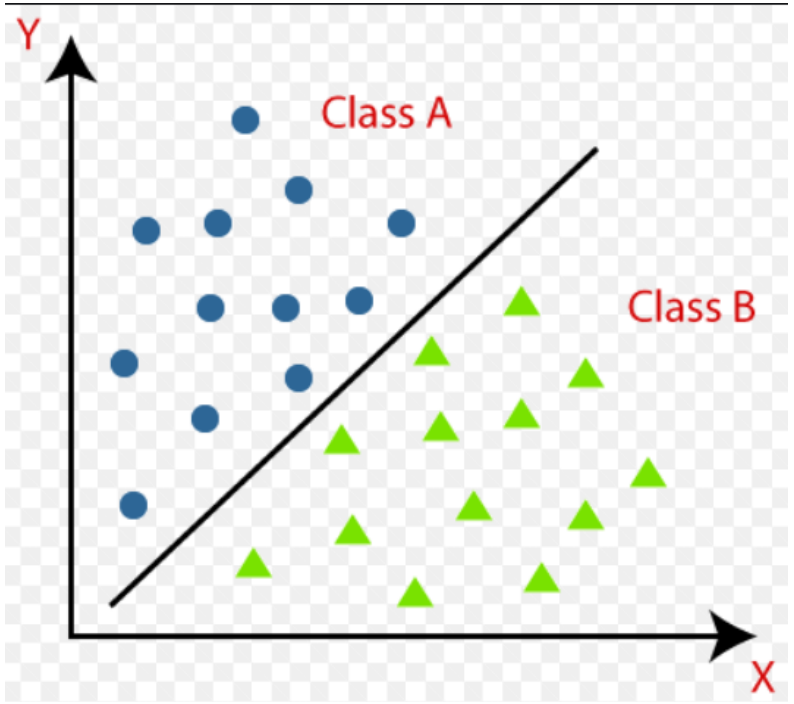
- **Classification**

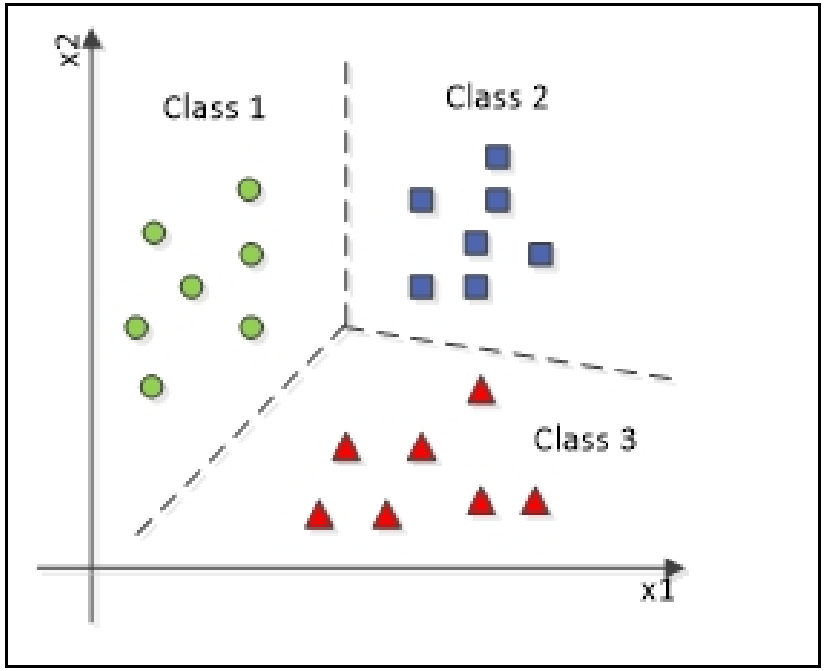




MACHINE LEARNING – METHODS - - SUPERVISED LEARNING ALGORITHMS - EXAMPLES

- **Classification**





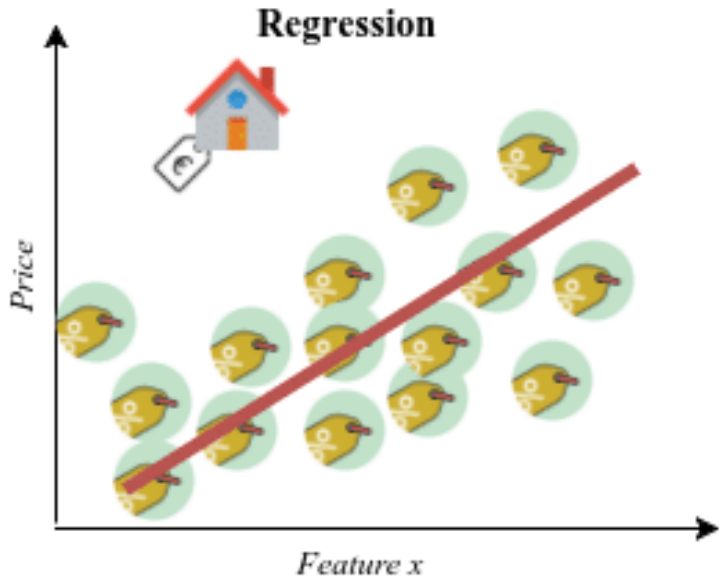
MACHINE LEARNING – METHODS - SUPERVISED LEARNING ALGORITHMS

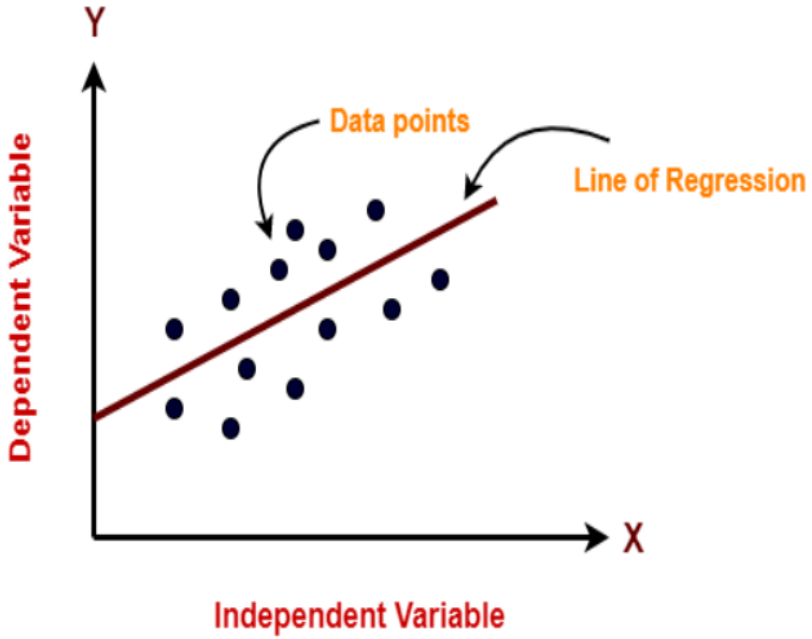
Regression

- The key objective of regression-based tasks is to predict output labels or responses which are continues numeric values, for the given input data.
- The output will be based on what the model has learned in its training phase.
- Basically, regression models use the input data features (independent variables) and their corresponding continuous numeric output values (dependent or outcome variables) to learn specific association between inputs and corresponding outputs.

MACHINE LEARNING – METHODS - SUPERVISED LEARNING ALGORITHMS - EXAMPLES

- **Regression**





MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING

2. Unsupervised Learning
   - It is opposite to supervised ML methods or algorithms which means in unsupervised machine learning algorithms we do not have any supervisor to provide any sort of guidance.
   - Unsupervised learning algorithms are handy in the scenario in which we do not have the liberty, like in supervised learning algorithms, of having pre-labeled training data and we want to extract useful pattern from input data.

MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING

For example, it can be understood as follows −

- Suppose we have −

  x: Input variables, then there would be no corresponding output variable and the algorithms need to discover the interesting pattern in data for learning.

  Examples of unsupervised machine learning algorithms includes K-means clustering, **K-nearest neighbors** etc.

- Based on the ML tasks, unsupervised learning algorithms can be divided into following broad classes −

  1. Clustering
  2. Association
  3. Dimensionality Reduction

MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING - CLUSTERING

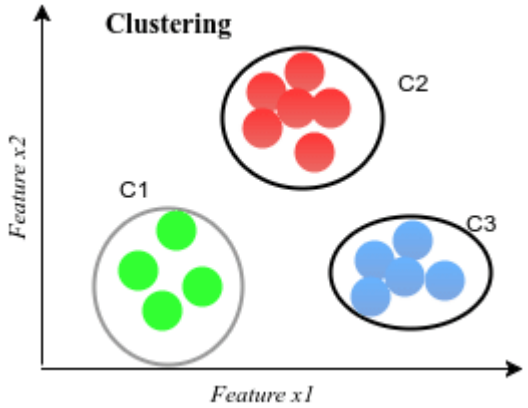
- **Clustering**
  - Clustering methods are one of the most useful unsupervised ML methods.
  - These algorithms used to find similarity as well as relationship patterns among data samples and then cluster those samples into groups having similarity based on features.

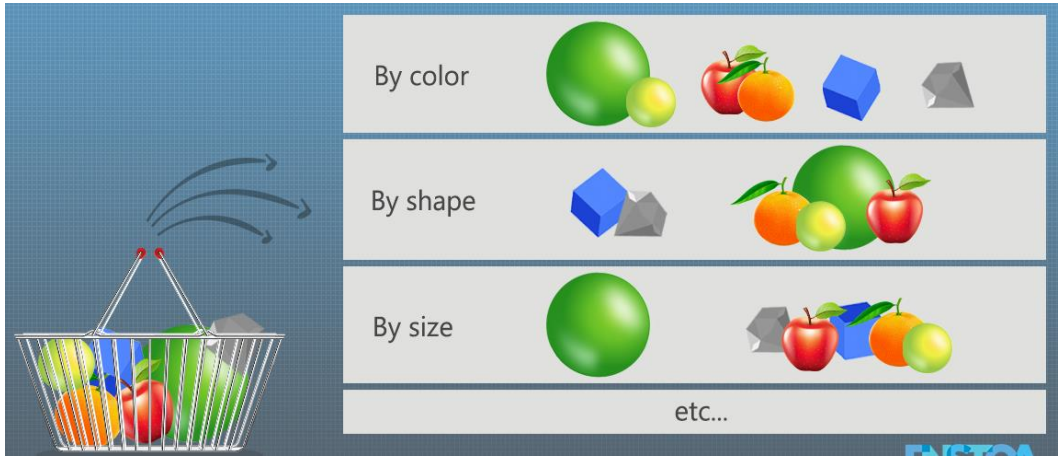
sample

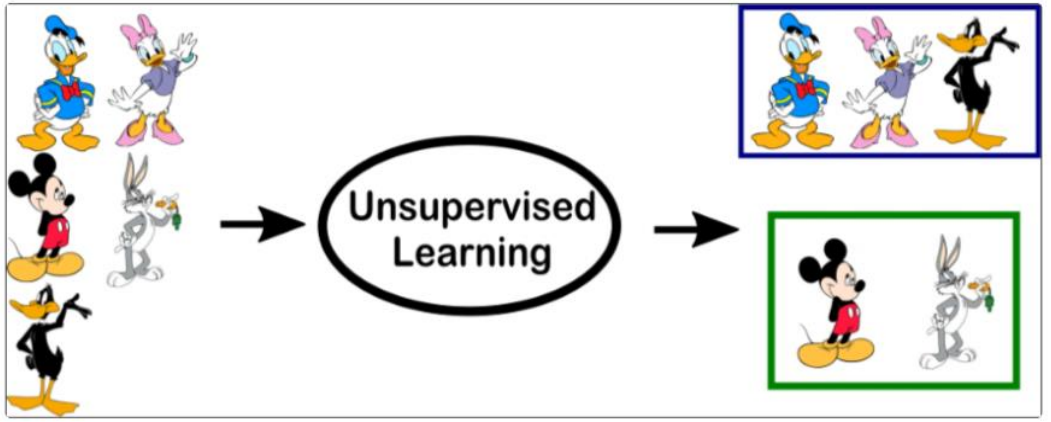
Cluster/group

MACHINE LEARNING – METHODS – UNSUPERVISED LEARNING - CLUSTERING - EXAMPLE

- **Clustering**
  - Clustering methods are one of the most useful unsupervised ML methods.
  - These algorithms used to find similarity as well as relationship patterns among data samples and then cluster those samples into groups having similarity based on features.





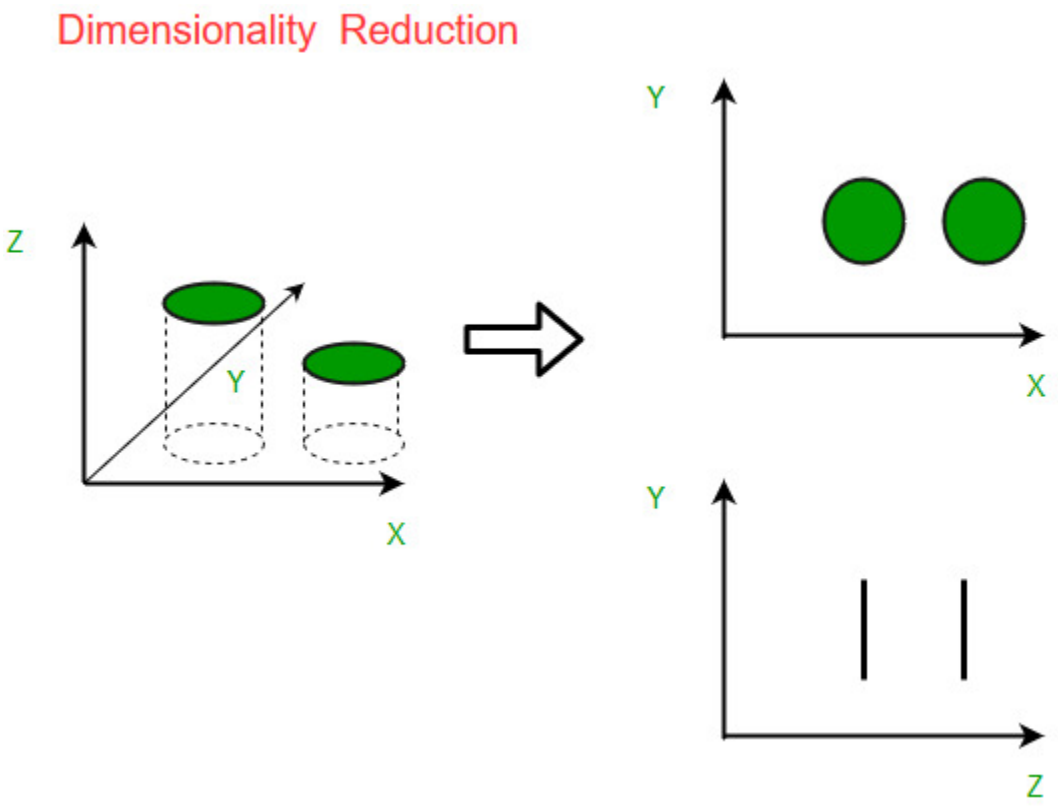


MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING - ASSOCIATION

- **Association**
  - Another useful unsupervised ML method is **Association** which is used to analyze large dataset to find patterns which further represents the interesting relationships between various items.
  - It is also termed as **Association Rule Mining** or **Market basket analysis** which is mainly used to analyze customer shopping patterns.

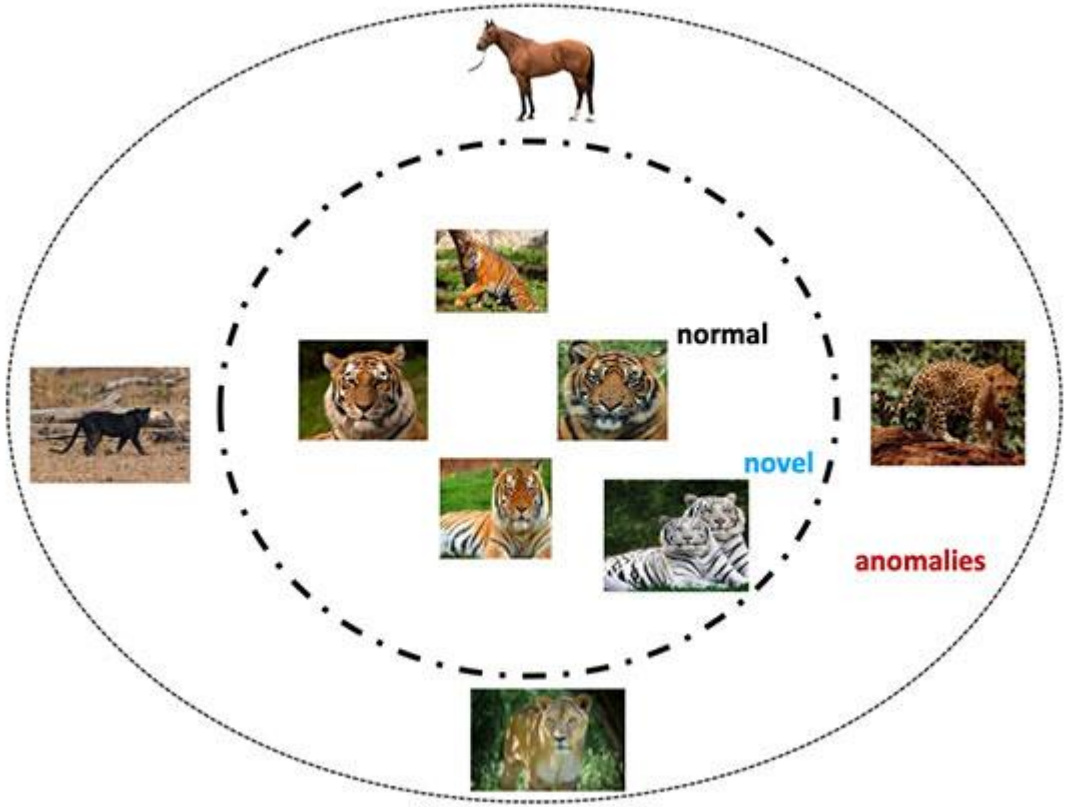
MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING - DIMENSIONALITY REDUCTION

- **Dimensionality Reduction**
  - This unsupervised ML method is used to reduce the number of feature variables for each data sample by selecting set of principal or representative features.
  - A question arises here is that why we need to reduce the dimensionality?
  - The reason behind is the problem of feature space complexity which arises when we start analyzing and extracting millions of features from data samples.
  - This problem generally refers to "curse of dimensionality". PCA (Principal Component Analysis), K-nearest neighbors and discriminant analysis are some of the popular algorithms for this purpose.



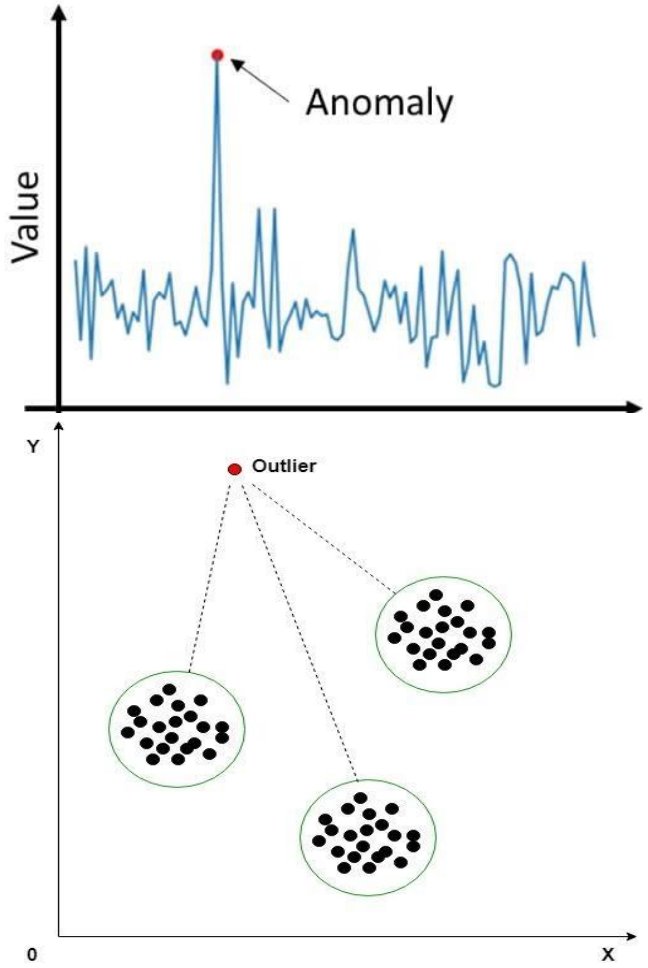
MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING - ANOMALY DETECTION

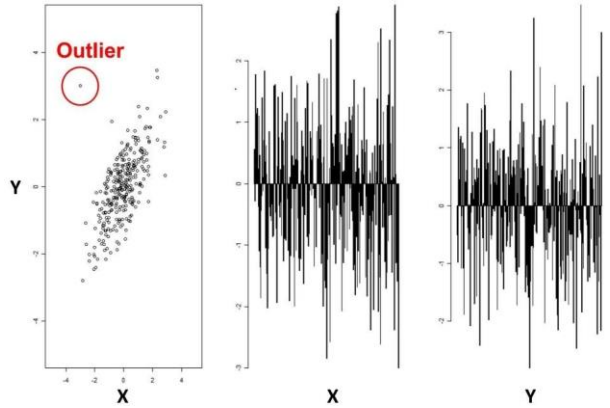
- **Anomaly Detection**
  - This unsupervised ML method is used to find out the occurrences of rare events or observations that generally do not occur.
  - By using the learned knowledge, anomaly detection methods would be able to differentiate between anomalous or a normal data point.
  - Some of the unsupervised algorithms like clustering, KNN can detect anomalies based on the data and its features.

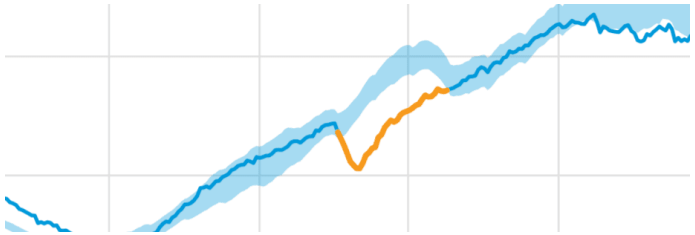


MACHINE LEARNING – METHODS - UNSUPERVISED LEARNING - ANOMALY DETECTION

- **Anomaly Detection**
  - This unsupervised ML method is used to find out the occurrences of rare events or observations that generally do not occur.
  - By using the learned knowledge, anomaly detection methods would be able to differentiate between anomalous or a normal data point.
  - Some of the unsupervised algorithms like clustering, KNN can detect anomalies based on the data and its features.

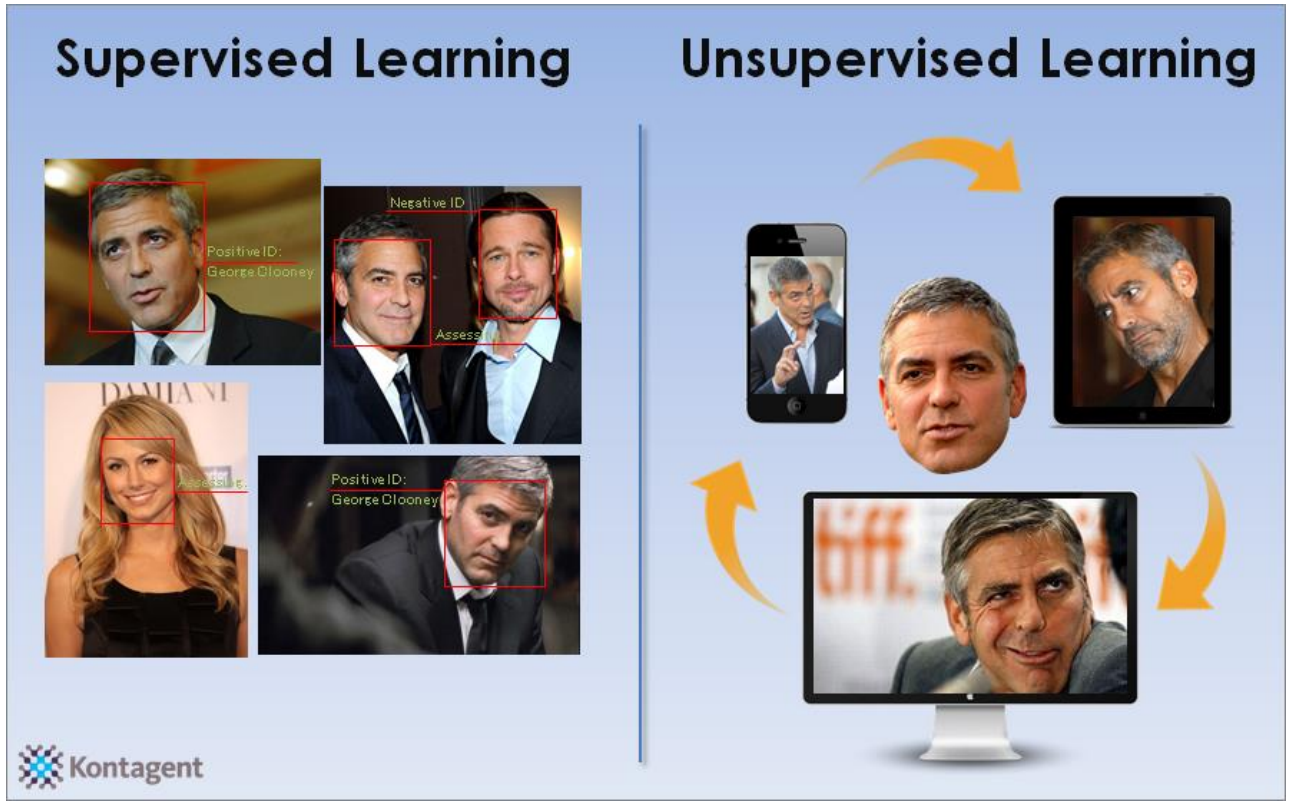




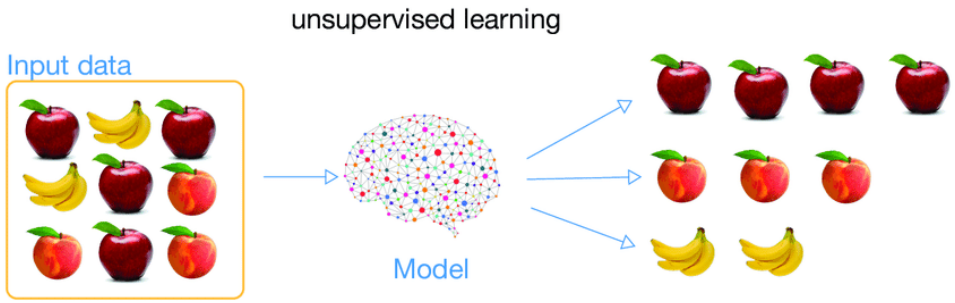


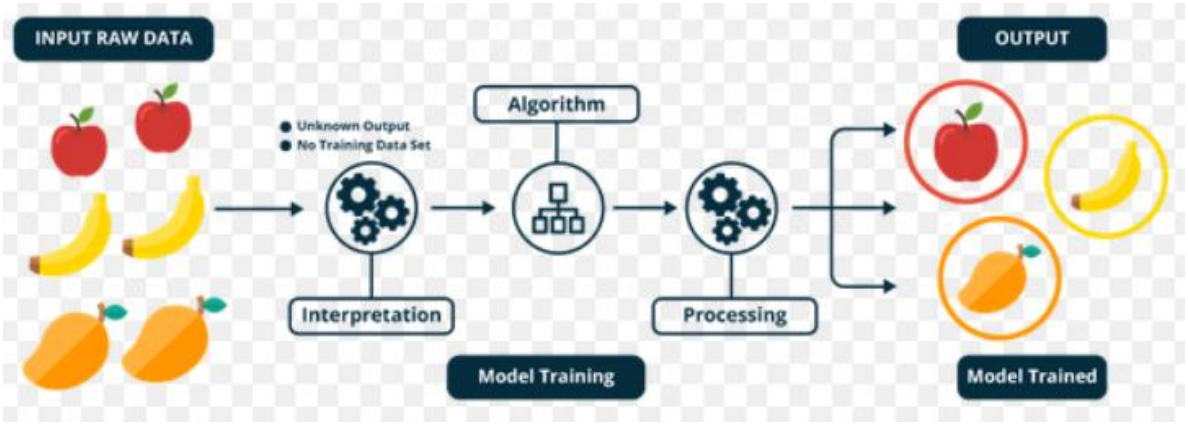
MACHINE LEARNING – METHODS - SEMI-SUPERVISED LEARNING

3. **Semi-supervised Learning**
   - Such kind of algorithms or methods are neither fully supervised nor fully unsupervised.
   - They basically fall between the two i.e. supervised and unsupervised learning methods.
   - These kinds of algorithms generally use small supervised learning component i.e. small amount of pre-labeled annotated data and large unsupervised learning component i.e. lots of unlabeled data for training.



MACHINE LEARNING – METHODS - SEMI-SUPERVISED LEARNING - EXAMPLE



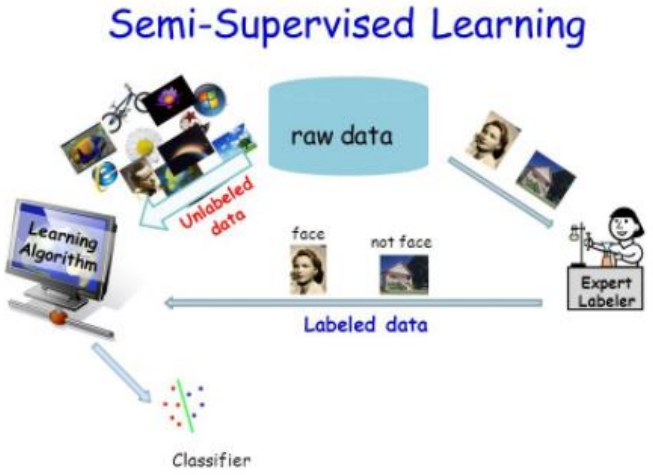


Supervised Learning

data	label
	Dog
	Bird
	Airplane
	Deer
	Cat
	Truck
	Ship

Semi-Supervised Learning

data	label
	Dog
	Bird
	No label
	No label
	No label
	No label
	No label



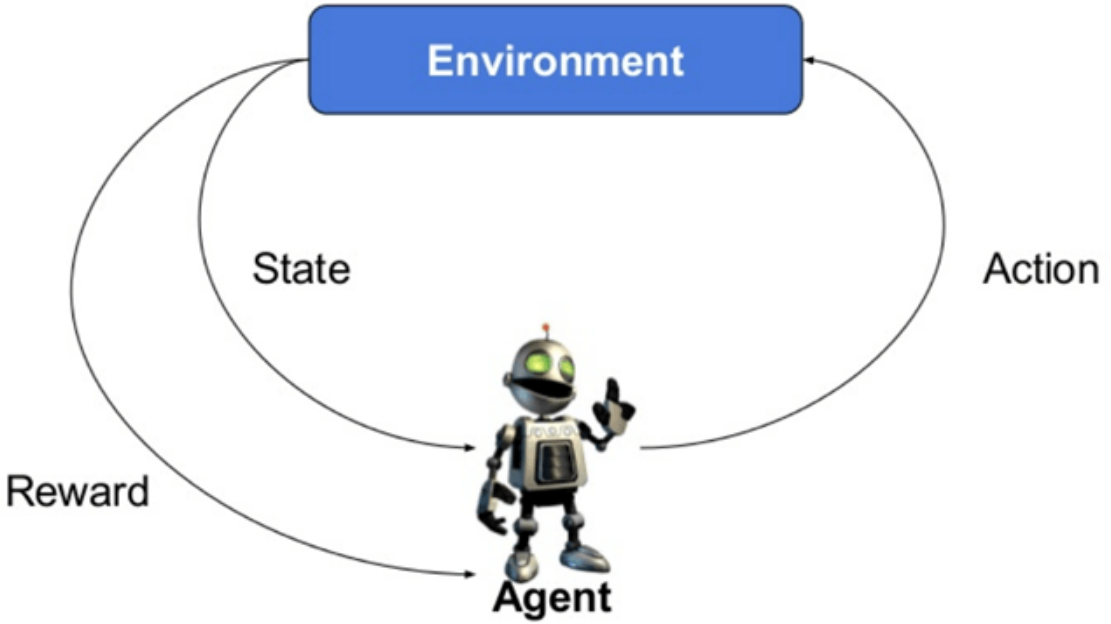
MACHINE LEARNING – METHODS - SEMI-SUPERVISED LEARNING - APPROACHES

The following approaches for implementing semi-supervised learning methods −

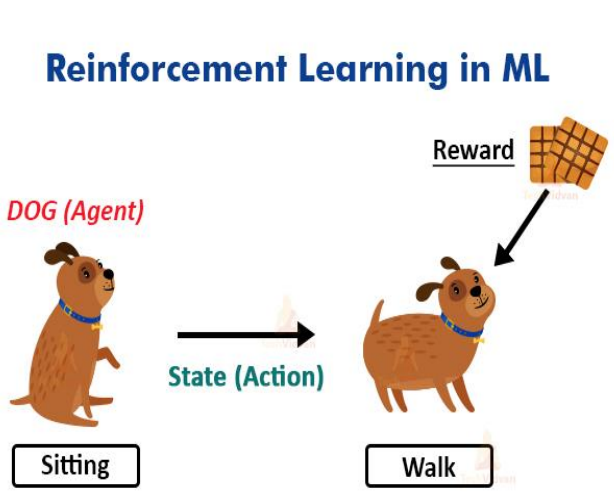
1. The first and simple approach is to build the supervised model based on small amount of labeled and annotated data and then build the unsupervised model by applying the same to the large amounts of unlabeled data to get more labeled samples. Now, train the model on them and repeat the process.
2. The second approach needs some extra efforts. In this approach, we can first use the unsupervised methods to cluster similar data samples, annotate these groups and then use a combination of this information to train the model.

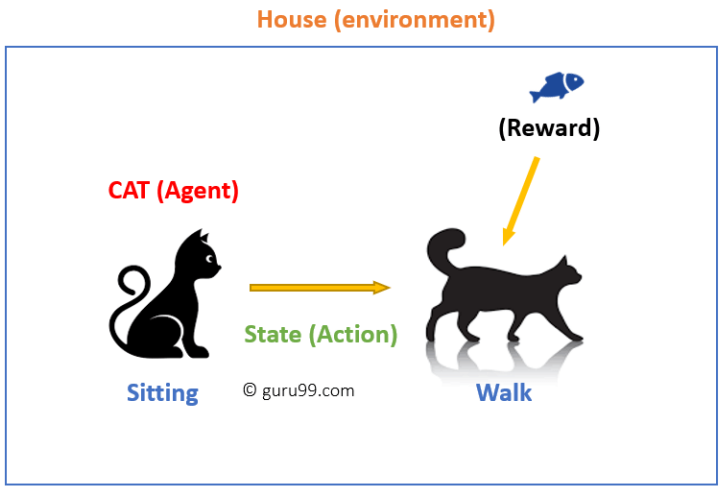
MACHINE LEARNING – METHODS - REINFORCEMENT LEARNING

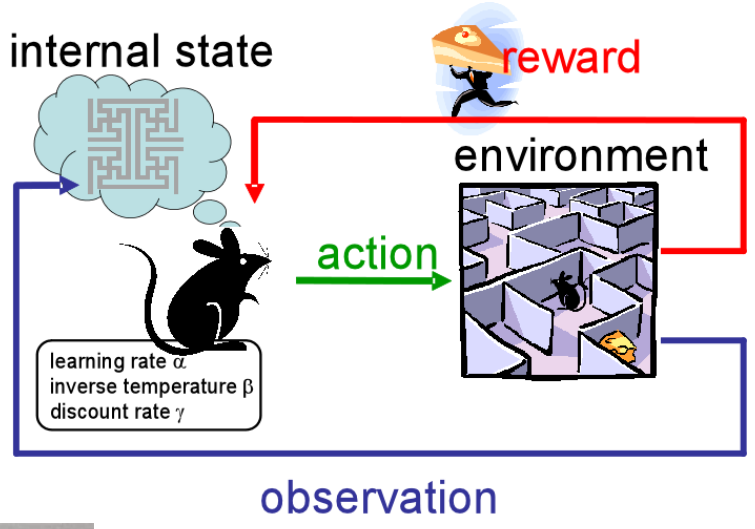
4. Reinforcement Learning
   - These methods are different from previously studied methods and very rarely used also.
   - In this kind of learning algorithms, there would be an agent that we want to train over a period of time so that it can interact with a specific environment.
   - The agent will follow a set of strategies for interacting with the environment and then after observing the environment it will take actions regards the current state of the environment.

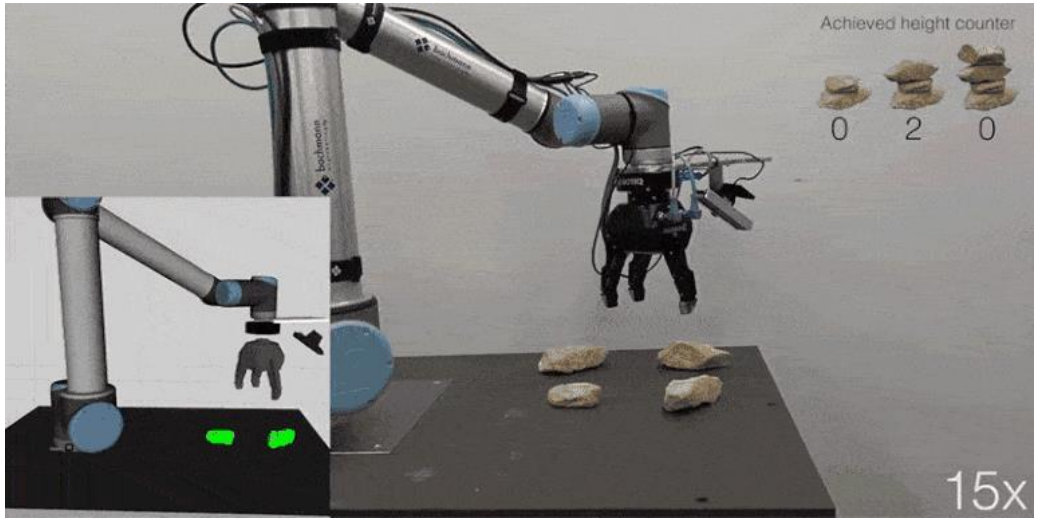


MACHINE LEARNING – METHODS - REINFORCEMENT LEARNING – EXAMPLE









MACHINE LEARNING – METHODS - REINFORCEMENT LEARNING

The following are the main steps of reinforcement learning methods −

Step 1 − First, we need to prepare an agent with some initial set of strategies.

Step 2 − Then observe the environment and its current state.

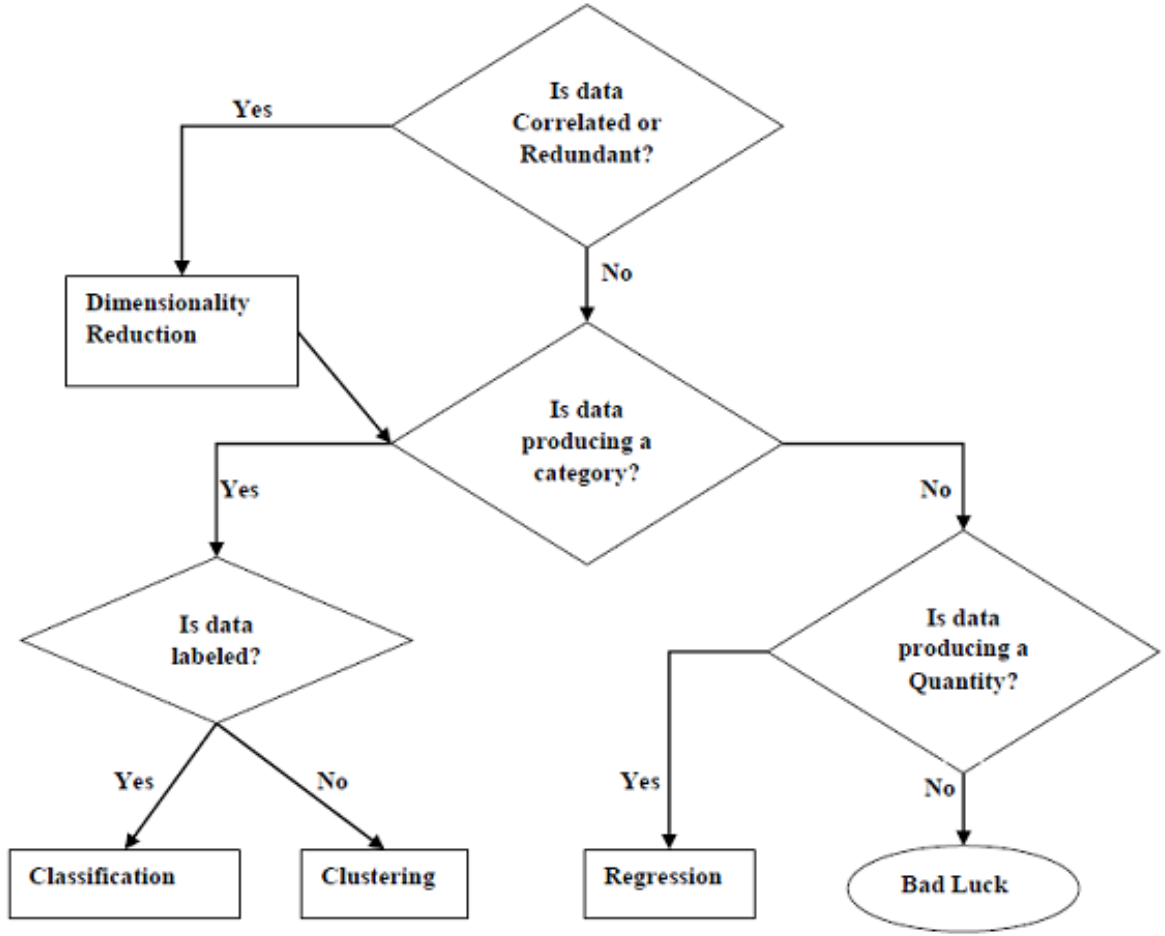
Step 3 − Next, select the optimal policy regards the current state of the environment and perform important action.

Step 4 − Now, the agent can get corresponding reward or penalty as per accordance with the action taken by it in previous step.

Step 5 − Now, we can update the strategies if it is required so.

Step 6 − At last, repeat steps 2-5 until the agent got to learn and adopt the optimal policies.

TASKS SUITED FOR MACHINE LEARNING



What type of task is appropriate for various ML problems −

MACHINE LEARNING PROCESS – BASED ON LEARNING ABILITY

- **Batch Learning**
  - In many cases, we have end-to-end Machine Learning systems in which we need to train the model in one go by using whole available training data.
  - Such kind of learning method or algorithm is called **Batch or Offline learning**.
  - It is called Batch or Offline learning because it is a one-time procedure and the model will be trained with data in one single batch.

The following are the main steps of Batch learning methods −

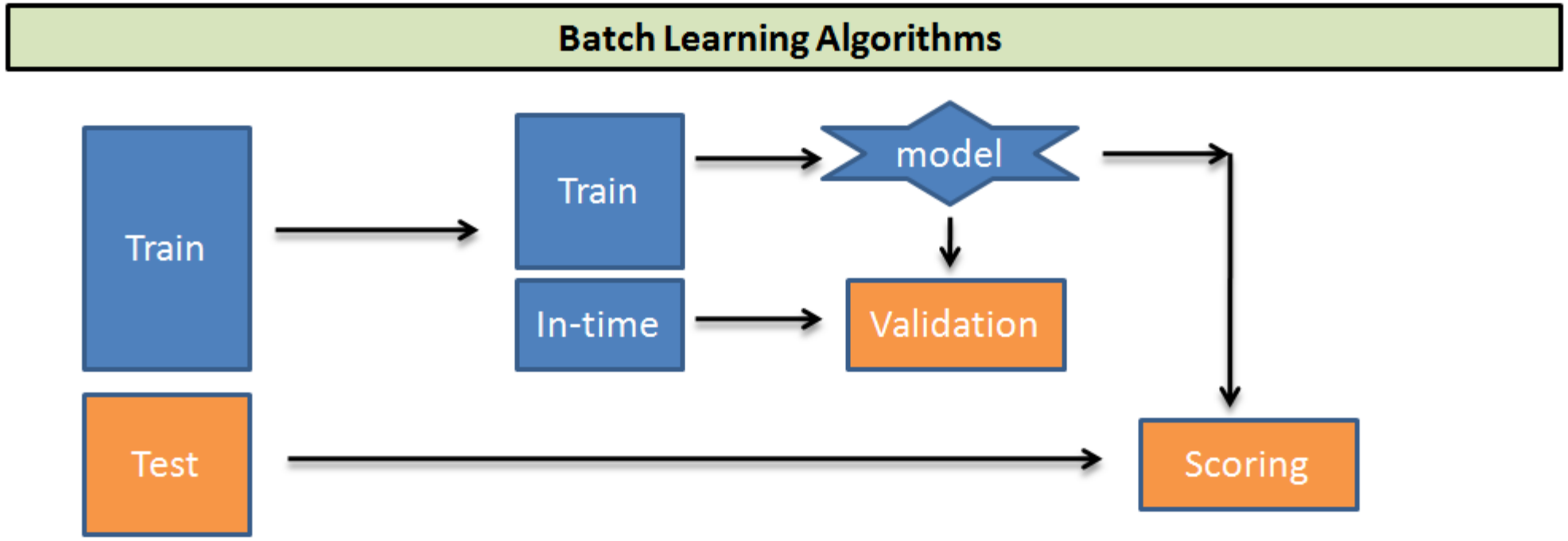
Step 1 − First, we need to collect all the training data for start training the model.

Step 2 − Now, start the training of model by providing whole training data in one go.

Step 3 − Next, stop learning/training process once you got satisfactory results/performance.

Step 4 − Finally, deploy this trained model into production. Here, it will predict the output for new data sample.

MACHINE LEARNING PROCESS – BASED ON LEARNING ABILITY



MACHINE LEARNING PROCESS – BASED ON LEARNING ABILITY

- **Online Learning**
  - It is completely opposite to the batch or offline learning methods.
  - In these learning methods, the training data is supplied in multiple incremental batches, called mini-batches, to the algorithm.

Followings are the main steps of Online learning methods −

Step 1 − First, we need to collect all the training data for starting training of the model.

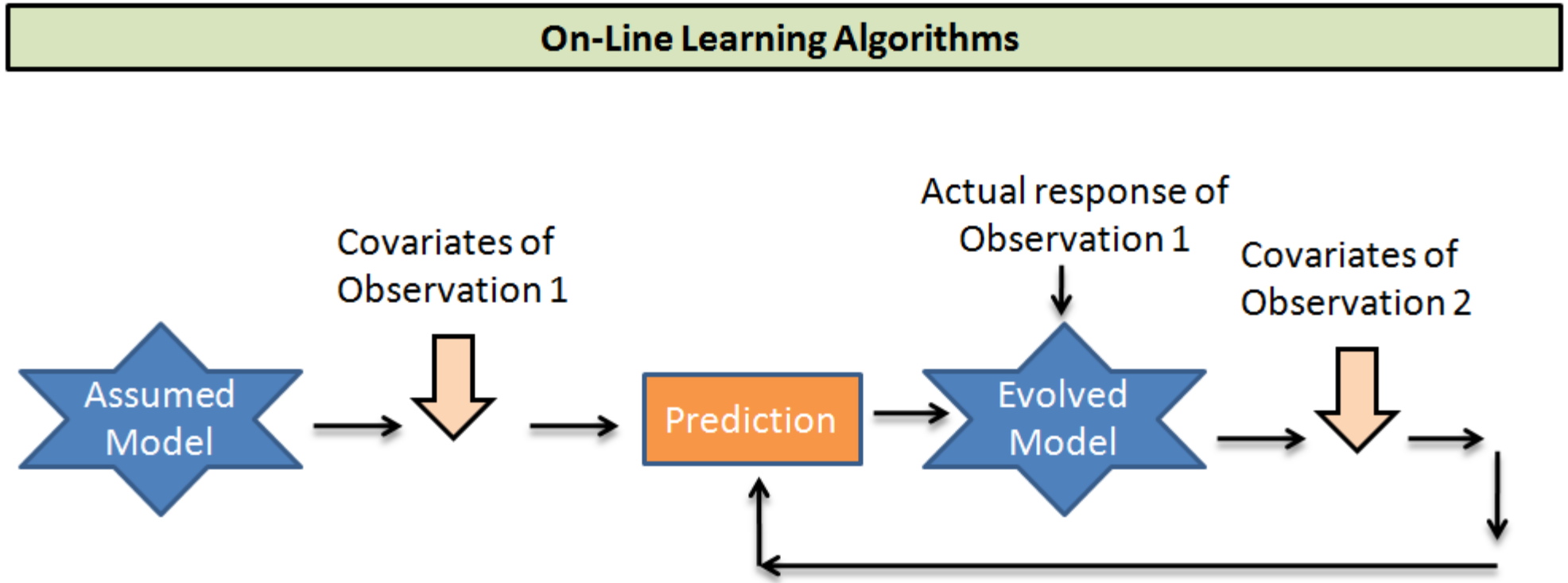
Step 2 − Now, start the training of model by providing a mini-batch of training data to the algorithm.

Step 3 − Next, we need to provide the mini-batches of training data in multiple increments to the algorithm.

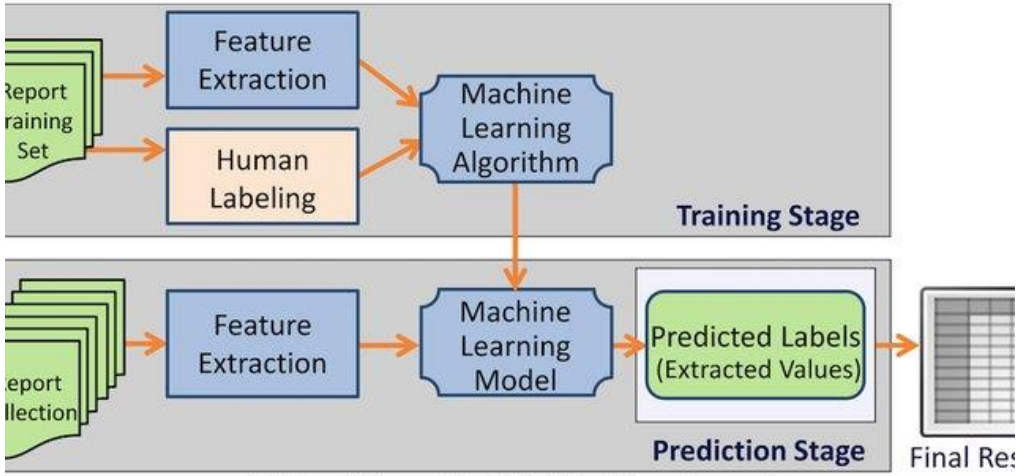
Step 4 − As it will not stop like batch learning hence after providing whole training data in mini-batches, provide new data samples also to it.

Step 5 − Finally, it will keep learning over a period of time based on the new data samples.

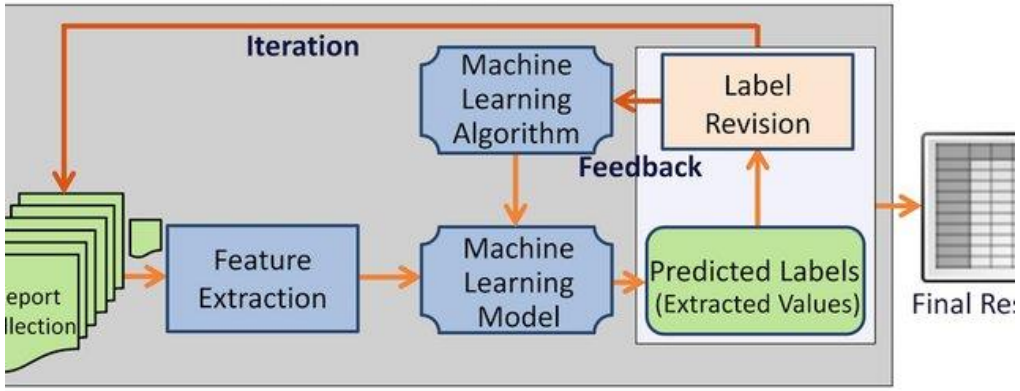
MACHINE LEARNING PROCESS – BASED ON LEARNING ABILITY



MACHINE LEARNING PROCESS – BASED ON LEARNING ABILITY



(a) Traditional Batch Machine Learning



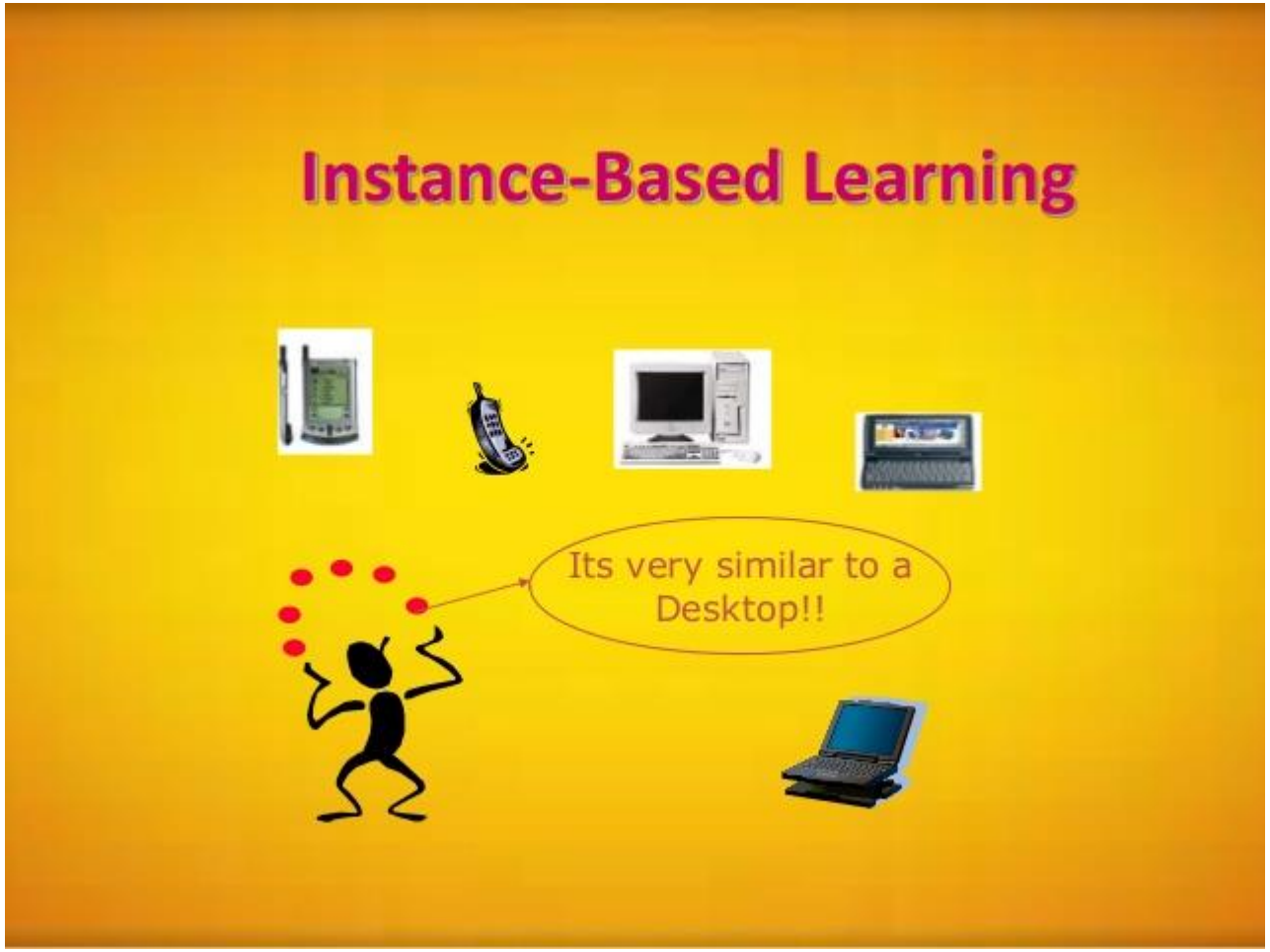
(b) Online Machine Learning

MACHINE LEARNING PROCESS – BASED ON GENERALIZATION APPROACH

Instance based Learning

- Instance based learning method is one of the useful methods that build the ML models by doing generalization based on the input data.
- It is opposite to the previously studied learning methods in the way that this kind of learning involves ML systems as well as methods that uses the raw data points themselves to draw the outcomes for newer data samples without building an explicit model on training data.
- In simple words, instance-based learning basically starts working by looking at the input data points and then using a similarity metric, it will generalize and predict the new data points.

MACHINE LEARNING PROCESS – BASED ON GENERALIZATION APPROACH



MACHINE LEARNING PROCESS – BASED ON GENERALIZATION APPROACH

- **Model based Learning**
  - In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called hyper parameters and in which input data is used to extract the features.
  - In this learning, hyper parameters are optimized based on various model validation techniques.
  - That is why we can say that Model based learning methods uses more traditional ML approach towards generalization.

LAB

PROBLEM - PIMA INDIANS DIABETES DATABASE

- Predict the onset of diabetes based on diagnostic measures
  - This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.
  - The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on

DATA - PIMA INDIANS DIABETES DATABASE

The columns of this datasets are as follows:

1. Pregnancies — Number of times pregnant
2. GlucosePlasma — glucose concentration 2 hours in an oral glucose tolerance test
3. Blood Pressure — Diastolic blood pressure (mm Hg)
4. SkinThickness — Triceps skin-fold thickness (mm)
5. Insulin — Two hours of serum insulin (mu U/ml)
6. BMI — Body mass index (weight in kg/(height in $m)^2$)
7. Diabetes Pedigree Function — Diabetes pedigree function
8. Age — Age in years
9. Outcome — Class variable (0 or 1)

- The first eight columns represent the independent variables, and the last column denotes the binary dependent variable. There are a total of 768 entries in the dataset. The outcome variable is set to 1 for 268 entries, and the rest are set to 0.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Looking at Raw Data
  - The very first recipe is for looking at your raw data.
  - It is important to look at raw data because the insight we will get after looking at raw data will boost our chances to better pre-processing as well as handling of data for ML projects.

```
from pandas import read_csv
path = "diabetes.csv"
data = read_csv(path)
print(data.head(15)) # Raw Data
print(data.shape) # Dimensions
print(data.dtypes) # Data Types
print(data.describe()) # Statistical Summary of Data
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Looking at Raw Data
  - Class distribution statistics is useful in classification problems where we need to know the balance of class values. It is important to know class value distribution because if we have highly imbalanced class distribution i.e. one class is having lots more observations than other class, then it may need special handling at data preparation stage of our ML project.

```
count_class = data.groupby('Outcome').size()

print(count_class)
```

From the above output, it can be clearly seen that the number of observations with class 0 are almost double than number of observations with class 1.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Looking at Raw Data
  - Correlation between Attributes
  - The relationship between two variables is called correlation. In statistics, the most common method for calculating correlation is Pearson's Correlation Coefficient. It can have three values as follows −
    - Coefficient value = 1 − It represents full positive correlation between variables.
    - Coefficient value = -1 − It represents full negative correlation between variables.
    - Coefficient value = 0 − It represents no correlation at all between variables.

```
correlations = data.corr(method='pearson')

print(correlations)
```

It is always good for us to review the pairwise correlations of the attributes in our dataset before using it into ML project because some machine learning algorithms such as linear regression and logistic regression will perform poorly if we have highly correlated attributes.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Looking at Raw Data
  - Skew of Attribute Distribution
    - Skewness may be defined as the distribution that is assumed to be Gaussian but appears distorted or shifted in one direction or another, or either to the left or right.
    - Reviewing the skewness of attributes is one of the important tasks due to following reasons −
      - Presence of skewness in data requires the correction at data preparation stage so that we can get more accuracy from our model.
      - Most of the ML algorithms assumes that data has a Gaussian distribution i.e. either normal of bell curved

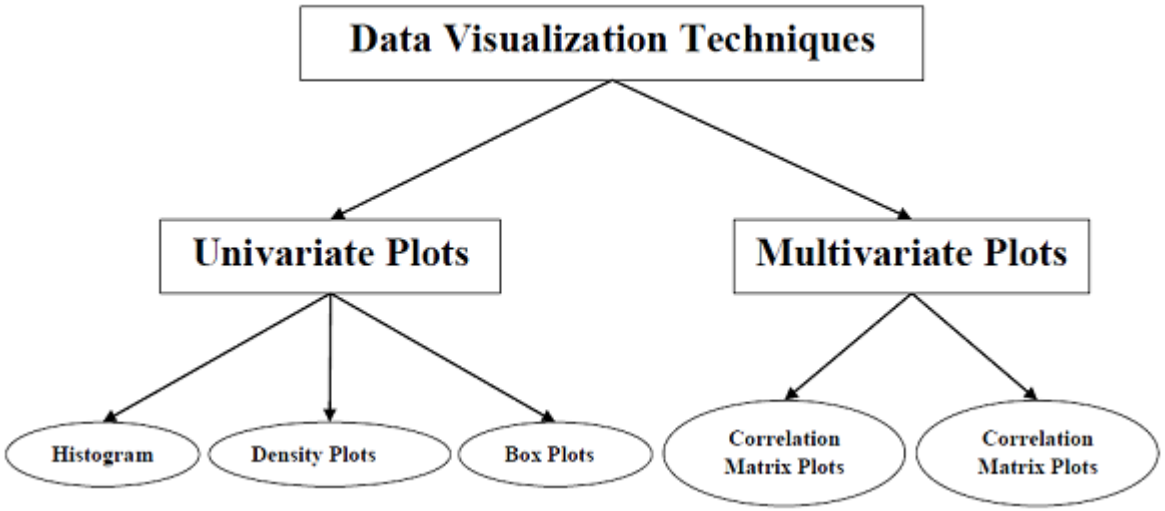
print(data.skew())

From the above output, positive or negative skew can be observed. If the value is closer to zero, then it shows less skew.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Visualization, to understand the data
  - With the help of data visualization, we can see how the data looks like and what kind of correlation is held by the attributes of data. It is the fastest way to see if the features correspond to the output.



```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Univariate Plots: Understanding Attributes Independently
  - The simplest type of visualization is single-variable or "univariate" visualization. With the help of univariate visualization, we can understand each attribute of our dataset independently. The following are some techniques in Python to implement univariate visualization −
- Histograms
  - Histograms group the data in bins and is the fastest way to get idea about the distribution of each attribute in dataset. The following are some of the characteristics of histograms −
    - It provides us a count of the number of observations in each bin created for visualization.
    - From the shape of the bin, we can easily observe the distribution i.e. weather it is Gaussian, skewed or exponential.
    - Histograms also help us to see possible outliers.

data.hist()

pyplot.show()

We can observe that perhaps Age, DiabetesPedigreeFunction and Insulin attribute may have exponential distribution while BMI and Glucose have Gaussian distribution.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Univariate Plots: Understanding Attributes Independently
  - The simplest type of visualization is single-variable or "univariate" visualization. With the help of univariate visualization, we can understand each attribute of our dataset independently. The following are some techniques in Python to implement univariate visualization –
- Density Plots
  - Another quick and easy technique for getting each attributes distribution is Density plots. It is also like histogram but having a smooth curve drawn through the top of each bin. We can call them as abstracted histograms.

data.plot(kind='density', subplots=True, layout=(3,3), sharex=False)

pyplot.show()

The difference between Density plots and Histograms can be easily understood

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Univariate Plots: Understanding Attributes Independently
  - Box and Whisker Plots
    - Box and Whisker plots, also called boxplots in short, is another useful technique to review the distribution of each attribute's distribution. The following are the characteristics of this technique −
    - It is univariate in nature and summarizes the distribution of each attribute.
    - It draws a line for the middle value i.e. for median.
    - It draws a box around the 25% and 75%.
    - It also draws whiskers which will give us an idea about the spread of the data.
    - The dots outside the whiskers signifies the outlier values. Outlier values would be 1.5 times greater than the size of the spread of the middle data.

data.plot(kind='box', subplots=True, layout=(3,3), sharex=False,sharey=False)

pyplot.show()

It can be observed that Age, Insulin and skinthickness appear skewed towards smaller values.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Multivariate Plots: Interaction Among Multiple Variables
  - Another type of visualization is multi-variable or "multivariate" visualization. With the help of multivariate visualization, we can understand interaction between multiple attributes of our dataset.
  - The following are some techniques in Python to implement multivariate visualization −
    - Correlation Matrix Plot
    - Correlation is an indication about the changes between two variables. Previously, the importance of Correlation Pearson's Correlation coefficients are seen. We can plot correlation matrix to show which variable is having a high or low correlation in respect to another variable.

```
import numpy
correlations = data.corr()
fig = pyplot.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(correlations, vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = numpy.arange(0,9,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
names=data.head(0)
ax.set_xticklabels(names)
ax.set_yticklabels(names)
pyplot.show()
```

From the above output of correlation matrix, we can see that it is symmetrical i.e. the bottom left is same as the top right. It is also observed that each variable is positively correlated with each other.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Multivariate Plots: Interaction Among Multiple Variables
  - Scatter Matrix Plot
    - Scatter plots shows how much one variable is affected by another or the relationship between them with the help of dots in two dimensions.
    - Scatter plots are very much like line graphs in the concept that they use horizontal and vertical axes to plot data points.

```
from matplotlib import pyplot
from pandas.plotting import scatter_matrix
%matplotlib
scatter_matrix(data)
pyplot.show()
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Preparing Data
  - Machine Learning algorithms are completely dependent on data because it is the most crucial aspect that makes model training possible.
  - On the other hand, if we won't be able to make sense out of that data, before feeding it to ML algorithms, a machine will be useless.
  - In simple words, we always need to feed right data i.e. the data in correct scale, format and containing meaningful features, for the problem we want machine to solve.
  - This makes data preparation the most important step in ML process.
  - Data preparation may be defined as the procedure that makes our dataset more appropriate for ML process.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Data Pre-processing
  - After selecting the raw data for ML training, the most important task is data pre-processing.
  - In broad sense, data preprocessing will convert the selected data into a form we can work with or can feed to ML algorithms.
  - We always need to preprocess our data so that it can be as per the expectation of machine learning algorithm.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Data Pre-processing Scaling Techniques
  - Most probably our dataset comprises of the attributes with varying scale, but we cannot provide such data to ML algorithm hence it requires rescaling.
  - Data rescaling makes sure that attributes are at same scale.
  - Generally, attributes are rescaled into the range of 0 and 1.
  - ML algorithms like gradient descent and k-Nearest Neighbors requires scaled data.
  - We can rescale the data with the help of MinMaxScaler class of scikit-learn Python library.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Data Pre-processing Scaling Technique Example

```
from numpy import set_printoptions
from sklearn import preprocessing
array = data.values
data_scaler = preprocessing.MinMaxScaler(feature_range=(0,1))
data_rescaled = data_scaler.fit_transform(array)
set_printoptions(precision=1)
print ("\nScaled data:\n", data_rescaled[0:10])
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Normalization
  - Another useful data preprocessing technique is Normalization.
  - This is used to rescale each row of data to have a length of 1.
  - It is mainly useful in Sparse dataset where we have lots of zeros.
  - We can rescale the data with the help of Normalizer class of scikit-learn Python library.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Types of Normalization
- In machine learning, there are two types of normalization preprocessing techniques as follows −
  - L1 Normalization
    - It may be defined as the normalization technique that modifies the dataset values in a way that in each row the sum of the absolute values will always be up to 1.
    - It is also called Least Absolute Deviations.
  - Example

```
from sklearn.preprocessing import Normalizer
Data_normalizer = Normalizer(norm='l2').fit(array)
Data_normalized = Data_normalizer.transform(array)
set_printoptions(precision=2)
print ("\nNormalized data:\n", Data_normalized [0:3])
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Types of Normalization
- In machine learning, there are two types of normalization preprocessing techniques as follows −
  - L2 Normalization
    - It may be defined as the normalization technique that modifies the dataset values in a way that in each row the sum of the squares will always be up to 1.
    - It is also called least squares.
  - Example

```
from sklearn.preprocessing import Normalizer
Data_normalizer = Normalizer(norm='l2').fit(array)
Data_normalized = Data_normalizer.transform(array)
set_printoptions(precision=2)
print ("\nNormalized data:\n", Data_normalized [0:3])
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Binarization
- This is the technique with the help of which we can make our data binary.
- We can use a binary threshold for making our data binary.
- The values above that threshold value will be converted to 1 and below that threshold will be converted to 0.
- For example, if we choose threshold value = 0.5, then the dataset value above it will become 1 and below this will become 0.
- That is why we can call it binarizing the data or thresholding the data.
- This technique is useful when we have probabilities in our dataset and want to convert them into crisp values.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Binarization – Example

```
from sklearn.preprocessing import Binarizer
binarizer = Binarizer(threshold=0.5).fit(array)
Data_binarized = binarizer.transform(array)
print ("\nBinary data:\n", Data_binarized [0:5])
```

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- Data Labeling
  - The importance of good for ML algorithms as well as some techniques to pre-process the data before sending it to ML algorithms.
  - One more aspect in this regard is data labeling. It is also very important to send the data to ML algorithms having proper labeling.
  - For example, in case of classification problems, lot of labels in the form of words, numbers etc. are there on the data.
  - Most of the sklearn functions expect that the data with number labels rather than word labels.
  - Hence, we need to convert such labels into number labels.
  - This process is called label encoding. We can perform label encoding of data with the help of LabelEncoder() function of scikit-learn Python library.

```
print(data.shape)
```

DATA - PIMA INDIANS DIABETES DATABASE

- **Data Labeling**

```
import numpy as np
from sklearn import preprocessing

input_labels = ['red','black','red','green','black','yellow','white']

encoder = preprocessing.LabelEncoder()
encoder.fit(input_labels)

test_labels = ['green','red','black']
encoded_values = encoder.transform(test_labels)
print("\nLabels =", test_labels)
print("Encoded values =", list(encoded_values))
encoded_values = [3,0,4,1]
decoded_list = encoder.inverse_transform(encoded_values)

print("\nEncoded values =", encoded_values)
print("\nDecoded labels =", list(decoded_list))
```

END OF UNIT I