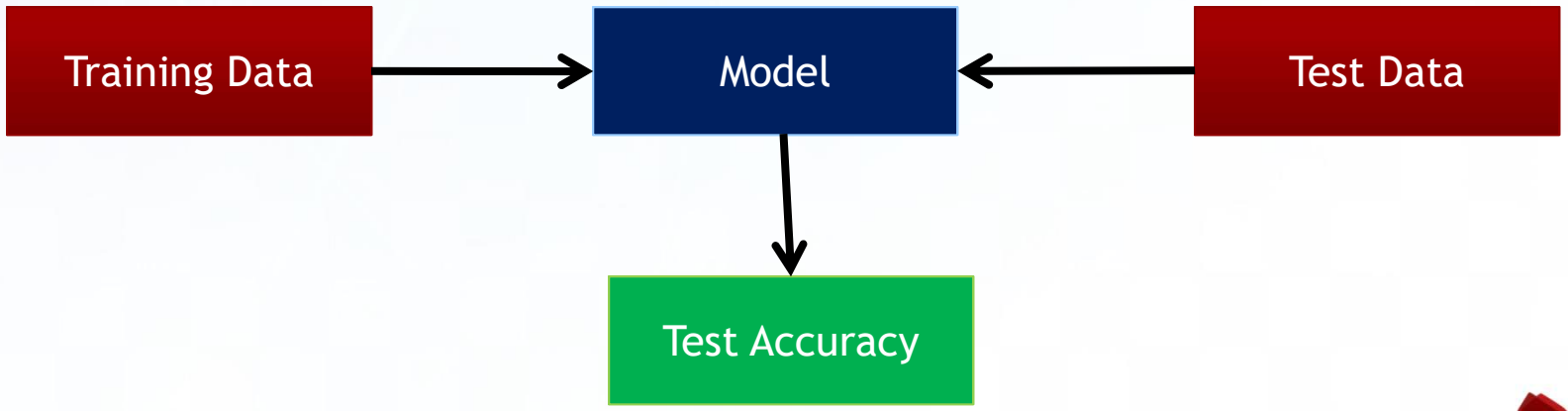
Spark MLlib

Machine Learning with Spark

What is Machine Learning ?

Machine learning is a **method of data analysis** that automates analytic model building. ML uses various algorithms to iteratively learn from data and there by finds hidden insights without explicitly being programmed.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

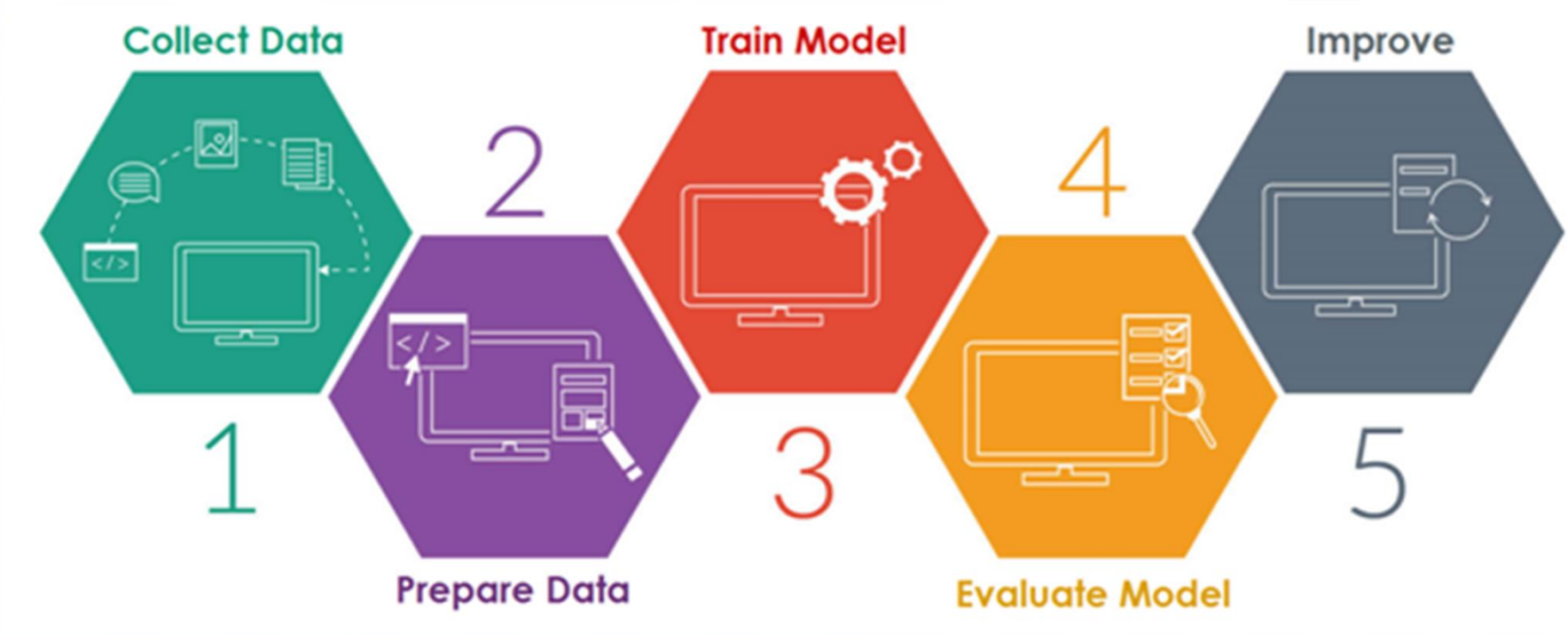


Machine Learning Features

1. ML uses data to **detect hidden patterns** in a dataset and **adjust program actions** accordingly

2. ML focuses on the development of computer programs called *models* that can teach themselves to grow and change when exposed to new data.

Machine Learning Phases



Steps of Machine Learning

Steps of Machine Learning

Collecting Data

- This stage involves collecting data from various sources

Data Wrangling

- It is he process of cleaning and converting “raw data” into a format that allows convenient consumption.

Analyze Data

- Data is analyzed to select and filter the data required to prepare the training data

Steps of Machine Learning

Train Algorithm

- The algorithm is trained on the training dataset through which the algorithm understands the pattern and rules that govern the data

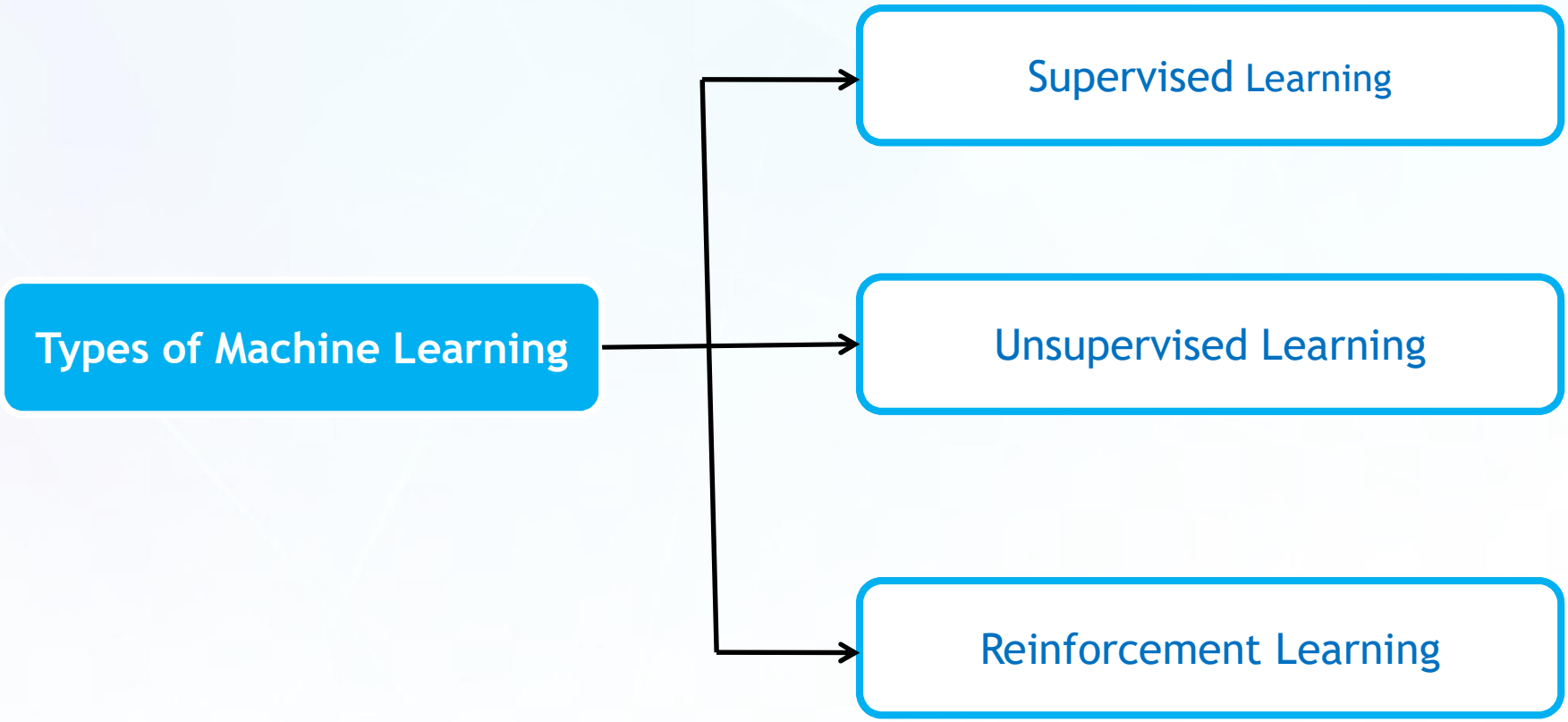
Test Algorithm

- The testing dataset determines the accuracy of our model

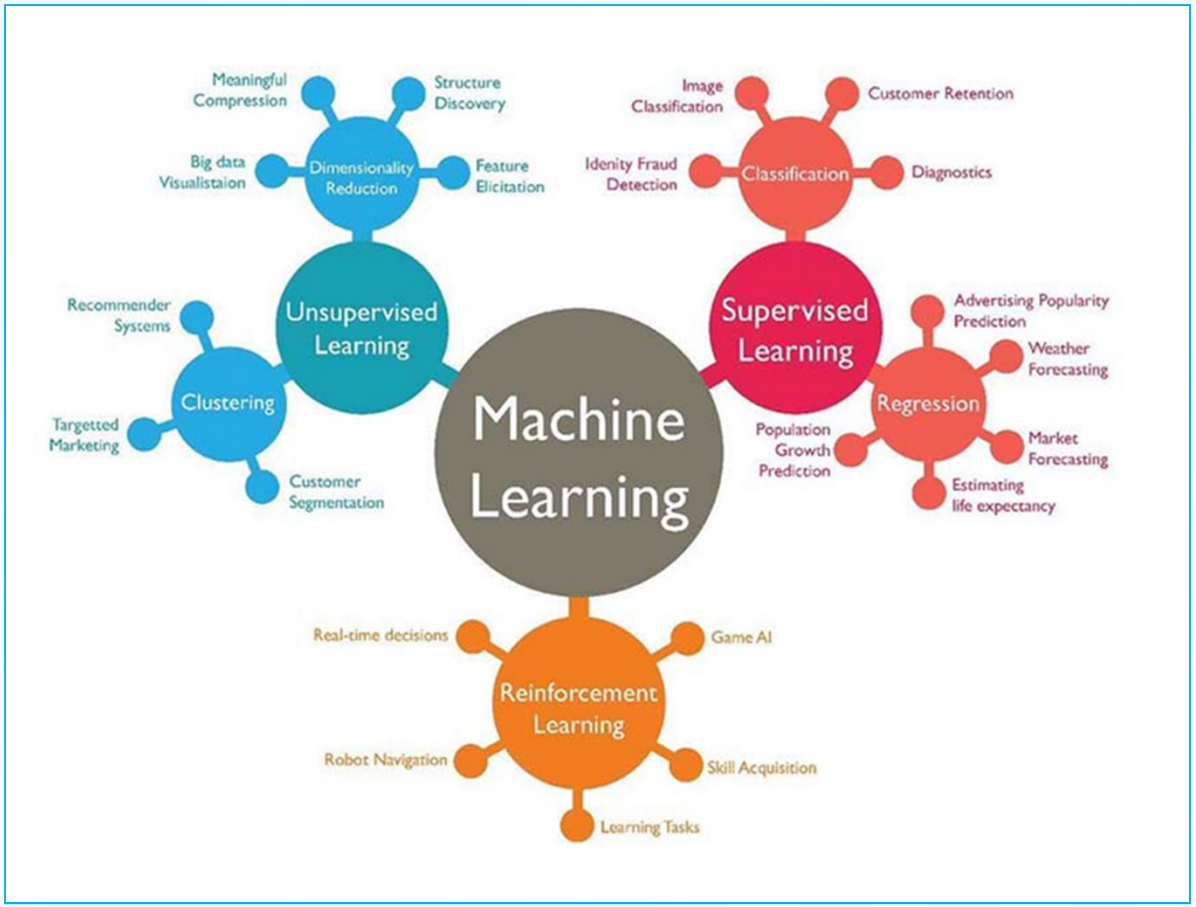
Deployment

- If the speed and accuracy of the model is acceptable, then that model can be deployed in the production to handle real data.

Types of Machine Learning



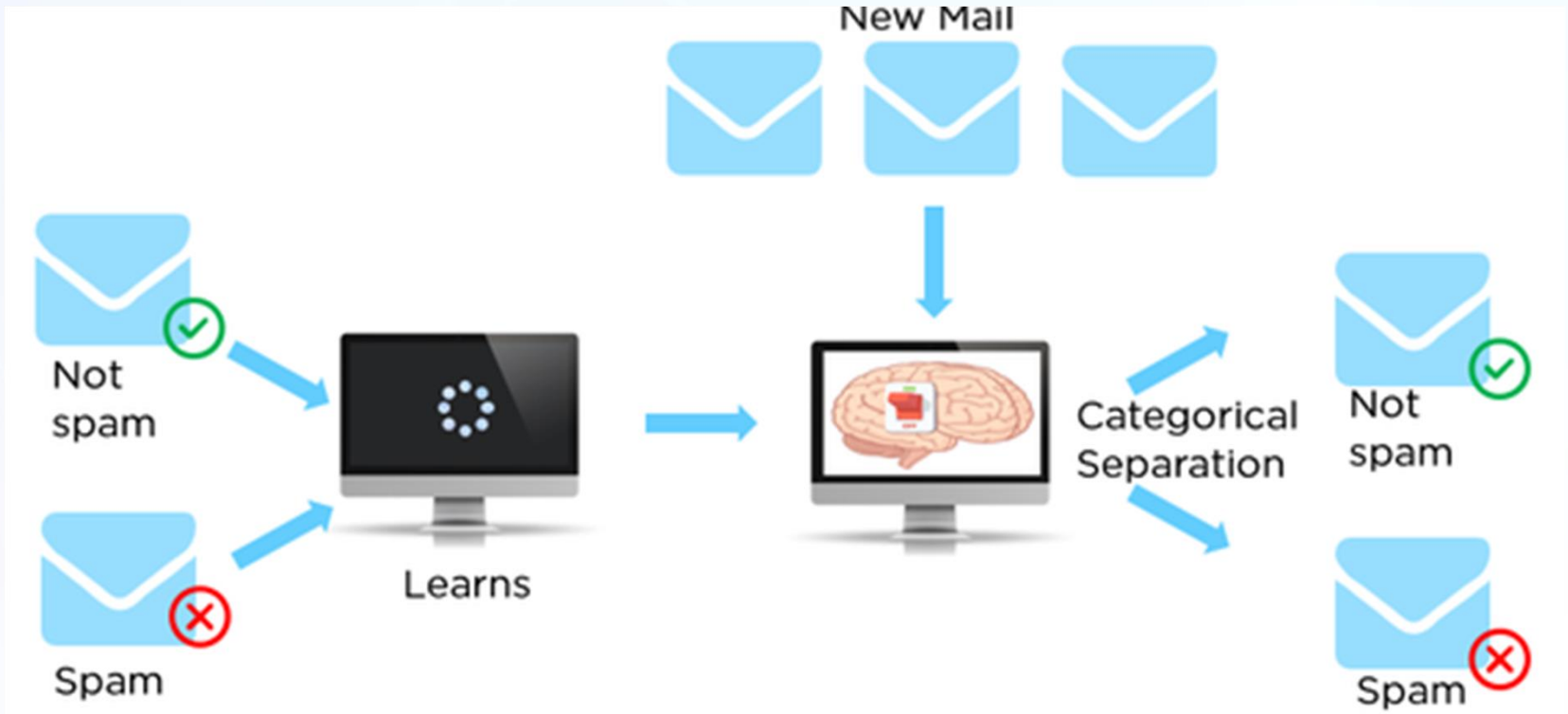
Types of Machine Learning



Supervised Learning

1. • Supervised Learning algorithms are trained using **labeled data** such as an input where the desired output is known.
2. • The **algorithm learns** by comparing the actual output with the correct outputs to find the errors
3. • It then modifies the model accordingly through methods like classification, regression, prediction and gradient boosting.
4. • Supervised Learning models use patterns to **predict the values of the label on additional unlabeled data**.

Supervised Learning - Example



Supervised Learning models are trained using "**labeled data**"

Supervised Learning - Example

- An email spam filter will be fed with thousands, possibly millions of emails. Each of these emails will already have a label - ‘spam’ or ‘not spam’.
- The supervised machine learning algorithm will then figure out which type of emails are being marked as spam.
- Next time an email is about to hit your inbox, the spam filter will use statistical analysis to figure out how likely it is that the email is spam. If the probability is high, it will label it as spam and the email won’t hit your inbox.

Training with Labeled Data

From a large collection of pictures containing faces and non-faces, we feed around 80% of the pictures along with the label to train the algorithm.

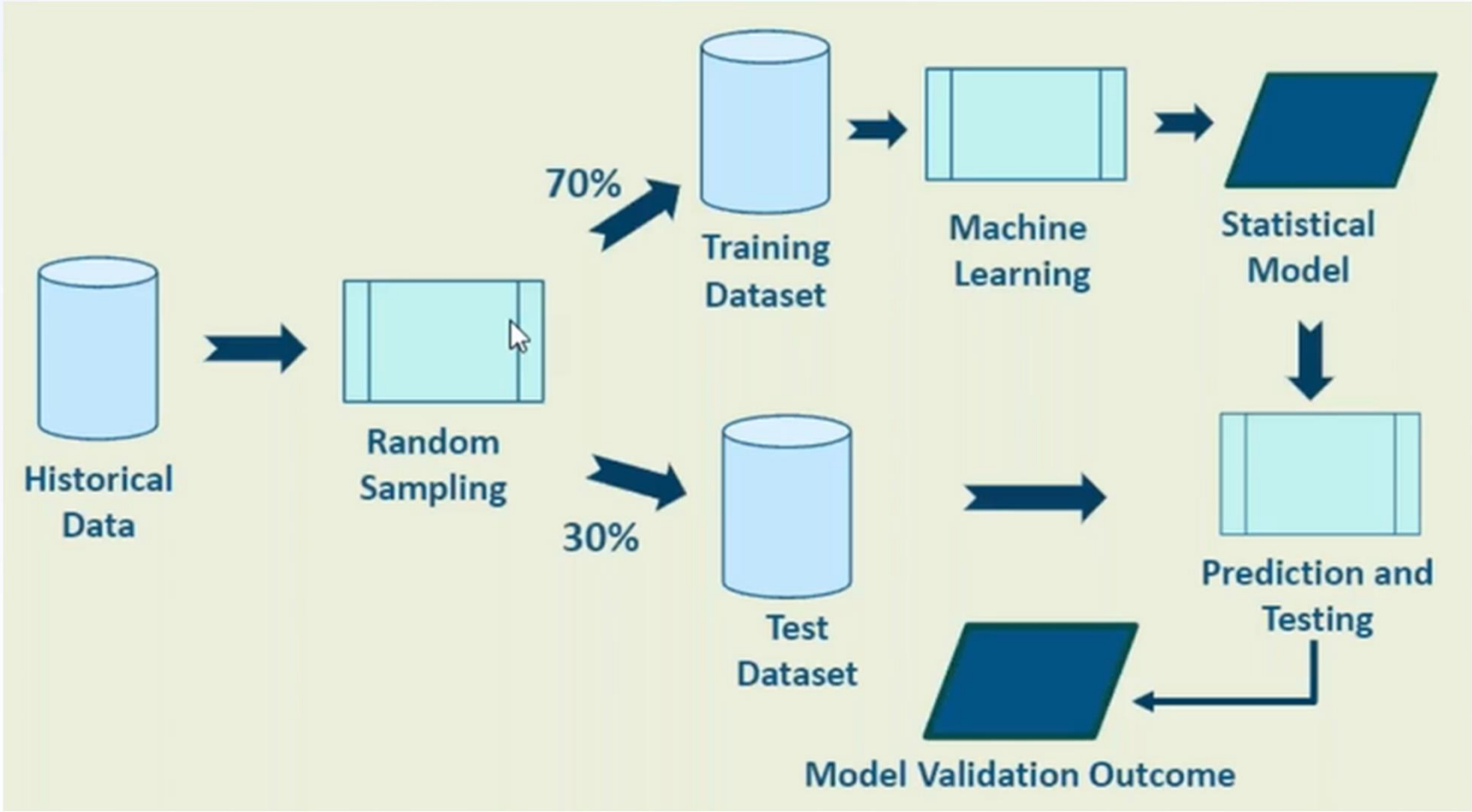
Label : "Face"

Label : "Non-Face"

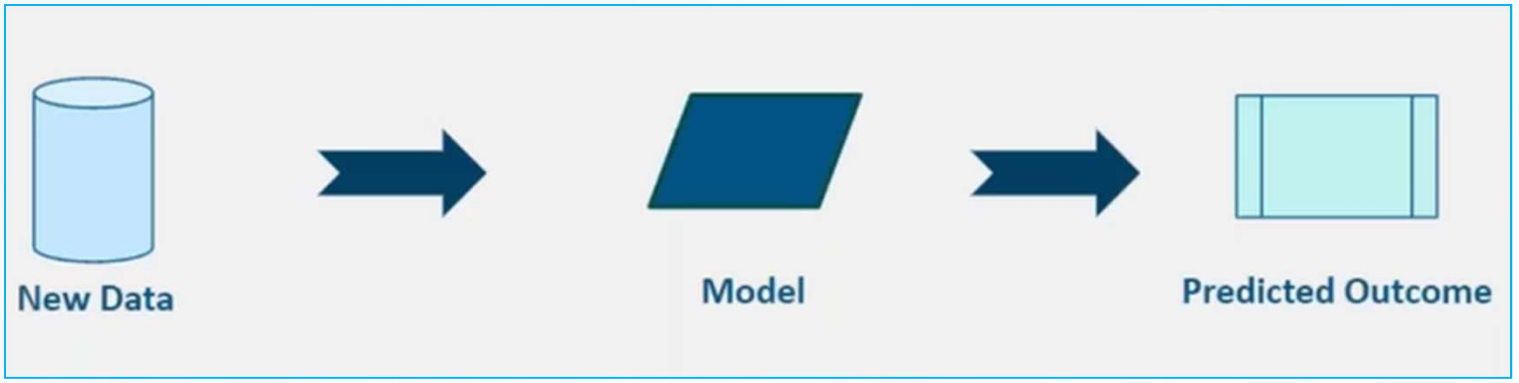
input to machine

The machine scans the images and finds all the pixel features of a picture that are particular to faces and create a **model**.

Process Flow: Testing & Training



Process Flow: Prediction



- The model is used to predict the outcome of a new dataset.
- Whenever the performance of the model degrades or fall below the acceptable accuracy level, the model need to be retrained.

Unsupervised Learning

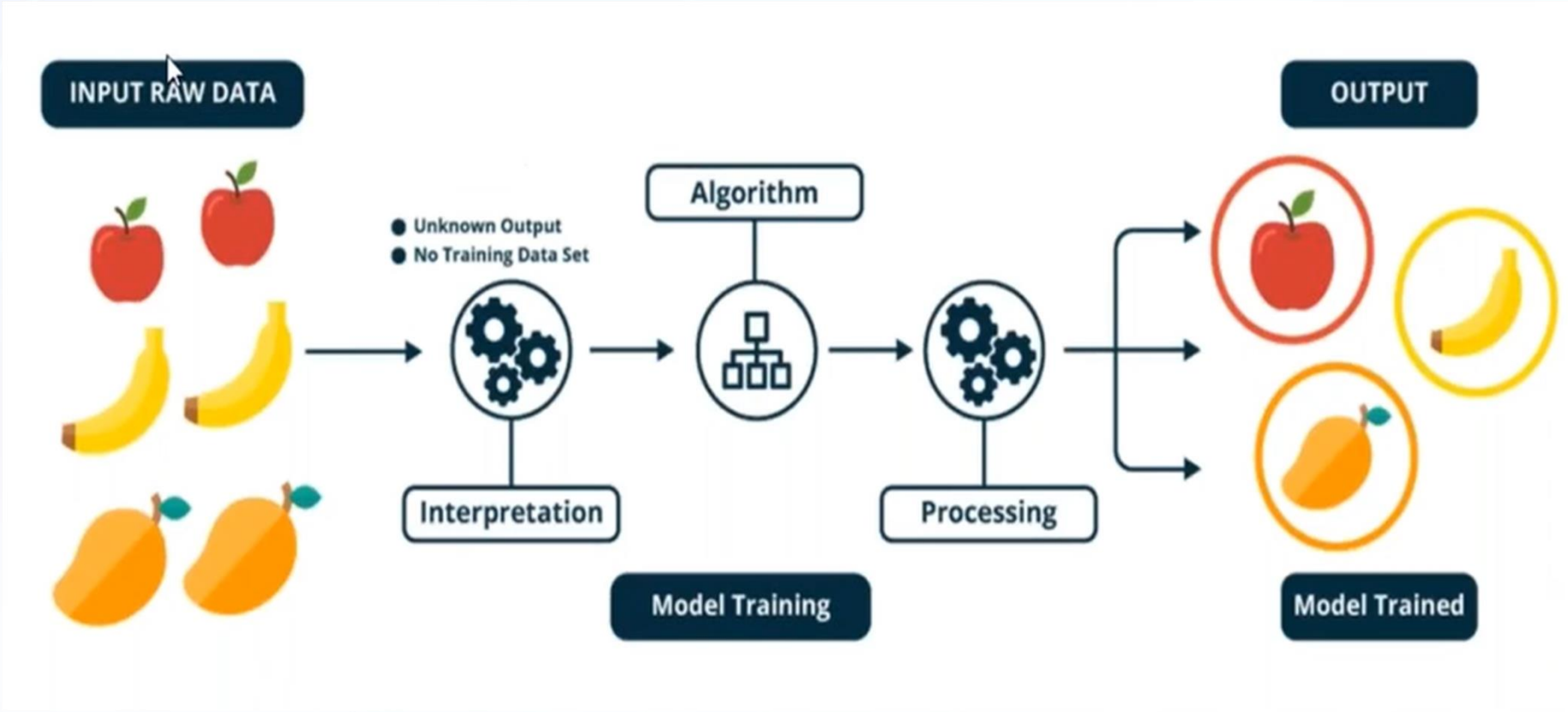
- Unsupervised Learning is used when you have data with no historic labels. The algorithm should figure out what is being shown.
- The goal is to learn the hidden patterns and correlations within the data
- Popular techniques include self-organizing maps, nearest neighbor mapping, K-means clustering etc.
- These algorithms are also used to segment text-topics, item recommendations, identifying outliers in data etc.

The machine learns the patterns all by itself and divide the data into categories. There are no labels provided in advance.

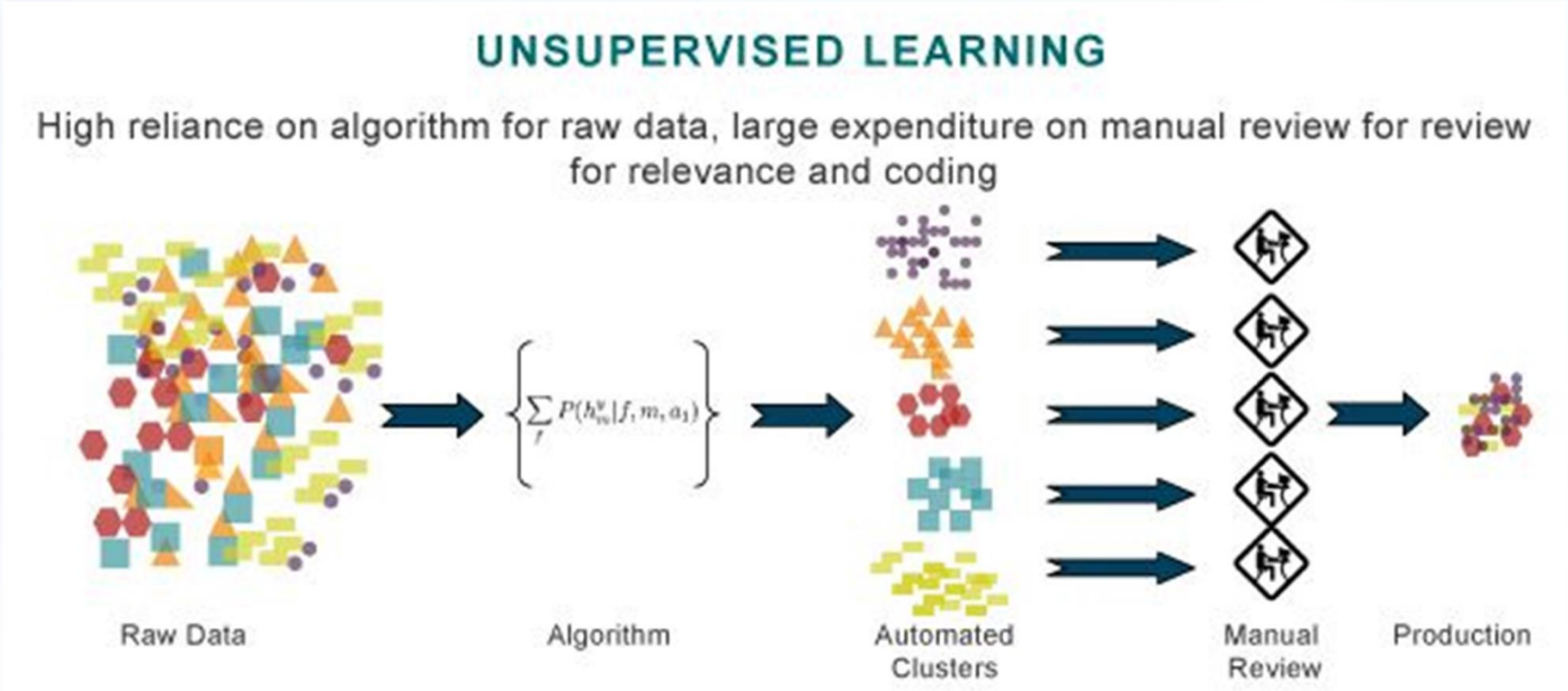
Unsupervised Learning

How will I arrange them?

Unsupervised Learning



Unsupervised Learning - Clustering



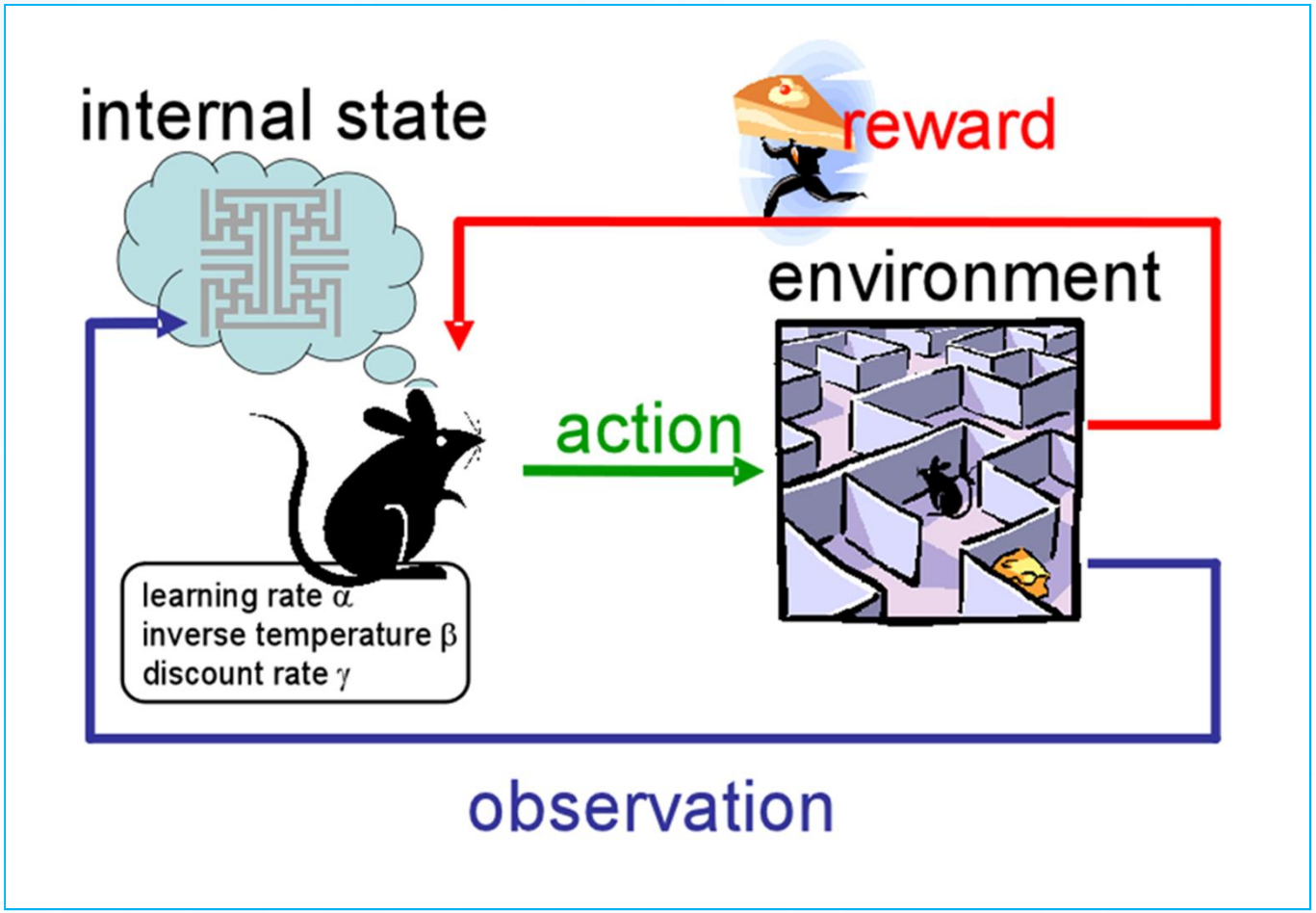
Reinforcement Learning

- **R**einforcement Learning allows the machine or software agent to learn its behaviour based on feedback from the environment.
- **R**einforcement Learning is learning how to map situations to actions so as to maximize a reward.
- If the problem is modelled with care, some RL algorithms can converge to the global optimum, the ideal behaviour that maximises the reward.
- Some common use cases are navigation, gaming and robotics.

Actions are what agents can do

Environment is everything the agent interacts with

Reinforcement Learning



Spark MLlib

Spark MLlib Overview

- Spark MLlib is used to perform machine learning in Apache Spark.
- MLlib consists popular classes of algorithms and utilities such as classification, regression, clustering, collaborative filtering, dimensionality reduction etc.
- Divided into two packages:
  - ***spark.mllib*** contains the original API built on top of RDDs.
  - ***spark.ml*** provides higher level API built on top of **DataFrames** for constructing ML pipelines.

Why MLlib ?

Speed: Compared to traditional MR programs Spark ML provides 100x faster in memory and 10x faster on disk

Scalability: ability to run the same ML code in a laptop or on large clusters

Spark MLlib

Streamlined: Building MLlib on top of Spark makes it possible to use Spark's unified stack components.

Simplicity and compatibility: Similar APIs familiar to data scientists from R & Python backgrounds

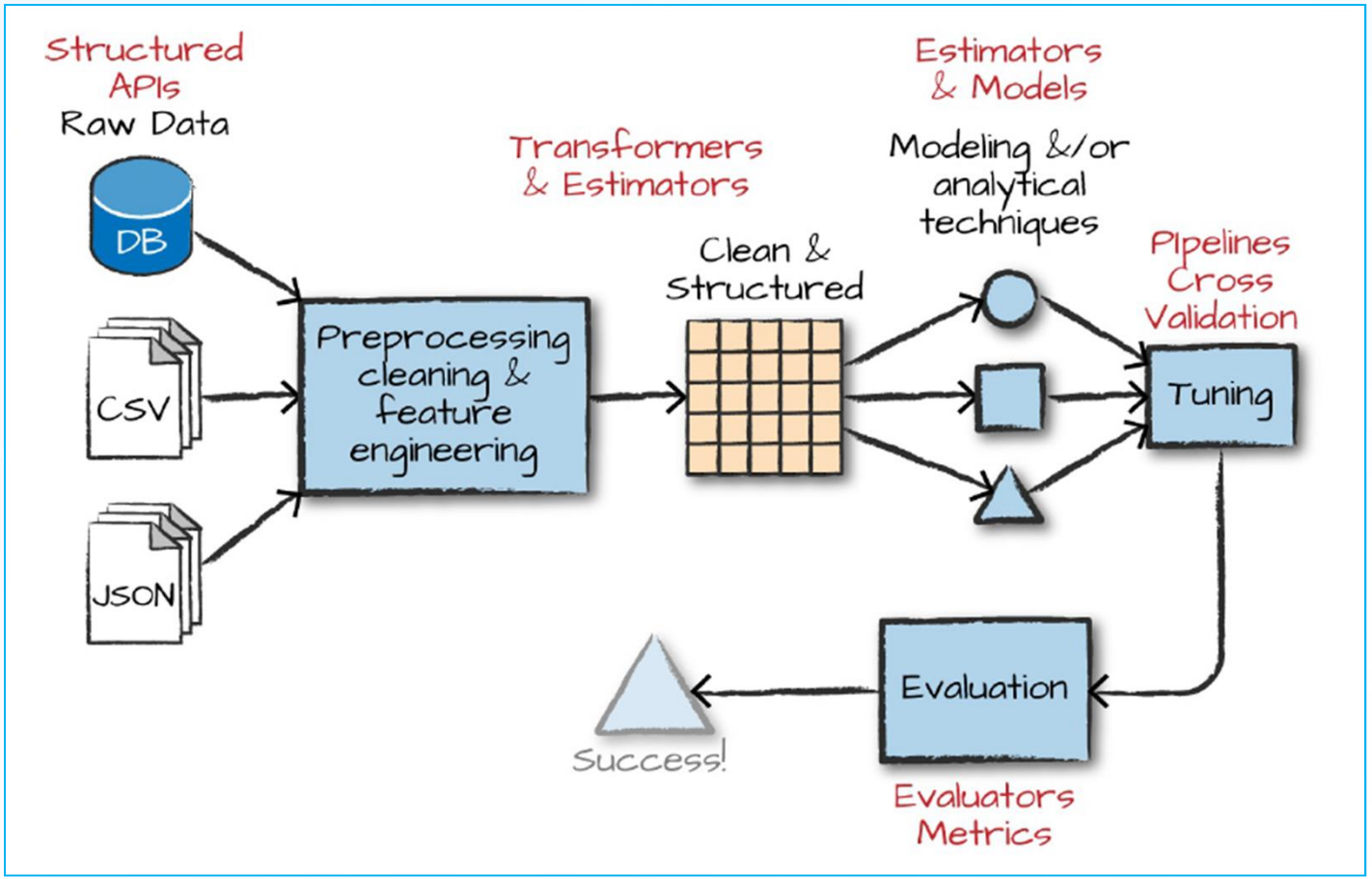
Spark MLlib - ML Tools

ML Algorithms	**ML Algorithms** form the core of MLlib. Include common learning algorithms such as classification, regression, clustering and collaborative filtering
Featurization	**Featurization** includes feature extraction, transformation, Dim. Reduction and selection
Pipelines	**Pipelines** provide tools of constructing, evaluating and tuning ML pipelines
Persistence	**Persistence** helps in saving and loading algorithms, models and pipelines.
Utilities	**Utilities** for linear algebra, statistics and data handling

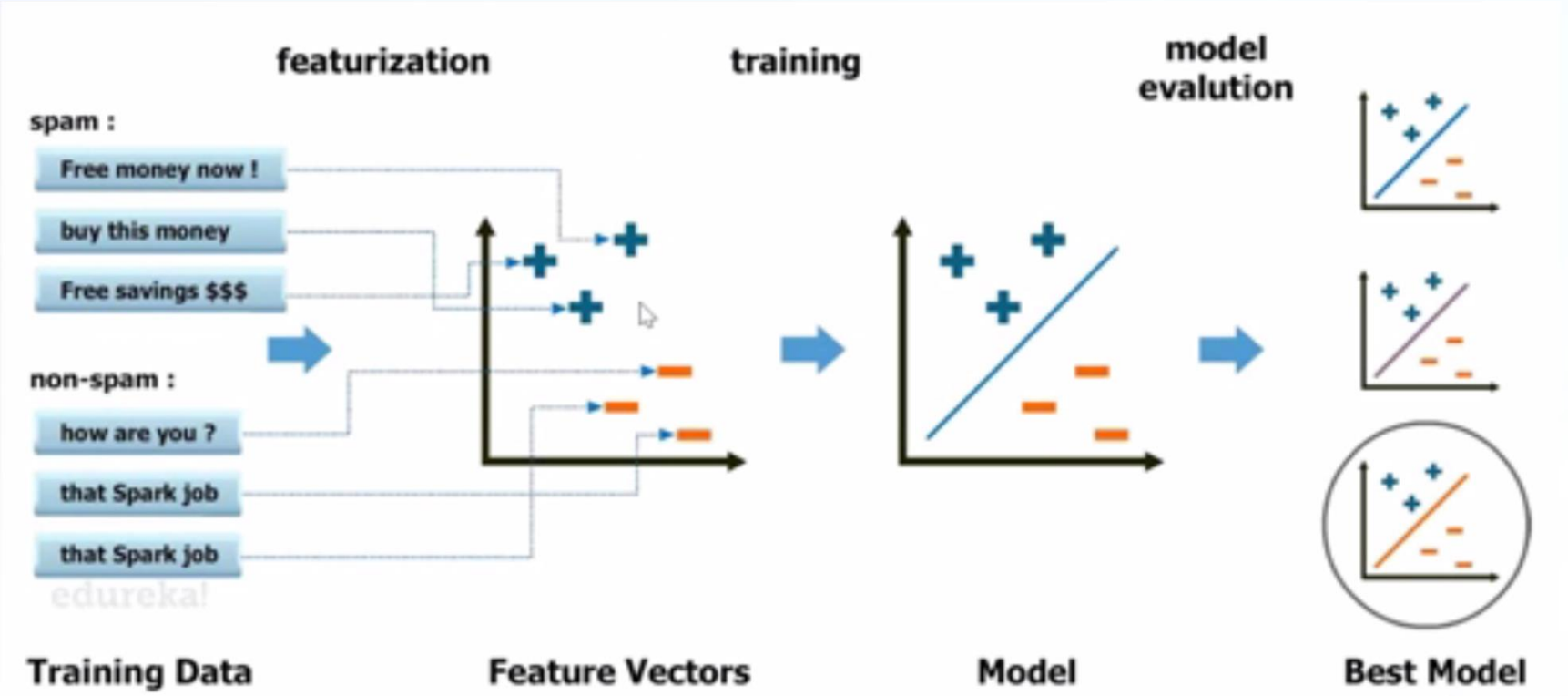
Spark ML Concepts

DataFrame	This ML API uses DataFrame from Spark SQL as an ML dataset, which can hold a variety of data types. E.g., a DataFrame could have different columns storing text, feature vectors, true labels, and predictions.
Transformer	Transformer is an algorithm which can transform one DataFrame into another DataFrame. E.g., an ML model is a Transformer which transforms a DataFrame with features into a DataFrame with predictions.
Estimator	Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer. E.g., a learning algorithm is an Estimator which trains on a DataFrame and produces a model.
Pipeline	A Pipeline chains Transformers and Estimators together to specify an ML workflow.
Parameter	Transformers and Estimators now share a common API for specifying parameters.

Spark ML Workflow



Steps in ML Pipeline



MLlib Concepts

Vector: A vector (or a point) is just a set of numbers. This set of numbers (coordinates) defines a point's position in space. The number of coordinates determines the dimensions of the space.

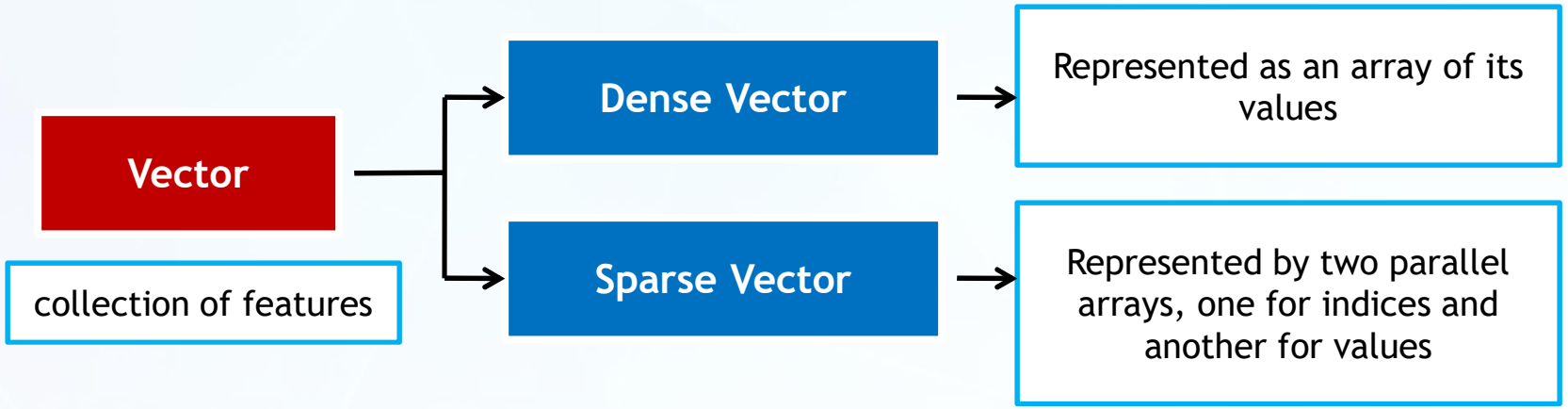
Hyperspace: A space with more than three dimensions is called **hyperspace.**

Features: Dimensions in vectors are called **features.** In another way, we can define a feature as an individual measurable property of a phenomenon being observed.

Features
Area: 4200 sq.ft, Lot size: 31000 sq.ft, Number of rooms: 4

Vectors	
Dense	[4200, 31000, 4]
Sparse	(3, [0, 1, 2], [4200.0, 31000.0, 4.0])

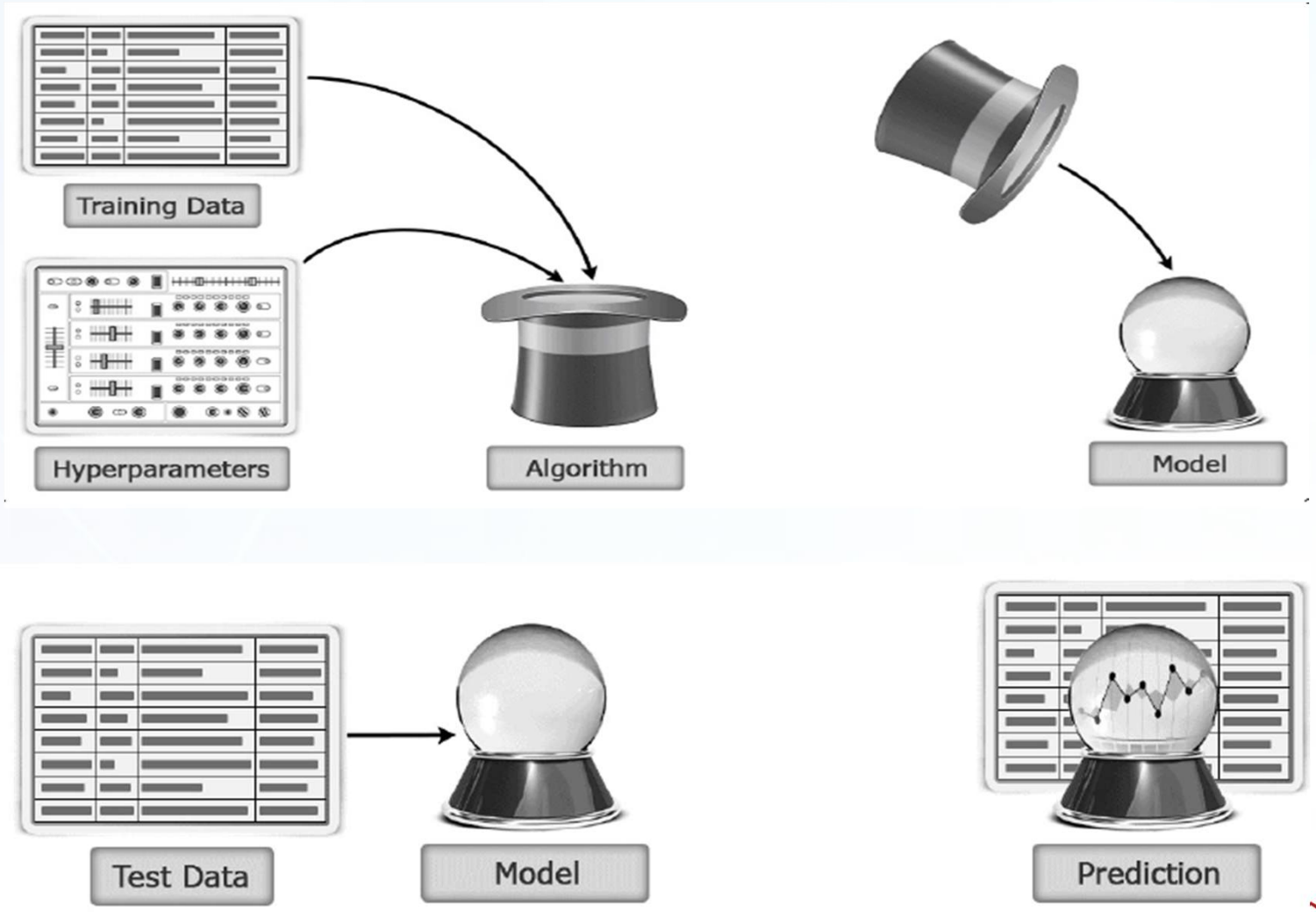
MLlib Concepts - Vectors




```
val dense = Vectors.dense(4500d,41000d,0d, 0d, 4d)

val sparse = Vectors.sparse(5, Array(0,1,4), Array(4500d,41000d,4d))
```

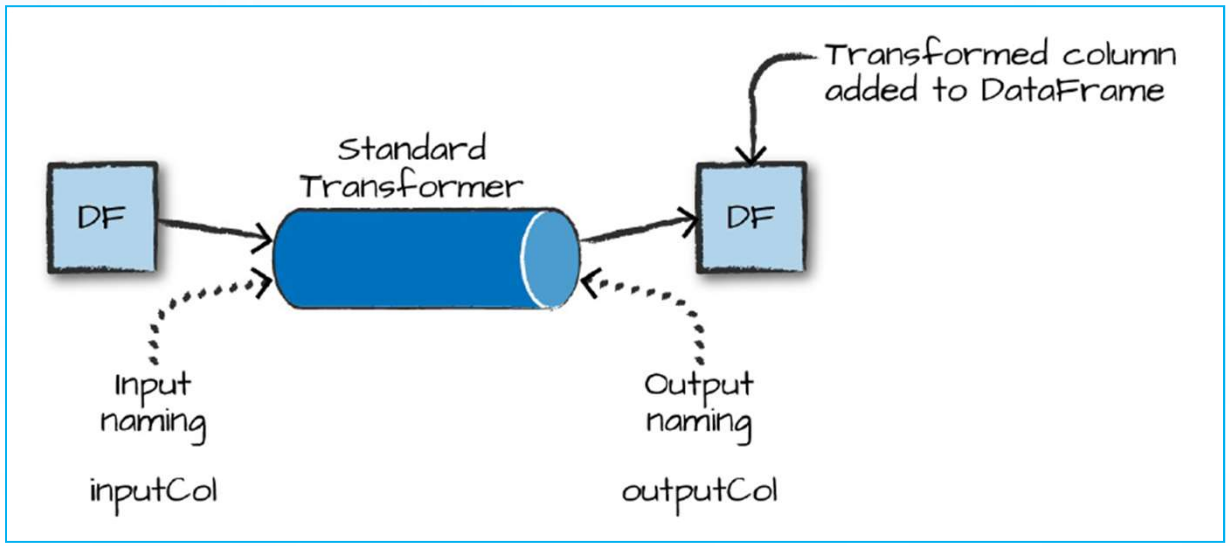
MLlib Concepts – Algorithms & Model



- A **hyperparameter** is a parameter whose value is set before the **learning** process begins

Transformers

- A Transformer is an abstraction that includes feature transformers and learned models.
- Technically, a Transformer implements a method `transform()`, which ***converts one DataFrame into another***, generally by appending one or more columns.



Transformers

For example:

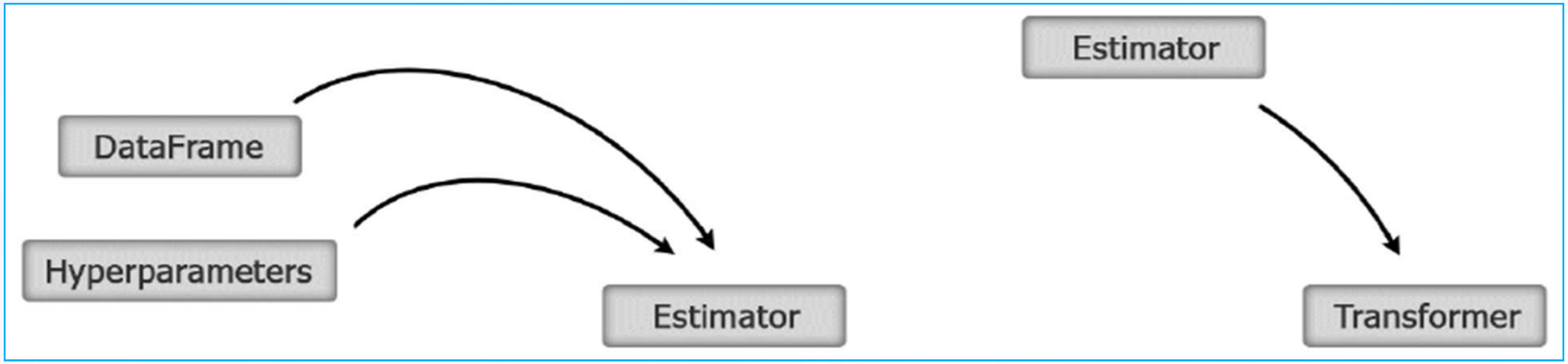
- A feature transformer might take a DataFrame, read a column (e.g., text), map it into a new column (e.g., feature vectors), and output a new DataFrame with the mapped column appended.

- A learning model might take a DataFrame, read the column containing feature vectors, predict the label for each feature vector, and output a new DataFrame with predicted labels appended as a column.

Estimators

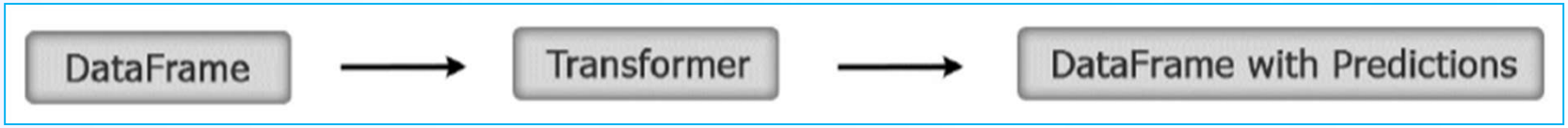
- An Estimator abstracts the concept of **a learning algorithm** or any algorithm that fits or trains on data.
- Technically, an Estimator implements a method `fit()`, which accepts a DataFrame and produces a Model, which is a Transformer.
- For example, a learning algorithm such as LogisticRegression is an Estimator, and calling fit() trains a LogisticRegressionModel, which is a Model and hence a Transformer.

Estimators & Transformers



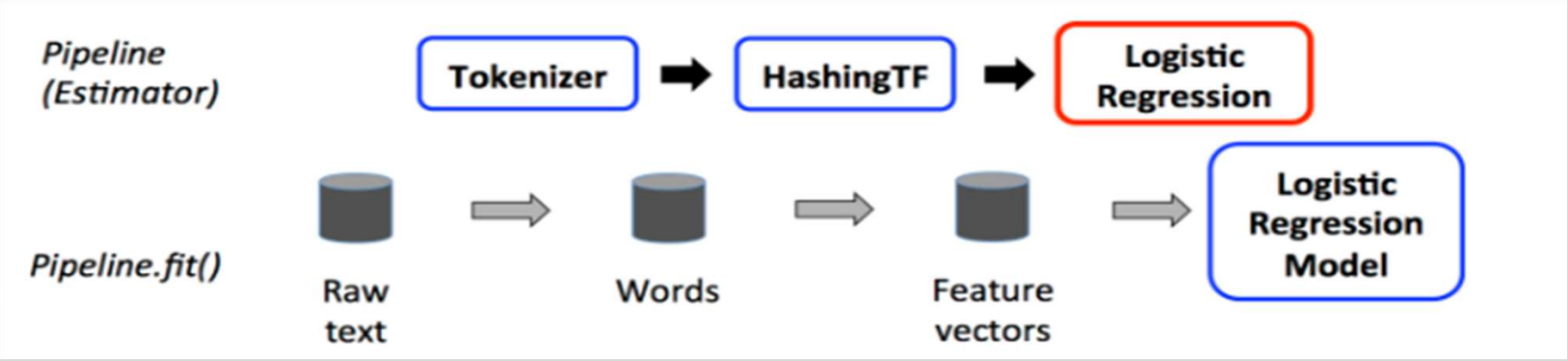
- In Spark ML, an estimator is provided as a DataFrame (via the **fit** method), and the output after training is a Transformer

Estimators & Transformers



- The Transformer takes one DataFrame as input and outputs another transformed (via the **transform** method) DataFrame.
- For example, it can take a DataFrame with the test data and enrich this DataFrame with an additional column for predictions and then output

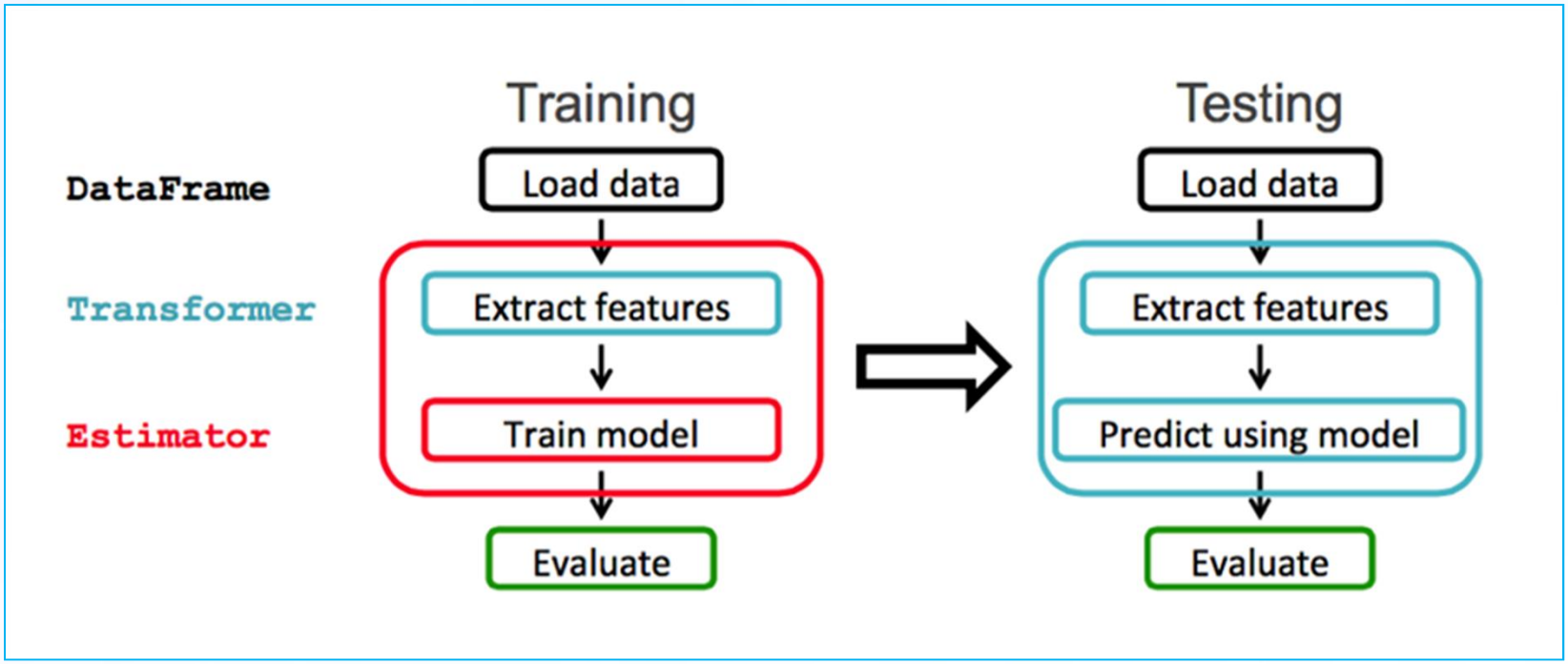
MLlib Concepts – Pipeline



A Pipeline is specified as a sequence of stages, and each stage is either a **Transformer** or an **Estimator**.

These stages are run in order, and the input DataFrame is transformed as it passes through each stage.

MLlib Concepts – Pipeline



Algorithms

Supervised Learning Algorithms

Used to estimate real values (total sales, house value etc.) based on continuous variables

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Naïve Bayes Classifier

Supervised Learning Algorithms

- Linear Regression — Used to estimate real values (total sales, house value etc.) based on continuous variables
- Logistic Regression — Used to estimate discrete values (0/1/2, true/false) based on given set of independent variables
- Decision Tree
- Random Forest
- Naïve Bayes Classifier

Supervised Learning Algorithms

- **Linear Regression** — Used to estimate real values (total sales, house value etc.) based on continuous variables
- **Logistic Regression** — Used to estimate discrete values (0/1/2, true/false) based on given set of independent variables
- **Decision Tree** — Used for classification problems based on both categorical and continuous dependent variables
- **Random Forest**
- **Naïve Bayes Classifier**

Supervised Learning Algorithms

Linear Regression	Used to estimate real values (total sales, house value etc.) based on continuous variables
Logistic Regression	Used to estimate discrete values (0/1/2, true/false) based on given set of independent variables
Decision Tree	Used for classification problems based on both categorical and continuous dependent variables
Random Forest	Randon forest is an ensemble of decision trees for getting better accuracy than decision tree.
Naïve Bayes Classifier	

Supervised Learning Algorithms

Linear Regression	Used to estimate real values (total sales, house value etc.) based on continuous variables
Logistic Regression	Used to estimate discrete values (0/1/2, true/false) based on given set of independent variables
Decision Tree	Used for classification problems based on both categorical and continuous dependent variables
Random Forest	Randon forest is an ensemble of decision trees for getting better accuracy than decision tree.
Naïve Bayes Classifier	Classification technique based on Bayes theorem with an assumption of independence between predictors.

Supervised Learning with MLlib — Linear Regression

There are two types of supervised learning algorithms:

1. **Regression**: This predicts a continuous valued output, such as a house price.

2. **Classification**: This predicts a discreet valued output (0 or 1) called label, such as whether an email is a spam or not. Classification is not limited to two values (binomial); it can have multiple values (multinomial), such as marking an e-mail important, unimportant, urgent, and so on (0, 1,2...).

MLlib — Linear Regression

- Linear Regression is a technique used to predict the unknown value of a (dependent) variable from the known value of (independent) variables.
- A dependent variable is a variable to be predicted in a regression model.
- An independent variable is the variable related to the dependent variable in the regression equation.

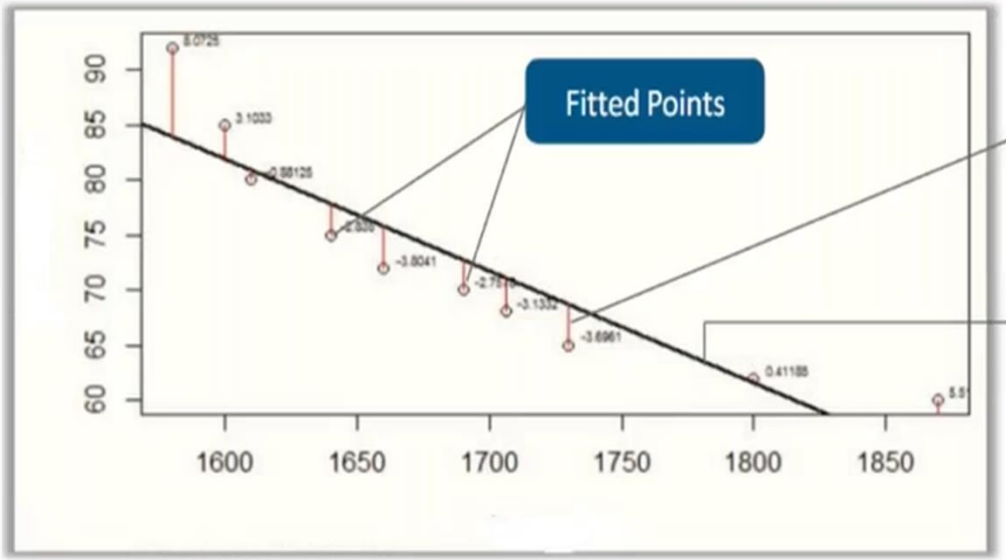
Independent Variables (features)

Area: 4200 sq.ft, Lot size: 31000 sq.ft,
Number of rooms: 4

Dependent Variable

House Price

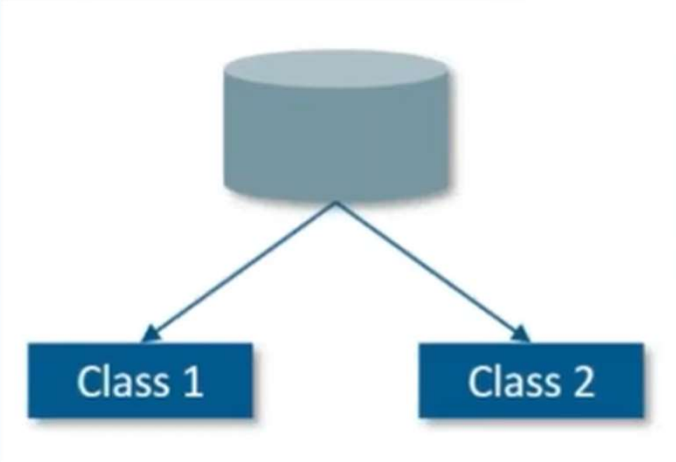
Regression Line



- The regression line simply a single line that best fits the data in terms of having the smallest overall distance from the lines to the point.
- This technique is used for finding the **best fitting line** using the ***least squares method***.

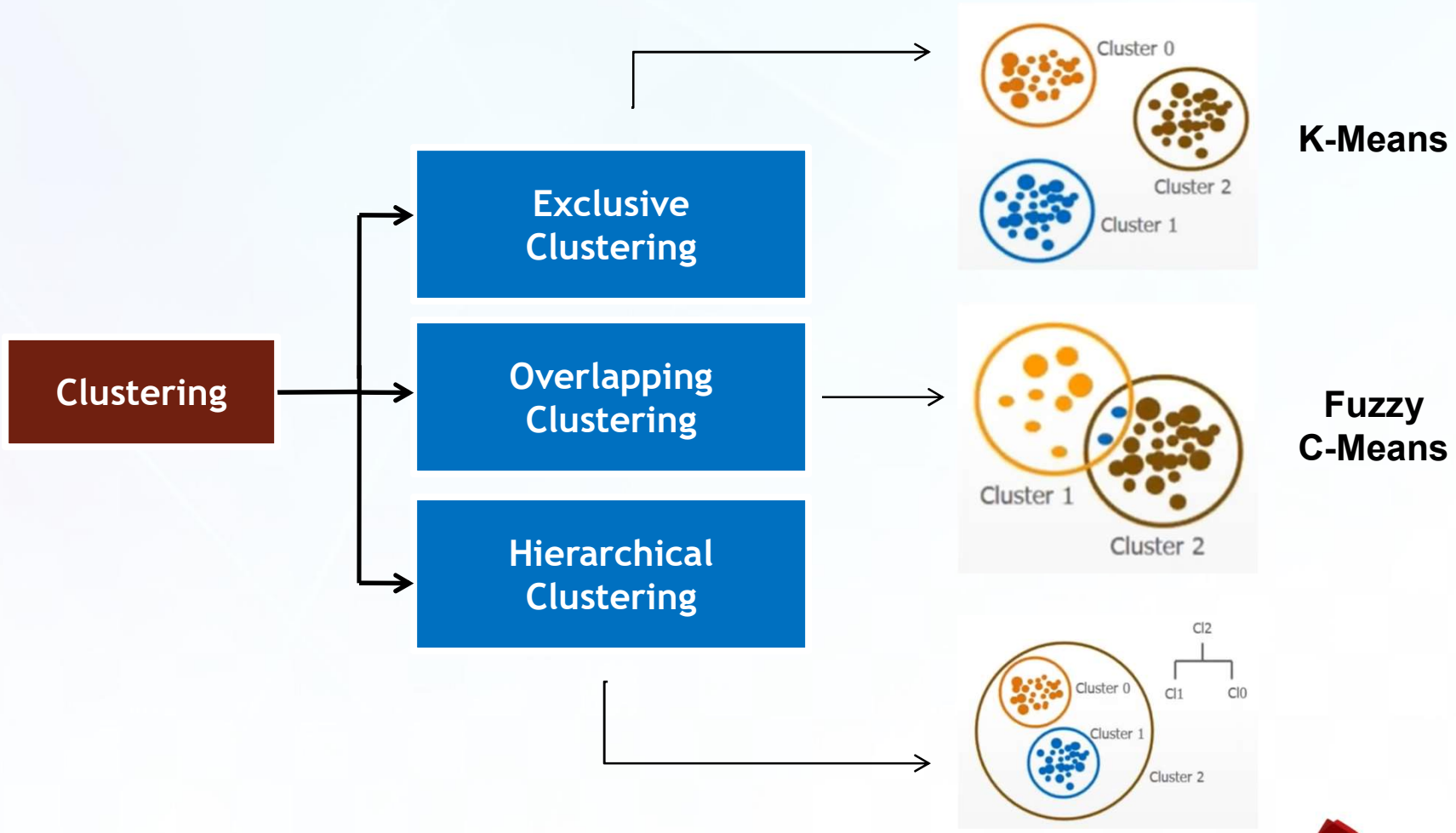
Unsupervised Learning - Clustering

Clustering means grouping of objects based on the information found in the data, describing the objects or their relations

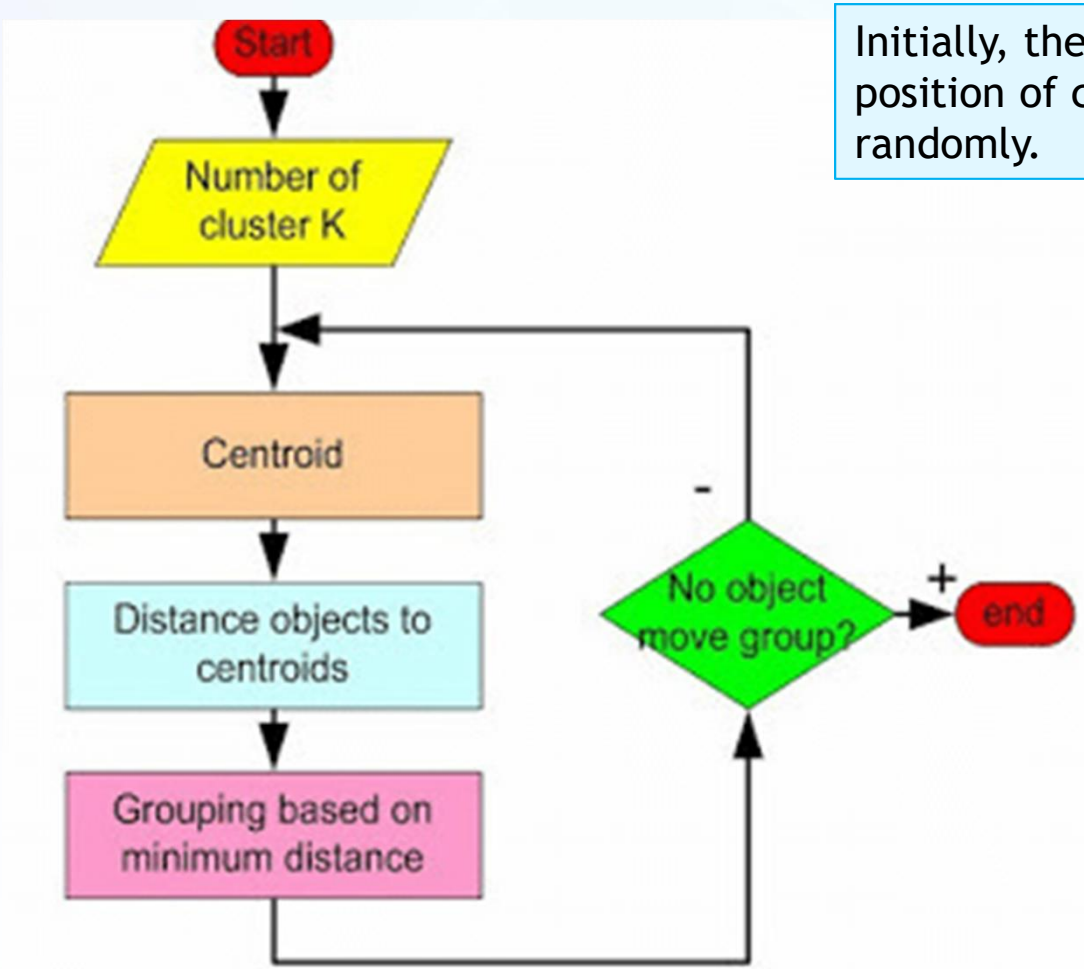


The goal is to organize the data into clusters, wherein the objects in one cluster are similar to each other and different from objects in other clusters.

Types of Clustering

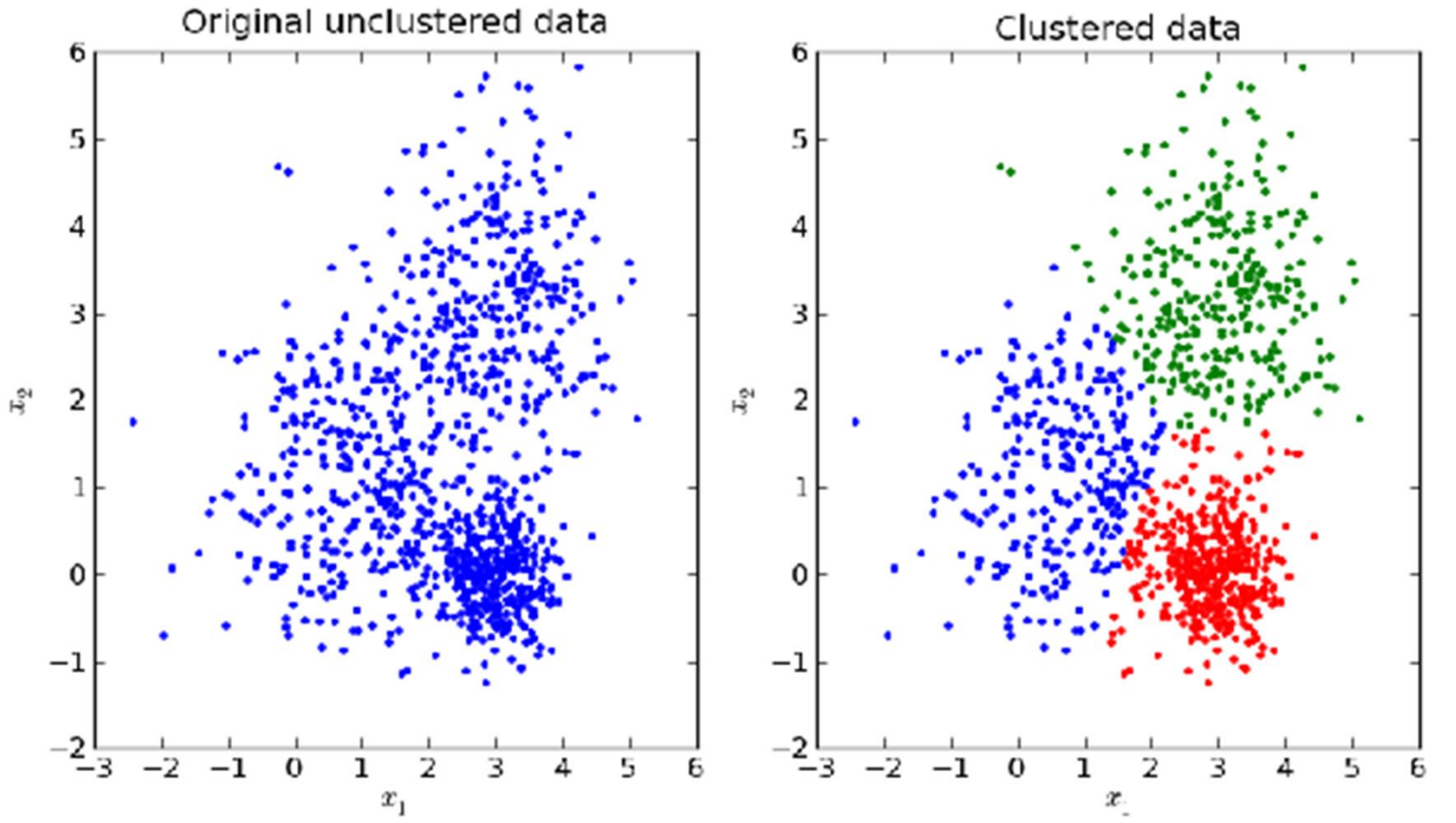


K-Means Clustering



Initially, the number of clusters and position of cluster centroids are chosen randomly.

K-Means Clustering



THANK YOU