1.1 Artificial Intelligence (AI)

Artificial intelligence is the intelligence acquired by a machine to solve problems usually solved by humans.

IBM Watson, MS Cortana, Apple Siri, Autonomous Vehicles are some of the examples of a large number of existing well-known AI applications.

1.1.1 Definition of Artificial Intelligence (AI)

AI is an art of creating machines that perform functions that require intelligence when performed by people. [KUR]Al is playing a leading role in healthcare, manufacturing, e-commerce, retail, social media, logistics and other industry sectors. Some of the reasons for increasing use of AI are the rise in the processing power, availability of data, and new technologies. AI use cases span from monitoring one's home, determining which stock to invest in, helping to decide which recipe to make to helping in choosing your life partner!

AI is an umbrella term which covers the science of making machines intelligent, whether it is a robot, a refrigerator, a television, a car, a firmware or a software component. ML is the subset of AI. ML and AI are often used interchangeably, but they are not the same thing.

AI is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion*

Applications of AI:

- Natural Language Processing
- Speech Processing
- Computer Vision
- Games and Search
- Knowledge Representation
- Reasoning and Logic
- Constraints and optimization

AI History

- 1950s
    a. The goal was to create an artificial brain
    b. The term AI was proposed in Dartmouth Artificial Intelligence conference
- 1980s
    a. "Expert systems" - applying rules on specific topics to new data
    b. Fuzzy logic, Genetic algorithms, Evolutionary Programming, Neural Nets, etc.
- Present
    c. Machine learning, Deep learning, Big Data
    d. Al is still at the stage of Narrow AI

ML is explained in more details in 1.2 Machine Learning (ML)

AI can broadly be categorized as Narrow, General or Super AI.

- **Narrow AI**: Machines that are programmed for carrying out a specific task with limited context. For example, game playing machines, voice assistants and all AI currently.
- **General AI**: Machines with general cognitive abilities are popularly called as Strong AI cases. These AIs can reason and understand their environment as humans do, and act accordingly. For instance, common-sense reasoning. Currently, General AI has not been realized and nobody knows when or if it will become a reality at all.
- **Super AI**: Machines that are capable of replicating human thoughts, ideas and emotions. It is that super state of intelligence where machines will become smarter and wiser than humans. Considering the current state of AI developments, Super AI will not become a reality anytime soon.

- Narrow (ANI)
    - AI that is good at performing a single task, such as playing chess or GO
    - A program trained on GO will not be able to play chess with the same performance levels Also,
    - termed as weak AI
- General (AGI)
    - Artificial Intelligence that can understand and reason with its environment as a human would
    - As a human would, the program will be able to perform a new task without needing significant reprogramming
- Super (ASI)
    - AI becomes much smarter than the best human brains in practically every field, including scientific
    - Creativity, general wisdom and social skills

1.2 Machine Learning (ML)

1.2.1 Definition of ML

Arthur Samuel defined ML as, “A field of study that gives computers the ability to learn without being explicitly programmed.” ML systems learn and improve with experience, and, with time, refine a model that can be used to predict the outcome of questions, based on the previous learning.

diverse scenarios. Notion of searching, scheduling and optimizing fall under the scope of AI, but not necessarily ML.

Some of the technologies used to accomplish AI are:

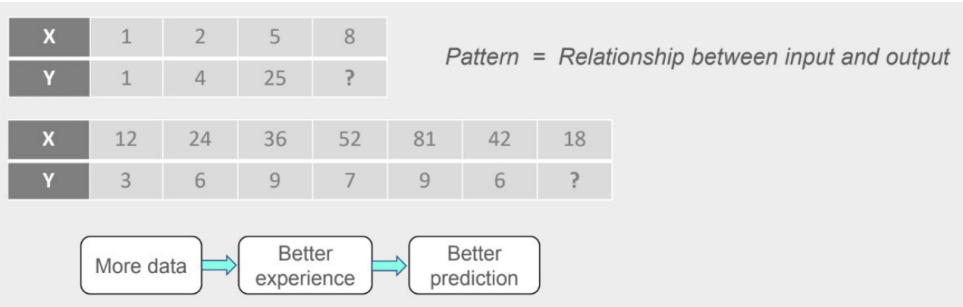
- Machine Learning (ML)
- Natural Language Processing (NLP)
- Robotics
- Speech Processing
- Computer Vision

There are a few ways in which ML algorithms can be categorized:

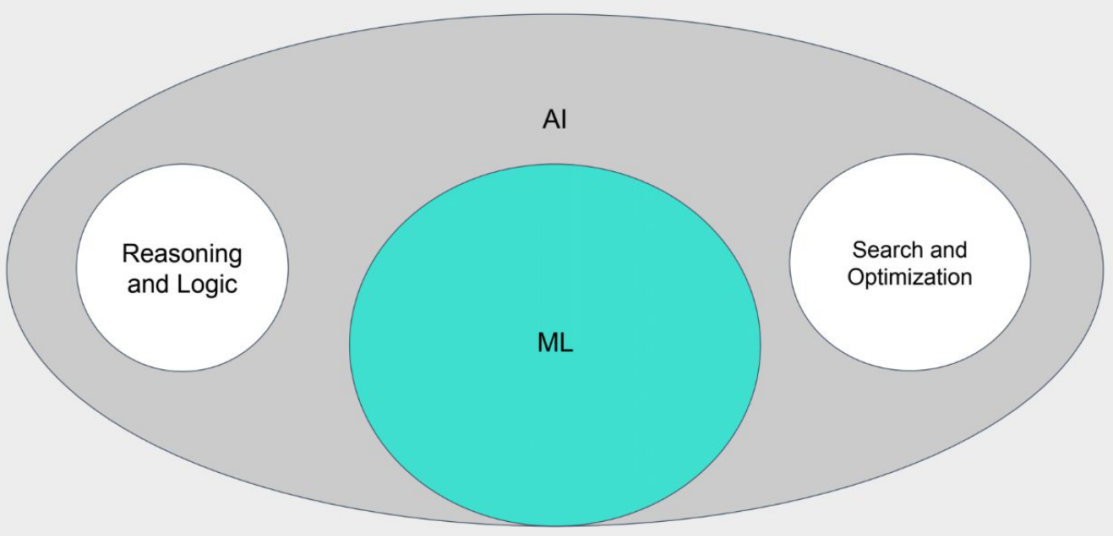
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning is a critical component of AI, which enables machines to automatically learn from experience without explicit programming

Find patterns from the past data and apply the patterns to predict future data



AI and ML



1.2.2 Supervised Learning - Classification and Regression

Supervised Learning: In this kind of learning, the model learns from labeled data during the training phase. The labeled data acts as a trainer/supervisor for the mapping function which infers the relationship between input data and the output label during the training.

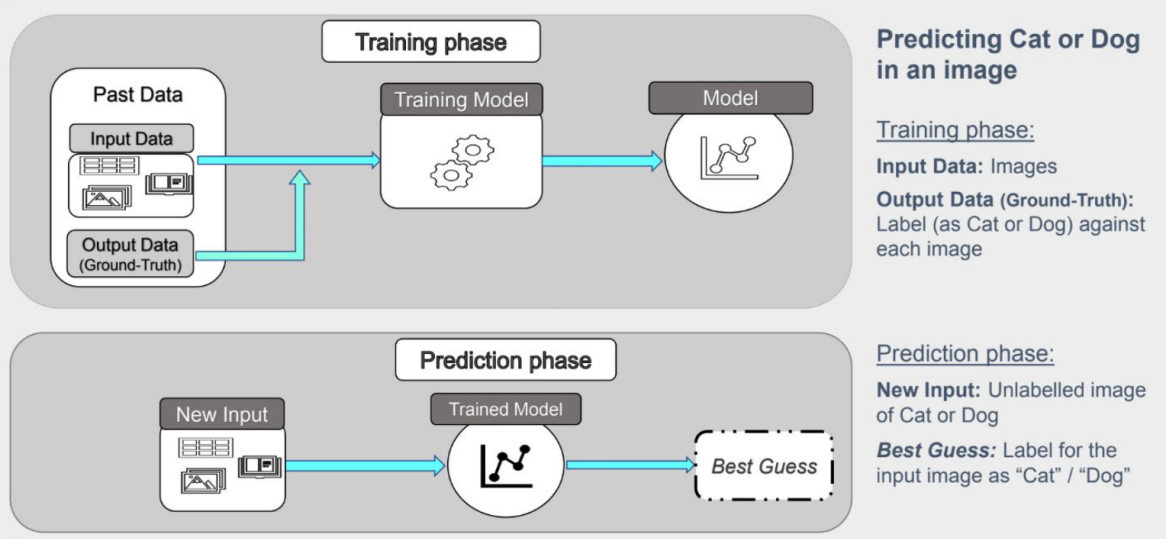
to predict the output which is also labeled. The model is deployed once the output accuracy level is satisfactory.

Problems solved by Supervised Learning are further divided into two categories:

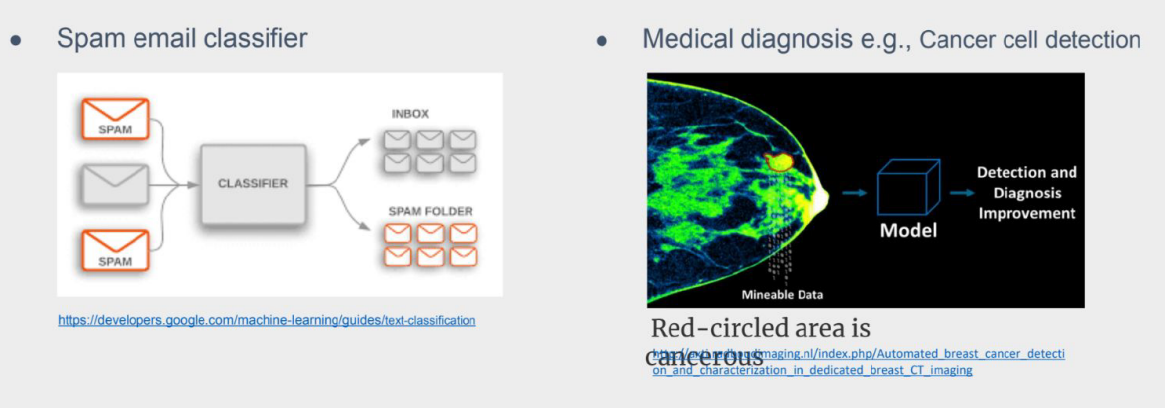
Classification: When the problem requires classifying an input into one of a few pre-decided classes, supervised learning is used. This kind of model is used when the output data is discrete or when the output falls among the number of classes fed during training. Face recognition or object detection in an image are examples of problems that can use classification. Some other applications of classification are spam detection (spam or no spam), the diagnosis of a disease on the basis of the likes of an X-ray, correct identification of road signs by a driver assistance system, etc. Some of the commonly used algorithms for classification are logistic regression, nearest neighbor, support vector machine, and neural nets.

Regression: When the output data is continuous or numeric in nature, e.g., predicting the age/weight of a person, predicting the future price of the stock, etc., regression learning is used. The most commonly used algorithm for this kind of problems is linear regression, a simple algorithm which explains the relation between inputs and the output and inputs as a linear equation. Some other algorithms are logistic regression, support vector machines, Lasso regression, etc.

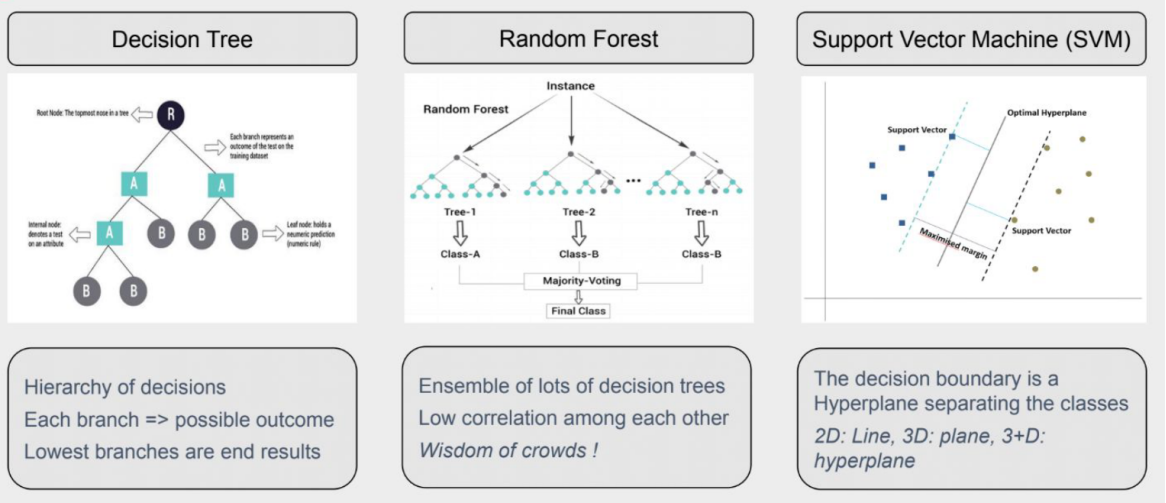
Supervised Learning



Supervised Learning - Classification



Common Algorithms



Quantifying Magic - Accuracy

- Accuracy means % correct predictions (of total number of predictions made)
  - Accuracy = (TP + TN) / (TP +TN + FP +FN)
- Does it reflect the quality of the system accurately?
  - Example - Mammography where the disease occurs < 1%
  - System labels all results as Negative
  - There is 1 positive in 100 cases (TP) which this system misses
  - There are 99 negatives that the system predicts
  - Accuracy = 99/100
- High accuracy
  - Isn't sufficient unless the classes of the dataset have nearly equal number of data points
  - Should not be used when one class dominates the data

Precision

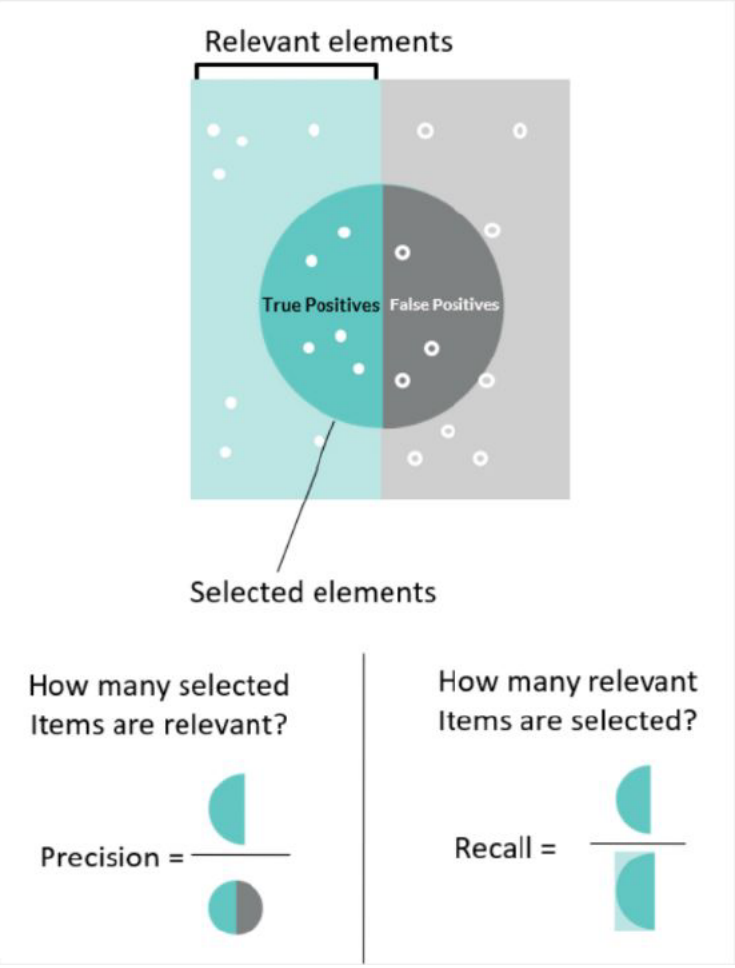
- If it is important to measure that don't have large number of FP (e.g., don't let an innocent suffer even if you have to let go many culprits)
- Precision = TP/ (TP+FP)
  - as FPs increase, precision drops.
- Note
  - You didn't classify any dog as cat (may have classified cat as dog). Precision is high. Are you OK with that? When cat is more important (Ancient Egyptians) for example.
  - Spam filter — classifying emails as Spam (positive) is another example. You want to err on the side of caution by not marking email as spam.
  - When the cost of FP is too high, you want to have high precision

Recall/Sensitivity

- Recall is used when you need a conservative approach, i.e., if you miss
- it, it will go undetected and may cause great harm
    - E.g., Gatekeeper MUST check all the people that (s)he suspe4s, even if a lot of them are found to be innocent. Wrongdoer = positive, innocent= negative
    - So, no wrongdoer should pass as an innocent (negative) i.e., no FN
- Recall = TP/ (TP+FN).
    - Measure of how many positives we did miss
    - FN should be zero for a 100% recall; as FN increases, recall drops
- We want to mark bugs as of high critically, when in doubt, as the cost of not labeling a bug as critical (and consequently not fixing it) is high
- Cancer detection test should not miss any positive cases

Recall or Precision?

- High precision does not mean high recall and vice-versa
- Recall is to minimize False Negatives
- Precision is to minimize False Positives



Specificity

- Specificity = TN / (TN -r FP)
    - TN = Number of correct negative identifications
    - TN + FP = Total number of actual negatives
- Measure how many negatives we classified correctly

- Specificity is the opposite of recall
    - if you switch the definitions of positives and negatives in your problem, Recall becomes

F1— Score

- F1 Score= 2 * (Precision * Recall)/(Precision + Recall)
- F1 Score = Harmonic Mean (Precision, Recall)
- Measure of Precision and Recall together
- Mean is not used because a large number will skew the results whereas Harmonic Mean remains
- closer to the lower number
    - Suppose, the model present poor precision, 0.1, but recall .is high (0.9)
    - Mean of precision & recall = 0.5 (fails to reflect poor precision)
    - Harmonic mean = 2 C (0.1 * 0.9) / (0.1 + 0.9) = 0.18 (low value!)
- Low F1 score reflects possible issues —at least one of Recall Or Precision is

Supervised Learning – Regression

- The output is a continuous quantity

  Examples:

    - Forecast temperature of the day
    - Predict share-prices,
    - Derive the fuel consumption (aka mileage) based on the model, time of day, age of vehicle, etc.

Common algorithms

Mean Absolute Error (MAE)

- lt is defined as the mean of differences between predicted and actual values

$$MAE = \frac{\sum_{t=1}^{N} \left| Predicted_t - Actual_t \right|}{N}$$

- For MAE, the effect of several small errors and the effect of one large error are equal, thus, it fails to represent the presence of large deviations between Predicted and Actual values (large errors)

Root Mean Squared Error (RMSE) and MSE

- RMSE uses the square value of the difference between the estimated and the actual value

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(Predicted_t - Actual_t)^2}{N}}$$

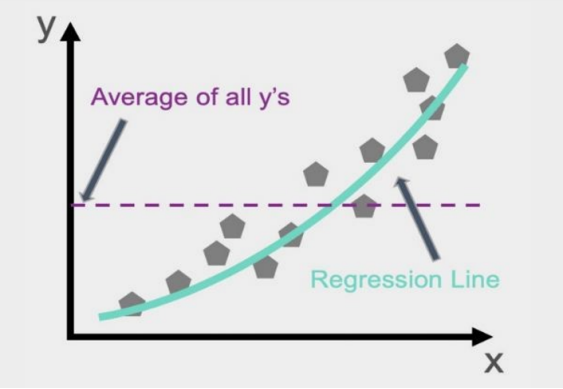
- The square
  - Prevents the cancelation e4ect of the positive and negative errors
  - Results in the amplification of outlier values so these should be removed from the data
- MSE is the square of RMSE
- The RMSE value is dependent on the unit of measurement
  - If, for instance, while predicting the fuel mileage of a vehicle engine, mileage is measured in miles per gallon (mpg) or kilometers per liter (kpl), the RMSE values will differ (lor a same model/algorithm)
- R-Square — Normalized Regression Metric

R-Square — Normalized Regression Metric

- A normalized metric
- Independent of the unit of the variable
- Measures how much better the model regression line is than the average of the output variable
- Formula for R-square is:

$$R^2 = 1 - \frac{SSE}{SST}$$

$$= 1 - \frac{MSE}{Variance}$$



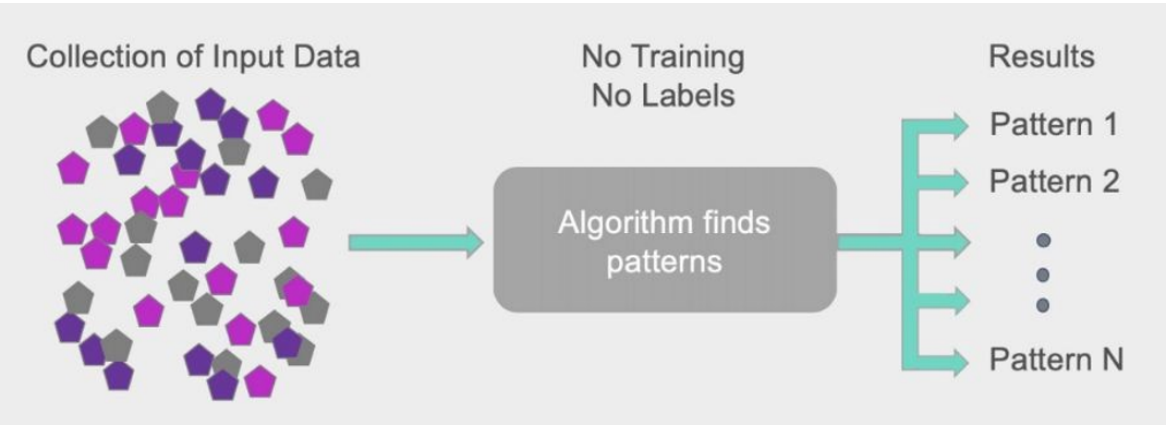
- SSE = sum of squared errors from the regression line
- SST = sum of squared deviations from mean of the output variable
- MSE = Mean Squared Error

- The closer is the value of R-square to 1, the be2er the model is

1.2.3 Unsupervised Learning – Clustering and Association

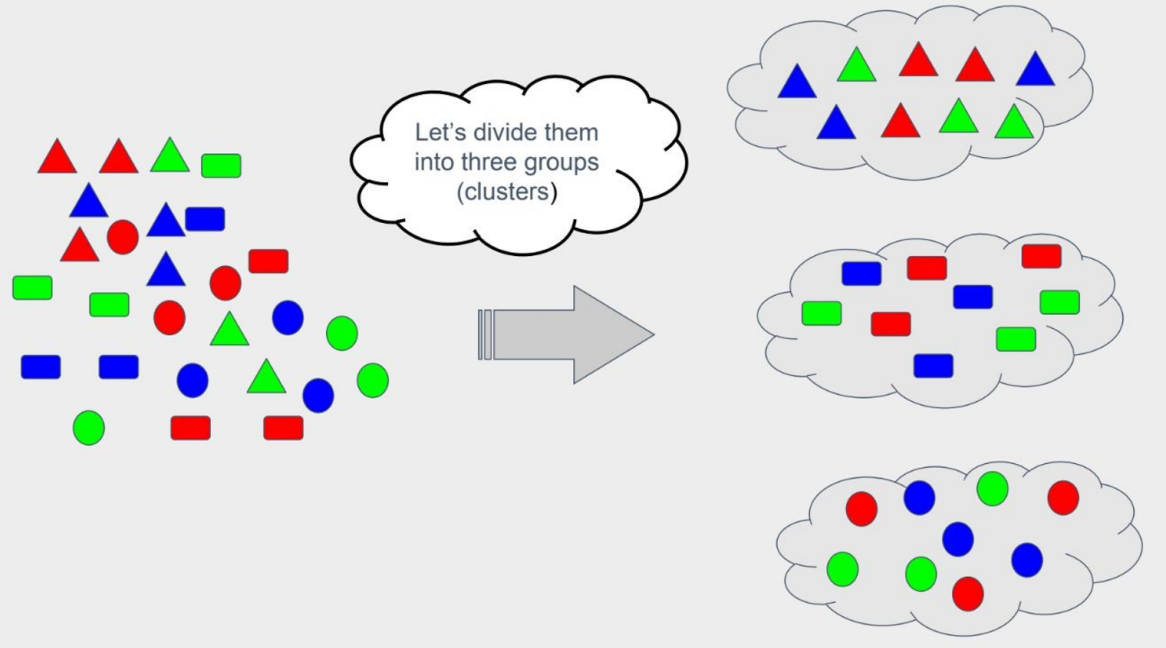
Unsupervised Learning: Clean, labeled data are not readily available all the time, so that certain problems need to be solved without an explicitly labeled training set. This kind of ML where no labeled data is provided explicitly is called Unsupervised Learning. The objective in such problems is to learn the pattern and structure of input data without any associated labels.

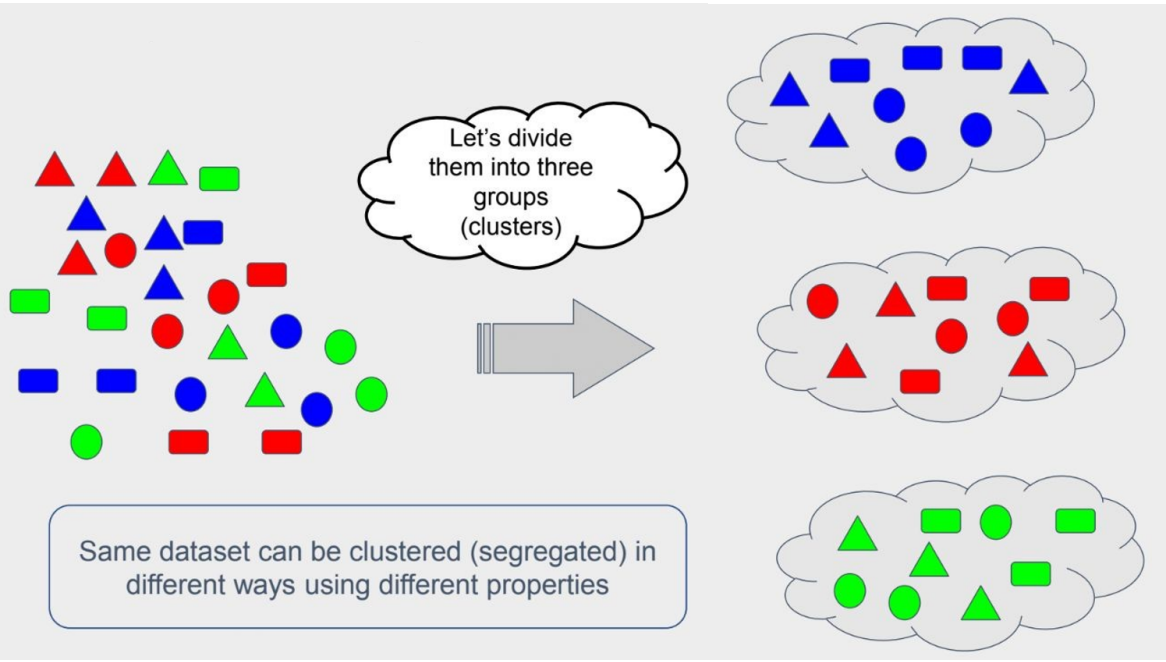
- No notion of Output Variable
- No need for training; the model learns without labels from data
- The: purpose is to learn the patterns and structure of the data without associated labels
- Used in
  - Pattern detection
  - Descriptive modeling



Unsupervised Learning is further classified in the following two methods, based on the type of outputs:

Clustering: This Unsupervised Learning model groups the input data based on some common characteristics or attributes. Input data with similar attributes (not labeled) are grouped in one cluster. Thus, the outputs are clusters of input data. For instance, customer segmentation in market analysis.





Association: Association Rule Mining finds interesting relationships or dependencies among the data attributes. The discovery of interesting associations provides a source of information often used for decision making.

For example, market-basket data analysis, product recommendation system based on learnings derived from customer shopping behavior are good examples of association rule-based modeling.

Clustering and Association.

Inertia

- Inertia is used to assess the quality of an unsupervised clustering model.
- E.g., the lemons segregated into one group also vary in their sizes and it is possible to segregate them further into finer groups/clusters Or combine two groups into one if they do not look too different.
- How to determine the true ideal number of clusters ? lnertia helps answering this

difference is modeled as size of the cluster e.g., a smaller or tight cluster means that objects within are highly similar and vice-versa

- lt is also known as "within-cluster sum of squares”

Adjusted Rand Score

- Another quality metric for unsupervised clustering
- You already have truth about cluster assignment for the data points, use Adjusted Rand score to assess quality of clustering algorithm
- lt is the measurement of the similarity between cluster assignments as discovered by the model and those available as ground truth (in reality)
- lt can also be used to determine how similar the two clustering models are
- The metric value is between 0 and 1
    - Zero means that the two data clustering do not agree on any pair of points
    - One means that both the data clustering are exactly the same.

When to use Inertia Or Adjusted Rand Score

- If you have the gold standard of expected ground truth labels, !

  => Adjusted rand index is preferred

- Else,

  =>Go with inertia

1.2.4 Reinforcement Learning

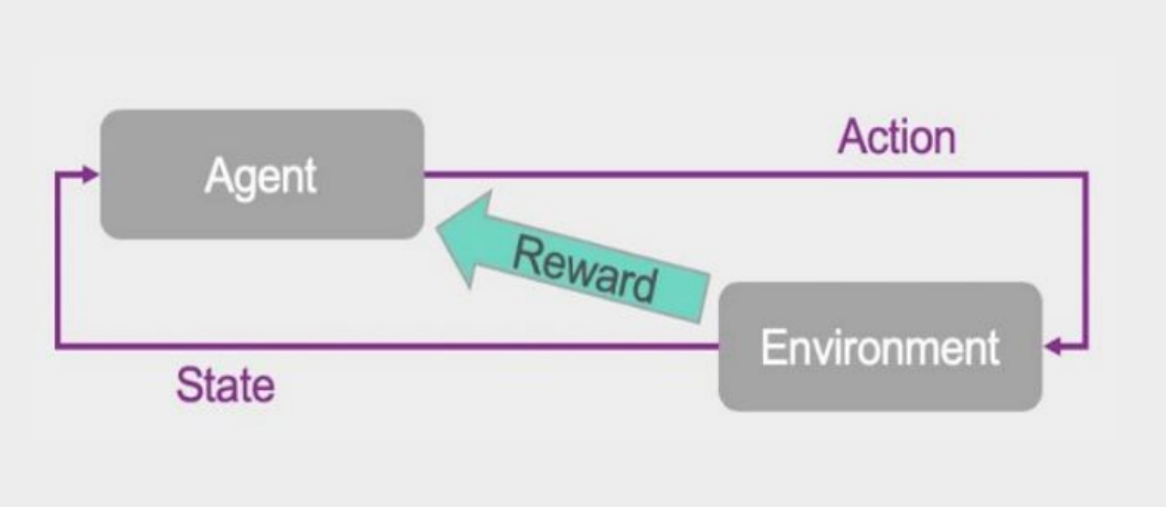
Recall the definition and applications of Reinforcement

Learning (RL).

It is a type of ML where an agent (algorithm) learns by interacting with the environment in an iterative manner and thereby learns from experience. The agent is rewarded when it makes a right decision and penalized when it makes a wrong one. This reward and penalty-based learning is thus defined as ‘reinforcement learning’ (RL). Setting up the proper environment, choosing the right strategy for the agent to meet the desired goal, and designing a reward function, are some of the key challenges in implementing RL. Robotics, Autonomous Vehicles, and Chatbots are examples of applications that can use RL.

- How RL works
    - Agent (algorithm) takes an action in a state
    - Receives a reward or penalty as a feedback from its Interactive environment (user)
    - The agent learns from the environment in an iterative manner
- RL is used for decision making
    - Robotic vision

    - Driverless cars, etc.
- Common algorithms
    - Q-Learning
    - Temporal Difference (TD)
    - Deep Adversarial Networks



1.3 Deep Learning (DL)

1.3.1 Deep Learning and the Types of Neural Networks

Deep Learning (DL) refers to the systems gaining experience from massive data sets. DL uses Artificial Neural Networks (ANN) to analyze large data sets, e.g.

Autonomous Vehicles, Large Text Processing, and Computer Vision applications among others. [AG1]

DL is a subset of ML and ML is a subset of AI. DL uses the same types of learning (Supervised, Unsupervised and Reinforcement Learning) as ML.
ML vs DL

- In ML, feature extraction is manual whereas DL detects the features on its own
- As the amount of data to be analyzed increases, DI network's performance far exceeds other on-DL mechanisms.

Neural Networks For DL

- Artificial Neural Network (ANN):
  It is similar to the 'Neuron' (a basic unit in the brain), it acts on input stimulus and produces an output signal
- Deep Neural Network (DNN):
  When a neural net has two or more hidden layers

Basics of Deep Learning (DL)

- A subset of Machine Learning
- Learns a hierarchy of representations automatically from data

- The first level of representation is called the input layer and the last one is called the output layer
- Any layer of representation between the input and output layer are called the hidden layers

Artificial Neural Networks: Artificial Neural Networks (ANN) are inspired by the architecture of the human brain. 'Neurons', as the basic unit of ANN, act upon the input stimulus and produce the output signal. The input goes through the layers of activation functions to generate the output. These layers form a mesh like network.

Every ANN has at least two layers – input and output layers. All the layers between these two layers are called hidden layers. Some of the various types of neural networks are:

Deep Neural Network (DNN): Deep Neural Network (DNN) is an ANN with two or more hidden layers.

Convolutional Neural Network (CNN): Convolutional Neural Network (CNN) is an ANN that emerged from the study of the brain's visual cortex, and they have been used in image recognition since the 1980s. Unlike other neural networks, CNNs work directly on input images without serializing/ vectorizing an input image and extracting features by filters. CNNs power image search services, autonomous vehicles, automatic video classification systems, and more.

Recurrent Neural Network (RNN): These ANNs can predict the future of time series problems. They follow a sequential approach on series of input data of arbitrary length rather than inputs of fixed length as in other neural networks.

Each input and output are independent of all the other layers. The feedback from the output layer is fed to the same network recurrently, till the right level of confidence is achieved. RNNs can analyze time series data such as stock prices, and tell you when to buy or sell. In autonomous vehicles, they can anticipate trajectories and help avoid accidents.

Neural Networks for DL

- Convolutional Neural Network (CNN): They power image search services, self-driving cars, etc. CNNs work directly on input images rather than focus on derived values.
- Recurrent Neural Network (RNN): The output is copied and sent back into the RNN. For making a decision, it considers the current input and the output that it has learned from the previous input
  RNNs are useful for Learning sequences, time-series data.

1.4 Stages of the ML Process

A typical ML project follows all the stages of the Cross Industry Standard Process for the Data Mining (CRISP-DM) framework – an industry standard, and a flexible framework.

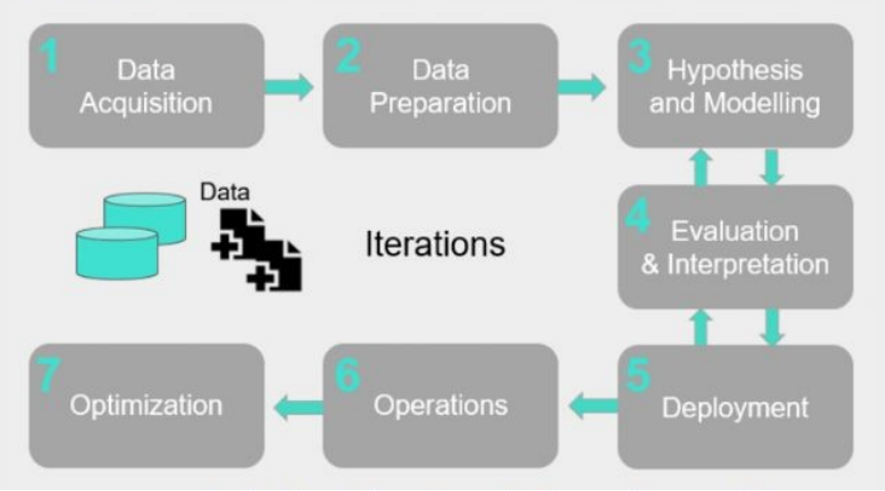
1.4.1 Stages of the ML Process – CRISP-DM Process

The Cross Industry Standard Process for Data Mining (CRISP-DM) model conceived in 1996 is used to gain knowledge and work with data

CRISP-DM has traditionally six stages in the data mining life cycle. It has been customized to meet the requirements of ML projects, by adding a seventh stage.

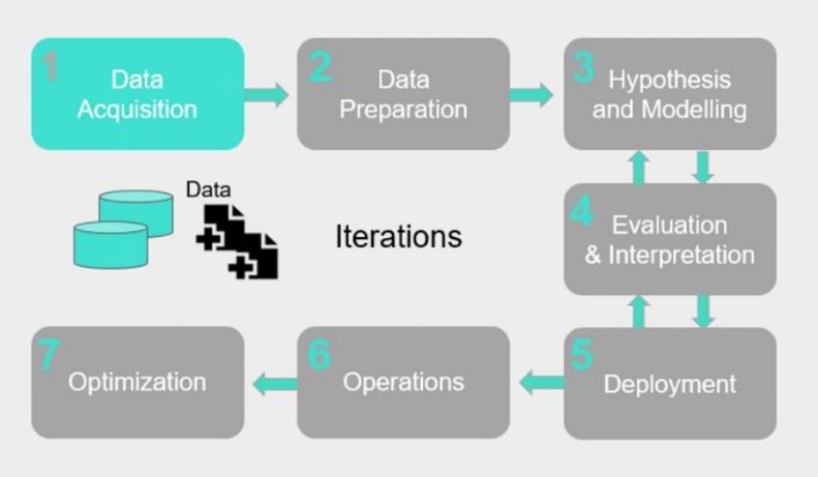
The seven stages of the CRISP-DM framework for ML are: [DSC1] [SMU]

1. Data Acquisition
2. Data Preparation
3. Hypothesis and Modeling
4. Evaluation and interpretation
5. Deployment
6. Operations
7. Optimization



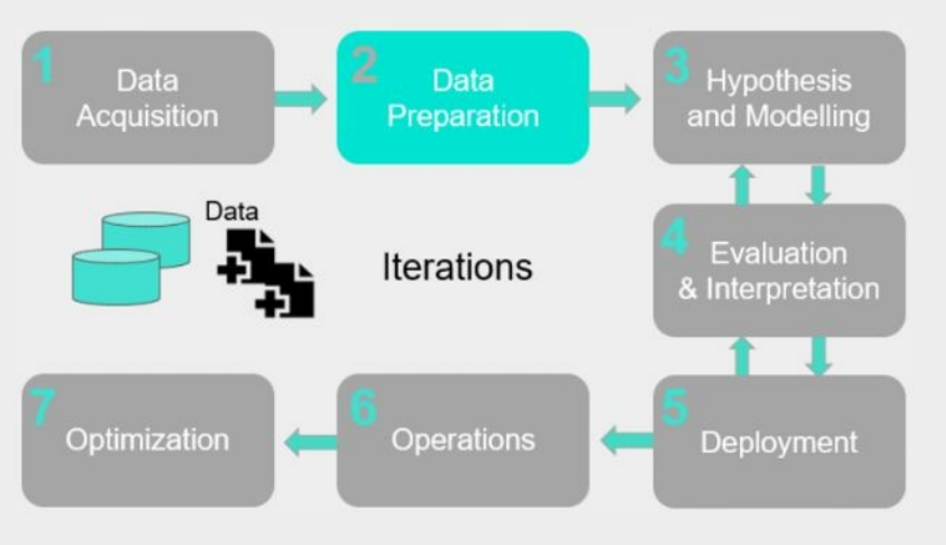
1. Data acquisition: Gather data from all internal and external sources (for example databases, CSV files, social media, etc.)

- Collect information from both internal and external sources, including
  - Databases
  - Scraping off the web, including social media
  - Putting together the aIready collected data into an excel or CSV file
- Data extraction schedules may also be set up to retrieve and store data at regular intervals
- Large data set and data from diverse sources are better
- Example: Reading a CSV file in PÇhon
  - data=pd.read_csv('obj.c:sv')



2. Data preparation: Clean the raw data and reshape it. New attributes are created with feature engineering, a process for creating new variables from existing data. Dimensionality reduction, data imputation, null value treatment for the missing values, etc., are some of the methods involved in data preparation.

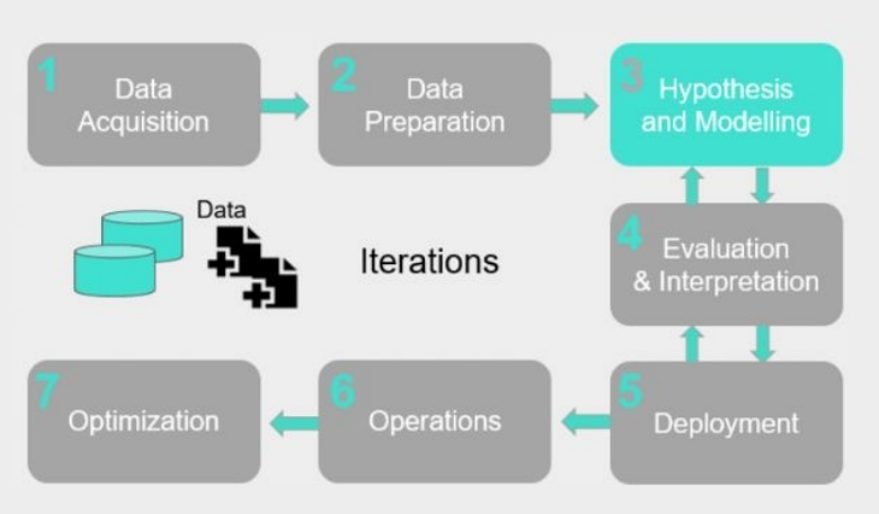
- This step Involves
  - Cleaning the data
  - Reshaping it in the required format
  - Creating a new attribute as a combination of two or more dependent attributes
  - lt may involve imputation — replacing null values
  - lt may also involve reducing dimensions by using techniques such as PCA



3. Modeling: Select the model or algorithm, divide the available data into training set and testing set. Models are obtained by executing ML algorithms on the training data set. Use the testing data set to evaluate and enhance the performance of the model until satisfactory performance is achieved.

- Core ML steps
  - Choose the model and algorithm for the target model to be deployed

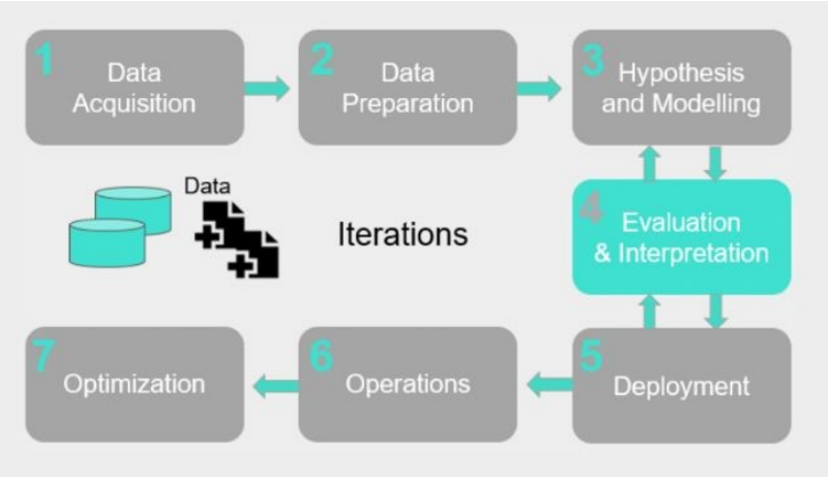
validate the results respectively
    - Repeat the process with different algorithms until the desired accuracy is reached
    - Baseline the chosen high accuracy algorithm and model for deployment



4. Evaluation: Evaluate the model on various metrics (discussed in 3.2 Metrics) and baseline it before it goes for final deployment.

Evaluation and Interpretation

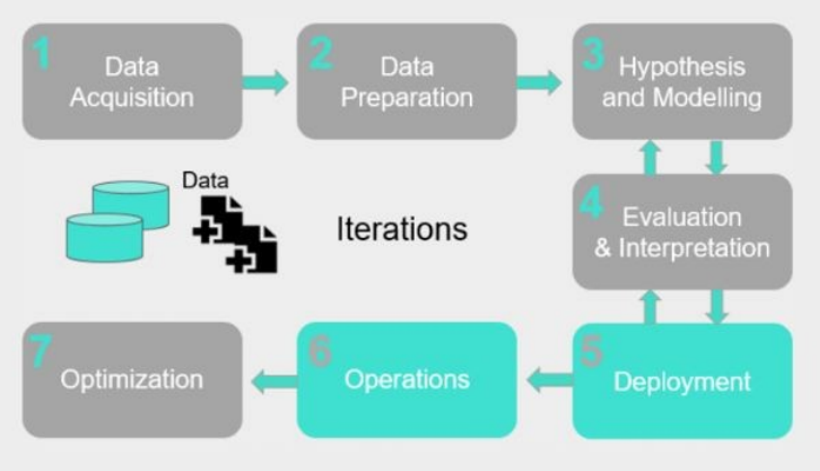
- A portion of the acquired data is used to evaluate the accuracy of the solution
- Evaluation is done by checking for the diverse combinations of hyperparameter values.
- Interpretation means providing data and answers as solution to the business problems
- Interpretation helps get useful answers for the organization



5. Deployment: Deploy and monitor the baselined model for metrics in the production environment.

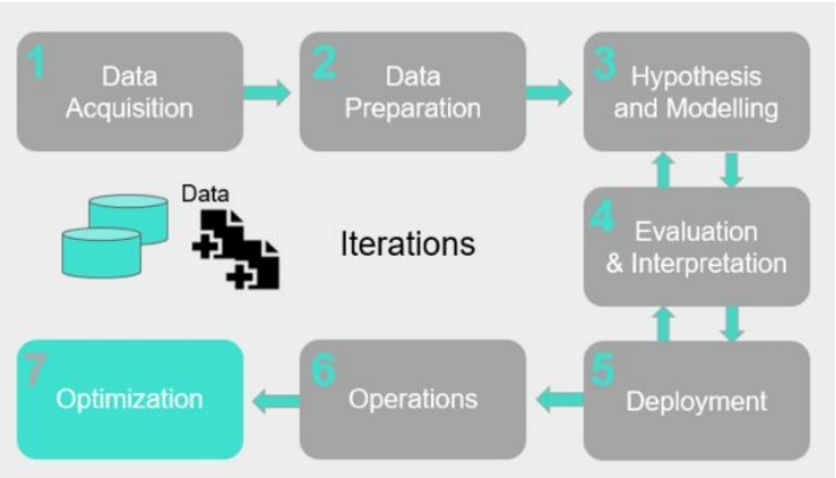
6. Operations: Carry out regular maintenance and operations. Regenerate and refine the model when the metrics fall below a certain threshold.

- Deployment of the baselined model, post evaluation
    - Typically deployed in a staging or pilot environment before full-scale/production deployment
    - infrastructure requirement may be a key concern here
- The Operations stage involves
    - Continuing to monitor the deployment for its behavior and performance
    - Regenerating the modal if the model performance/accuracy falls below the threshold because of concept drift



7. Optimization: The deployed solution may be replaced due to concept drift (see 6.1.3 Risk of Concept Drift (CD)), as better algorithms become available, or because of some major failures in performance.

- The deployed solution may need to be replaced because of
    - Falling performance because of due to concept drift
    - The need to add new data sources and to retraining the model
- The model may also be improved by using different/better algorithms and the retained model is then deployed
- Optimization will need a recalibration of the model, hence there will be retraining in a batch mode again to get a new model

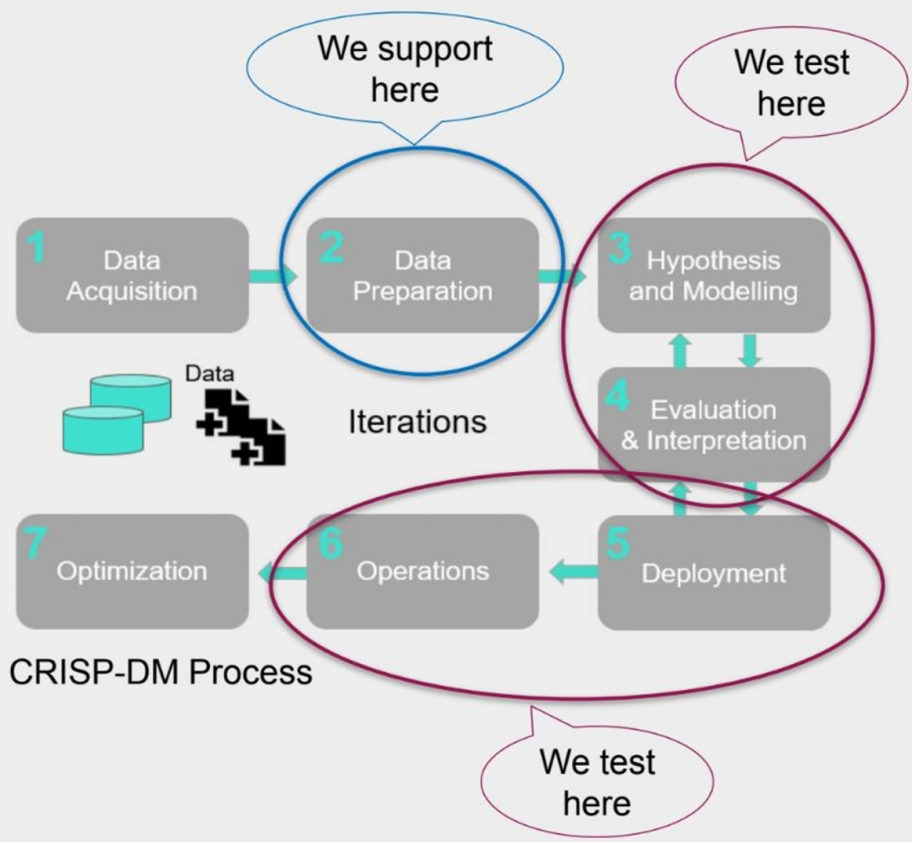


Steps 1-4 can be classified as part of the offline phase, the output of which is the trained model. Steps 5-6 are a part of the online phase, where the model trained in the offline

The optimization step involves re-execution of steps 1-6.

Testing in the ML Lifecycle

- Stages of ML lifecycle
  - Data acquisition
  - Data Preparation
  - Hypothesis and modeling
  - Evaluation and interpretation
  - Deployment
  - Operation
  - Optimization



1.4.2 Steps for the Identification of the ML Problem Type

It is important to understand the problems that we are trying to solve and the type of learning required to solve those problems. One way we can identify the ML problem type is discussed below:

1. If the problem involves the notion of multiple states, and involves moves at each state, then explore RL.

2. If there is an output variable it is supervised learning.

2.1. In the case that the output is discrete and categorical, it is a classification problem.

2.2. In the case that the output is numeric and continuous in nature, then it is a regression problem

learning.

3.1. If the problem involves grouping similar data, then it is a clustering problem.

3.2. If the problem involves finding co-occurring data items, then apply association rule mining

3.3. If the raw data is unstructured, extracting features automatically can be explored with deep learning algorithms.

The prerequisite to the above steps is that there should be enough data available for the analysis of the appropriate ML problem type