**Machine learning**

**Types Of Machine Learning Algorithms**

**Supervised**

**Regression**

1. linear regression
2. Decsion Tree
3. Random Forest

**Classification ( Where we predict in like yes or No )**

1) Logistic regression

2) Decsion Tree

3) Random Forest

4) naive bayaes

5) XGboost

6) KNN

7) SVM

**Unsupervised learning ( where the data is not labeled )**

Clustering ( use to organized the data use in ecom website )

\* K- means

\* Hierarchical

\* Mean shift

\* Density based

\* Dimensionality Reduction ( process to reduce the dimension of your feature set )

\* feature elimination

\* feature extraction

**Reinforcement Learning**

**Decision Tree:**

disadvantages of decision Tree

Overfitting

High variance ( unstable with small change in data )

Low biased Tree ( complicated trees make difficult to predict new data )

entropy ( measure of randomness in dataset )

information Gain ( measure of decrease in entropy in each step )

leaf node ( end noot )

**entropy ( measure of randomness in dataset )**

information Gain ( measure of decrease in entropy in each step )

leaf node ( end noot )

**For classifiers  
**criterion***{“gini”, “entropy”}, default=”gini”***

***For reg***

****criterion***{“squared\_error”, “friedman\_mse”, “absolute\_error”, “poisson”},***

**Random Forest**

advantages

no overfitting

good for large datasets

maintain acuracy with also missing data in dataset

### 1. Features which make predictions of the model better

1a max features:

1b n\_estimators:

1c min\_sample\_leaf

****max\_features***{“auto”, “sqrt”, “log2”}, int or float, default=”auto”***

The number of features to consider when looking for the best split:

If int, then consider max\_features features at each split.

If float, then max\_features is a fraction and round(max\_features \* n\_features) features are considered at each split.

1. Auto/None : This will simply take all the features which make sense in every tree.Here we simply do not put any restrictions on the individual tree.
2. sqrt : This option will take square root of the total number of features in individual run. For instance, if the total number of variables are 100, we can only take 10 of them in individual tree.”log2″ is another similar type of option for max\_features.
3. 0.2 : This option allows the random forest to take 20% of variables in individual run. We can assign and value in a format “0.x” where we want x% of features to be considered.

****How does “max\_features” impact performance and speed?****

Increasing max\_features generally improves the performance of the model as at each node now we have a higher number of options to be considered. However, this is not necessarily true as this decreases the diversity of individual tree which is the USP of random forest. But, for sure, you decrease the speed of algorithm by increasing the max\_features. Hence, you need to strike the right balance and choose the optimal max\_features.

N\_ estimators:

This is the number of trees you want to build before taking the maximum voting or averages of predictions. High for better but depend on the power of your processor

Min Sample leaf:

A smaller leaf makes the model more prone to capturing noise in train data. Generally I prefer a minimum leaf size of more than 50.

**2. Features which will make the model training easier**

#### 2.a. n\_jobs :

%timeit

model = RandomForestRegressor(n\_estimator = 100, oob\_score = TRUE, n\_jobs = -1,random\_state =50, max\_features = "auto", min\_samples\_leaf = 50)

**Classification ( Where we predict in like yes or No )**

1) Logistic regression

2) Decsion Tree

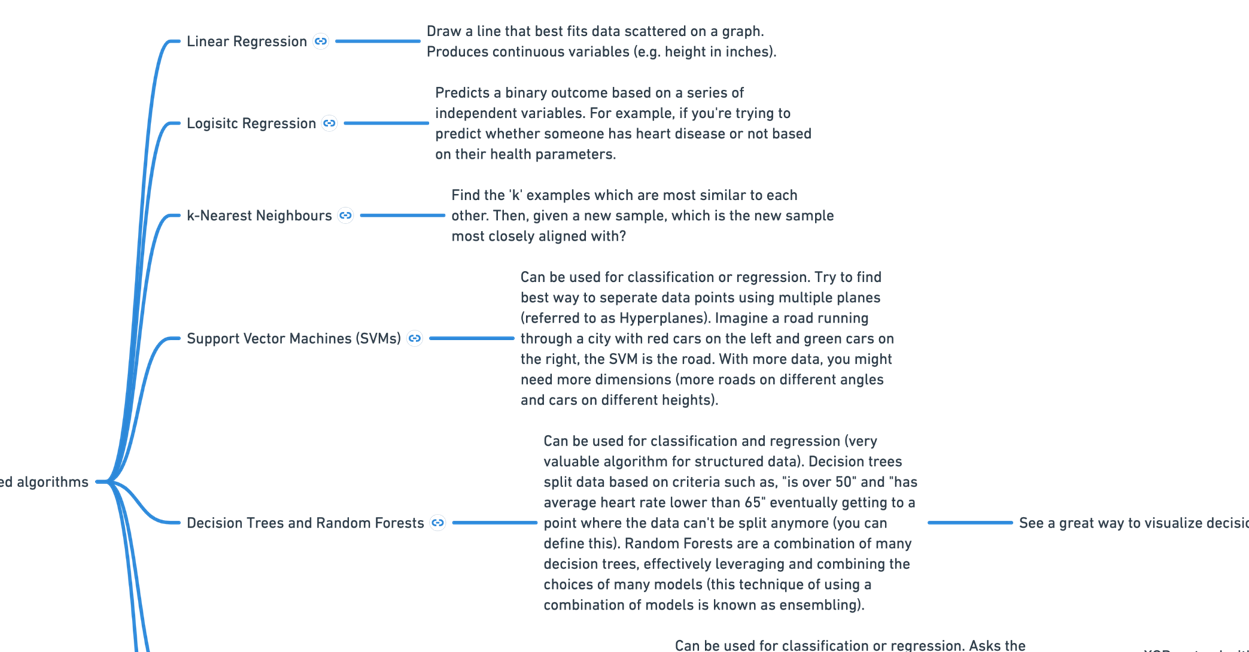
3) Random Forest

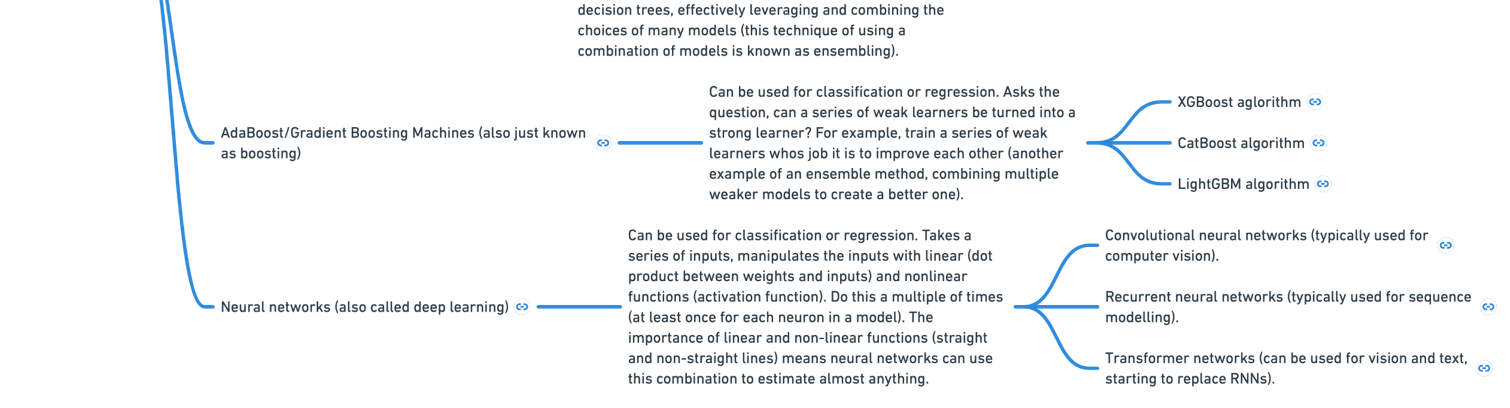
4) naive bayaes

5) XGboost

6) KNN

7) SVM





## **Naive Bayes**

[Conditional probability](https://www.analyticsvidhya.com/blog/2020/08/probability-conditional-probability-monty-hall-problem/" \t "https://www.analyticsvidhya.com/blog/2021/09/naive-bayes-algorithm-a-complete-guide-for-data-science-enthusiasts/_blank) is defined as the likelihood of an event or outcome occurring, based on the occurrence of a previous event or outcome. Conditional probability is calculated by multiplying the probability of the preceding event by the updated probability of the succeeding, or conditional, event.

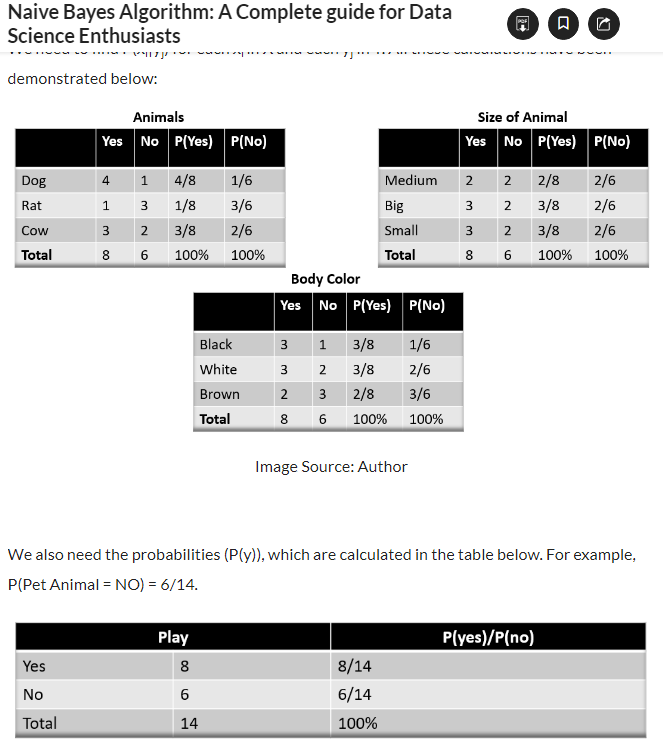
## **Assumptions of Naive Bayes**

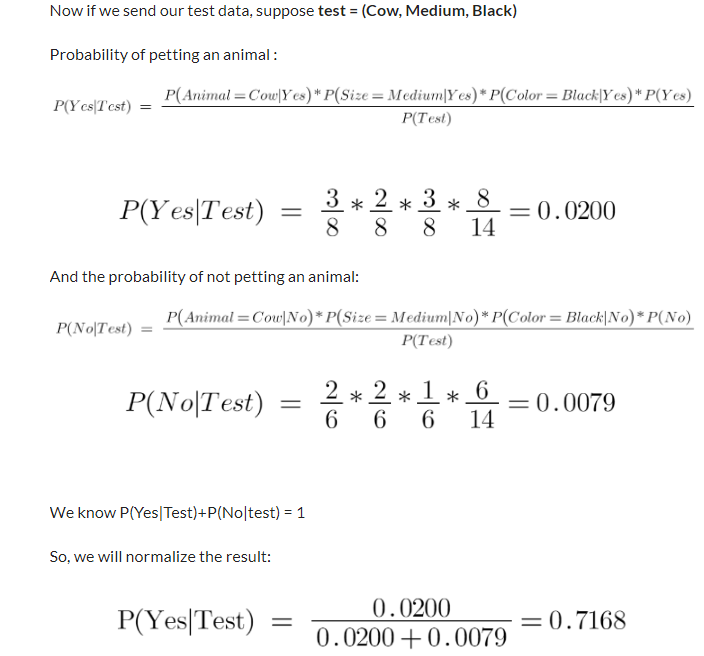
· All the variables are independent. That is if the animal is Dog that doesn’t mean that Size will be Medium

· All the predictors have an equal effect on the outcome. That is, the animal being dog does not have more importance in deciding If we can pet him or not. All the features have equal importance.

We should try to apply the Naive Bayes formula on the above dataset however before that, we need to do some precomputations on our dataset.

We need to find P(xi|yj) for each xi in X and each yj in Y. All these calculations have been demonstrated below:





## Pros and Cons

### Pros

* Simple, Fast in processing, and effective in predicting the class of test dataset. So you can use it to make real-time predictions for example to check if an email is a spam or not. Email services use this excellent algorithm to filter out spam emails.
* Effective in solving a multiclass problem which makes it perfect for identifying Sentiment. Whether it belongs to the positive class or the negative class.
* Does well with few samples for training when compared to other models like Logistic Regression.
* Easy to obtain the estimated probability for a prediction. This can be obtained by calculating the mean, for example, print(result.mean()).
* It performs well in case of text analytics problems.
* It can be used for multiple class prediction problems where we have more than 2 classes.

### Cons

* Relies on and often an incorrect assumption of independent features. In real life, you will hardly find independent features. For example, Loan eligibility analysis would depend on the applicant’s income, age, previous loan, location, and transaction history which might be interdependent.
* Not ideal for data sets with a large number of numerical attributes. If the number of attributes is larger then there will be high computation cost and it will suffer from the Curse of dimensionality.
* If a category is not captured in the training set and appears in the test data set then the model is assign 0 (zero) probability which leads to incorrect calculation. This phenomenon is referred to as ‘Zero frequency’ and to overcome ‘Zero frequency’ phenomena you will have to use smoothing techniques.

## **Endnotes**

Naive Bayes algorithms are mostly used in face recognition, weather prediction, Medical Diagnosis, News classification, Sentiment Analysis, etc.

<https://www.analyticsvidhya.com/blog/2021/09/naive-bayes-algorithm-a-complete-guide-for-data-science-enthusiasts/>

**SVM**

<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

it separate data and set aline between them

Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’

kernal // function is used to convert 1D data set in to 2D dataset

use when second type of data is in line of first type of data

more complicated use

transform // to convert 2D dataset in to 3D

## Pros and Cons associated with SVM

* ****Pros:****
  + It works really well with a clear margin of separation
  + It is effective in high dimensional spaces.
  + It is effective in cases where the number of dimensions is greater than the number of samples.
  + It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* ****Cons:****
  + It doesn’t perform well when we have large data set because the required training time is higher
  + It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping
  + SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of Python scikit-learn library.

## When do we use KNN algorithm?

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output

2. Calculation time

3. Predictive Power

when data is overlaps logistic regression is not best to use

if alot of overlaps use KNN ALGO

IF not or less then use Random forest Decision tree

**Unsupervised learning ( where the data is not labeled )**

Clustering ( use to organized the data use in ecom website )

\* K- means

\* Hierarchical

// https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

for more understanding use above link

type of hier

Agglomerative hierarchical clustering

Divisive Hierarchical clustering

\* Mean shift

\* Density based

\* Dimensionality Reduction ( process to reduce the dimension of your feature set )

\* feature elimination

\* feature extraction

## Introduction to K-Means Clustering

1. means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

***The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.***

### Steps for K mean

### Step 1: Choose the number of clusters k **Step 2: Select k random points from the data as centroids**

**Step 3: Assign all the points to the closest cluster centroid**

### Step 4: Recompute the centroids of newly formed clusters

### Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations are reached

This is where K-Means++ helps. ****It specifies a procedure to initialize the cluster centers before moving forward with the standard k-means clustering algorithm.****

he steps to initialize the centroids using K-Means++ are:

1. The first cluster is chosen uniformly at random from the data points that we want to cluster. This is similar to what we do in K-Means, but instead of randomly picking all the centroids, we just pick one centroid here
2. Next, we compute the distance (D(x)) of each data point (x) from the cluster center that has already been chosen
3. Then, choose the new cluster center from the data points with the probability of x being proportional to (D(x))2
4. We then repeat steps 2 and 3 until k clusters have been chosen

***The maximum possible number of clusters will be equal to the number of observations in the dataset.***

# **Hierarchical Clustering**

## Types of Hierarchical Clustering

There are mainly two types of hierarchical clustering:

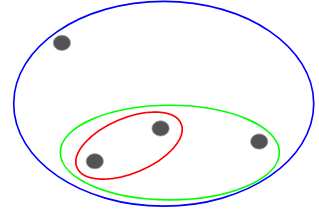
1. Agglomerative hierarchical clustering
2. Divisive Hierarchical clustering

Let’s understand each type in detail.

### Agglomerative Hierarchical Clustering

We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning:

Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/05/Screenshot-from-2019-05-15-13-31-06.png)

We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as ****additive hierarchical clustering.****

### Divisive Hierarchical Clustering

Divisive hierarchical clustering works in the opposite way. Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster.

So, it doesn’t matter if we have 10 or 1000 data points. All these points will belong to the same cluster at the beginning:

### Steps to Perform Hierarchical Clustering

****Step 1:**** First, we assign all the points to an individual cluster:

****Step 2:**** Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. We then update the proximity matrix:

****Step 3:**** We will repeat step 2 until only a single cluster is left.

## Choose the Number of Clusters in Hierarchical Clustering?

To get the number of clusters for hierarchical clustering, we make use of an awesome concept called a ****Dendrogram****

## Choose the Number of Clusters in Hierarchical Clustering?

****. Our aim is to make clusters from this data that can segment similar clients together****. We will, of course, use Hierarchical Clustering for this problem.

**to remove outliers**

Q1,Q3=fin['Bank Balance'].quantile([.25,.75])

interQuartileRange=Q3-Q1

uperlimit=Q3+1.5\*(interQuartileRange)

lowerlimit=Q1-1.5\*(interQuartileRange)

outliers=fin[fin['Bank Balance']>uperlimit]

outliers['Defaulted?'].value\_counts()

fin['Bank Balance']=np.where(fin['Bank Balance']>uperlimit,uperlimit,fin['Bank Balance'])

sns.boxplot(y=fin['Bank Balance'])

# **overfit in decisionTree regresser**

## **Using better evaluation technique - Cross Validation**

from sklearn.model\_selection import cross\_val\_score

scores=cross\_val\_score(model,housing,housing\_label,scoring='neg\_mean\_squared\_error',cv=10)

rmse\_scores=np.sqrt(-scores)

if we have situiation like our dependent value is small in count then others

we use this to make our pridiction model more accurate

!pip install imblearn

if we have situiation like our dependent value is small in count then others

we use this to make our pridiction model more accurate

from imblearn.over\_sampling import SMOTE

sm=SMOTE(random\_state=33,sampling\_strategy=0.75)

X\_res,Y\_res=sm.fit\_resample(x\_train,y\_train)

// random\_state=33,sampling\_strategy=0.75 // is used to divide our data in such a way

if there are 3( 0 or No) then there will be 1 ( 1 or Yes) values

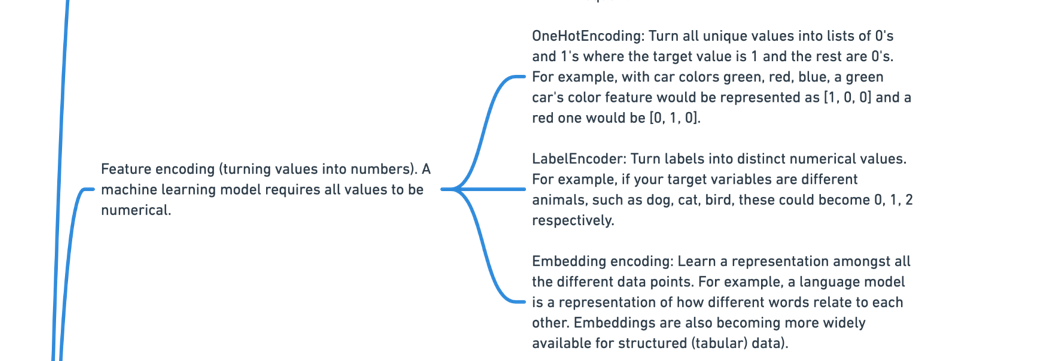
# fin=pd.get\_dummies(fin,drop\_first=True)

to remove (yes and No) into ( 1 or 0 )

# random\_state=21,stratify=y is used because in data is imbalance

means value of dependent value is small

we use it to equally divide dependant in test and train data



Principal component analysis ( PCA )

check model testing

classifier test

from sklearn.metrices import

// confusion matrix

f1\_score

accuracy score

from sklearn.naive\_bayes import MultinomialNB

## Feature Selection

<https://github.com/krishnaik06/Feature-Selection-techniques/blob/master/Feature%20Selection.ipynb>

## 3 Feature selection techniques that are easy to use and also gives good results.

1. Univariate Selection
2. Feature Importance
3. Correlation Matrix with Heatmap

## Univariate Selection

Statistical tests can be used to select those features that have the strongest relationship with the output variable.

The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features.

The example below uses the chi-squared (chi²) statistical test for non-negative features to select 10 of the best features from the Mobile Price Range Prediction Dataset.

## Feature Importance

You can get the feature importance of each feature of your dataset by using the feature importance property of the model.

Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

Feature importance is an inbuilt class that comes with Tree Based Classifiers, we will be using Extra Tree Classifier for extracting the top 10 features for the dataset.

## Correlation Matrix with Heatmap

Correlation states how the features are related to each other or the target variable.

Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)

Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

## **What is Reinforcement Learning?**

****Reinforcement Learning**** is defined as a Machine Learning method that is concerned with how software agents should take actions in an environment. Reinforcement Learning is a part of the deep learning method that helps you to maximize some portion of the cumulative reward.

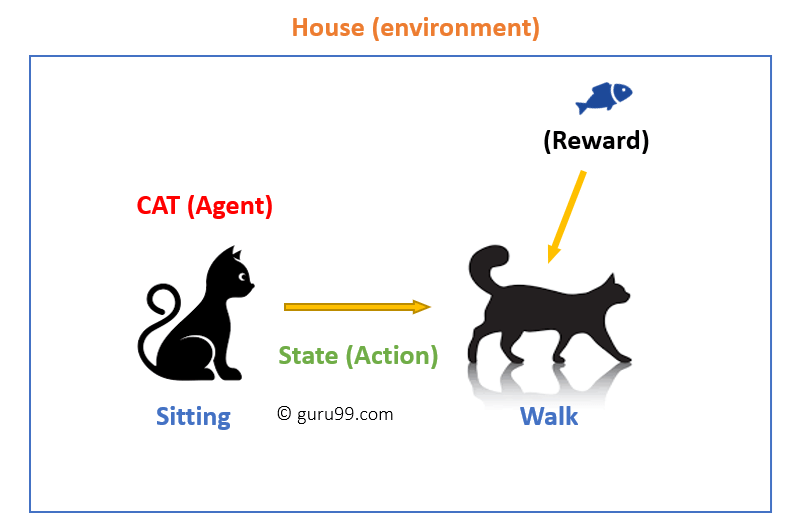
This neural network learning method helps you to learn how to attain a complex objective or maximize a specific dimension over many steps.

## **How Reinforcement Learning works?**

Let’s see some simple example which helps you to illustrate the reinforcement learning mechanism.

Consider the scenario of teaching new tricks to your cat

* As cat doesn’t understand English or any other human language, we can’t tell her directly what to do. Instead, we follow a different strategy.
* We emulate a situation, and the cat tries to respond in many different ways. If the cat’s response is the desired way, we will give her fish.
* Now whenever the cat is exposed to the same situation, the cat executes a similar action with even more enthusiastically in expectation of getting more reward(food).
* That’s like learning that cat gets from “what to do” from positive experiences.
* At the same time, the cat also learns what not do when faced with negative experiences.



## **Reinforcement Learning Algorithms**

There are three approaches to implement a Reinforcement Learning algorithm.

### **Value-Based:**

In a value-based Reinforcement Learning method, you should try to maximize a value function ****V(s)****. In this method, the agent is expecting a long-term return of the current states under policy ****π****.

### **Policy-based:**

In a policy-based RL method, you try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future.

Two types of policy-based methods are:

* Deterministic: For any state, the same action is produced by the policy π.
* Stochastic: Every action has a certain probability, which is determined by the following equation.Stochastic Policy :

### **Model-Based:**

In this Reinforcement Learning method, you need to create a virtual model for each environment. The agent learns to perform in that specific environment.

## **Types of Reinforcement Learning**

Two types of reinforcement learning methods are:

#### **Positive:**

It is defined as an event, that occurs because of specific behavior. It increases the strength and the frequency of the behavior and impacts positively on the action taken by the agent.

#### **Negative:**

Negative Reinforcement is defined as strengthening of behavior that occurs because of a negative condition which should have stopped or avoided. It helps you to define the minimum stand of performance.

## **Learning Models of Reinforcement**

There are two important learning models in reinforcement learning:

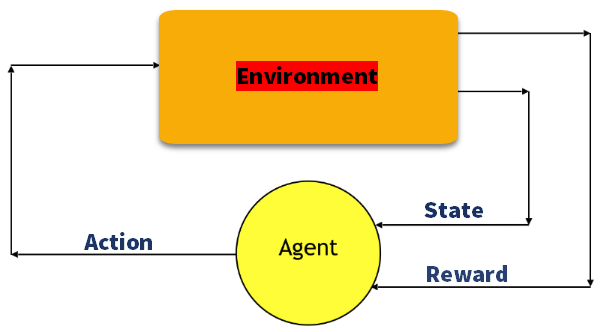
* Markov Decision Process
* Q learning

### **Markov Decision Process**

The following parameters are used to get a solution:

* Set of actions- A
* Set of states -S
* Reward- R
* Policy- n
* Value- V

The mathematical approach for mapping a solution in reinforcement Learning is recon as a Markov Decision Process or (MDP).

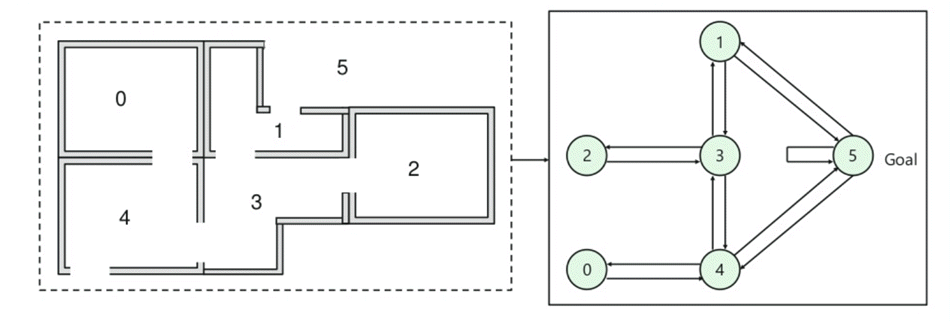


### **Q-Learning**

Q learning is a value-based method of supplying information to inform which action an agent should take.

Let’s understand this method by the following example:

* There are five rooms in a building which are connected by doors.
* Each room is numbered 0 to 4
* The outside of the building can be one big outside area (5)
* Doors number 1 and 4 lead into the building from room 5



Next, you need to associate a reward value to each door:

* Doors which lead directly to the goal have a reward of 100
* Doors which is not directly connected to the target room gives zero reward
* As doors are two-way, and two arrows are assigned for each room
* Every arrow in the above image contains an instant reward value

## **Summary:**

* Reinforcement Learning is a Machine Learning method
* Helps you to discover which action yields the highest reward over the longer period.
* Three methods for reinforcement learning are 1) Value-based 2) Policy-based and Model based learning.
* Agent, State, Reward, Environment, Value function Model of the environment, Model based methods, are some important terms using in RL learning method
* The example of reinforcement learning is your cat is an agent that is exposed to the environment.
* The biggest characteristic of this method is that there is no supervisor, only a real number or reward signal
* Two types of reinforcement learning are 1) Positive 2) Negative
* Two widely used learning model are 1) Markov Decision Process 2) Q learning
* Reinforcement Learning method works on interacting with the environment, whereas the [supervised learning](https://www.guru99.com/supervised-machine-learning.html) method works on given sample data or example.
* Application or reinforcement learning methods are: Robotics for industrial automation and business strategy planning
* You should not use this method when you have enough data to solve the problem
* The biggest challenge of this method is that parameters may affect the speed of learning