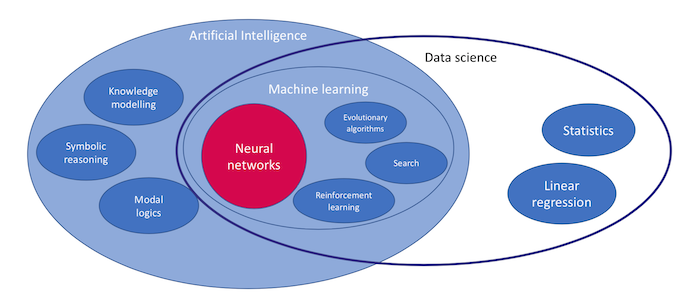
**1.Explain Artificial Intelligence.**

* “AI” is used when a machine simulates functions that humans associate with other human minds, such as learning and problem solving.
* Artifical Intelligence is to impart human intellect to machines and the areas of artificial intelligence are classified as they are reasoning, programming, artificial life, belief revision, data mining, distributed AI, expert systems, genetic algorithms, systems, knowledge representation, machine learning, natural language understanding, neural networks, theorem proving, constraint satisfaction, and theory of computation
* AI has become an important area of research in all fields: Engineering, science, education, medicine, business, accounting, finance, marketing, economics, stock market, and law.
* Subfields of AI, such as machine learning, natural language processing, image processing.

ARTIFICIAL INTELLIGENCE NETWORK 

**MACHINE LEARNING**

* Machine Learning is the science of teaching machines how to learn by themselves. Now, you might be thinking – why on earth would we want machines to learn by themselves? Well – it has a lot of benefits AND **Machines can do high-frequency repetitive tasks with high accuracy without getting bored.**
* For example – the task of mopping and cleaning the floor. When a human does the task – the quality of outcome would vary. We get exhausted/bored after a few hours of work and the chances of getting sick also impact the outcome.

## Overview of the Workflow of ML

## https://miro.medium.com/max/1400/1*QV1rVgh3bfaMbtxueS-cgA.png

# Understanding the machine learning workflow

We can define the machine learning workflow in 3 stages.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

## Applications of Machine Learning in day-to-day life

* Now that you get the hang of it, you might be asking what are some of the examples of machine learning and how does it affect our life? Unless you have been living under a rock – your life is already heavily impacted by machine learning.

**Let us look at a few examples where we use the outcome of machine learning already:**

* Smartphones detecting faces while taking photos or unlocking themselves
* Facebook, LinkedIn or any other social media site recommending your friends and ads you might be interested in
* Amazon recommending you the products based on your browsing history
* Banks using Machine Learning to detect Fraud transactions in real-time.

DEEP LEARNING

* **Deep learning**is a type of machine learning that imitates the way humans gain certain types of knowledge,
* It got more popular over the years compared to standard models while traditional algorithms are linear, deep learning models, generally neural networks, are stacked in a hierarchy of increasing complexity and abstraction (therefore the “deep” in deep learning).
* Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge.
* Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let’s say that we had a set of photos of different pets, and we wanted to categorize by “cat”, “dog”, “hamster”, et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

How deep learning works

* Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.
* Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called *visible*layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.
* Another process called backpropagationuses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.
* The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,
* [*Convolutional neural networks (CNNs),*](https://www.ibm.com/cloud/learn/convolutional-neural-networks)used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
* [*Recurrent neural network (RNNs)*](https://www.ibm.com/cloud/learn/recurrent-neural-networks)are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

## Deep learning applications

* Real-world deep learning applications are a part of our daily lives, but in most cases, they are so well-integrated into products and services that users are unaware of the complex data processing that is taking place in the background. Some of these examples include the following:

### Law enforcement

* Deep learning algorithms can analyze and learn from transactional data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision, and other deep learning applications can improve the efficiency and effectiveness of investigative analysis by extracting patterns and evidence from sound and video recordings, images, and documents, which helps law enforcement analyze large amounts of data more quickly and accurately.

### Financial services

* Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

### Customer service

* Many organizations incorporate deep learning technology into their customer service processes. [Chatbots](https://www.ibm.com/cloud/learn/chatbots-explained)—used in a variety of applications, services, and customer service portals—are a straightforward form of AI. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more [sophisticated chatbot solutions](https://www.ibm.com/products/watson-assistant) attempt to determine, through learning, if there are multiple responses to ambiguous questions. Based on the responses it receives, the chatbot then tries to answer these questions directly or route the conversation to a human user.
* Virtual assistants like Apple's Siri, Amazon Alexa, or Google Assistant extends the idea of a chatbot by enabling speech recognition functionality. This creates a new method to engage users in a personalized way.

### Healthcare

* The healthcare industry has benefited greatly from deep learning capabilities ever since the digitization of hospital records and images. Image recognition applications can support medical imaging specialists and radiologists, helping them analyze and assess more images in less time..

**Q2 .Explain supervised learning.**

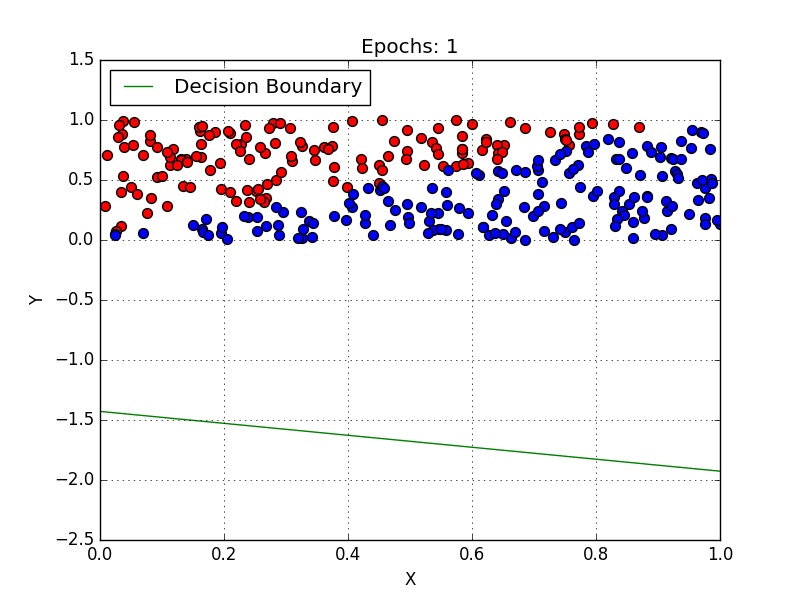
In Supervised learning, an AI system is presented with data which is labelled, which means that each data tagged with the correct label.

The supervised learning is categorized into 2 other categories which are “Classification” and “Regression”.

Classification:

Classification problem is when the target variable is categorical (i.e. the output could be classified into classes — it belongs to either Class A or B or something else).

A classification problem is when the output variable is a category, such as “red” or “blue” , “disease” or “no disease” or “spam” or “not spam”.



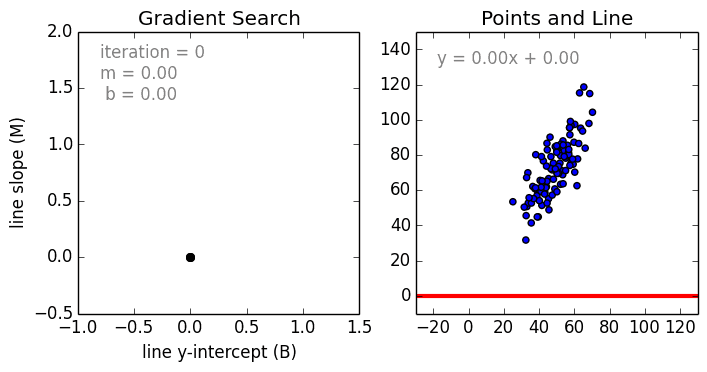
Classification [|](https://www.cs.toronto.edu/~frossard/post/classification/)

As shown in the above representation, we have 2 classes which are plotted on the graph i.e. red and blue which can be represented as ‘setosa flower’ and ‘versicolor flower’, we can image the X-axis as ther ‘Sepal Width’ and the Y-axis as the ‘Sepal Length’, so we try to create the [best fit line](https://mathbits.com/MathBits/TISection/Statistics1/LineFit.htm) that separates both classes of flowers.

These some most used classification algorithms.

* K-Nearest Neighbor
* Naive Bayes
* Decision Trees/Random Forest
* Support Vector Machine
* Logistic Regression
* Regression:

While a Regression problem is when the target variable is continuous (i.e. the output is numeric).



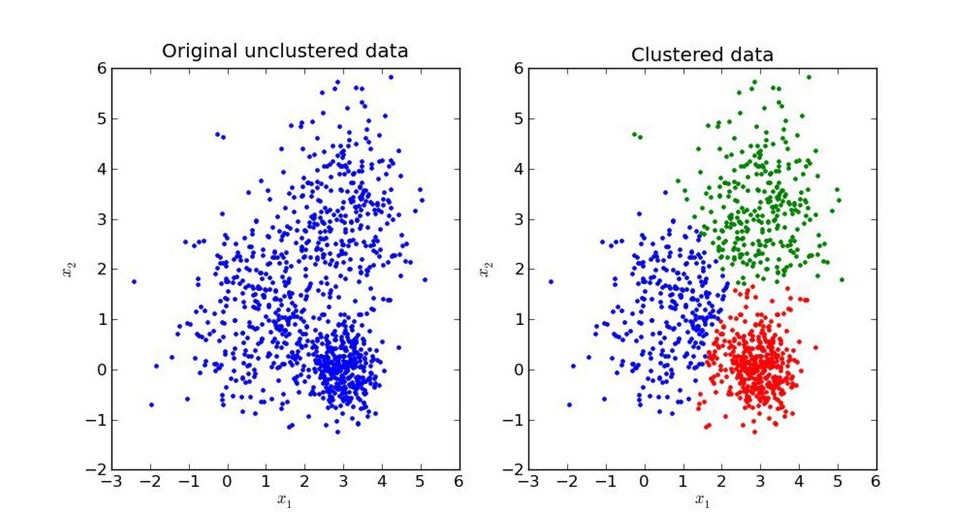
Regression | GIF: techburst.io

As shown in the above representation, we can imagine that the graph’s X-axis is the ‘Test scores’ and the Y-axis represents ‘IQ’. So we try to create the [best fit line](https://mathbits.com/MathBits/TISection/Statistics1/LineFit.htm) in the given graph so that we can use that line to predict any approximate IQ that isn’t present in the given data.

These some most used regression algorithms.

* Linear Regression
* Support Vector Regression
* Decision Tress/Random Forest
* Gaussian Progresses Regression
* Ensemble Methods

## 3. Unsupervised Learning:

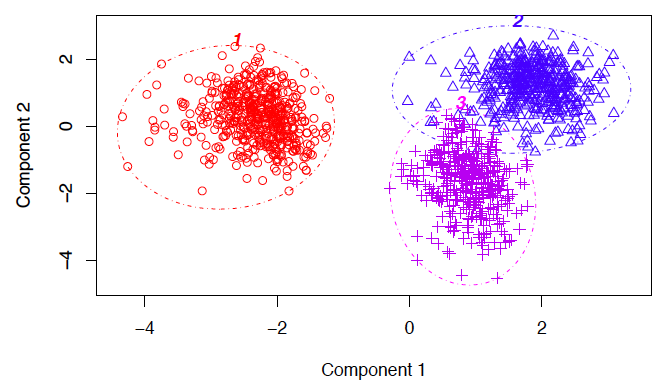


In unsupervised learning, an AI system is presented with unlabeled, un-categorized data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

The unsupervised learning is categorized into 2 other categories which are “Clustering” and “Association”.

Clustering:

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.



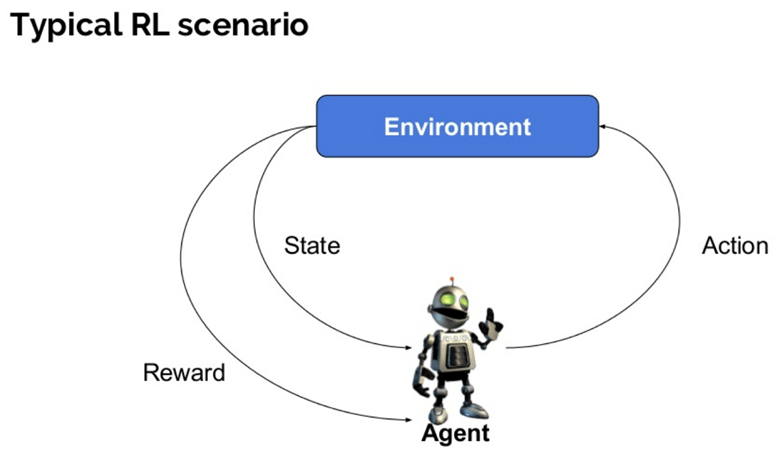
Clustering

Methods used for clustering are:

* Gaussian mixtures
* K-Means Clustering
* Boosting
* Hierarchical Clustering
* K-Means Clustering
* Spectral Clustering

**4.Explain 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 mamize a specific dimension over many steps.



Here are some important terms used in Reinforcement AI:

* **Agent:**It is an assumed entity which performs actions in an environment to gain some reward.
* **Environment (e):**A scenario that an agent has to face.
* **Reward (R):**An immediate return given to an agent when he or she performs specific action or task.
* **State (s):**State refers to the current situation returned by the environment.
* **Policy (π):**It is a strategy which applies by the agent to decide the next action based on the current state.
* **Value (V):**It is expected long-term return with discount, as compared to the short-term reward.
* **Value Function:**Itspecifies the value of a state that is the total amount of reward. It is an agent which should be expected beginning from that state.
* **Model of the environment:**This mimics the behavior of the environment. It helps you to make inferences to be made and also determine how the environment will behave.
* **Model based methods:** It is a method for solving reinforcement learning problems which use model-based methods.
* **Q value or action value (Q):**Q value is quite similar to value. The only difference between the two is that it takes an additional parameter as a current action.

## 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.
* This type of Reinforcement helps you to maximize performance and sustain change for a more extended period. However, too much Reinforcement may lead to over-optimization of state, which can affect the results.

#### Negative:

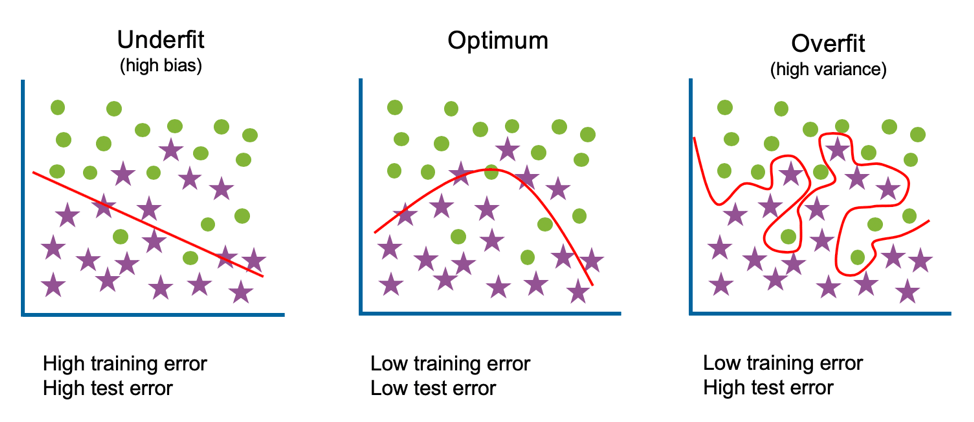
* Negative Reinforcement is defined as strengthening of behavior that occurs because of a negative condition which should have stopped or avoided. It helps you to define the minimum stand of performance. However, the drawback of this method is that it provides enough to meet up the minimum behavior.

5.How to find the best machine learning algorithm for your problem.

* First, We choose, justify, and apply a model performance indicator to assess your model and justify the choice of an algorithm. Examples of model performance indicators include the F1 score, true positive rate, and within cluster sum of squared error.
* Implement your algorithm in at least one deep-learning and at least one non-deep learning algorithm. Then, compare and document model performance.
* Apply at least one more iteration in the process model that involves at least the feature creation task. Record the impact on model performance, such as data normalizing and principal component analysis (PCA). Depending on the algorithm class and data set size, you might choose specific technologies or frameworks to solve your problem.
* As you evaluate algorithms, it's useful to know related terminology:

|  |  |
| --- | --- |
| * **Term** | * **Description** |
| * Descriptive analytics | * Descriptive analytics is likely the most common type of analytics that is used to create dashboards and reports. It describes and summarizes events that already occurred. One example is a grocery store owner who wants to know how items of each product were sold in all stores within a region in the last five years. |
| * Predictive analytics | * Predictive analytics refers to using mathematical and statistical methods to forecast future outcomes. The grocery store owner wants to understand how many products can be sold in the next couple of months so that he can decide on inventory levels. |
| * Prescriptive analytics | * Prescriptive analytics is used to optimize business decisions by simulating scenarios that are based on a set of constraints. The grocery store owner wants to creating a staffing schedule for employees, but to do so, he must account for factors such as availability, vacation time, number of hours of work, and potential emergencies (constraints). He also must create a schedule that works for everyone and ensure that his business is able to function on a daily basis. |
| * Supervised learning | * Supervised learning teaches a model from labeled training data and helps you to make predictions about unseen or future data. During the training, you give the algorithm a dataset that contains correct answers (label y). Then, you validate the model accuracy with a test data set with correct answers. A data set must be split into training and test sets. |
| * Classification | * With classification, you're trying to predict one of a small number of discrete-valued outputs. For example, you might try to predict whether your label is binary (binary classification) or categorical (multiclass classification). |
| * Regression | * In regression, the goal of the learning problem is to predict continuous value output. |
| * Unsupervised learning | * Given a data set, try to find tendencies in the data by using techniques like clustering. |
| * Feature | * Feature is an attribute that is used as input for the model to train. Other names include dimension or column. |
| * Bias | * Bias is the expected difference between the parameters of a model that perfectly fits your data and the parameters that your algorithm learned. Low-bias algorithms, such as Decision Trees, K-nearest Neighbors, and Support Vector Machines, tend to find more complex patterns than high-bias algorithms. |
| * Variance | * Variance is how much the algorithm is impacted by the training data and how much the parameters change with new training data. Low-variance algorithm, such as Linear Regression, Logistic Regression, and Naive Bayes, tend to find less complex patterns than high-variance algorithms. |
| * Underfitting | * The model is too simple to capture the patterns within the data. The model performs poorly on data that it was trained on and on unseen data. High bias, low variance. High training error and high test error. |
| * Overfitting | * The model is too complicated or too specific, capturing trends that don't generalize. The model accurately predicts data that it was trained on but doesn't accurately predict unseen data. Low bias, high variance. Low training error and high test error. |
| * Bias-Variance Trade-off | * Bias-Variance Trade-off refers to finding a model with the right complexity, minimizing both the train and test error. |

### **Train, validate, and test algorithms**

* Machine learning algorithms learn from examples. If you have good data, the more examples you provide, the better the model is at finding patterns in the data. However, be cautious of overfitting. As described in the first table, overfitting is where a model can accurately make predictions for data that it was trained on but can't generalize to other data.
* Overfitting is why you split your data into training data, validation data, and test data. Validation data and test data are often referred to interchangeably, but they have distinct purposes.
* 
* Get familiar with the terminology around training, testing, and validating data:

|  |  |
| --- | --- |
| * **Term** | * **Description** |
| * Training data | * This data is used to train the model and to fit the model parameters. It accounts for the largest proportion of data because you want the model to see as many examples as possible. |
| * Validation data | * This data is used to fit hyperparameters and for feature selection. Although the model never sees this data during training, by selecting particular features or hyperparameters based on this data, you introduce bias and risk overfitting. |
| * Test data | * This data is used to evaluate and compare your tuned models. Because this data wasn't seen during training or tuning, it can provide insight into whether your models generalize well to unseen data. |
| * Cross-validation | * The purpose of cross-validation is to evaluate whether a particular algorithm is suited for your data and use case. Cross-validation is also used for hyperparameter tuning and feature selection. The data is split into train and validation sets, though you need test data put to one side, too. You build a model with each of the slices of data. The final evaluation of the algorithm is the average performance of each of the models. |
| * Hold-out method | * The simplest version of cross-validation is the hold-out method, where you randomly split your data into two sets: a training set and a validation set. This method is the quickest because it requires building a model only once. However, with only one validation data set, you risk that it contains easy or difficult observations to predict. Therefore, you might find that you're overfitting to this validation data, and it will perform poorly on a test set. |
| * K-fold cross-validation | * This method involves splitting the data into k-subsets. You then train a model k times, each time by using one of the k-subsets as its validation data. The training data is all other observations that aren't in the validation set. Your final evaluation is the average across all k folds. |
| * Leave-one-out cross-validation | * This method is the most extreme version of k-fold cross-validation. In it, your k is N (the number of observations in your data set). You train a model N separate times by using all data except for one observation and then validate its accuracy with its prediction for that observation. Although you're thoroughly evaluating how well this algorithm works with your data set, this method is expensive because it requires you to build N models. |
| * Stratified cross-validation | * This method enforces that the k-fold sets have similar proportions of observations for each class in either categorical features or the label. |

### **Algorithm exploration**

* The algorithms that you explore must be driven by your use case. By first identifying what you're trying to achieve, you can narrow the scope of searching for solutions. Possible methods include, but aren't limited to algorithms for regression, classification, clustering, recommendations, and anomaly detection. You can also use the Algorithm Explorer to guide algorithm selection.

#### **Regression**

* Regression algorithms are machine learning techniques for predicting continuous numerical values. They're supervised learning tasks, so they require labeled training examples.

#### **Classification**

* Classification algorithms are machine learning techniques for predicting which category the input data belongs to. They're supervised learning tasks, so they require labeled training examples.

#### **Clustering**

* Clustering algorithms are machine learning techniques to divide data into a number of groups where points in the groups have similar traits. They're unsupervised learning tasks and don't require labeled training examples.

#### **Recommendation engines**

* Recommendation engines are created to predict a preference or rating that indicates a user’s interest in an item or product. The algorithms that are used to create this system find similarities between the users, the items, or both.

#### **Anomaly detection**

* Anomaly detection is a technique that is used to identify unusual events or patterns that don't conform to expected behavior. The items that are identified are often referred to as anomalies or outliers.

### **Hyperparameter optimization**

* Although the terms parameters and hyperparameters are occasionally used interchangeably, distinctions exist between them. Parameters are properties that the algorithm is learning during training. For linear regression, those parameters are the weights and biases. For random forests, they're the variables and thresholds at each node.
* Hyperparameters are properties that must be set before training. For k-means clustering, you must define the value of k. For neural networks, an example is the learning rate. Hyperparameter optimization is the process of finding the best possible values for the hyperparameters to optimize your performance metric, such as highest accuracy and lowest RMSE (root-mean-square error). You train a model for different combinations of values and evaluate which models find the best solution. Methods to search for the best combinations include grid search, random search, coarse-to-fine, and bayesian optimization.

#### **Grid search**

* In grid search, you specify values for each hyperparameter, and all combinations of those values are evaluated. For example, to evaluate hyperparameters for random forest, you might provide three options for the number of trees hyperparameter: 10, 20 and 50. For the maximum depth of each tree, you also provide three options: no limit, 10 and 20. This selection results in a random forest model being built for each of the nine possible combinations: (10, no limit), (10, 10), (10, 20), (20, no limit), (20, 10), (20, 20), (50, no limit), (50, 10), and (50, 20).
* The combination that provides the best performance is the one that you use for your final model. This method is simple to use. You can find the best combination of the values that you provided, and you can run each of the experiments in parallel. However, it's also computationally expensive because so many models are being built. If a hyperparameter isn't important, you might explore different possibilities unnecessarily.

#### **Random search**

* In random search, you specify ranges or options for each hyperparameter and random values of each are selected. Continuing with the random forest example, you might provide the range for the number of trees to be 10 - 50 and max\_depth to be either no limit, 10, or 20. This time, rather than compute all the permutations, you can specify the number of iterations you want to run. You want only five. You might test something like (19, 20), (32 no limit), (40, no limit), (10, 20), (27, 10).
* This method is simple to use, efficient, and can outperform grid search when only a few hyperparameters affect the overall performance. You can run each of the experiments in parallel. However, it involves random sampling, so it finds the best combination only if it searches that space.

#### **Coarse-to-fine**

* For both grid search and random search, you can also use the coarse-to-fine technique. This technique involves exploring a broader range of variables with wide intervals or all possible options. After you have your results from your initial search, you explore the results to find any patterns or regions that look promising. If so, you can repeat the process but refine your search.
* For the random forest example, you might notice that results are promising when the maximum depth has no limit and when the number of trees hyperparameter is either 10 or 20. The search process is repeated but keeps the maximum depth hyperparameter constant and increases the granularity of the number of tree options, testing values of 12, 14, 16, 18, to see whether you can find a better result.
* This method can find more optimized hyperparameters, improving the performance metric. However, the evaluation of the results to find the best regions to explore can be cumbersome.

#### **Bayesian optimization**

* Bayesian optimization uses the prior knowledge of success with hyperparameter combinations to choose the next best. This technique uses a machine learning approach, building a model where the hyperparameters are the features and the performance is the target variable. After each experiment, a new data point is added and a new model is built. It assumes that similar combinations have similar results, and it prioritizes exploring regions where promising results were found.
* However, it also takes into consideration uncertainty as a possibility for large gain. If large unexplored areas exist, it also prioritizes those areas. Taking only one hyper-parameter, the number of trees, the algorithm might first try 10 and get good performance. It then tries 32 and the performance is significantly better.
* Bayesian optimization builds a model based on the first two data points to predict performance. The model is likely to be linear with only the two data points, so the next value that it chooses is 40, with the expectation that the performance improves as the number of trees increases. It doesn't.
* It builds another model that suggests that there might be improvement around 32 where it saw the best result so far. However, a large gap still exists between 10 and 32 that hasn’t been explored, and because of large uncertainty, it chooses 21. Again, the model is tweaked with this new data and another value is chosen.
* This method can find more optimized hyperparameters, improving the performance metric. It can reduce the time spent searching for an optimum solution when the number of parameters is high and each experiment is computationally expensive. However, you can't run each experiment in parallel because the next combination of hyperparameter values is determined by the prior runs. It also requires tuning: choosing a scale for each hyperparameter and an appropriate kernel.

### **Ensemble learning**

* Ensembles combine several machine learning models, each finding different patterns within the data to provide a more accurate solution. These techniques can improve performance, as they capture more trends. They can also reduce overfitting, as the final prediction is a consensus from many models.

#### **Bagging**

* Bagging, or bootstrap aggregations, is the method of building models in parallel and averaging their prediction as the final prediction. These models can be built with the same algorithm. For example, the random forest algorithm builds many decision trees. You can also build different types of models, such as a linear regression model and a support vector machine (SVM) model.

#### **Boosting**

* Boosting builds models sequentially by evaluating the success of earlier models. The next model prioritizes learning trends for predicting examples that the current models perform poorly on. Three common techniques are AdaBoost, Gradient Boosting, and XGBoosted.

#### **Stacking**

* Stacking involves building models and using their output as features into a final model. For example, your goal is to create a classifier and you build a KNN model and a Naïve Bayes model. Rather than choose between the two, you can pass their two predictions into a final logistic regression model. This final model might result in better results than the two intermediate models.