Introduction to Machine Learning

It will demystify the fascinating field of Machine Learning and learn about the basic concepts and terminologies that are essential for understanding how machines can learn from data.

Machine Learning: What is it?

Machine Learning is like teaching a computer to learn and make predictions, just like we learn from our experiences. Imagine you have a friend who always guesses the price of things correctly. Over time, your friend learns from their mistakes and gets better at guessing. That's what we do with machines - we teach them to improve their predictions by learning from data.

Demystifying ML Terminologies:

Let's use a simple example to demystify some key ML terms:

Example: Imagine you have a table with X and Y values.

X	1	2	3	4	5	….	16
Y	3	5	7	9	11	….	**?**

Now, your task is to predict the Y value for X = 16.

Answer: Y = 33

Justification: In this example, Y = 2X + 1, applying X = 16, Y = 2*16 + 1 = 33

What process happened on your mind now, you try to identify some pattern in relationship between X and Y. Then you figured out the relationship between X and Y, that is Y = 2X + 1. Then you are applying the unforeseen value of X for predicting the Y value for X = 16. Machine Learning is mimicking the process happened on your brain now, to the computer programs.

Now, let's demystify the common terminologies in Machine Learning Lingo.

- **Features**: In our example, X is the feature, which helps for doing the prediction. Just like the features in real estate prediction could be the number of bedrooms, location, or square footage.
- **Label**: Y is the label we want to predict. In real estate, it's the property price.
- **ML Algorithms**: Your friend's guessing strategy is an algorithm. Similarly, in machine learning, we have algorithms to make predictions. Basically, all the machine learning algorithms are mathematical tools to identify the hidden pattern from features and labels.
- **Training**: The process of identifying the pattern by passing the data to the ML algorithm is known as training.
- **Model**: Model is the pattern identified from the data. It is the output of training process.
- **Parameters:** Parameters are the optimized variables in the model. The model's mathematical structure depends on the machine learning algorithm you are using. For example, Y = 2X+1, is mathematically known as Linear Regression. Where in linear regression Y = mX + c, where m and c are parameters. You will start m and c as random numbers or zeroes initially. Then by optimization algorithm like Gradient Descent Algorithm will update and optimize those values during the training (m = 2, and c=1 is the optimized values for the given data).
- **Hyperparameters**: Some parameters in the model will not optimize (change) during the training process. These parameters are called the Hyperparameters. For example, learning rate is parameter which decides how quickly the parameters m and c are varying. This Learning rate is a hyperparameter since it will be constant throughout the training process.
- **Inferencing**: When you make a guess (infer) without looking at the answer, you are doing inferencing. In machine learning, inferencing is making predictions using the trained model.

What is machine learning and how does it work?

Machine learning (ML) is the process of using mathematical models of data to help a computer learn without direct instruction. It's considered a subset of artificial intelligence (AI). Machine learning uses algorithms to identify patterns within data, and those patterns are then used to create a data model that can make predictions. With increased data and experience, the results of machine learning are more accurate—much like how humans improve with more practice.

The adaptability of machine learning makes it a great choice in scenarios where the data is always changing, the nature of the request or task are always shifting, or coding a solution would be effectively impossible.

How machine learning relates to AI

Machine learning is considered a subset of AI. An “intelligent” computer thinks like a human and carries out tasks on its own. One way to train a computer to mimic human reasoning is to use a neural network, which is a series of algorithms that are modeled after the human brain.

The benefits of machine learning

Machine learning has many applications—and the possibilities are constantly expanding. Here are some of the top benefits that businesses have achieved with their machine learning projects:

- **Uncover insight**

Machine learning can help identify a pattern or structure within both structured and unstructured data, helping to identify the story the data is telling.

- **Improve data integrity**

Machine learning is excellent at data mining and can take it a step further, improving its abilities over time.

- **Enhance user experience**

Adaptive interfaces, targeted content, chatbots, and voice-enabled virtual assistants are all examples of how machine learning can help optimize the customer experience.

- **Reduce risk**

As fraud tactics constantly change, machine learning keeps pace—monitoring and identifying new patterns to catch attempts before they’re successful.

- **Anticipate customer behavior**

Machine learning can mine customer-related data to help identify patterns and behaviors, letting you optimize product recommendations and provide the best customer experience possible.

- **Lower costs**

One machine learning application is process automation, which can free up time and resources, allowing your team to focus on what matters most.

What are machine learning algorithms?

Machine learning algorithms are pieces of code that help people explore, analyze, and find meaning in complex data sets. Each algorithm is a finite set of unambiguous step-by-step instructions that a machine can follow to achieve a certain goal. In a machine learning model, the goal is to establish or discover patterns that people can use to make predictions or categorize information.

Machine learning algorithms use parameters that are based on training data—a subset of data that represents the larger set. As the training data expands to represent the world more realistically, the algorithm calculates more accurate results.

Different algorithms analyze data in different ways. They’re often grouped by the machine learning techniques that they’re used for: supervised learning, unsupervised learning, and reinforcement learning. The most commonly used algorithms use regression and classification to predict target categories, find unusual data points, predict values, and discover similarities.

Machine learning techniques

As you learn more about machine learning algorithms, you’ll find that they typically fall within one of three machine learning techniques:

Supervised learning

In supervised learning, algorithms make predictions based on a set of labeled examples that you provide. This technique is useful when you know what the outcome should look like.

For example, you provide a dataset that includes city populations by year for the past 100 years, and you want to know what the population of a specific city will be four years from now. The outcome uses labels that already exist in the data set:population,city,andyear.

Unsupervised learning

In unsupervised learning, the data points aren't labeled—the algorithm labels them for you by organizing the data or describing its structure. This technique is useful when you don't know what the outcome should look like.

For example, you provide customer data, and you want to create segments of customers who like similar products. The data that you're providing isn't labeled, and the labels in the outcome are generated based on the similarities that were discovered between data points.

Reinforcement learning

Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral, or incorrect. It's a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

For example, you're designing an autonomous car, and you want to ensure that it's obeying the law and keeping people safe. As the car gains experience and a history of reinforcement, it learns how to stay in its lane, go the speed limit, and brake for pedestrians.

What you can do with machine learning algorithms

Machine learning algorithms help you answer questions that are too complex to answer through manual analysis. There are many different machine learning algorithm types, but use cases for machine learning algorithms typically fall into one of these categories.

Predict a target category

Two-class (binary) classification algorithms divide the data into two categories. They're useful for questions that have only two possible answers that are mutually exclusive, including yes/no questions. For example:

- Will this tire fail in the next 1,000 miles: yes or no?
- Which brings in more referrals: a $10 credit or a 15% discount?

Multiclass (multinomial) classification algorithms divide the data into three or more categories. They're useful for questions that have three or more possible answers that are mutually exclusive. For example:

- In which month do the majority of travelers purchase airline tickets?
- What emotion is the person in this photo displaying?

Find unusual data points

Anomaly detection algorithms identify data points that fall outside of the defined parameters for what's "normal." For example, you would use anomaly detection algorithms to answer questions like:

- Where are the defective parts in this batch?
- Which credit card purchases might be fraudulent?

Predict values

Regression algorithms predict the value of a new data point based on historical data. They help you answer questions like:

- How much will the average two-bedroom home cost in my city next year?
- How many patients will come through the clinic on Tuesday?

See how values change over time

Time series algorithms show how a given value changes over time. With time series analysis and time series forecasting, data is collected at regular intervals over time and used to make predictions and identify trends, seasonality, cyclicity, and irregularity. Time series algorithms are used to answer questions like:

- Is the price of a given stock likely to rise or fall in the coming year?
- What will my expenses be next year?

Discover similarities

Clustering algorithms divide the data into multiple groups by determining the level of similarity between data points. Clustering algorithms work well for questions like:

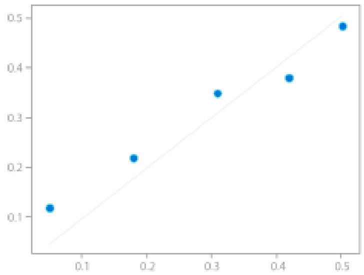
- Which viewers like the same types of movies?
- Which printer models fail in the same way?

Classification

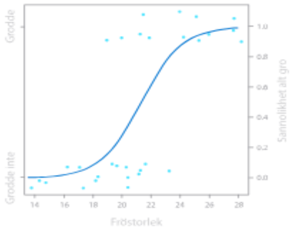
Classification algorithms use predictive calculations to assign data to preset categories. Classification algorithms are trained on input data, and used to answer questions like:

- Is this email spam?
- What is the sentiment (positive, negative, or neutral) of a given text?

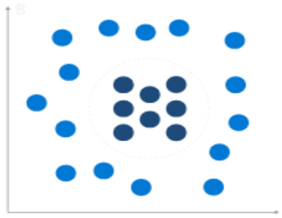
Linear regression algorithms show or predict the relationship between two variable or factors by fitting a continuous straight line to the data. The line is often calculated using the Squared Error Cost function. Linear regression is one of the most popular types of regression analysis.



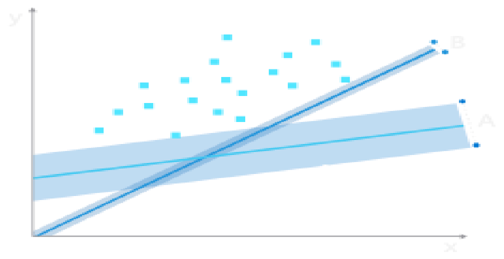
Logistic regression algorithms fit a continuous S-shaped curve to the data. Logistic regression is another popular type of regression analysis.



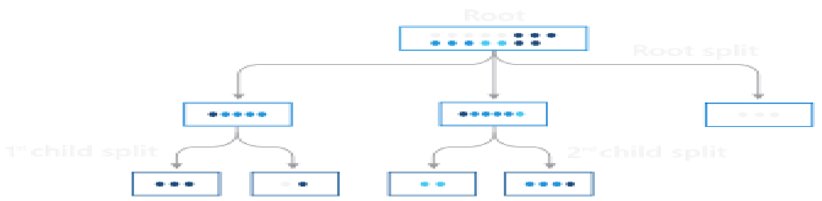
Naïve Bayes algorithms calculate the probability that an event will occur, based on the occurrence of a related event.



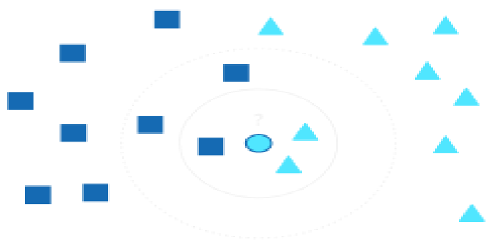
Support Vector Machines draw a hyperplane between the two closest data points. This marginalizes the classes and maximizes the distances between them to more clearly differentiate them



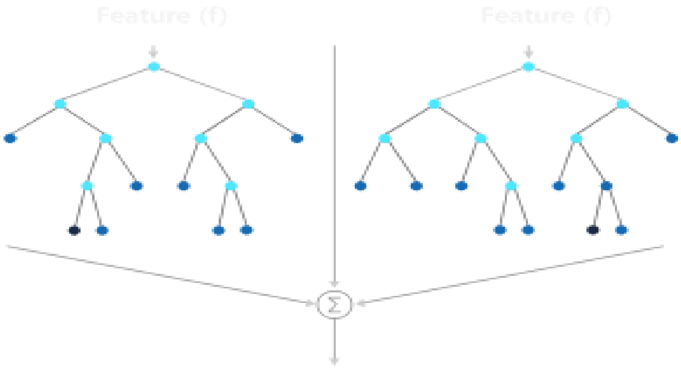
Decision tree algorithms split the data into two or more homogeneous sets. They use if–then rules to separate the data based on the most significant differentiator between data points.



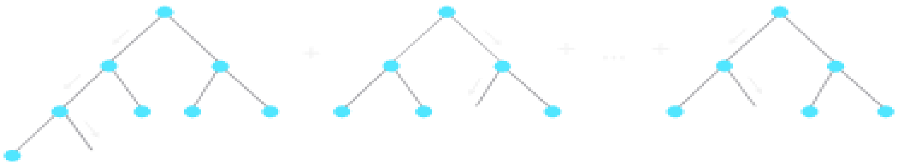
K-Nearest neighbor algorithms store all available data points and classify each new data point based on the data points that are closest to it, as measured by a distance function.



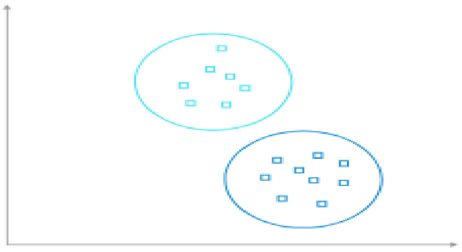
Random forest algorithms are based on decision trees, but instead of creating one tree, they create a forest of trees and then randomize the trees in that forest. Then, they aggregate votes from different random formations of the decision trees to determine the final class of the test object.



Gradient boosting algorithms produce a prediction model that bundles weak prediction models—typically decision trees—through an ensembling process that improves the overall performance of the model.



K-Means algorithms classify data into clusters—where K equals the number of clusters. The data points inside of each cluster are homogeneous, and they're heterogeneous to data points in other clusters.



What are machine learning libraries?

A machine learning library is a set of functions, frameworks, modules, and routines written in a given language. Developers use the code in machine learning libraries as building blocks for creating machine learning solutions that can perform complex tasks. Instead of having to manually code every algorithm and formula in a machine learning solution, developers can find the functions and modules they need in one of many available ML libraries, and use those to build a solution that meets their needs.

Types of ML Algorithms:

Let's explore the different types of Machine Learning algorithms using classroom teaching as an analogy, making it relatable and easy to understand.

Supervised Learning: Teaching and Testing

Imagine a teacher (machine) who's helping students (models) learn. In supervised learning, the teacher guides the students by teaching them a subject and then conducts a test.

1. **Teaching Phase:** The teacher prepares a question paper (features) and an answer key (label). The teacher teaches the students based on these materials.

2. **Testing Phase:** Students write their answers (predictions) based on what they've learned from the teacher. The teacher then compares these answer scripts to the answer key.

3. **Evaluation:** Any differences between the answer script and the answer key are like errors. The teacher analyzes these errors and modifies their teaching approach, adjusting some parameters (teaching methods), to improve student learning performance.

The same learning process in machine learning where labels are utilized for training is known as supervised learning.

Unsupervised Learning: Grouping without Labels

Now, let's consider unsupervised learning. Imagine the teacher wants to group students based on their similarities, but without knowing their names or any labels.

1. **Grouping Phase:** The teacher observes student behaviors and interactions, trying to find patterns and similarities.

2. **Clustering:** Based on these observations, the teacher groups students who behave similarly, creating clusters of students with common characteristics.

Similarly, for the learning process if labels are not utilized that is unsupervised learning.

Reinforcement Learning: Learning Through Rewards and Punishments

In reinforcement learning, we'll liken the process to teaching a dog new trick. The dog (agent) learns through a series of actions and outcomes, which are associated with rewards or punishments.

1. **Action and Consequence:** The dog takes actions (like performing tricks) without knowing which action is right or wrong. After each action, it receives rewards (treats) or punishments (a gentle scolding).
2. **Learning and Improvement:** Over time, the dog learns which actions result in more rewards and fewer punishments. It adjusts its behavior accordingly to maximize rewards.

In this way, supervised learning is like a teacher guiding students through structured lessons and tests, unsupervised learning is like a teacher grouping students based on observed behavior, and reinforcement learning is like teaching a dog new trick through rewards and punishments.

ML Tasks: Classification, Regression, Clustering

Machine learning encompasses various tasks, and each task is tailored to different types of predictions. Let's delve into some of the most common ML tasks and provide real-life examples to make them more relatable.

Regression:

In regression, the goal is to predict a numeric value based on features. It's like trying to estimate a continuous quantity, such as:

Predicting the number of ice creams sold on a given day based on factors like temperature, rainfall, and windspeed. As the temperature rises, people tend to buy more ice creams.

Estimating the selling price of a property based on features like its square footage, the number of bedrooms, and the socio-economic metrics of its location. Larger properties with more bedrooms tend to have higher prices.

Determining the fuel efficiency (miles-per-gallon) of a car based on features like engine size, weight, width, height, and length. Smaller and lighter cars often have better fuel efficiency.

Classification:

Classification is about categorizing data into distinct classes or categories. There are two common scenarios:

Binary Classification:

In binary classification, the model predicts whether an item belongs to a specific class or not. For example:

Identifying whether a patient is at risk for diabetes based on clinical metrics like weight, age, and blood glucose level. The model predicts if the patient is at risk (positive) or not (negative).

Predicting whether a bank customer will default on a loan based on factors like income, credit history, and age. The model determines if the customer is likely to default or not.

Assessing whether a mailing list customer will respond positively to a marketing offer based on attributes like demographic information and past purchases. The model predicts whether the customer will respond positively or not.

Multiclass Classification:

In multiclass classification, the task extends beyond two classes, allowing prediction into multiple possible categories. For example:

Classifying the species of a penguin as Adelie, Gentoo, or Chinstrap based on physical measurements. Each penguin falls into one of these distinct categories.

Determining the genre of a movie as comedy, horror, romance, adventure, or science fiction based on features like the cast, director, and budget. Movies are categorized into one of these genres.

Occasionally, some algorithms allow for multilabel classification, where an observation can have more than one valid label. For instance, a movie could be categorized as both science fiction and comedy.

Clustering:

Clustering is a form of unsupervised learning that groups data into clusters based on similarities between observations. This is akin to sorting items without predefined categories, such as:

Grouping similar types of flowers based on features like size, number of leaves, and petals. Clustering helps identify patterns in flower species.

Identifying groups of customers with similar demographic attributes and purchasing behavior. Clustering allows businesses to understand their customer segments.

Clustering is different from classification in that there are no predefined classes; the algorithm groups data based solely on the similarity of features. In some cases, clustering is used to identify potential classes before training a classification model.

Understanding these ML tasks helps us address a wide range of problems, from predicting quantities to categorizing and sorting data in a meaningful way, making machine learning a powerful tool in various fields.

Introduction to Scikit-Learn

Scikit-Learn, also known as sklearn, is like a toolbox for machine learning. Just as a carpenter uses tools to build furniture, data scientists use Scikit-Learn to build machine learning models. It provides various tools, algorithms, and functions to make our machine learning tasks easier.

Data Preprocessing

To prepare data for machine learning, we often need to clean and transform it. Think of this as getting ingredients ready before cooking.

Data Normalization:

It's like making sure all ingredients in a recipe are in the same unit of measurement. In machine learning, we want to scale our features, so they all have a similar range, helping algorithms work more effectively.

```
from sklearn.preprocessing import MinMaxScaler

# Sample data for normalization

data = [[1, 2], [2, 4], [3, 6]]

# Create a MinMaxScaler

scaler = MinMaxScaler()

# Fit and transform the data

scaled_data = scaler.fit_transform(data)

print(scaled_data)
```

Output

```
[[0. 0.]
 [0.5 0.5]
 [1. 1.]]
```

Standardization:

Similar to converting temperatures from Fahrenheit to Celsius. It makes sure our features have a mean of 0 and a standard deviation of 1, making it easier for machine learning models to compare them.

```
from sklearn.preprocessing import StandardScaler

# Sample data for standardization
data = [[1, 2], [2, 4], [3, 6]]

# Create a StandardScaler
scaler = StandardScaler()

# Fit and transform the data
standardized_data = scaler.fit_transform(data)
print(standardized_data)
```

Output:

```
[[-1. -1.]
[ 0. 0.]
[ 1. 1.]]
```

Label Encoding:

Think of this as assigning numbers to categories. For example, converting colors like "red," "green," and "blue" into 1, 2, and 3

```
from sklearn.preprocessing import LabelEncoder

# Sample data for label encoding
data = ['red', 'green', 'blue', 'red', 'blue']

# Create a LabelEncoder
encoder = LabelEncoder()

# Fit and transform the data
encoded_data = encoder.fit_transform(data)
print(encoded_data)
```

Output:

```
[2 1 0 2 0]
```

One-Hot Encoding:

Imagine you have a menu with options like "Burger," "Pizza," and "Sushi." One-hot encoding is like turning these options into checkboxes (0 or 1) for each item.

```
from sklearn.preprocessing import OneHotEncoder
# Sample data for one-hot encoding
data = ['Burger', 'Pizza', 'Sushi', 'Burger', 'Sushi']
# Create a OneHotEncoder
encoder = OneHotEncoder(sparse=False)
# Fit and transform the data
one_hot_data = encoder.fit_transform(np.array(data).reshape(-1, 1))
print(one_hot_data)
```

Output:

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]
 [1. 0. 0.]
 [0. 0. 1.]]
```

Feature Engineering

Feature engineering is like creating new ingredients or flavors to enhance a recipe. In machine learning, we create new features from existing ones to improve model performance.

Let's say you have a dataset of houses with features like "number of bedrooms" and "number of bathrooms." You can engineer a new feature "bedrooms_per_bathroom" by dividing the number of bedrooms by the number of bathrooms. This could be a valuable predictor of house prices.

```
import pandas as pd


# Sample dataset
data = {'bedrooms': [2, 3, 4, 2, 3],
      'bathrooms': [1, 2, 2, 1, 1]}


# Create a DataFrame
df = pd.DataFrame(data)


# Feature engineering
df['bedrooms_per_bathroom'] = df['bedrooms'] / df['bathrooms']


print(df)
```

Output

```
   bedrooms  bathrooms      bedrooms_per_bathroom
0        2         1                  2.0
1        3         2                  1.5
2        4         2                  2.0
3        2         1                  2.0
4        3         1                  3.0
```

Train-Test Split

Splitting data into training and testing sets is like having two kitchens. You cook and experiment in one (training) without worrying about ruining the meal you serve in the other (testing).

```
from sklearn.model_selection import train_test_split

# Sample data and labels

data = [1, 2, 3, 4, 5, 6]

labels = [0, 0, 1, 1, 1, 0]

# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

print("X_train:", X_train)

print("X_test:", X_test)

print("y_train:", y_train)

print("y_test:", y_test)
```

Output

```
X_train: [4, 6, 3, 1]

X_test: [2, 5]
```

y_train: [1, 0, 1, 0]

y_test: [0, 1]

These data pre-processing techniques and train-test splitting are essential to ensure that your machine learning models learn effectively and can be tested for their performance. Just as in cooking, proper preparation ensures a better final dish.