**Getting started with Machine Learning**

ML is a process of training a piece of software (Model/Algorithm) to make predictions.

**Types of machine learning problems**

**1. On basis of the nature of the learning “signal” or “feedback” available to a learning system**

* **Supervised learning**: The model or algorithm is presented with example inputs and their desired outputs and then finds patterns and connections between the input and the output. The goal is to learn a general rule that maps inputs to outputs. The training process continues until the model achieves the desired level of accuracy on the training data. Some real-life examples are:
  + **Image Classification:**You train with images/labels. Then in the future, you give a new image expecting that the computer will recognize the new object.
  + **Market Prediction/Regression:**You train the computer with historical market data and ask the computer to predict the new price in the future.
* **Unsupervised learning**: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. It is used for clustering populations in different groups. Unsupervised learning can be a goal in itself (discovering hidden patterns in data).
  + **Clustering:** You ask the computer to separate similar data into clusters, this is essential in research and science.
  + **High-Dimension Visualization:** Use the computer to help us visualize high-dimension data.
  + **Generative Models:** After a model captures the probability distribution of your input data, it will be able to generate more data. This can be very useful to make your classifier more robust.

A simple diagram that clears the concept of supervised and unsupervised learning is shown below:  
A diagram of a graph

Description automatically generatedA diagram of a graph

Description automatically generated   
As you can see clearly, the data in supervised learning is labeled, whereas data in unsupervised learning is unlabelled.

* **Semi-supervised learning**: Problems where you have a large amount of input data and only some of the data is labeled, are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning. For example, a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.
* **Reinforcement learning**: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments as it navigates its problem space.

A diagram of a agent and action

Description automatically generated

**2. Two most common use cases of Supervised learning are:**

* **Classification**: Inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes and predicts whether or not something belongs to a particular class. This is typically tackled in a supervised way.  Classification models can be categorized in two groups: Binary classification and Multiclass Classification. Spam filtering is an example of binary classification, where the inputs are email (or other) messages and the classes are “spam” and “not spam”.
* **Regression**: It is also a supervised learning problem, that predicts a numeric value and outputs are continuous rather than discrete. For example, predicting stock prices using historical data.

An example of classification and regression on two different datasets is shown below:A diagram of a graph

Description automatically generated with medium confidence

**3. Most common Unsupervised learning are:**

* **Clustering**: Here, 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. As you can see in the example below, the given dataset points have been divided into groups identifiable by the colors red, green, and blue.
* **Density estimation**: The task is to find the distribution of inputs in some space.
* **Dimensionality reduction**: It simplifies inputs by mapping them into a lower-dimensional space. Topic modeling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover similar topics.

On the basis of these machine learning tasks/problems, we have a number of algorithms that are used to accomplish these tasks. Some commonly used machine learning algorithms are Linear Regression, Logistic Regression, Decision Tree, SVM(Support vector machines), Naive Bayes, KNN(K nearest neighbors), K-Means, Random Forest, etc. **Note:**All these algorithms will be covered in upcoming articles.

**Terminologies of Machine Learning**

* **Model** A model is a **specific representation** learned from data by applying some machine learning algorithm. A model is also called a **hypothesis**.
* **Feature** A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a **feature vector**. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, **etc.** **Note:** Choosing informative, discriminating and independent features is a crucial step for effective algorithms. We generally employ a **feature extractor** to extract the relevant features from the raw data.
* **Target (Label)** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the features section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training** The idea is to give a set of inputs(features) and its expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label). But make sure if the machine performs well on unseen data, then only we can say the machine performs well.

The figure shown below clears the above concepts:A diagram of a process

Description automatically generated

Here are the steps to get started with machine learning:

1. *Define the Problem*: Identify the problem you want to solve and determine if machine learning can be used to solve it.
2. *Collect Data*: Gather and clean the data that you will use to train your model. The quality of your model will depend on the quality of your data.
3. *Explore the Data:* Use data visualization and statistical methods to understand the structure and relationships within your data.
4. *Pre-process the Data*: Prepare the data for modeling by normalizing, transforming, and cleaning it as necessary.
5. *Split the Data*: Divide the data into training and test datasets to validate your model.
6. *Choose a Model*: Select a machine learning model that is appropriate for your problem and the data you have collected.
7. *Train the Model:*Use the training data to train the model, adjusting its parameters to fit the data as accurately as possible.
8. *Evaluate the Model:* Use the test data to evaluate the performance of the model and determine its accuracy.
9. *Fine-tune the Model:* Based on the results of the evaluation, fine-tune the model by adjusting its parameters and repeating the training process until the desired level of accuracy is achieved.
10. *Deploy the Model:* Integrate the model into your application or system, making it available for use by others.
11. *Monitor the Model:* Continuously monitor the performance of the model to ensure that it continues to provide accurate results over time.

Example :

Here is a simple machine-learning example in Python that demonstrates how to train a model to predict the species of iris flowers based on their sepal and petal measurements:

**Python**

|  |
| --- |
| # Load the necessary libraries  **import** pandas as pd  **from** sklearn.model\_selection **import** train\_test\_split  **from** sklearn.svm **import** SVC    # Load the iris dataset  df **=** pd.read\_csv('iris.csv')    # Split the data into features and labels  X **=** df[['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width']]  y **=** df['species']    # Split the data into training and testing sets  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_state**=**42)    # Create an SVM model and train it  model **=** SVC()  model.fit(X\_train, y\_train)    # Evaluate the model on the test data  accuracy **=** model.score(X\_test, y\_test)    print('Test accuracy:', accuracy) |