

Introduction to Machine Learning

Define what is Machine learning and why it is useful

What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

Traditional Computing



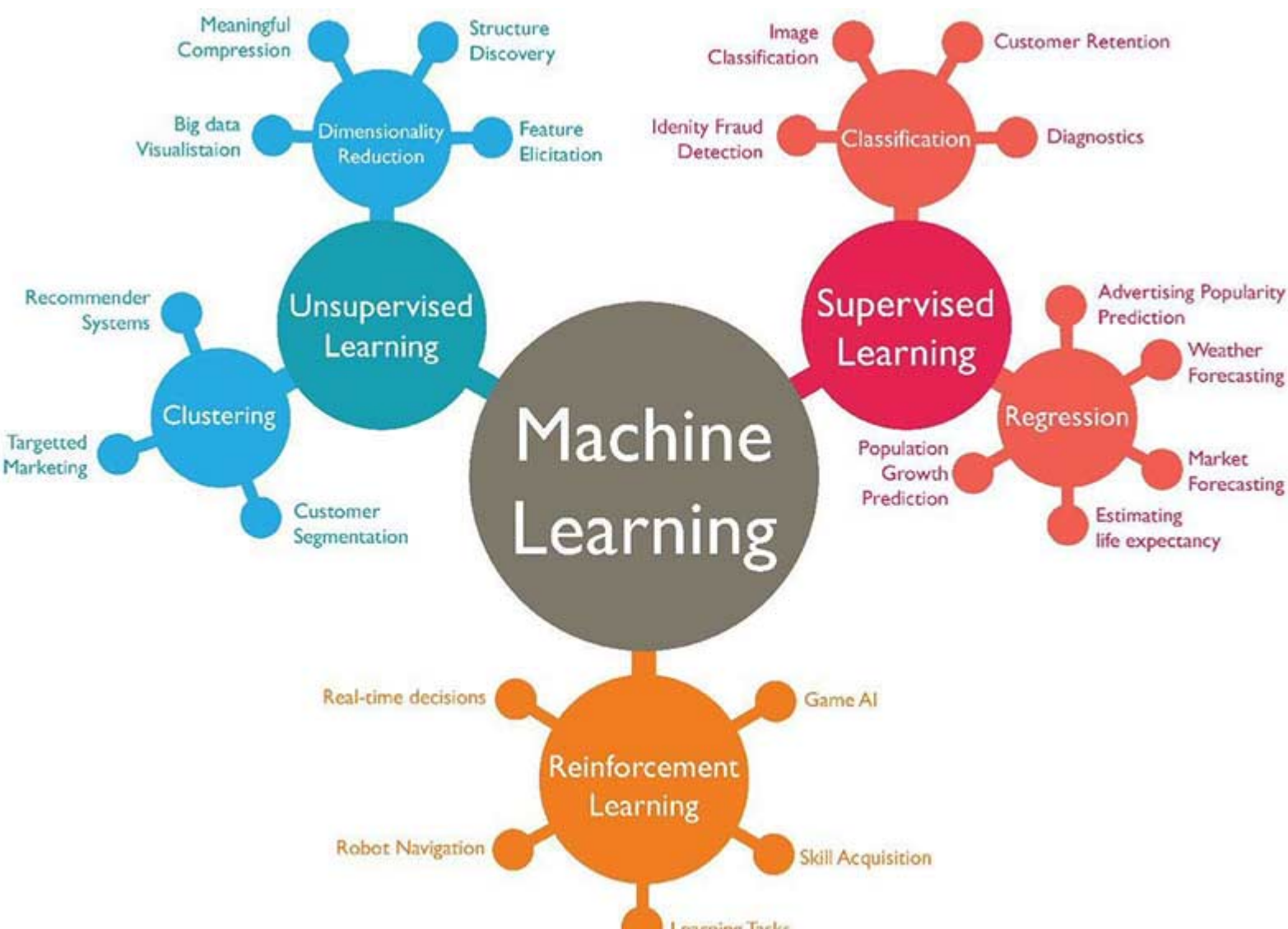
Machine Learning



When we use Machine Learning?

- Problems for which existing solutions require a lot of fine-tuning or long lists of rules. In this case one Machine Learning algorithm can often simplify code and improve the performance.
- Complex problems for which using traditional approach(for example search algorithms on graphs like DFS or BFS) can't give us good solutions.
- fluctuating System(System that their data tend to change) - A ML(Machine Learning) system can adapt to new data.
- Getting Insight about complex problems and large amount of data

Types of Machine Learning



photo's source: <https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08>

Supervised Learning

Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labeled data. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution, and data points to be dealt with. The training dataset is also very similar to the final dataset in its characteristics and provides the algorithm with the labeled parameters required for the problem.

The algorithm then finds relationships between the parameters given, essentially establishing a cause and effect relationship between the variables in the dataset. At the end of the training, the algorithm has an idea of how the data works and the relationship between the input and the output.

This solution is then deployed for use with the final dataset, which it learns from in the same way as the training dataset. This means that supervised machine learning algorithms will continue to improve even after being deployed, discovering new patterns and relationships as it trains itself on new data.

Types of Supervised Learning

1. **Classification** - A classification problem is when the output variable is a category. such as "red" or "blue", "Positive" or "Negative" and etc.
2. **Regression** - A regression problem is when the output variable is a real value, such as predict revenue or weight.

Unsupervised Learning

Unsupervised machine learning holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program.

In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings.

The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post-deployment development than supervised learning algorithms.

Reinforcement Learning

Reinforcement learning directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged or 'reinforced', and non-favorable outputs are discouraged or 'punished'.

Based on the psychological concept of conditioning, reinforcement learning works by putting the algorithm in a work environment with an interpreter and a reward system. In every iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not.

In case of the program finding the correct solution, the interpreter reinforces the solution by providing a reward to the algorithm. If the outcome is not favorable, the algorithm is forced to reiterate until it finds a better result. In most cases, the reward system is directly tied to the effectiveness of the result.

In typical reinforcement learning use-cases, such as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

Recommendations(Recommender Systems)

A recommendation system/engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a customer and based on that, recommends products which the users might be likely to buy.

Basic Definitions in ML

1. Learning Algorithm

A learning algorithm is a method used to process data to extract patterns appropriate for application in a new situation. The goal is to adapt a system to a specific input-output transformation task.

2. Training data

Training data is the data you use to train an algorithm or machine learning model to predict the outcome you design your model to predict.

If you are using supervised learning or some hybrid that includes that approach, your data will be enriched with data labeling or annotation.

When we use a Dataframe/Table in machine learning, rows and columns represent **observations** and **features** respectively. The feature that we need to predict is referred to as **target**.

3. Learning Model

A machine learning model is a file that has been trained to recognize certain types of patterns. We train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

Once you have trained the model, you can use it to reason over data that it hasn't seen before, and make predictions about those data.

4. Evaluation Metrics

Not every model is good and we use **Evaluation Metrics** to measure the quality of the model: Lets check 4 kinds of used Evaluation Metrics.

Precision is how close measure values are to each other, basically how many decimal places are at the end of a given measurement. The precision is ratio of how much observations that predicted as positive are True positive.

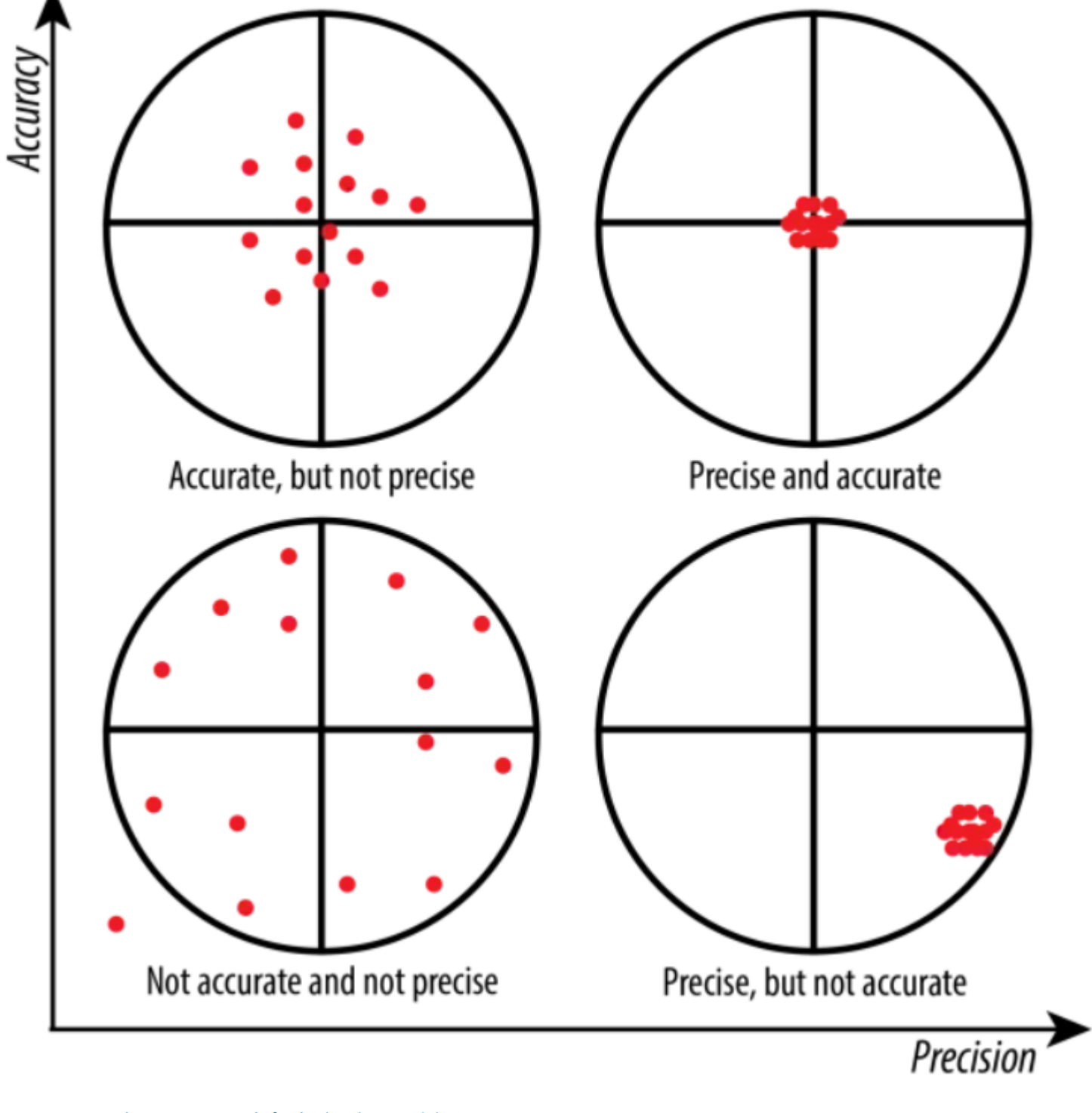
$$Precision = \frac{TP}{TP+FP}$$

when TP = the number of True Positive and FP = the number of False Positive.

Accuracy is how close a measure value is to the true value. Accuracy matters too, but it's best when measurements are both precise and accurate.

$$Accuracy = \frac{N-E}{N}$$

when N = total number of questions and E = number of errors



photo's source: <https://wp.stolaf.edu/it/gis-precision-accuracy/>

Recall ratio of the observations that are Positive are founded by the model.

$$Recall = \frac{TP}{TP+FN}$$

when TP = the number of True Positive and FN = the number of False Negative.

F1 Score is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.

The F-score is a way of combining the precision and recall of the model.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5. Overfitting vs Underfitting

Machine learning algorithms learn by training with data. The algorithms are so powerful that they can learn each and every detail of the training data. However, we do not want this.

The main purpose of training a model is to be able to use them on new, previously unseen data (e.g. test data). There will always be small differences between the training data and test data. Thus, a model should generalize well instead of memorizing the details of training data.

Overfitting arises when a model tries to fit the training data so well that it cannot generalize to new observations. Well generalized models perform better on new observations. If a model is more complex than necessary, it is highly likely we end up with overfitting.

In a sense, **Underfitting** is the opposite of overfitting. An underfit model fails to capture enough detail on training data. Thus, it does not perform well on either training or test data.

Sources

- [Introduction to Machine Learning for Beginners - Towards Data Science](#)
- [A Glossary for Machine Learning - Towards Data Science](#)

