Chapter 1: The Machine Learning Landscape

What Is Machine Learning?

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

—Tom Mitchell,1997

In my words, it is to train on data to make predictions.

Why Use Machine Learning?

Machine Learning is great for:

- Problems for which existing solutions require a lot of fine-tuning or long lists of rules: one Machine Learning model can often simplify code and perform better than the traditional approach.
- Complex problems for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution.
- Fluctuating environments: a Machine Learning system can easily be retrained on new data, always keeping it up to date.
- Getting insights about complex problems and large amounts of data.

Examples of Applications

- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans
- Automatically classifying news articles
- Automatically flagging offensive comments on discussion forums
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Forecasting your company's revenue next year, based on many performance metrics
- Making your app react to voice commands
- Detecting credit card fraud
- Segmenting clients based on their purchases so that you can design a different marketing strategy for each segment
- Representing a complex, high-dimensional dataset in a clear and insightful diagram

- Recommending a product that a client may be interested in, based on past purchases
- Building an intelligent bot for a game

<u>And my suggestion is: Detecting mental illnesses.</u>

Types of Machine Learning Systems

Training Supervision:

Supervised learning: In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels.
A typical supervised learning task is classification.
Another typical task is to predict a target numeric value, this sort of task is called regression.
some regression models can be used for classification as well, and vice versa.

Unsupervised learning: The training data is unlabeled.The system tries to learn without a teacher.

Semi-supervised learning: It deals with data that’s partially labeled.
Most semi-supervised learning algorithms are combinations of unsupervised and supervised algorithms.

Self-supervised learning: It is generating a fully labeled dataset from a fully unlabeled one.
Once the whole dataset it labeled, any supervised learning algorithm can be used.
You usually want to tweak and fine-tune the model for a slightly different task. One that you actually care about.
Some people consider self-supervised learning to be a part of unsupervised learning, since it deals with fully unlabeled datasets.
But self-supervised learning uses (generated) labels during training, so in that regard it’s closer to supervised learning.
self-supervised learning focuses on the same tasks as supervised learning: mainly classification and regression.

Reinforcement Learning: The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return or penalties in the form of negative rewards. It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

Batch vs Online Learning

In batch learning, the system is incapable of learning incrementally: it must be trained using all the available data. This will generally take a lot of time and computing resources, so it is typically done offline. First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned. This is called offline learning.

In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or in small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives.

Instance-Based Vs Model-Based Learning

In instance-based learning, the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples (or a subset of them).

is to build a model of

Model-based learning is to build a model of these examples and then use that model to make predictions.

Main Challenges of Machine Learning

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

Testing and Validating

The only way to know how well a model will generalize to new cases is to actually try it out on new cases. One way to do that is to put your model in production and monitor how well it performs. A better option is to split your data into two sets: the training set and the test set. As these names imply, you train your model using the training set, and you test it using the test set. The error rate on new cases is called the generalization error (or out-of-sample error), and by evaluating your model on the test set, you get an estimate of this error. This value tells you how well your model will perform on instances it has never seen before.