What is Machine Learning

Tal Linzen

Center for Data Science, NYU

Jan 25, 2022

Machine Learning Problems

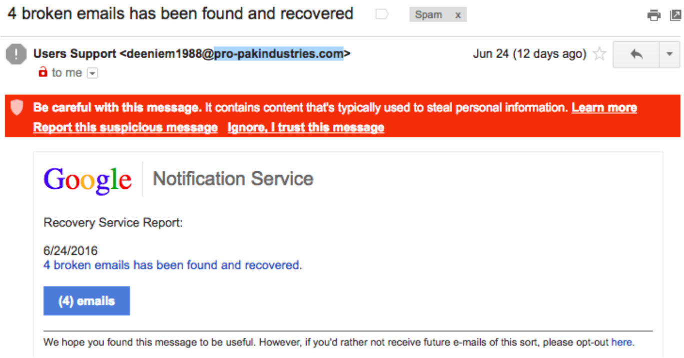
Typically our goal is to solve a prediction problem of the format:

- Given an **input** x,
- **Predict** an **output** y.

We'll start with a few canonical examples.

Example: Spam Detection

- **Input:** Incoming email



- **Output:** “SPAM” or “NOT SPAM”

- This is a **binary classification** problem: there are two possible outputs.

Example: Medical Diagnosis

- **Input**: Symptoms (fever, cough, fast breathing, shaking, nausea, ...)
- **Output:** Diagnosis (pneumonia, flu, common cold, bronchitis, ...)
- A **multiclass classification** problem: choosing an output out of a *discrete* set of possible outputs.

How do we express uncertainty about the output?

- **Probabilistic classification** or **soft classification**:

$$
\begin{aligned}
\mathbb{P}(\text{pneumonia}) &= 0.7 \\
\mathbb{P}(\text{flu}) &= 0.2 \\
\vdots \quad &\quad \vdots
\end{aligned}
$$

Example: Predicting a Stock Price

- **Input:** History of the stock's prices
- **Output:** The price of the stock at the close of the next day
- This is called a **regression** problem (for historical reasons): the output is *continuous*.

Comparison to Rule-Based Approaches (Expert Systems)

- Consider the problem of medical diagnosis.
  1. Talk to experts (in this case, medical doctors).
  2. Understand how the experts come up with a diagnosis.
  3. Implement this process as an algorithm (a **rule-based system**): e.g., a set of symptoms $\rightarrow$ a particular diagnosis.
  4. Potentially use logical deduction to infer new rules from the rules that are stored in the knowledge base.

Rule-Based Approach

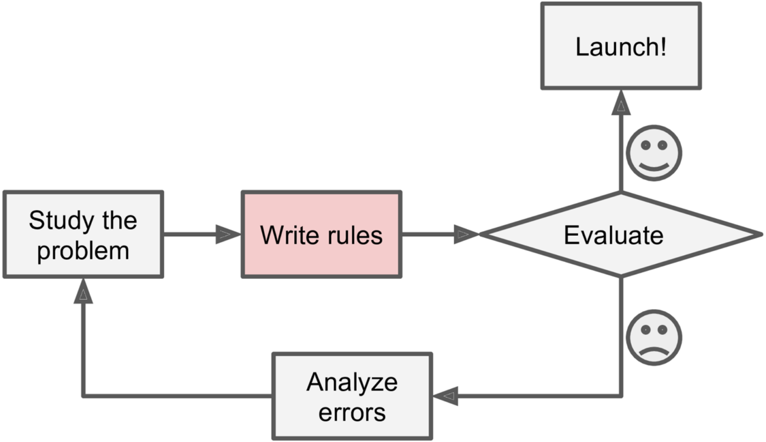


Fig 1-1 from *Hands-On Machine Learning with Scikit-Learn and TensorFlow* by Aurelien Geron (2017).

Advantages of Rule-Based Approaches

- Leverage existing domain expertise.
- Generally **interpretable**: We can describe the rule to another human
- Produce reliable answers for the scenarios that are included in the knowledge bases.

Limitations of Rule-Based Systems

- Labor intensive to build: experts' time is expensive.
- Rules work very well for areas they cover, but often do not **generalize** to unanticipated input combinations.
- Don't naturally handle uncertainty.

The Machine Learning Approach

- Instead of explicitly engineering the process that a human expert would use to make the decision...
- We have the machine **learn** on its own from inputs and outputs (decisions).
- We provide **training data**: many examples of (input x , output y) pairs, e.g.
  - A set of videos, and whether or not each has a cat in it.
  - A set of emails, and whether or not each one should go to the spam folder.
- Learning from training data of this form (inputs and outputs) is called **supervised learning**.

Machine Learning Algorithm

- A **machine learning algorithm** learns from the training data:
  - **Input**: Training Data (e.g., emails x and their labels y)
  - **Output**: A prediction function that produces output y given input x.
- The goal of machine learning is to find the "best" (to be defined) prediction function **automatically, based on the training data**
- The success of ML depends on
  - The availability of large amounts of data;
  - **Generalization** to unseen samples (the test set): just memorizing the training set will not be useful.

Machine Learning Approach

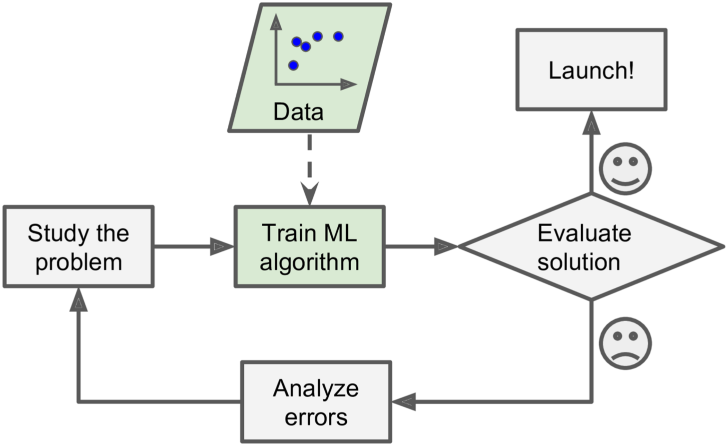


Fig 1-2 from *Hands-On Machine Learning with Scikit-Learn and TensorFlow* by Aurelien Geron (2017).

Key concepts

- The most common **ML problem types**:

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function
- Beyond prediction

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function
- Beyond prediction
  - **Unsupervised learning**: finding structures in data, e.g. clustering

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function
- Beyond prediction
  - **Unsupervised learning**: finding structures in data, e.g. clustering
  - **Reinforcement learning**: optimizing long-term objective, e.g. Go

Key concepts

- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function
- Beyond prediction
  - **Unsupervised learning**: finding structures in data, e.g. clustering
  - **Reinforcement learning**: optimizing long-term objective, e.g. Go
  - **Representation learning**: learning good features of real-world objects, e.g. text

Core Questions in Machine Learning

Given any task, the following questions need to be answered:

- **Modeling**: What class of prediction functions are we considering?
- **Learning**: How do we learn the "best" prediction function in this class from our training data?
- **Inference**: How do we compute the output of the prediction function for a new input?