L01: The Fundamentals of Machine Learning

CSci 574 Machine Learning (Géron) Ch. 1

Derek Harter

Department of Computer Science
East Texas A&M University

Summer 2025

EAST TEXAS A&M

What is Machine Learning?

Machine learning is the science (and art) of programming computers so they can **learn from data**.

Some definitions that may more or less be useful:

- Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed (Samuel, 1959).
- 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` (Mitchell, 1997).

EAST TEXAS A&M

Artificial Intelligence, Machine Learning, Deep Learning, Generative AI

- **Artificial Intelligence** (AI): *the effort to automate intellectual tasks normally performed by humans*
- **Machine Learning** (ML): *machine looks at input and answer and figures out the rules*
- **Deep Learning** (DL, DNN): *learning successive layers of increasingly meaningful representations*
- **Generative AI** (GenAI): *extend from reactive to creative activities*

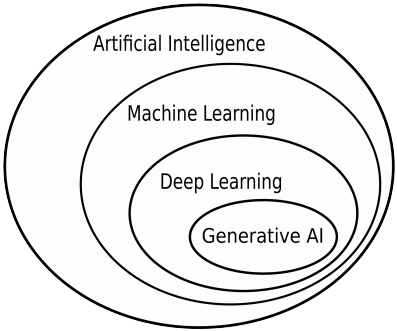


Figure 1: Relation of AI with ML, DL and GenAI. Each is a more specialized subset of the larger discipline. (Chollet, 2021, pg.2)

EAST TEXAS A&M

Why Use Machine Learning?

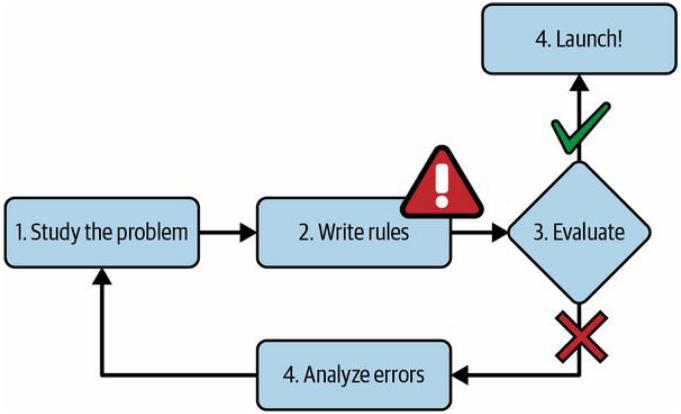


Figure 2: Traditional approach to building a spam filter. (Géron, 2023, ch.1)

The traditional way to write a spam filter:

1. Examine what spam typically looks like and collect patterns.
2. Write a detection algorithm for each of the patterns that you identified.
3. Test program and repeat steps 1 and 2 until it was good enough at detecting spam.

EAST TEXAS A&M

Why Use Machine Learning?

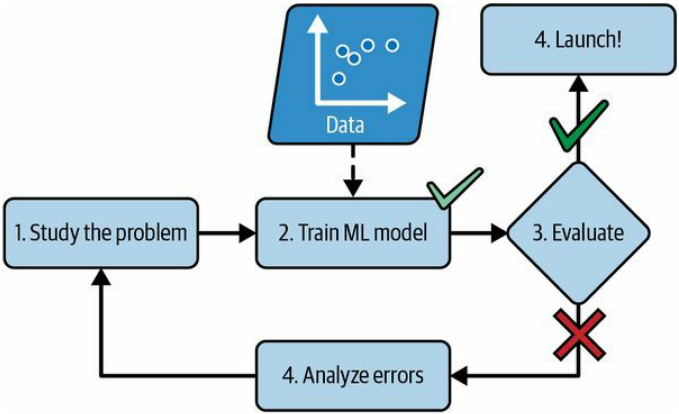


Figure 3: Machine learning approach, have ML model learn rules form data. (Géron, 2023, ch.1)

The ML way to write a spam filter:

1. Collect data and label the data as spam or not span.
2. Train a ML model to learn the rules from the labeled data.
3. Test program and repeat steps 1 and 2 until it was good enough at detecting spam.

EAST TEXAS A&M

Machine Learning

- Usually have human programmer write down rules (a computer program) that turns input into appropriate answers.
- In ML paradigm, you give a ML algorithm the input and answers, and it "learns" the rules.

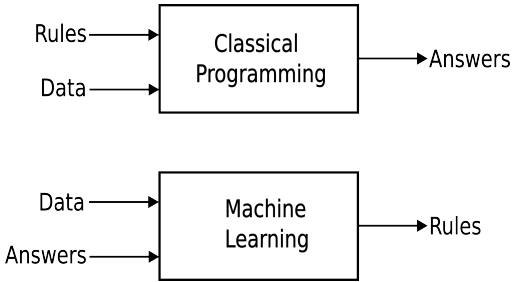


Figure 4: Machine Learning coordinate change and learning representations.. (Chollet, 2021, Fig 1.2)

EAST TEXAS A&M

Learning Rules and Representations from Data

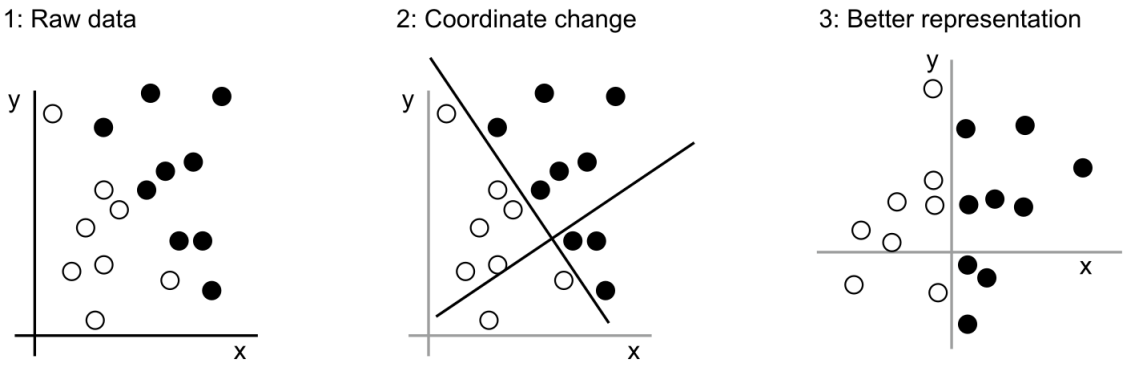


Figure 5: ML a new programming paradigm. (Chollet, 2021, Fig. 1.4)

Some Benefits of ML vs. Traditional Programming

- Problems for which existing solutions require a lot of fine-tuning or long lists of rules might be simpler and more maintainable as ML systems that learn rules from data.
- Complex problems for which a traditional approach is too complex for humans usually to program a traditional solution directly might be amenable to a ML algorithm.
- Fluctuating environments might make it difficult to keep traditional rule based systems updated, will be easier to update for shifting targets by retraining ML models.
- (Some) ML approaches can inspect their solutions to gain insights into complex problems and large amounts of data.

EAST TEXAS A&M

Types of Machine Learning Systems

We can classify ML using several broad categories

- How are they supervised during training
  - supervised learning
  - unsupervised learning
  - self-supervised and reinforcement type learning
- Whether the ML algorithm can learn incrementally on the fly, or not (online learning vs. batched learning).
- Whether they work by simply comparing new data points to known data points, or instead build a predictive model that interpolates and detects patterns in the training data.
  - Instance-based system
  - Model-based machine learning

EAST TEXAS A&M

Supervised Learning

Supervised Learning: the training set you feed to the algorithm includes the desired solution, called **labels** or **target** values.

Classification consists of labels with discrete values. For example, binary classification has 2 labels, like **spam** vs. **ham**.

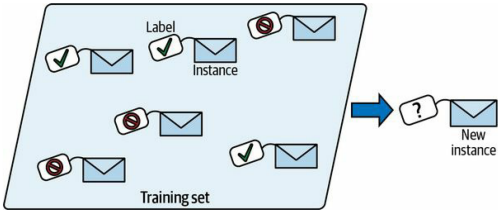


Figure 6: A labeled training set for spam classification, an example of supervised learning. (Géron, 2023, Fig 1.5)

Regression the target value is a numeric (continuous) value, like predict the price of a house based on its size and age.

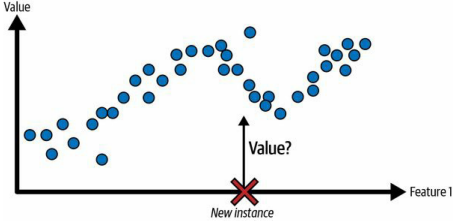


Figure 7: A regression problem: predict a value given an input feature. (Géron, 2023, Fig 1.6)

EAST TEXAS A&M

Unsupervised Learning

In **unsupervised learning** the training data is unlabeled, the system tries to learn without a teacher.

Clustering detect groups of similar items in data (e.g. possible category labels)

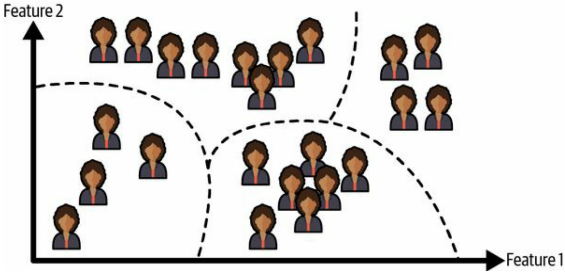


Figure 8: Example of clustering an unlabeled set of data. (Géron, 2023, Fig 1.8)

Visualization like dimensionality reduction and feature extraction.

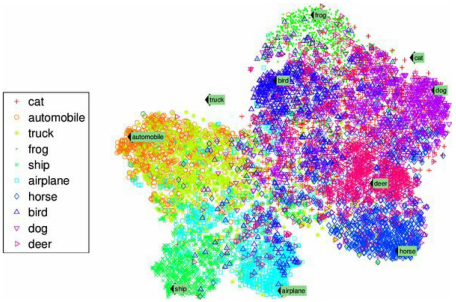


Figure 9: Example of a t-SNE 2D dimension reduction visualizing semantic clusters. (Géron 2023, Fig 1.9)

Batch vs. Offline Learning

- In **batch learning** the system is incapable of learning incrementally: it must be trained using all the available data.
  - Typically takes time so needs to be done *offline*.
  - First trained, then once works sufficiently well, put into production.
  - In many situations, performance decays because world continues to evolve while the model remains uncahnged. **model rot** or **data drift**
- **online learning** train system incrementally by feeding it data sequentially or in small groups (*mini-batches*)
  - Useful for systems that need to adapt to change extremely rapidly.
  - Also for huge datasets that cannot fit in a typical computer memory *out-of-core* learning.

EAST TEXAS A&M

Instance-Based vs. Model-Based Learning

The goal is given a number of training examples, the system needs to be able to make good predictions (**generalization**) on examples it has never seen before.

Instance-based learning: measure similarity to known instances, and predict most similar or average of 3 most similar (KNN).

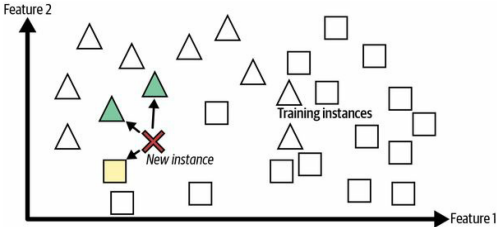


Figure 10: Measure similarity and take 3-closest neighbors (e.g. K-nearest neighbors). (Géron, 2023, Fig 1.16)

Model-based learning: build a model of labeled examples, and use the model to make *predictions* e.g. generalize to unseen input.

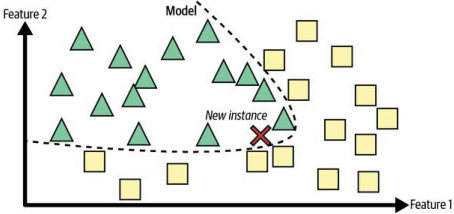


Figure 11: Learn a model based on labeled training data, for classification results in partitioning the space using decision boundaries. (Géron, 2023, Fig 1.17)

Example of Model-Based Learning: Linear Regression

Given a set of data that looks like the following:

$$\text{satisfaction} = \theta_0 + \theta_1 \times \text{GDP} \tag{1}$$

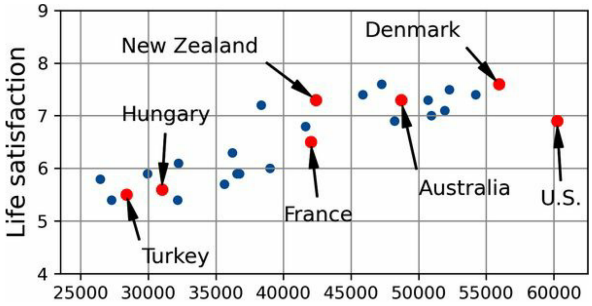


Figure 12: Data possible linear relationship of GDP to satisfaction. (Géron, 2023, Fig 1.18)

EAST TEXAS A&M

Example of Model-Based Learning: Linear Regression

This is a linear regression. But which model is the "best" fitting model of this data? We will need a **cost** function to measure the goodness or badness of fit of different models to choose.

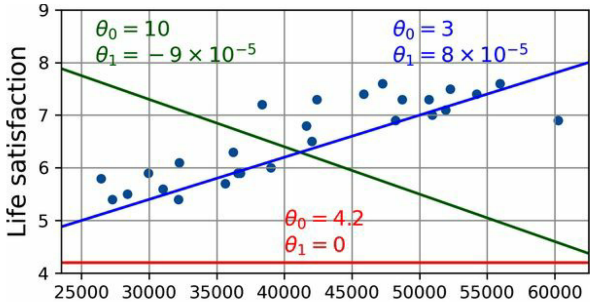


Figure 13: A few possible linear models among the infinite number of possible lines. (Géron, 2023, Fig 1.19)

EAST TEXAS A&M

Challenges of ML: Bad Data

- Insufficient quantity of training data: it takes a LOT of data for ost ML algorithms to fit a model properly.
- Nonrepresentative training data: in order to generalize well, your training data must be representative of the new cases you want to generalize to.
- Poor-quality data: if the data is full of errors, outliers missing data and noise, it will make it harder for a ML algorithm to create a model that generalizes well to the underlying patterns. **data cleaning**
  - Remove outliers
  - Fill in or interpolate missing data
- Irrelevant features: features can be gathered that have no predictive power for the target, garbage in garbage out. **feature engineering**
  - *feature selection* select most useful / correlated features for training and detect nonpredictive ones
  - *feature extraction* combine existing features to produce more useful ones

EAST TEXAS A&M

Challenges of ML: Bad Model

- **Overfitting**: model performs well on the training data, but it does not generalize well
  - simplify / regularize the model
  - gather more training data
  - reduce the noise in the training data
- **Underfitting**: model is too simple to learn the underlying structure of the data
  - select a more powerful model with more parameters
  - feed better features to the learning algorithm (feature engineering)
  - reduce the constraints on the model (remove regularization)

EAST TEXAS A&M

Evaluating Generalization Performance: Train and Test Sets

The only way to know how well a model will generalize to new cases is to actually try it out on new cases.

- Standard practice, always split data into a **training set** and a **test set**:
  - **training error**: train model using the training set,
  - **generalization error**: evaluate a fitted model to see how well it generalizes on the test set.
  - If training error is low but generalization error is high it means model is **overfitting**.

EAST TEXAS A&M

Hyperparameter Tuning and Model Selection

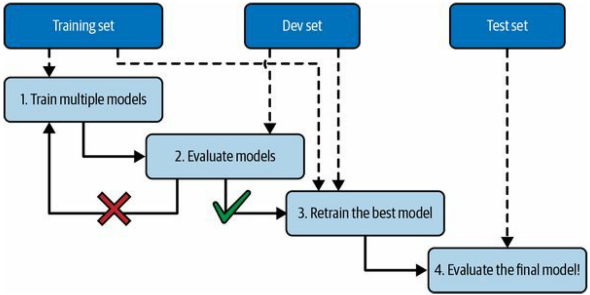
- It is common to try many different kinds of models (model selection), and to fit a type of model with many different values of **hyperparameters** (hyperparameter tuning)
- If you evaluate generalization multiple times on the test set, you are indirectly adapting model type and hyperparameters to produce best model for that particular test set.
  - This may end up not being completely indicative of overall generalization performance on a different unseen set of data.

Common Solution: **holdout validation** data

- Typically from the test set, again split into test and validation data.
  - Perhaps do this 1 time for all hyperparameter turning, or randomly holdout different validation data each time.
  - Iterate model selection and hyperparameter search evaluating on holdout validation data.
  - Best model / parameters for this search, then train a final model on full training/validation data
  - Final check of best model on held back test data then can determine if you overtuned hyperparameters and it doesn't really generalize well or not on real unseen data.

EAST TEXAS A&M

train / validation / test Set Model Selection



1. Train multiple models on reduced data set (training data minus held back validation data).
2. Evaluate different models and parameter searches generalization on held back validation data
3. Select best model from search based on validation performance, retrain final model on full train/validation data.
4. Evaluate final model on test data to get estimate of generalization error, and detect possible mistakes/leaks of information during model tuning search.

EAST TEXAS A&M

Summary

- Machine Learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules.
- There are many different categories of ML systems: supervised or unsupervised, batch or online, instance-based or model-based systems.
- In ML system you gather data in training set and you feed training set to learning algorithm.
  - If model-based it "fits" some parameters to the training set, which hopefully can make good predictions (generalize) on unseen data.
  - If instance-based, it saves/remembers the examples and uses a measure of similarity to find most similar item it knows to unknown item to make a prediction with.
- A ML system will not perform well if
  - your training set is too small
  - the data is not representative of what you ultimately want to make predictions of
  - the data is polluted with irrelevant features, is too noisy or has a lot of missing data
- A ML model-based system needs to be neither
  - too simple (underfitting)
  - too complex (overfitting)

EAST TEXAS A&M

Bibliography

- Chollet, F. (2021). *Deep learning with python* (second). Manning.
- Géron, A. (2023). *Hands-on machine learning with scikit-learn, keras and tensorflow* (third). O'Reilly Media, Inc.
- Mitchell, T. M. (1997). Does machine learning really work? *AI magazine*, *18*(3), 11–11.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210–229.

EAST TEXAS A&M