ML intro

ML definition: we will learn a task from input data (e) and expect the result to improve given more input data.

ML paradigms

Labeled data X, y ⇒ we know that input X gives response y

Supervised Learning

Given labeled data consisting of features X and response y, find a model y = f(X)

if y continuous ⇒ regression

if y discrete ⇒ classification

Unsupervised Learning

Given unlabeled dataset X, fund a structure. E.g. group of customers into different categories based on their purchase patterns.

Semi-supervised learning

Improve unsupervised learning by making use of a set of labeled data. Or vice versa, improve supervised learning by making use of unlabeled data.

Supervised Learning - Goals and Results

We start with some data X (called the training set), and corresponding responses y (called labels).

Our aim is to ‘learn’ an optimal function f, called the **hypothesis**, such that

$$y = f(x) + epsilon$$

where epsilon is the error term representing the error. Optimal f minimizes the error.

Unsupervised Learning

Often referred to as a descriptive task whereas supervised learning is a predictive task.

Notations and Vocabulary

Dataset

each row is a *sample* or *observation*

the columns are *features* or *attributes*

the result is a *response* or *label*

Feature Types

discrete vs continuous

numeric, nominal, or ordinal

name	example	description
numeric	Temperature	numeric value
ordinal	grades (A-F)	the objects can be ranked by this feature
nominal	color	the object can be categorized (but not ranked) by this feature

Data preprocessing

Sometimes the learning itself requires us to preprocess the data in order to facilitate learning. Some common strategies for preprocessing data are:

- dimensionality reduction
- feature subset selection
- feature extraction
- discretization
- normalization or standardization

Model (parametric) vs instance (non-parametric) based learning

Parametric example: When searching for $y = f(X)$, assume $f(x)$ of the form $y = \beta 0 + \beta 1*X + \beta 2*X^2 + \beta 3*X^3$, where βi are our parameters.

Model-based learning: assumes some fixed set of parameters β which describes future predictions independent of the observed data.

Non-parametric example: KNN

Instance-based learning: No model, we use the entire training set to make a prediction. Don't make assumptions about the functional form and instead directly estimate the underlying data distribution.

Hyper-parameters

often denoted by alpha, are set before the learning procedure begins, such as: for how many interactions should we allow this training to proceed?

Summary: parameters beta are part of the model and will be decided during training, hyper-params alpha are set before training and configured during the training phase.

Flexibility and interpretability

Some models are more restricted than others, such as linear regression. This is not a very flexible model, but to its advantage, it's very interpretable, the response Y is described as a linear function of the variables X.

On the other hand, some models are not as interpretable, such as SVM, but are very flexible in the sense that they can capture many different forms of data.

Model Quality

The choice of method to determine model quality depends on several factors:

- Type of task
- Your data
- Domain knowledge, e.g. spam-detection

Quality of Classification Models

often expressed using a confusion matrix:

	actual yes	actual no
predicted yes	TP = 100	FP = 10
predicted no	FN = 5	TN = 50

number of samples = 50 + 10 + 5+ 100 = 165

sample yes count = 100 + 5 = 105

predicted yes count = 100 + 10 = 110

number of correct classifications: 50 + 100 = 150

number of errors = 5 + 10 = 15

accuracy and error rate:

- accuracy = (50 + 100) / 165 = 0.91,
- error rate = (5 + 10) / 165 = 0.09

Quality of Regression Models

(yi - f(xi))^2 ⇒ squared y-distance between estimate f(xi) and actual value yi

The most common regression error measure is mean squared error (MSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2.$$

The MSE will be small if the discrepancy between the estimate model f and the response Y is small over all training examples.

loss function: measures the discrepancy between the function estimate and the response

cost function: it's averaged sum over all training examples. It quantifies the model's performance and provides a measure of how well the model is fitting the training data.

The Root Mean Square Error (Mean euclidian distance)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2} = \sqrt{MSE}$$

The Mean Absolute Error (Manhattan distance)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(x_i)|.$$

Test set and test error

Training set

Used to build the model

Used to select which type of model to use

Uset to fine-tune the hyper-parameters

The error measured by the training set is called the training error.

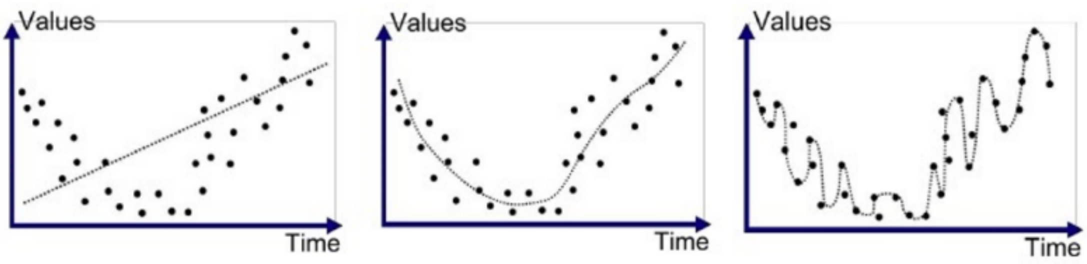
Test set

Used to evaluate the model

Represents unseen data

The error measured by the test set is called the test error, hence, we determine whether a model is good by its test error

Over and underfitting



Underfitted Good Fit/Robust Overfitted

Underfitting: The model is not flexible enough. Large test and training errors, no matter what size of the dataset.

Overfitting: Model much too flexible ⇒ makes unrealistic fitting to given training set ⇒ very small training error and large test error.