

Machine Learning

A Gentle introduction

Roozbeh H. Pazuki

Uncertainty

Uncertainty is everywhere.

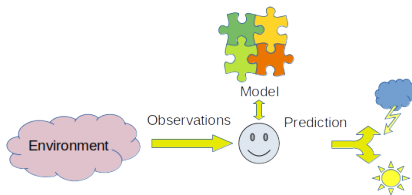
- Next train arrival time.
- Tomorrow's weather.
- Next month's inflation rate.
- The sex of an unborn baby.
- Finding a sit in your favourite restaurant.
- How many people will order food from the seniors' canteen today?
- Your mark and performance in each course.
- How long do we live?

Uncertainty

Or ones with less clear-cut answers.

- How long does it take to boil a kettle?
- Is fast food good or bad for health?
- Number of stars in Milky Way galaxy.
- Number of cells in your liver.
- How many rain droplets would wet my cloth if I ran from the Sherfield building to the SEC?
- How fast should I move my racket to hit the ball?

Uncertainty



- At any instance, we continuously eliminate the uncertainty in our decision-making and planning for the future.
- We learn from our experiences (**observation**), and based on that, we construct a notion for what should be expected (**model**). And finally, we guess about the future (**prediction**).

Uncertainty

The quality of our guesses depends on the following:

- 1 **The quality and amount of past data.** e.g., I have used the bus route since last week or last year – Of course, it was not an issue if the past observations were infinite, but we know it is impossible.
- 2 **The model's suitability and sensibility of expectations.** e.g. expecting one day, there would be no bus at all since the TFL is on strike.

Uncertainty

- ③ **The predictability of the phenomena.** e.g., there is a difference between the predictability of the next bus arrival time, the closing price of a stock, or the upcoming lottery winner number.
- ④ **Considering the inaccuracy in the prediction and having an estimate of its error.** e.g. putting all the eggs in one basket – No matter how experienced or good we are at guessing, we can never be sure about the accuracy of the guess. Otherwise, the future is not uncertain.

Sources of Uncertainty

- **Physical Uncertainty.** e.g. Heisenberg Uncertainty principle.
- **Lack of complete knowledge.** e.g. Weather systems, stock exchange, molecular details of gene expression inside a specific cell, ...
- **Measurement.** e.g. Every lab measurement, a faulty instrument, a careless bookkeeper, or fluorescent imaging of a gene expression, ...

Sources of Uncertainty

Depending on the sources of uncertainty, there are usually two possibilities for the predictions we make:

- ① The phenomenon is deterministic, but the data is finite, or the model is not accurate or detailed. So, our prediction is inaccurate.
- ② The phenomenon is inherently probabilistic. So, the best that we can predict is a probability distribution.

A Machine That Can Learn

- Let's say we have some data (observations). Can we write a computer program that learns the data and predicts the future?
- Of course, "**machine learning**" is the practice of creating such a program.
- Nevertheless, what is true for us is also valid for our machine.

A Machine That Can Learn

The followings are similar for machines:

- ① The data for teaching the machine is finite and limited.
- ② Each program that can learn is inherently a **model**. It may or may not be a suitable choice for what we expect.
- ③ The phenomena that the machine predicts may be deterministic or stochastic.

How to Make a Program That Can Learn

What should we look for? Ask yourself:

- ① **Data:** Is it the right data to learn from? Is it enough?
- ② **Model:** Is this model expectation appropriate for the phenomena?
- ③ **Prediction accuracy:** How reliable is this prediction? Can we reduce its inaccuracy? Can we find a bound for its inaccuracy?

How to Make a Program That Can Learn

Examples:

- ❶ **Data:** Using bus arrival time is not appropriate for trains.
- ❷ **Model:** Not including the fact that some days train strike happens makes the model less suitable.
- ❸ **Prediction accuracy:** Even including all available timetables and using sophisticated models, e.g., we are sure we can not predict when faulty signalling causes havoc. So, the predictions are always inaccurate. But how much?

Practical ML

So, these are what you should remember as an ML practitioner:

- 1 **Data:** collect your data. Make sure it is high quality and contains relevant information.
- 2 **Model:** Select the right model by studying its assumptions.
- 3 **Prediction accuracy (Generalisation error):** Estimate the prediction error. If it is not good enough, start again.

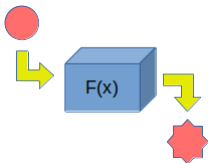
Practical ML

In this course, we focus on items two and three. However, collecting the data is the most important and time-consuming part of a machine learning workflow. As we all agree, collecting relevant data is one of the main objectives of any science project, and we leave it here.

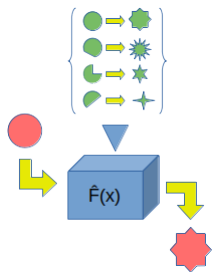
- ① **Data**
- ② **Model**
- ③ **Prediction accuracy (Generalisation error)**

Demystify the “*Model*”

- Almost all machine learning models are nothing but a **mathematical function**.
- It is easy to have an intuitive notion about a mathematical function when one imagines it as a black box that takes in **inputs** and turns it into **outputs**.



Demystify the “*Model*”



- To tackle the uncertainty by taking the experienced **observations** into account and turning them into a function, one can **predict** the future.
- In machine learning, **selecting** a **mathematical function** based on data is called **training**.

Demystify the “*Model*”

Let us give an example of how selecting a function works:

- 1 We assume polynomials are the class of functions that are the right choice for our modelling.
- 2 A scalar polynomial degree n , denotes by $f : \mathbb{R} \rightarrow \mathbb{R}$, in its general form writes as

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0,$$

for $n + 1$ coefficients $(a_n, a_{n-1}, \dots, a_0)$

- 3 And since there are infinite possibilities for coefficients $(a_n, a_{n-1}, \dots, a_0)$, therefore, there are infinite polynomial to select from.

Demystify the “*Model*”

- ④ For instance, for $n = 3$,

$$f_1(x) = x^3 + 2x^2 + \sqrt{2}x - 1,$$

and

$$f_2(x) = 7x^3 - 5x^2 - 4,$$

are two different polynomials.

- ⑤ At the same time,

$$f_3(x) = 5x^5 + 4x^4 - 9x^2 + x + 3,$$

and

$$f_4(x) = 3x^2 - x - 8.$$

are two polynomials with different degrees.

Demystify the “*Model*”

- As long as we believe the **family of polynomials** is the right choice for modelling the observed data and prediction, we can start by using an **algorithm** to feed the data, and the algorithm **selects** a polynomial among this family for us.
- In this case, **selecting** means finding the **best** coefficients and degrees by which
 - 1 The observed data fits.
 - 2 The least uncertain prediction can be made.
- In ML jargon, the family of polynomials is called the **model hypothesis**, and the set of all its functions is **hypothesis space**.

Demystify the “*Model*”

- Every machine learning technique has an algorithm for selecting a function from a **model hypothesis**.
- Based on the data we feed in, the algorithm must select the one that **best** describes the data from this family of functions.
- From the humble logistic regression to a deep learning model, even if it is a Large Language Model that can deceive us as a sentient being or a fancy algorithm that drives a car, all are a function that takes in the observations and turns them into a prediction about the future.
- It is the ML practitioner's job to ensure the model he or she selects has an appropriate hypothesis concerning the phenomena being modelled.

Demystify “*Selecting the Best Model*”

- Let us assume we are happy with our data and the model hypothesis. In other words, the data is ready, and we have a favourite algorithm and want to train it.
- How can we tell our favourite algorithm which functions to choose? Which one is the **best**?
- We have already answered these questions:
 - ① The model must fit the observed data as best as possible.
 - ② At the same time, it must also predict the unseen inputs as best as possible.
- So, we need a measure for the **goodness of fit** of the observed data and an estimation for the **error of unseen data**.

Demystify “*Selecting the Best Model*”

These measures are respectively called:

- 1 The **training performance metric**.
- 2 The **generalisation error**.

Let us see these concepts in action for a hypothetical data set.

Fitting a Function

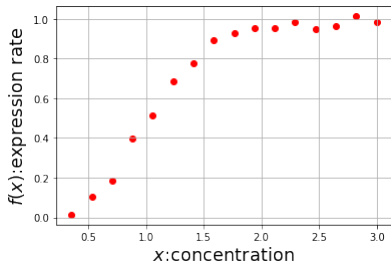


Figure 1: Training data.

- The y axis is the expression rate (observed fluorescent intensity), and x is the substrate concentration (an independent variable).
- The x can be a parameter in an experiment or a quantity that can change independently.
- The model assumes the existence of a causal relation $f(x)$ from x to y , for instance the Hill function.

Fitting a Function

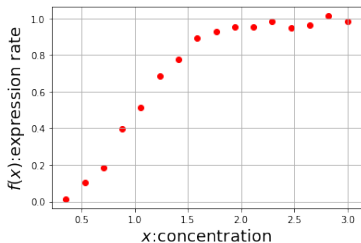


Figure 2: Training dataset.

- The data points are called **training dataset**.
- The causality assumption implies the function $f(x)$ models the dependence between x and y (Hill function).
- An approximated function $\hat{f}(x)$ exists such that it minimises the generalisation error.

Generalisation Error

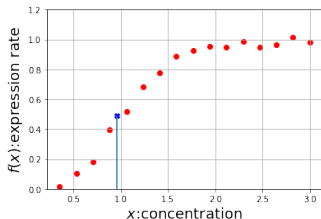


Figure 3: An new data point.

- For example, when the value of the fluorescent intensity at the concentration $x = 0.95$ is unknown, its value predicts as $y = f(0.95)$.
- However, by using machine learning to find a reasonable estimate for $f(x)$, there is always an error $\epsilon = \hat{f}(0.95) - f(0.95)$ that our candidate function $\hat{f}(0)$ will introduce for the point $x = 0.95$.

Generalisation Error

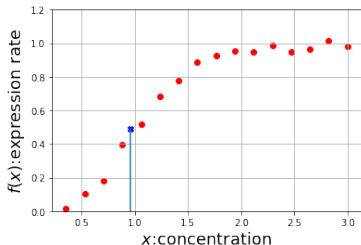


Figure 4: An new data point.

- $\mathbb{E}[\hat{f}(x) - f(x)]$, or the average error for all data points, is the generalisation error.
- Remember that we don't know $f(x)$, so we don't know the generalisation error either. We can only estimate that too, and hopefully, find a $\hat{f}(x)$ with a small generalisation error.
- One can say the main objective of an ML practitioner is **reducing the generalisation error**.

ML Cookbook

So, these are the ingredients for cooking an ML function:

- 1 A measure for the **training performance** (or how good our estimated function is).
- 2 A **function family** or **hypothesis space**.
- 3 An **algorithm** that uses the measure to select a function from this family.
- 4 If the algorithm has extra parameters, a strategy for **tuning** (which is called **validation**).
- 5 And finally, an estimation of **generalisation error**.

We will see them in action.

Curve Fitting - Measure

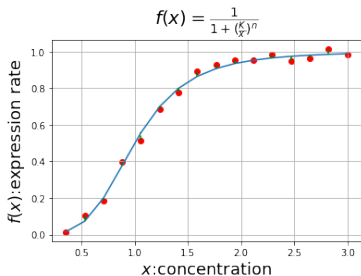


Figure 5: A fitted function.

- The curve on the left figure is one candidate for the estimated Hill function.
- Although it is not pass through all points, it is more or less very close to all.
- One way to measure its overall closeness is the sum of squares of deviations from the data (the green dash lines).

Curve Fitting - Measure

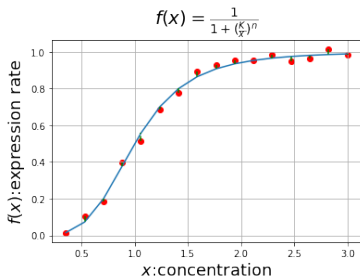


Figure 6: A fitted function.

- Mathematically, we define the measure as

$$\begin{aligned} \|\hat{f}(x) - f(x)\|^2 &= (\hat{f}(x_1) - f(x_1))^2 + \dots \\ &\quad + (\hat{f}(x_n) - f(x_n))^2 \\ &= \sum_{i=1}^n (\hat{f}(x_i) - f(x_i))^2 \end{aligned}$$

- This measure is called Euclidean distance, empirical error, mean square error, Norm-2, or even L^2 , and is your first choice for continuous quantities.

Curve Fitting - Family of function

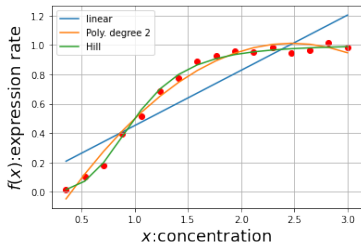


Figure 7: Different fitted functions.

- Now that we can measure how well the estimated function replicates the observed data, we need a family of functions.
- Naturally, we can calculate $||\hat{f}(x) - f(x)||^2$ for each function in the left figure and select the one with the smallest deviation.
- To test this idea in action, we assume the family of polynomials is a good choice

$$\hat{f}(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0.$$

Curve Fitting - Algorithm

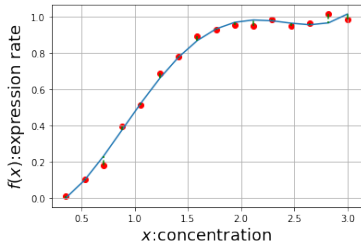


Figure 8: Polynomial degree 3.

- So, the goal is to find the coefficients of a polynomial (a_n, \dots, a_0) that has the least deviation for observed values.
- **Optimisation** is a branch of mathematics that studies such a task (e.g. check scipy optimisation package).
- One of the algorithms that can find the coefficients of a polynomial with the least $\|\hat{f}(x) - f(x)\|^2$ is called **least square**.

Curve Fitting - Algorithm

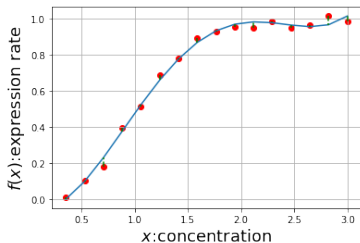
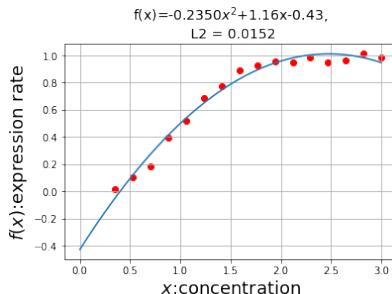
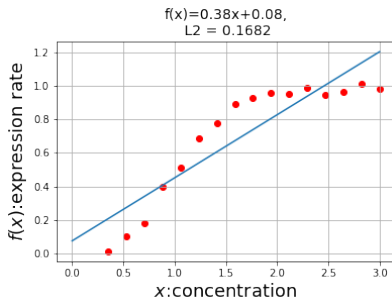


Figure 9: Polynomial degree 3.

- In practice, there are many implementations of an algorithm as different libraries in your favourite programming language, and usually, we need to tune their parameter.
- Fortunately, the least square algorithm for what we need is very modest and does not need too much.

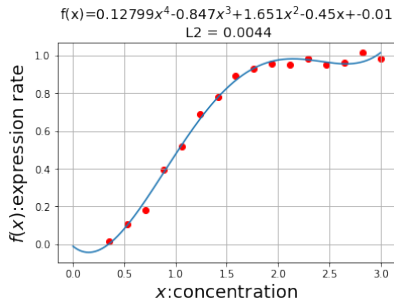
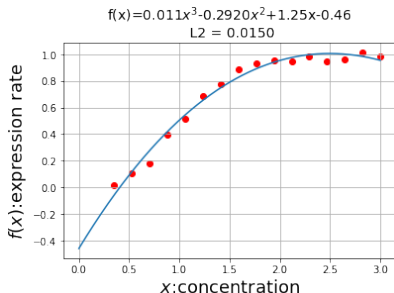
Curve Fitting - Selecting

Here we used the least square to fit four polynomials to find their coefficients.



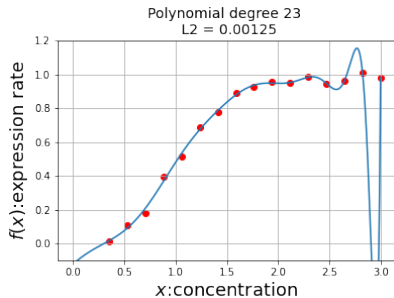
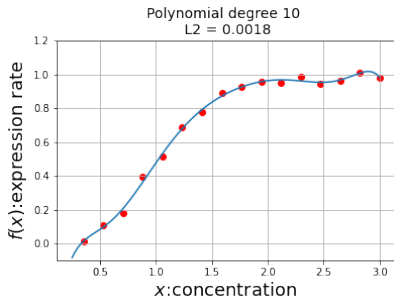
Curve Fitting - Selecting

The polynomial with the smallest $L^2 = \|\hat{f}(x) - f(x)\|^2$ is our choice. You can see that the polynomial degree four has the least L^2 among four of them.



Curve Fitting - Selecting

When we use polynomial degree ten, L^2 reduces to 0.0018. Even more, if we use polynomial degree 23, the curve passes through almost all points with $L^2 = 0.00125$. However, this curve makes a very bad judgment for the saturated expression rate.



Curve Fitting - Generalisation Error

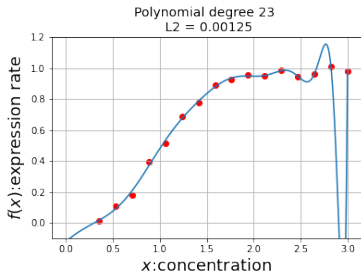


Figure 10: Polynomial degree 23.

- So, common sense tells us there is a balance between how well the estimated function, say $\hat{f}(x)$, can reproduce the observed data (L^2) while the generalisation error remains small.
- To estimate the generalisation error, one can set aside some of the training data. Since these data points are unseen in finding $\hat{f}(x)$, they can be used to estimate the generalisation error.

Curve Fitting - Generalisation Error

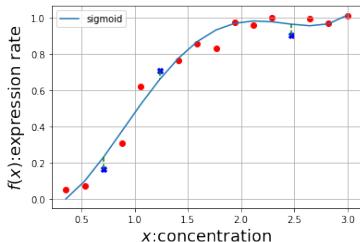


Figure 11: Test vs train data.

- For instance, we divide the data set into a **train set** (red dots) and a **test set** (blue crosses).
- Next, using the same algorithm, we can select polynomials with different degrees over the training dataset and their corresponding train costs.
- Finally, using the selected polynomials, the test set can be used to find the generalisation error.

Curve Fitting - Generalisation Error

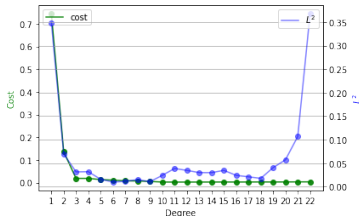


Figure 12: Cost vs generalisation error.

- The value of L^2 is an estimate for the generalisation error.
- We can see on the left figure the cost decreases as the degree of polynomials increases.
- Meanwhile, the generalisation error, or its L^2 estimate, first decreases and later increases.
- The sweet spot, or where we should select the estimated function, is at the minimum of the U shape of the generalisation estimate.

ML Cookbook - Recap

We did a workflow that all ML practitioners do to train a model.
These are your take-homes:

- 1 The most important quantity a model must reduce is the generalisation error.
- 2 Like a real-life situation, the estimated function must reduce uncertainty about the unseen data. The measure of that uncertainty is the generalisation error.
- 3 A trained model must be able to replicate the output very close to the observed values. But not exactly.

ML Cookbook - Recap

- ④ Otherwise, you will end with an estimated function that predicts poorly.
- ⑤ To prevent that and have an assessment of its prediction error. We set aside some of the training data as a test set. Then, after training, check the model's performance on the test set.
- ⑥ The balance between the accuracy of prediction on the training data and the test data is the sweet spot of model selection.
- ⑦ And finally, never use the test data on training time.

Classification

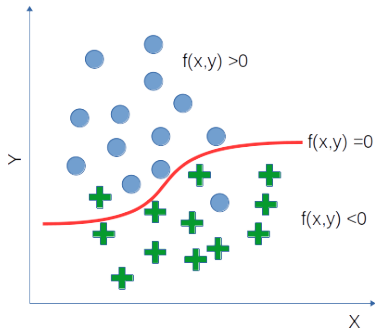


Figure 13: Classifying the data points to two categories.

- In the previous examples, the output was continuous data.
- For discrete labelled data points, the classification model divides the data into some subsets.
- For instance, in the left figure, the red line divides the circles and pluses into two subsets.
- The input of the function is the coordinates x and y .
- The class of the data point is determined by the sign of the function $f(x, y)$.

Type of Uncertainty

- 1 For deterministic phenomena whereby it always gets the same outputs for the same inputs, the values for new inputs are uncertain.

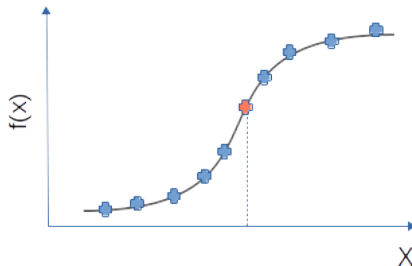


Figure 14: Deterministic function for interpolating the unseen values.

Type of Uncertainty

- 1 For deterministic phenomena with noise or error in measurement, the output varies around the average for a given input. So, the model finds a function for the expectation or a confidence interval.

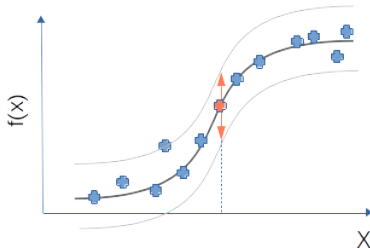


Figure 15: Deterministic function with noise.

Type of Uncertainty

- 1 For stochastic phenomena, which it always results in a distribution of outputs for the same inputs, the probability distribution is modelled.

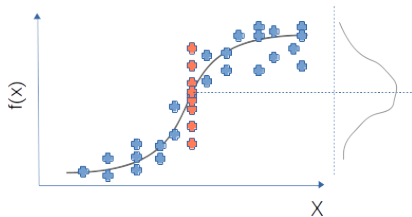


Figure 16: For a stochastic function, repeating the same experiment results in different values. So, the ultimate goal is estimating a distribution for any given input.

Types of ML Approaches

- ➊ **Supervised learning:** The dataset comprises inputs and outputs (the records are labelled).
- ➋ **Unsupervised learning:** The dataset comprises inputs only.
- ➌ **Reinforcement learning:** Learning by interacting with an environment.
- ➍ **Semi-supervised learning:** Incomplete labels.

Types of ML Approaches: Examples

- 1 **Supervised learning:** Images with labelled benign and metastasis cells. Sentences tokens labelled as a verb, subject, etc.
- 2 **Unsupervised learning:** Gene sequences or sentences.
- 3 **Reinforcement learning:** Game of checkers.
- 4 **Semi-supervised learning:** Protein sequence with incomplete classification.

Measures For Continuous Data

- ① **Euclidean distance:** for a D -dimensional dataset

$$\sqrt{(\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}})} = \sqrt{(x_1 - \hat{x}_1)^2 + \cdots + (x_D - \hat{x}_D)^2}.$$

- ② L_p norms:

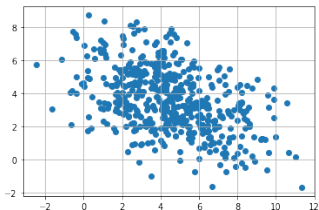
$$\|\mathbf{x} - \hat{\mathbf{x}}\|_p = [(x_1 - \hat{x}_1)^p + \cdots + (x_D - \hat{x}_D)^p]^{1/p}.$$

- ③ L_∞ norms:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_\infty = \max [|x_1 - \hat{x}_1|, \dots, |x_D - \hat{x}_D|].$$

Measures For Continuous Data

- ④ **Mahalanobis distance:** every $n \times n$ square matrix represents n directions. So, we can put some weights on measuring distance along each vector differently.



Measures For Continuous Data

① Formula

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{(\mathbf{x} - \hat{\mathbf{x}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \hat{\mathbf{x}})}.$$

② Example:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 6 & -2 \\ -2 & 3.5 \end{bmatrix}, \quad \mathbf{x} = (1, 2), \quad \hat{\mathbf{x}} = (2, 3),$$

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} 0.20 & 0.12 \\ 0.12 & 0.35 \end{bmatrix},$$

$$d(\mathbf{x}, \hat{\mathbf{x}}) = (1, 2) \cdot \begin{bmatrix} 0.20 & 0.12 \\ 0.12 & 0.35 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Measures For Discrete Data

- ① **Jaccard index:** Similarity of two sets is the ratio of the intersect volume to the union $J = \frac{|A \cap B|}{|A \cup B|}$. For example

$$A = \{a, b, c, d\}, \quad B = \{a, c, f\} \implies$$







$$A \cap B = \{a, c\}, \quad A \cup B = \{a, b, c, d, f\} \implies$$

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{5}.$$

- ② **Edit Distance:** Number of deletions, insertions and substitutions between two strings. For example,
 $d(\text{"ABBC"}, \text{"CBBA"}) = 2$.

Confusion Matrix

- ① **Binary Data:** Ratio of the correct match to the total number. The confusion matrix.

Prediction			
Real			
			
		True Positive	False Negative
		False Positive	True Negative

Confusion Matrix

Prediction \ Real	+	-
+	True Positive	False Negative
-	False Positive	True Negative

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}.$$

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

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

Baseline Models

Always try to find a **Null Model** to have a baseline for your model performance. Some good starts:

- ① **Pure chance.**
- ② **Naive Bayesian** or **Logistics regression** for classification of discrete output.
- ③ **Linear regression** for continuous output.
- ④ **Two dense layer neural network** for high-dimensional inputs. e.g. images.

We will try the Naive Bayesian and the Logistics regression in the lab.

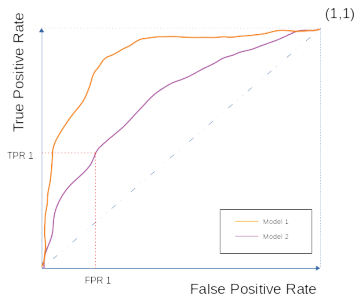
Model Performance: ROC curve

The Receiver Operating Characteristic curve, or the ROC, shows a classification model's performance at different thresholds.

① **True Positive Rate (Recall)** = $\frac{TP}{TP+FN}$

② **False Positive Rate** = $\frac{FP}{FP+TN}$

Model Performance: ROC curve



- **Area Under the curve (AUC):** Value closer to one shows better performance.

Validation

When an algorithm has hyper-parameters, we need a way to select values that results in the best outcome. This process is called

Validation is as follow:

- 1 Similar to measuring generalisation error that we separated some data as a test set, we can put aside some data to compare the model's performance for tunable parameters.
- 2 We then select the parameters that have the best performance on the validation set.
- 3 After that, re-train the model with all the data (train and validation), if possible.

Validation

Different strategies for separating the data:

- 1 **Type 1:** Randomly selects a subset. Make sure the distribution of labels is the same in the subset.
- 2 **Type 2:** Divide the data to k disjoint subsets. Train on $k - 1$ subsets, validate on the last and repeat it. The performance is the average of the k validations.
- 3 **Type 3:** The extreme case is setting one point aside and training on $n - 1$ points.

Validation

Schematically for $k = 4$, type 2 is depicted below.

Train	Train	Train	Validation
Train	Train	Validation	Train
Train	Validation	Train	Train
Validation	Train	Train	Train

References - Books

- ① *"Element of Statistical Learning"* - T. Hastie, R. Tibshirani and J. Friedman.
- ② *"Pattern recognition and machine learning"* - Christopher M. Bishop.
- ③ *"Machine Learning: a Probabilistic Perspective"* - Kevin Murphy.
- ④ *"Reinforcement Learning: An Introduction"* - R. A. Sutton.
- ⑤ *"Deep Learning"* - Ian Goodfellow.

References - Practical Books

- ① *"Deep Learning with Python"* - Francois Chollet.
- ② *"Hands-On Machine Learning with Scikit-Learn and TensorFlow"* - A. Geron.
- ③ *"Python Data Science Handbook"* - Jacob T. Vanderplas .

References

websites

- ① Internet. e.g. <https://paperswithcode.com/>
- ② Libraries' documentation.
- ③ People on Twitter who curate educational resources.
- ④ List of some free resources
 - ① <https://github.com/dair-ai/ML-Course-Notes>
 - ② <https://github.com/dair-ai/ML-YouTube-Courses>
 - ③ <https://github.com/dair-ai/ML-Notebooks>