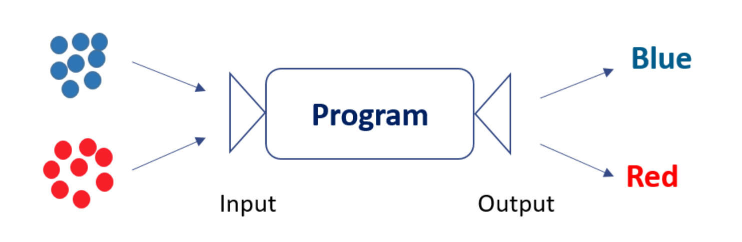
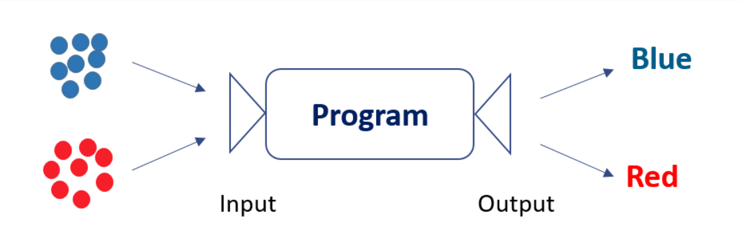
Introduction to Machine Learning

Traditional Computer Science vs. Machine Learning

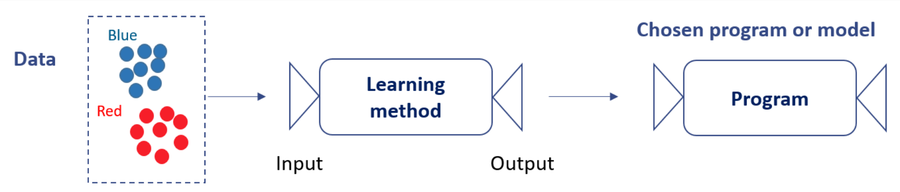


Traditional Way

Traditional Computer Science vs. Machine Learning



Traditional Way



Machine Learning

Machine Learning: Definition

- Term introduced in 1959 by Arthur L. Samuel [1]

Arthur Samuel (1901-1990)

Tom M. Mitchell
Computer Scientist and
Professor @CMU

Machine Learning: Definition

- Term introduced in 1959 by Arthur L. Samuel [1]
- Formal definition (Tom M. Mitchell [2]):
  A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

Arthur Samuel (1901-1990)

Tom M. Mitchell
Computer Scientist and
Professor @CMU

Machine Learning: Definition

- Term introduced in 1959 by Arthur L. Samuel [1]
- Formal definition (Tom M. Mitchell [2]):
  A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.
- In simple words: Algorithms that improve on a task with experience

Arthur Samuel (1901-1990)

Tom M. Mitchell
Computer Scientist and
Professor @CMU

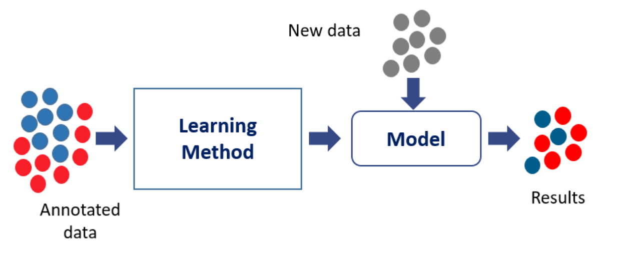
[1] A. Samuel. Some studies in machine learning using the game of Checkers. IBM Journal of Research and Development (1959)

[2] T.M. Mitchel. Machine Learning (1997)

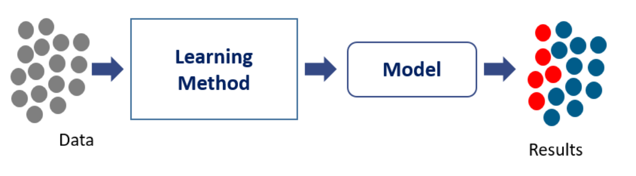
Key Issues in Machine Learning

- What data (E) to use?
- How to represent it?
- Which algorithm should be used to learn?
- How to pick the best model?
- Can we be confident in the results?
- How to model a problem as a Machine Learning problem?

Types of Learning



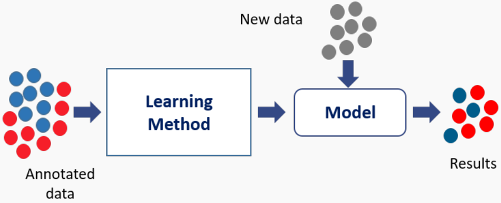
Supervised Learning (80%)



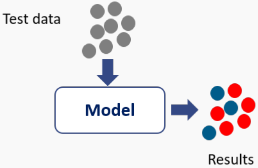
Unsupervised Learning (20%)

Supervised Learning: Procedure

Condensed View of Supervised Learning:



Decompressed View:



Training Phase

Testing Phase

Training Data

The training data comes in input pairs $(\mathbf{x}, y)$, with $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathcal{C}$.

The entire training set is denoted as:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \subseteq \mathbb{R}^D \times \mathcal{C}$$

with

- $\mathbb{R}^D$ - D-dimensional feature space
- $\mathcal{C}$ - label space
- $\mathbf{x}_i$ - input vector of the i^{th} training sample
- y_i - label of the i^{th} training sample
- N - number of training samples

Question: In the previous slide, what is $\mathbf{x}$? and y?

Training Data

The **training set** points $(\mathbf{x}_i, y_i)$ are drawn from an unknown probability distribution $\mathcal{P}(X, Y)$.

Training Data

The **training set** points $(\mathbf{x}_i, y_i)$ are drawn from an unknown probability distribution $\mathcal{P}(X, Y)$.

Goal of Supervised Learning:
Use $\mathcal{D}$ to learn a function h, such that for an **unseen point** $(\mathbf{x}, y) \sim \mathcal{P}$:

$$h(\mathbf{x}) \approx y$$

with high probability

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red ($1/+1$), blue ($0/-1$)
2) Spam filtering (how?)

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification

$\mathcal{C} = \{0, 1\}$ or

$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$

2) Spam filtering (how?)

Multi-class Classification

$\mathcal{C} = \{1, 2, \ldots, K\}$ with $K \geq 2$

Example: Fruit classification from photos (how?)

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$
2) Spam filtering (how?)

Multi-class Classification
$\mathcal{C} = \{1, 2, \ldots, K\}$ with $K \geq 2$

Example: Fruit classification from photos (how?)

Regression
$\mathcal{C} = \mathbb{R}^O$
In this course, $O = 1$

Example: Predict MALIS grades $(O = 1)$
Predict weight and height of a person $(O = 2)$

Setup: Where are we?

Training data

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

Classification/Regression

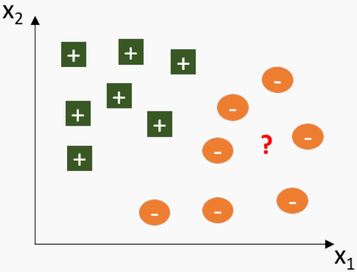
The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

Example:



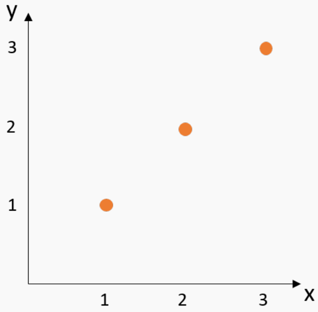
- D=
- $h(\mathbf{x}) = +1$
- Is this a hypothesis?
- Is this a good hypothesis?

The Hypothesis Class

- We have $h \in \mathcal{H}$, where $\mathcal{H}$ denotes the hypothesis class
- **Examples:**
  - Linear Classifiers
  - Decision Trees
  - Neural Networks
  - Support Vector Machines
- First task: Pick a hypothesis class
- Warning: No Free Lunch Theorem

No Free Lunch

- Which hypothesis class $\mathcal{H}$ to choose?
- Every ML algorithm has to make assumptions
- The choice will depend on the data
- $\mathcal{H}$ encodes assumptions about the data and its distribution



$h(2.5) = ?$

No Free Lunch: There is no single perfect choice for all problems

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

- **Idea:** Pick $h \in \mathcal{H}$ making the least mistakes in $\mathcal{D}$ and, preferably, the simplest.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

- **Idea:** Pick $h \in \mathcal{H}$ making the least mistakes in $\mathcal{D}$ and, preferably, the simplest.
- **Measure:** Loss function

The Loss Function

- A loss or risk function $l : \mathbb{R} \to \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

The Loss Function

- A loss or risk function $l : \mathbb{R} \rightarrow \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

- The lower the value of $l(y, h(\mathbf{x}))$ the better the approximation
- $l(y, y) = 0$
- Typically (but not always) $l(y, h(\mathbf{x})) \geq 0$ for all $y, h(\mathbf{x})$

The Loss Function

- A loss or risk function $l : \mathbb{R} \to \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

- The lower the value of $l(y, h(\mathbf{x}))$ the better the approximation
- $l(y, y) = 0$
- Typically (but not always) $l(y, h(\mathbf{x})) \geq 0$ for all $y, h(\mathbf{x})$

Loss	**Expression**	**Task**
0/1 Loss	$l(y, h(\mathbf{x})) = \begin{cases} 1 & \text{if } h(\mathbf{x}) \neq y \\ 0 & \text{otherwise} \end{cases}$	Classification
Quadratic loss	$l(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$	Regression
Absolute loss	$l(y, h(\mathbf{x})) = \lvert y - h(\mathbf{x}) \rvert$	Regression

Table 2: Common loss functions

Loss Minimization

- Using the training data $\mathcal{D}$, we can compute the average loss over all the data points

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

Loss Minimization

- Using the training data $\mathcal{D}$, we can compute the average loss over all the data points

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

- Finding the best hypothesis means finding the h that minimizes the loss.
- This can be formalized as

$$h^* = \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

Summary: Supervised Learning Formalization

Back to the Definition

hypothesis h

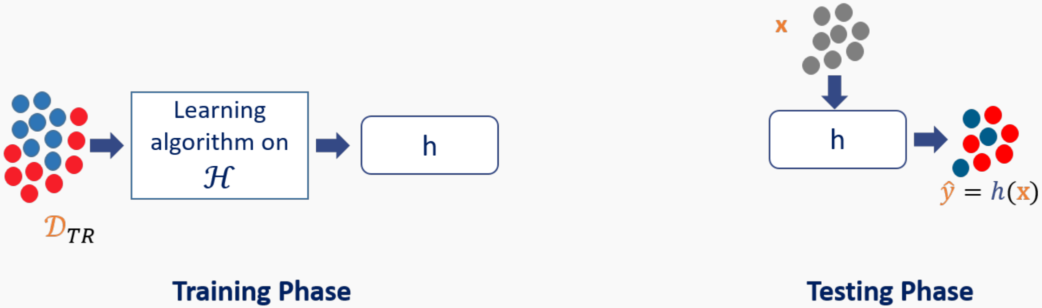
Training data

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

loss

Classification/Regression

Back to the Supervised Learning Process



Generalization

Suppose the following hypothesis:

$$h(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists\, (\mathbf{x}_i, y_i) \in \mathcal{D} \text{ s.t.} \mathbf{x} = \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Generalization

Suppose the following hypothesis:

$$h(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists\,(\mathbf{x}_i, y_i) \in \mathcal{D} \text{ s.t.} \mathbf{x} = \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Questions:

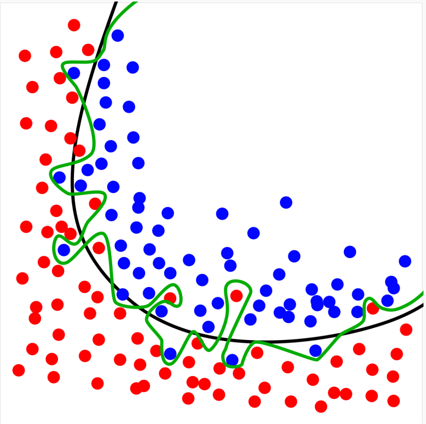
- What is the value of the loss $\mathcal{L}$? Pick the loss you prefer.
- When new samples arrive $\mathbf{x} \notin \mathcal{D}$, how will $h(\cdot)$ perform?

When $h(\cdot)$ has a very low loss, but it does not perform well in unseen data, we say there is **overfitting** causing that our model does not **generalize** well.

Overfitting

Reminder...

- **Overfitting** occurs when a model fits the data too well
- It is associated to models of high complexity
- It will lead to failure to generalize
  Training error $<$ Testing error

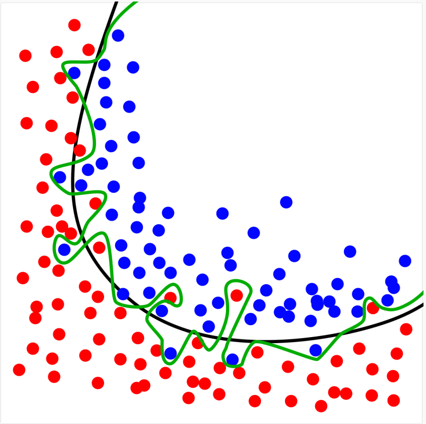


Overfitting

Reminder...

- **Overfitting** occurs when a model fits the data too well
- It is associated to models of high complexity
- It will lead to failure to generalize
  Training error $<$ Testing error

- **Underfitting** occurs when a model cannot adequately capture the underlying structure of the data



Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

However, the loss over the training set does not give us information about the generalization capabilities of the trained model.

Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

However, the loss over the training set does not give us information about the generalization capabilities of the trained model.

Generalization loss:

$$\epsilon = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{P}}[l(y, h(\mathbf{x}))]$$

We can resort to data splitting to obtain an estimate of the generalization loss.

Train/Test Splits

- We split $\mathcal{D}$ into three sets:
  - Training set $\mathcal{D}_{TR}$ - Used to learn h
  - Validation set $\mathcal{D}_{VAL}$ - To check for overfitting
  - Test set $\mathcal{D}_{TEST}$ - Used to evaluate the chosen h and have an estimate of the **generalization error** or loss

- Typical splits are 70/10/20, 80/10/10, 60/20/20.
- If the samples are drawn i.i.d. from the same distribution P, then the testing loss is an unbiased estimator of the true generalization loss.

Train/Test Splits

- It is important to split the data properly to simulate a real life scenario and to avoid **data leakage**.

- How to split?
  - **By time:** if the data is collected temporally, the split needs to be done in time. Example: First 70% point will be for training, next 10% for validation, last 20% for test.
  - **Uniformly at random** if the data is independent and identically distributed

Validation

Generalization and Model Selection

- **Generalization:** Ability of a model to perform well on unseen data

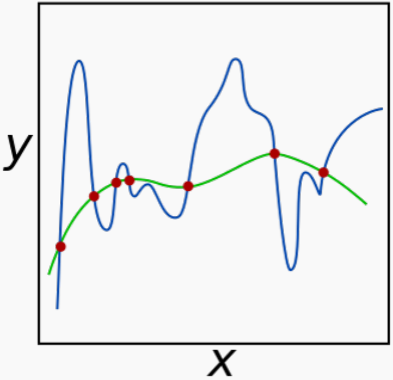
$$\epsilon = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}}[l(y, h(\mathbf{x}))]$$

Generalization loss

- **Model Selection:** Task of selecting a model from a set of candidate models given the data

Model Selection

- For a set of candidate models we choose that one with the smallest test error
- **Reminder:** We prefer simpler models
- Therefore, we might choose:
  - Slightly higher validation errors
  - Simpler models



Validation

1. Split $\mathcal{D}$ into $\mathcal{D}_{TR}, \mathcal{D}_{VAL}$ and $\mathcal{D}_{TEST}$
2. Train candidate models using $\mathcal{D}_{TR}$, e.g. different λ for regularization, network hyper-parameters
3. Use $\mathcal{D}_{VAL}$ to evaluate the candidate models
4. Pick the best
5. Retrain the best using $\mathcal{D}_{TR} + \mathcal{D}_{VAL}$
6. Test the generalization capabilities using $\mathcal{D}_{TEST}$

Validation

1. Split $\mathcal{D}$ into $\mathcal{D}_{TR}, \mathcal{D}_{VAL}$ and $\mathcal{D}_{TEST}$
2. Train candidate models using $\mathcal{D}_{TR}$, e.g. different λ for regularization, network hyper-parameters
3. Use $\mathcal{D}_{VAL}$ to evaluate the candidate models
4. Pick the best
5. Retrain the best using $\mathcal{D}_{TR} + \mathcal{D}_{VAL}$
6. Test the generalization capabilities using $\mathcal{D}_{TEST}$

Drawback

- Easy when there is a very large amount of data
- Was the split the good one?

Cross-Validation

Better known as K-fold cross-validation

Algorithm

1. Split the data into $\mathcal{D}_{TR}, \mathcal{D}_{TEST}$
2. Split $\mathcal{D}_{TR}$ into K-folds
3. For each fold $k \in \{1, \ldots, K\}$, a candidate model is trained in all but the k^{th} fold
4. Test on the k^{th} fold
5. Average the error across folds
6. Use the resulting average error of each candidate model to select one
7. Retrain the chosen one using $\mathcal{D}_{TR}$
8. Test the generalization capabilities using $\mathcal{D}_{TEST}$

Cross-Validation

Better known as K-fold cross-validation

Algorithm

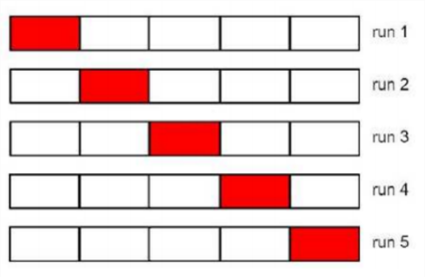
1. Split the data into $\mathcal{D}_{TR}, \mathcal{D}_{TEST}$
2. Split $\mathcal{D}_{TR}$ into K-folds
3. For each fold $k \in \{1, \ldots, K\}$, a candidate model is trained in all but the k^{th} fold
4. Test on the k^{th} fold
5. Average the error across folds
6. Use the resulting average error of each candidate model to select one
7. Retrain the chosen one using $\mathcal{D}_{TR}$
8. Test the generalization capabilities using $\mathcal{D}_{TEST}$

Note

If $K = N$, it is denoted leave-one-out CV (LOOCV)

K-fold Cross-validation

- CV gives an idea of the variability of the test error
- It can assess stability of the method by looking at the models parameter obtained for each fold
- A common value for K is 5



Using CV Properly

- Checking generalization and doing model selection should be two different tasks
- **Model selection:** Estimates the performance of different models in order to choose the best one (validation set via CV)
- **Model assessment:** Having chosen a final model, estimates its prediction error (generalization) on new data (test set)

References

Further Reading and Useful Material

Source	Notes
The Elements of Statistical Learning	Ch 3, 4, 7
The Elements of Statistical Learning	Sec. 11.5 - Training of Neural Networks
Sci-kit Learn	Model Selection and Evaluation
Selection bias in the reported performances of AD classification pipelines	(link)

Any questions?