Machine Learning

Lecture 1: Introduction to ML, Statistical learning: concepts, Bayes classifier, LDA, QDA

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

BIG DATA EXPERIENCE CENTER

Teaching Team

Aj. Yai

Aj. Por

Lecturers

Aj. Joe

P’ Noina

TA

Course Learning Outcome

CLO1: Demonstrate mastery in concepts and details of supervised learning algorithms.

CLO2: Demonstrate mastery in concepts and details of unsupervised and reinforcement learning algorithms.

CLO3: Implement machine learning models and workflows.

Grading

Midterm Exam	30%
Final Exam	35%
Homework and assignments	35%

A: ≥ 85, B+: [84,80], B: [79,75], C+: [74,65],
C: [64,55], D+: [54,50], D: [49,45], F: < 45

The instructor reserves the right to change the grading policy as deemed appropriate.

References

- G. James, et.al., Introduction to Statistical Learning with Application with Python, 2023
- A. Geron, Hands-On Machine Learning with Scikit-Learn & Tensorflow, O'Reilly, 2019

Class Policies

- Assignments are due in one week before class in LEB2.
- Late submissions are only accepted under reasonable excuses and explicit permission from the instructor. No submission is accepted after the solution has been posted.
- Posted solutions will be brief and does not show routine works. You should attempt to work out detailed solutions on your own.
- Academic integrity is strictly enforced.

Class Schedule

Date	Topic
Aug 7	Introduction to ML, Statistical learning: concepts, Bayes classifier, LDA, QDA
Aug 14	Training models: Direct (OLS) and iterative (gradient descent) approaches
Aug 21	Support vector machines: Linear and nonlinear SVM
Aug 28	Regression, Constraints, Generalized linear models
Sep 4	Tree-based models: Decision tree and Regression tree
Sep 9 - 13	1st University Exam Period - No class
Sep 18	Ensemble models
Sep 25	MLOps
Oct 2	Neural networks and deep learning
Oct 9	Convolutional neural networks
Oct 16	Dimensionality reduction
Oct 21 - 29	2nd University Exam Period - No class
Nov 6	Clustering (1): Dissimilarity measures, K-means clustering, Cluster evaluation
Nov 13	Clustering (2): Gaussian mixture model, Hierarchical clustering, Density-based clustering
Nov 20	Anomaly detection
Nov 27	Reinforcement learning
Dec 2 - 13	3rd University Exam Period - No class

Topics

- Introduction to ML
- Statistical learning concepts
- Bayes' classifier
- Discriminant analysis

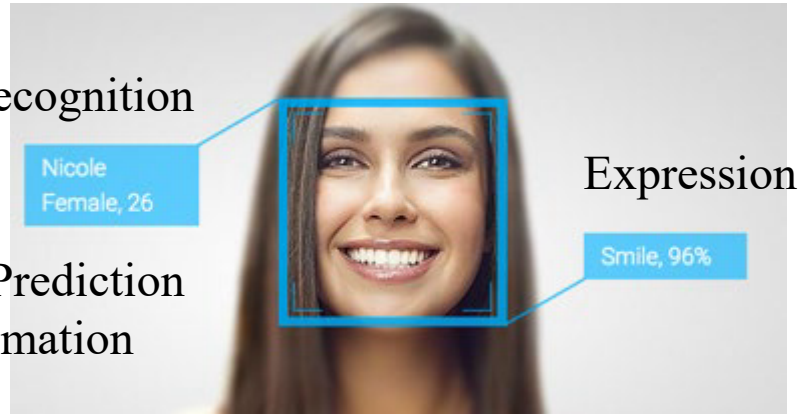
Introduction to ML
Statistical Learning Concepts

Section 1

Consumer ML

Face Detection

Face Recognition



Expression Recognition

Gender Prediction
Age Estimation

Voice Interface

Smart Home

What exactly is machine learning?

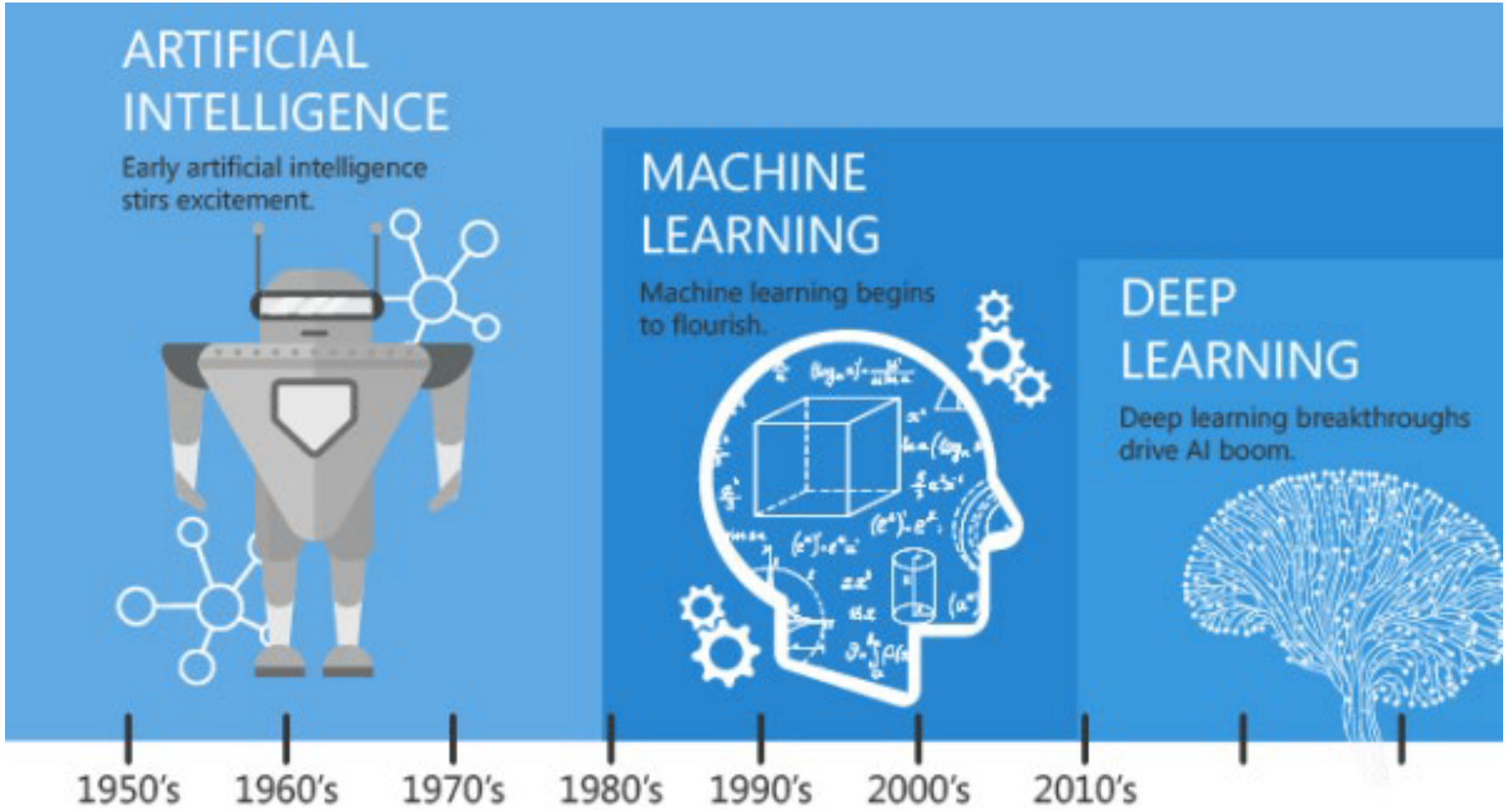
- Machine learning (ML) is a method of data analysis that automates analytical model building.

- Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

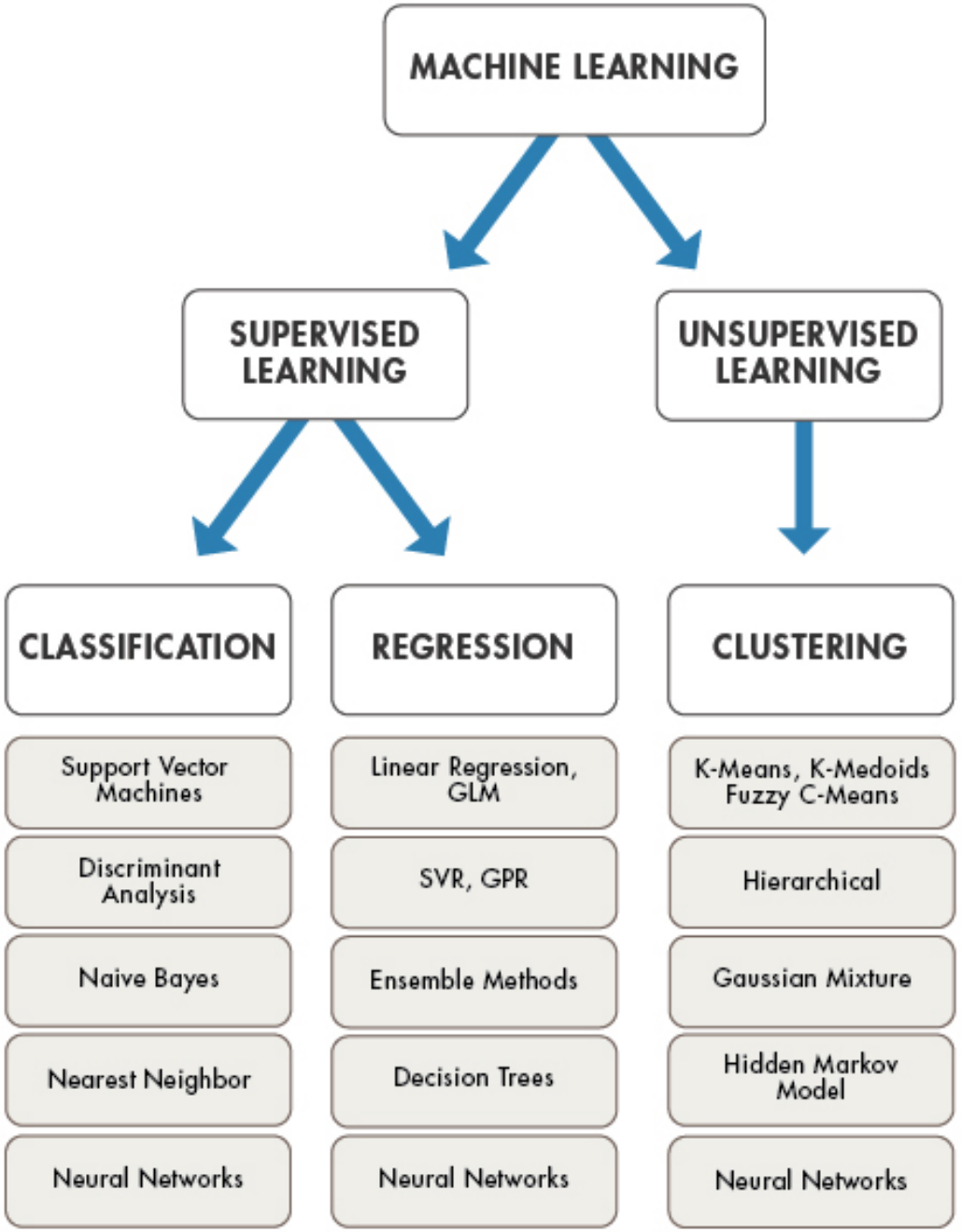
Why machine learning matter

- Automatic: Train it once and it can be run automatically
- Fast: With big data, work faster than human
- Accurate: Can predict groups more accurately than manual methods
- Scale: Able to handle large data

Timeline


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Types of ML


MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Neural Networks
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Hidden Markov Model
Neural Networks

Style of Learning

• Data has **known labels** or output	• Labels or output unknown • Focus on **finding patterns and gaining insight** from the data	• Labels or output known for **a subset of** data • A blend of supervised and unsupervised learning	• Focus on **making decisions** based on previous experience • Policy-making with feedback
• Insurance underwriting • Fraud detection	• Customer clustering • Association rule mining	• Medical predictions (where tests and expert diagnoses are expensive, and only part of the population	• Game AI • Complex decision problems • Reward systems

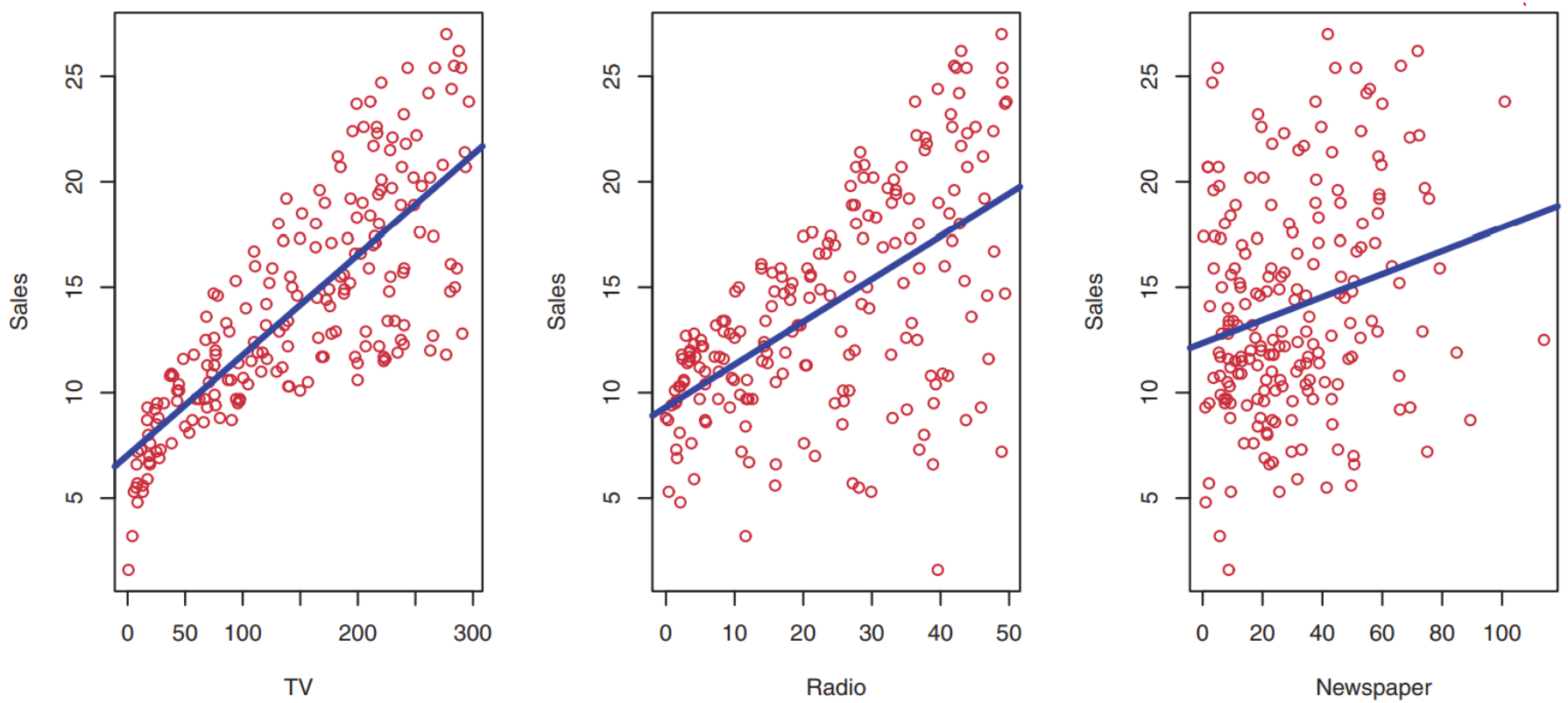


FIGURE 2.1. *The* `Advertising` *data set. The plot displays* `sales`*, in thousands of units, as a function of* `TV`*,* `radio`*, and* `newspaper` *budgets, in thousands of dollars, for* 200 *different markets. In each plot we show the simple least squares fit of* `sales` *to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict* `sales` *using* `TV`*,* `radio`*, and* `newspaper`*, respectively.*

Representing relationship

- Suppose that we observe a quantitative response Y and p different predictors, X_1, X_2,...,X_p.
- Assume that there is some relationship between Y and X = (X_1, X_2,...,X_p),

$$Y = f(X) + \epsilon.$$

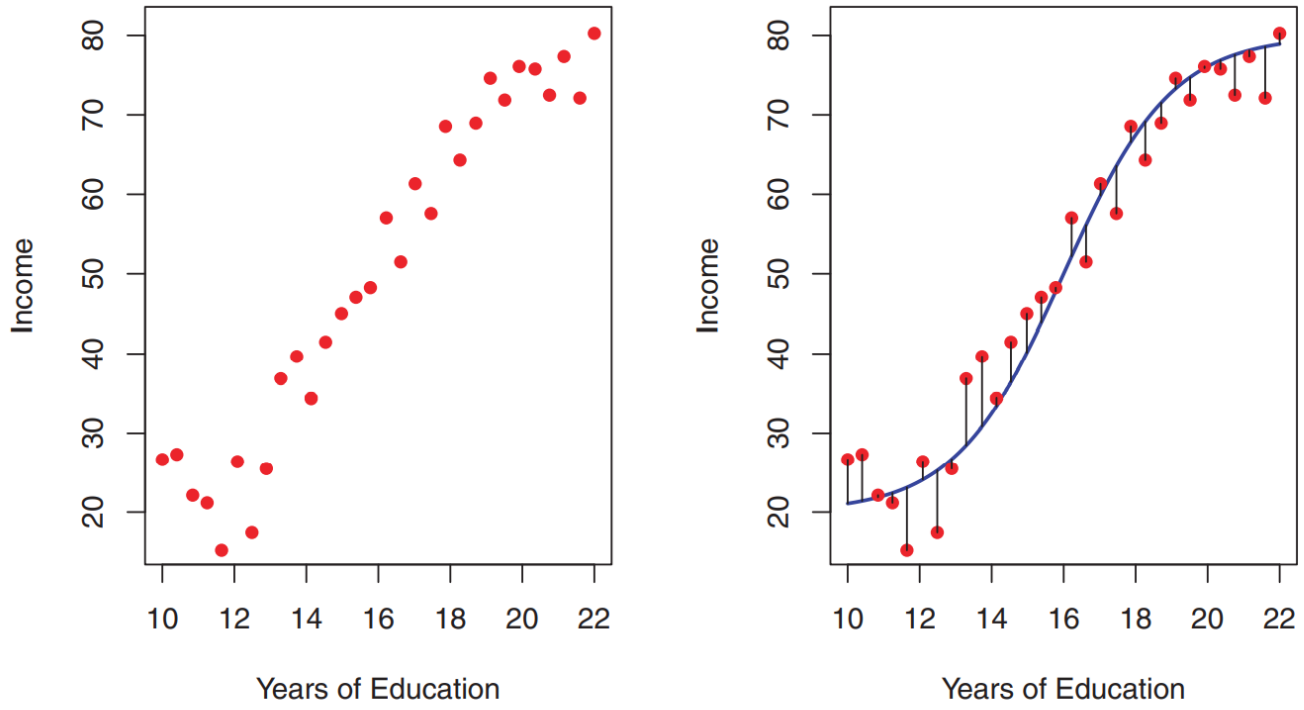


FIGURE 2.2. *The* `Income` *data set.* Left: *The red dots are the observed values of* `income` *(in tens of thousands of dollars) and* `years of education` *for* 30 *individuals.* Right: *The blue curve represents the true underlying relationship between* `income` *and* `years of education`*, which is generally unknown (but is known in this case because the data were simulated). The black lines represent the error associated with each observation. Note that some errors are positive (if an observation lies above the blue curve) and some are negative (if an observation lies below the curve). Overall, these errors have approximately mean zero.*

Why estimate f?

- Prediction

$$\hat{Y} = \hat{f}(X),$$

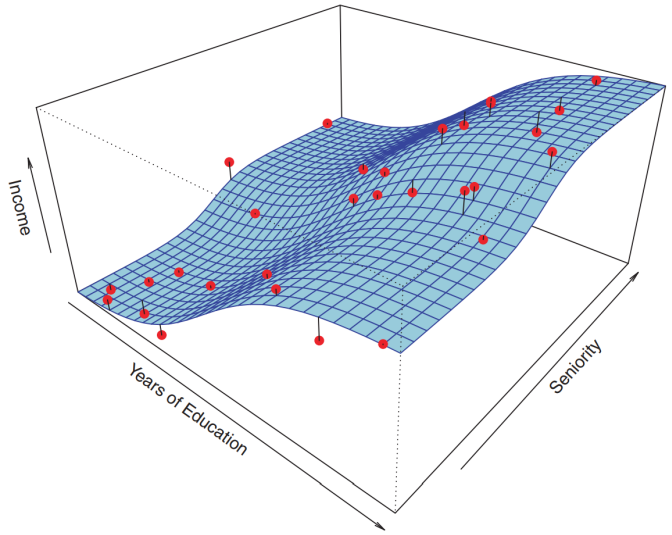


FIGURE 2.3. *The plot displays* income *as a function of* years of education *and* seniority *in the* Income *data set. The blue surface represents the true underlying relationship between* income *and* years of education *and* seniority*, which is known since the data are simulated. The red dots indicate the observed values of these quantities for* 30 *individuals.*

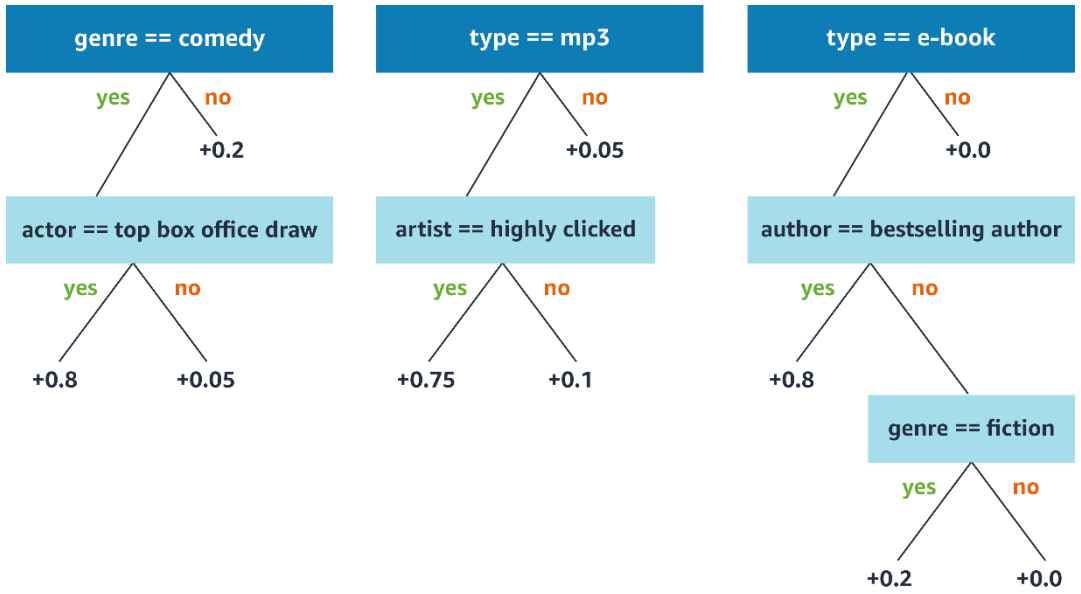
- Inference
  - Which predictors are associated with the response?
  - What is the relationship between the response and each predictor?
  - Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

How do we estimate f?

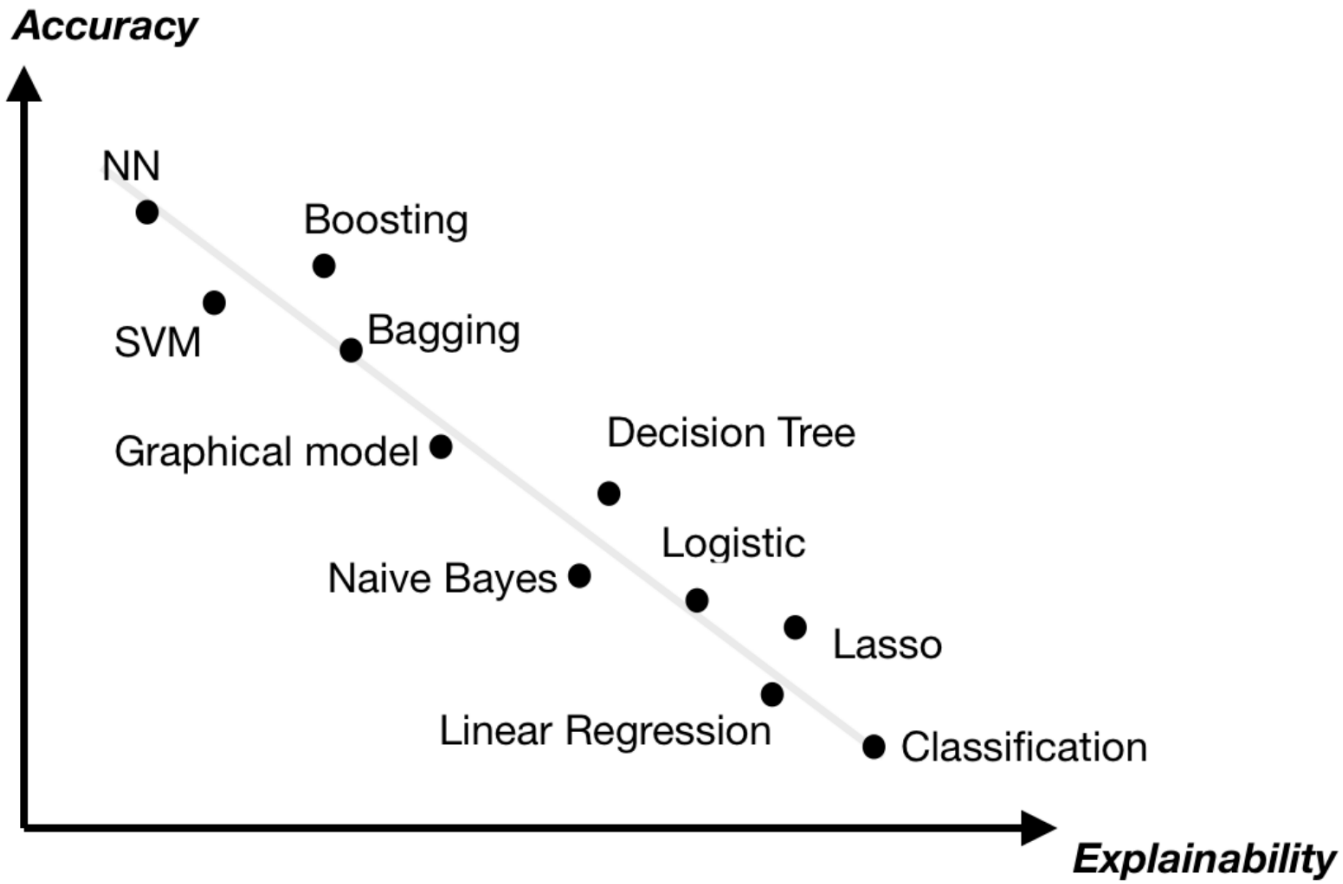
- Linear models

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p.$$

- Non-linear models

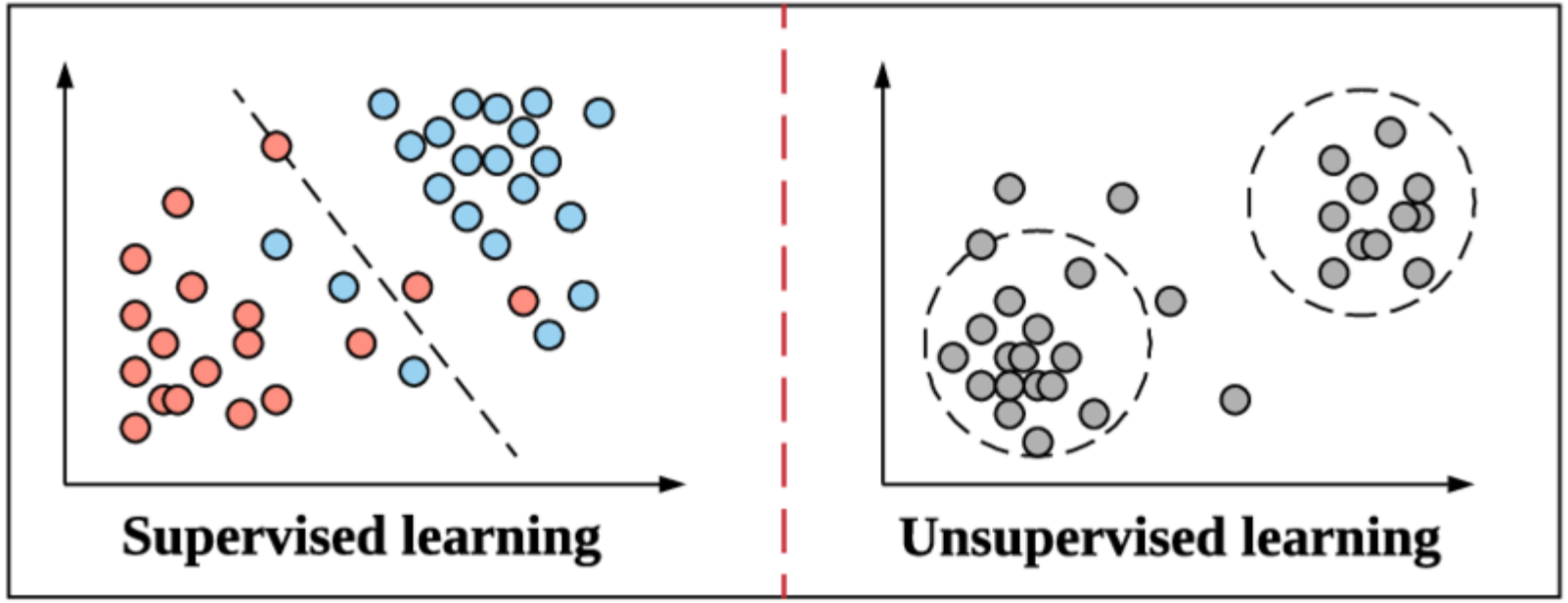


Explainability vs Accuracy



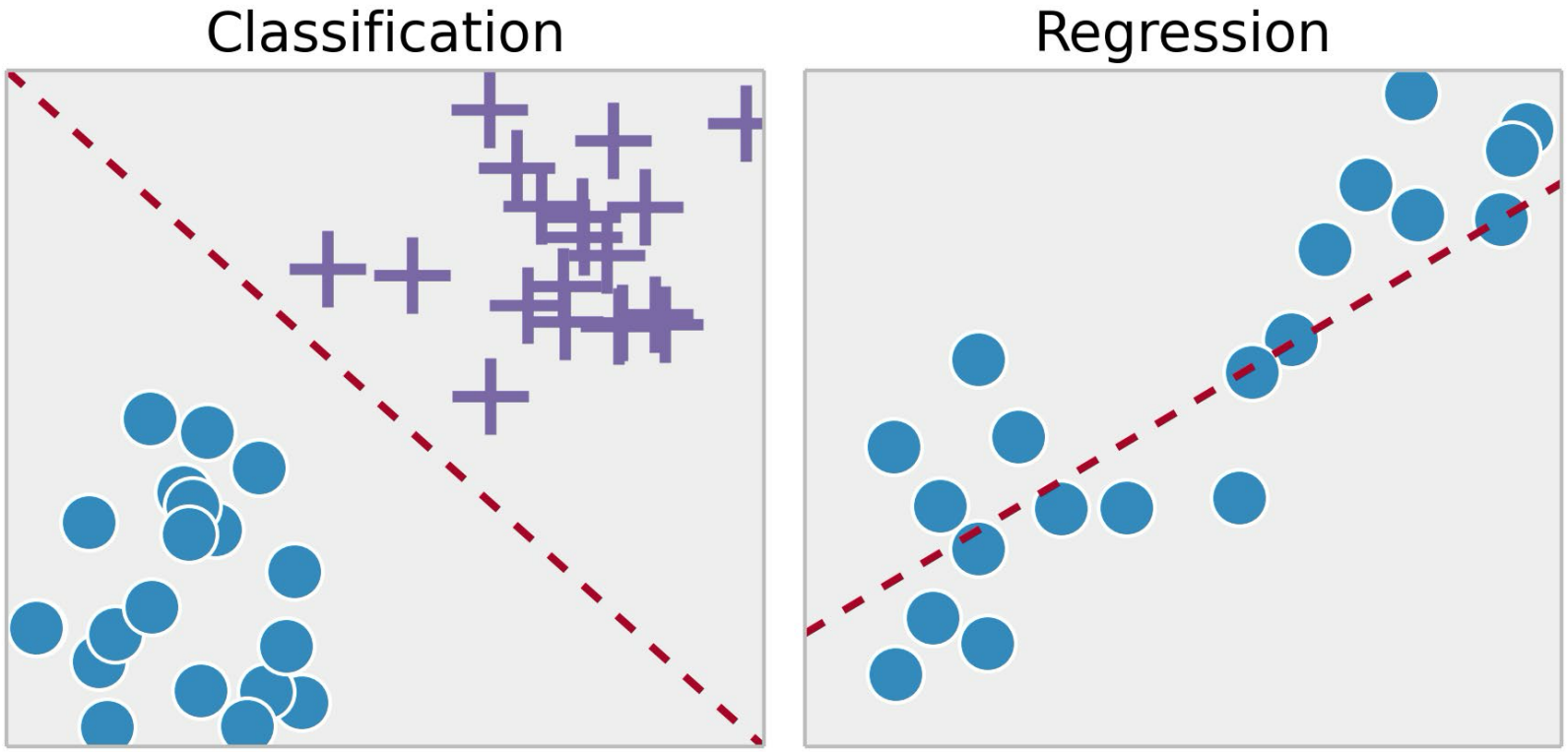
Duval, Alexandre. (2019). Explainable Artificial Intelligence (XAI). 10.13140/RG.2.2.24722.09929.

Supervised vs Unsupervised



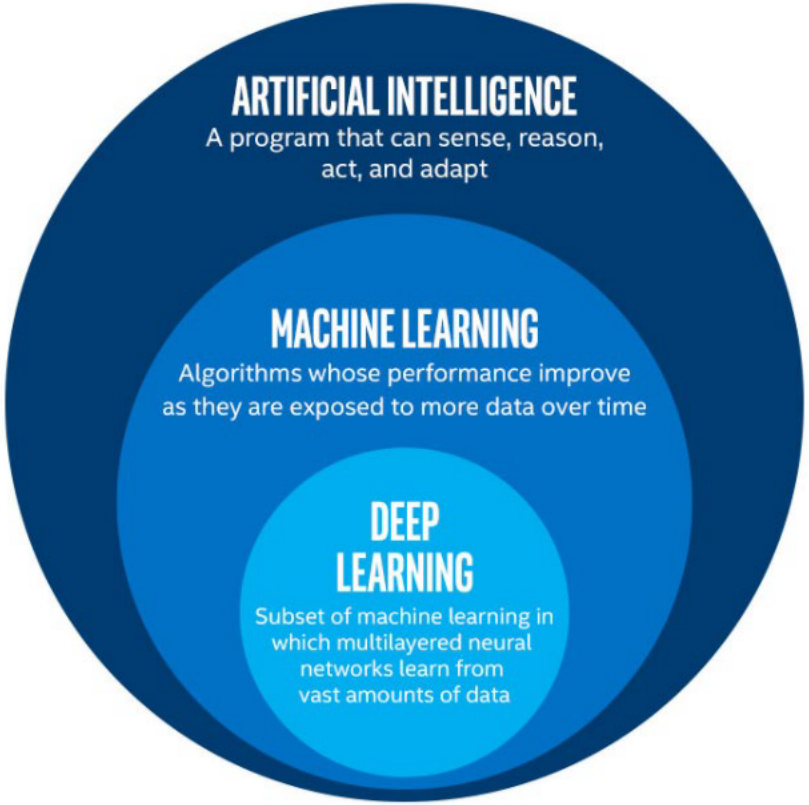
Qian, Bin & Su, Jie & Wen, Zhenyu & Yang, Renyu & Zomaya, Albert & Rana, Omer. (2019). Orchestrating the Development Lifecycle of Machine Learning-Based IoT Applications: A Taxonomy and Survey.

Regression vs Classification



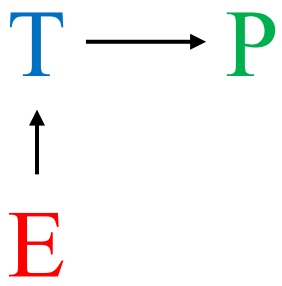
Machine Learning Basics

- Deep learning is a specific kind of machine learning.
- To understand deep learning well, one must have a solid understanding of the basic principles of machine learning.



Learning Algorithm - Definition

"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." (Mitchell, 1997)



Tasks, T

Classification

- In this type of task, the computer program is asked to specify which of *k* categories some input belongs to.
- To solve this task, the learning algorithm is usually asked to produce a function f: $R^n \rightarrow \{1,..., k\}$.
- When $y = f(x)$, the model assigns an input described by vector x to a category identified by numeric code y.

Tasks, T

Regression

- In this type of task, the computer program is asked to predict a numerical value given some input.
- To solve this task, the learning algorithm is asked to output a function f: $R^n \rightarrow R$.
- This type of task is similar to classification, except that the format of output is a value instead of a class.

Tasks, T

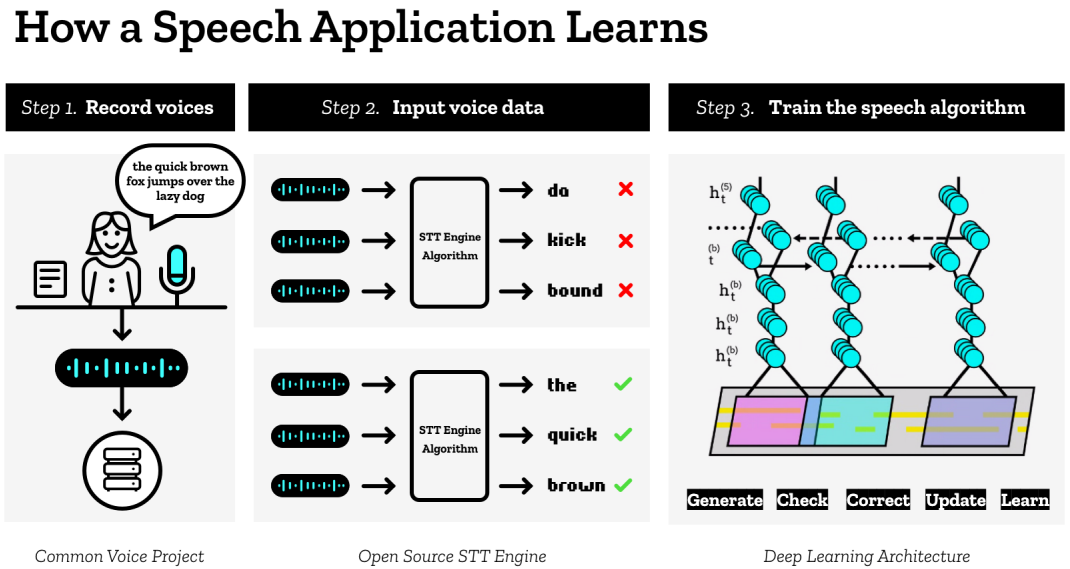
Clustering

- In this type of task, the computer program is asked to divide data into groups.
- To solve this task, the learning algorithm is asked to output a function $f: R^n \rightarrow \{1,\ldots,k\}$, but without the label.
- The goal is usually to divide data into groups based on specific features, e.g. Recency-Frequency-Monetary customer segmentation

Tasks, T

Transcription

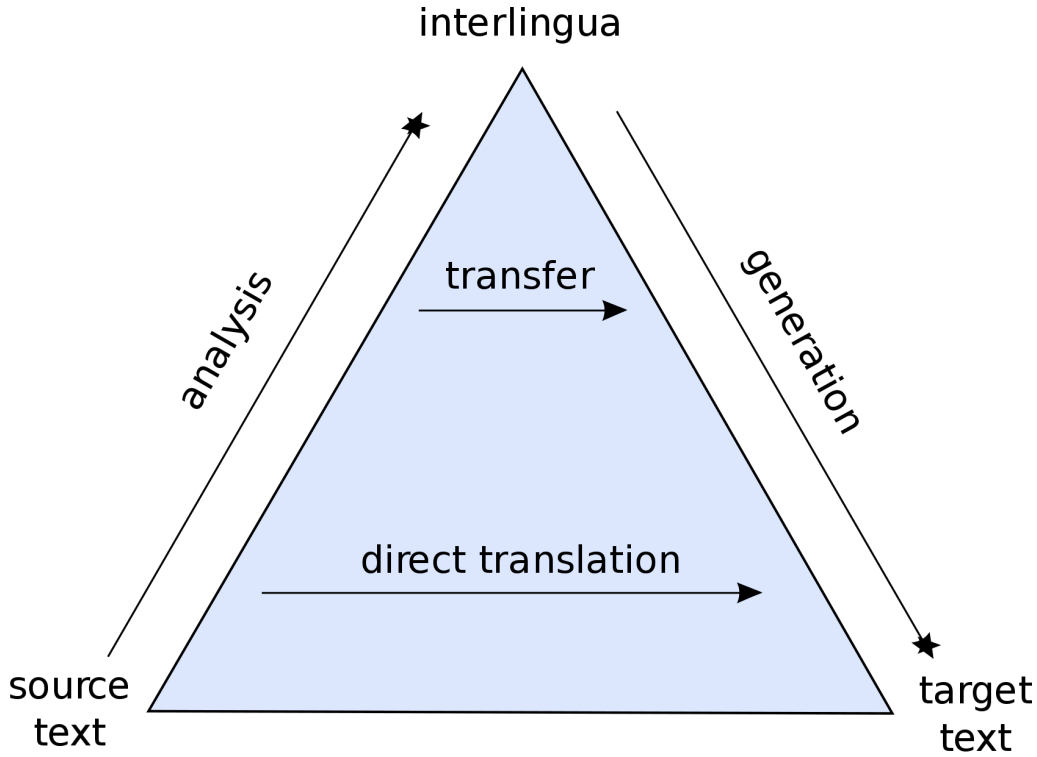
- In this type of task, the machine learning system is asked to observe a relatively unstructured representation of some kind of data and transcribe the information into discrete textual form.



Tasks, T

Machine Translation

- In a machine translation task, the input already consists of a sequence of symbols in some language, and the computer program must convert this into a sequence of symbols in another language.



Tasks, T

Anomaly Detection

- In this type of task, the computer program sifts through a set of events or objects and flags some of them as being unusual or atypical.

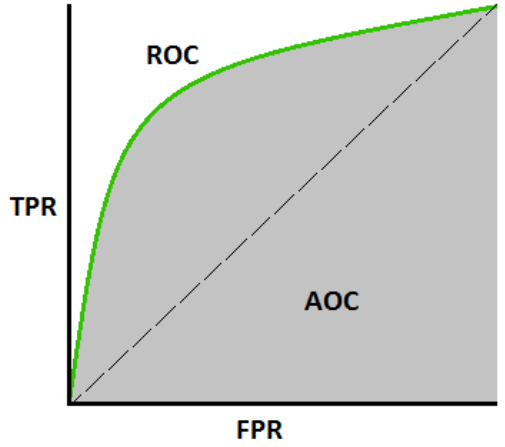
Tasks, T

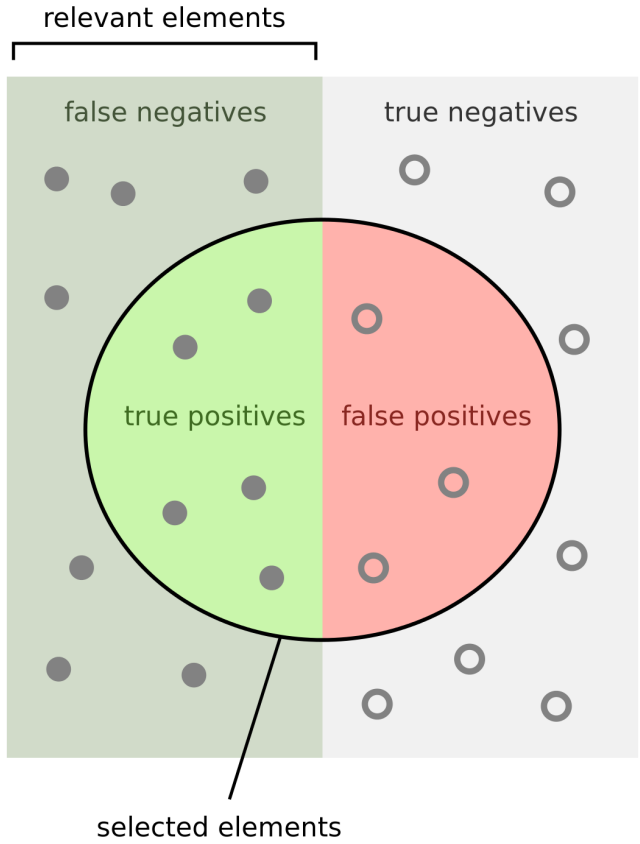
Synthesis and sampling

- In this type of task, the machine learning algorithm is asked to generate new examples that are similar to those in the training data.

Performance, P

- Classification

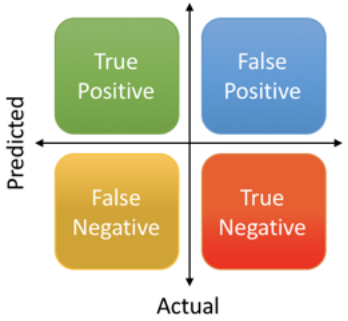




Precision = $\frac{\text{True Positive}}{\text{Actual Results}}$ or $\frac{\text{True Positive}}{\text{True Positive + False Positive}}$

Recall = $\frac{\text{True Positive}}{\text{Predicted Results}}$ or $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$

Accuracy = $\frac{\text{True Positive + True Negative}}{\text{Total}}$



How many selected items are relevant?

Precision = $\frac{\text{green}}{\text{green + red}}$

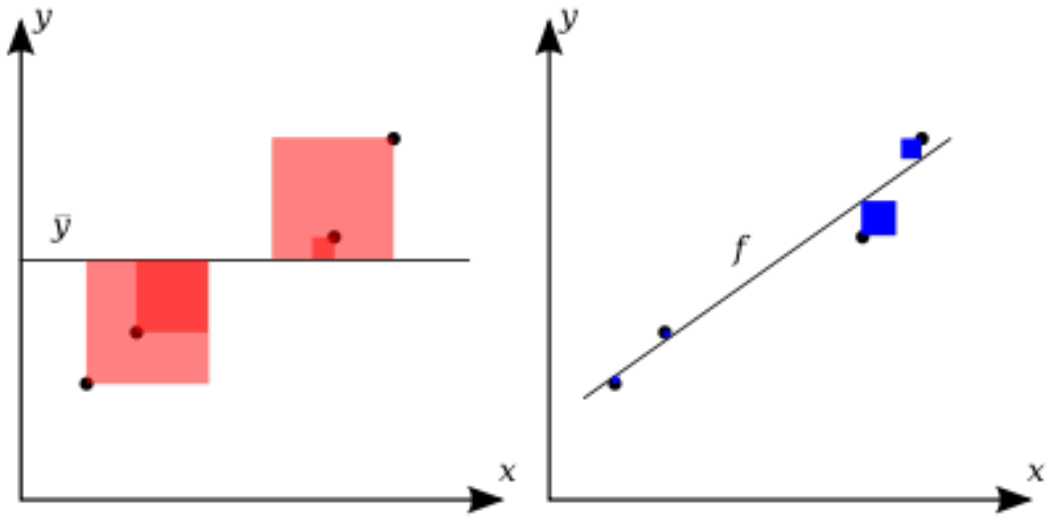
How many relevant items are selected?

Recall = $\frac{\text{green}}{\text{green + yellow}}$

Performance, P

- Regression

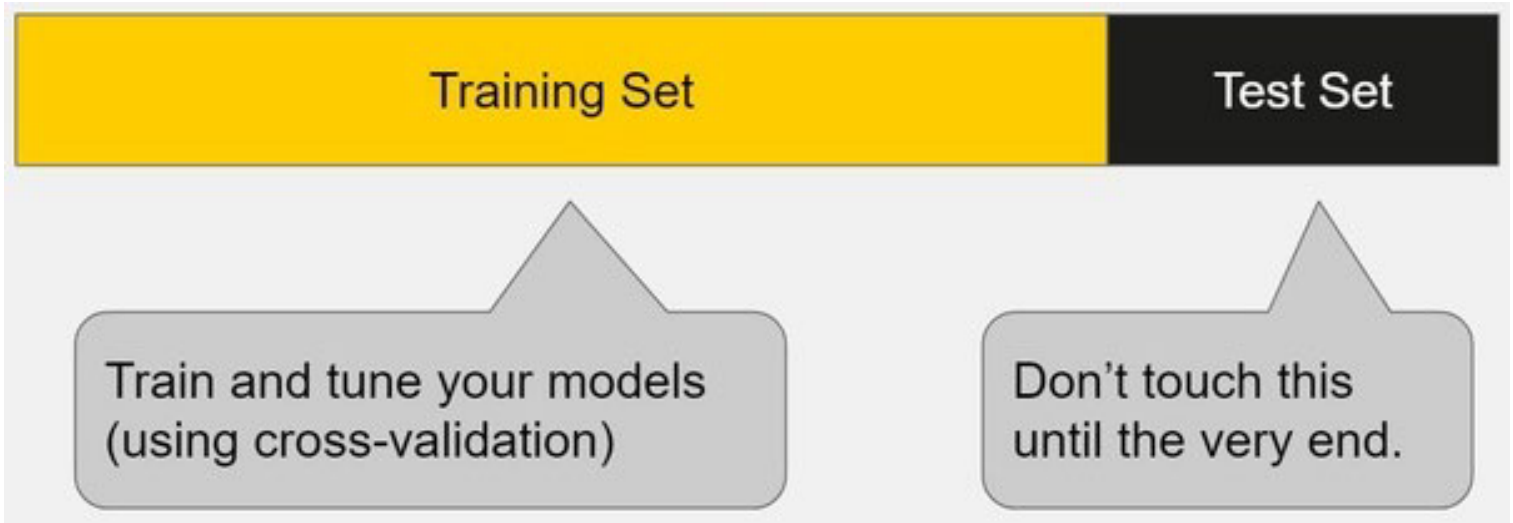
R^2



$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\, x_i - x \,|$$

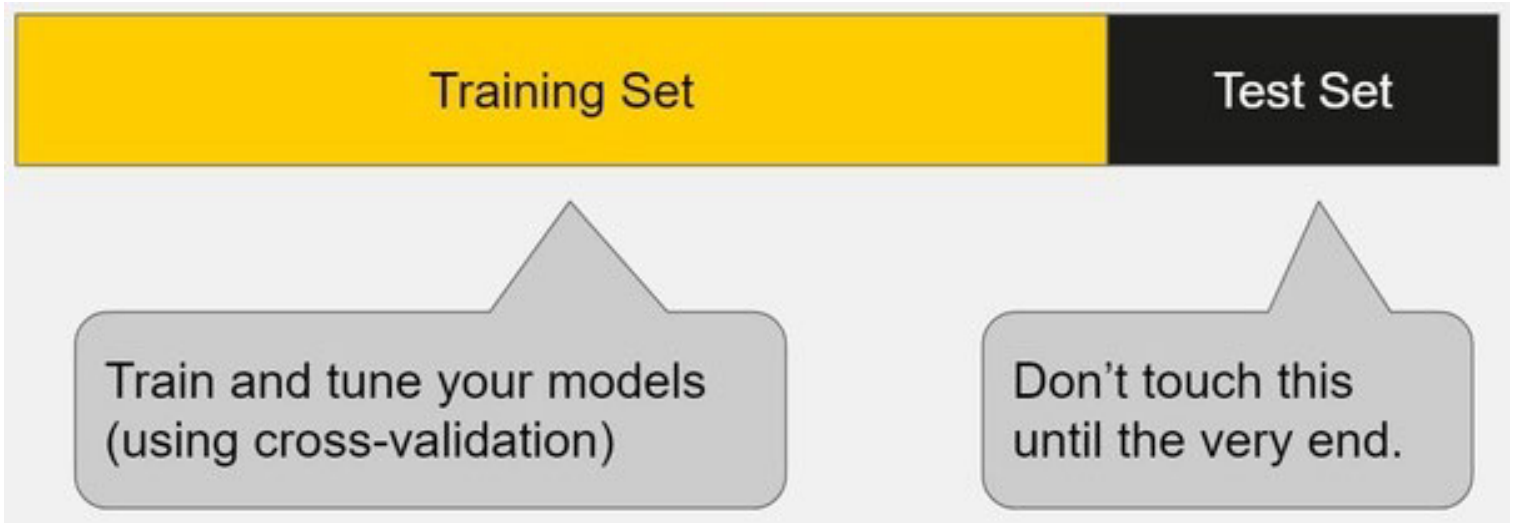
Performance, P

- Train set and Test set

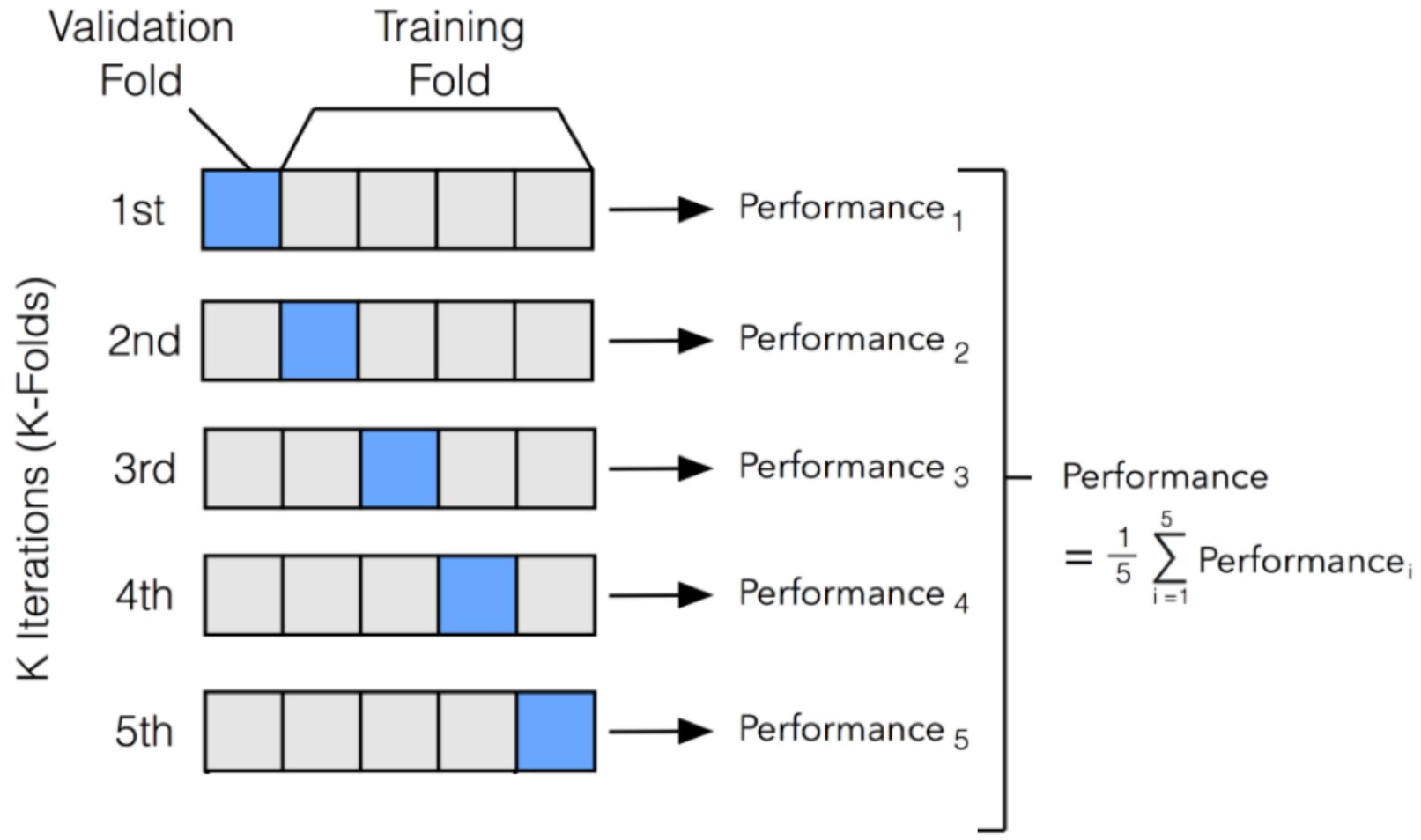


Experience, E

- Train set and Test set



Cross-validation



Example: Linear Regression

- Linear regression solves a regression problem.
- In other words, the goal is to build a system that can take a vector $\mathbf{x} \in \mathrm{R}^n$ as input and predict the value of a scalar $y \in \mathrm{R}$ as its output.
- The output of linear regression is a linear function of the input.
- Let $\hat{y}$ be the value that our model predicts y should take on. We define the output to be

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x},$$

where $\boldsymbol{w} \in \mathrm{R}^n$ is a vector of parameters

Linear regression: Task, T

To predict y from **x** by outputting

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x},$$

Performance measure, P

Mean square error

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{m} \sum_i (\hat{\boldsymbol{y}}^{(\mathrm{test})} - \boldsymbol{y}^{(\mathrm{test})})_i^2.$$

Experience, E

$$\nabla_{\boldsymbol{w}} \mathrm{MSE}_{\mathrm{train}} = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \frac{1}{m} ||\hat{\boldsymbol{y}}^{(\mathrm{train})} - \boldsymbol{y}^{(\mathrm{train})}||_2^2 = 0$$

$$\Rightarrow \frac{1}{m} \nabla_{\boldsymbol{w}} ||\boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})}||_2^2 = 0$$

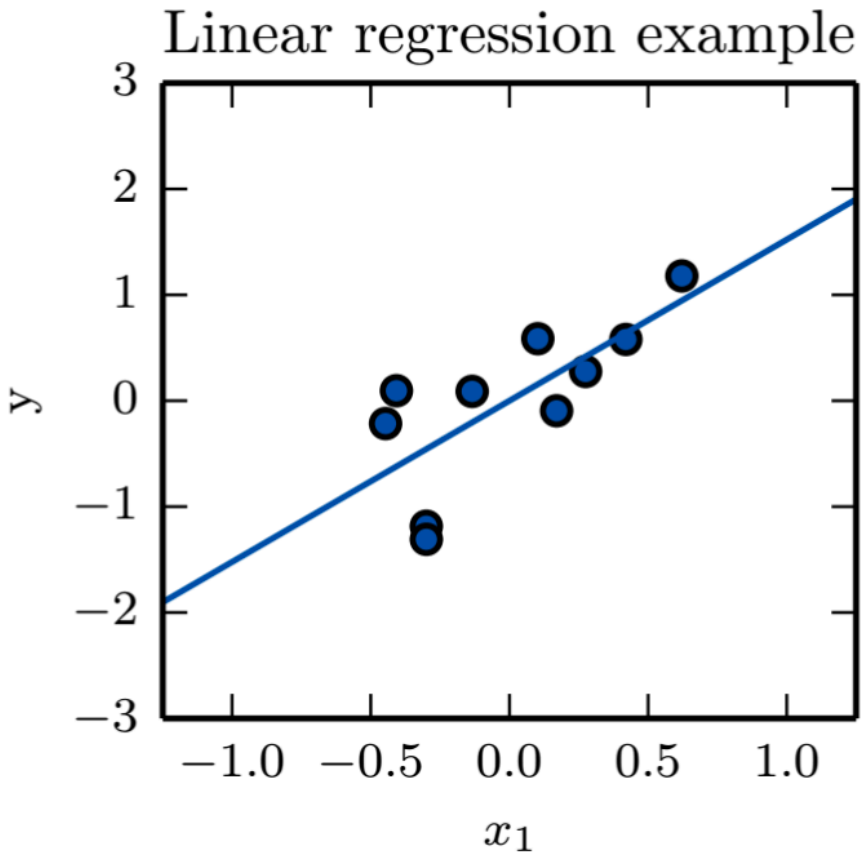
$$\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})} \right)^\top \left(\boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})} \right) = 0$$

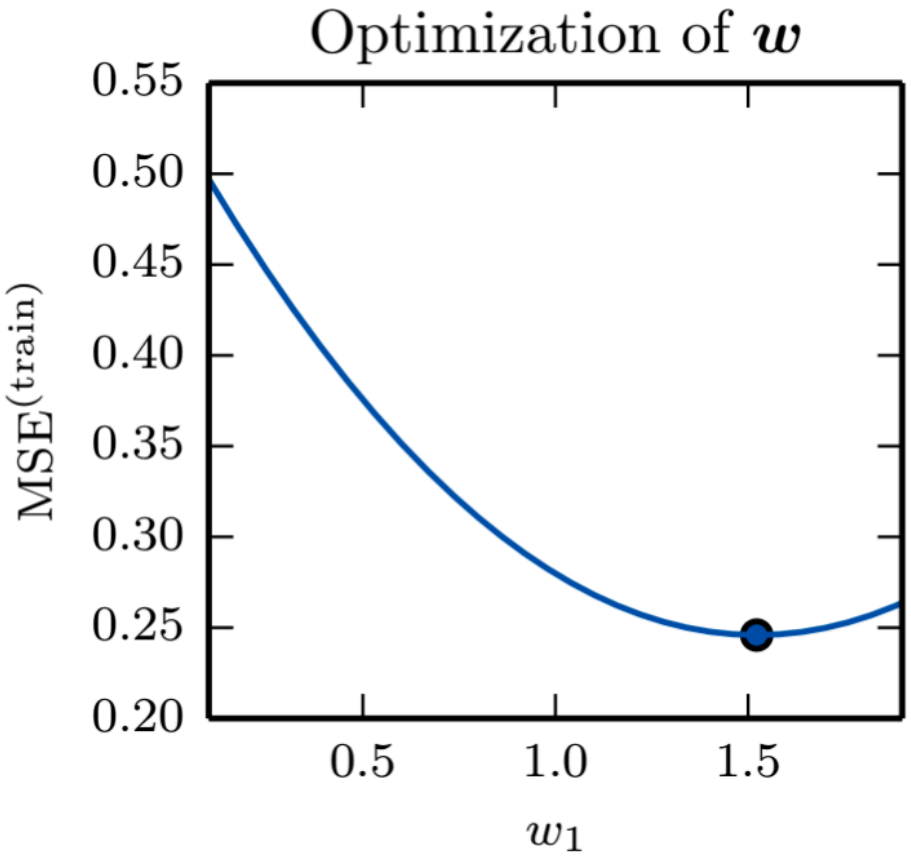
$$\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{w}^\top \boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - 2\boldsymbol{w}^\top \boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{y}^{(\mathrm{train})} + \boldsymbol{y}^{(\mathrm{train})\top} \boldsymbol{y}^{(\mathrm{train})} \right) = 0$$

$$\Rightarrow 2\boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - 2\boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{y}^{(\mathrm{train})} = 0$$

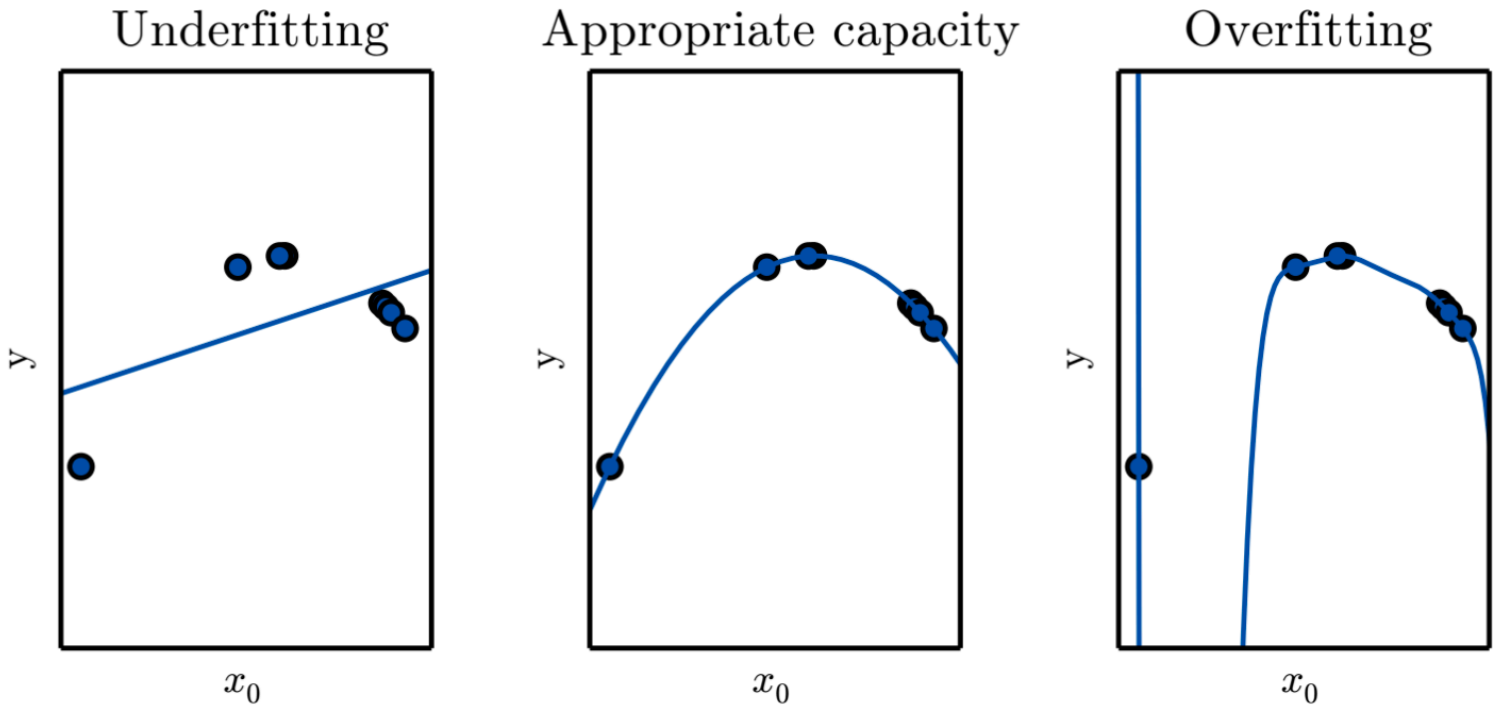
$$\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{X}^{(\mathrm{train})} \right)^{-1} \boldsymbol{X}^{(\mathrm{train})\top} \boldsymbol{y}^{(\mathrm{train})}$$

Linear regression – learning weight





Regression, underfitting and overfitting

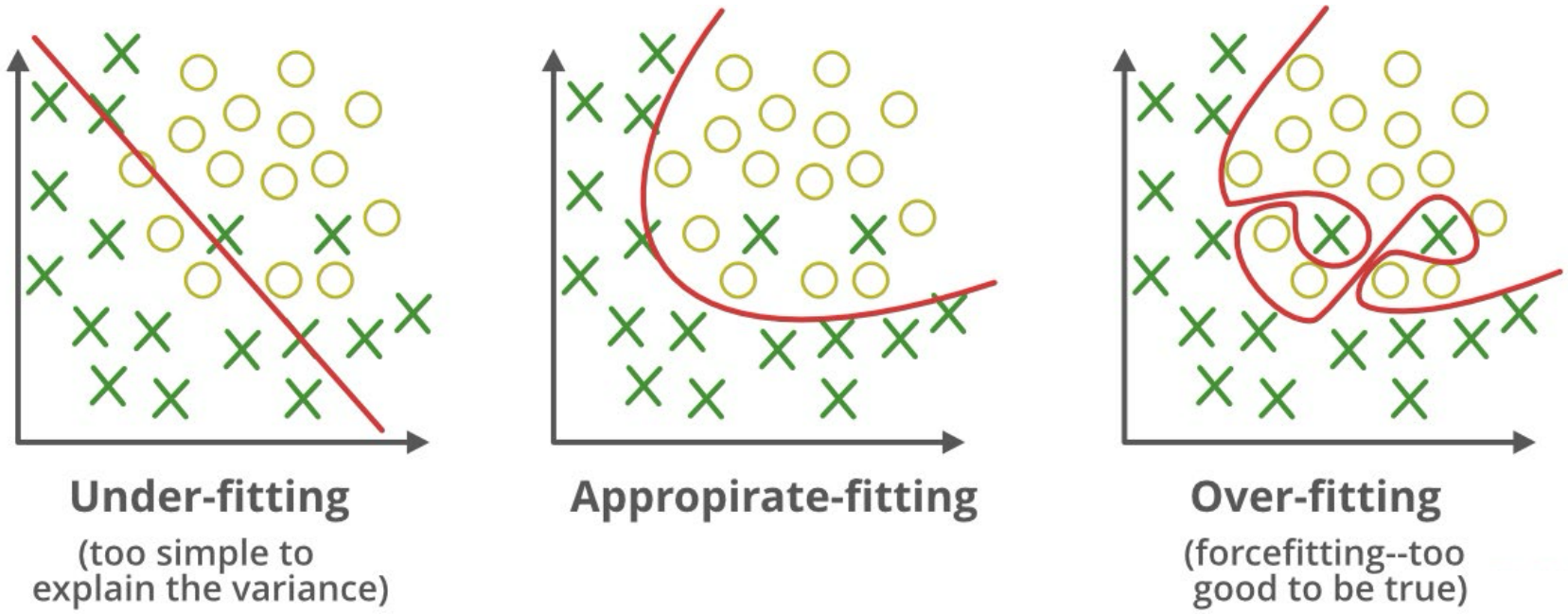


$$\hat{y} = b + wx.$$

$$\hat{y} = b + w_1 x + w_2 x^2.$$

$$\hat{y} = b + \sum_{i=1}^{9} w_i x^i.$$

Classification, underfitting and overfitting



Bayes' Classifier

Section 2

Bayes' Classifier

- assigns each observation to the most likely class, given its predictor values.
- In other words, we should simply assign a test observation with predictor vector x_0 to the class j for which

$$\Pr(Y = j \mid X = x_0)$$

  is the largest
- This is a conditional probability, the probability conditional that Y = j, given the observed predictor vector x_0.

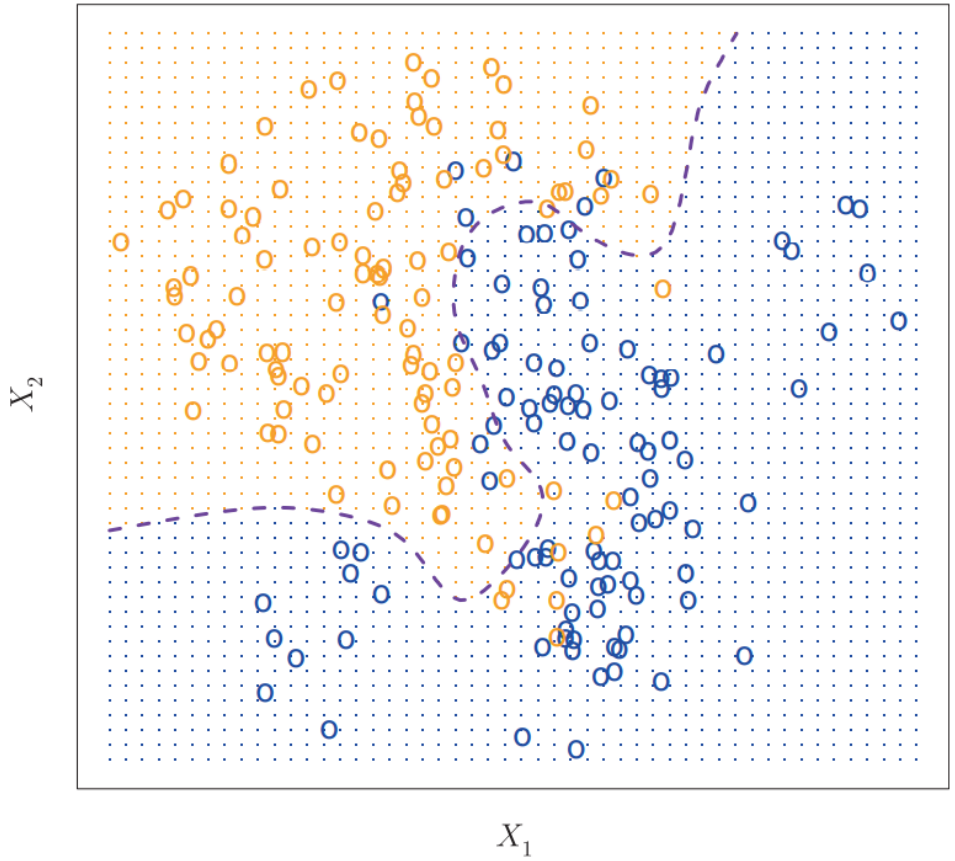
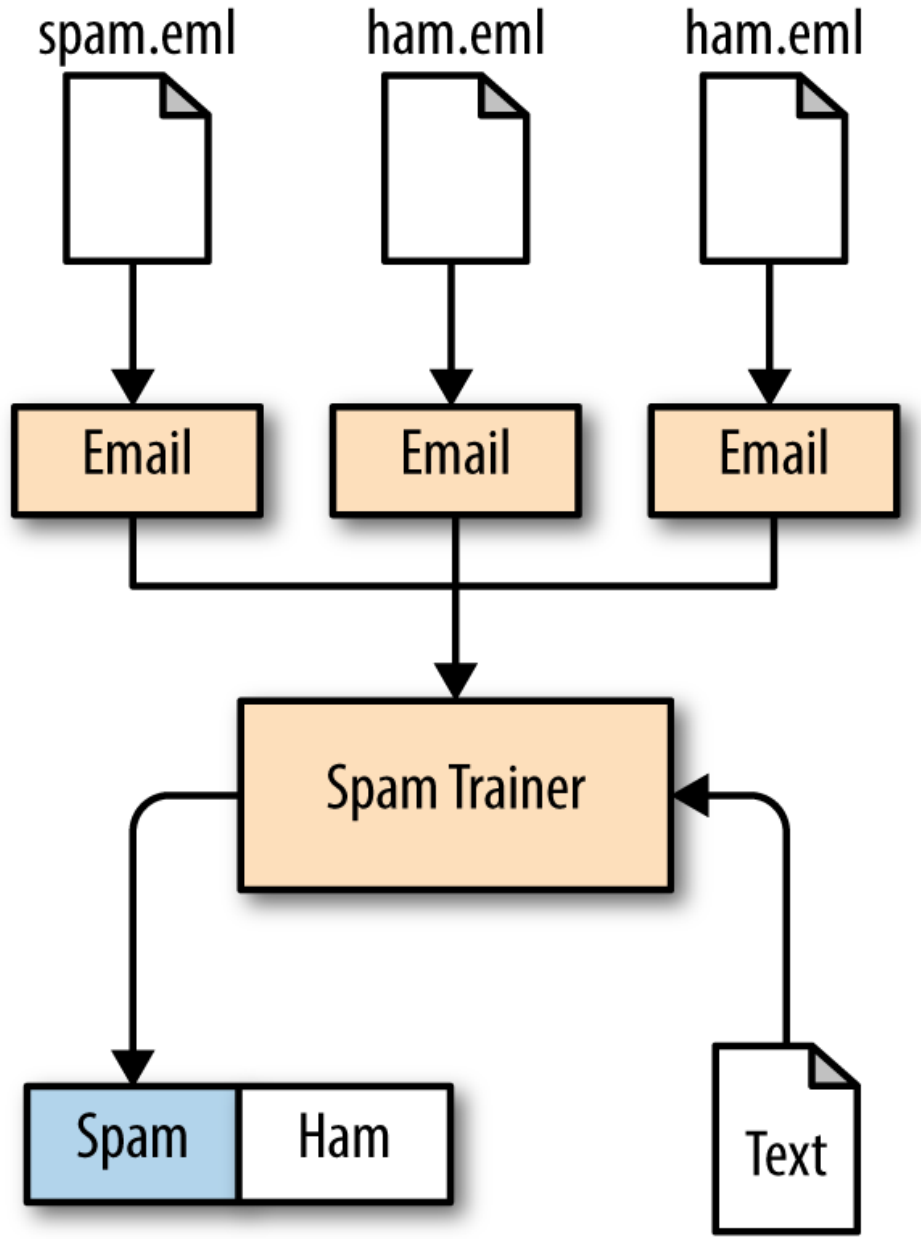


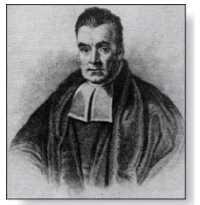
FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*


spam.eml
ham.eml
ham.eml
Email
Email
Email
Spam Trainer
Spam
Ham
Text

Motivation for Bayes' Theorem

- Bayes' theorem allows us to use probability to answer questions such as the following:
  - Given that someone tests positive for having a particular disease, what is the probability that they actually do have the disease?
  - Given that someone tests negative for the disease, what is the probability, that in fact they do have the disease?
- Bayes' theorem has applications to medicine, law, artificial intelligence, engineering, and many diverse other areas.

Bayes' Theorem



Thomas Bayes
(1702-1761)

Bayes' Theorem: Suppose that E and F are events from a sample space S such that $p(E) \neq 0$ and $p(F) \neq 0$. Then:

$$p(F|E) = \frac{p(E|F)p(F)}{p(E|F)p(F) + p(E|\overline{F})p(\overline{F})}$$

$p(E)$

Derivation of Bayes' Theorem

- Recall the definition of the conditional probability $p(E|\mathrm{F})$:

$$p(E|F) = \frac{p(E \cap F)}{p(F)}$$

- From this definition, it follows that:

$$p(E|F) = \frac{p(E \cap F)}{p(F)}, \qquad p(F|E) = \frac{p(E \cap F)}{p(E)}$$

Derivation of Bayes' Theorem

$$p(E|F)p(F) = p(E \cap F),\ p(F|E)p(E) = p(E \cap F)$$

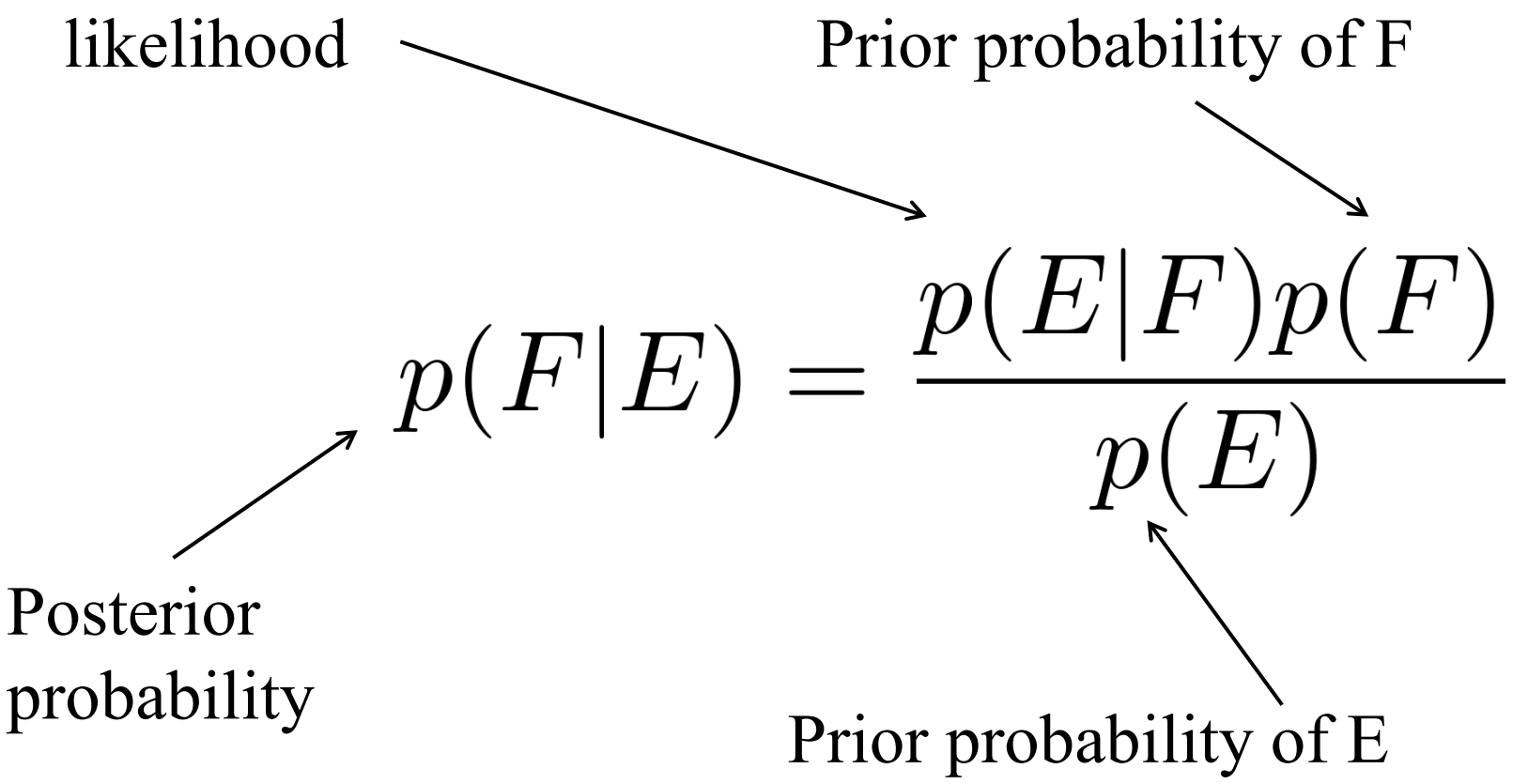
Equating the two formulas for p(E F) shows that

$$p(E|F)p(F) = p(F|E)p(E)$$

Solving for $p(E|F)$ and for $p(F|E)$ tells us that

$$p(E|F) = \frac{p(F|E)p(E)}{p(F)},\ p(F|E) = \frac{p(E|F)p(F)}{p(E)}$$

Bayes' Theorem



Derivation of Bayes' Theorem

$$p(F|E) = \frac{p(E|F)p(F)}{p(E)}$$

Note that $$p(E) = p(E|F)p(F) + p(E|\overline{F})p(\overline{F})$$

since $p(E) = p(E \cap F) + p(E \cap \overline{F})$

$$p(E) = p(E \cap F) + p(E \cap \overline{F}) = p(E|F)p(F) + p(E|\overline{F})p(\overline{F})$$

Hence, $$p(F|E) = \frac{p(E|F)p(F)}{p(E|F)p(F) + p(E|\overline{F})p(\overline{F})}$$

Applying Bayes' Theorem

Example: Suppose that one person in 100,000 has a particular disease. There is a test for the disease that gives a positive result 99% of the time when given to someone with the disease. When given to someone without the disease, 99.5% of the time it gives a negative result. Find

a) the probability that a person who test positive has the disease.

b) the probability that a person who test negative does not have the disease.

- Should someone who tests positive be worried?

$p(D) = 10^{-5}$ $\quad p(+|D) = 0.99$ $\quad p(-|\bar{D}) = 0.995$

$p(\bar{D}) = 0.99999$ $\quad p(+|\bar{D}) = 0.01$ $\quad p(-|D) = 0.005$

Applying Bayes' Theorem

Solution: Let D be the event that the person has the disease, and E be the event that this person tests positive. We need to compute $p(D|E)$ from $p(\mathrm{D})$, $p(E|D)$, $p(E|\overline{D})$, $p(\overline{D})$.

$$p(D) = 1/100,000 = 0.00001 \quad p(\overline{D}) = 1 - 0.00001 = 0.99999$$

$$p(E|D) = .99 \quad p(\overline{E}|D) = .01 \quad p(E|\overline{D}) = .005 \quad p(\overline{E}|\overline{D}) = .995$$

$$p(D|E) = \frac{p(E|D)p(D)}{p(E|D)p(D) + p(E|\overline{D})p(\overline{D})}$$

$$= \frac{(0.99)(0.00001)}{(0.99)(0.00001) + (0.005)(0.99999)}$$

$$\approx 0.002$$

Can you use this formula to explain why the resulting probability is surprisingly small?

So, don't worry too much, if your test for this disease comes back positive.

Applying Bayes' Theorem

- What if the result is negative?

So, the probability you have the disease if you test negative is

$$p(D|\overline{E}) \approx 1 - 0.9999999 = 0.0000001.$$

$$p(\overline{D}|\overline{E}) = \frac{p(\overline{E}|\overline{D})p(\overline{D})}{p(\overline{E}|\overline{D})p(\overline{D}) + p(\overline{E}|D)p(D)}$$

$$= \frac{(0.995)(0.99999)}{(0.995)(0.99999) + (0.01)(0.00001)}$$

$$\approx 0.9999999$$

- So, it is extremely unlikely you have the disease if you test negative.

Generalized Bayes' Theorem

Generalized Bayes' Theorem: Suppose that E is an event from a sample space S and that $F_1, F_2, \ldots, F_n$ are mutually exclusive events such that

$$\bigcup_{i}^{n} F_i = S.$$

Assume that $p(E) \neq 0$ for $i = 1, 2, \ldots, n$. Then

$$p(F_j|E) = \frac{p(E|F_j)p(F_j)}{\sum_{i=1}^{n} p(E|F_i)p(F_i)}.$$

Bayesian Spam Filters

- How do we develop a tool for determining whether an email is likely to be spam?
- If we have an initial set B of spam messages and set G of non-spam messages. We can use this information along with Bayes' law to predict the probability that a new email message is spam.
- We look at a particular word w, and count the number of times that it occurs in B and in G; $n_B(w)$ and $n_G(w)$.
  - Estimated probability that an email containing w is spam: $p(w) = n_B(w)/|B|$
  - Estimated probability that an email containing w is not spam: $q(w) = n_G(w)/|G|$

Bayesian Spam Filters

- Let S be the event that the message is spam, and E be the event that the message contains the word w.
- Using Bayes' rule

$$p(S|E) = \frac{p(E|S)p(S)}{p(E|S)p(S) + p(E|\overline{S})p(\overline{S})}$$

$$p(S|E) = \frac{p(E|S)}{p(E|S) + p(E|\overline{S})}$$

Assuming that it is equally likely that an arbitrary message is spam and is not spam; i.e., $p(S) = ½$.

Bayesian Spam Filters

Using our empirical estimates of $p(E \mid S)$ and $p(E \mid \overline{S})$.

$$r(w) = \frac{p(w)}{p(w) + q(w)}$$

$r(w)$ estimates the probability that the message is spam. We can class the message as spam if $r(w)$ is above a **threshold**.

Bayesian Spam Filters

Example: We find that the word "Rolex" occurs in 250 out of 2000 spam messages and occurs in 5 out of 1000 non-spam messages. Estimate the probability that an incoming message is spam. Suppose our threshold for rejecting the email is 0.9.

Solution: $p(Rolex) = 250/2000 = .125$ and $q(Rolex) = 5/1000 = 0.005$.

$$r(Rolex) = \frac{p(Rolex)}{p(Rolex) + q(Rolex)} = \frac{0.125}{0.125 + .005} = \frac{0.125}{0.125 + .005} \approx 0.962$$

We class the message as spam and reject the email!

Bayesian Spam Filters using Multiple Words

- Accuracy can be improved by considering more than one word as evidence.
- Consider the case where E_1 and E_2 denote the events that the message contains the words w_1 and w_2 respectively.
- We make the simplifying assumption that the events are independent. And again we assume that $p(S) = ½$.

$$p(S|E_1 \cap E_2) = \frac{p(E_1|S)p(E_2|S)}{p(E_1|S)p(E_2|S) + p(E_1|\overline{S})p(E_2|\overline{S})}$$

$$r(w_1, w_2) = \frac{p(w_1)p(w_2)}{p(w_1)p(w_2) + q(w_1)q(w_2)}$$

Bayesian Spam Filters using Multiple Words

Example: We have 2000 spam messages and 1000 non-spam messages.

- The word “stock” occurs 400 times in the spam messages and 60 times in the non-spam.
- The word “undervalued” occurs in 200 spam messages and 25 non-spam.

Bayesian Spam Filters using Multiple Words

Solution: $p(stock) = 400/2000 = .2$

$q(stock) = 60/1000 = .06$

$p(undervalued) = 200/2000 = .1$

$q(undervalued) = 25/1000 = .025$

$$r(stock, undervalued) = \frac{p(stock)\,p(undervalued)}{p(stock)\,p(undervalued) + q(stock)\,q(undervalued)}$$

$$= \frac{(0.2)(0.1)}{(0.2)(0.1) + (0.06)(0.025)} \approx 0.930$$

If our threshold is .9, we class the message as spam and reject it.

Bayesian Spam Filters using Multiple Words

- In general, the more words we consider, the more accurate the spam filter. With the independence assumption if we consider *k* words:

$$p(S|\bigcap_{i=1}^{k} E_i) = \frac{\Pi_{i=1}^{k} p(E_i|S)}{\Pi_{i=1}^{k} p(E_1|S) + \Pi_{i=1}^{k} p(E_i|\overline{S})}$$

$$r(w_1, w_2, ...w_n) = \frac{\Pi_{i}^{k} p(w_i)}{\Pi_{i=1}^{k} p(w_i) + \Pi_{i=1}^{k} q(w_i)}$$

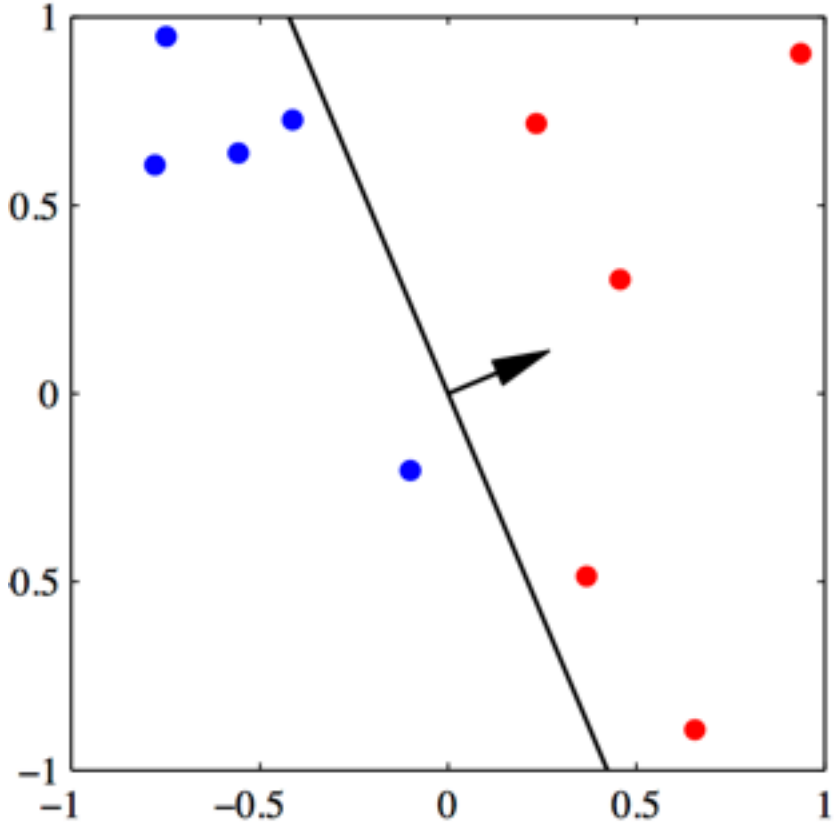
We can further improve the filter by considering pairs of words as a single block or certain types of strings.

Discriminant analysis

Section 3

Linear classification

- Focus on linear classification model, i.e., the decision boundary is a linear function of $\boldsymbol{x}$
  - Defined by (D – 1)-dimensional hyperplane
- If the data can be separated exactly by linear decision surfaces, they are called linearly separable
- Implicit assumption: Classes can be modeled well by Gaussians
- Simply speaking, treat classification as a projection problem

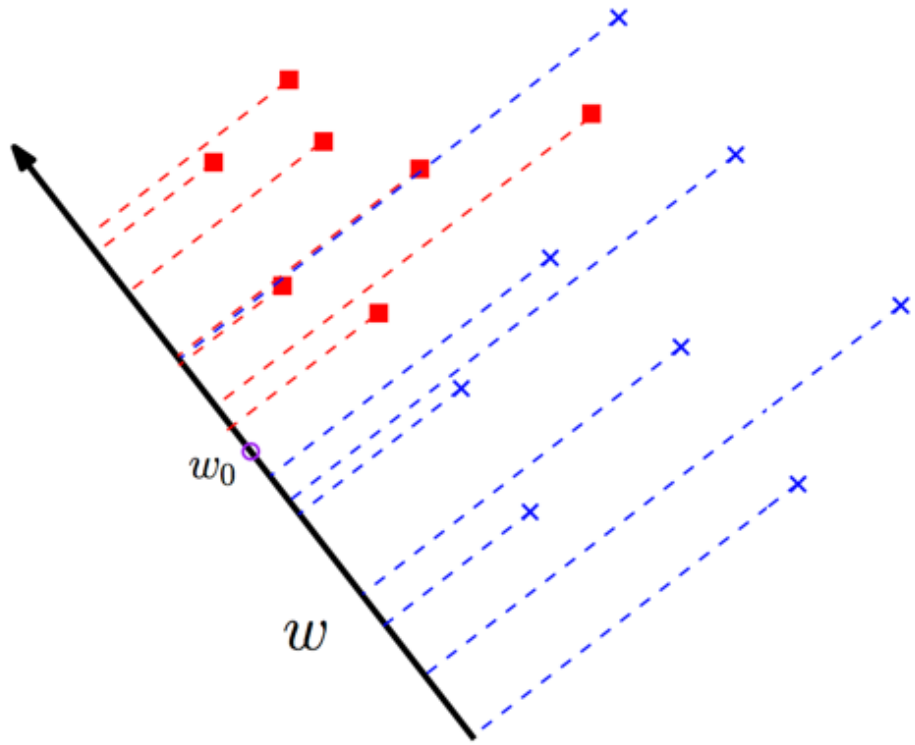


From PRML (Bishop, 2006)

Projection

- Assume we know the basic vector $\boldsymbol{w}$, we can compute the projection, y, of any points, x.

- Threshold w_0, such that we decide on C_1 if $y \geqq w_0$ and C_2 otherwise.



+7, 14, 21, 28, ..., 70

7, 77 ≈ 1 นาที 17 วิ

747

2 นาที 27 วิ

มีนาฬิกาปลุก 2 เรือนเหมือนกัน

เรือนแรกทุก 7 วิ : 7, 14, 21, 28, ... → $7n = 3600$ (514)

6 นาที ≈ 7

60 นาที ≈ 70

} 89

เรือนที่ 2 ทุกวินาทีที่ลงท้าย 7 : 7, 17, 27, 37, ...

เปิดพร้อมกัน 1 ชั่วโมง : 3600 วินาที

3597

3587

$10n + 7$ (n = 359, 7 วินาที)

3600

3598 / 7

3577

เรือนแรกตีไป : 514 ครั้ง

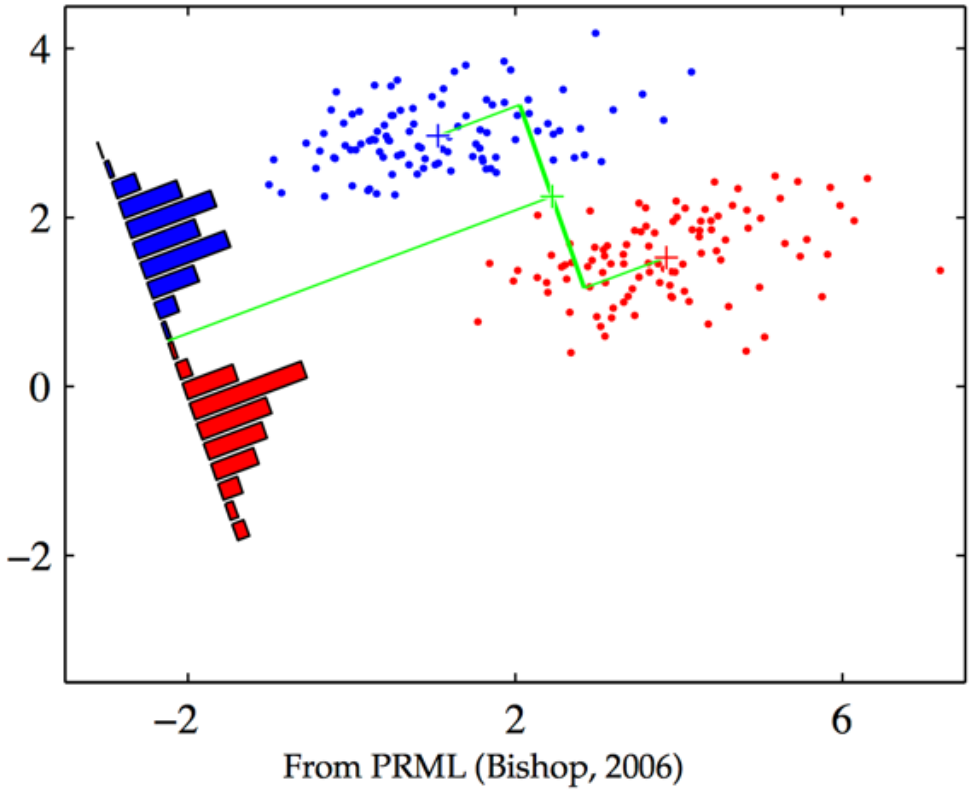
เรือนสอง : 359 ครั้ง

} 873 - 51 = 822

$70n + 7 = 3600$

$70n = 3593$

LDA



From PRML (Bishop, 2006)

- Separate samples of distinct groups by projecting them onto a space that
  - Maximize their between-class separability while
  - Minimize their within-class variability

Linear discriminant analysis

- We model the distribution of the predictors X separately in each of the response classes (i.e. given Y), and then use Bayes' theorem to flip these around into estimates for Pr(Y = k | X = x)
- Assume that we have 1 predictor
- Suppose we assume that $f_k(x)$ is normal or Gaussian

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

Bayes' theorem for classification

- Let π_k be the overall or prior probability of the kth class;
- This is the probability that a given observation is associated with the k^{th} category of the response variable Y.
- Let $f_k(x) \equiv \Pr(X = x|Y = k)$ denote the density function of X for an observation that comes from the k^{th} class

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

Bays' theorem for classification

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}.$$

- Taking a log, we get

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

- We will assign x to class k if $\delta_k(x)$ is largest.

Deriving Bayes' classifier solution

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{p(X=x)}$$

$$\log\left(p_k(x)\right) = log(\pi_k) + log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + log\left(exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)\right) - log\left(p(X=x)\right)$$

$$\log\left(p_k(x)\right) = log(\pi_k) + log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right) - log\left(p(X=x)\right)$$

$$\log\left(p_k(x)\right) = log(\pi_k) + log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{x^2}{2\sigma^2} + \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} - log\left(p(X=x)\right)$$

$$\delta_k(x) = log(\pi_k) + \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2}$$

Linear discriminant analysis (LDA)

- The linear discriminant analysis method approximates the Bayes classifier by plugging estimates for π_k, μ_k, and σ_2:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{\pi}_k = n_k/n.$$

LDA discriminant function

- Assign x to class k if

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

is the largest

Bayes decision boundary and LDA

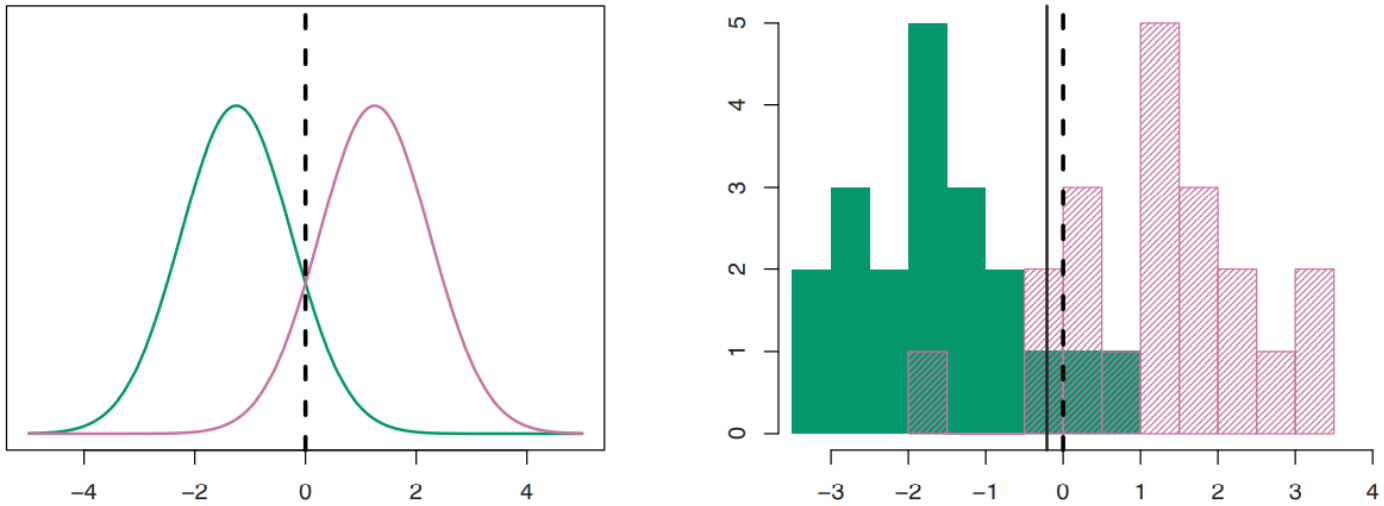


FIGURE 4.4. Left: *Two one-dimensional normal density functions are shown. The dashed vertical line represents the Bayes decision boundary.* Right: *20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is again shown as a dashed vertical line. The solid vertical line represents the LDA decision boundary estimated from the training data.*

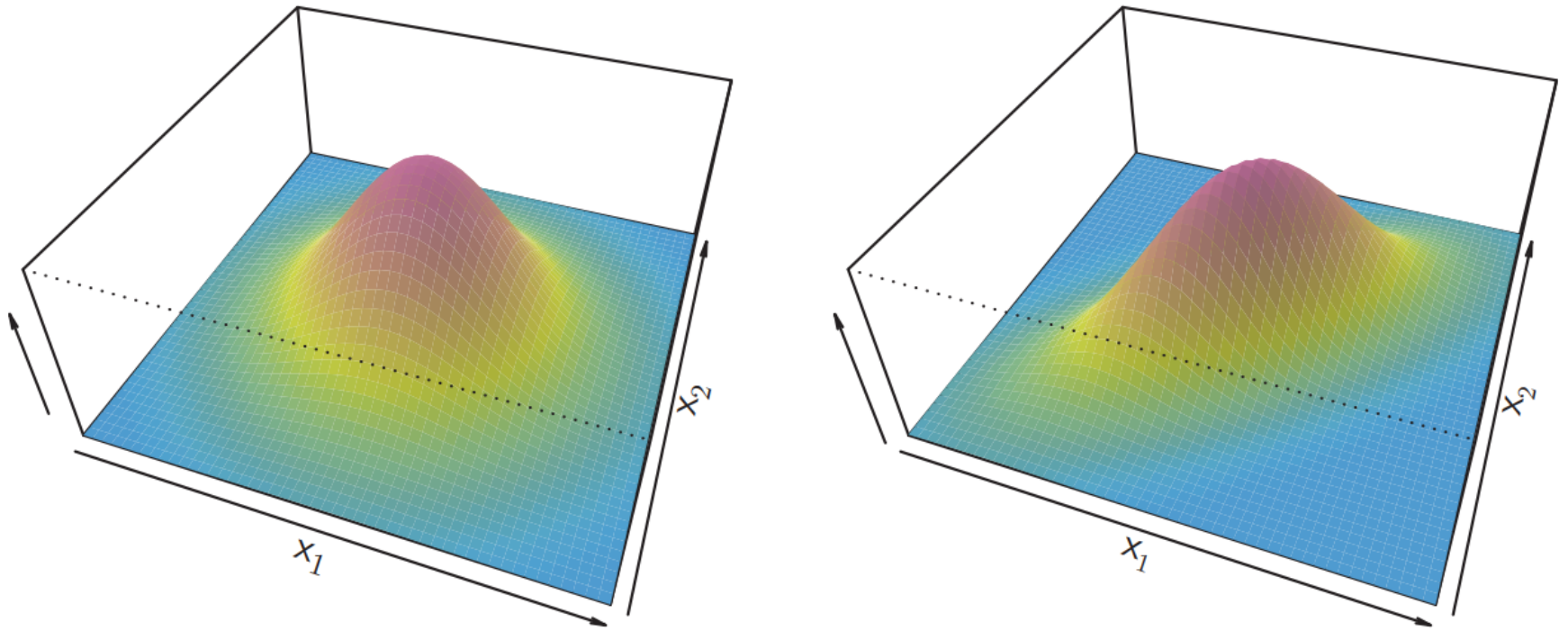


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

Discriminant analysis with more than one predictors

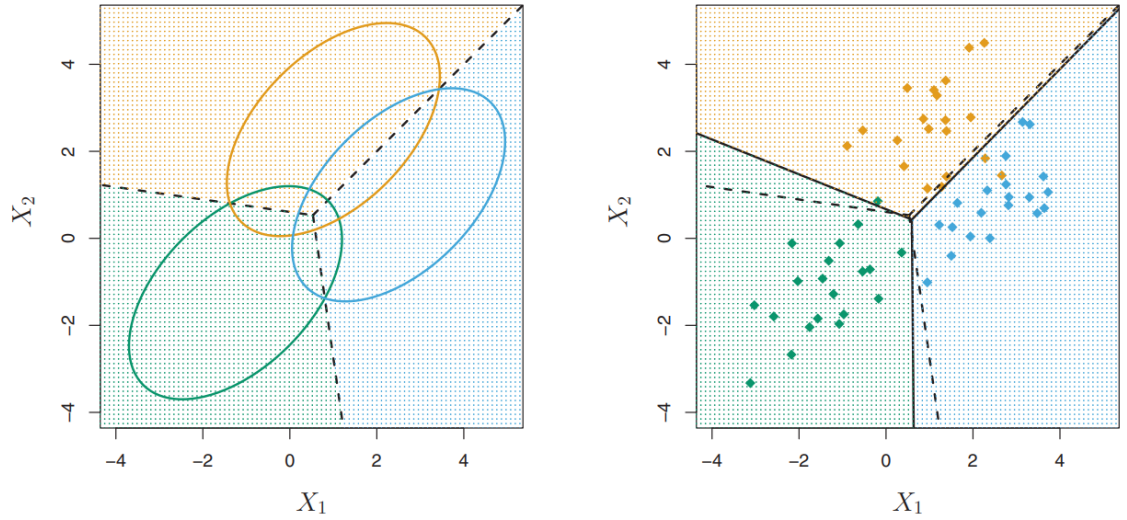


FIGURE 4.6. *An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.* Left: *Ellipses that contain 95 % of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries.* Right: 20 *observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.*

Quadratic Discriminant Analysis

A generic class of discriminant analysis

From the distribution function, if we take log of the function

$$\log P(y=k|x) = \log P(x|y=k) + \log P(y=k) + Cst$$
$$= -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^t\Sigma_k^{-1}(x-\mu_k) + \log P(y=k) + Cst,$$

where the constant term Cst corresponds to the denominator $P(x)$, in addition to other constant terms from the Gaussian. The predicted class is the one that maximises this log-posterior.

LDA is a special case of LDA when each class assume shared covariance.

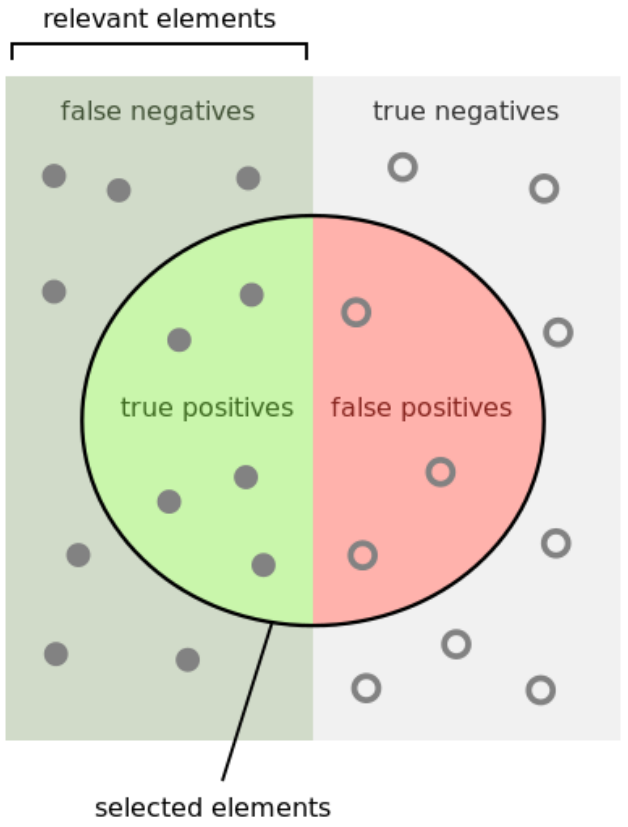
Accuracy, Precision and Recall

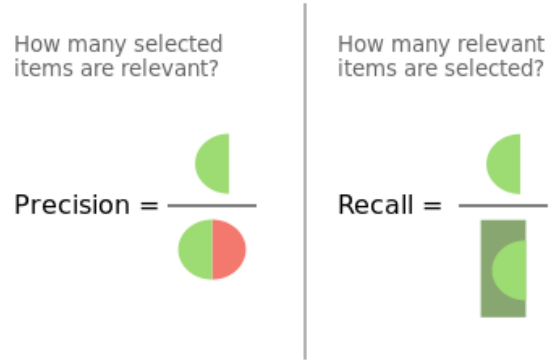
	Actual Positive (p)	Actual Negative (n)
The model says "Yes" = positive (y)	True positives	False positives
The model says "No" = not positive (n)	False negatives	True negatives

- Accuracy = (TP + TN)/(TP + FP + TN + FN)
- Recall (Completeness) = true positive rate = TP/(TP + FN)
- Precision (Exactness) = the accuracy over the cases predicted to be positive, TP/(TP + FP)
- F-measure = the harmonic mean of precision and recall

  = the balance between recall and precision

  $= 2 . \frac{precision * recall}{precision + recall}$





Receiver operating characteristics Area under the ROC curve

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

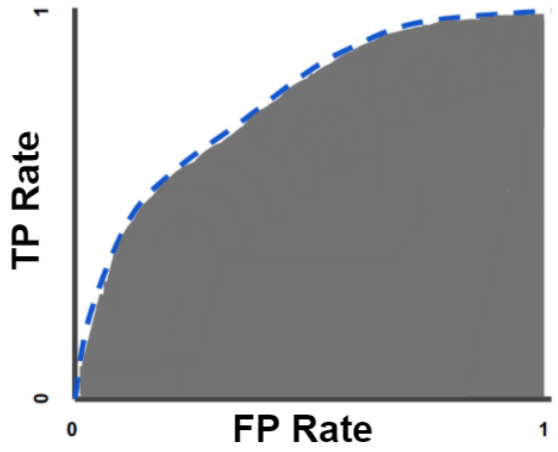


Figure 5. AUC (Area under the ROC Curve).

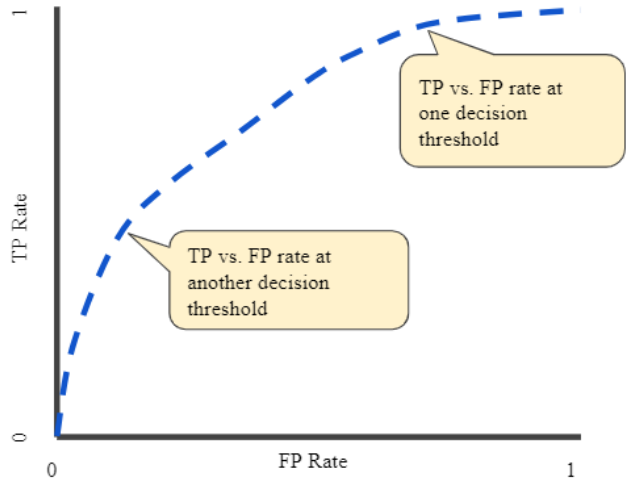
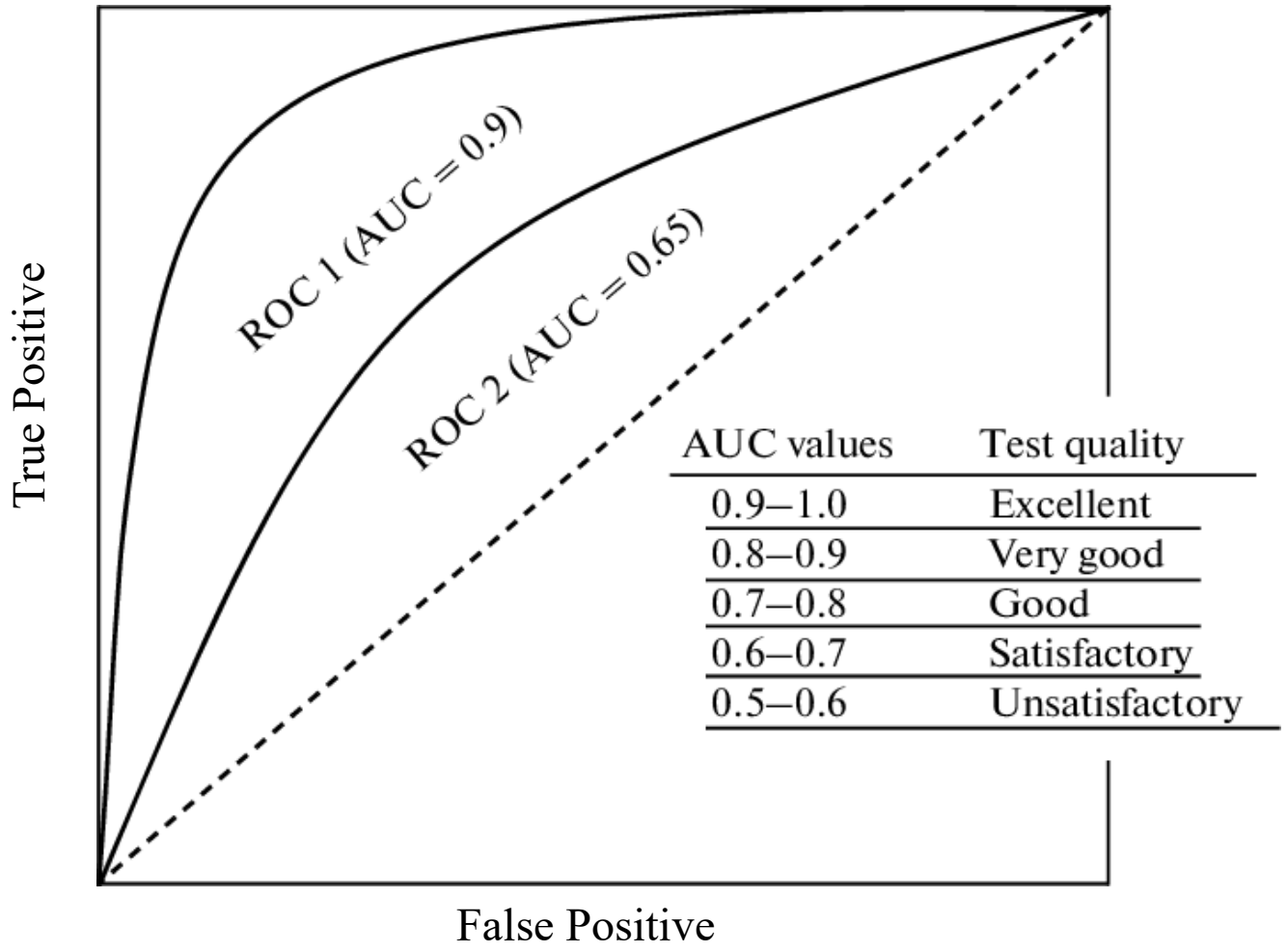


Figure 4. TP vs. FP rate at different classification thresholds.

AUC - ROC



AUC values	Test quality
0.9–1.0	Excellent
0.8–0.9	Very good
0.7–0.8	Good
0.6–0.7	Satisfactory
0.5–0.6	Unsatisfactory

Coding Practice

- Implementing linear discriminant analysis (LDA) in Python - IBM Developer
- Linear Discriminant Analysis Made Simple & How To Tutorial (spotintelligence.com)

End of Lecture 1

Question?