Lecture 1. Introduction to Machine Learning

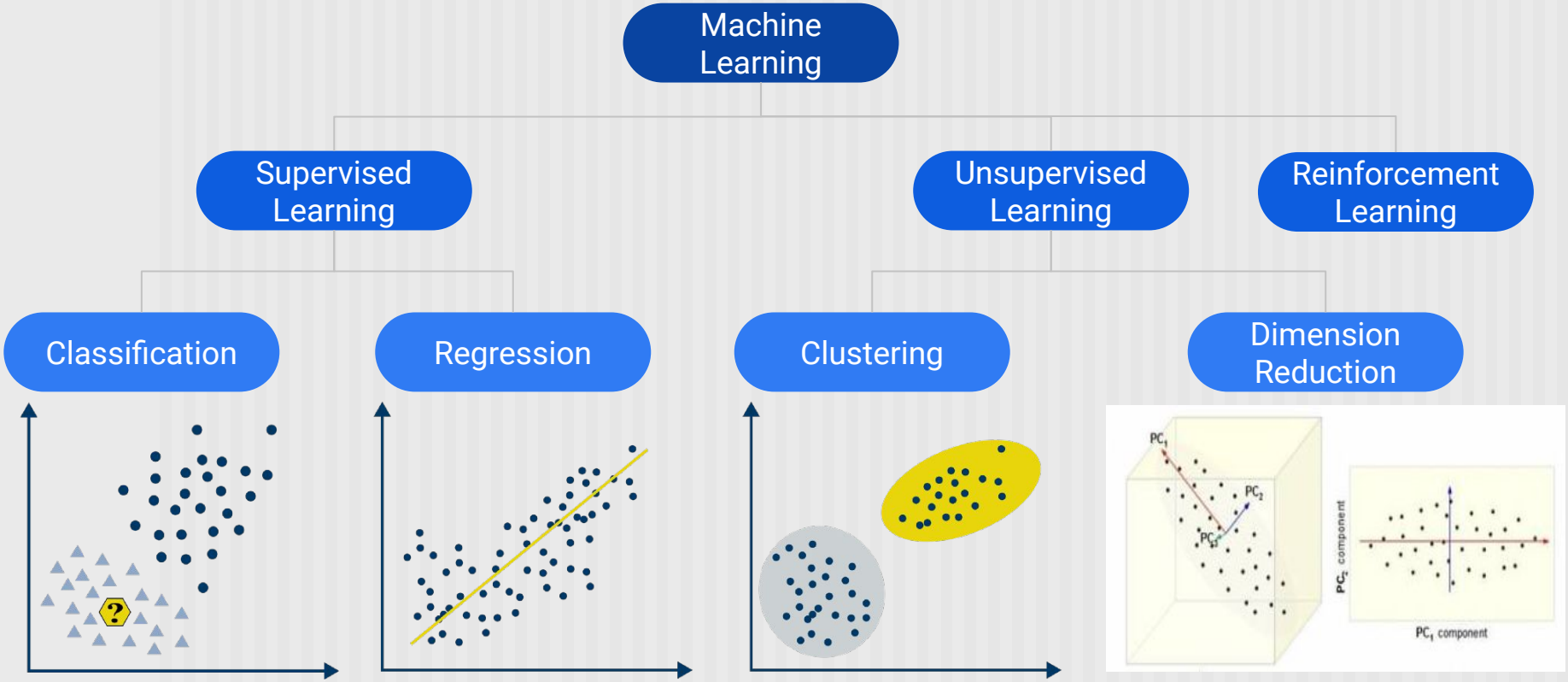
Intro to Image Processing

KNN Classifier

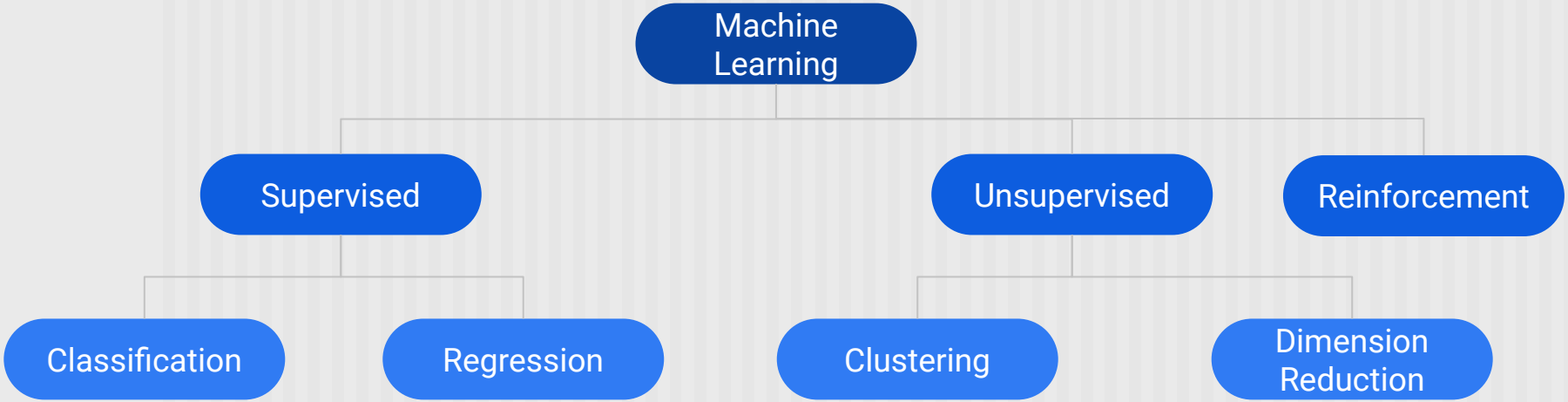
Naive Bayes Classifier

Suthep “Jogie” Madarasmi, Ph.D.

Three Classes of ML Algorithms



Three Classes of ML Algorithms



Predict a category
Input: Labeled data set
Output: Discrete category
Algorithms: Decision Trees, ML Perceptrons, …

Predict a value
Input: Labeled data set
Output: Continuous Values
Algorithms: Linear Regression, Decision Trees, Random Forests.

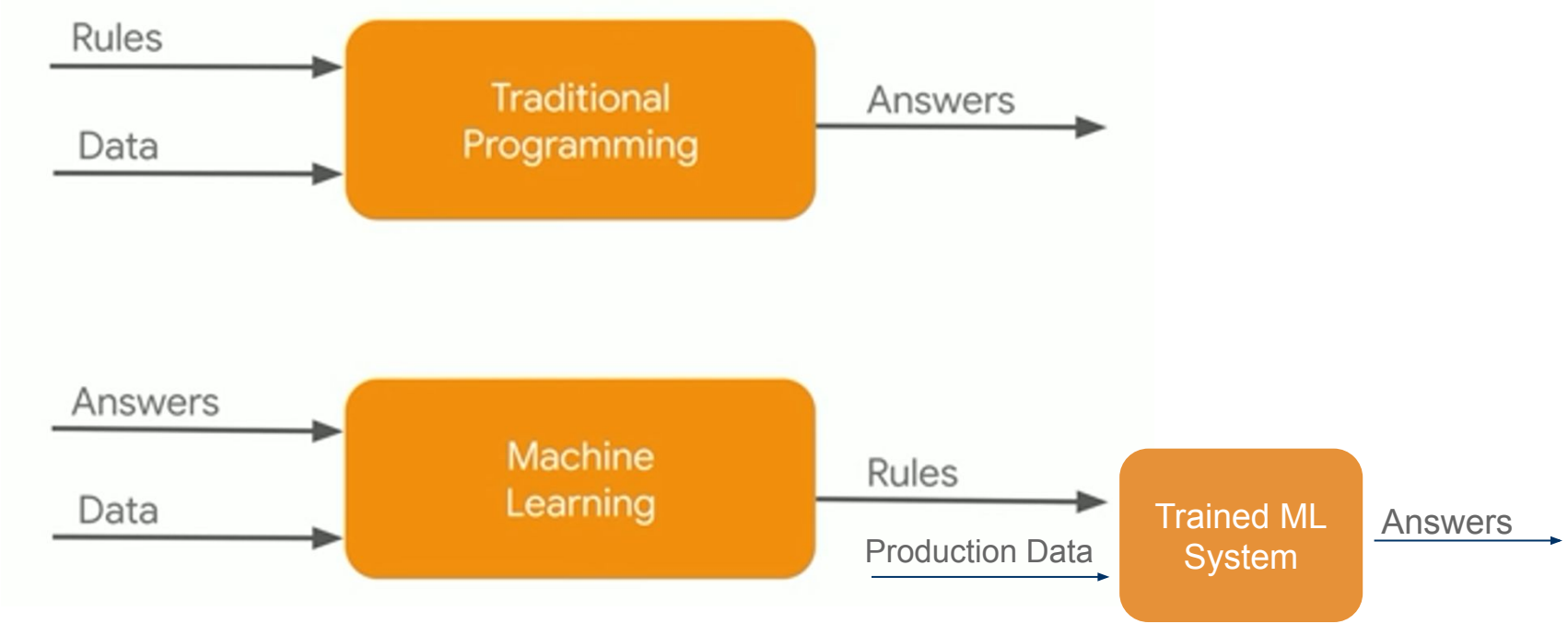
Identify groups of objects.
Input: Data set
Output: Groups
Algorithms: K-means Clustering, ANN

Reduces number of variables
Input: Data set
Output: New Data Set
Algorithm: Principal Component Analysis (PCA)

Recent Developments and Trends

- ChatGPT https://openai.com/blog/chatgpt/
- Dall-e https://openai.com/dall-e-2/

Typical Machine Learning Scenario



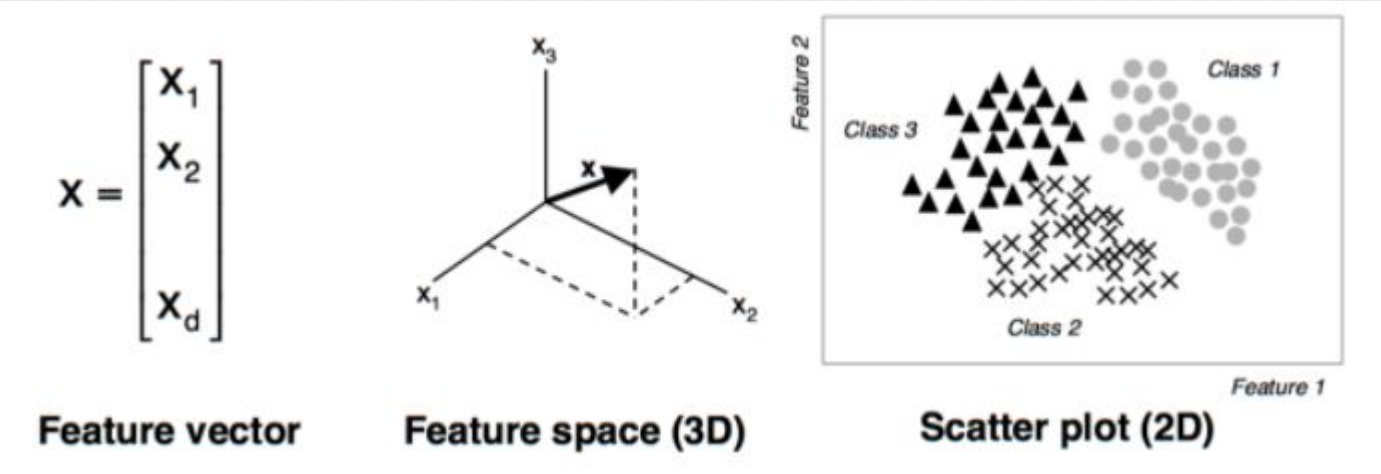
Input: Labeled dataset (Answers) of feature vectors (Data)

Pattern Recognition: Classification of Feature Vectors

https://brilliant.org/wiki/feature-vector/

Area of study is formally called “Pattern Recognition”

- Input: a feature vector called “*pattern*”
- Output: “*recognition*” class label of the input pattern.

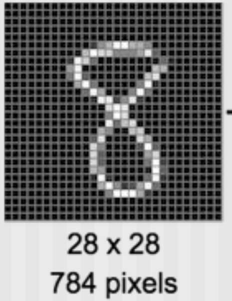


Example Feature vector for characters

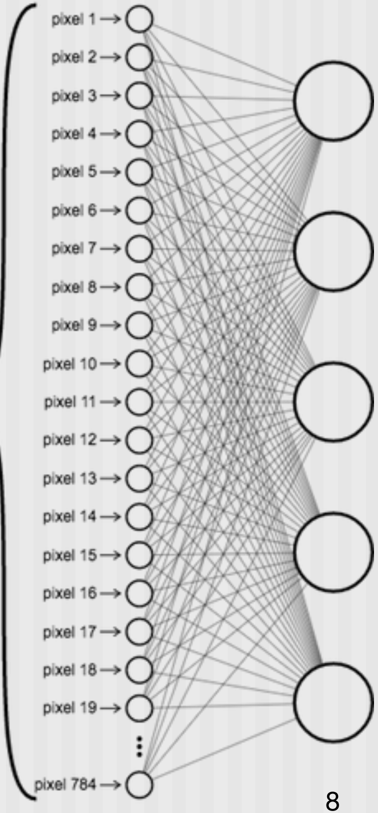
(class) character	area	height	width	number #holes	number #strokes	(cx,cy) center	best axis	least inertia
'A'	medium	high	3/4	1	3	1/2,2/3	90	medium
'B'	medium	high	3/4	2	1	1/3,1/2	90	large
'8'	medium	high	2/3	2	0	1/2,1/2	90	medium
'O'	medium	high	2/3	1	0	1/2,1/2	90	large
'1'	low	high	1/4	0	1	1/2,1/2	90	low
'W'	high	high	1	0	4	1/2,2/3	90	large
'X'	high	high	3/4	0	2	1/2,1/2	?	large
'*'	medium	low	1/2	0	0	1/2,1/2	?	large
'-'	low	low	2/3	0	1	1/2,1/2	0	low
'/'	low	high	2/3	0	1	1/2,1/2	60	low

Raw Feature Vector for characters (Images)

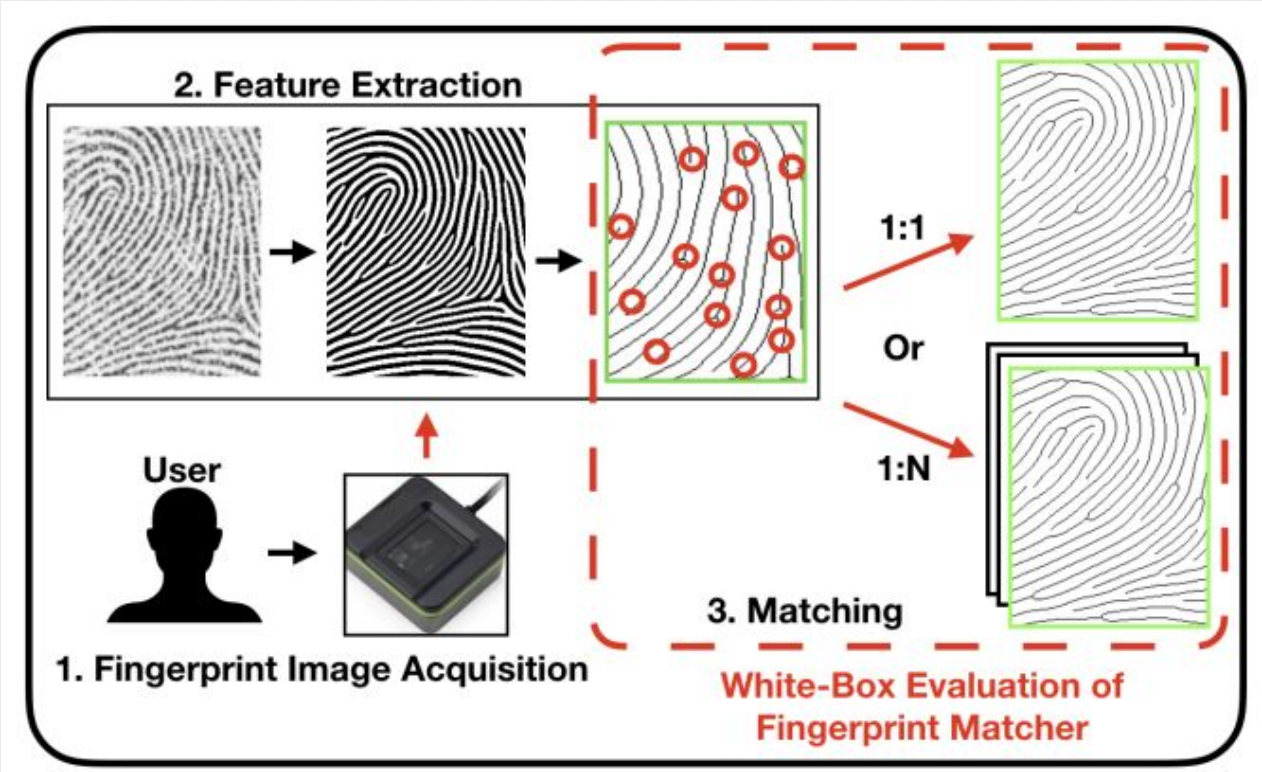
No Preprocessing of the feature.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Example: Fingerprint Recognition System



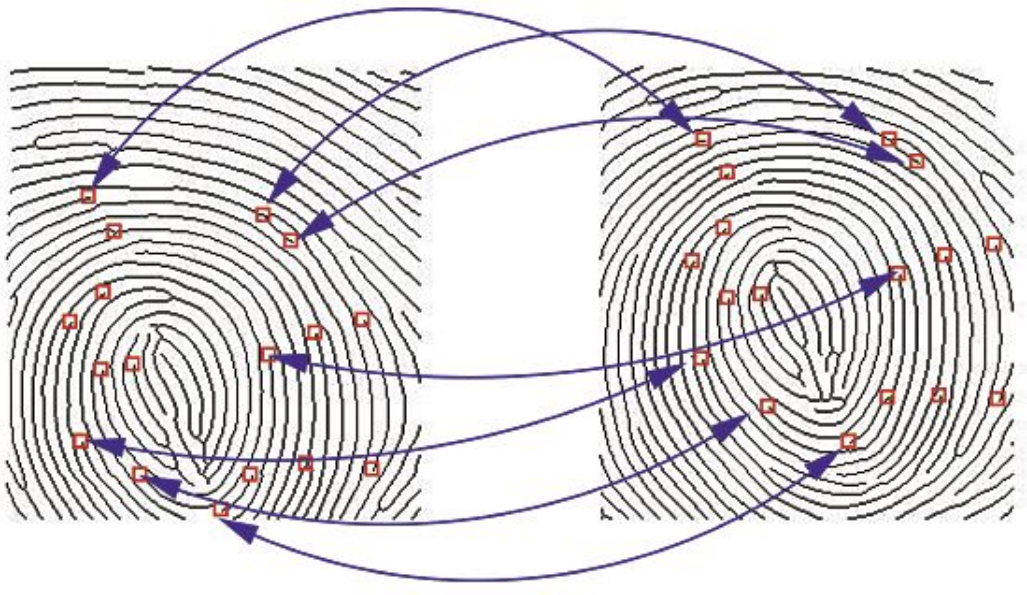
Verification vs. Identification

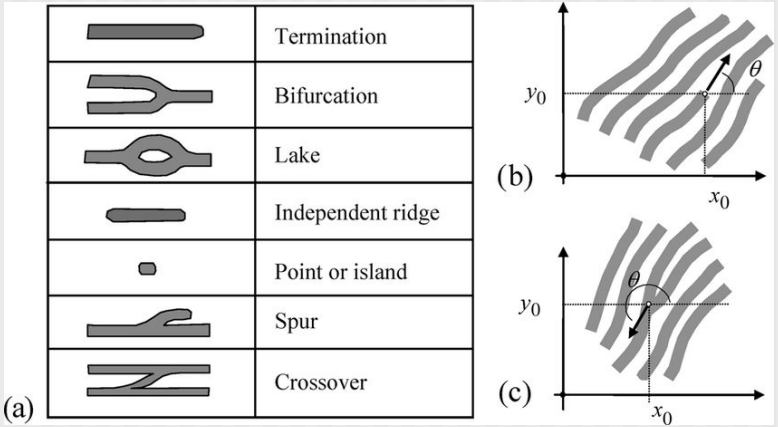
https://arxiv.org/pdf/2008.00128.pdf

Fingerprint Features

Based on Minutiae

https://www.researchgate.net/figure/Fingerprint-matching-based-on-minutiae_fig1_301253150

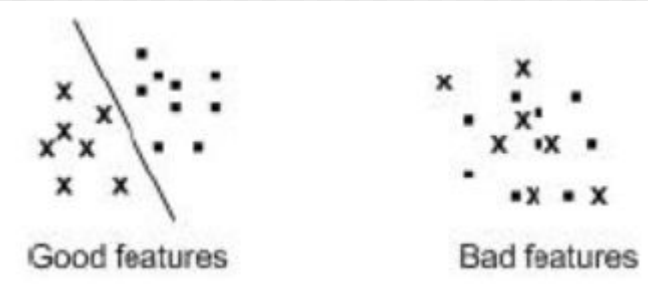


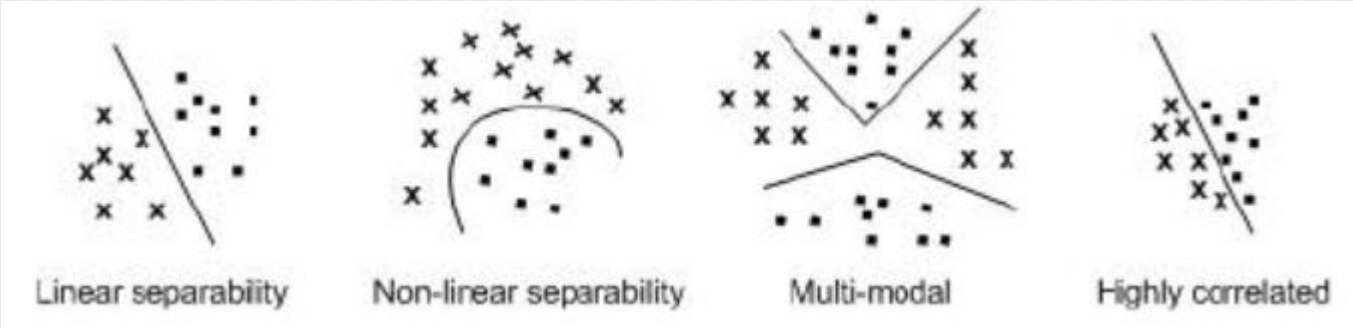


https://www.researchgate.net/figure/Fingerprint-minutiae-a-The-common-fingerprint-minutiae-types-b-ridge-ending-x-y_fig4_3455239

Pattern Recognition Example

The goal of a classifier is to partition feature space into class-labeled decision regions or to "Find Decision Boundaries". Borders between decision regions are called decision boundaries. https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm

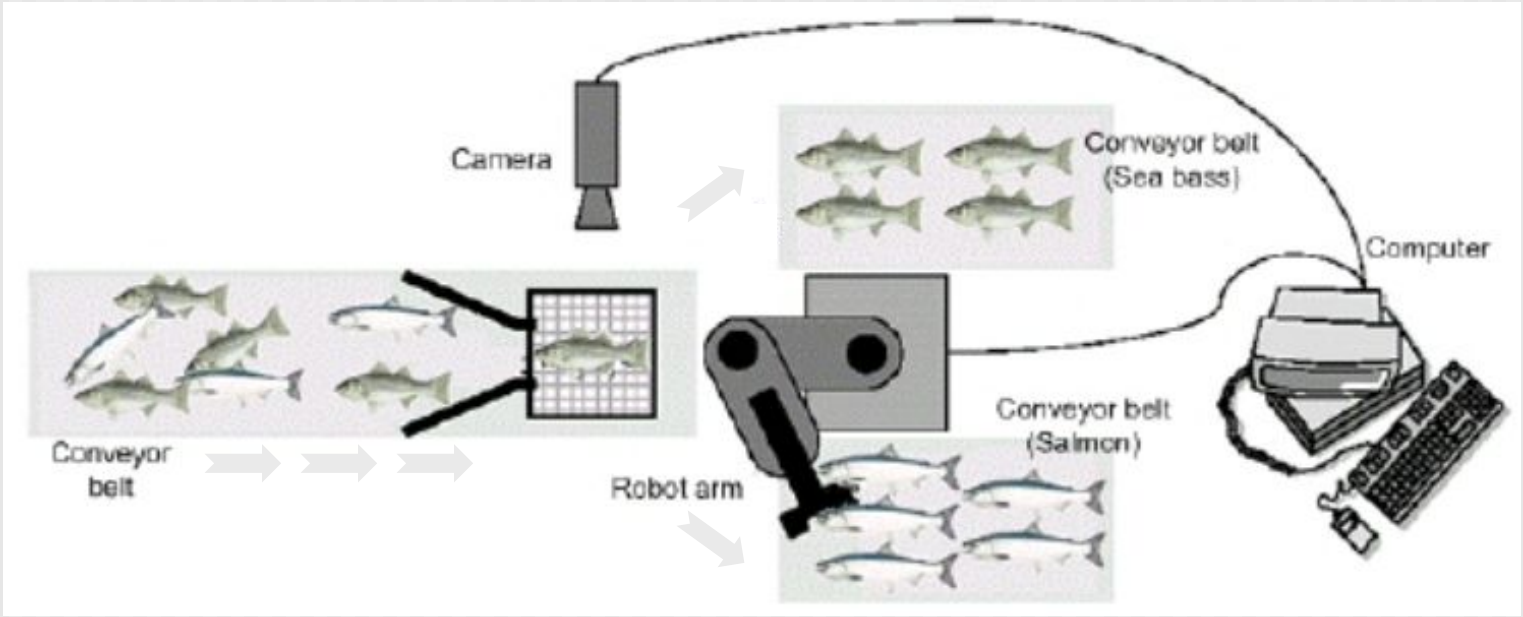




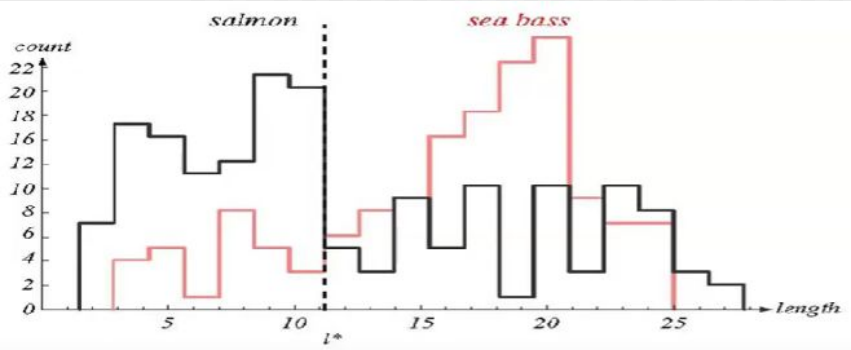
Example of a 2-feature classifier

Automated Sea Bass (ปลากระพง) and Salmon Packing System

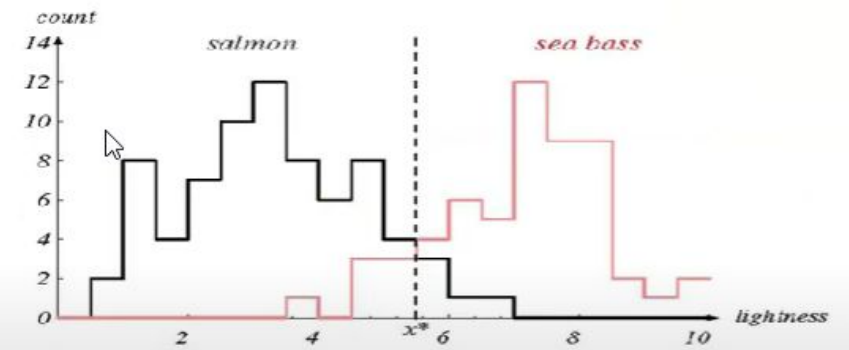
https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm



Sea bass vs. Salmon



Width Histogram



Brightness Histogram

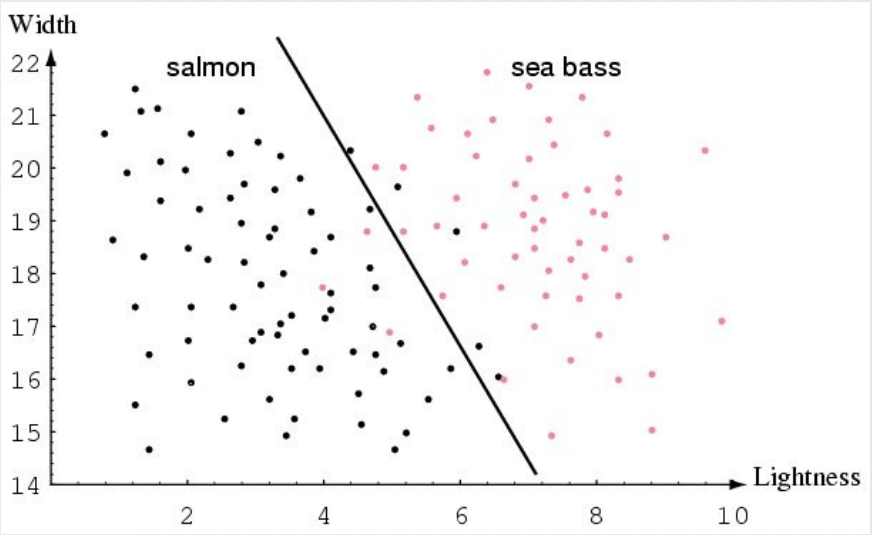
https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm

Nearly all probability functions are conditional. Here the condition is "given the fish accepted by factory".

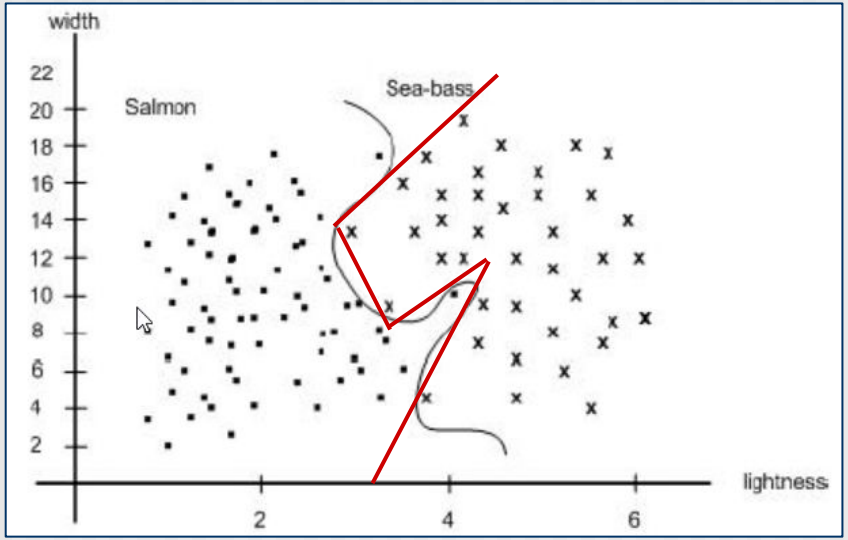
Photo from Makro. Which one is Salmon?

Sea bass vs. Salmon

https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm



Linear Classifier

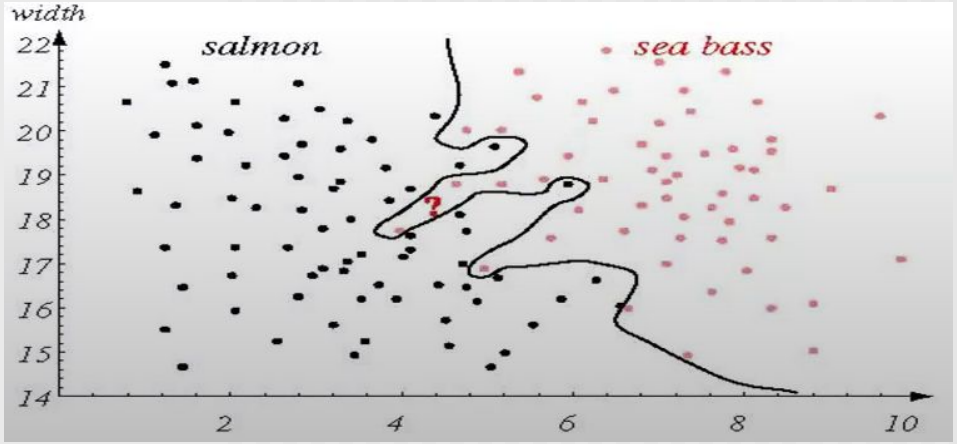


Piecewise Linear Classifier
vs. Non-Linear Classifier

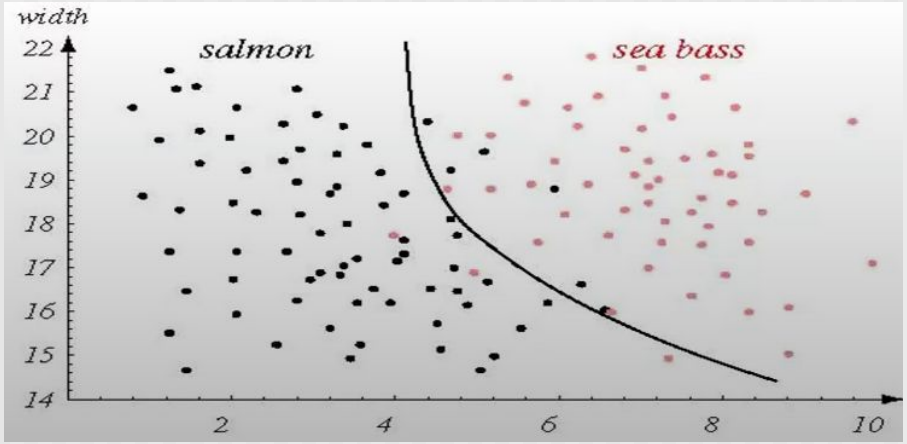
Problem: We often want classification to be invariant to rotation, translation, scale, lighting.

Sea bass vs. Salmon

https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm

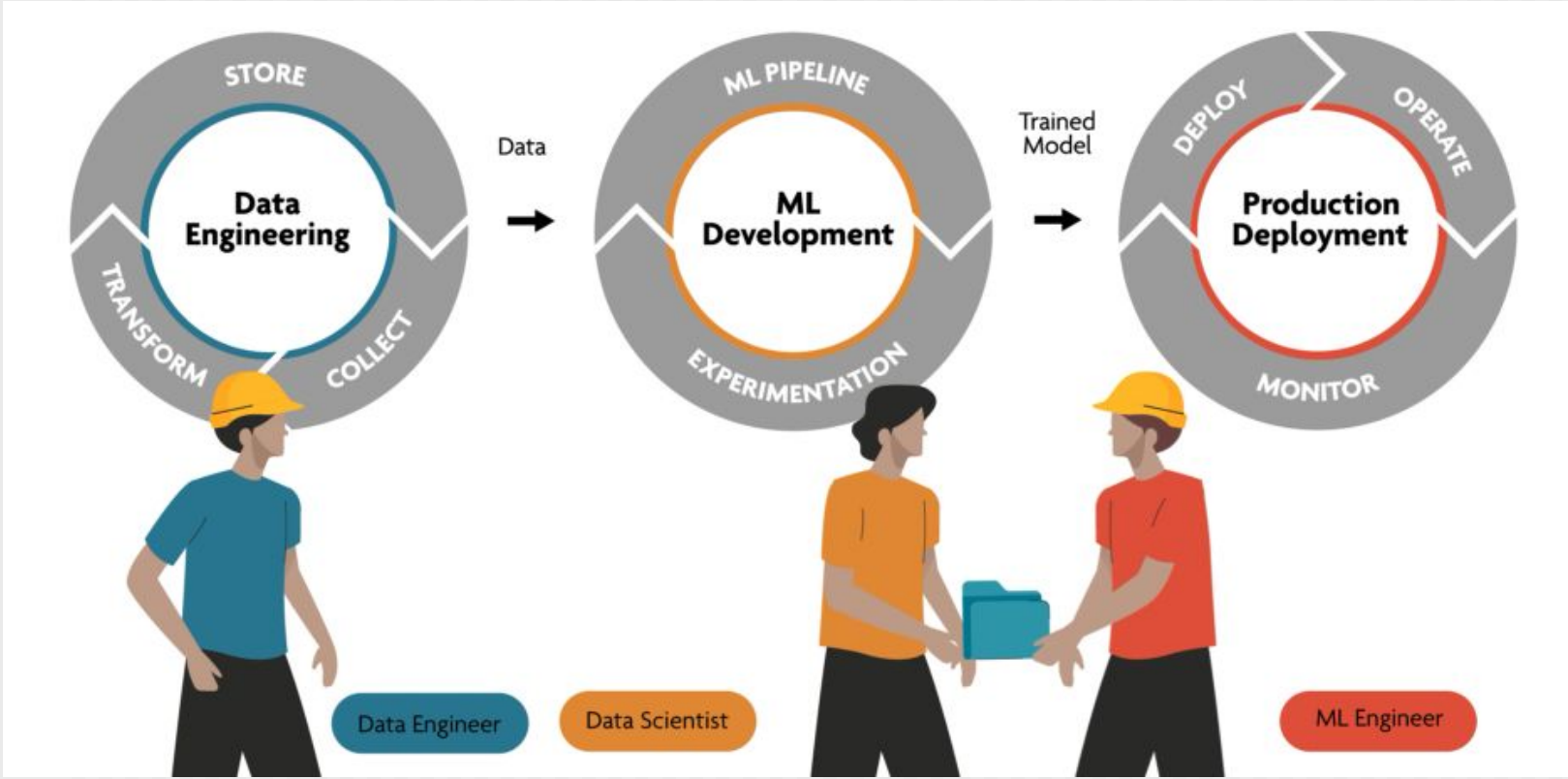


Complex Model - Overfitting. Performs 100% on training data but will likely fail on testing dataset and in production.

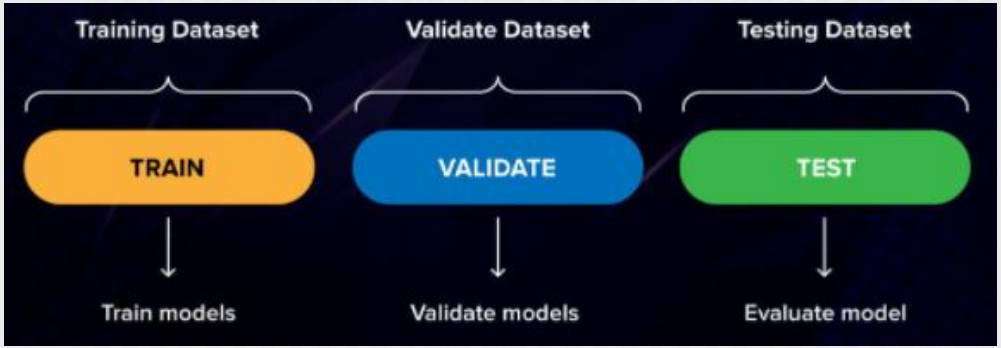


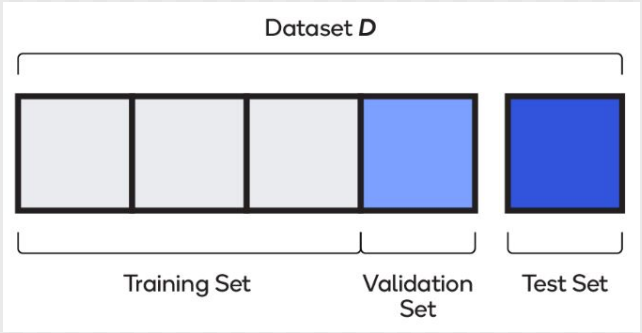
Simpler Fitting Model, but better on testing dataset and in production.

ML Development Pipeline



Dataset partitioning for supervised learning



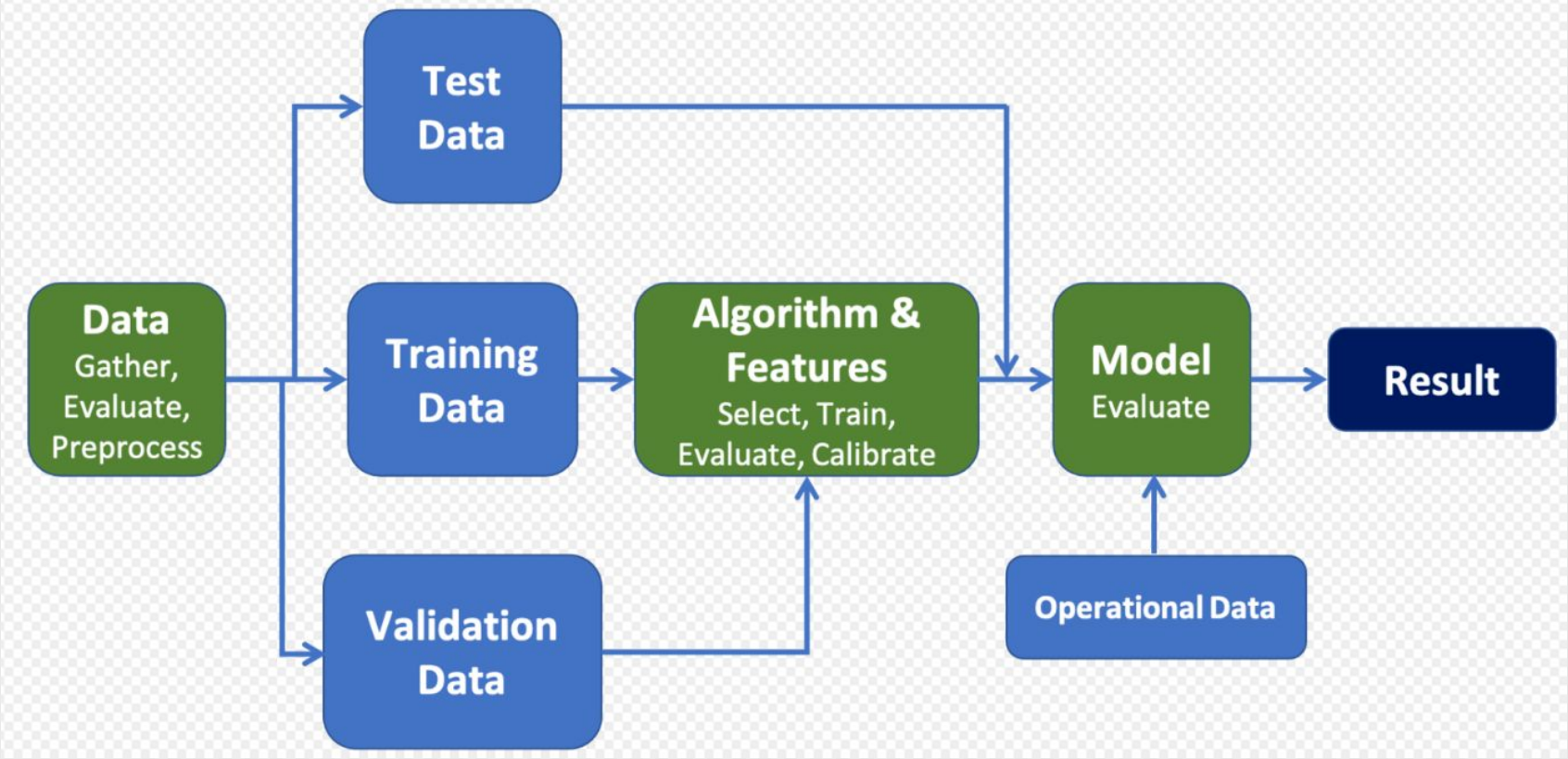


Training - About 75%-80% of the total dataset
Validation - About 5% (later added to test set)
Testing - About 20%-25%

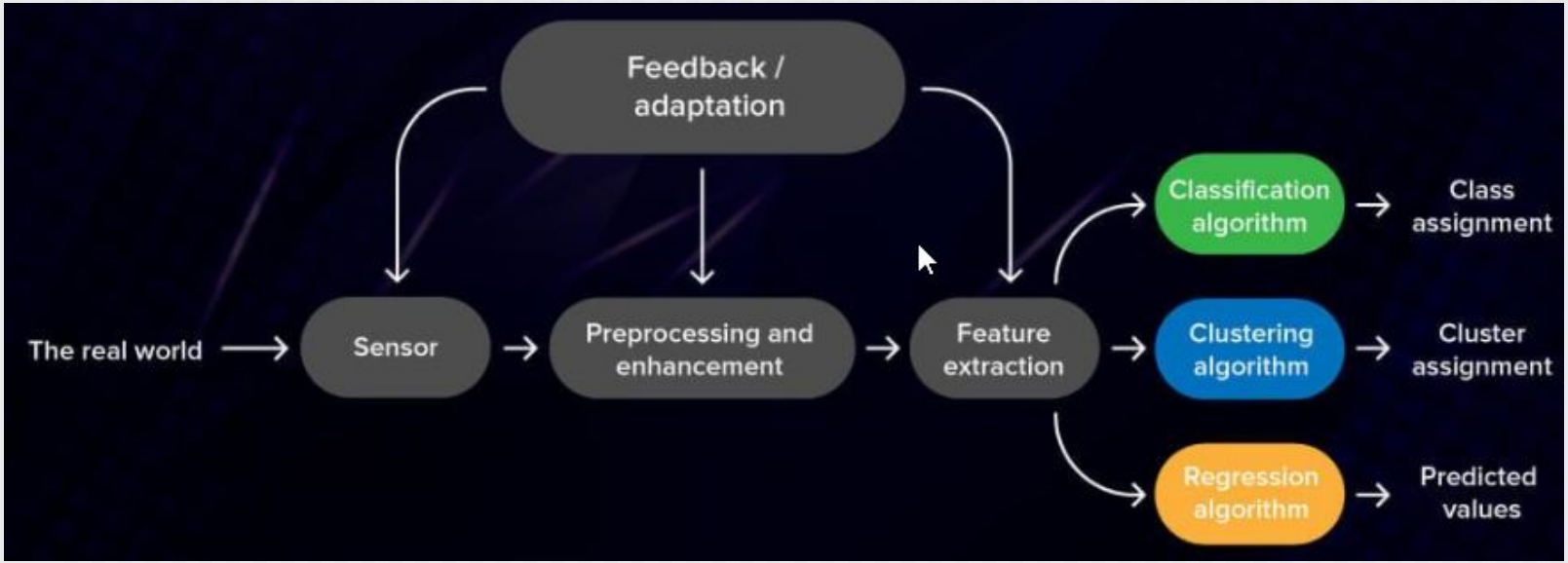
Validation - Use this dataset to guide the training. If the accuracy of training data set increases, but the accuracy of validation data set is same or decreases, then model **overfitting** and should **stop training**. Otherwise, it is added to training dataset.

Example of Overfitting case ater training with the added validate dataset:

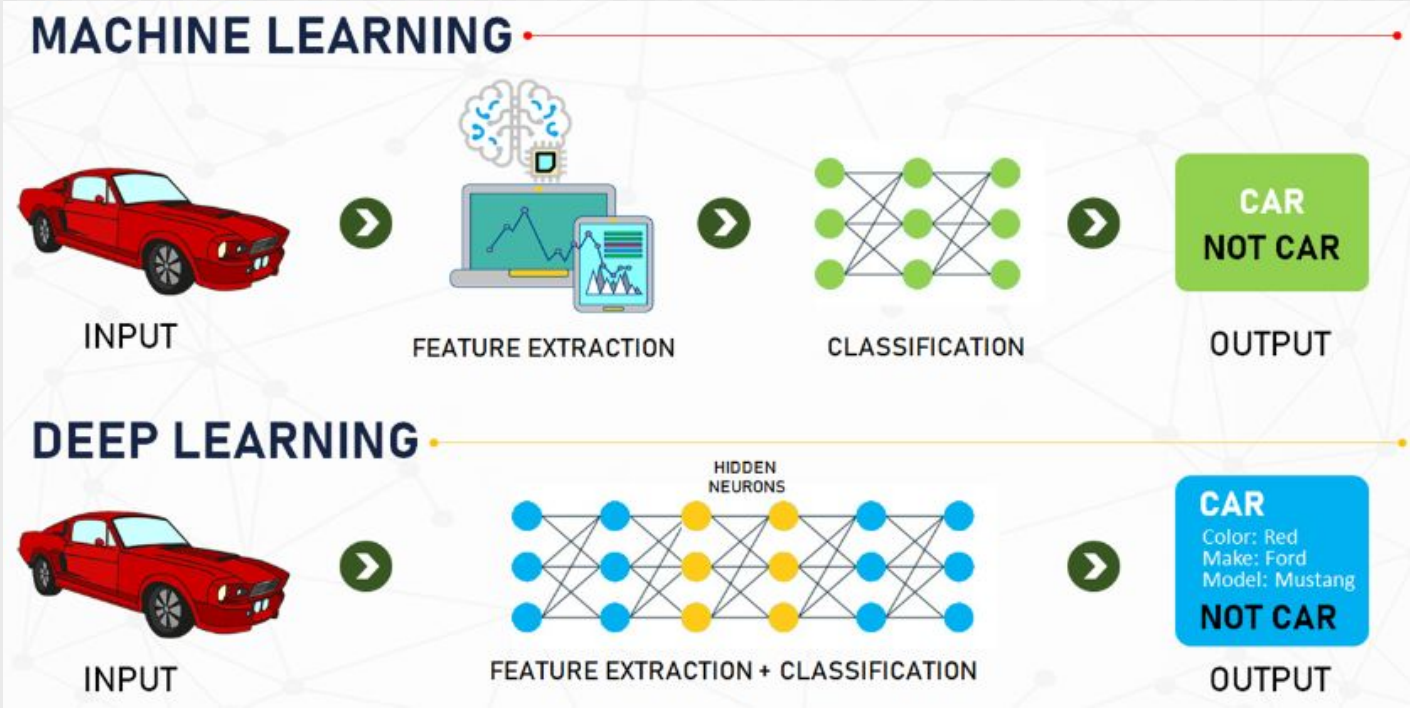
- Training dataset accuracy improves: 95% → 97%
- Validate dataset accuracy declines: 94% → 93%


Test Data
Data
Gather, Evaluate, Preprocess
Training Data
Algorithm & Features
Select, Train, Evaluate, Calibrate
Model
Evaluate
Result
Validation Data
Operational Data

Generic Machine Learning System



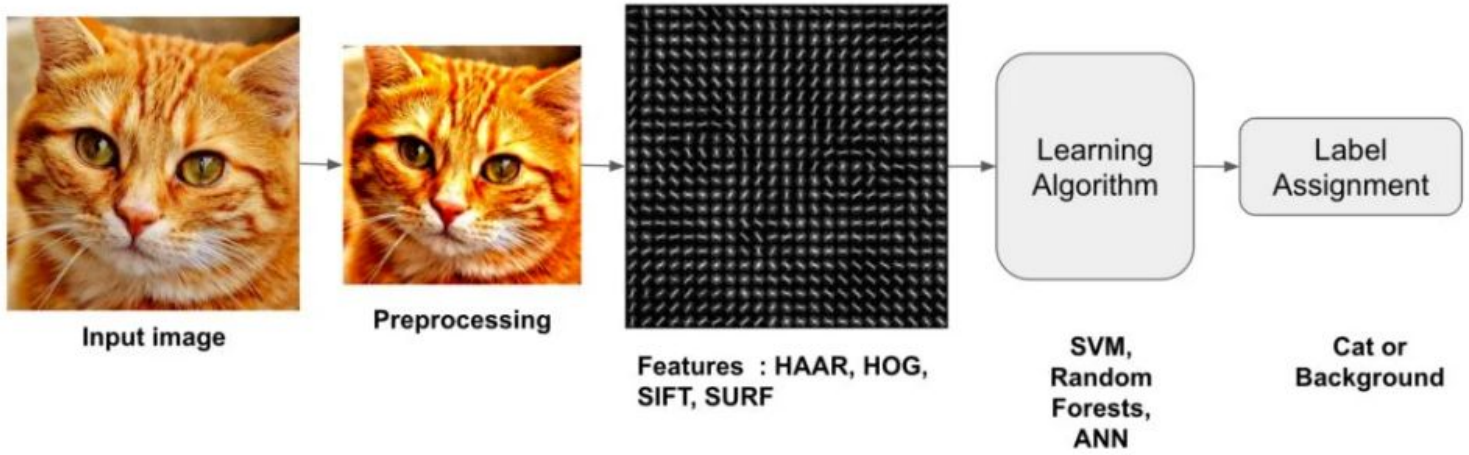
Classical Machine Learning vs. Deep Learning



In this course we include Deep Learning into the Machine Learning topic.

Deep Learning vs. Traditional Classification

https://learnopencv.com/image-recognition-and-object-detection-part1/



Interestingly, many traditional computer vision image classification algorithms follow this pipeline, while Deep Learning based algorithms bypass the feature extraction step completely.

What are Features?

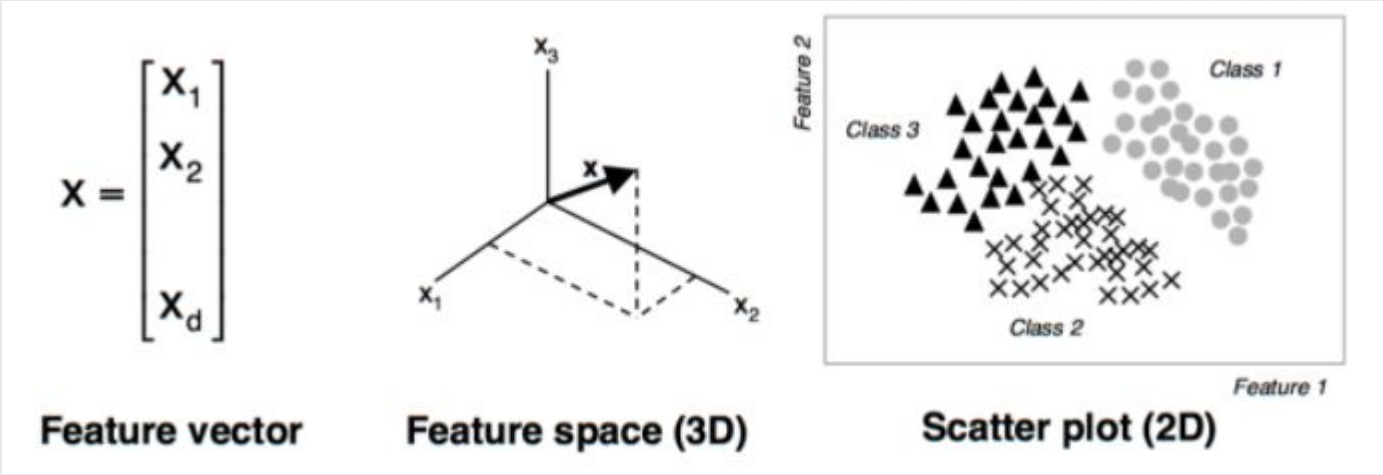
What are Class Labels?

Pattern Recognition: Classification of Feature Vectors

https://brilliant.org/wiki/feature-vector/

Area of study is formally called “Pattern Recognition”

- Input: a **feature vector** called “*pattern*”
- Output: “*recognition*” **class label** of the input pattern.

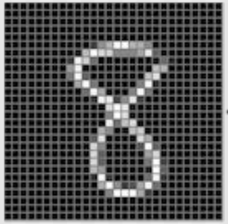


Example Feature vector for characters

(class) character	area	height	width	number #holes	number #strokes	(cx,cy) center	best axis	least inertia
'A'	medium	high	3/4	1	3	1/2,2/3	90	medium
'B'	medium	high	3/4	2	1	1/3,1/2	90	large
'8'	medium	high	2/3	2	0	1/2,1/2	90	medium
'O'	medium	high	2/3	1	0	1/2,1/2	90	large
'1'	low	high	1/4	0	1	1/2,1/2	90	low
'W'	high	high	1	0	4	1/2,2/3	90	large
'X'	high	high	3/4	0	2	1/2,1/2	?	large
'*'	medium	low	1/2	0	0	1/2,1/2	?	large
'-'	low	low	2/3	0	1	1/2,1/2	0	low
'/'	low	high	2/3	0	1	1/2,1/2	60	low

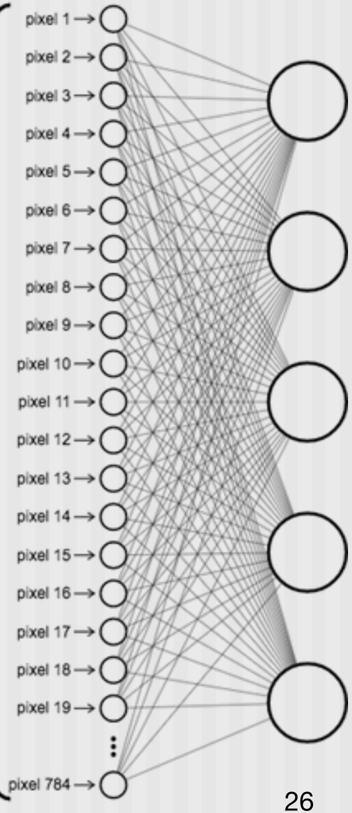
Raw Feature Vector for characters (Images)

No Preprocessing
of the feature.



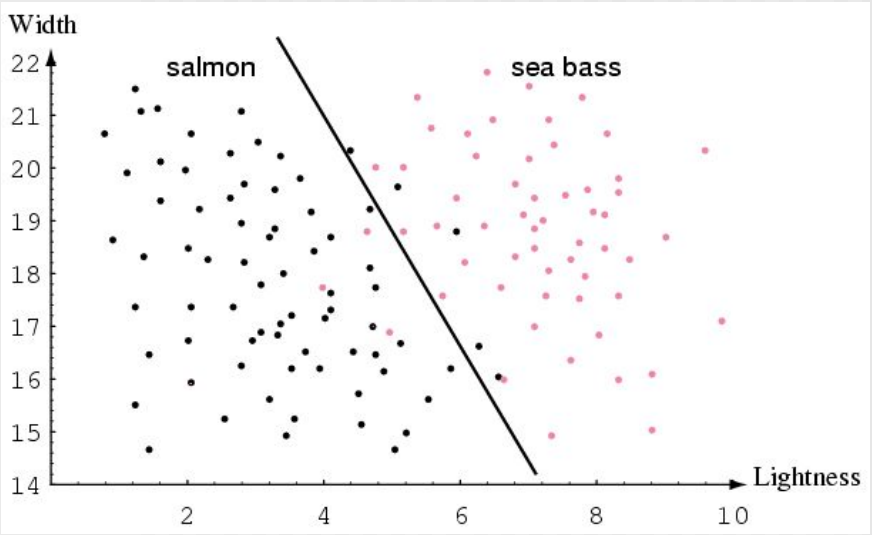
28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	86	253	99	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	3	162	214	11	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	118	253	68	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	40	254	98	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	112	244	58	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

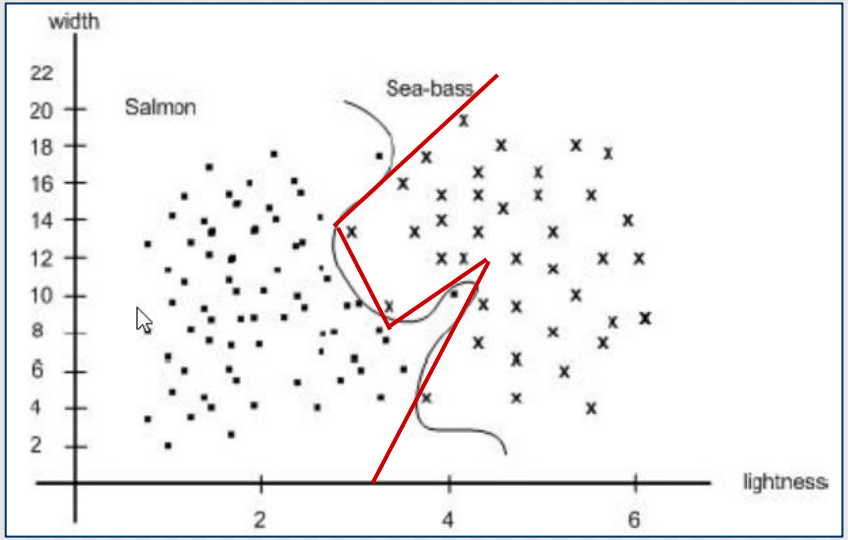


Sea bass vs. Salmon

https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm



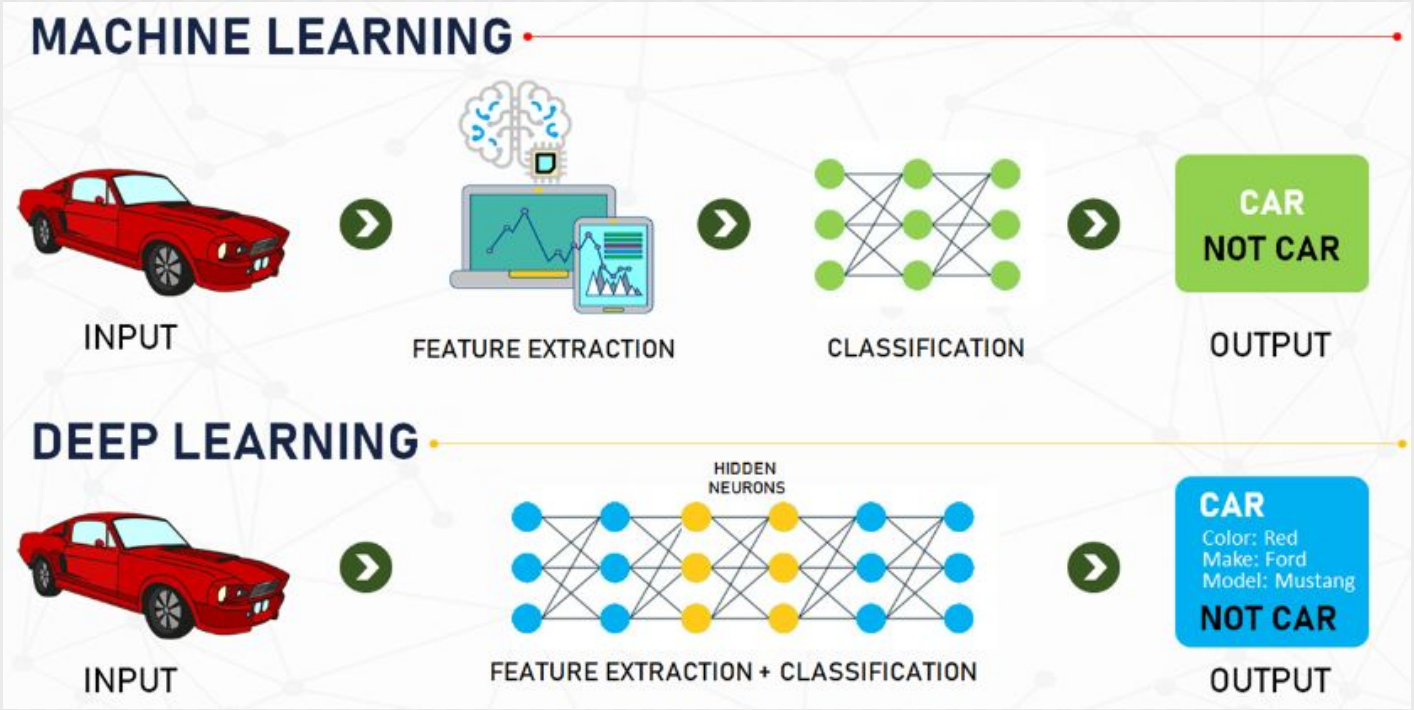
Linear Classifier



Piecewise Linear Classifier
vs. Non-Linear Classifier

Problem: We often want classification to be invariant to rotation, translation, scale, lighting.

Classical Machine Learning vs. Deep Learning



In this course we include Deep Learning into the Machine Learning topic.

First, a Quick Introduction to Computer Vision

Image Convolution and Filtering

Quiz 1. Some Image Processing Background.

King Mongkut's University of Technology **Name:** _____

Machine Learning **I.D. Number:** _____

Suthep Madarasmi, Ph.D.

Take Home Quiz 1 due Sun Jan 29, 2022 **Score:** _____ / 70

1. **Image Convolution.** Create your own Gaussian Kernel

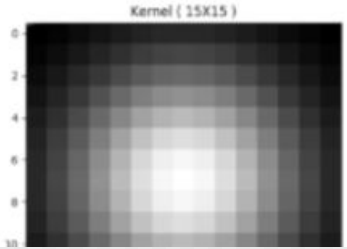
1.1. *10 points.* 1 hrs. Using Python, compute and print the matrix for a Gaussian kernel with σ = 3 using kernel size of 19x19 (we use width = ceiling (6*σ), but an odd no.). Print the kernel as output.

1.2. *10 points.* 0.5 hrs. Modify the OpenCV code shown in class to show the result of the convolution of your 19 x 19 Gaussian kernel using the Lenna image.

$$for\ i = -n..n, j = -n..n$$

$$g(i,j) = e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

$$G(i,j) = \frac{1}{}round(\frac{g(i,j)}{})$$

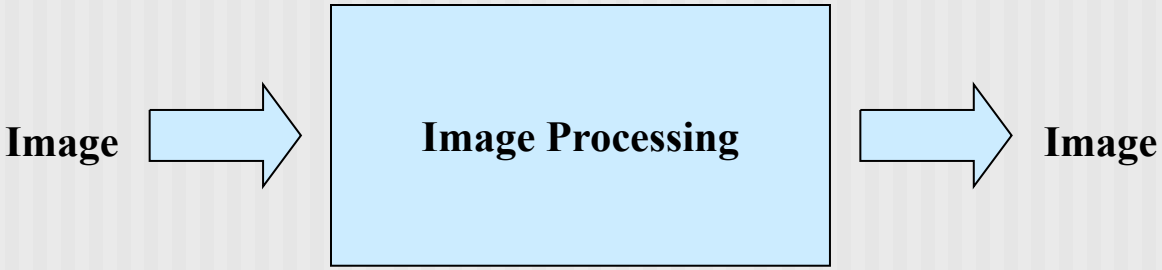


List of Topics

- Filters: Convolution Operation in OpenCV.
- Filter Examples: Blur, Edge detection, contrast.
- Noise removal: blurring and median filter
- Template Matching
- SIFT Features
- Use OpenCV

What is Image Processing?

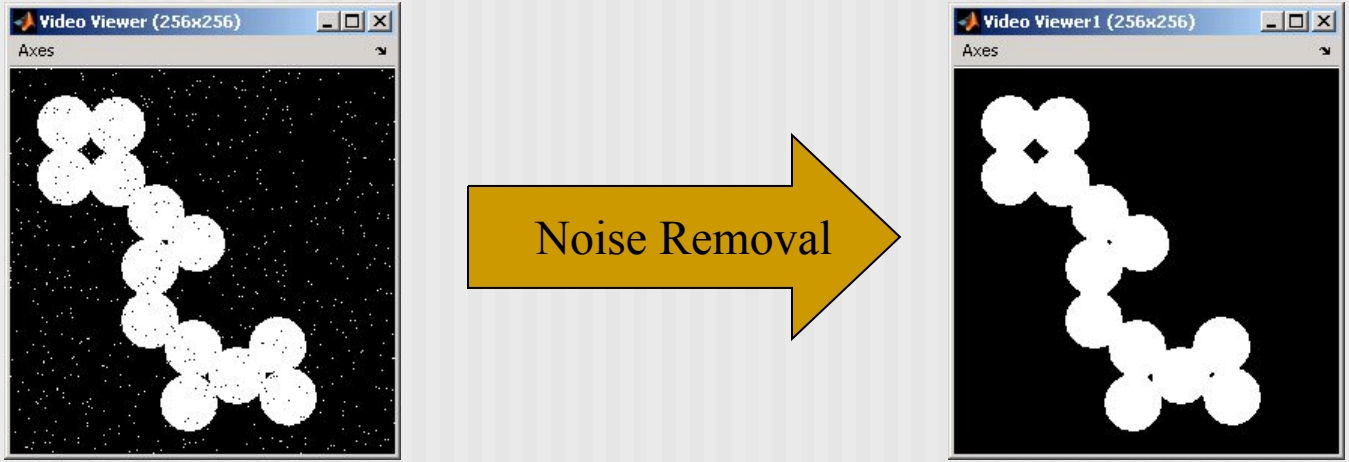
Image processing: Transformation of an input image into an output image with desired properties



Examples:

- **Image Enhancement**
- **Edge Detection**
- **Image Segmentation**

Examples of image processing

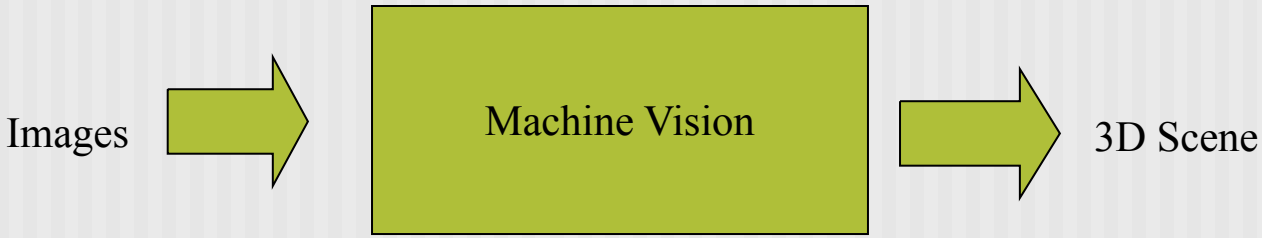


Examples of image processing

Contrast Adjustment

What is Computer Vision?

Machine Vision: Inferring the properties of the world from one or more images

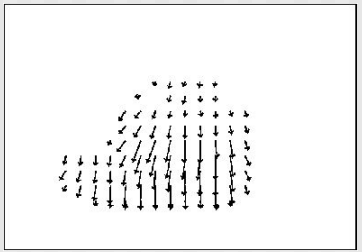


Examples:

- **Scene Description**
- **Shape Information**
- **Object Recognition**

Examples of Computer Vision

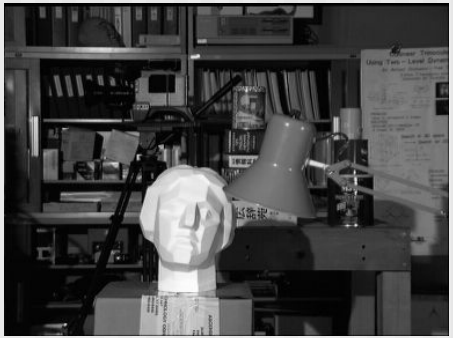
Motion Estimation

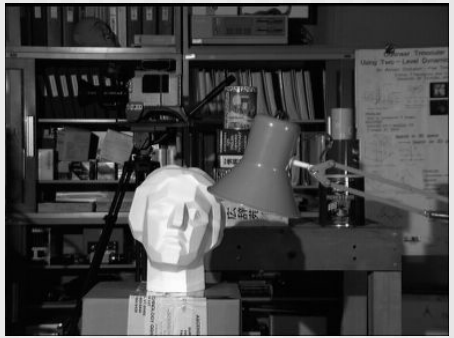


Optical Flow Field

Examples of Computer Vision

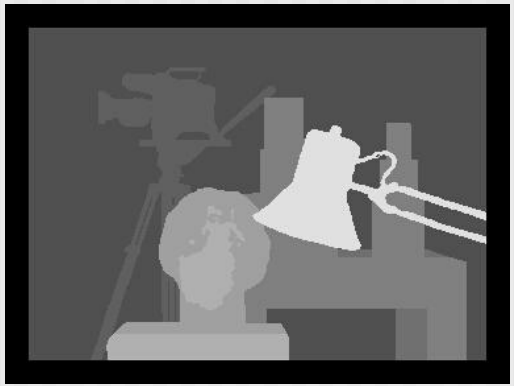
Left





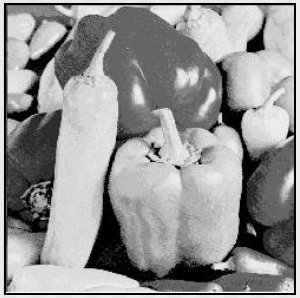
Right

Stereo Matching



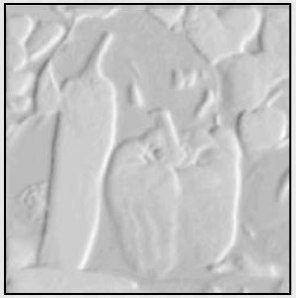
Depth

Examples of Computer Vision



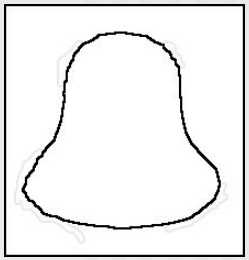
Image

Shape from Shading



Surface Orientation

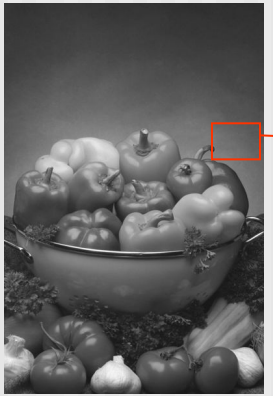
Examples of Computer Vision



Contour Search

Digital Image Representation

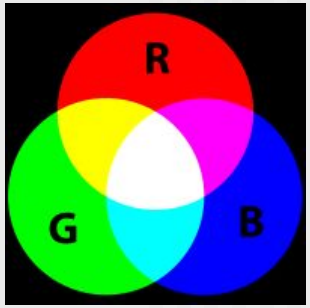
- Unit: pixel
- Each pixel has intensity value: (8 Bit) dark = 0 … bright = 255
- Spatial Resolution : number of pixels (width*height) per inch.
- Temporal Resolution: number of frames per second for video.

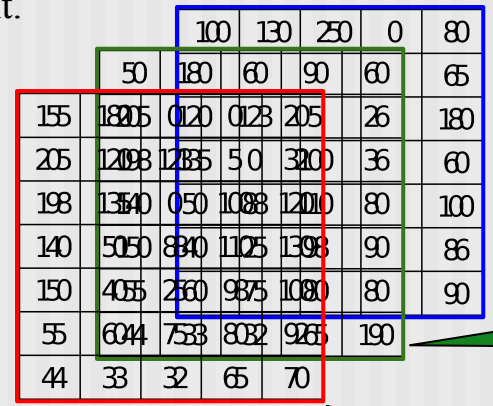


155	180	0	0	20
205	120	123	5	32
198	135	0	100	120
140	50	88	110	130
150	40	25	98	100
55	60	75	80	92
44	33	32	65	70

Color Images – RGB Channels

- Red, green, and blue light are added together in various ways to reproduce a broad array of colors. 0-1 is standard (0-255 in case of 1 byte encoding) where 0 is “no light” or black and 1 is maximum light of that channel.
- Gray = 0.2989 * R + 0.5870 * G + 0.1140 * B
- Alpha channel has 0-1 (0-255 for 1 byte) for opacity where lower is more transparent.





3 Intensities: 1 per color.

B: 0..255

G: 0..255

R: 0..255

1 Byte Per Pixel Color Image

Image Processing Example

- Convert to Gray. Gray = 0.2989 * R + 0.5870 * G + 0.1140 * B, due to human eye sensitivity to G, then R, then B.
- Threshold Image
- Color -> Channels -> Decompose
- View Color Curves and Transforms
- Histogram equalize and see histogram
- Show Alpha Channel: .png vs. .jpg
- The TIFF and PNG formats support the 8-bit alpha channel “RGBA”, whereas JPEGs have none. GIF supports a 1-bit channel: is this pixel fully transparent or not.

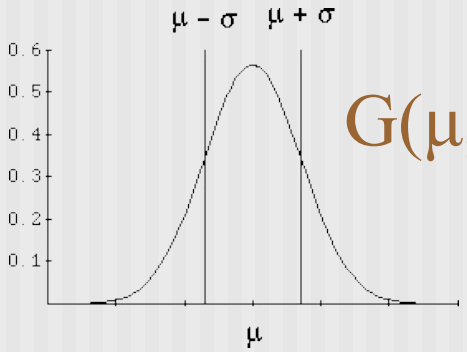
1. การวิเคราะห์ระบบ (System Analysis) และการออกแบบระบบ (System Design) ทางบริษัทฯจะทำการวิเคราะห์พร้อมทั้งทำการออกแบบระบบ เพื่อเสนอแก่ท่านจนเป็นที่พอใจ หลังจากนั้นจึงเริ่มขั้นตอนของงานการพัฒนาระบบ
2. การตั้งรหัสสินค้าสำหรับสินค้าประเภทต่างๆ เช่น วัตถุดิบ (Raw Materials) สินค้าที่อยู่ระหว่างการผลิต (WIP) และสินค้าที่ผลิตเสร็จสมบูรณ์แล้ว (Finished Goods)

คุณสมบัติพิเศษ: รองรับหน่วยของสินค้า ได้ถึง 2 หน่วย (หน่วยหลัก/หน่วยบรรจุภัณฑ์) สำหรับสินค้าทุกๆชนิด, สนับสนุนการแปลงระหว่างหน่วยต่างๆ, การใช้หมวดสินค้าแทนแต่ละรหัสสินค้า โดยไม่ต้องระบุเลขที่ล็อตและเกรด และรหัสสินค้าแบบเต็ม ซึ่งระบุ สี/แบบ/ เกรดและเลขที่ล็อต และหาวิธีที่จะตั้งรหัสสินค้าเพื่อให้เหมาะสมกับงานของลูกค้าแต่ละรายให้มากที่สุด

Gaussian Additive Noise

- For example, an ideal white paper

220	220	220	220	220
220	220	220	220	220
220	220	220	220	220
220	220	220	220	220

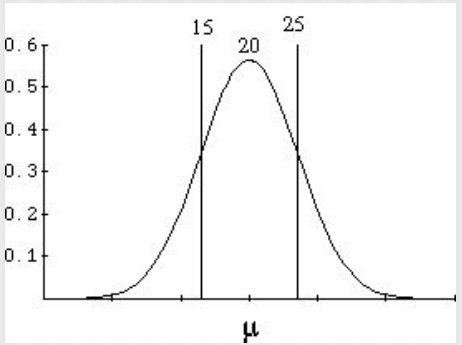


$G(\mu,\sigma)$

- Scanned white paper will spread around 220 + 20:

242	239	240	220	238
242	240	240	241	245
247	237	230	235	238
244	240	237	232	243

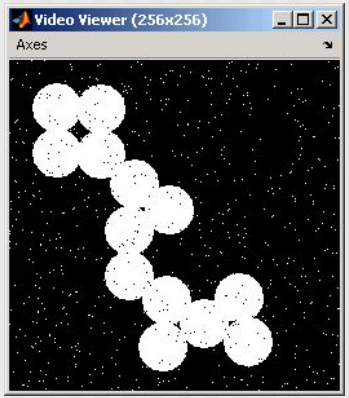
$G(\mu=20,\sigma = 5)$

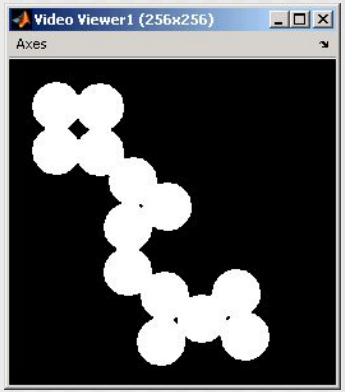


The Convolution Filter

- Gaussian Noise
  - Use Gaussian Filter
  - Can also use the Mean (Average) Filter

- Salt/Pepper Noise
  - Use Median Filter





Noise Removal Techniques

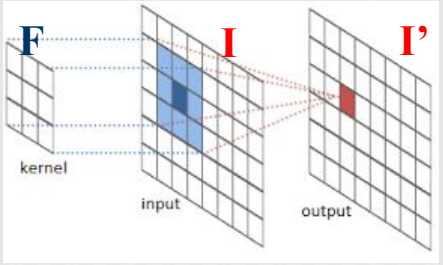
- Gaussian Filter (convolution). For Noise model of Gaussian Noise where $N = G(\mu, \sigma)$. Convolution by Gaussian Kernel.

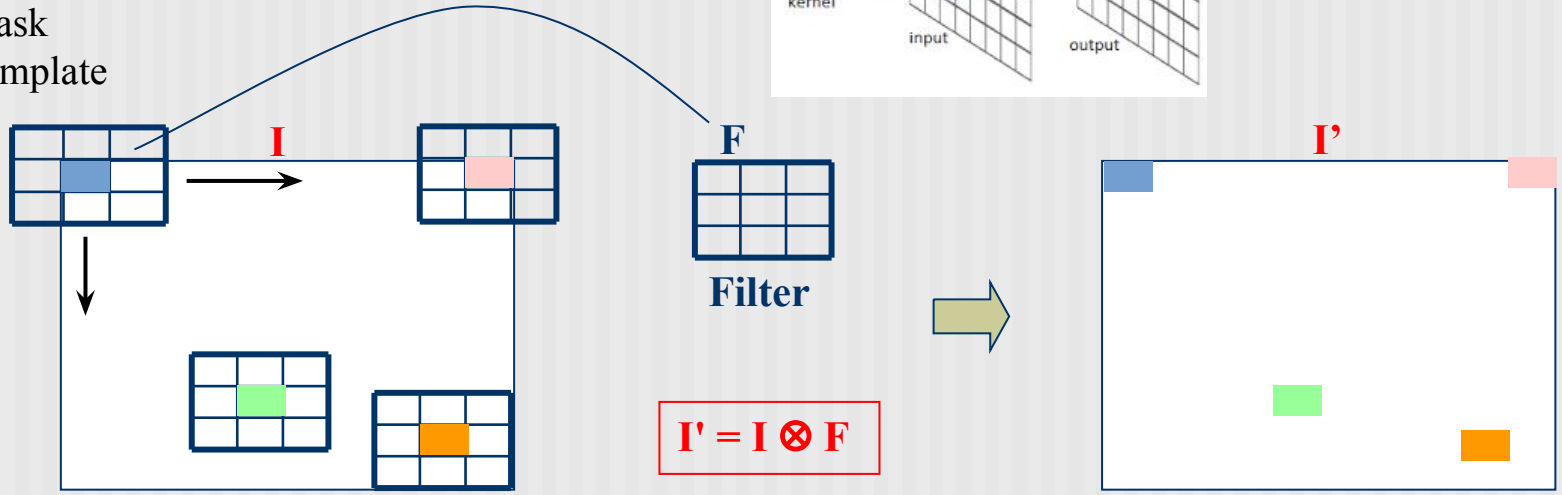
- Median Filter. For Salt/Pepper Noise Model. Sort pixels in region by intensity. Pick the median intensity as output.

The Convolution Filter

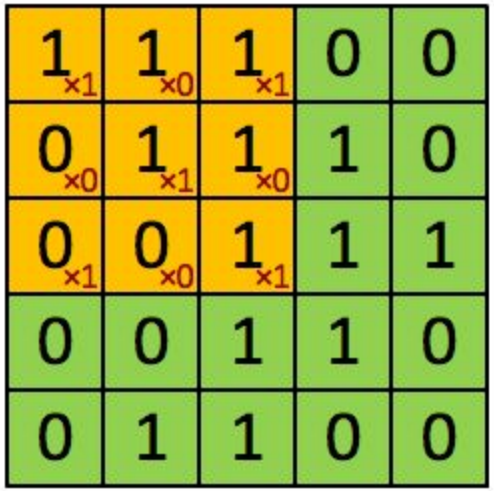
Synonyms for Convolution Matrix

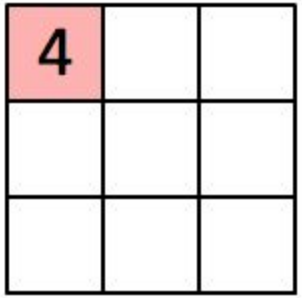
- Kernel
- Filter
- Mask
- Template





Convolution





Filter:

1	0	1
0	1	0
1	0	1

Average or Mean Convolution Filter

Average = (50*1+50*1+45*1+50*1+50*1+45*1+49*1+49*1+52*1) / 9

(50*1/9)+(50*1/9)+(45*1/9)+
(50*1/9)+(50*1/9)+(45*1/9)+
(49*1/9)+(49*1/9)+(52*1/9)

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

50	50	45	47	52	54
50	50	45	47	52	54
49	49	52	50	50	51
51	51	52	51	45	46
50	50	47	54	47	52

Mirror

Input Image

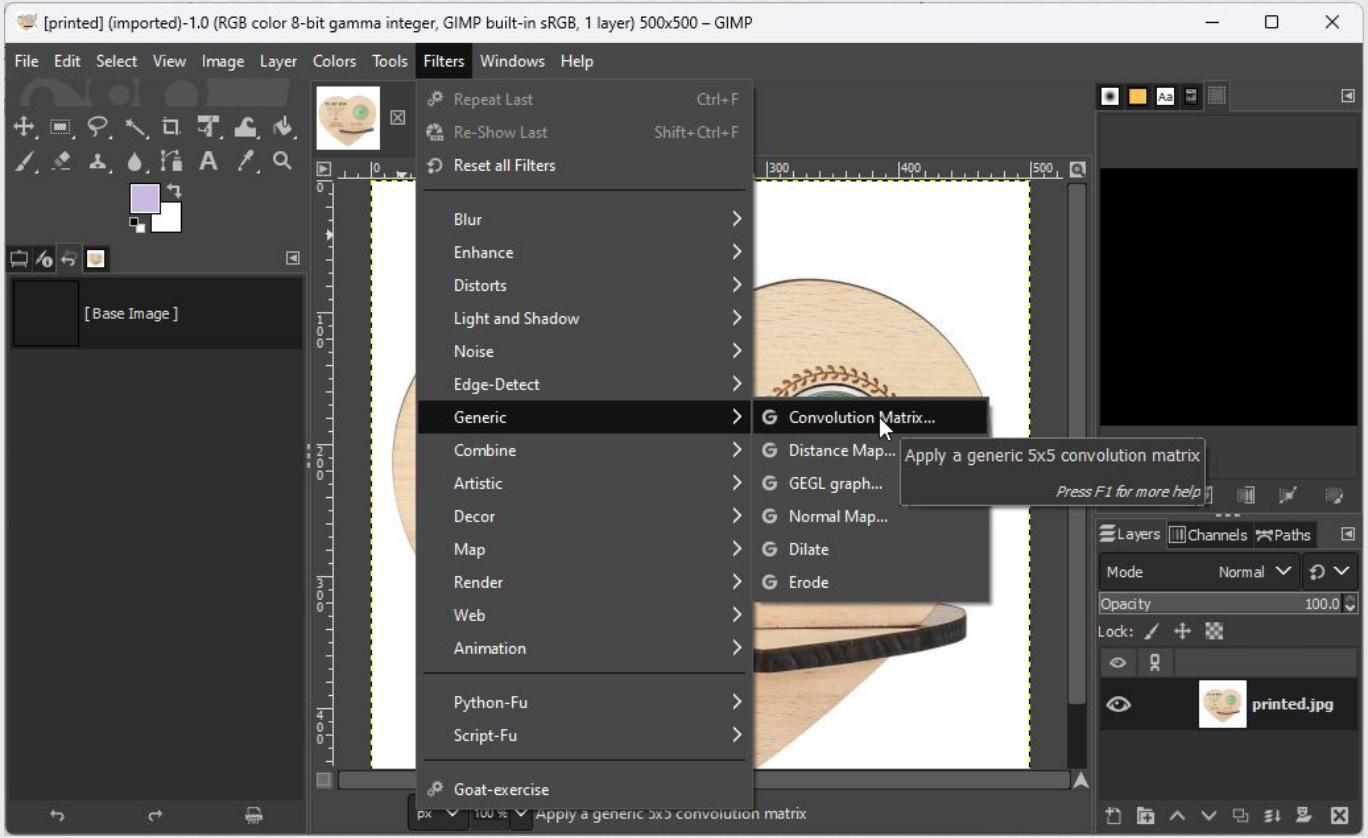
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3
Mean Filter

49	48	49	51	
50	50	49	50	
50	50	50	50	

Output Image

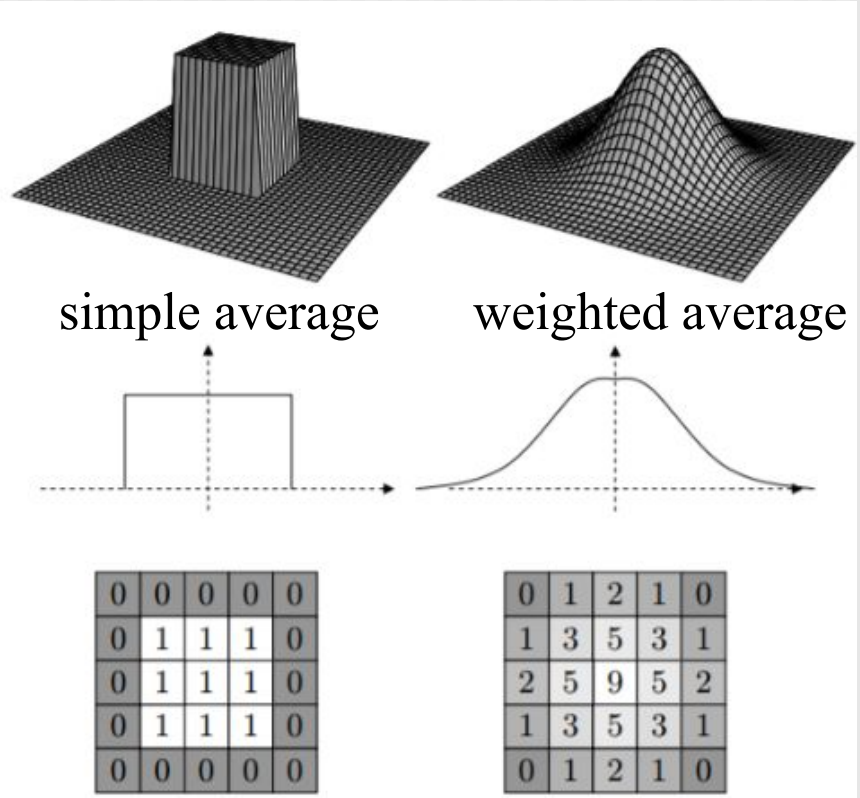
GIMP: Mean Convolution and Median Filter



GIMP:
GNU
Image
Manipulation
Program

GNU:
GNU's
NOT
UNIX

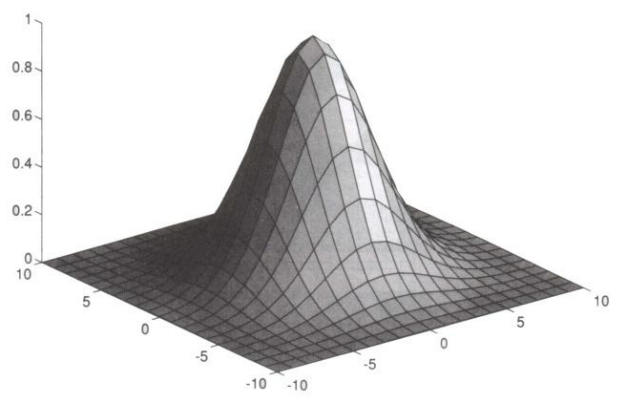
The Mean Filter vs. The Gaussian Filter

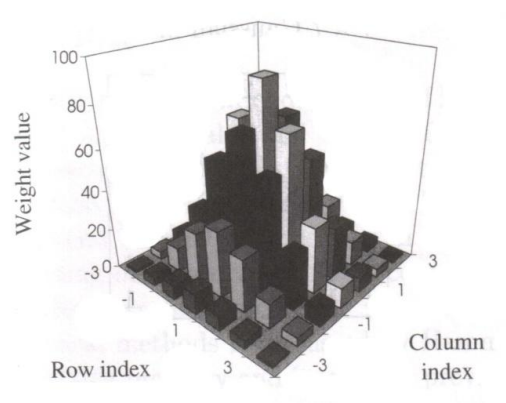


0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

The Gaussian Filter (Convolution)





$$C = \sum_{i=-n}^{n} \sum_{j=-n}^{n} G(i,j)$$

$G(i,j)$

$$\frac{1}{C}$$

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

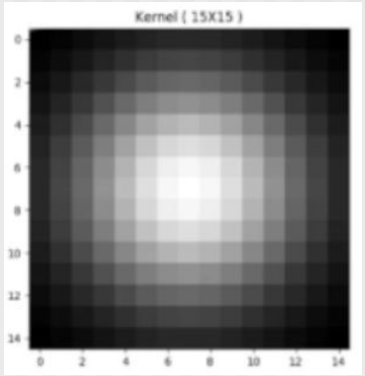
The discrete Gaussian Filter

$$g(i,j) = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \; for \; i = -n..n, j = -n..n$$

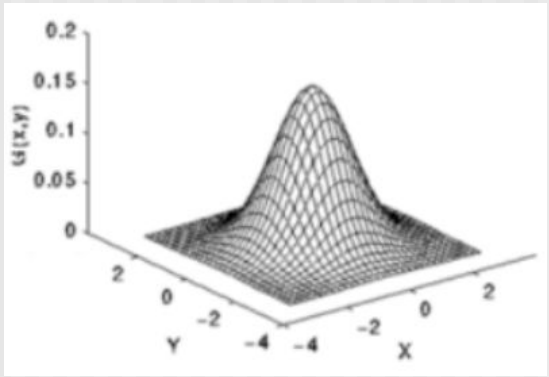
$$G(i,j) = \frac{1}{C} round(\frac{g(i,j)}{g(n,n)})$$ divide each by min or g(n,n) to get integers

$$C = \sum_{i=-n}^{n} \sum_{j=-n}^{n} G(i,j)$$

C is normalizing constant.



$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Pseudocode

```
for i = -n .. n
  for j = -n ..n
    i2 = i+n
    j2 = j+n
    g[i2, j2] = exp(-(i^2 + j^2)/(2*sigma^2))

C = 0
for i = 0 .. 2*n
  for j = 0 .. 2*n
    G[i,j] = round(g[i,j]/g[i2, j2])
    C += G[i,j]
```

Find values for 13 x 13 matrix to create Gaussian filter with σ = 2.

Your result G(i,j) should look something like (here max scaled to 255):

$\frac{1}{C}$

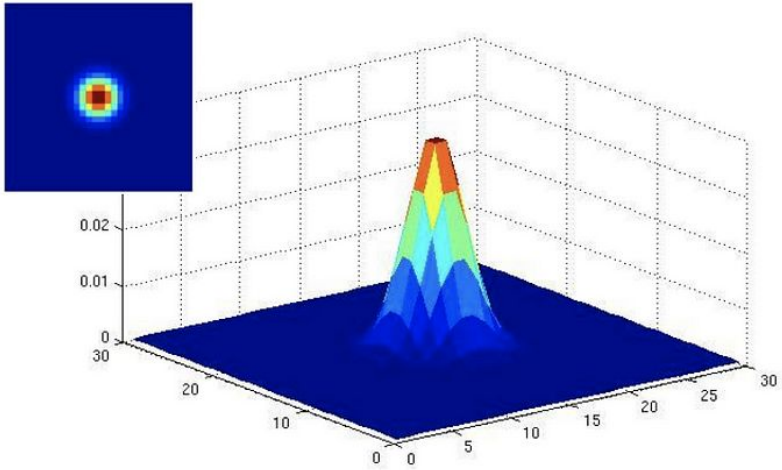
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	1	3	6	9	11	9	6	3	1	0	0
0	1	4	11	20	30	34	30	20	11	4	1	0
0	3	11	26	50	73	82	73	50	26	11	3	0
1	6	20	50	93	136	154	136	93	50	20	6	1
2	9	30	73	136	198	225	198	136	73	30	9	2
2	11	34	82	154	225	255	225	154	82	34	11	2
2	9	30	73	136	198	225	198	136	73	30	9	2
1	6	20	50	93	136	154	136	93	50	20	6	1
0	3	11	26	50	73	82	73	50	26	11	3	0
0	1	4	11	20	30	34	30	20	11	4	1	0
0	0	1	3	6	9	11	9	6	3	1	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0

$$C = \sum_{i=-n}^{n} \sum_{j=-n}^{n} G(i,j)$$

Mask width = ceiling (6*σ)

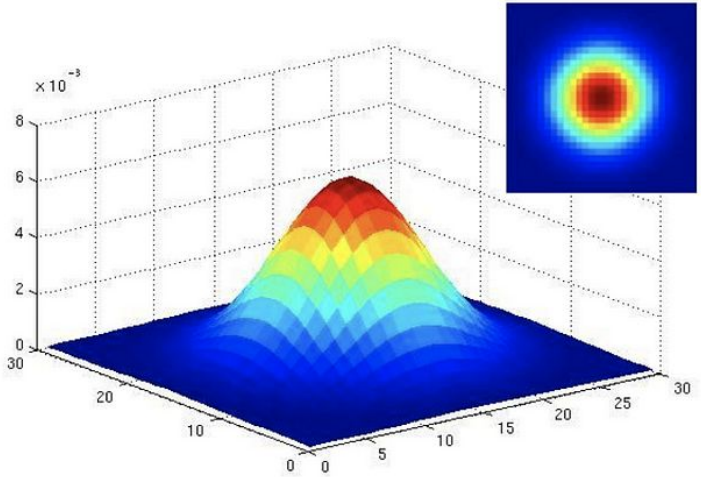
Why?

Gaussian Filters of varying σ.



Can reduce the kernel size to keep only non-zero weights.

$\sigma = 2$ with 30×30 kernel



Must keep values from -3σ to +3σ.

$\sigma = 5$ with 30×30 kernel

Median Filter: Rids Salt/Pepper Noise

- 20, 21, 32, 23, 17, 19 , 20
- Order by value

17
19
20
20
21
23
32

Use the middle

- Use 20

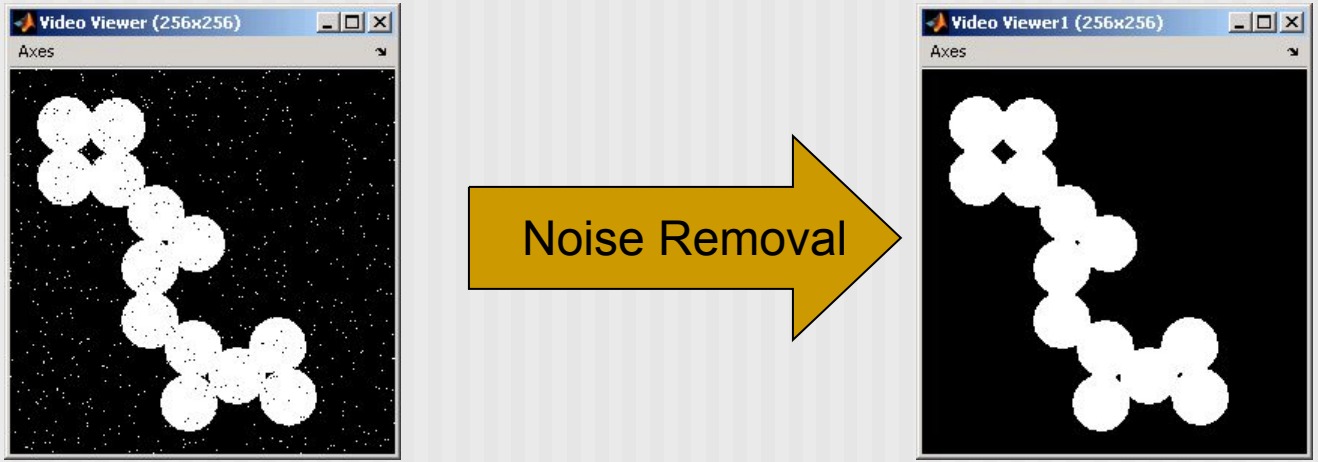
- 20, 22, 32, 23, 17, 19 , 20,24
- Order by value

17
19
20
20
22
23
24
32

Average the middle
(20+22)/2 = 21

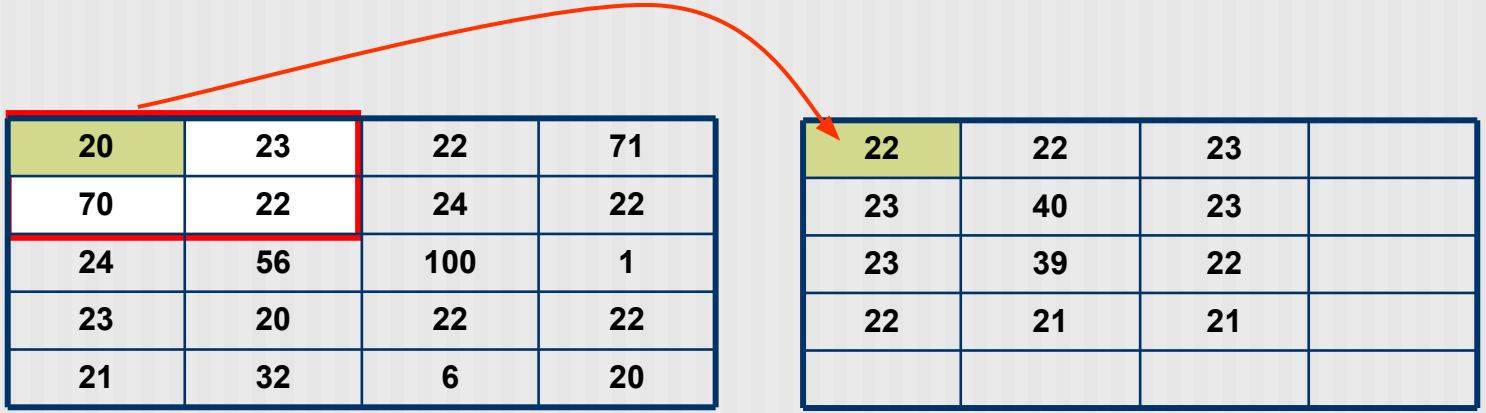
- Use 21

Salt/Pepper Noise Removal



Example of Median Filter

- **2 x 2 Median Filter (most filters are odd numbered)**
- **Notice Salt "70, 71, 56, 100" removed.**
  Pepper "1, 6" removed.

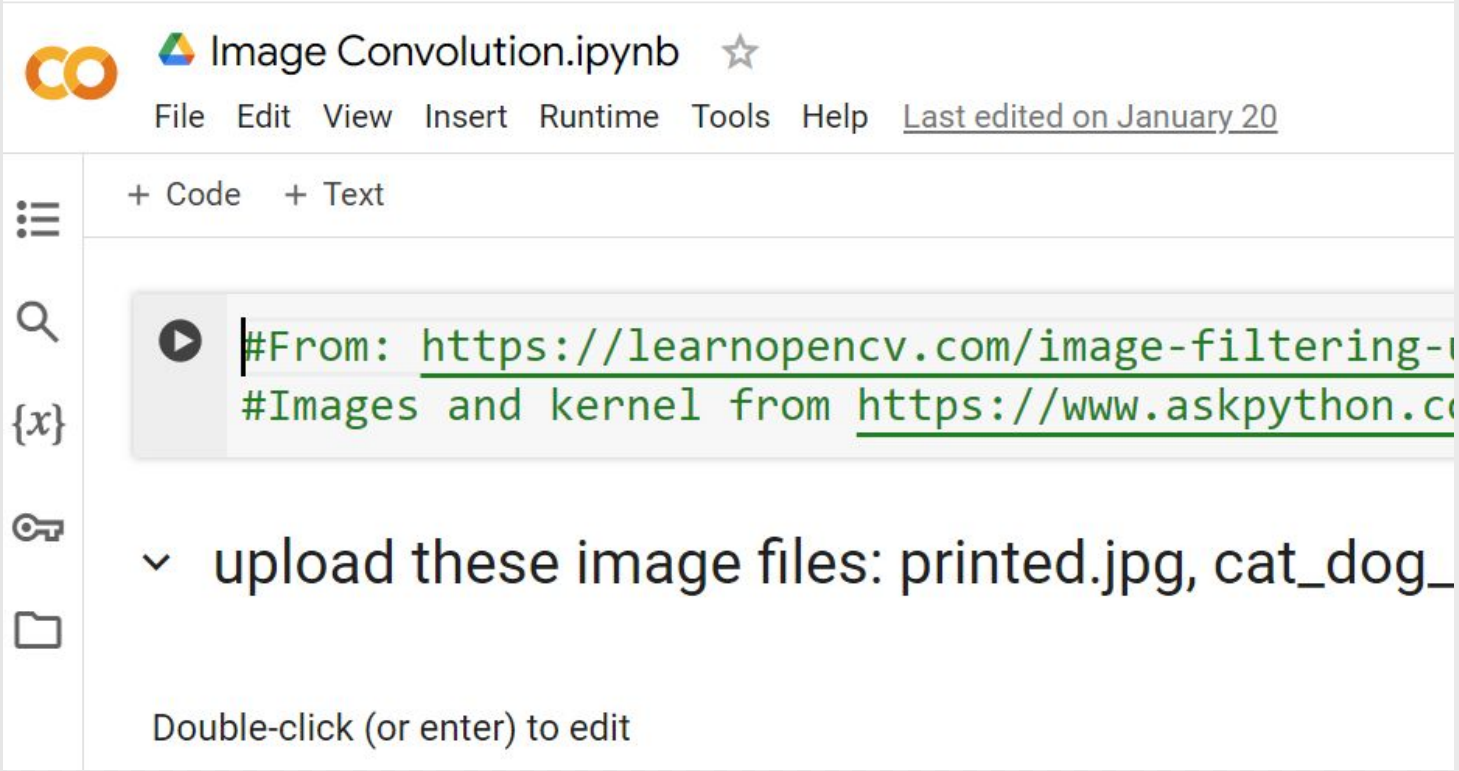


20	23	22	71
70	22	24	22
24	56	100	1
23	20	22	22
21	32	6	20

22	22	23	
23	40	23	
23	39	22	
22	21	21	

Step 1. Remove Image Noise (Gaussian and Salt/Pepper)

Image Convolutional Colab



Quiz 1
Problem 1

1. **Image Convolution.** Create your own Gaussian Kernel

1.1. *10 points.* 1 hrs. Using Python, compute and print the matrix for a Gaussian kernel with σ = 3 using kernel size of 19x19 (we use width = ceiling (6*σ), but an odd no.). Print the kernel as output.

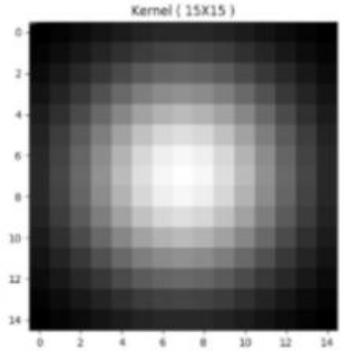
1.2. *10 points.* 0.5 hrs. Modify the OpenCV code shown in class to show the result of the convolution of your 19 x 19 Gaussian kernel using the Lenna image.

$$for\ i = -n..n, j = -n..n$$

$$g(i,j) = e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

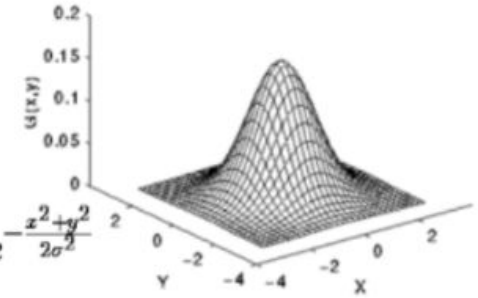
$$G(i,j) = \frac{1}{C} round(\frac{g(i,j)}{g(n,n)})$$

divide each by min or g(n,n) to get integers



$$C = \sum_{i=-n}^{n} \sum_{j=-n}^{n} g(i,j)$$

C is normalizing constant.



$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Uses of Image Convolution

1. Blur to remove noise
2. Edge Detection
3. Feature detection: lines, corners, endpoints, dots, holes, …

vertical edge:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

horizontal edge:

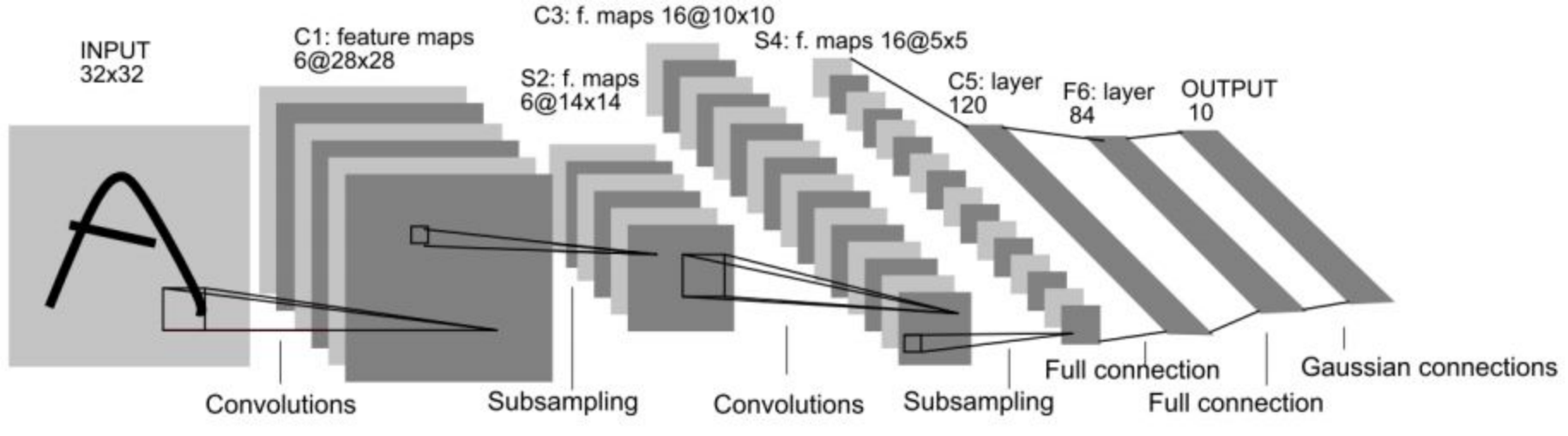
$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

100	100	100	200	200	200
100	100	100	200	200	200
100	100	100	200	200	200
200	200	200	100	100	100
200	200	200	100	100	100
200	200	200	100	100	100

CNNs (Convolutional Neural Networks)

2 stages:

1. Learn ideal features (convolutional kernel values) from training data.
2. Flatten 2D to 1D and use Multi-layer Perceptron to classify.

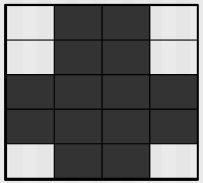


Introduction to Computer Vision

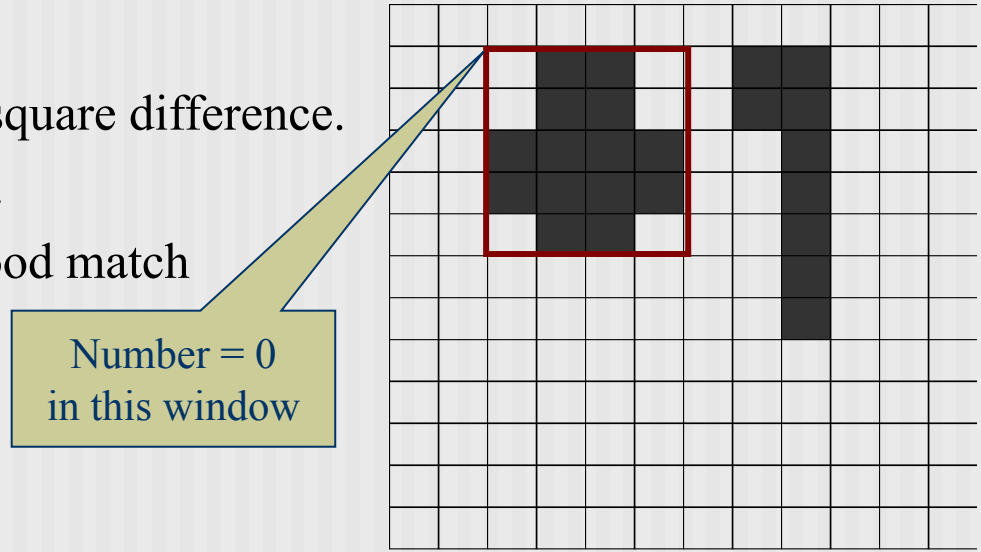
Template Matching

Template matching to image

- Given this template:



- Place template, do sum square difference.
- Move throughout image.
- Low number means a good match



Min Template Match = Max Convolution

$$\underset{\text{minimize}}{SSD} = \sum_k \sum_l \left(f(k,l) - h(i+k, j+l)\right)^2 \quad \text{Sum of Squares Difference}$$

$$SSD = \sum_k \sum_l \left(f(k,l)^2 - 2h(i+k, j+l) f(k,l) + h(i+k, j+l)^2\right)$$

$$\underset{\text{minimize}}{SSD} = \sum_k \sum_l \left(-2h(i+k, j+l) f(k,l)\right)$$

$$\underset{\text{maximize}}{SSD} = \sum_k \sum_l \left(2h(i+k, j+l) f(k,l)\right)$$

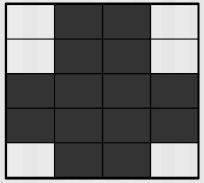
f(k,l) is template image. h(i, j) is input image.

Template matching is done at h(i, j).

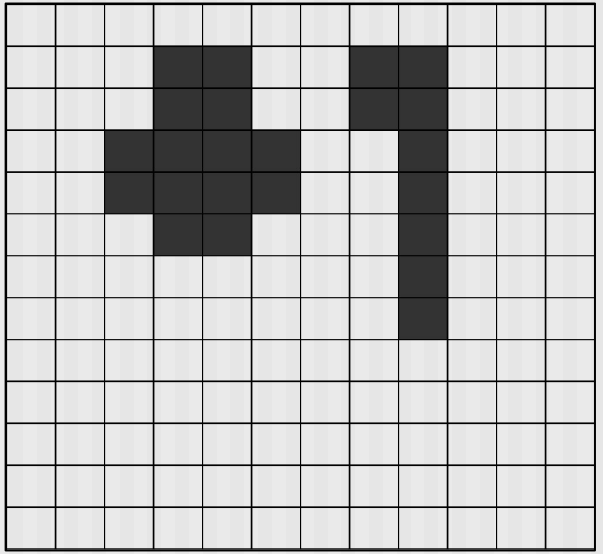
Range of k is template height, range of l is image width.

Template matching pseudocode

T(k, l):



I(i,j):



```
for i = 1 .. 13
for j = 1.. 12
  R(i,j) = 0
  for k = 1..4
  for l = 1..5
     R(i,j) += (T(k, l) - I(i+k, j+l))^2
```

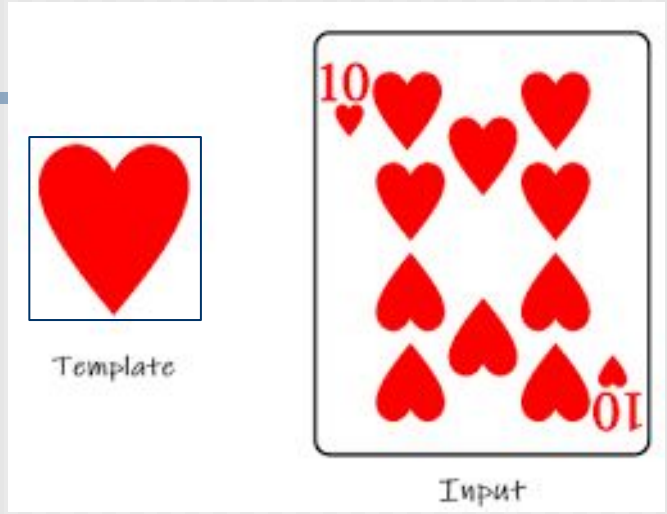
T(k,l) is template image. I(i, j) is input image. R is result.

Here, template matching is centered at I(i, j) to get R(i, j).

Range of k is template height, range of l is template width.

Template Matching

https://theailearner.com/2020/12/12/template-matching-using-opencv/



Use error measures for corresponding pixels:

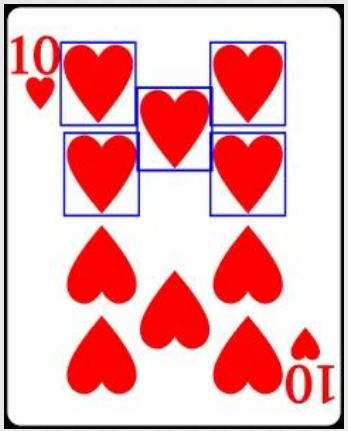
- |L1 Norm|. Sum absolute error.
- ||L2 Norm||. Sum squared error.

Issues of scale, rotation, contrast. Handles translation.

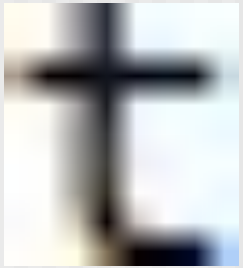
Should scale template first. Can rotate template a bit.
Color normalize of template and image area 0-1.

Could improve by allowing

perspective distortions.



Quiz 1. Problem 2.



```
import numpy as np
import cv2 as cv2
import matplotlib.pyplot as plt


# In[9]:


def myImshow(title, img):
    """
    function to make windows display work in jupyter notebook
    - shows image in a separate window,
    - waits for any key to close the window.

    """

    cv2.startWindowThread()
    cv2.imshow(title, img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()


# In[10]:


path = "D:/data/Dropbox/ML/"
#RGB images in BGR order in penCV
img1 = cv2.imread(path+'box.png',cv2.IMREAD_GRAYSCALE)          # queryImage
# Print error message if image is null
if img1 is None:
    print('Could not read query image')
else:
    print("Query Image read success...")

img2 = cv2.imread(path+'box_in_scene.png',cv2.IMREAD_GRAYSCALE) # targetImage
# Print error message if image is null
```

2. *10 points.* 1.5 hrs. **Template Matching.** Search for the 't' using "t_character.png" as template in the text image "text_image.png". Use a bounding box to mark where 't' were found. Use the Euclidean norm. You may use OpenCV to only read and write the image, but not to call the template matching routine.

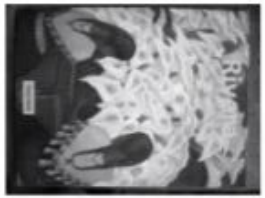
Scale-Invariant Feature Transform
SIFT Matching

https://www.youtube.com/watch?v=PU6BKIh1y4o

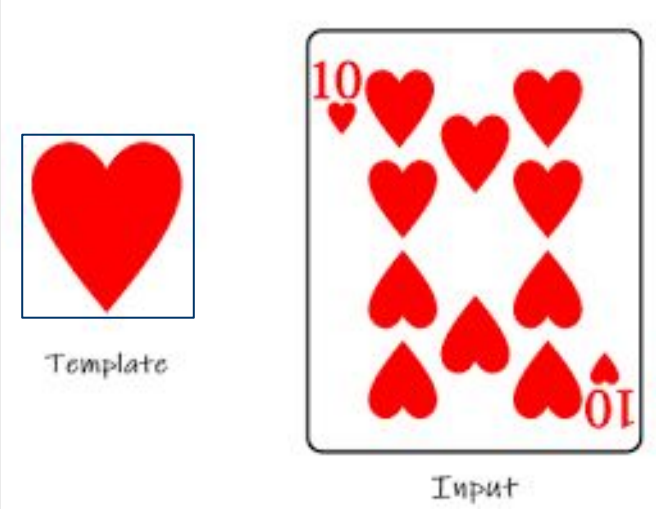
Test Image:

http://www.computing.dundee.ac.uk/staff/jessehoey/teaching/vision/project1.html

Training Images

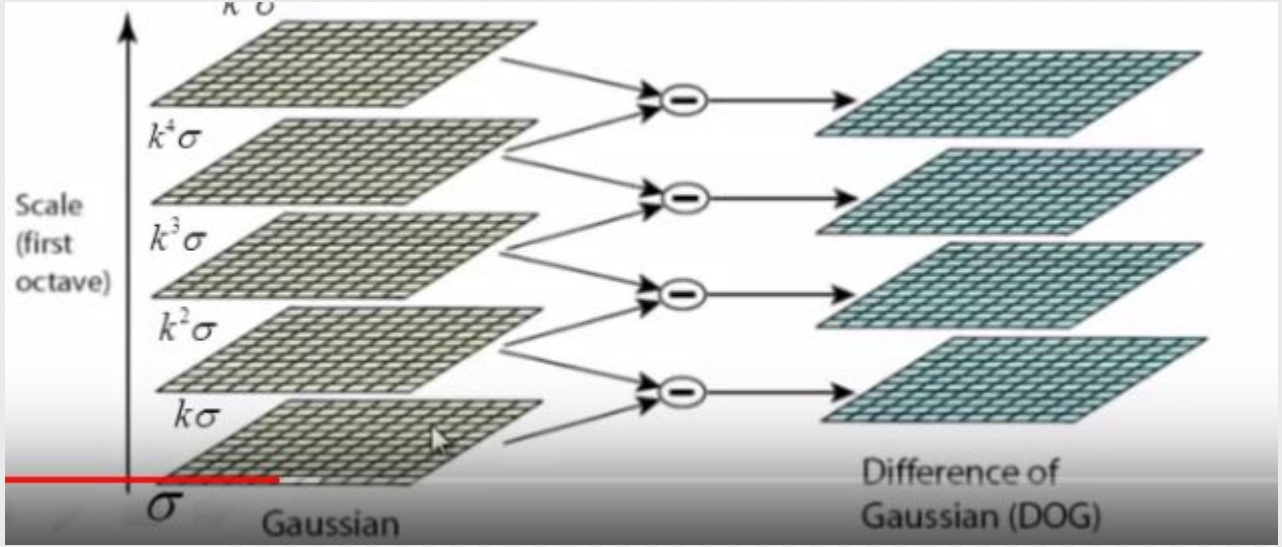


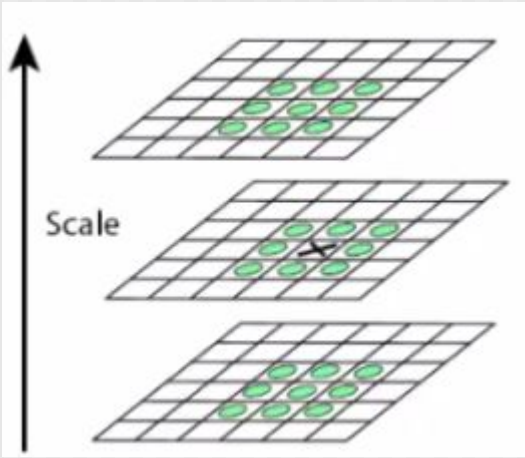
Scale-Invariant Feature Transform
SIFT Matching



Selecting a SIFT Feature

$$G(x,y,k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2+y^2)/2k^2\sigma^2}$$





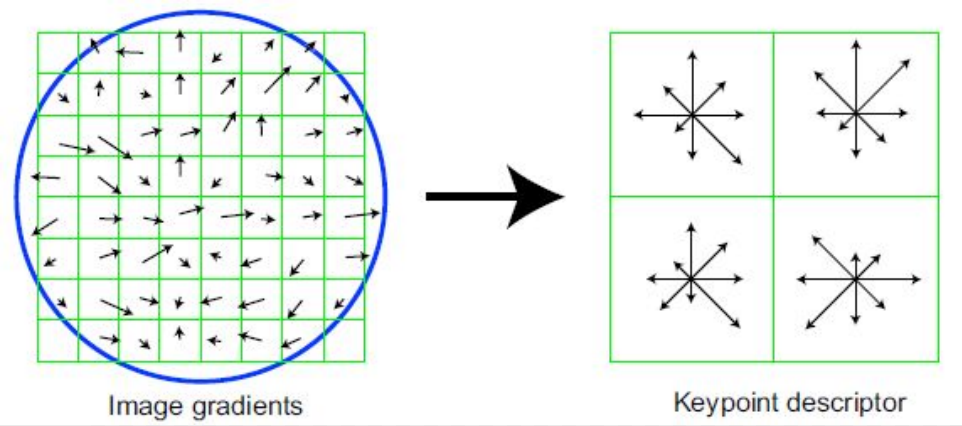
$k = \sqrt{2}$

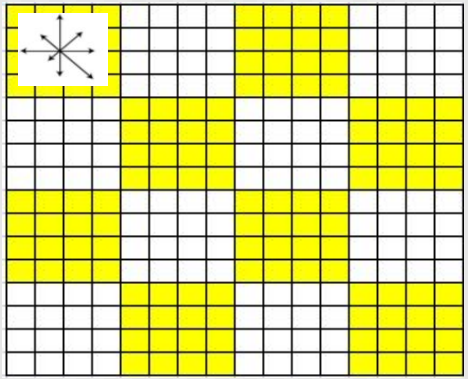
Regarding different parameters, the paper gives some empirical data which can be summarized as, number of octaves = 4, number of scale levels = 5, initial $\sigma = 1.6$, $k = \sqrt{2}$ etc as optimal values.

The SIFT Feature

SIFT is a rich feature descriptor to address problem of matching across frames with changes in scale and orientation. Each SIFT keypoint feature i has:

- points where local maxima in scale space of LoG filter (via DoG): $Lxx(\sigma) + Lyy(\sigma)$
- f_i(x, y, scale, angle) is position, scale, and main gradient angle of the keypoint i.
- d_i(x1, x2, …, x128) descriptor of keypoint i. For 16 x 16 pixel area around feature center (x, y), for each 4 x 4 region, find gradient orientation count for 8 key orientations (0, 45, 90, 135, 180, 225, 270, 315 degrees), each weighted by gradient magnitude. 16 x 8 = 128.



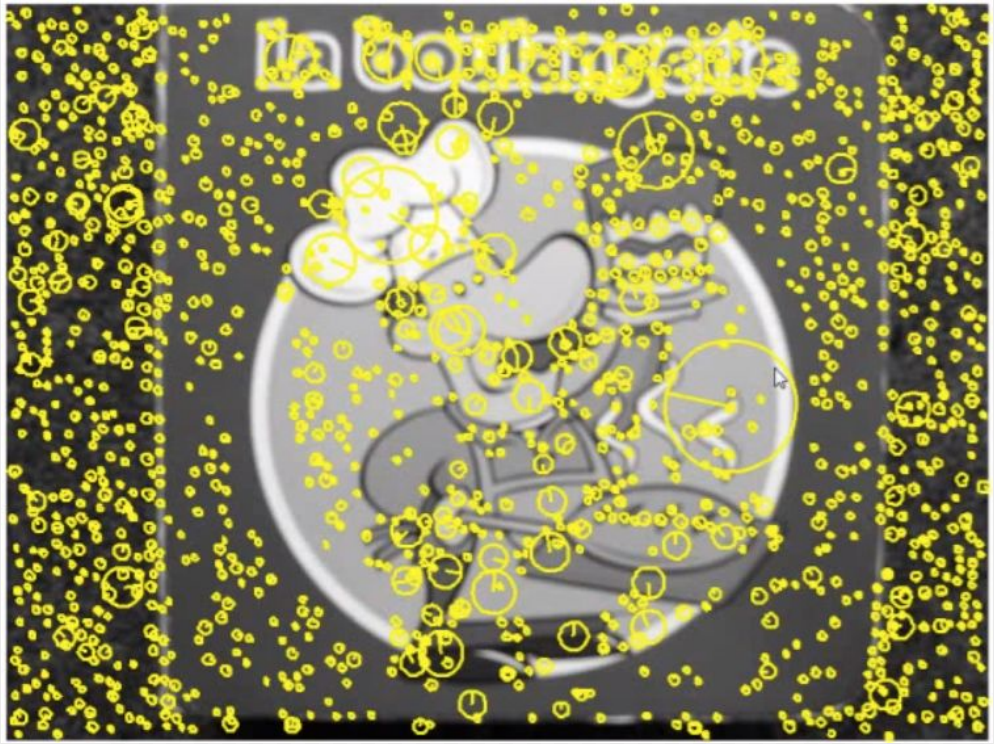


Shows 8x8 nbr.
Actual: 16x16

Shows 2x2 descriptor array of 8 orientations.
Actual: 4x4 descriptor array of 8 orientations.

The SIFT Feature

SIFT Features extracted from a training image. Total: 1,815 features.



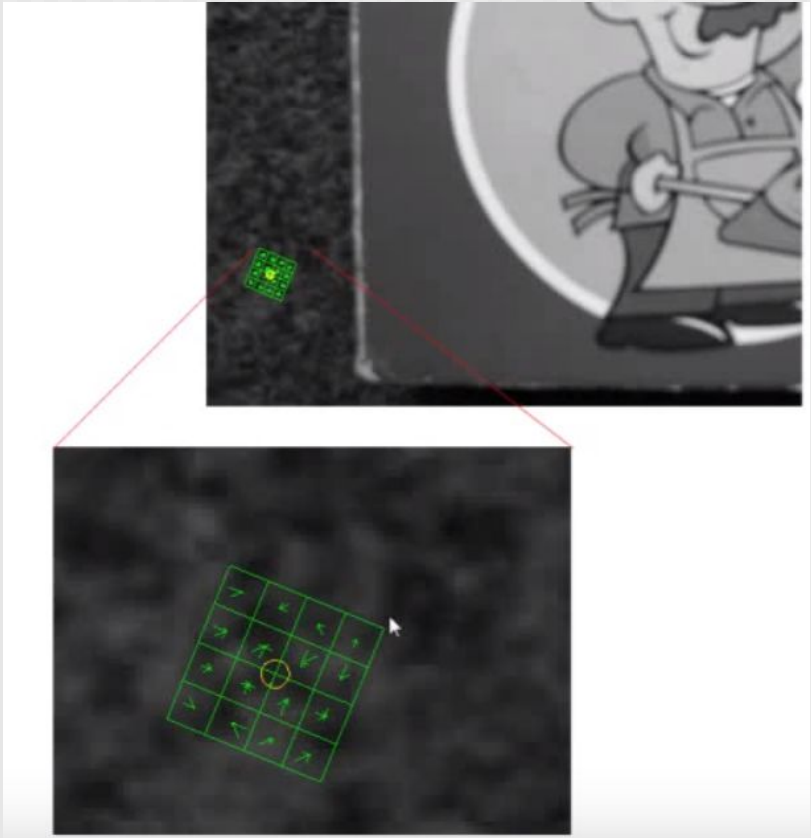
Center of circle is the feature position (x, y).

The size of circle indicates scale where larger size means its a bigger “blob” feature.

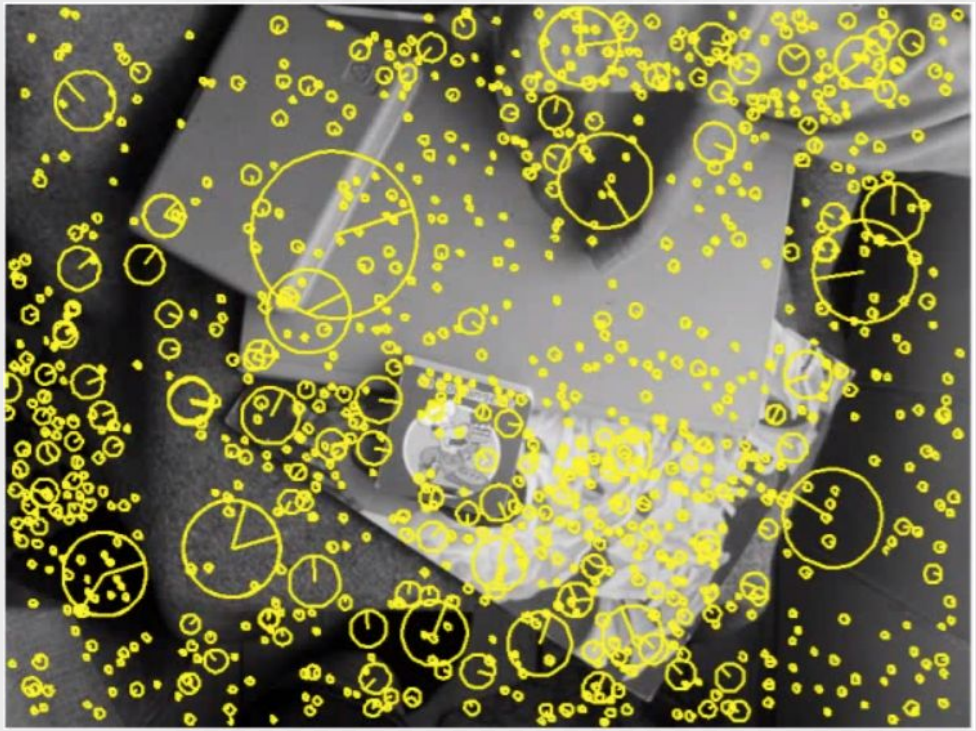
Radius of circle indicates main gradient direction (normal to edge) of the feature.

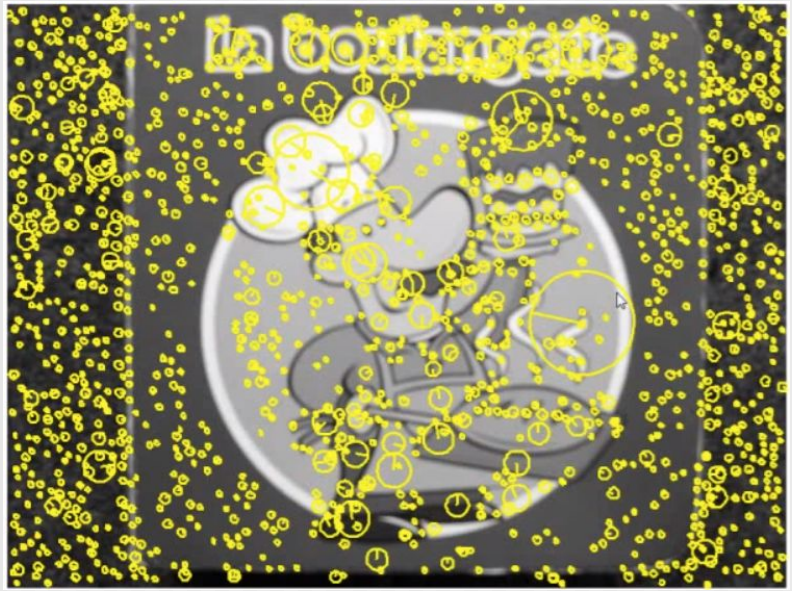
Descriptor Example

Descriptors are extracted relative to the main orientation of the feature and at the corresponding scale.



Features in Test Image

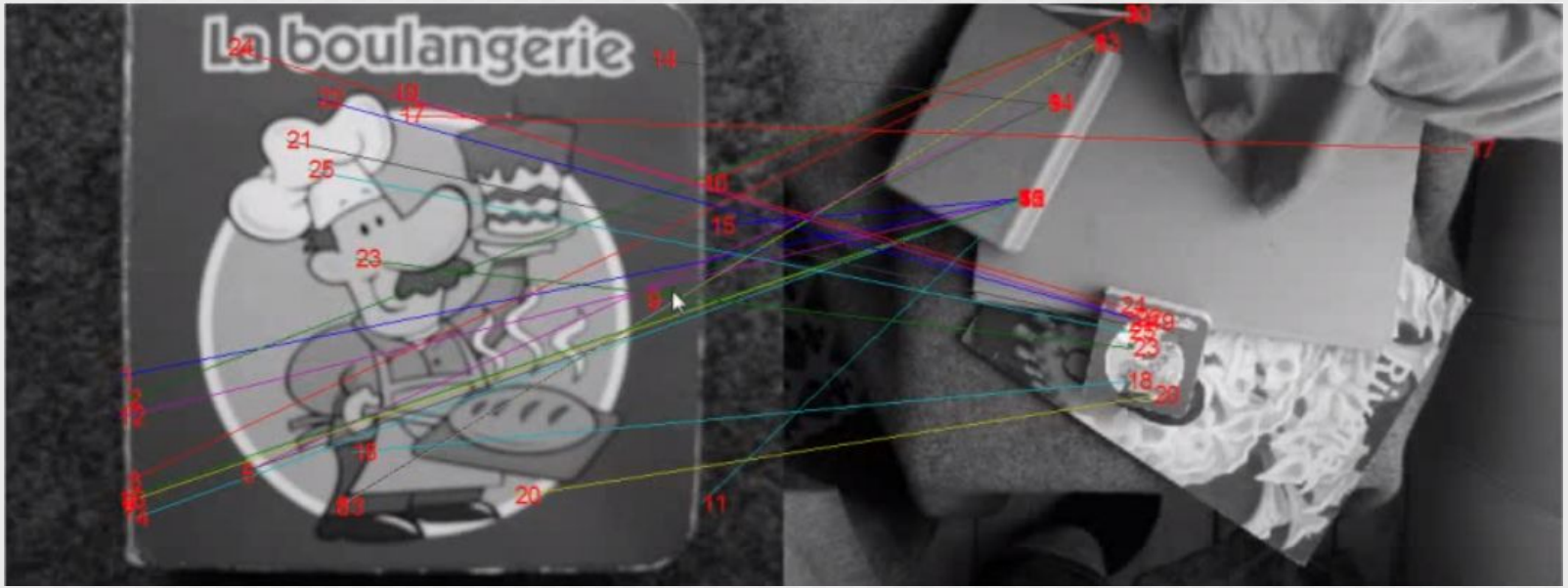




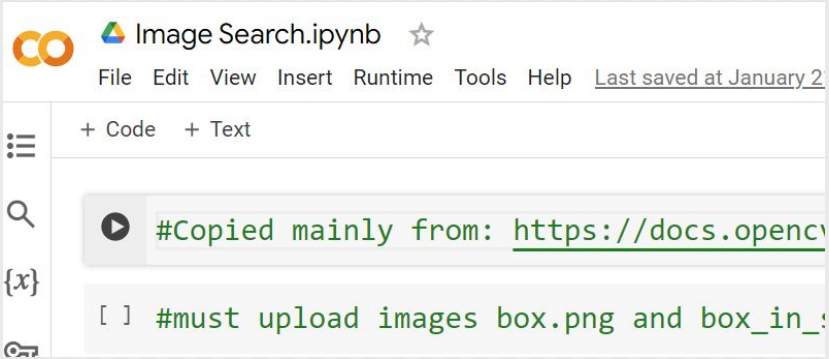
Test image has 1,108 features.

Match Output

Match features in query image with target image. It will take each of the N features in query image and match with M features in target, creating N*M possible matches. Each paired match (n, m) will have a score which is the Euclidean distance (template match) among all the attributes of the 2 features being matched. N = 25 in the test image below.



Quiz 1. Problem 5.

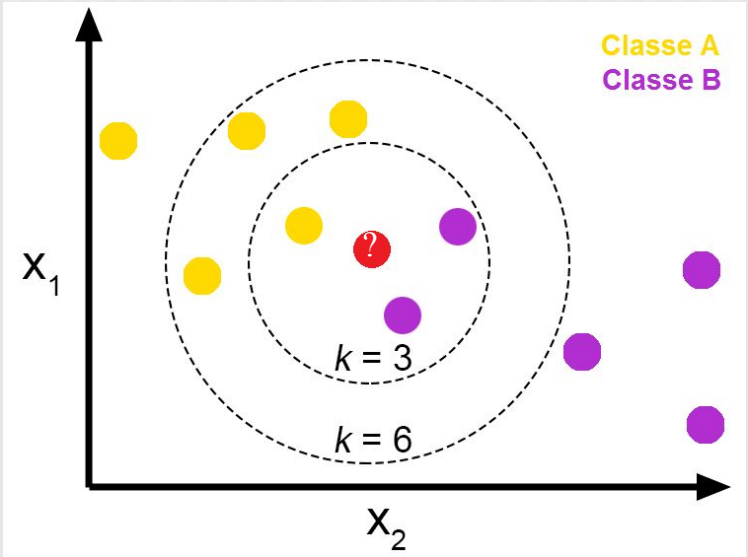


5. *10 points.* 0.5 hrs. **Image Matching with KNN.** Create a query image of an object you photographed yourself. Then photograph that same object but in a different environment with some other background objects as a target image. Try the provided image search colab to search for the query image. Show the "good matches" that includes the query and target images with SIFT features and matching lines as done in the colab.

The K Nearest Neighbor (KNN) Classifier

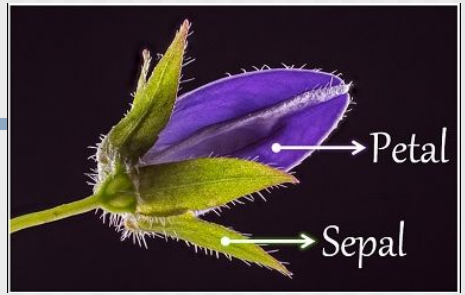
KNN Classifiers

Machine Learning Based such as K-Nearest Neighbors (KNN). To label an unknown point with 2 features (x_1, x_2), look at the class of nearest data points. First must decide on K = ?. If K = 3, **Class B** wins with count **2** vs. **1**. If K = 6, **Class A** wins with count **4** vs. **2**.



https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

KNN Algorithm Example



Labelled Dataset using 2 features:

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

When K = 3, Setosa gets all 3 votes.
Also when K = 5, Setosa is chosen:

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

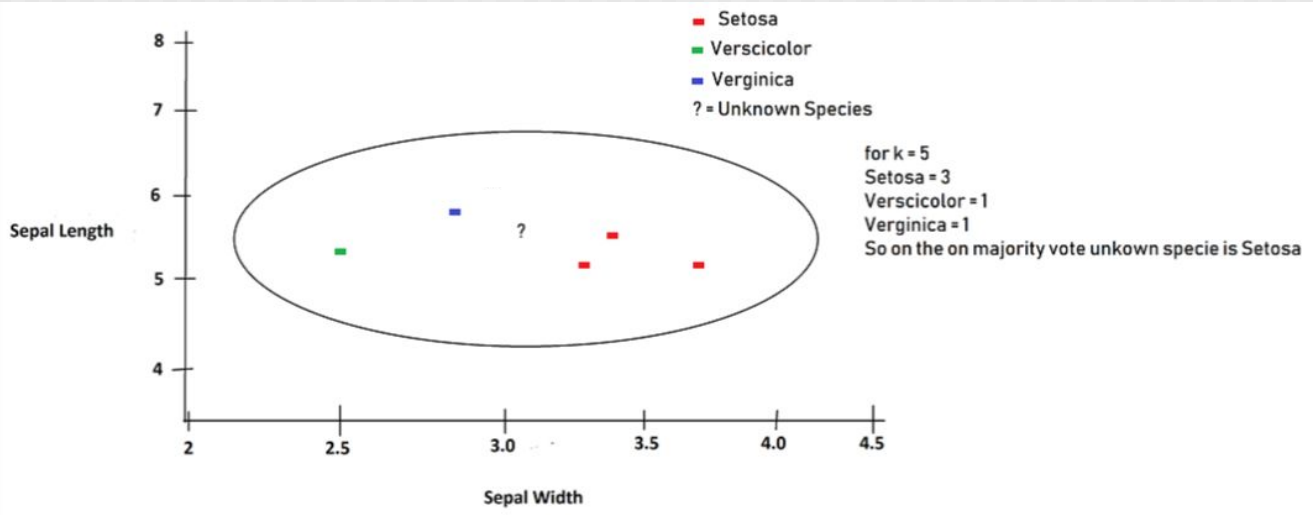
Which flower species?:

Sepal Length	Sepal Width	Species
5.2	3.1	?

KNN Algorithm Example

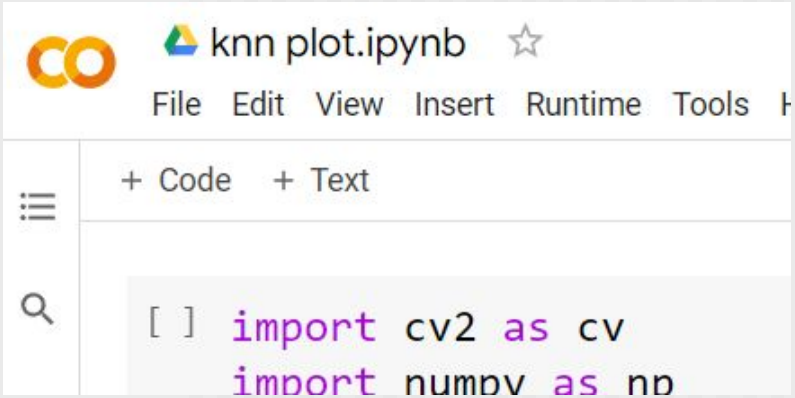
Sepal Length	Sepal Width	Species	Distance	Rank
5.1	3.3	Setosa	0.22	1
5.4	3.4	Setosa	0.36	2
5.1	3.7	Setosa	0.608	3
5.1	2.5	Verscicolor	0.6	4
5.8	2.8	Virginica	0.67	5

For K = 5, setosa has a count of 3, others have only 1.

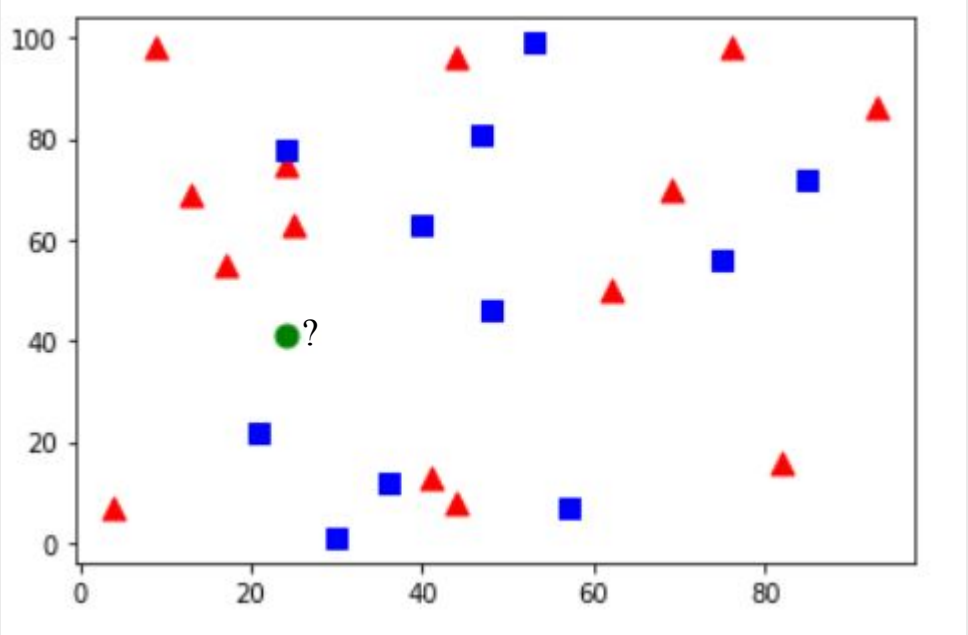


Sepal Length	Sepal Width	Species
5.2	3.1	Setosa

KNN in Colab



2 randomly generated classes of (x, y) values 1-100 (uniform distribution). Classify and unknown random (x, y) using KNN.



Quiz 1. Change to 3 random classes.

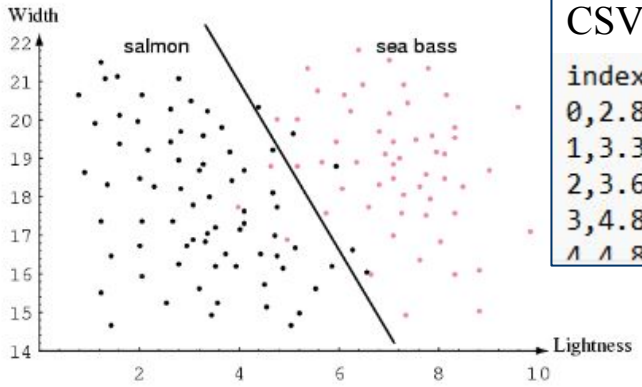
Quiz 2. Change class distribution to Gaussian Distribution.

Quiz 1. Problem 3.

3. *10 points.* 1 hrs. **KNN (K nearest neighbor) for 3 Classes.** Modify the provided program for KNN with 2 random red/blue classes shown in class to have 3 classes of red/blue/yellow instead. Then use K = 4 to classify a randomly generated sample as red, yellow, or blue.

Quiz 1. Problem 4.

4. *15 points.* 1.5 hrs. **KNN (K nearest neighbor) for Salmon/Sea Bass classification.**



CSV Dataset with 130 samples:

```
index,lightness,width,species
0,2.834754098360656,21.087142857142855,0
1,3.329180327868852,18.877142857142857,0
2,3.6904918032786886,19.824285714285715,0
3,4.812459016393442,17.759999999999998,0
4,4.812459016393442,16.497142857142855,0
```

Use the salmon vs. sea bass dataset "salmon_seabass.csv" provided in the shared folder. For each (x_i, y_i) data point do the following:

1. Assume you do not know whether this point (x_i, y_i) is a salmon or sea bass, but you know the label of all other points.
2. Use KNN with K = 3 to classify this point (x_i, y_i). Plot correctly classified salmon as a red circle, incorrectly classified salmon as a red "X", correctly classified sea bass as a blue triangle, and incorrectly classified sea bass as a blue "X".

Classification vs. Regression vs. Clustering

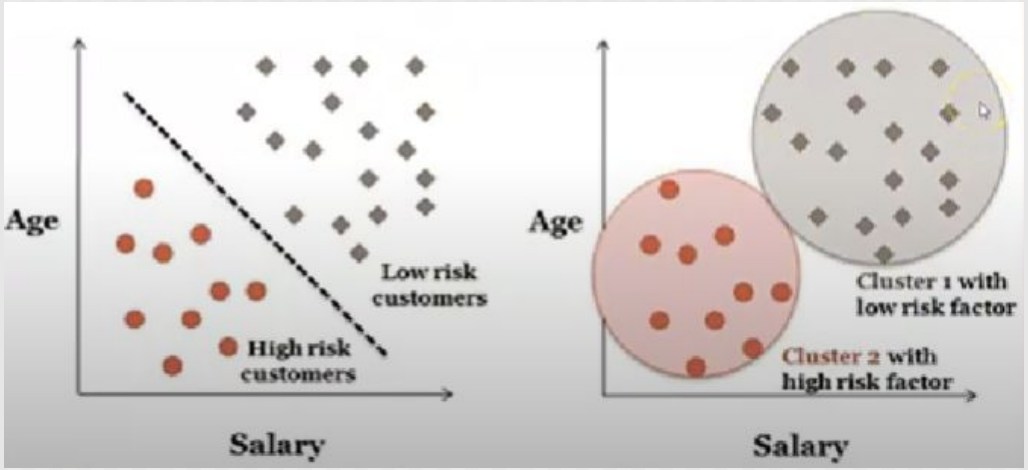
Some more introductory notes

Classification vs. Clustering

How risky is a customer based on features: (Age, Salary)?

https://www.youtube.com/watch?v=obih0Jikwmk&t=183s

Classification. In classification, the algorithm assigns labels to data based on the predefined features. This is an example of **supervised learning**. Can find what group new data falls under.



Clustering. An algorithm splits *unlabelled (no class)* data (features) into a number of clusters based on the similarity of features. *May not know how many clusters.* **Unsupervised learning**. Can find what chance new data belongs to each group.

Another example: Use of GPA and Communications Skills to cluster "good" employee called Worker Profiling.

Linear Discrimination. Supervised Learning.

A	B	C	D	E
	salary (x)	age (y)		
Key point 1:	50,000	22	**Slope:**	-0.00127
Key point 2:	20,000	60	**Intercept:**	85.3

Line Equation: age = -0.00127*salary + 85.3

Rule:

if age > threshold value of -0.00127*salary + 85.3 then:

    low risk

else:

    high risk.

Linear Discrimination

Rule:

threshold_age = -0.00127*salary + 85.3

if age > threshold_age then:

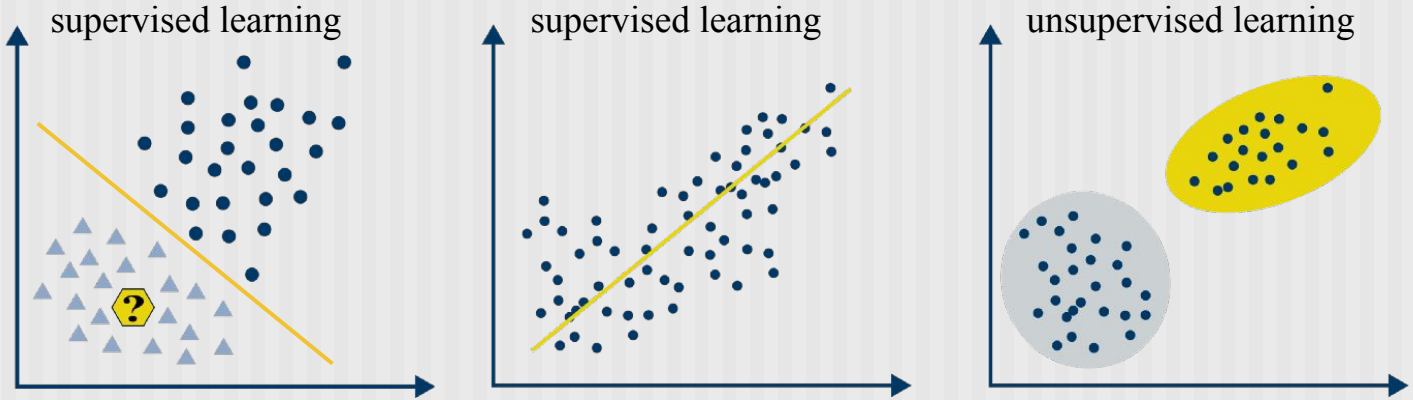
    low risk

else:

    high risk

Salary	Age	Threshold	Hi/Lo Risk?
10,000	22	72.7	High
31,000	22	46.1	High
20,000	22	60.0	High
50,000	30	22.0	Low
50,000	55	22.0	Low
50,000	20	22.0	High
100,000	40	-41.3	Low
70,000	40	-3.3	Low
60,000	35	9.3	Low
~~10,000~~		~~72.7~~	
20,000		60.0	
30,000		47.3	
40,000		34.7	
50,000		22.0	
60,000		9.3	
70,000		-3.3	
80,000		-16.0	

Classification vs. Regression vs. Clustering



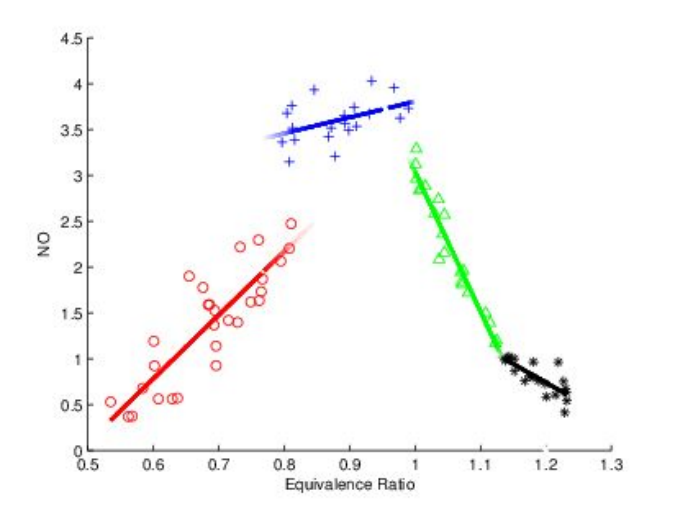
Regression algorithms try to find a relationship between variables and predict unknown dependent variables based on known data. It is based on supervised learning

Regression Example:

- We have historical time estimate data based on human judgement for software projects completed
- We capture certain features used in the pricing: # c1..c3 forms, # c1..c3 reports, # modules, users, …
- We try a linear fit to model time based on n features $x_1, \ldots, x_n$ with n unknowns $a_0, a_1, \ldots, a_n$

$$t\,(x_1, \ldots, x_n) = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n$$

Regression within each cluster



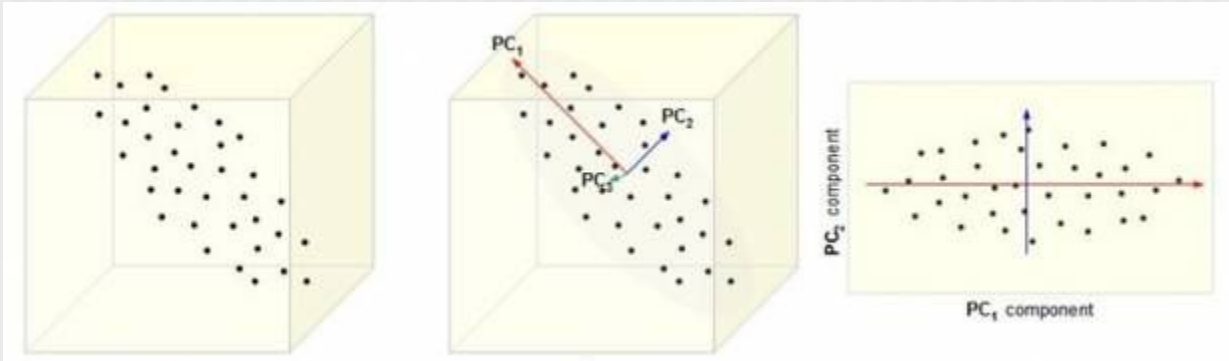
Example. How much to charge a client for visual effects service on a shot (video clip)?

In many cases a single regression will not work. The data must first be clustered into groups and regression fits better within each group. **Example:** Time taken to complete software projects that are small and similar to previous projects vs. software projects that are large and new.

Dimension Reduction Example

PCA - Principal Component Analysis. Goal: Reduce the dimensionality of a data set while retaining the variation present in it. Convert dataset into new features to reduce covariance

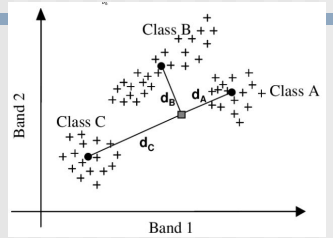
1. Find a new “origin” and orthonormal coordinate axis of the dataset by fitting an ellipsoid.
2. Eliminate the dimension(s) with the smallest radius.

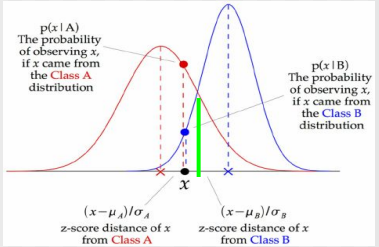


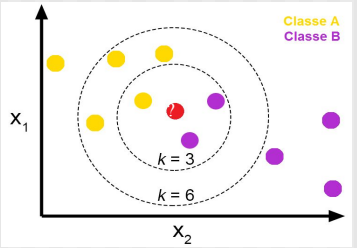
Reduction of 3D data to 2D data

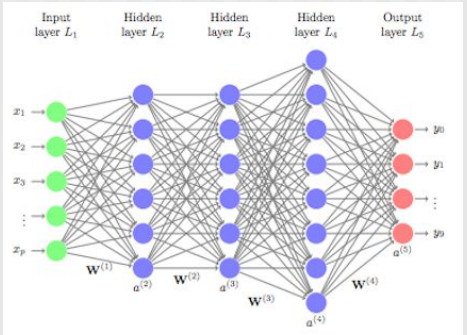
Common Pattern Classifiers

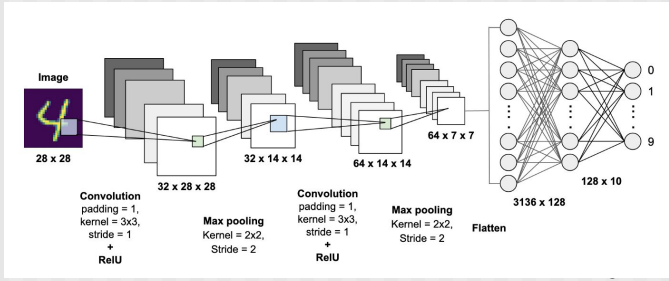
1. Minimum Distance to Mean Classifier
2. Gaussian Naive Bayesian Classifier
3. Machine Learning Based. eg. K-Nearest Neighbors (KNN)
4. Neural Networks based. eg. *Multi-Layer Perceptrons (MLP)* and *Convolutional Neural Networks* (CNN)



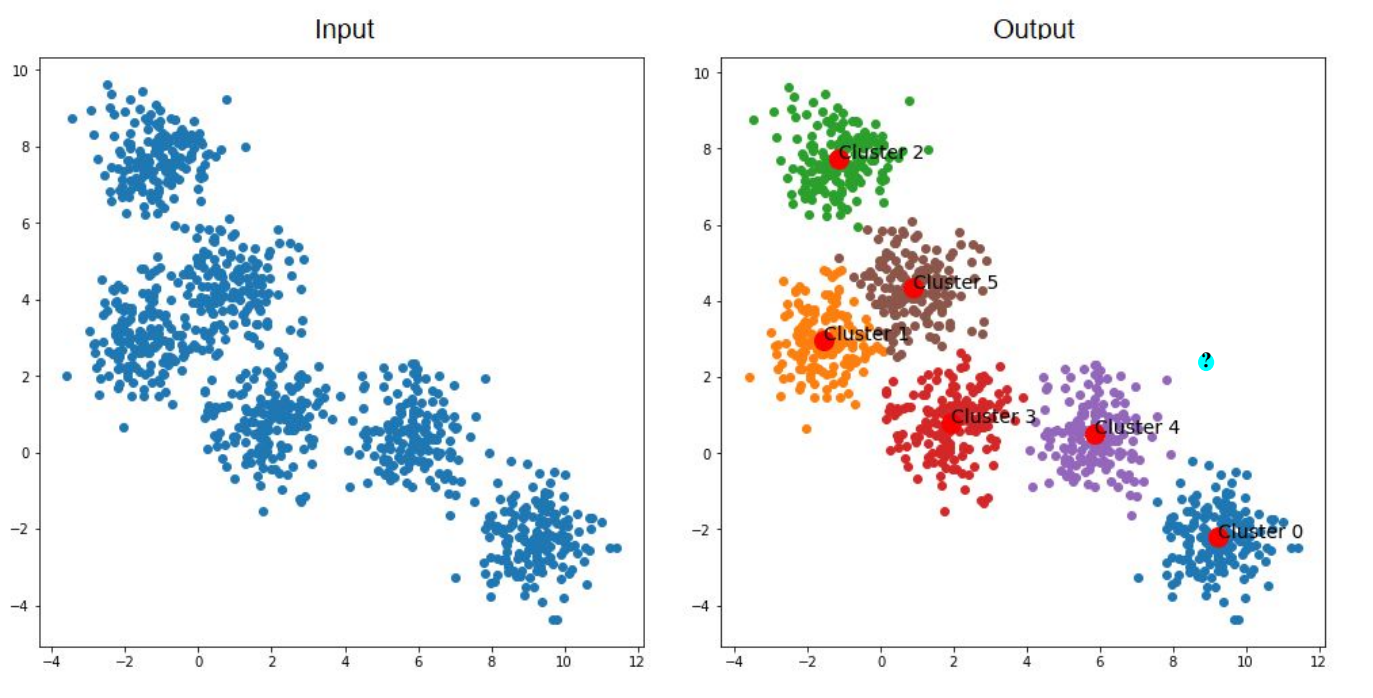








Classes may be learned via unsupervised clustering



Movie "?" is similar to which cluster of movies (belongs to which cluster)? I can suggest this movie to those who liked another movie of the same cluster.

Movie Features:

- Suspense
- Horror
- Thriller
- Action
- Romance
- Comedy
- Satire
- Political
- …

K-means clustering may produce 6 clusters. For the unknown feature "?" at (4, 3), the Min distance classifier to cluster mean will report cluster 3, but KNN with K = 5 will report cluster 5.

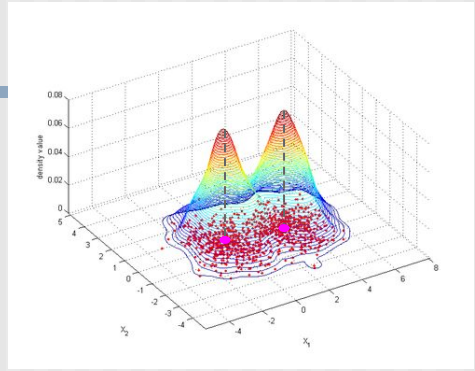
The minimum distance classifier

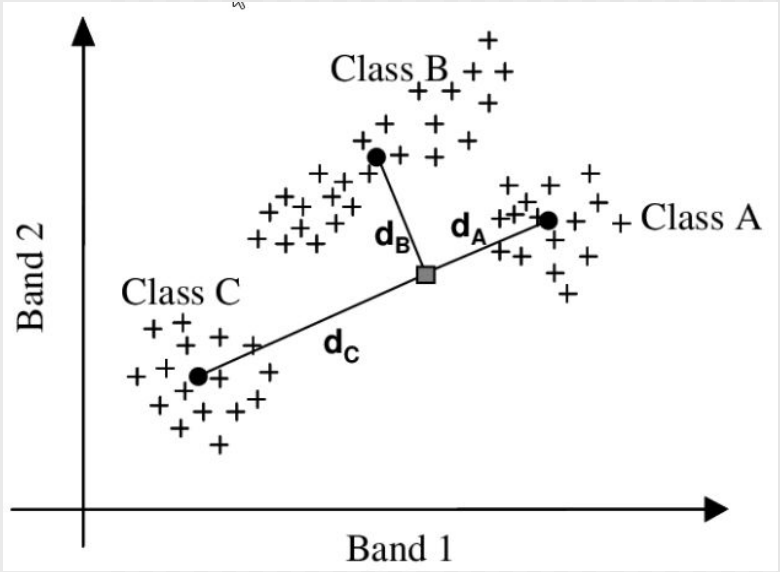
To the cluster mean or to the cluster SD

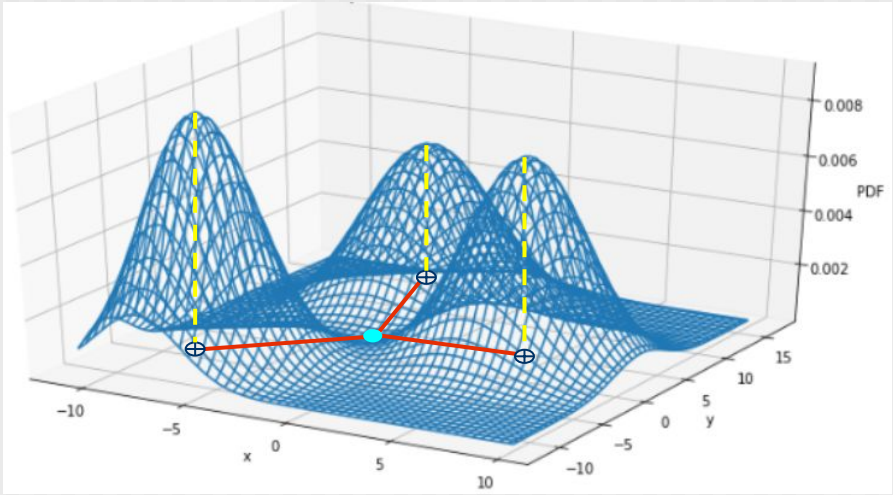
Common Pattern Classifiers

1. Minimum Distance to Mean Classifier.

Minimum Distance Classifier.



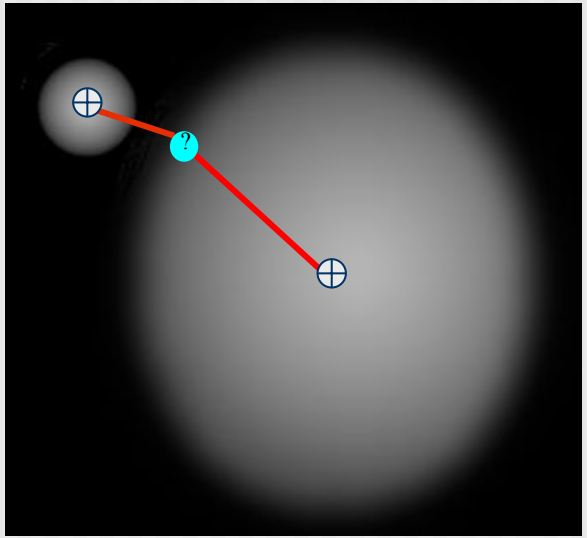


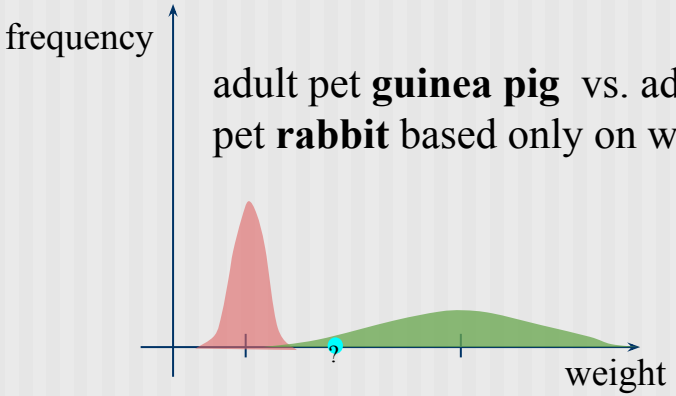


Common Pattern Classifiers

1. Minimum Distance Classifier

A clear weakness when standard deviation is not take into account.

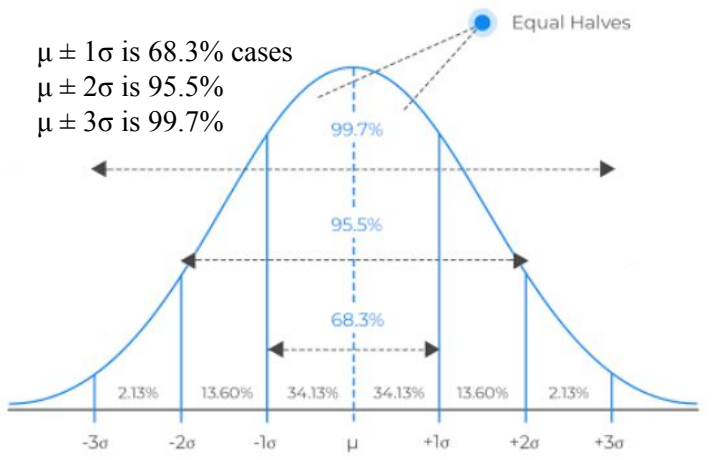




Understanding the Gaussian Distribution

The Gaussian Distribution

https://medium.com/@gasmielhadi/gaussian-normal-distribution-27ee4380f1f8



Almost all data fits within ± 3 standard deviations from the mean.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$Z = \int_{-\infty}^{\infty} f(x)dx$$

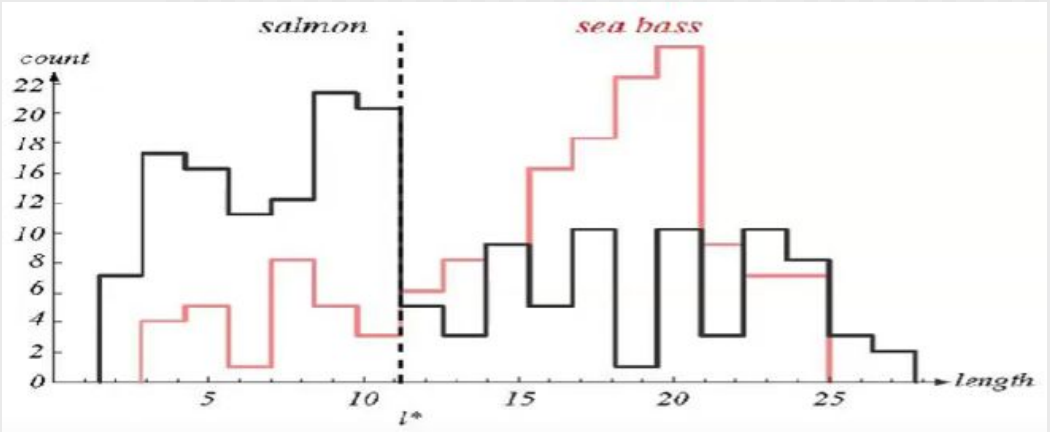
$$P(x) = \frac{1}{Z} f(x)$$

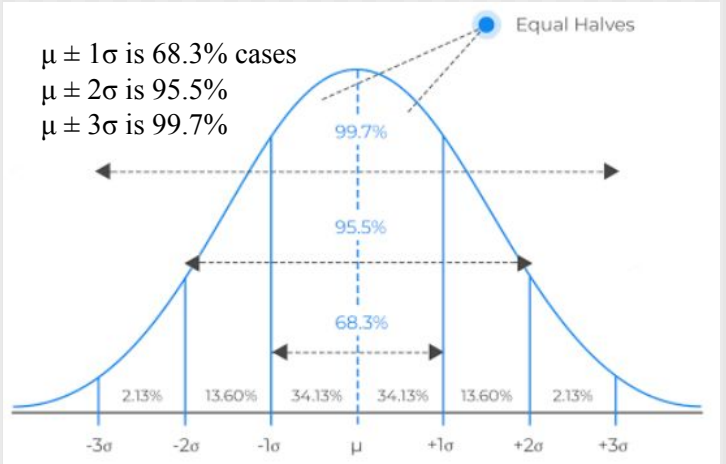
$$P(x_0) = P(x = x_0) = \frac{1}{Z} f(x_0)$$

(scratch slide)

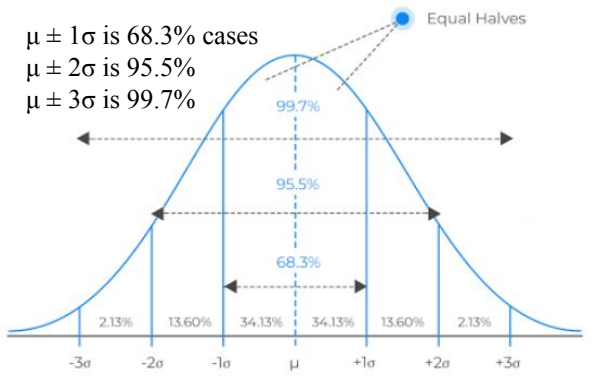
Most distributions can be well-approximated by Gaussians.

Let's try case of Salmon width $\mu = 13.0$, $\sigma = 4$. Sea bass width $\mu = 17.0$, $\sigma = 3.5$.



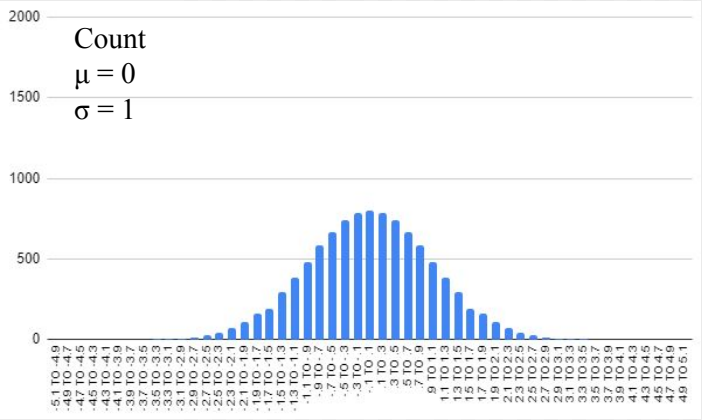


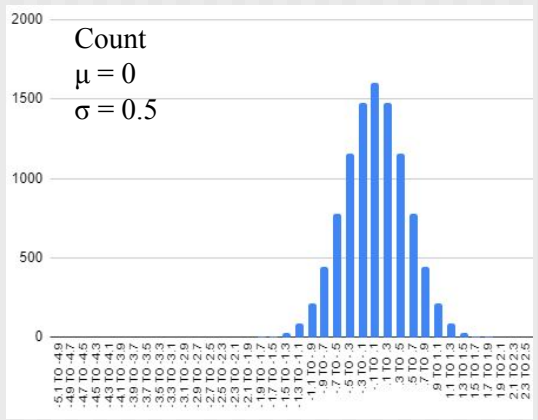
Gaussian Probability Continued



Range	Count	Cum
-1.0 TO 1.0	6827	68.27%
-2.0 TO 2.0	9545	95.45%
-3.0 TO 3.0	9973	99.73%

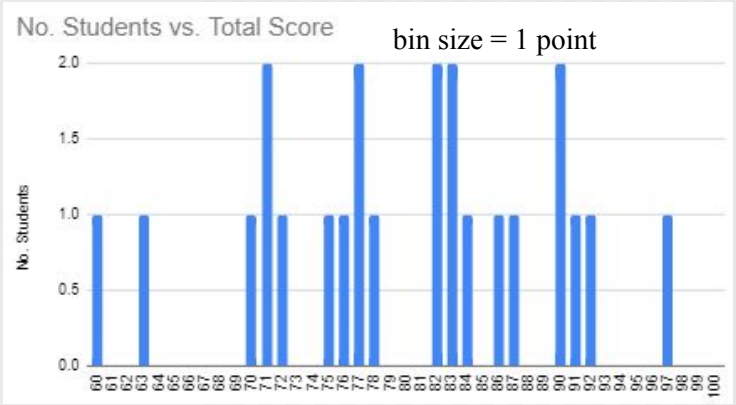
Lower sigma is thinner and taller

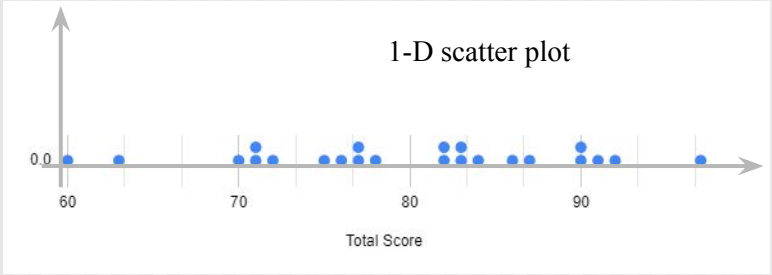


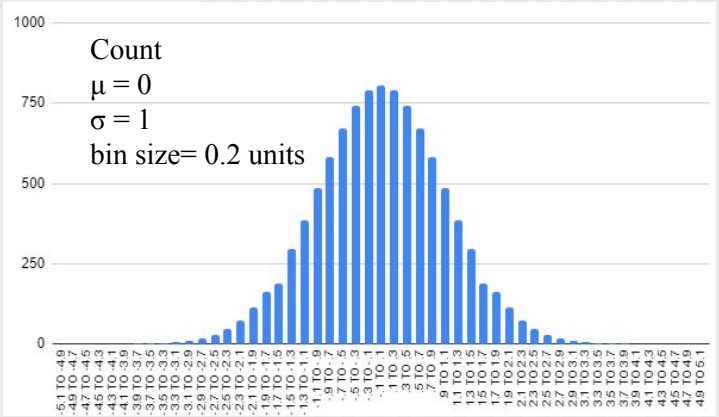


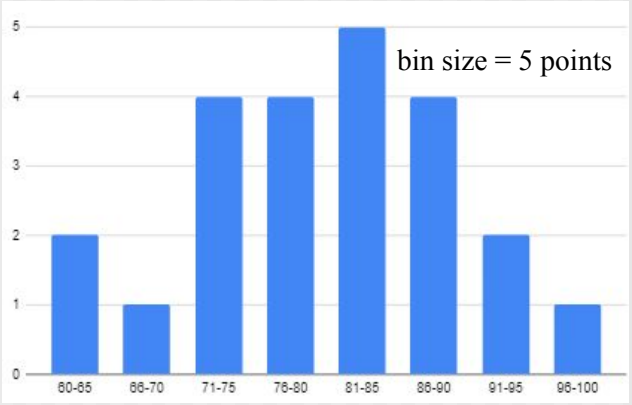
Range	Count	Cum
-3.9 TO -3.7	1	1
-3.7 TO -3.5	1	2
-3.5 TO -3.3	3	5
-3.3 TO -3.1	5	10
-3.1 TO -2.9	9	19
-2.9 TO -2.7	16	35
-2.7 TO -2.5	28	63
-2.5 TO -2.3	46	109
-2.3 TO -2.1	72	181
-2.1 TO -1.9	109	290
-1.9 TO -1.7	160	450
-1.7 TO -1.5	225	675
-1.5 TO -1.3	302	977
-1.3 TO -1.1	391	1368
-1.1 TO -.9	486	1854
-.9 TO -.7	581	2435
-.7 TO -.5	668	3103
-.5 TO -.3	737	3840
-.3 TO -.1	782	4622
-.1 TO .1	796	5418
.1 TO .3	780	6198
.3 TO .5	734	6932
.5 TO .7	664	7596
.7 TO .9	577	8173
.9 TO 1.1	481	8654
1.1 TO 1.3	387	9041
1.3 TO 1.5	297	9338
1.5 TO 1.7	221	9559
1.7 TO 1.9	157	9716
1.9 TO 2.1	108	9824
2.1 TO 2.3	70	9894
2.3 TO 2.5	45	9939
2.5 TO 2.7	27	9966
2.7 TO 2.9	16	9982
2.9 TO 3.1	8	9990
3.1 TO 3.3	5	9995
3.3 TO 3.5	3	9998
3.5 TO 3.7	1	9999
3.7 TO 3.9	1	10000

Normal Curve for Student Exam scores







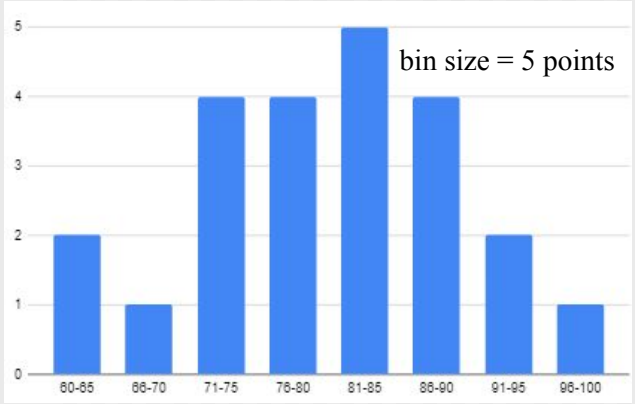


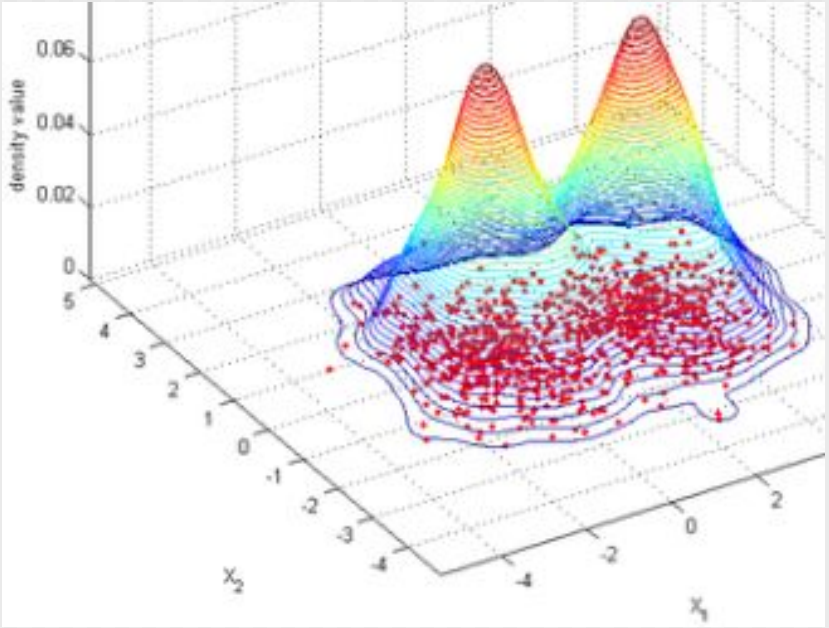
Bin Size is important in helping view the data distribution.

Total Score	No. Students	Cumulative
60	1	1
61	0	1
62	0	1
63	1	2
64	0	2
65	0	2
66	0	2
67	0	2
68	0	2
69	0	2
70	1	3
71	2	5
72	1	6
73	0	6
74	0	6
75	1	7
76	1	8
77	2	10
78	1	11
79	0	11
80	0	11
81	0	11
82	2	13
83	2	15
84	1	16
85	0	16
86	1	17
87	1	18
88	0	18
89	0	18
90	2	20
91	1	21
92	1	22
93	0	22
94	0	22
95	0	22
96	0	22
97	1	23
98	0	23
99	0	23
100	0	23

2-D Gaussian Probability

Even scatter plots are summed by 2-D bin to get 3D histogram. The height is the number of data points in each rectangular grid.





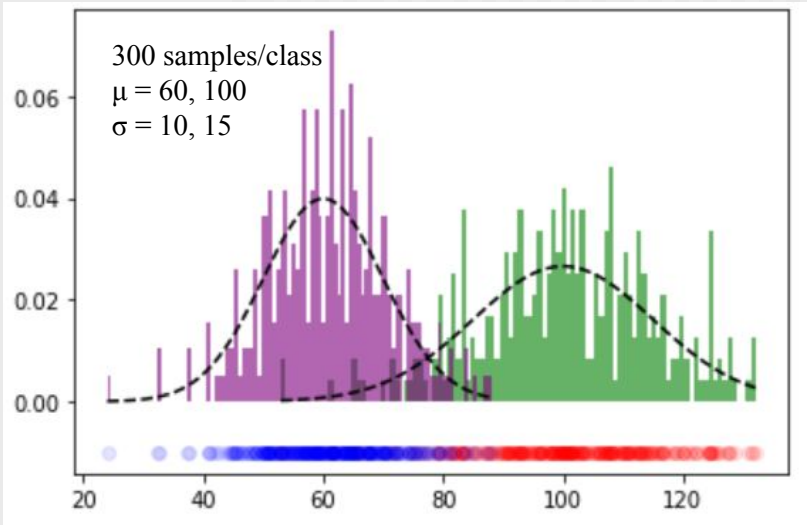
Hands on with Gaussian Distribution (1-D)

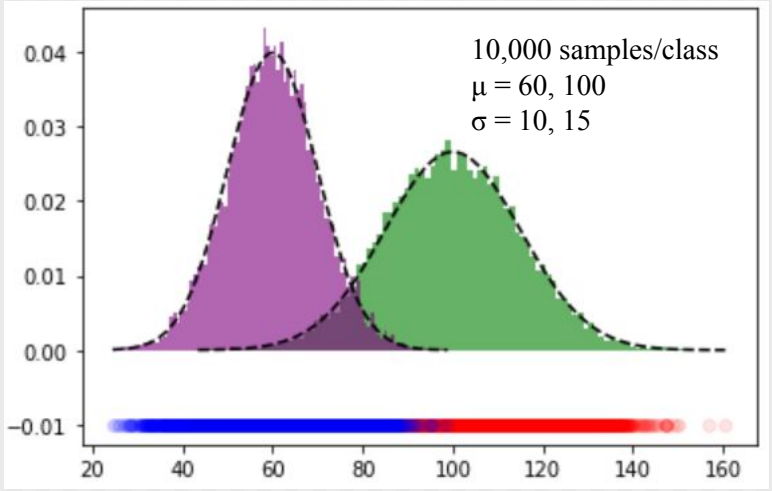
Gaussian Data.ipynb ☆

File Edit View Insert Runtime Tools Help All changes sa

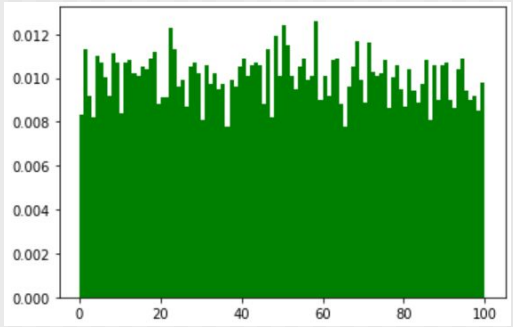
+ Code + Text

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
```

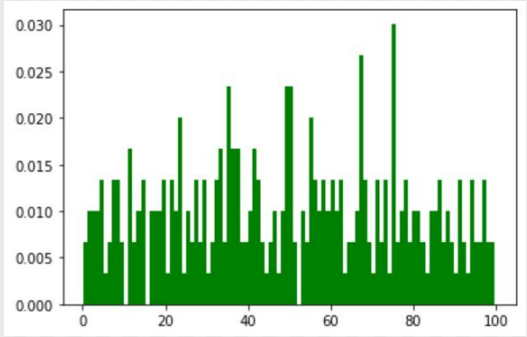




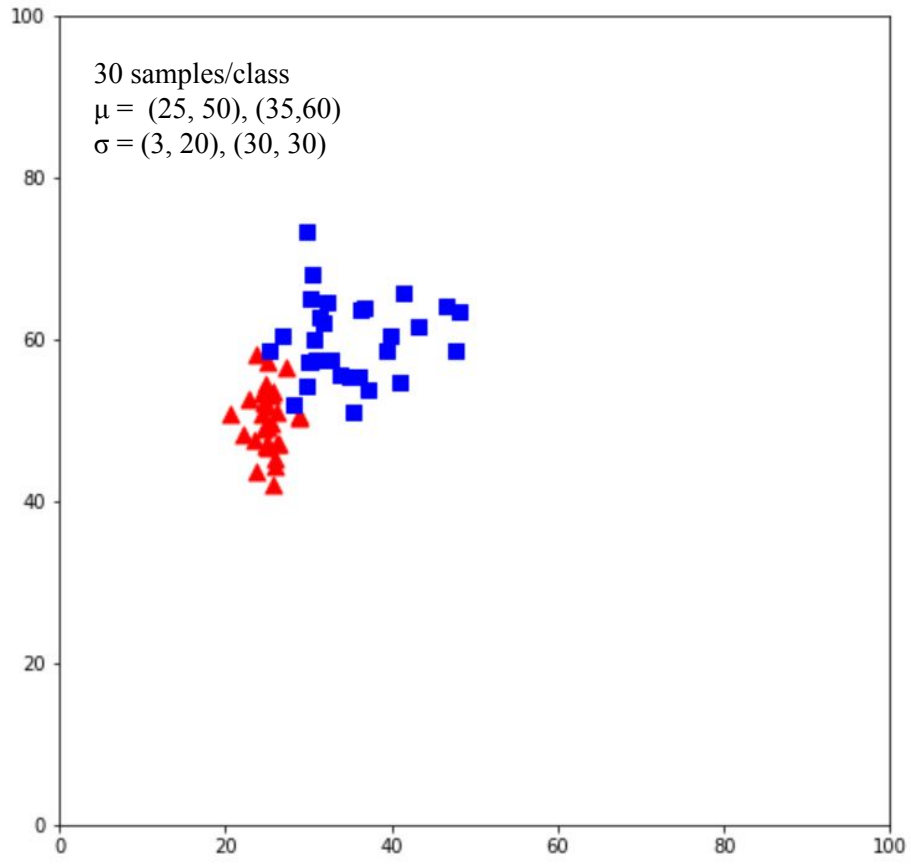
Uniform distribution 0-100. 10,000 samples:

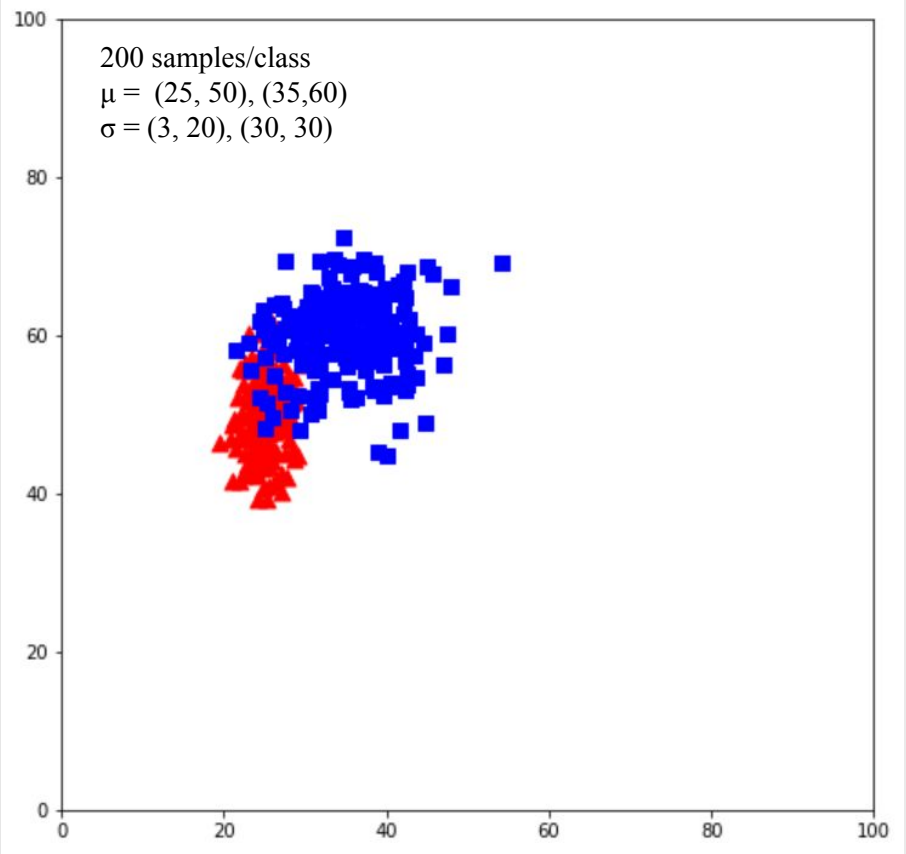


300 samples:



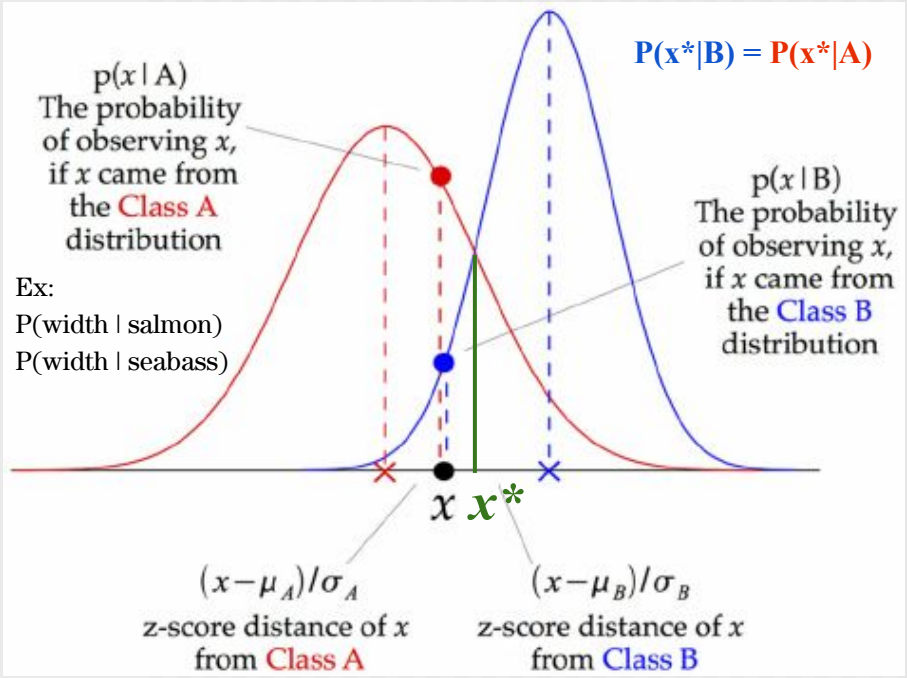
Hands on with Gaussian Distribution (2D)





Common Pattern Classifiers

2. Gaussian Naive Bayesian Classifier



Classifier decides based on P(A | **x**) and P(B | **x**). The class with higher probability is chosen.

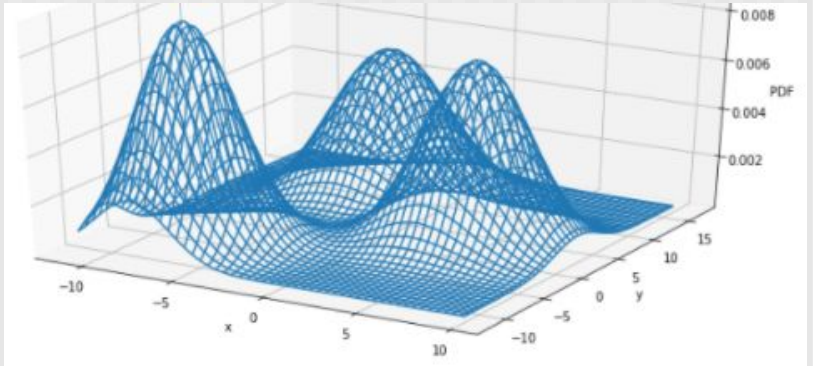
$$P\left(A \mid \vec{x}\right) = \frac{P\left(\vec{x} \mid A\right) \cdot P(A)}{P\left(\vec{x}\right)} \propto P\left(\vec{x} \mid A\right) \cdot P(A)$$

known from the samples

$$P\left(B \mid \vec{x}\right) = \frac{P\left(\vec{x} \mid B\right) \cdot P(B)}{P\left(\vec{x}\right)} \propto P\left(\vec{x} \mid B\right) \cdot P(B)$$

if x_i are independent

In general **x** = [x_1, x_2, …, x_n] is a n-dimensional feature vector. P(c_i | x_1, x_2) is shown below for 3 classes.



Scratch

Classifier decides based on P(A | **x**) and P(B | **x**). The class with higher probability is chosen.

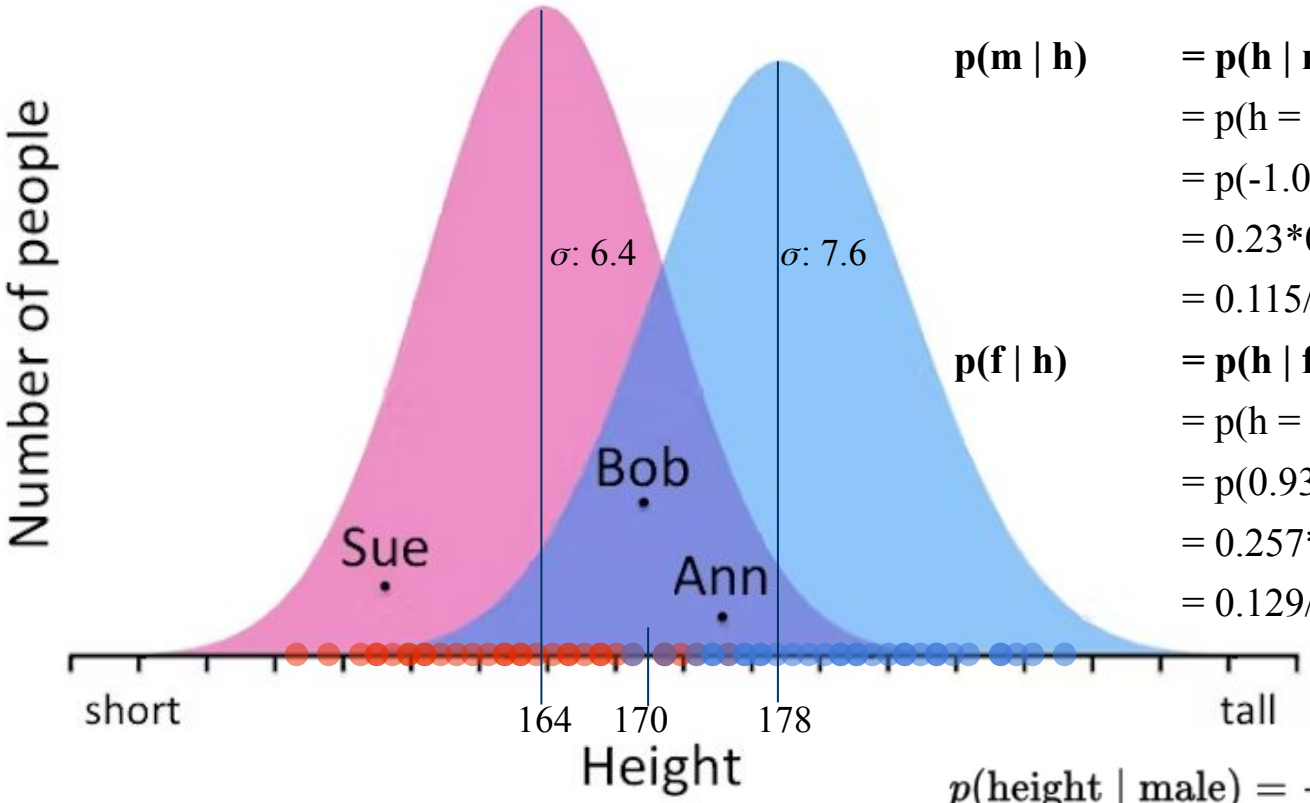
$$P\left(A \mid \vec{x}\right) = \frac{P\left(\vec{x} \mid A\right) \cdot P(A)}{P\left(\vec{x}\right)} \propto \boxed{P\left(\vec{x} \mid A\right)} \cdot \boxed{P(A)}$$

known from the samples

$$P\left(B \mid \vec{x}\right) = \frac{P\left(\vec{x} \mid B\right) \cdot P(B)}{P\left(\vec{x}\right)} \propto \boxed{P\left(\vec{x} \mid B\right)} \cdot \boxed{P(B)}$$

if x_i are independent

1-D Feature Example to classify Male vs. Female



Higher Bayesian Probability.

p(m | h) **= p(h | m) * p(m)/p(h)**

= p(h = 170, N(μ:178, σ:7.6)) * 0.5/p(h)

= p(-1.05, N(0,1)) * 0.5 / p(h)

= 0.23*0.5/p(h)

= 0.115/p(h)

p(f | h) **= p(h | f) * p(f)/p(h)**

= p(h = 170, N(μ:164, σ:6.4)) * 0.5/p(h)

= p(0.937, N(0,1)) * 0.5 / p(h)

= 0.257*0.5/p(h)

= 0.129/p(h)

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(h-\mu)^2}{2\sigma^2}\right)$$

Z Score makes it easier to calculate p(class | feature)

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$Z = \frac{X-\mu}{\sigma}$$

$$Z \sim N(0,1)$$

$$\mu = 0 \quad \sigma = 1$$

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

p(m | h)

x	f(x). G(mean, sd).
-5.000000000	0.0000014867
-4.990000000	0.0000015629
-1.050000000	0.2298821407

p(f | h)

x	f(x). G(mean, sd).
-5.000000000	0.0000014867
-4.990000000	0.0000015629
0.930000000	0.2588805467
0.940000000	0.2564712944

= p(h | m) * p(m)/p(h)

= p(h = 170, N(μ:178, σ:7.6)) * 0.5/p(h)

= p(-1.05, N(0,1)) * 0.5 / p(h)

= 0.23*0.5/p(h)

= 0.115/p(h)

= p(h | f) * p(f)/p(h)

= p(h = 170, N(μ:164, σ:6.4)) * 0.5/p(h)

= p(0.937, N(0,1)) * 0.5 / p(h)

= 0.257*0.5/p(h)

= 0.129/p(h)

Bayesian Classifier

$$P(A \cap B) = P(A|B) \cdot P(B)$$

$$P(B \cap A) = P(A \cap B)$$

$$P(B|A) \cdot P(A) = P(A|B) \cdot P(B)$$

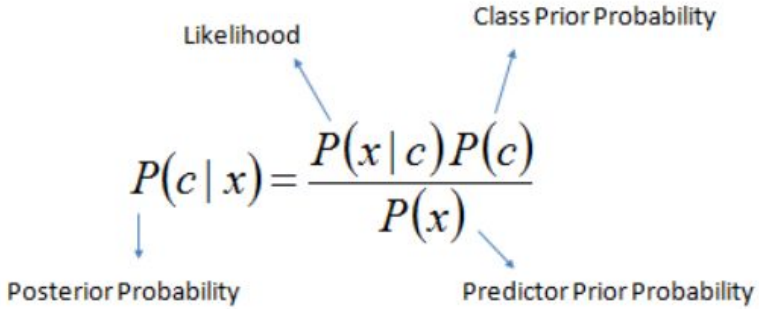
$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

$$\begin{aligned} P(x_1, x_2, x_3|c) &= P(x_1, x_2, x_3, c)/P(c) \\ &= P(x_1|x_2, x_3, c) \cdot P(x_2, x_3, c)/P(c) \\ &= P(x_1|x_2, x_3, c) \cdot P(x_2|x_3, c) \cdot P(x_3, c)/P(c) \\ &= P(x_1|x_2, x_3, c) \cdot P(x_2|x_3, c) \cdot P(x_3|c) \cdot P(c)/P(c) \\ &= P(x_1|x_2, x_3, c) \cdot P(x_2|x_3, c) \cdot P(x_3|c) \end{aligned}$$

$$P(x_1, x_2, ..., x_n|c) = P(x_1|x_2, ..., x_n, c) \cdot P(x_2|x_3, ..., x_n, c) \cdot ... \cdot P(x_n|c)$$

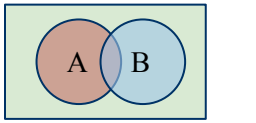
$$P(c|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|c) \cdot P(c)}{P(x_1, ..., x_n)}$$

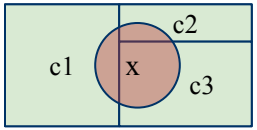
$$P(c_k|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|c_k) \cdot P(c_k)}{P(x_1, ..., x_n)}$$

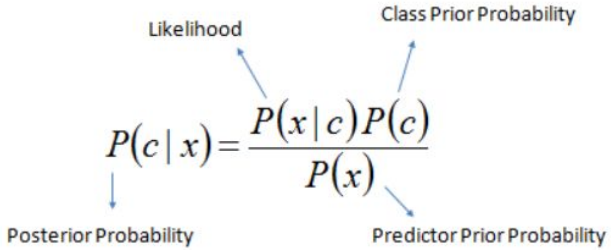


- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*features*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Naive Bayes Classifier







$$P(A \cup B) = P(A) + P(B) - P(A, B)$$

$$U = c_1 \cap c_2 \cap c_m$$

$$\begin{aligned} P(x) &= P(x, U) = P(x, c_1, c_2, ..., c_m) \\ &= P(x, c_1) + ... + P(x, c_m) \\ &= P(x|c_1) \cdot P(c_1) + ... + P(x|c_m) \cdot P(c_m) \end{aligned}$$

$$P(x_1, x_2, ..., x_n|c_k) = P(x_1|x_2, ..., x_n, c_k) \cdot P(x_2|x_3, ..., x_n, c_k) \cdot ... \cdot P(x_n|c_k)$$

$$P(c_k|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|c_k) \cdot P(c_k)}{P(x_1, ..., x_n)}$$

$$p(C_k|x_1, \dots, x_n) = p(C_k) \prod_{i=1}^{n} \frac{p(x_i|C_k)}{p(x_i)}$$

$$Naive\ Bayes : P(c_k|x_1, ..., x_n) = \frac{P(x_1|c_k) \cdot P(x_2|c_k) \cdot ... \cdot P(x_n|c_k) \cdot P(c_k)}{P(x_1) \cdot P(x_2) \cdot ... \cdot P(x_n)}$$

Mutually exclusive events are when two events cannot happen at the same time: $P(A \cap B) = 0$

While Independent events are when two events do not influence each other: $P(A \cap B) = P(A) \cdot P(B)$

Naive Bayes

$$P(x) = P(x|c_1) \cdot P(c_1) + ... + P(x|c_m) \cdot P(c_m)$$

$$P(c_k|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|c_k) \cdot P(c_k)}{P(x_1, ..., x_n)}$$

Naive Bayes :

$$P(c_k|x_1, ..., x_n) = \frac{P(x_1|c_k) \cdot P(x_2|c_k) \cdot ... \cdot P(x_n|c_k) \cdot P(c_k)}{P(x_1) \cdot P(x_2) \cdot ... \cdot P(x_n)}$$

$$p(C_k|x_1, \dots, x_n) = p(C_k) \prod_{i=1}^{n} \frac{p(x_i|C_k)}{p(x_i)}$$

$$= \frac{P(x_1|c_k) \cdot P(x_2|c_k) \cdot ... \cdot P(x_n|c_k) \cdot P(c_k)}{[P(x_1|c_1) \cdot P(c_1) + ... + P(x_1|c_m) \cdot P(c_m)] \cdot ... \cdot [P(x_n|c_1) \cdot P(c_1) + ... + P(x_n|c_m) \cdot P(c_m)]}$$

Naive Bayes should work well since length and color and not correlated. **Highest probability class wins.**

$$P(salmon|len, color) = \frac{P(len|salmon) \cdot P(color|salmon)}{[P(len|salmon) \cdot P(salmon) + P(len|sbass) \cdot P(sbass)] \cdot [P(color|salmon) \cdot P(salmon) + P(color|sbass) \cdot P(sbass)]}$$

$$P(sbass|len, color) = \frac{P(len|sbass) \cdot P(color|sbass)}{[P(len|salmon) \cdot P(salmon) + P(len|sbass) \cdot P(sbass)] \cdot [P(color|salmon) \cdot P(salmon) + P(color|sbass) \cdot P(sbass)]}$$

Naive Bayes can ignore denominator

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Since $P(x_1, \ldots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Naive Bayes Example. Male or Female from 3 features?

Classify 2 classes y_i= {male, female} based on the 3 features x_i = {height, weight, foot size}

Classify Test Sample:

Gender	height	weight	foot size
?	6	130	8

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Example Training Data:

Gender	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92	190	11
male	5.58	170	12
male	5.92	165	10
female	5	100	6
female	5.5	150	8
female	5.42	130	7
female	5.75	150	9

Finding Mean, SD, and Distribution

11.2, 11.9, 12.0, 12.8, 13.4, 14.3

$$\bar{x} = \frac{\sum x}{n}$$

$$= \frac{11.2+11.9+12.0+12.8+13.4+14.3}{6}$$

$$= \frac{75.6}{6} = 12.6$$

$$s = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}$$

Calculate the mean

SD divided by n for population, divided by (n-1) for sample of a population.

Population: KMUTT.

Sample: Class samples KMUTT

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$P(x) = \frac{1}{Z} \cdot f(x)$$

x	$\bar{x}$	$x - \bar{x}$	$(x - \bar{x})^2$
11.2	12.6	-1.4	1.96
11.9	12.6	-0.7	0.49
12.0	12.6	-0.6	0.36
12.8	12.6	0.2	0.04
13.4	12.6	0.8	0.64
14.3	12.6	1.7	2.89
			6.38

$$s = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}$$

$$= \sqrt{\frac{6.38}{5}}$$

Naive Bayes Example. Male or Female?

Test sample:

Gender	height	weight	foot size
?	6	130	8

Gender	Height Mean	Height SD	Weight Mean	Weight SD	Foot Size Mean	Foot Size SD	P(x)
Male	5.86	0.1872	176	11.0868	11	0.9574	0.49
Female	5.42	0.3118	133	23.6291	8	1.2910	0.51

$$P(c_k|x_1, \ldots, x_n) \propto P(x_1|c_k) \cdot P(x_2|c_k) \cdot \ldots \cdot P(x_n|c_k) \cdot P(c_k)$$

$$P(male|height, weight, foot) \propto P(male) \cdot P(height|male) \cdot P(weight|male) \cdot P(foot|male))$$

$$P(female|height, weight, foot) \propto P(female) \cdot P(height|female) \cdot P(weight|female) \cdot P(foot|female))$$

Naive Bayes Example. Male or Female?

Test sample:

Gender	height	weight	foot size
?	6	130	8

Gender	height	weight	foot size
Female	6	130	8

$$P(c_k|x_1, \ldots, x_n) \propto P(x_1|c_k) \cdot P(x_2|c_k) \cdot \ldots \cdot P(x_n|c_k) \cdot P(c_k)$$

$$P(male|height, weight, foot) \propto P(male) \cdot P(height|male) \cdot P(weight|male) \cdot P(foot|male))$$

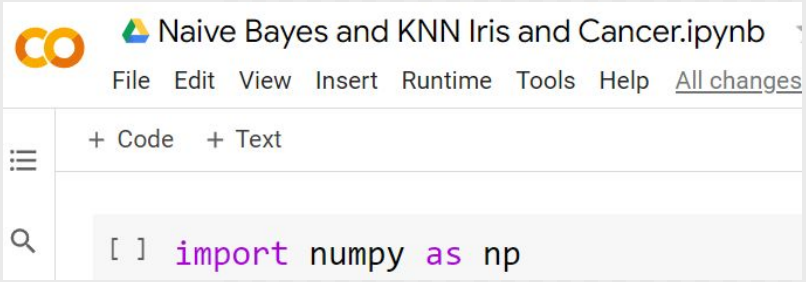
$$P(female|height, weight, foot) \propto P(female) \cdot P(height|female) \cdot P(weight|female) \cdot P(foot|female))$$

gender	P(h = 6 |gender)	P(w = 130 |gender)	P(foot =8 |gender)	P(gender)	P(gender | h, w, f)
Male	1.5788831833	0.0000059867	0.0013112210	0.4900000000	0.0000000061
Female	0.2234587268	0.0167892979	0.2866906999	0.5100000000	0.0005485467

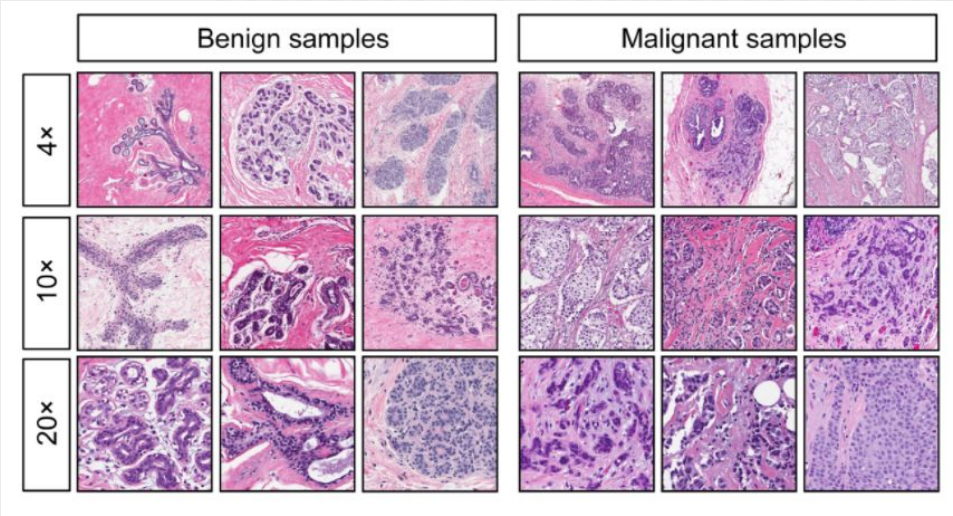
Answer: Female.

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789$$

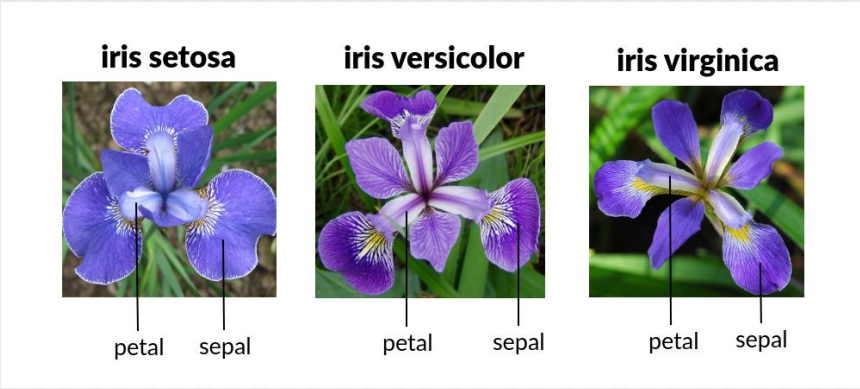
Naive Bayes for Iris and breast cancer classification using sklearn



Breast cancer dataset



Iris dataset



Code: Naive Bayes and KNN for Iris and breast cancer

Naive Bayes Iris and Cancer.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[ ]  #copied mainly from https://scikit-learn.org/stable/modules/naive_bayes.html
```

```
[23] from sklearn.datasets import load_iris, load_breast_cancer
     from sklearn.model_selection import train_test_split
     from sklearn.naive_bayes import GaussianNB
     from sklearn.metrics import accuracy_score
```

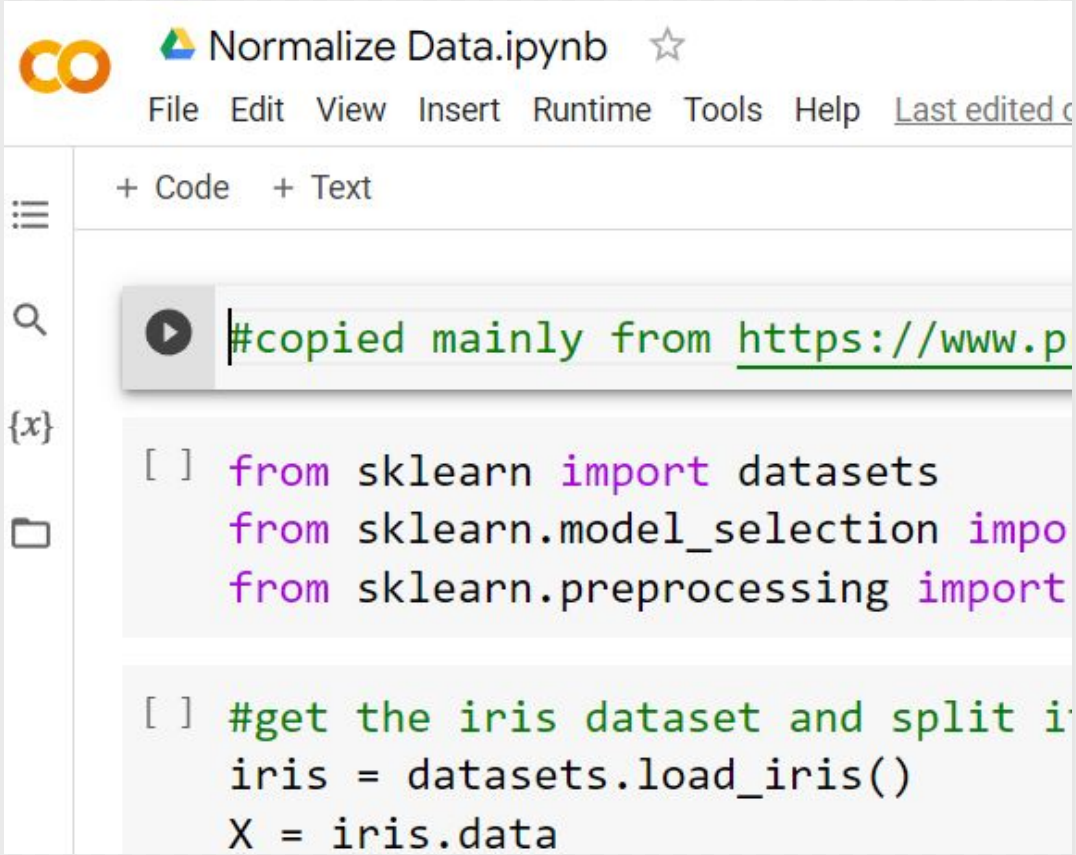
```
[28] #see https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

     dataset_type = 'none'
     while dataset_type not in {'iris', 'cancer'}:
       dataset_type = input('Enter iris for the Iris dataset and cancer for the Breast Cancer dataset ')
       if dataset_type not in {'iris', 'cancer'}:
         print("Invalid response: '%s'. Please try again." % dataset_type)
```

```
Enter iris for the Iris dataset and cancer for the Breast Cancer dataset cancer
```

```
if dataset_type == 'iris':
  the_data = load_iris() #get the data from sklearn
else:
  the_data = load_breast_cancer() #get the data from sklearn
```

Normalizing Example



Normalizing the data (Regularization) may help

For machine learning, normalization is required only when features have different ranges.

Example of working adults features:

- x: Age Range 18–100.
- y: Monthly Salary Range 5,000 – 1,000,000 baht.

These 2 features have very different ranges. Consider distances (age, salary):

- (30, 30000) vs. (40, 35000). The distance in age is only 10, but the distance in salary is 5000. This biases the system towards trying to fit the salary data more because it is generally a bigger number.
- Fitting models such as linear regression will bias towards salary more to minimize the error.
- KNN will face this problem if not scaled. Naive Bayes will not because it normalizes to Z-score standard normal N(0, 1).

Normalizing the data

A common normalization to make the data in units of σ from 0:

$$\vec{x}_{norm} = \frac{(\vec{x} - \vec{\mu})}{\vec{\sigma}}$$

The above converts the features into a Z distribution of N(0, 1)

Sometimes to normalize the data is simply scaled to be 0-1:

$$\vec{x}_{norm} = \frac{(\vec{x} - \vec{x}_{min})}{\vec{x}_{max} - \vec{x}_{min}}$$

K-fold cross validation

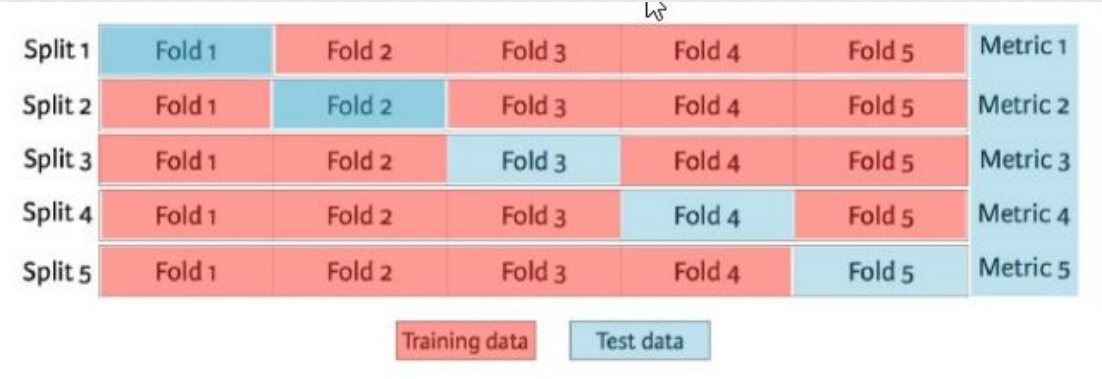
Cross-validation - dataset is randomly split up into ‘k’ groups or “folds”

If k = 5, then ⅕ of the data or 20% becomes test set, remaining 80% training.

The model is trained on the training set and scored on the test set.

This is done 5 times so each of the 5 groups has become the test set.

At the end an average of all these models is used to get the final accuracy.

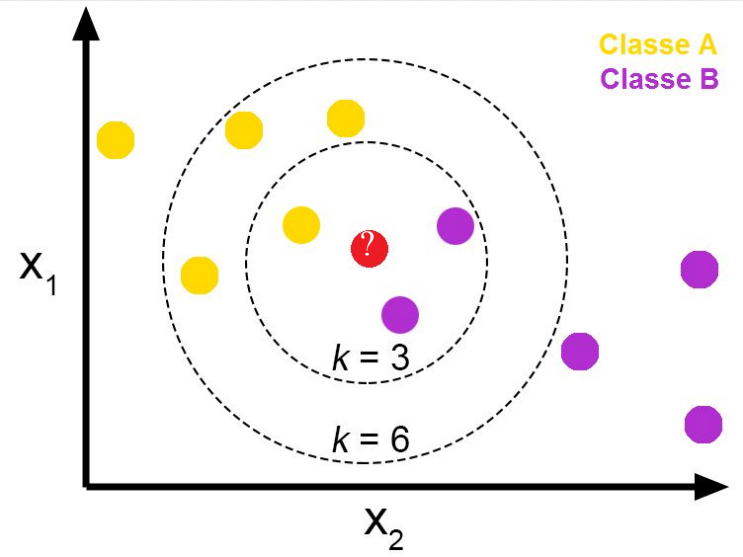


K fold cross validation is a method used to prevent overfitting, where the model fits the training data too well but does not generalize well enough to other data.

Common Pattern Classifiers (1. Nearest Nbr, 2. Naive Bayes)

3. Machine Learning Based such as K-Nearest Neighbors (KNN)

First must decide on K = ?. If K = 3, **Class B** wins with count **2** vs. **1**. If K = 6, **Class A** wins with count **4** vs. **2**.



KNN Algorithm Example

https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

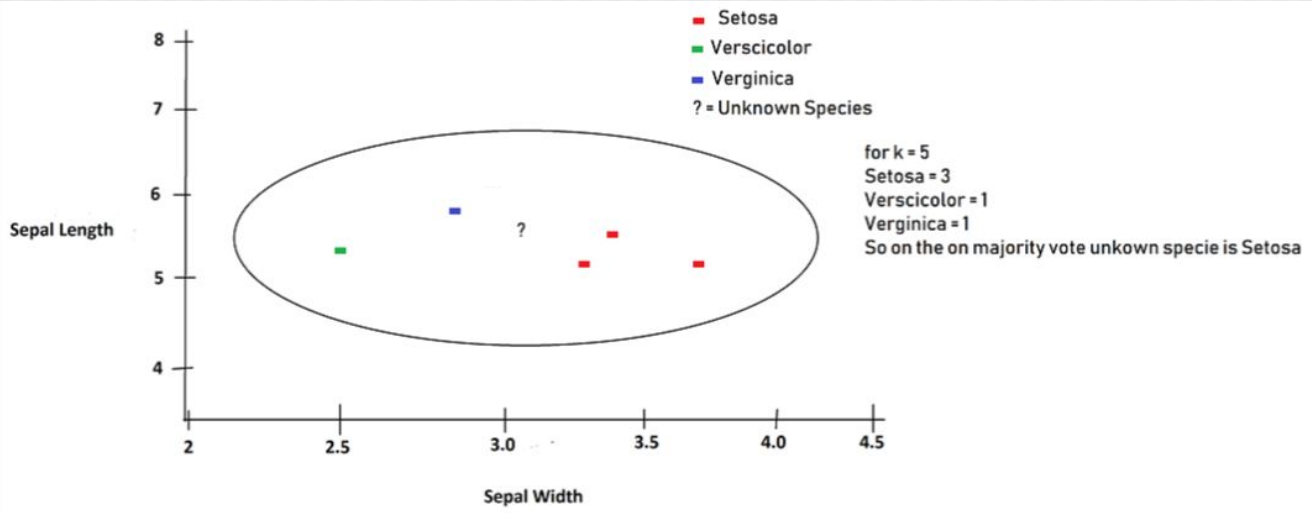
Sepal Length	Sepal Width	Species
5.2	3.1	?

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

KNN Algorithm Example

Sepal Length	Sepal Width	Species	Distance	Rank
5.1	3.3	Setosa	0.22	1
5.4	3.4	Setosa	0.36	2
5.1	3.7	Setosa	0.608	3
5.1	2.5	Verscicolor	0.6	4
5.8	2.8	Virginica	0.67	5

For K = 5, setosa has a count of 3, others have only 1.

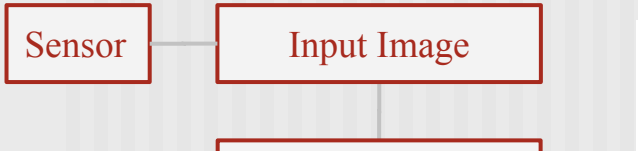


Sepal Length	Sepal Width	Species
5.2	3.1	Setosa

Common Pattern Classifiers

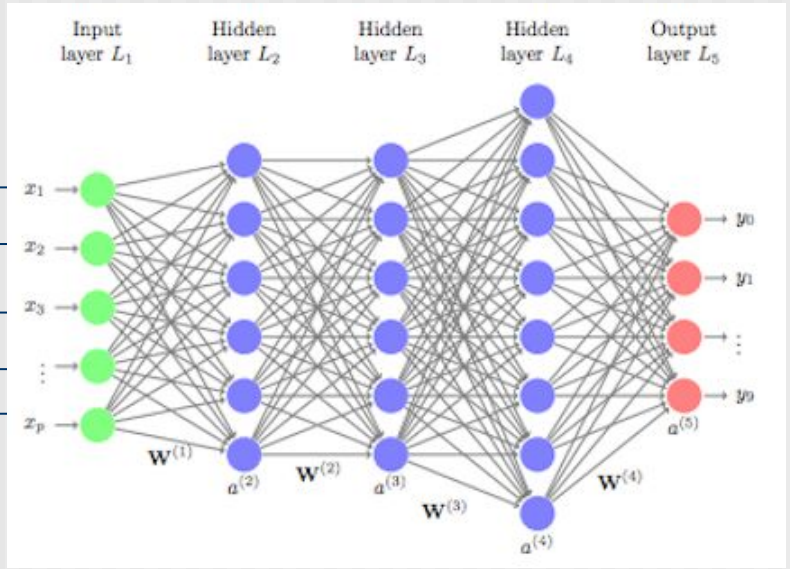
4. Neural Networks based such as *Feedforward Neural Network* (multi-layer perceptron) for digit classification.

http://uc-r.github.io/feedforward_DNN



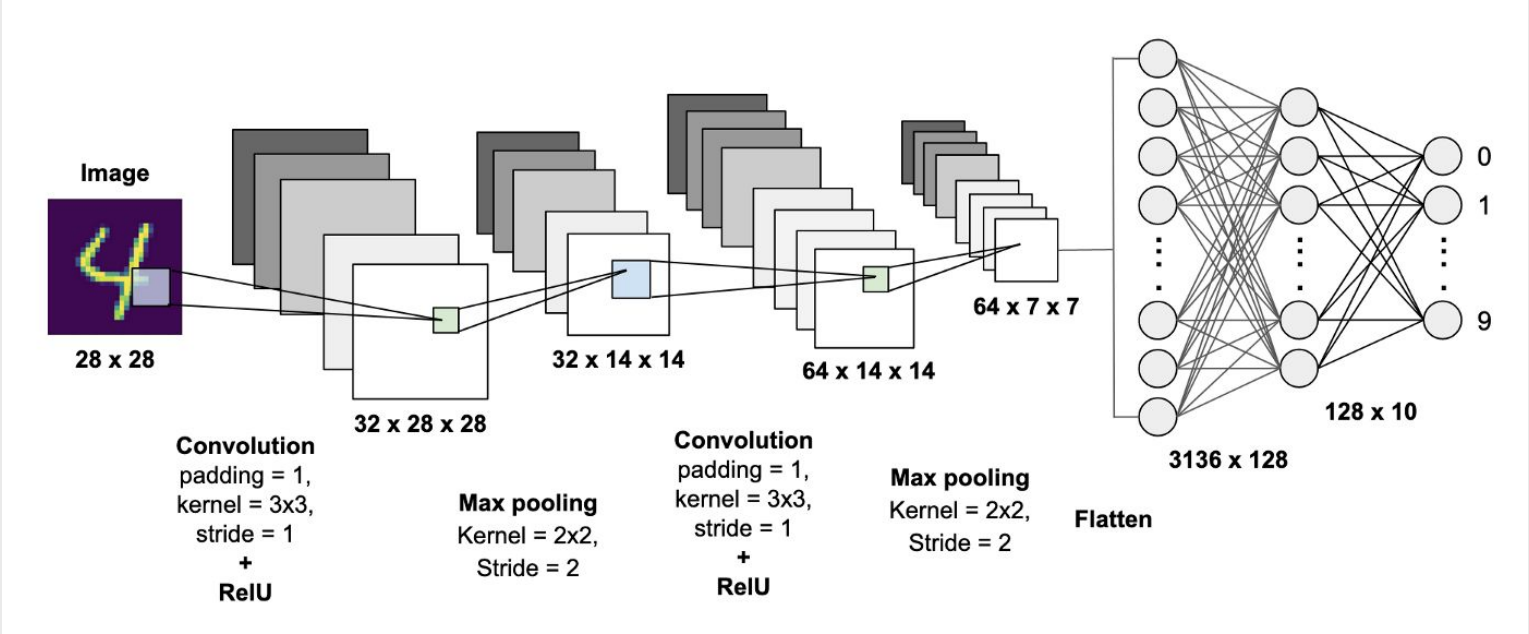
Extracted Feature Vector:

$$\vec{x} = [x_1, x_2, \ldots, x_n]^T$$



Common Pattern Classifiers

4. Neural Networks based such as *Convolutional Neural Networks* or CNN for handwritten digit classification.



After flattening it's like a feedforward NN: 64x7x7 = 3136. 128 hidden layers. 10 output layers

Breast Cancer Classifier `breast_cancer.csv`

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
569	30	malignant	benign																											
17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0.01587	0.03003	0.006193	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189	0
20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186	0.0134	0.01389	0.003532	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902	0
19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.004571	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758	0
11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.009208	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173	0

30 Features:

'mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoothness', 'mean compactness', 'mean concavity','mean concave points', 'mean symmetry', 'mean fractal dimension', 'radius error', 'texture error', 'perimeter error', 'area error', 'smoothness error', 'compactness error', 'concavity error', 'concave points error', 'symmetry error', 'fractal dimension error', 'worst radius', 'worst texture', 'worst perimeter', 'worst area', 'worst smoothness', 'worst compactness', 'worst concavity', 'worst concave points', 'worst symmetry', 'worst fractal dimension'.

2 Classes: 0 - malignant (cancer), 1 - benign (not cancer).

569 rows of data

Iris Dataset

	A	B	C	D	E	F
1	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
2	1	5.1	3.5	1.4	0.2	Iris-setosa
3	2	4.9	3	1.4	0.2	Iris-setosa
52	51	7	3.2	4.7	1.4	Iris-versicolor
53	52	6.4	3.2	4.5	1.5	Iris-versicolor
54	53	6.9	3.1	4.9	1.5	Iris-versicolor
102	101	6.3	3.3	6	2.5	Iris-virginica
103	102	5.8	2.7	5.1	1.9	Iris-virginica
104	103	7.1	3	5.9	2.1	Iris-virginica

4 Features:

sepal length, sepal width, petal length, petal width

3 classes as species: setosa, versicolor, virginica.

Wine Dataset

https://www.goeduhub.com/3426/demonstrate-and-implement-the-naive-bayesian-classifier

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Proanth	Color.int	Hue	OD	Proline
2	1	14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065
3	1	13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050
91	2	12.08	1.33	2.3	23.6	70	2.2	1.59	0.42	1.38	1.74	1.07	3.21	625
92	2	12.08	1.83	2.32	18.5	81	1.6	1.5	0.52	1.64	2.4	1.08	2.27	480
93	2	12	1.51	2.42	22	86	1.45	1.25	0.5	1.63	3.6	1.05	2.65	450
135	3	12.7	3.55	2.36	21.5	106	1.7	1.2	0.17	0.84	5	0.78	1.29	600
136	3	12.51	1.24	2.25	17.5	85	2	0.58	0.6	1.25	5.45	0.75	1.51	650

13 Features:

alcohol %, malic acid, ash, acl, mg, phenols, flavanoids, nonflavanoid phenols, proanth, color, hue, odor, proline.

3 Classes of wine: 1, 2, 3

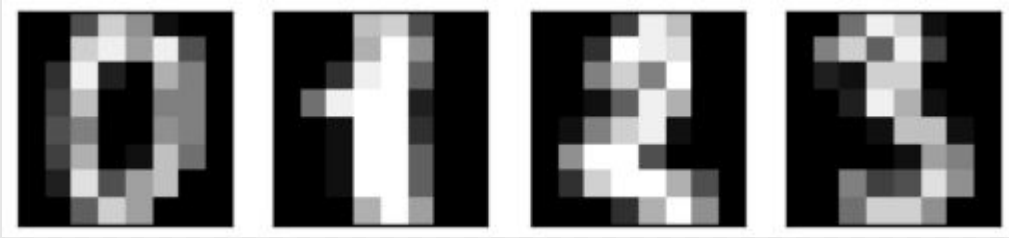
Digits Dataset

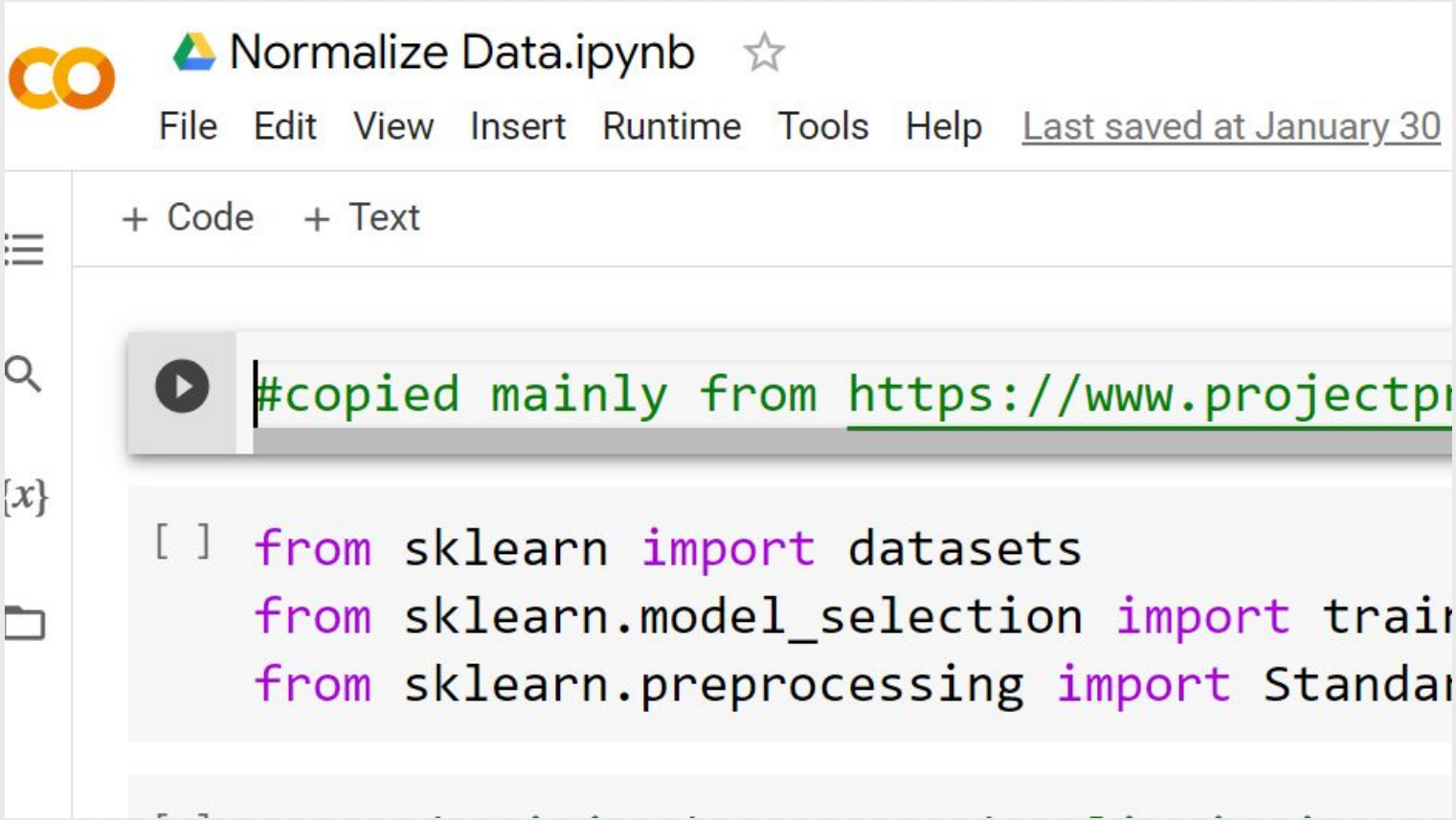
https://www.goeduhub.com/3426/demonstrate-and-implement-the-naive-bayesian-classifier

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM
1	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	digit
2	15	5	0	0	3	15	2	0	11	8	0	0	4	12	0	0	8	8	0	0	5	8	0	0	9	8	0	0	4	11	0	1	12	7	0	0	2	14	5	10	12	0	0	0	0	6	13	10	0	0	0	0
3	9	0	0	0	0	3	15	16	6	0	0	0	7	15	16	16	2	0	0	0	0	1	16	16	3	0	0	0	0	1	16	16	6	0	0	0	0	1	16	16	6	0	0	0	0	0	11	16	10	0	0	1
4	14	0	0	0	0	8	13	8	16	0	0	0	0	1	6	15	11	0	0	0	1	8	13	15	1	0	0	0	9	16	16	5	0	0	0	0	3	13	16	16	11	5	0	0	0	0	3	11	16	9	0	2
5	4	0	0	0	2	1	13	13	0	0	0	0	0	2	15	11	1	0	0	0	0	0	1	12	12	1	0	0	0	0	0	1	10	8	0	0	0	8	4	5	14	9	0	0	0	7	13	13	9	0	0	3
6	0	0	0	0	0	1	13	6	2	2	0	0	0	7	15	0	9	8	0	0	5	16	10	0	16	6	0	0	4	15	16	13	16	1	0	0	0	0	3	15	10	0	0	0	0	0	2	16	4	0	0	4
7	14	0	0	0	0	13	16	15	10	1	0	0	0	11	16	16	7	0	0	0	0	0	4	7	16	7	0	0	0	0	0	4	16	9	0	0	0	5	4	12	16	4	0	0	0	9	16	16	10	0	0	5
8	0	0	0	0	0	13	16	3	0	0	0	0	0	14	13	0	0	0	0	0	0	15	12	7	2	0	0	0	0	13	16	13	16	3	0	0	0	7	16	11	15	8	0	0	0	1	9	15	11	3	0	6
9	11	12	0	0	0	0	0	8	13	1	0	0	4	8	8	15	15	6	0	0	2	11	15	15	4	0	0	0	0	0	16	5	0	0	0	0	0	9	15	1	0	0	0	0	0	13	5	0	0	0	0	7
10	12	0	0	0	0	9	10	0	15	4	0	0	0	3	16	12	14	2	0	0	0	4	16	16	2	0	0	0	3	16	8	10	13	2	0	0	1	15	1	3	16	8	0	0	0	11	16	15	11	1	0	8
11	13	0	0	0	3	16	12	10	14	0	0	0	1	16	1	12	15	0	0	0	0	13	16	9	15	2	0	0	0	0	3	0	9	11	0	0	0	0	0	9	15	4	0	0	0	9	12	13	3	0	0	9
12	14	6	0	0	2	16	10	0	9	9	0	0	1	16	4	0	8	8	0	0	4	16	4	0	8	8	0	0	1	16	5	1	11	3	0	0	0	12	12	10	10	0	0	0	0	1	10	13	3	0	0	0
13	16	2	0	0	0	0	14	16	12	0	0	0	1	10	16	16	12	0	0	0	3	12	14	16	9	0	0	0	0	0	5	16	15	0	0	0	0	0	4	16	14	0	0	0	0	0	1	13	16	1	0	1

64 Features: pixel 1, pixel 2, pixel 3, …, pixel 64 for 8x8 image of digits. Each pixel has intensity 0-16.

10 classes for digits: 0, 1, …, 9.




Normalize Data.ipynb
File Edit View Insert Runtime Tools Help Last saved at January 30
+ Code + Text
#copied mainly from https://www.projectpr
[ ] from sklearn import datasets
from sklearn.model_selection import trair
from sklearn.preprocessing import Standar

Core Neural Network Models for ML

1. Multilayer Perceptrons (MLP). Most Basic. Used for supervised learning tasks such as image and speech recognition, and natural language processing.
2. Convolutional Neural Networks (CNN). Ideal for data with grid-like topology, suited for images and video.
3. Recurrent Neural Networks (RNN). These are neural networks that are designed to handle sequential data, such as time series or natural language. Ideal for language translation, speech recognition, and text generation.
    a. Gated Recurrent Units (GRU). A variant of RNN that uses gating mechanism to control the flow of information, which allows it to better handle long-term dependencies in sequential data.
    b. Long Short-Term Memory (LSTM). A Variant of RNN that uses a memory cell to store information and control gates to regulate the flow of information, allowing it to better handle long-term dependencies in sequential data.
4. Transformer Neural Networks (TNN). Neural Networks designed to handle sequential data, such as natural language. GPT: Generative Pre-trained Transformer. RNNs process sequences in a sequential manner and struggle with long sequences, while Transformers use self-attention to process input sequences in parallel and effectively handle long sequences.

Quiz 2. Naive Bayes and KNN Classifiers