AI VIETNAM
All-in-One Course

Introduction to KNN

Quang-Vinh Dinh
Ph.D. in Computer Science

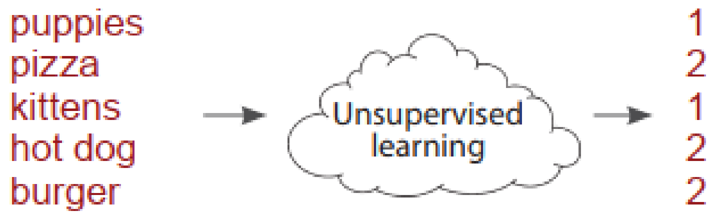
Year 2023

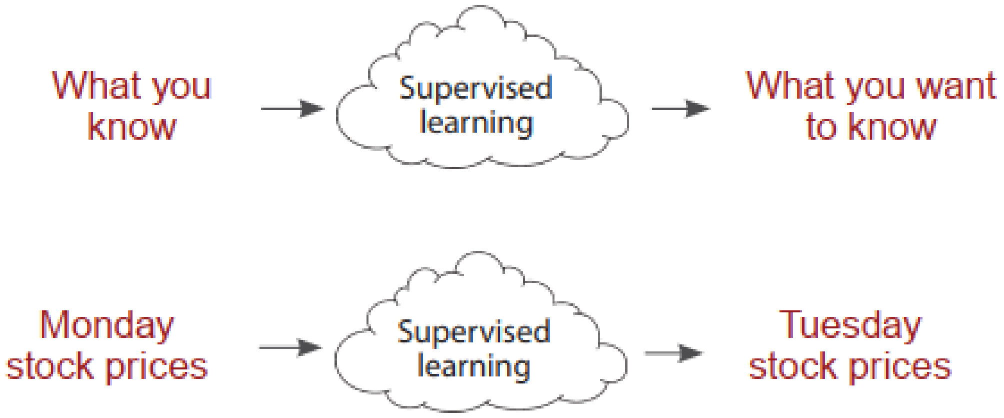
Machine Learning

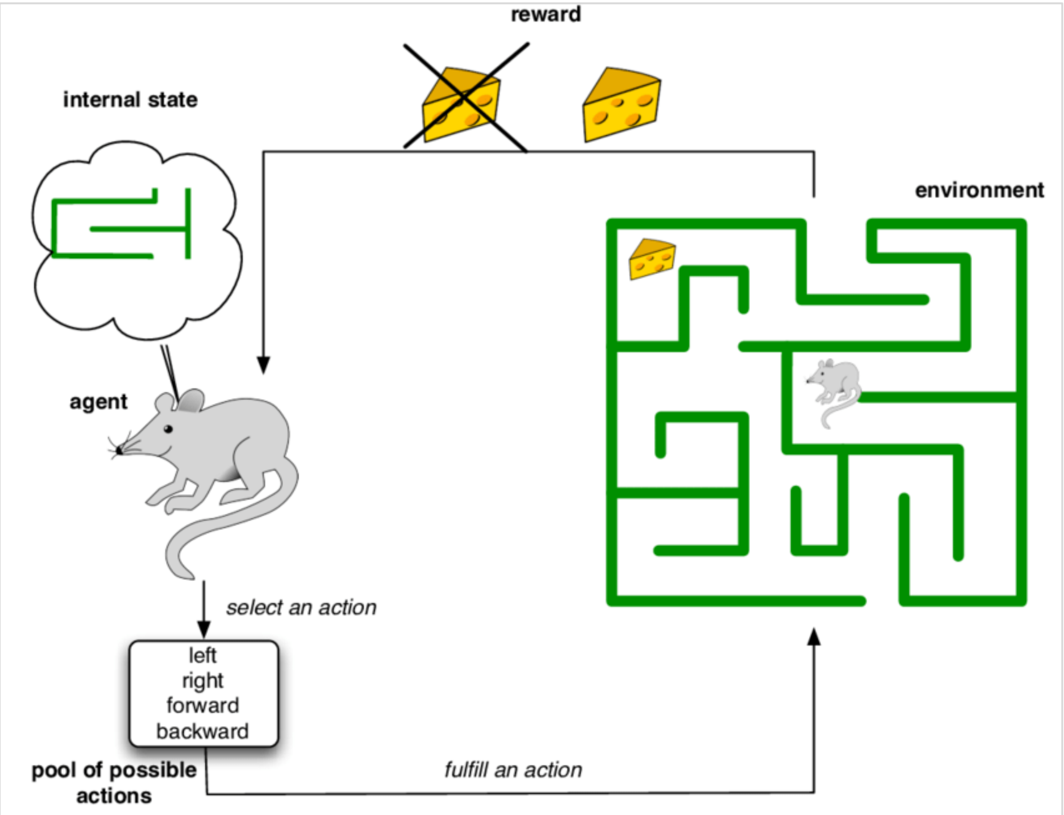
❖ Definition

What is machine learning?

> A field of study that gives computers the ability to learn without being explicitly programmed.
>
> —Attributed to Arthur Samuel







Machine Learning

❖ Supervised learning

❖ Data

Input and output data are provided

- Training data
- Cats
- Dogs

From Cat-Dog dataset

Machine Learning

Supervised learning

Data

Training data

Used to teach

Machine learning model

This is a cat

This is a dog

Machine learning model

Training phase

From Cat-Dog dataset

Testing data (≠ training data)

Machine learning model

Make decision

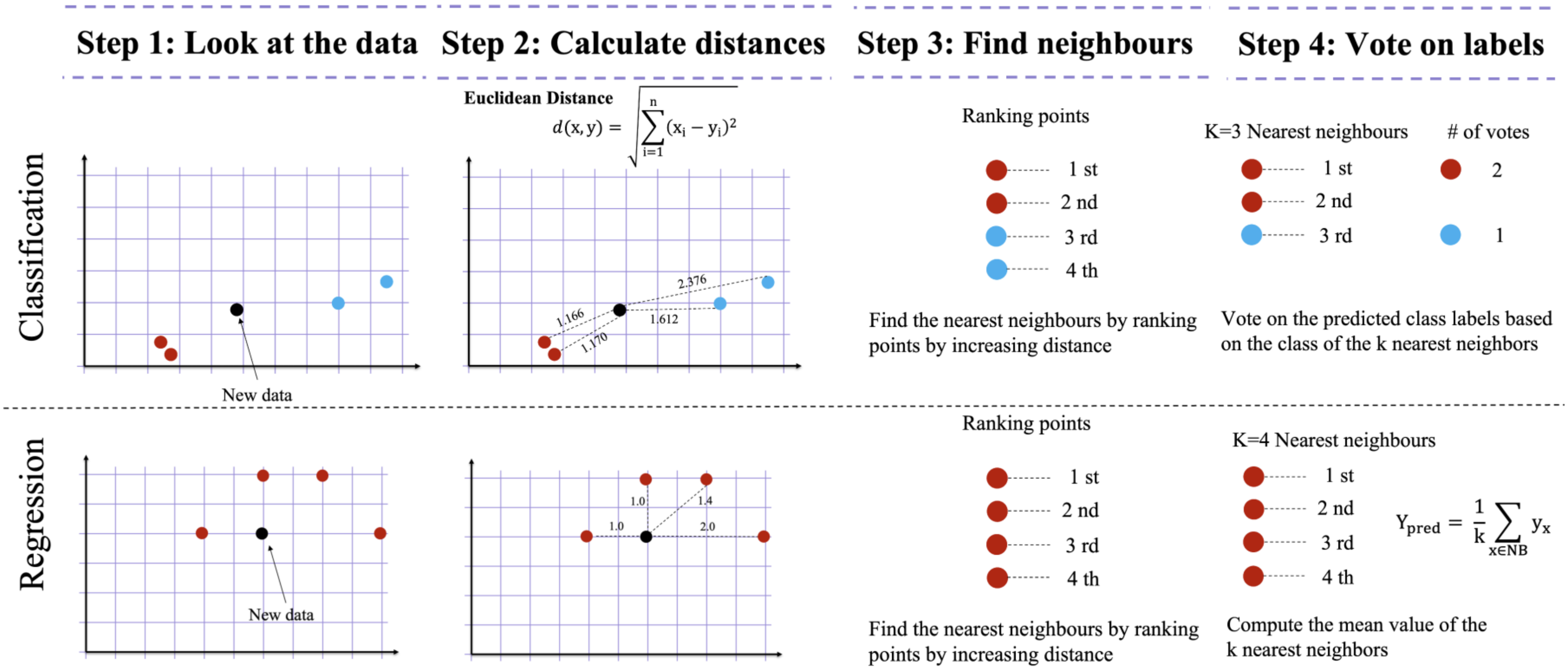
Cat or Dog?

Testing phase

K-Nearest Neighbors

Overview

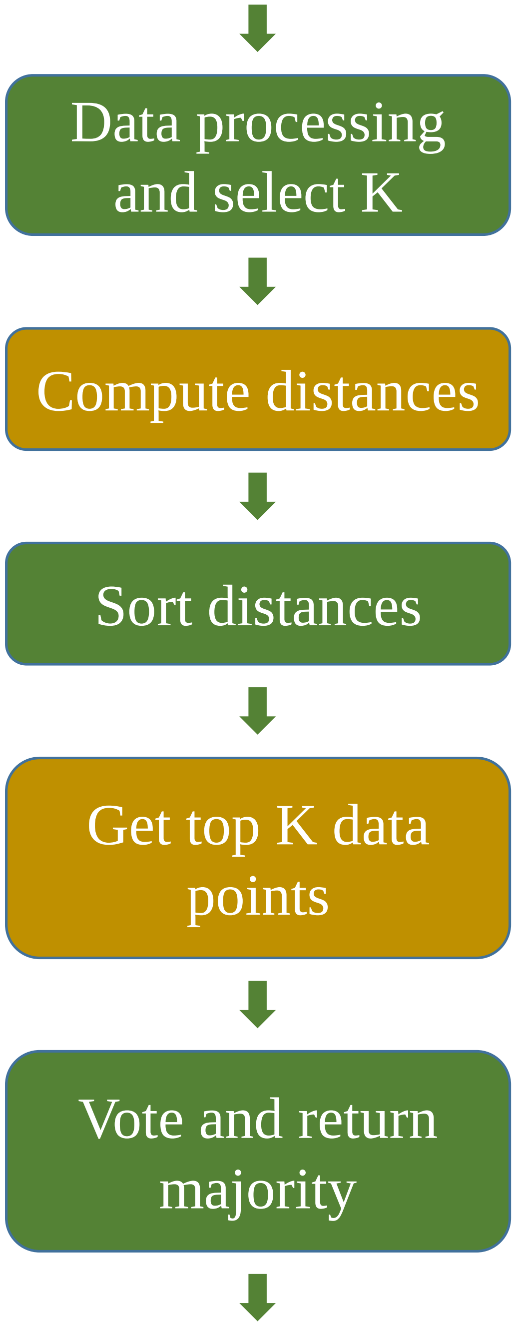
From TA Thái



K-Nearest Neighbors

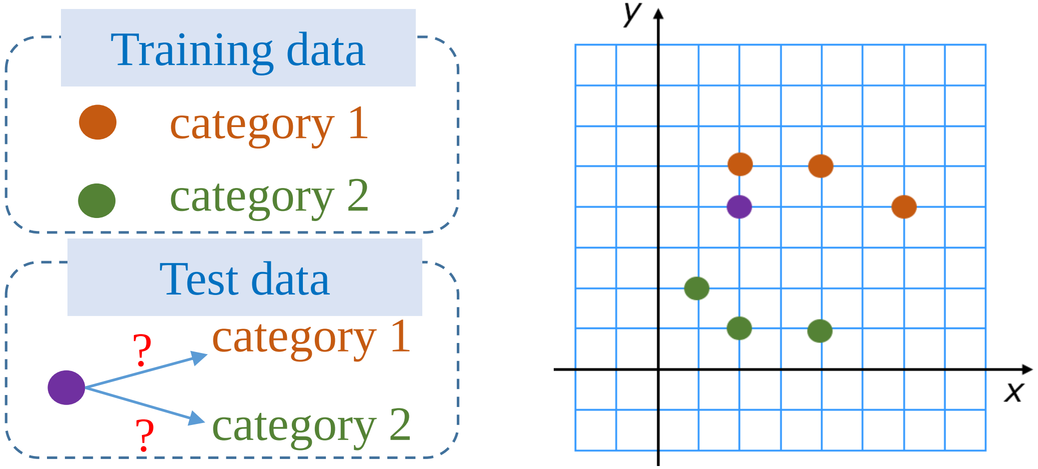
❖ Procedure

1. Initialize the value of k
2. Iterate from 1 to total number of training data points. Calculate the distance between test data and each row of training dataset.
3. Sort the calculated distances in ascending order based on distance values
4. Get top k rows from the sorted array
5. Get the most frequent class of these rows
6. Return the predicted class

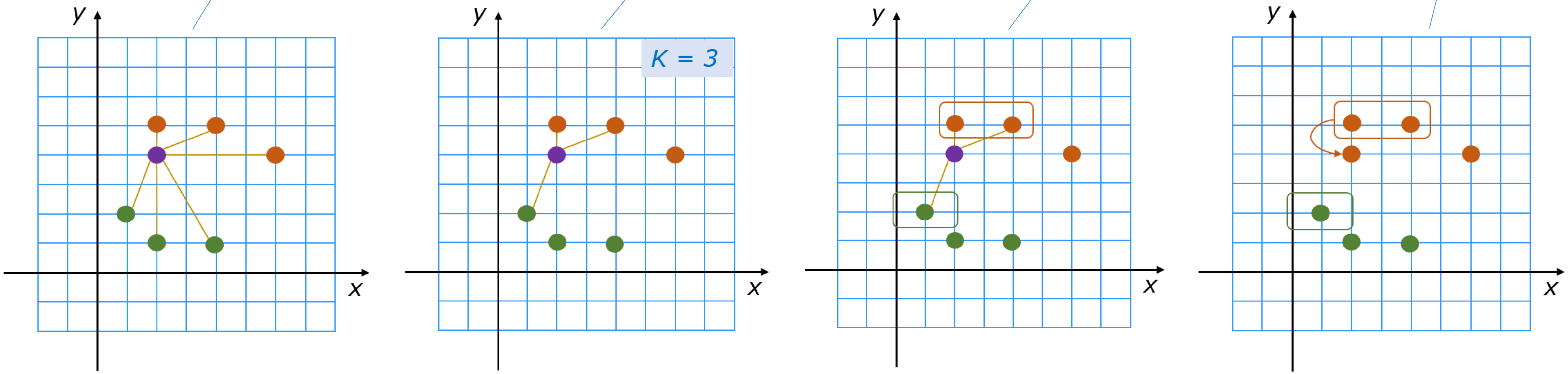


K-NN Algorithm

Petal_Length (cm)	Petal_Width (cm)	Label
1.4	0.2	0
1.3	0.4	0
1.4	0.3	0
4	1	1
4.7	1.4	1
3.6	1.3	1



Prepare data and select K → Compute distances between a testing point and points in training data → Take the K nearest neighbors → Voting → Output



KNN

❖ Example

Petal_Length	Label	Distance
1.4	0	1
1	0	1.4
1.5	0	0.9
3.1	1	0.7
3.7	1	1.3
4.1	1	1.7

New input data
x_test = 2.4

Petal_Length	Label	Distance
1.4	0	1
1	0	1.4
1.5	0	0.9
3.1	1	0.7
3.7	1	1.3
4.1	1	1.7

k=1
→ y_test = 1

k=3
→ y_test = 0

KNN

❖ Example

Petal_Length	Petal_Width	Label	Distance
1.4	0.2	0	1.166
1.3	0.4	0	1.17
1.4	0.3	0	1.118
4	1	1	1.612
4.7	1.4	1	2.376
3.6	1.3	1	1.3

New input data
x_test = (2.4, 0.8)

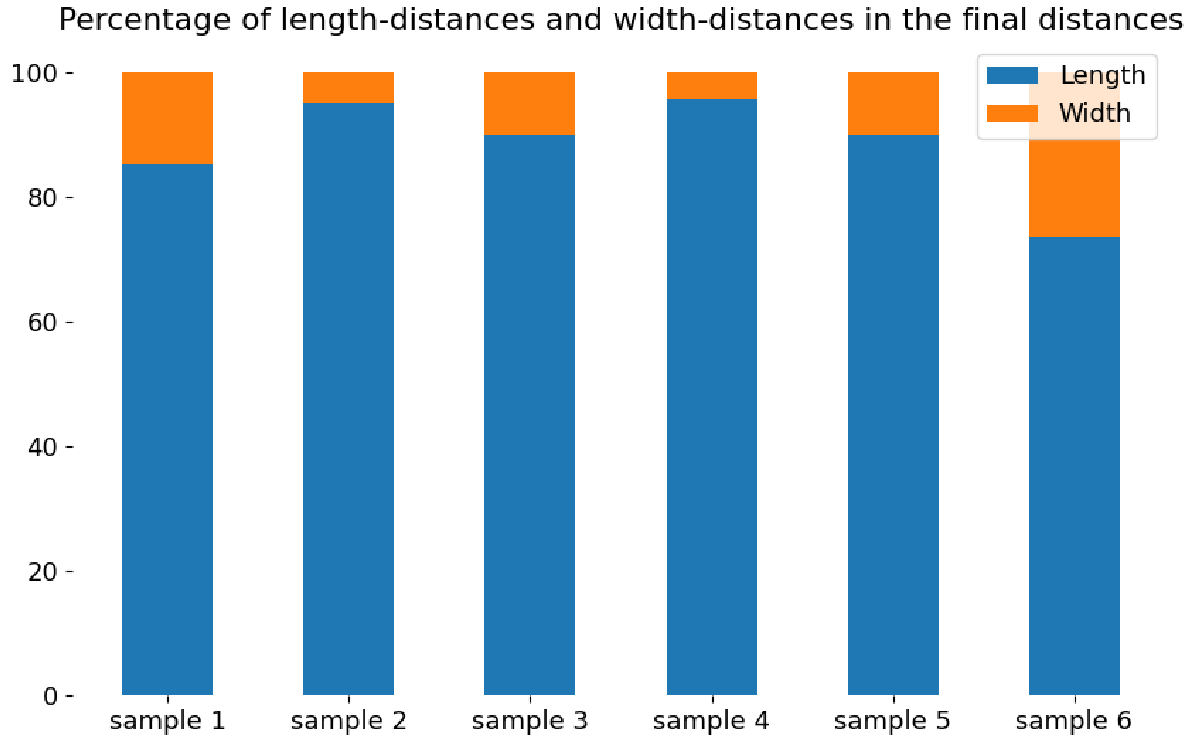
K = 1

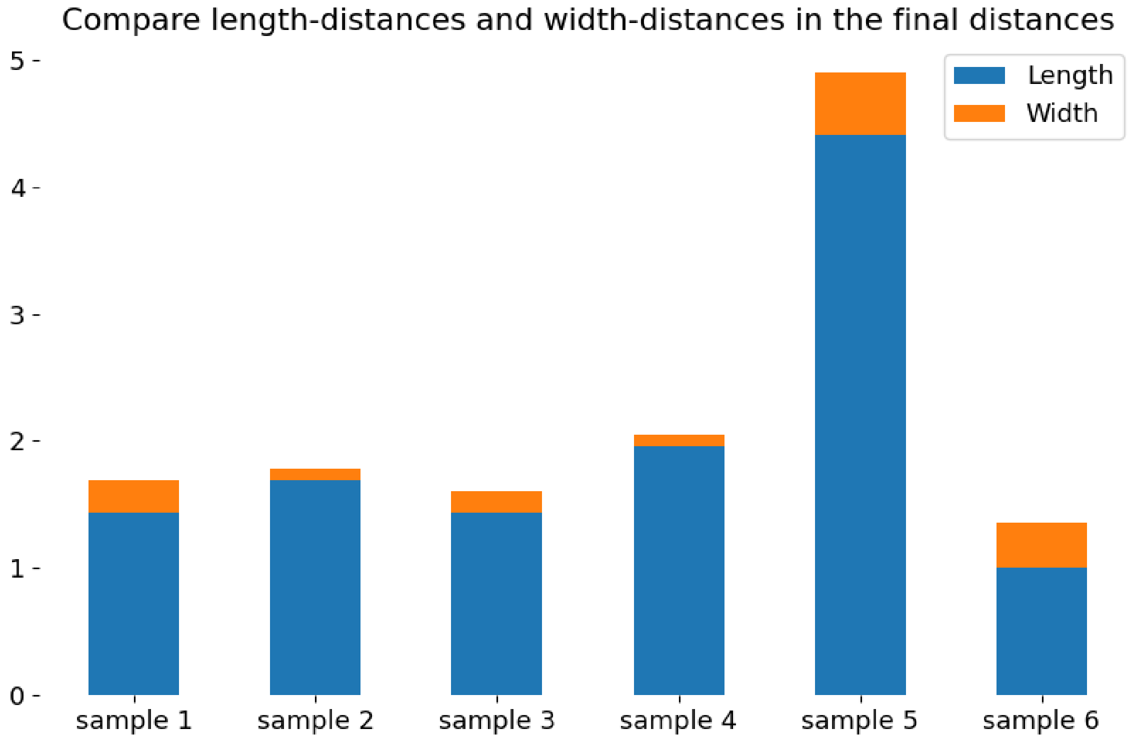
K = 3

Example (1)

Unnormalized 2D data

Petal_Length (cm)	Petal_Width (cm)	Label	Length_distance	Width_distance	Distance
1.4	0.2	0	1.44	0.25	1.3
1.3	0.4	0	1.69	0.09	1.33
1.4	0.3	0	1.44	0.16	1.26
4	1	1	1.96	0.09	1.43
4.7	1.4	1	4.41	0.49	2.21
3.6	1.3	1	1	0.36	1.16

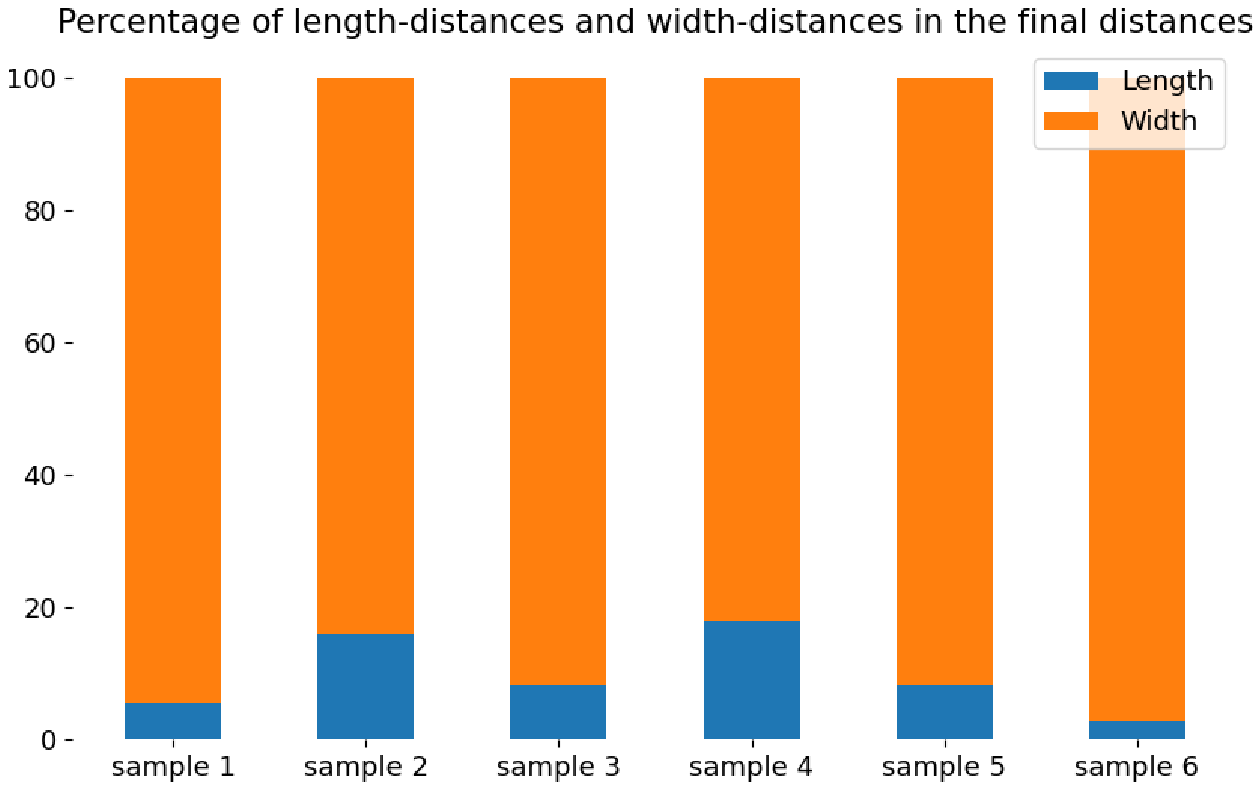


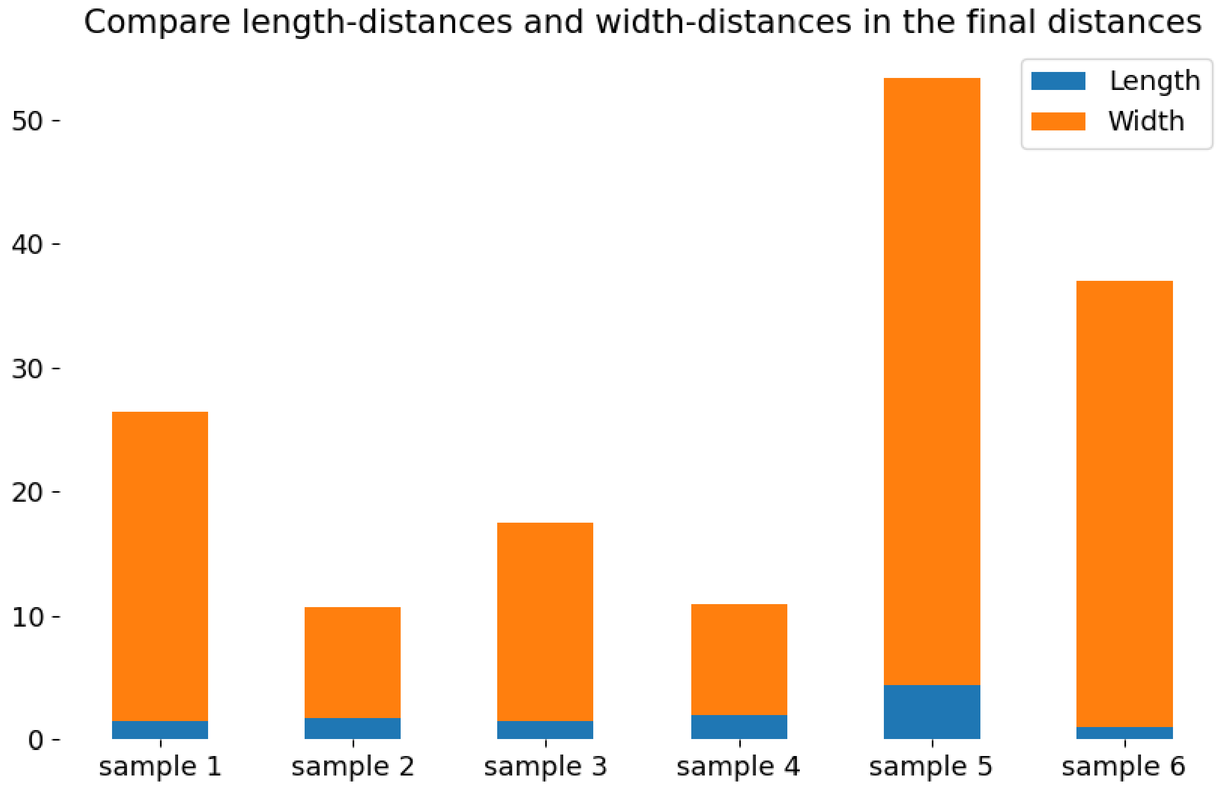


Example (2)

Unnormalize
d 2D data

Petal_Length (cm)	Petal_Width (mm)	Label	Length_distance	Width_distance	Distance
1.4	2	0	1.44	25	5.14
1.3	4	0	1.69	9	3.26
1.4	3	0	1.44	16	4.17
4	10	1	1.96	9	3.31
4.7	14	1	4.41	49	7.31
3.6	13	1	1	36	6.08





Data normalization

$$d = \sqrt{\left(x_1^{test} - x_1^{train}\right)^2 + \left(x_2^{test} - x_2^{train}\right)}$$

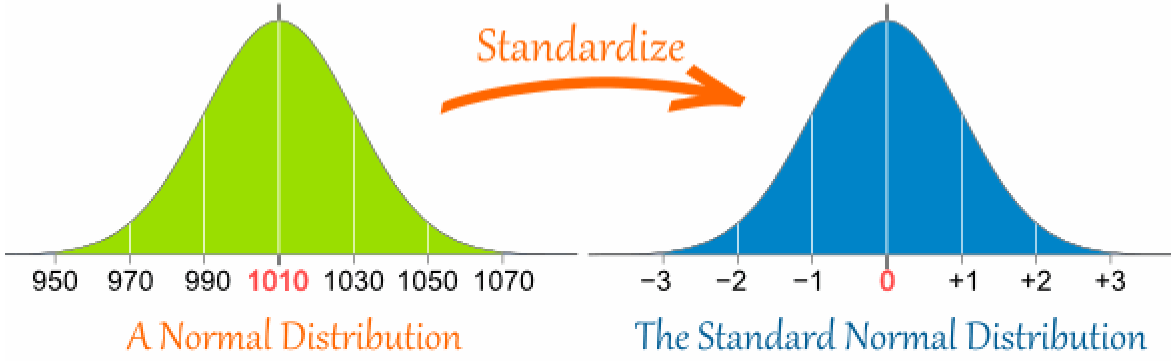
Petal_Length (cm) (x_1)	Petal_Width (mm) (x_2)	Label	Distance (d)
1.4	2	0	5.14
1.3	4	0	3.26
1.4	3	0	4.17
4	10	1	3.31
4.7	14	1	7.31
3.6	13	1	6.08

Training Data 1

Petal_Length (cm) (x_1)	Petal_Width (cm) (x_2)	Label	Distance (d)
1.4	0.2	0	1.3
1.3	0.4	0	1.33
1.4	0.3	0	1.26
4	1	1	1.43
4.7	1.4	1	2.21
3.6	1.3	1	1.16

Training Data 2

$$x = \frac{x - \bar{x}}{\sigma}$$



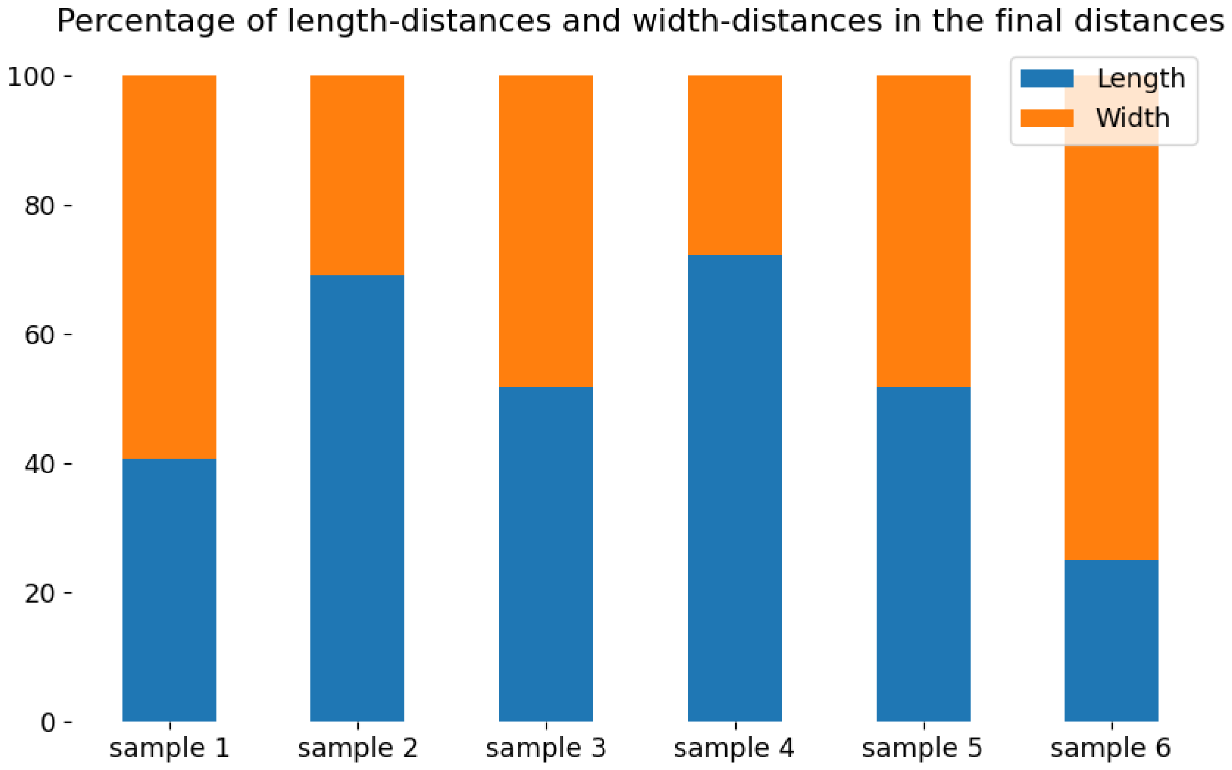
https://www.quora.com/Why-is-the-mean-of-normal-distribution-equal-to-zero

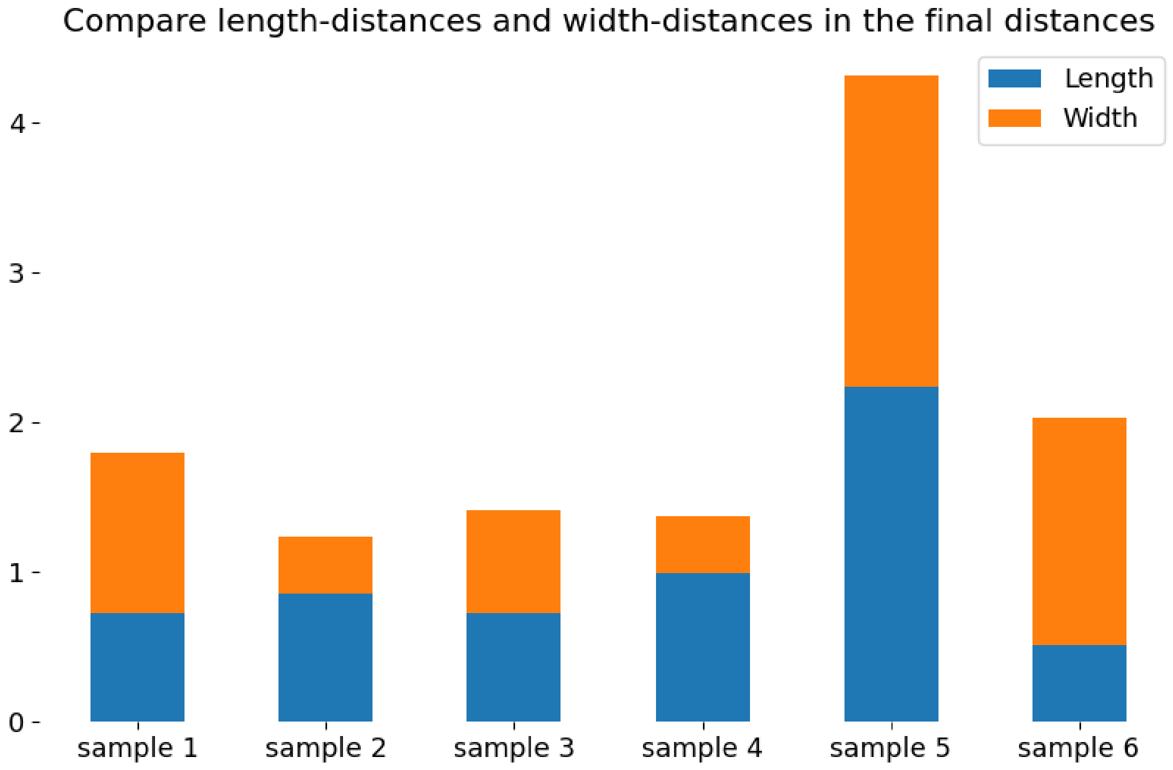
Petal_Length	Petal_Width	Label	Distance
-0.949	-1.167	0	1.338
-1.021	-0.755	0	1.113
-0.949	-0.961	0	1.187
0.901	0.481	1	1.172
1.4	1.304	1	2.077
0.617	1.098	1	1.426

Example (3)

normalized 2D data

Petal_Length	Petal_Width	Label	Length_distance	Width_distance	Distance
-0.949	-1.167	0	0.73	1.061	1.338
-1.021	-0.755	0	0.856	0.382	1.113
-0.949	-0.961	0	0.73	0.679	1.187
0.901	0.481	1	0.993	0.382	1.172
1.4	1.304	1	2.236	2.08	2.077
0.617	1.098	1	0.507	1.528	1.426





KNN

❖ **Implementation**

```python
from sklearn import neighbors, datasets
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd

data = pd.read_csv('iris_2D.csv')

# get x
x_data = data[['Petal_Length', 'Petal_Width']].to_numpy()
x_data = x_data.reshape(6, 2)

# get y
y_data = data['Label'].to_numpy()

# training
classifier = KNeighborsClassifier(n_neighbors=1)
classifier.fit(x_data, y_data)

# prediction
x_test = [[2.6, 0.7]]
y_pred = classifier.predict(x_test)
print(y_pred)
```

Text classification with KNN

Vectorization with Bag of Words

Text Representation

❖ Bag of words

Corpus

doc1 = “deep learning book” → ['deep', 'learning', 'book']

doc2 = “machine learning algorithm” → ['machine', 'learning', 'algorithm']

doc3 = “learning ai from scratch” → ['learning', 'ai', 'from', 'scratch']

doc4 = “ai vietnam” → ['ai', 'vietnam']

Tokenization

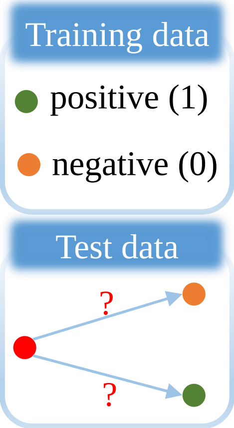
Vocabulary =

deep	learning	book	machine	algorithm	ai	from	scratch	vietnam

☞ **Given a string =** “vietnam machine learning deep learning book”

	deep	learning	book	machine	algorithm	ai	from	scratch	vietnam
BoW	1	2	1	1	0	0	0	0	1
Binary BoW	1	1	1	1	0	0	0	0	1

Doc	Label
góp gió gặt bão	1
có làm mới có ăn	1
đất lành chim đậu	1
ăn cháo đá bát	0
gậy ông đập lưng ông	0
qua cầu rút ván	0



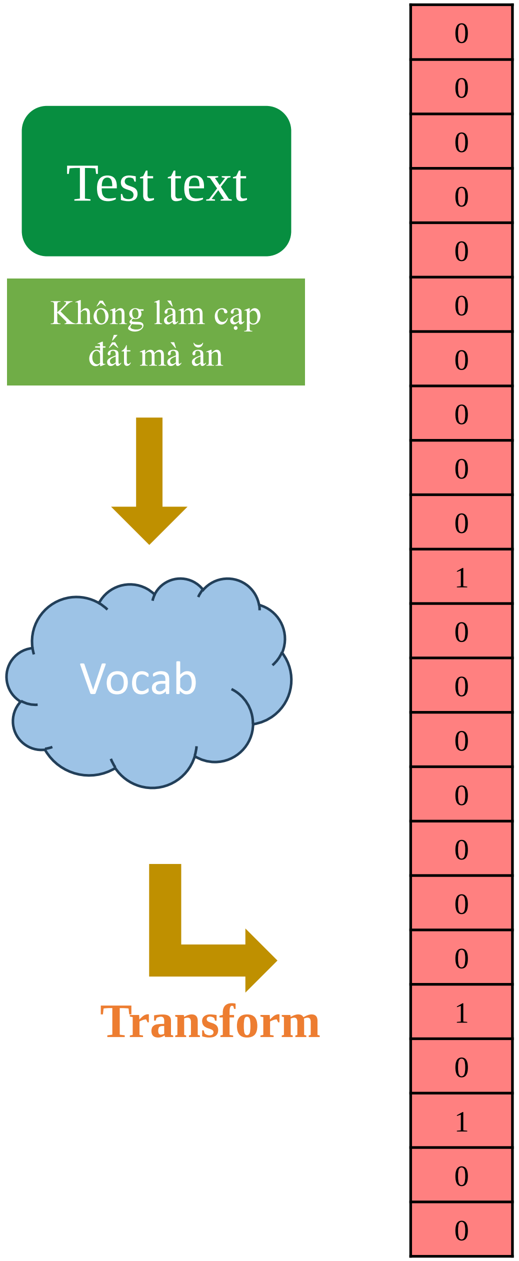
Tokenization

Vocabulary
bát
bão
chim
cháo
có
cầu
gió
góp
gậy
gặt
làm
lành
lưng
mới
qua
rút
ván
ông
ăn
đá
đất
đập
đậu

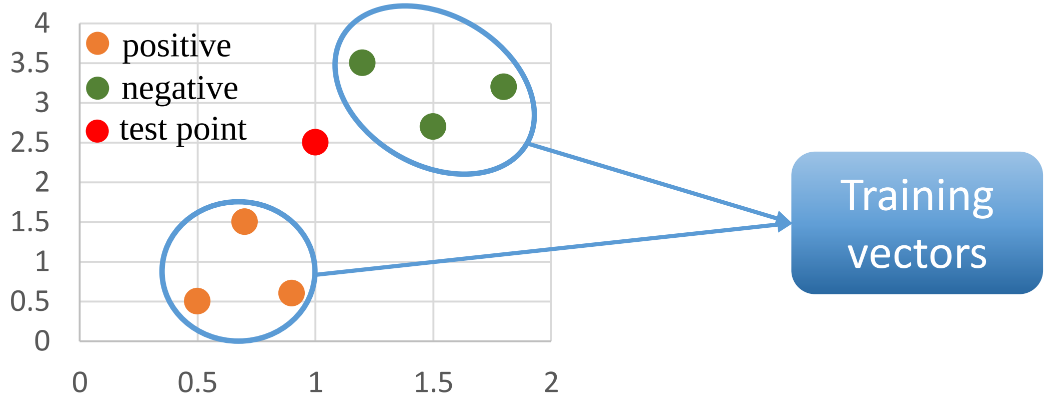
gậy ông đập lưng ông

	doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
bát	0	0	0	1	0	0
bão	1	0	0	0	0	0
chim	0	0	1	0	0	0
cháo	0	0	0	1	0	0
có	0	2	0	0	0	0
cầu	0	0	0	0	0	1
gió	1	0	0	0	0	0
góp	1	0	0	0	0	0
gậy	0	0	0	0	1	0
gặt	1	0	0	0	0	0
làm	0	1	0	0	0	0
lành	0	0	1	0	0	0
lưng	0	0	0	0	1	0
mới	0	1	0	0	0	0
qua	0	0	0	0	0	1
rút	0	0	0	0	0	1
ván	0	0	0	0	0	1
ông	0	0	0	0	2	0
ăn	0	1	0	1	0	0
đá	0	0	0	1	0	0
đất	0	0	1	0	0	0
đập	0	0	0	0	1	0
đậu	0	0	1	0	0	0

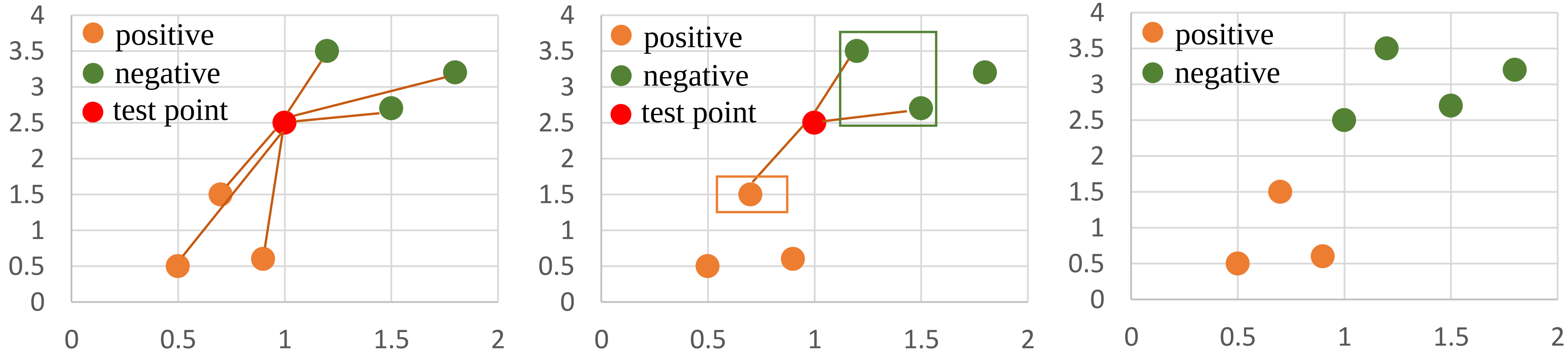
BoW vectors



Doc	Label	Distance
góp gió gặt bão	1	2.645
có làm mới có ăn	1	2.449
đất lành chim đậu	1	2.236
ăn cháo đá bát	0	2.236
gậy ông đập lưng ông	0	3.162
qua cầu rút ván	0	2.645



Select K (K=3) → Compute distance between test vector and training vectors → Take the K nearest neighbors and Voting → Output



Text classification with KNN

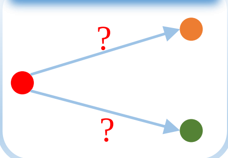
TF-IDF vectorizer (extension)

Doc	Label
góp gió gặt bão	0
có làm mới có ăn	0
đất lành chim đậu	0
ăn cháo đá bát	1
gậy ông đập lưng ông	1
qua cầu rút ván	1

Training data

- positive (1)
- negative (0)

Test data



Clean data

Build Doc-Term matrix

Compute IDF vector

	doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
bát	0	0	0	1	0	0
bão	1	0	0	0	0	0
chim	0	0	1	0	0	0
cháo	0	0	0	1	0	0
có	0	2	0	0	0	0
cầu	0	0	0	0	0	1
gió	1	0	0	0	0	0
góp	1	0	0	0	0	0
gậy	0	0	0	0	1	0
gặt	1	0	0	0	0	0
làm	0	1	0	0	0	0
lành	0	0	1	0	0	0
lưng	0	0	0	0	1	0
mới	0	1	0	0	0	0
qua	0	0	0	0	0	1
rút	0	0	0	0	0	1
ván	0	0	0	0	0	1
ông	0	0	0	0	2	0
ăn	0	1	0	1	0	0
đá	0	0	0	1	0	0
đất	0	0	1	0	0	0
đập	0	0	0	0	1	0
đậu	0	0	1	0	0	0

Doc-term matrix

$$\log\left(\frac{6+1}{1+1}\right)+1$$

$$IDF_t = \log\left(\frac{N+1}{DF_t+1}\right)+1$$

Smothing

$$\log\left(\frac{6+1}{2+1}\right)+1$$

N = number of documents

IDF vector
2.25
2.25
2.25
2.25
1.84
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
1.84
1.84
2.25
2.25
2.25
2.25

	doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
bát	0	0	0	1	0	0
bão	1	0	0	0	0	0
chim	0	0	1	0	0	0
cháo	0	0	0	1	0	0
có	0	2	0	0	0	0
cầu	0	0	0	0	0	1
gió	1	0	0	0	0	0
góp	1	0	0	0	0	0
gậy	0	0	0	0	1	0
gặt	1	0	0	0	0	0
làm	0	1	0	0	0	0
lành	0	0	1	0	0	0
lưng	0	0	0	0	1	0
mới	0	1	0	0	0	0
qua	0	0	0	0	0	1
rút	0	0	0	0	0	1
ván	0	0	0	0	0	1
ông	0	0	0	0	2	0
ăn	0	1	0	1	0	0
đá	0	0	0	1	0	0
đất	0	0	1	0	0	0
đập	0	0	0	0	1	0
đậu	0	0	1	0	0	0

Doc-term matrix

$$TF_{(t,d)} = log(count(t,d) + 1)$$

$$log(0 + 1)$$

$$log(1 + 1)$$

Compute TF matrix

doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
0.0	0	0	0.69	0	0
0.69	0	0	0	0	0
0	0	0.69	0	0	0
0	0	0	0.69	0	0
0	1.09	0	0	0	0
0	0	0	0	0	0.69
0.69	0	0	0	0	0
0.69	0	0	0	0	0
0	0	0	0	0.69	0
0.69	0	0	0	0	0
0	0.69	0	0	0	0
0	0	0.69	0	0	0
0	0	0	0	0.69	0
0	0.69	0	0	0	0
0	0	0	0	0	0.69
0	0	0	0	0	0.69
0	0	0	0	0	0.69
0	0	0	0	1.09	0
0	0.69	0	0.69	0	0
0	0	0	0.69	0	0
0	0	0.69	0	0	0
0	0	0	0	0.69	0
0	0	0.69	0	0	0

TF matrix

doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
0.0	0	0	0.69	0	0
0.69	0	0	0	0	0
0	0	0.69	0	0	0
0	0	0	0.69	0	0
0	1.09	0	0	0	0
0	0	0	0	0	0.69
0.69	0	0	0	0	0
0.69	0	0	0	0	0
0	0	0	0	0.69	0
0.69	0	0	0	0	0
0	0.69	0	0	0	0
0	0	0.69	0	0	0
0	0	0	0	0.69	0
0	0.69	0	0	0	0
0	0	0	0	0	0.69
0	0	0	0	0	0.69
0	0	0	0	0	0.69
0	0	0	0	1.09	0
0	0.69	0	0.69	0	0
0	0	0	0.69	0	0
0	0	0.69	0	0	0
0	0	0	0	0.69	0
0	0	0.69	0	0	0

TF matrix

x

IDF vector
2.25
2.25
2.25
2.25
1.84
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
2.25
1.84
1.84
2.25
2.25
2.25
2.25

=

0
0
0
0
2.02
0
0
0
0
0
1.56
0
0
1.56
0
0
0
0
1.28
0
0
0
0

$$TFIDF_{(t,d)} = TF_{(t,d)} \times IDF_t$$

$$L2\_norm(v) = \frac{v}{\|v\|_2}$$

$$\frac{2.02}{\sqrt{(2.02^2 + 1.56^2 + 1.56^2 + 1.28^2)}}$$

$$\frac{1.56}{\sqrt{(2.02^2 + 1.56^2 + 1.56^2 + 1.28^2)}}$$

Compute and normalize TF-IDF vectors

doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
0.0	0	0	0.52	0	0
0.5	0	0	0	0	0
0	0	0.5	0	0	0
0	0	0	0.52	0	0
0	0.62	0	0	0	0
0	0	0	0	0	0.5
0.5	0	0	0	0	0
0.5	0	0	0	0	0
0	0	0	0	0.46	0
0.5	0	0	0	0	0
0	0.47	0	0	0	0
0	0	0.5	0	0	0
0	0	0	0	0.46	0
0	0.47	0	0	0	0
0	0	0	0	0	0.5
0	0	0	0	0	0.5
0	0	0	0	0	0.5
0	0	0	0	0.6	0
0	0.39	0	0.42	0	0
0	0	0	0.52	0	0
0	0	0.5	0	0	0
0	0	0	0	0.46	0
0	0	0.5	0	0	0

TF-IDF Matrix

Test text

Không làm cạp đất mà ăn

Compute TF

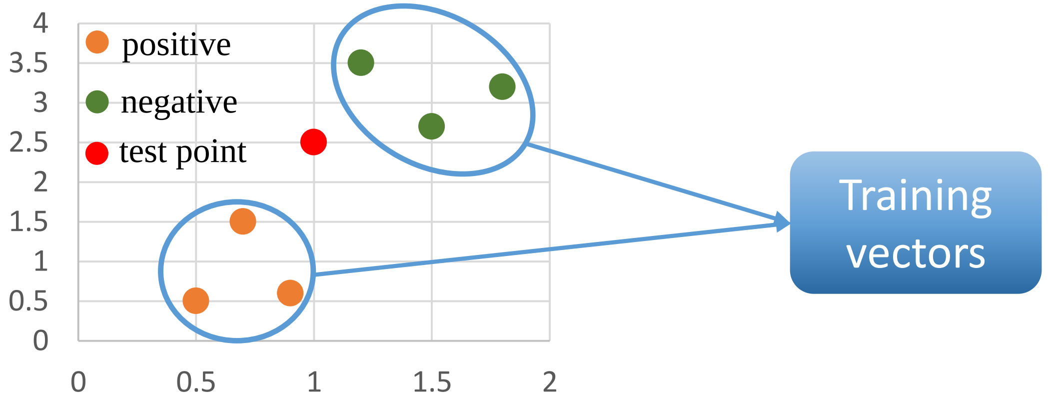
Compute TF-IDF

Normalize

Test vector

0
0
0
0
0
0
0
0
0
0
0.61
0
0
0
0
0
0
0
0.5
0
0.61
0
0

Doc	Label	Distance
góp gió gặt bão	1	1.41
có làm mới có ăn	1	1.01
đất lành chim đậu	1	1.17
ăn cháo đá bát	0	1.25
gậy ông đập lưng ông	0	1.41
qua cầu rút ván	0	1.41

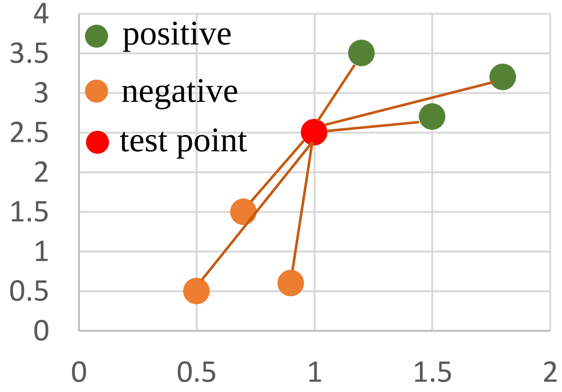


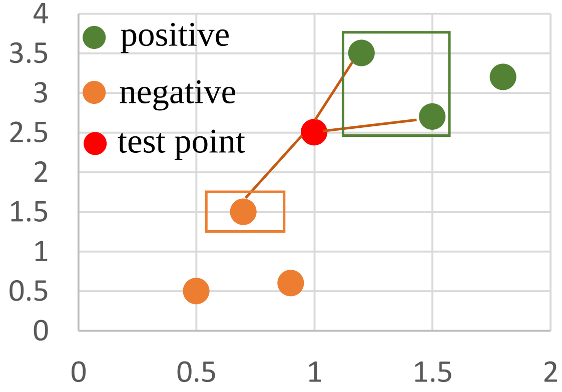
Select K (K=3)

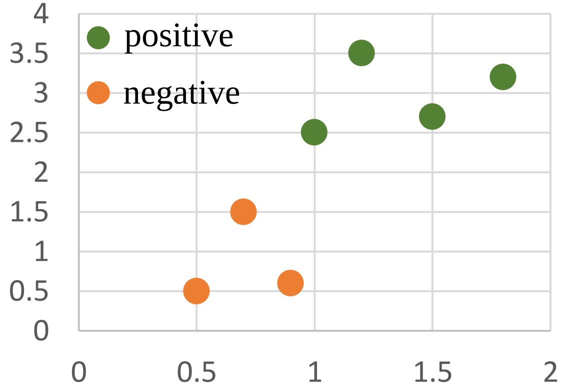
Compute distance between test vector and training vectors

Take the K nearest neighbors and Voting

Output

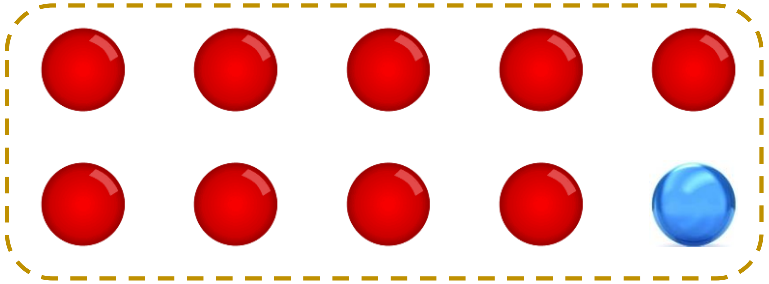






Entropy

❖ Motivation



A: Get a red ball

B: Get a blue ball

$$p(A) = \frac{9}{10} = 0.9$$

$$p(B) = \frac{1}{10} = 0.1$$

E: Pick a ball from the basket

Experiment 1

Got a red ball

Experiment 2

Got a blue ball

Which experiment makes you more surprised?

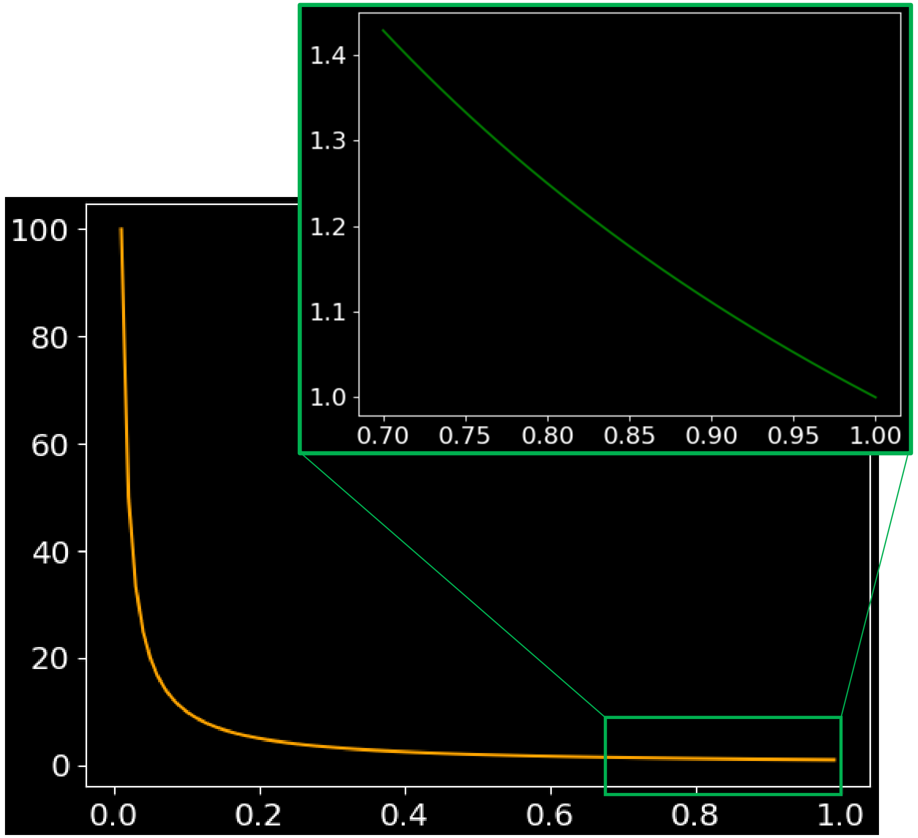
How to measure the surprises?

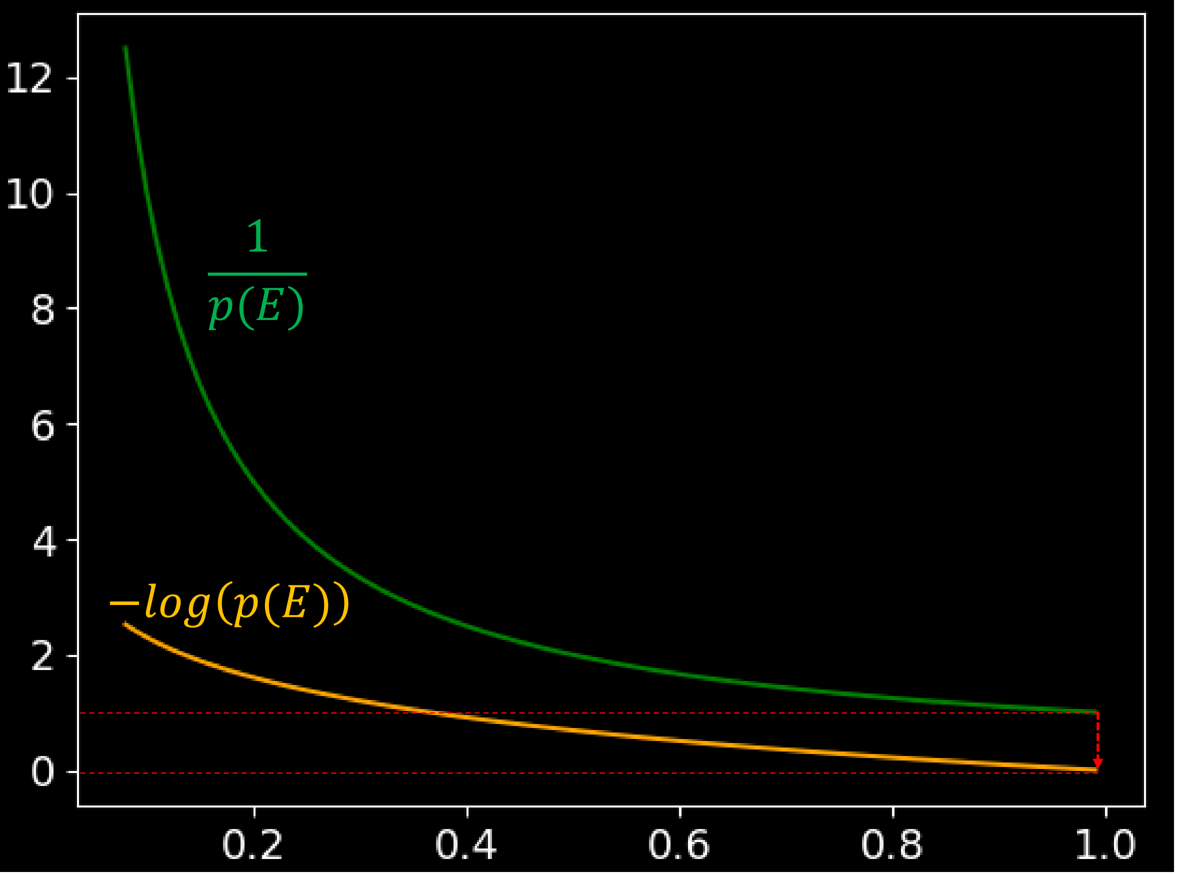
Observation

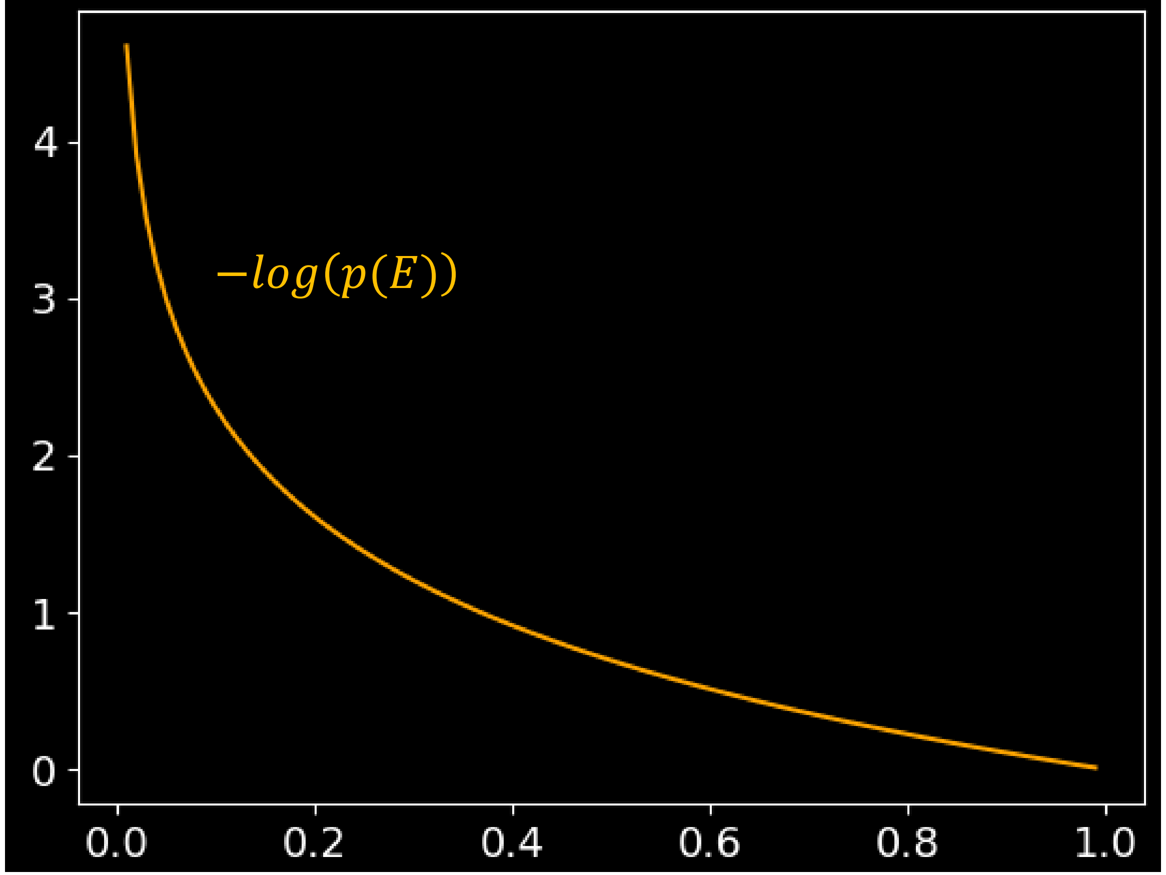
$$Surprise(E) \downarrow \uparrow p(E)$$

➡ $$Surprise(E) = \frac{1}{p(E)}$$

Problem?







Monotonic decrease of the function surprise(E)

$$log(Surprise(E)) = log\left(\frac{1}{p(E)}\right)$$

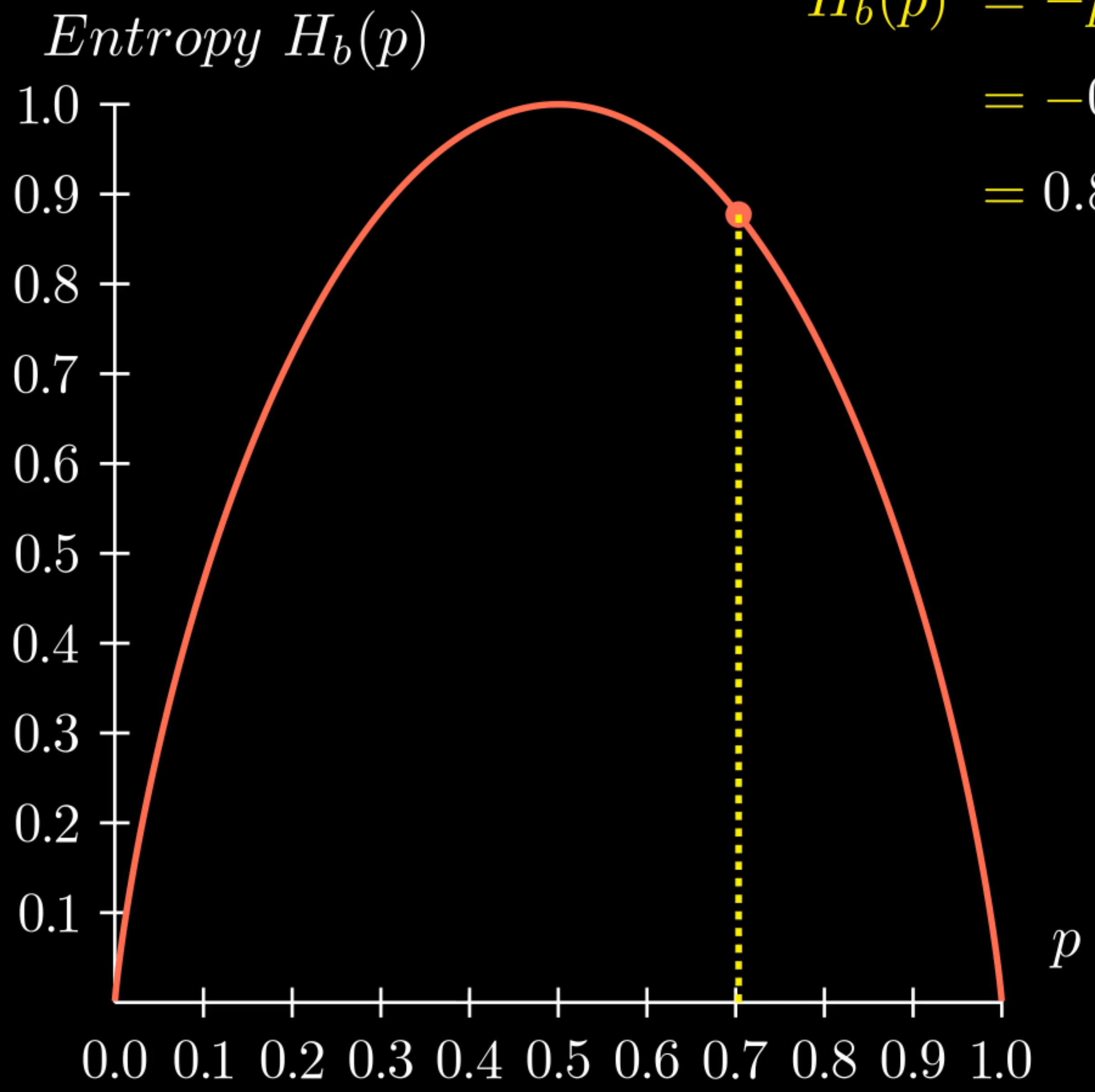
$$= -log(p(E))$$

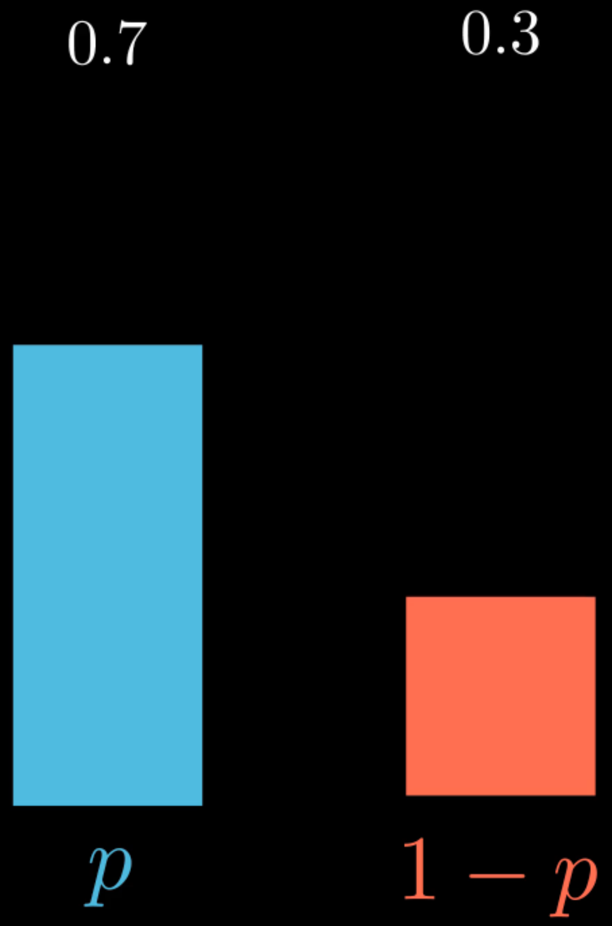
In information theory

$$\text{Information}(x) = -log(p(x))$$

AI VIETNAM

$$H_b(p) = -p\,log_2(p) - (1-p)\,log_2(1-p)$$
$$= -0.7\,log_2(0.7) - (1-0.7)\,log_2(1-0.7)$$
$$= 0.88$$

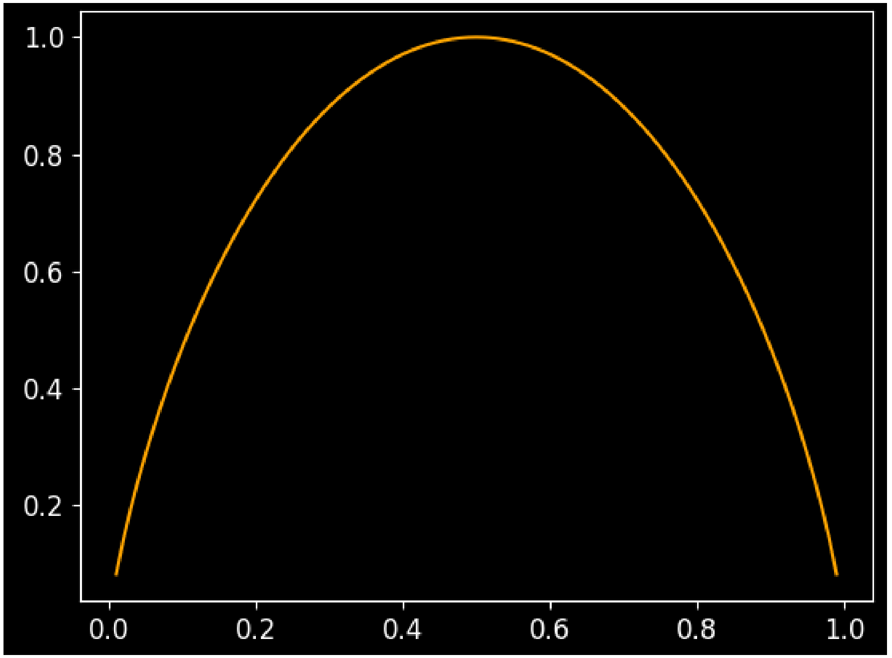


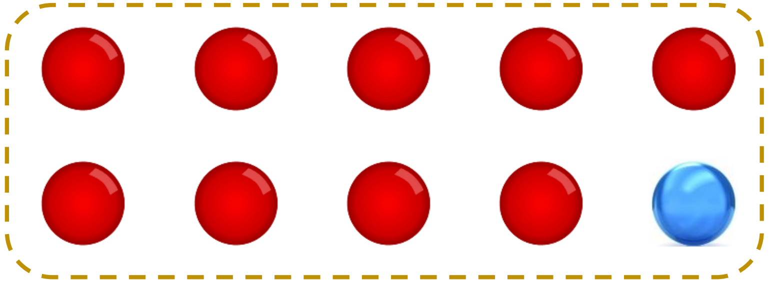


Entropy

Entropy: Average of information

$$H(X) := -\sum_{x \in X} p(x)\, log(p(x))$$





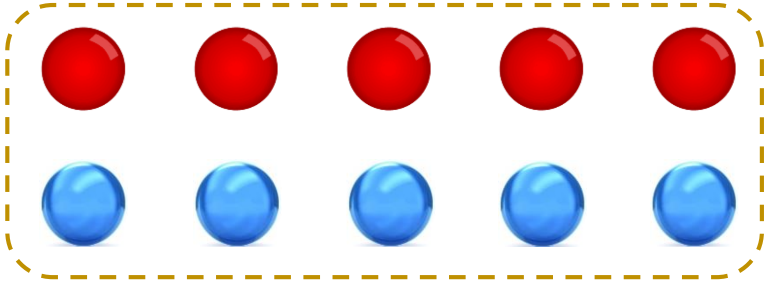
$$p(X = 0) = \frac{9}{10} = 0.9$$

$$p(X = 1) = \frac{1}{10} = 0.1$$

$$H(X) = -\sum_{x \in X} p(x)\, log(p(x))$$

$$= -0.9log(0.9) - 0.1log(0.1)$$

$$= 0.468$$



$$p(X = 0) = \frac{5}{10} = 0.5$$

$$p(X = 1) = \frac{5}{10} = 0.5$$

$$H(X) = -\sum_{x \in X} p(x)\, log(p(x))$$

$$= -0.5log(0.5) - 0.5log(0.5)$$

$$= 1.0$$