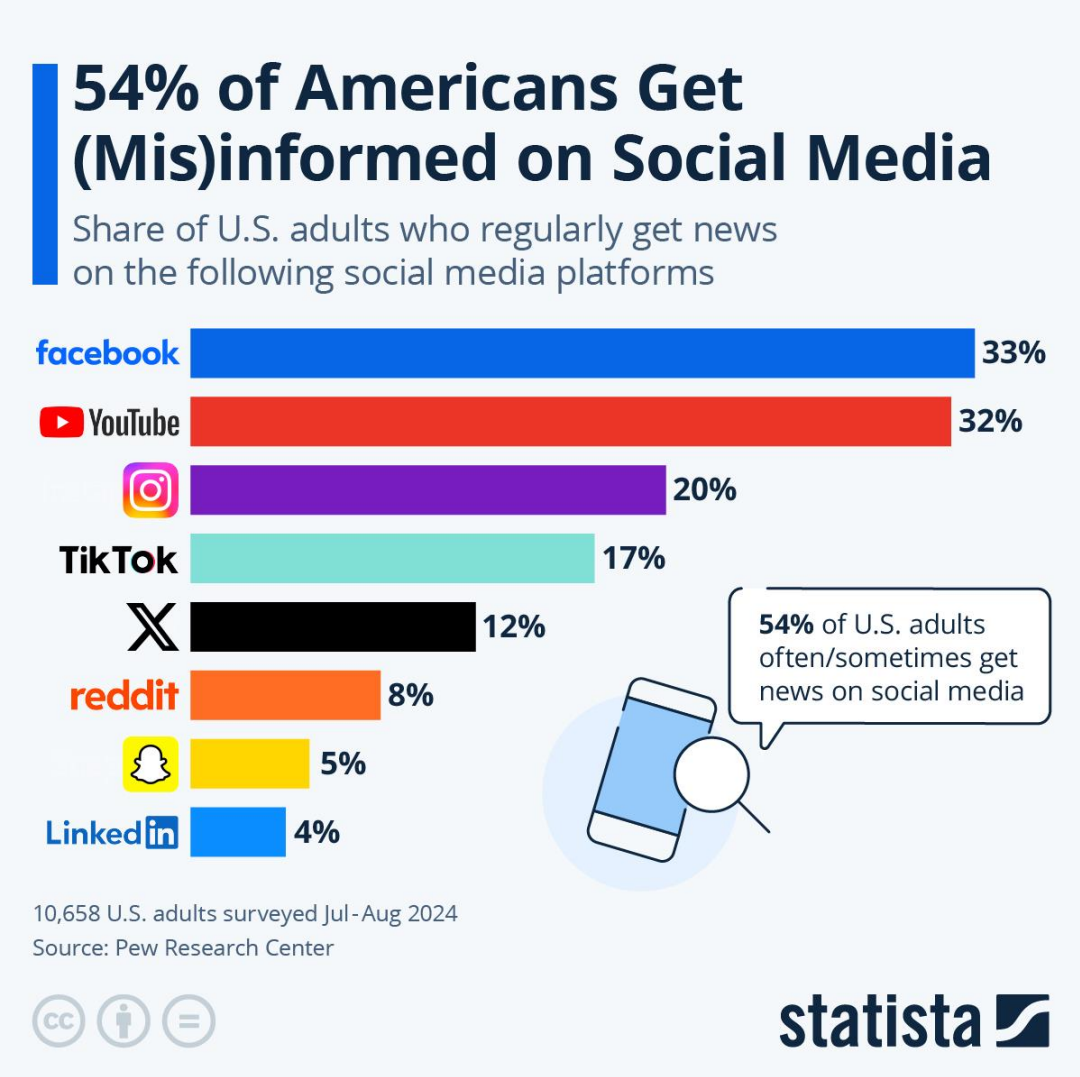
Data Mining

Naïve Bayes/KNN

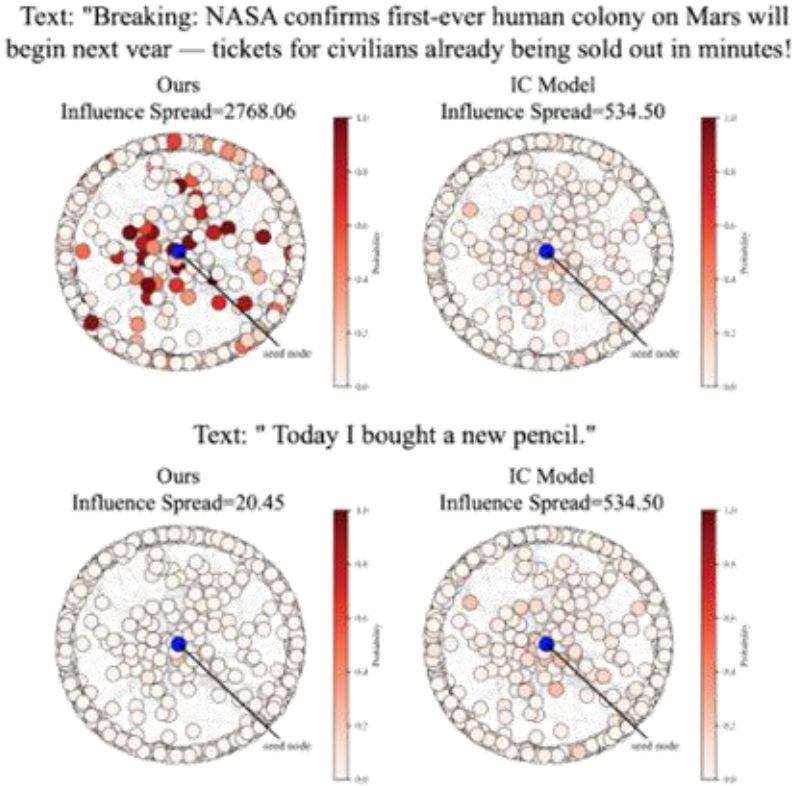
https://data-mining.github.io/winter-2026/

CS 453/553 – Winter 2026
Yu Wang, Ph.D.
Assistant Professor
Computer Science
University of Oregon

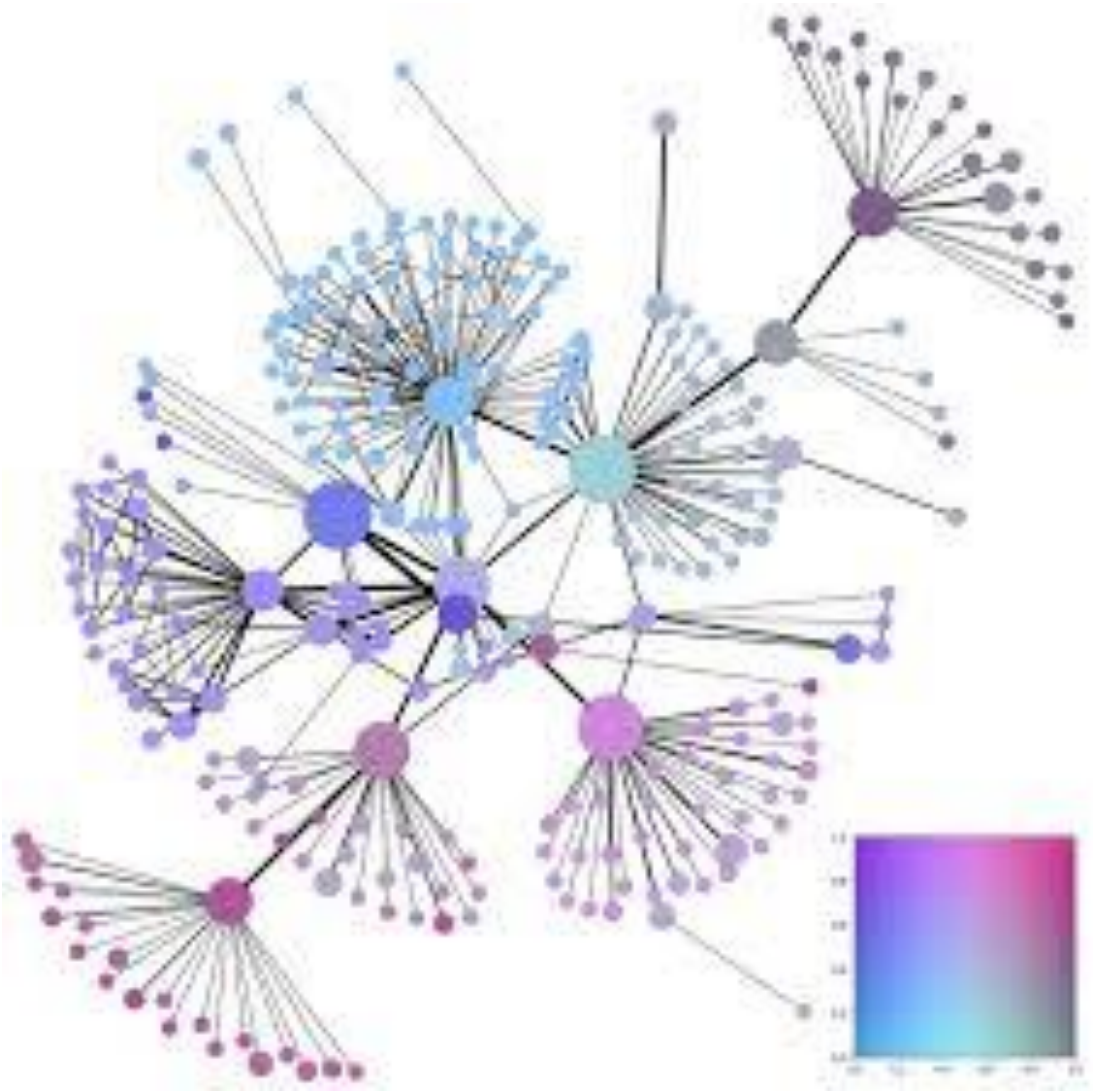
Motivation – What is Classification and Why?



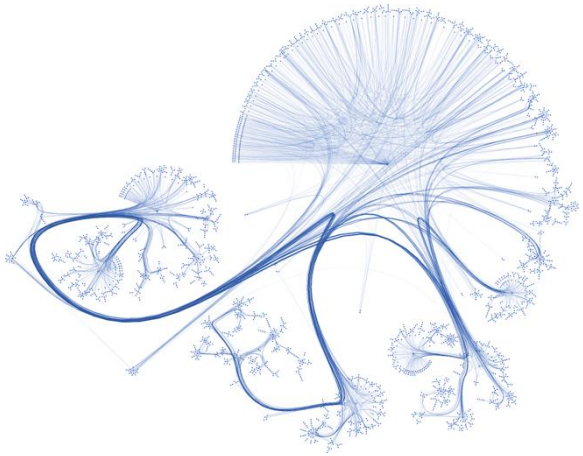
Misinformation Detection



Motivation – What is Classification and Why?



Paper Classification



Some Basic Methods

- **Naïve Bayes Classifier**
- **Nearest Neighbor Classifiers**
- **Decision-Tree**
- **Advanced Methods:**
  - **Neural Network**
  - **Geometric Neural Networks**

Naïve Bayes Classifier

$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$

Whether Evade?

Yes?
No?

$P(\text{Yes} \mid X)$

$P(\text{No} \mid X)$

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

P(No) = 7/10,
P(Yes) = 3/10

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$P(X|Yes)$ $P(X|No)$

$P(X_{re}, X_d, X_{in}|Yes)$ $P(X_{re}, X_d, X_{in}|No)$

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

$$P(X_{re}, X_d, X_{in} | Yes) = P(X_{re} | Yes) \\ P(X_d | Yes) P(X_{in} | Yes)$$

$$P(X_{re}, X_d, X_{in} | No) = P(X_{re} | No) \\ P(X_d | No) P(X_{in} | No)$$

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

P(X | Yes) =

P(Refund = No | Yes) x
P(Divorced | Yes) x
P(Income = 120K | Yes)

P(X | No) =

P(Refund = No | No) x
P(Divorced | No) x
P(Income = 120K | No)

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

For categorical attributes:

$P(X_i = c \mid y) = n_c / n$

- where $|X_i = c|$ is number of instances having attribute value $X_i = c$ and belonging to class y
- Examples:

P(Status=Married|No) = 4/7
P(Refund=Yes|Yes)=0

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

P(X | Yes) =

P(Refund = No | Yes) x

P(Divorced | Yes) x

P(Income = 120K | Yes)

P(X | No) =

P(Refund = No | No) x

P(Divorced | No) x

P(Income = 120K | No)

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

$$P(\text{Yes} \mid X) = \frac{P(X \mid \text{Yes})P(\text{Yes})}{P(X)}$$

$$P(\text{No} \mid X) = \frac{P(X \mid \text{No})P(\text{No})}{P(X)}$$

$$P(X|Y = No) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\mu)^2}{2\sigma^2}}$$

If Class=No

- sample mean = 110
- sample variance = 2975

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120\text{K})$$

Naïve Bayes Classifier:

P(Refund = Yes | No) = 3/7
P(Refund = No | No) = 4/7
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/7
P(Marital Status = Divorced | No) = 1/7
P(Marital Status = Married | No) = 4/7
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0

For Taxable Income:
If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

- P(X | No) = P(Refund=No | No)
  × P(Divorced | No)
  × P(Income=120K | No)
  = 4/7 × 1/7 × 0.0072 = 0.0006

- P(X | Yes) = P(Refund=No | Yes)
  × P(Divorced | Yes)
  × P(Income=120K | Yes)
  = 1 × 1/3 × 1.2 × 10^{-9} = 4 × 10^{-10}

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Naïve Bayes Classifier

Even in absence of information about any attributes, we can use Apriori Probabilities of Class Variable:

Naïve Bayes Classifier:

P(Refund = Yes | No) = 3/7
P(Refund = No | No) = 4/7
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/7
P(Marital Status = Divorced | No) = 1/7
P(Marital Status = Married | No) = 4/7
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0

For Taxable Income:
If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

P(Yes) = 3/10

P(No) = 7/10

If we only know that marital status is Divorced, then:

P(Yes | Divorced) = 1/3 x 3/10 / P(Divorced)

P(No | Divorced) = 1/7 x 7/10 / P(Divorced)

If we also know that Refund = No, then

P(Yes | Refund = No, Divorced) = 1 x 1/3 x 3/10 / P(Divorced, Refund = No)

P(No | Refund = No, Divorced) = 4/7 x 1/7 x 7/10 / P(Divorced, Refund = No)

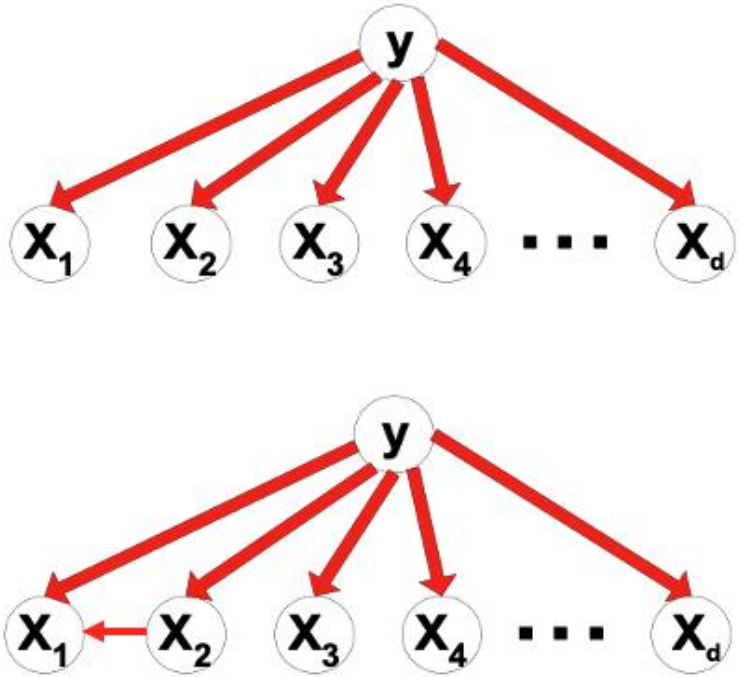
If we also know that Taxable Income = 120, then

P(Yes | Refund = No, Divorced, Income = 120) = 1.2×10^{-9} x 1 x 1/3 x 3/10 / P(Divorced, Refund = No, Income = 120)

P(No | Refund = No, Divorced Income = 120) = 0.0072 x 4/7 x 1/7 x 7/10 / P(Divorced, Refund = No, Income = 120)

Naïve Bayes Classifier

- Assume independence among attributes X_i when class is given:
  - $P(X_1, X_2, \ldots, X_d \mid Y_j) = P(X_1 \mid Y_j)\, P(X_2 \mid Y_j) \ldots P(X_d \mid Y_j)$

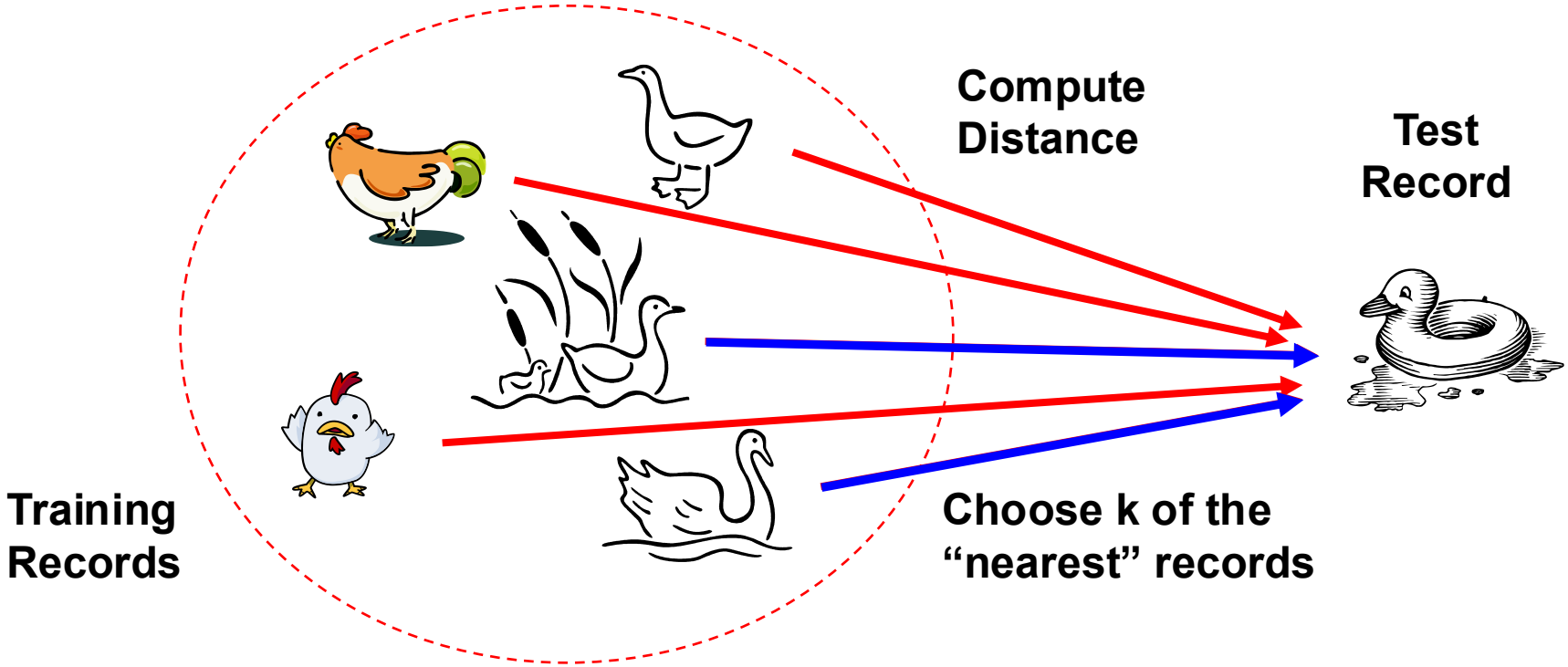


Some Basic Methods

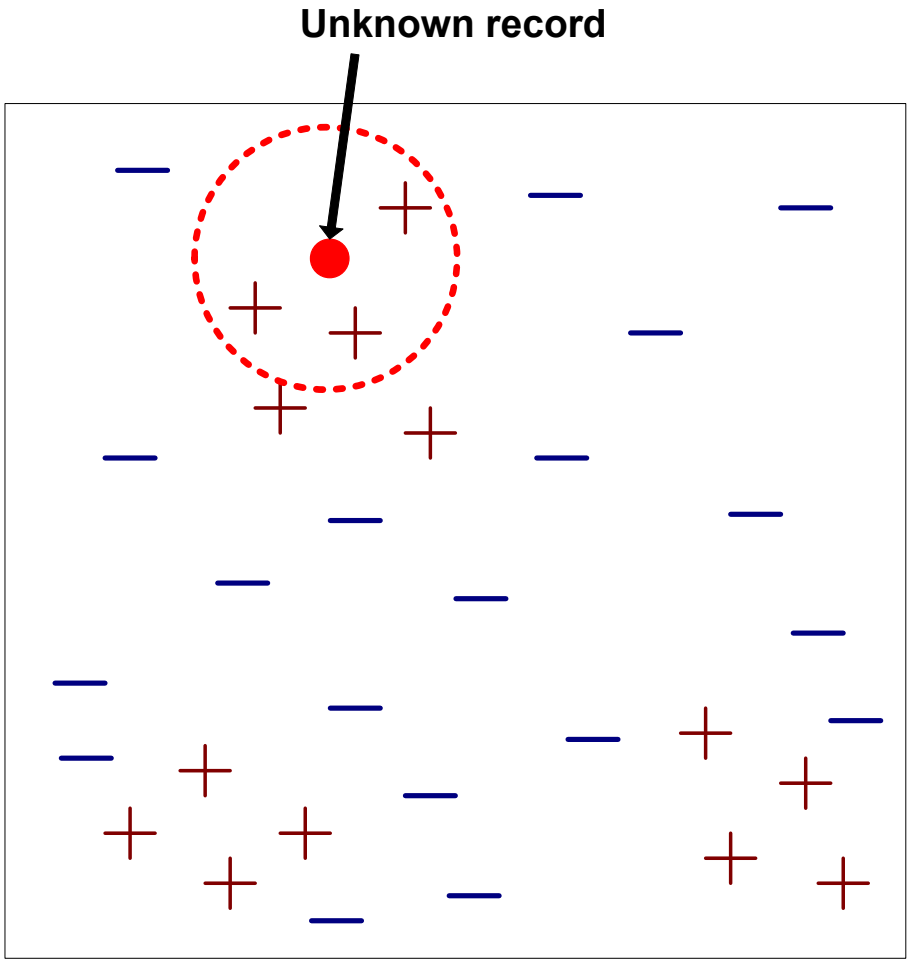
- Naïve Bayes Classifier
- Nearest Neighbor Classifiers
- Decision-Tree
- Advanced Methods:
  - Neural Network
  - Geometric Neural Networks

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck,
    quacks like a duck, then it's probably a duck



Nearest Neighbor Classifiers



What is training and testing data?

- Requires the following:
  - A set of labeled records
  - Proximity (Distance/Similarity) metric between a pair of records
    - e.g., Euclidean distance
  - The value of *k*, the number of nearest neighbors to retrieve
  - class labels of K nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

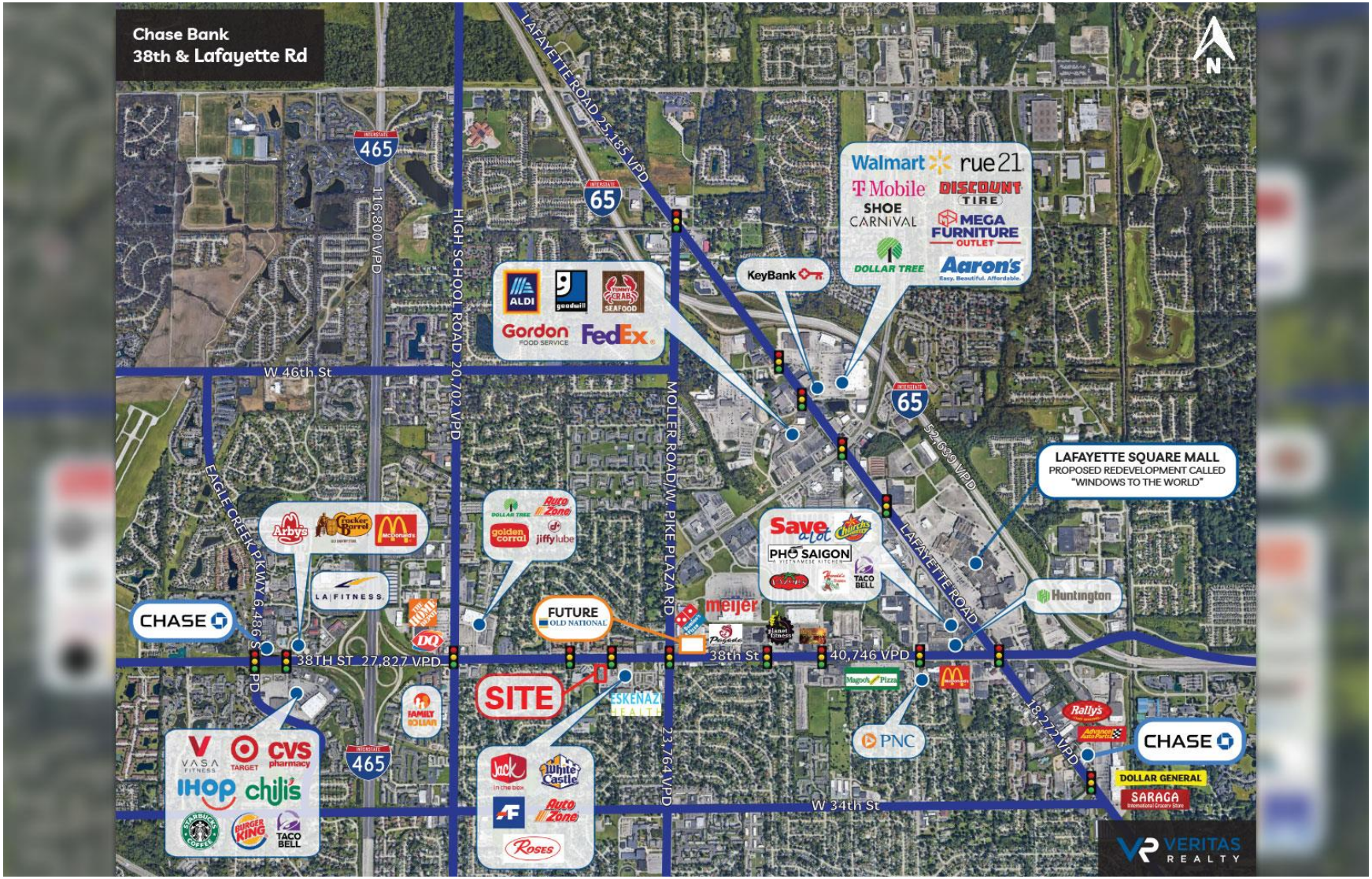
Or you can do $w = 1/d^2$

Nearest Neighbor Classifiers – Real Estate Example

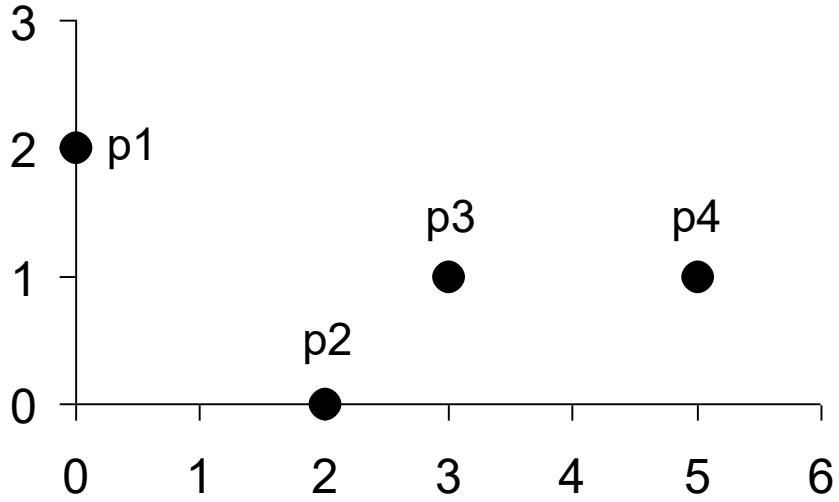
Commercial

Residential

Nearest Neighbor Classifiers – Real Estate Example



Nearest Neighbor Classifiers – Distance/Similarity Metric



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Nearest Neighbor Classifiers – Paper Classification

KW1 KW2 KW4
[0, 1, 1, 1]

[0, 1, 1, 0]

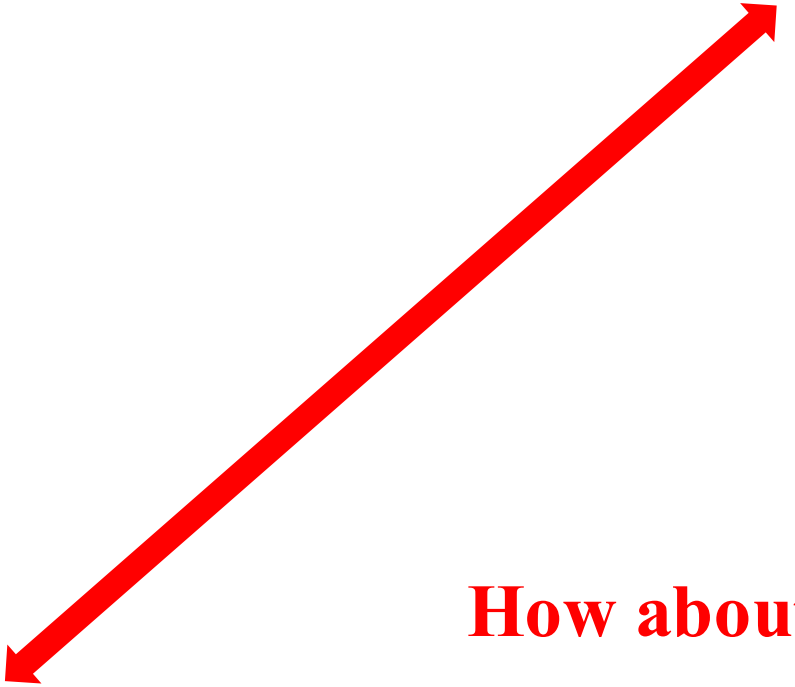
[1, 0, 0, 0]

Nearest Neighbor Classifiers – Paper Classification

KW1 KW2 KW4
[0, 1, 1, 1]

[0, 0, 0, 1]

[0, 1, 1, 0]



How about these two?

[1, 0, 0, 0]

Nearest Neighbor Classifiers – Distance Metric

You need to think about the distance metric!

Inner Product	Minkowski	Intersection	Entropy	χ^2 Family	Fidelity (Squared-Chord)	String Rearrangement	String Similarity
Subsec. 2.1	Subsec. 2.2	Subsec. 2.3	Subsec. 2.4	Subsec. 2.5	Subsec. 2.6	Subsec. 2.7	Subsec. 2.7
Inner Product Cosine Angular Jaccard Dice	Euclidean $(\text{Euclidean})^2$ L_1 L_p L_∞	Intersection Wave Hedges Sørensen Kulczynski Jaccard	Kullback-Leibler J-Divergence K-Divergence Topsøe Jensen-Shannon Jensen Diff. SED	Pearson Neyman Add. Sym.χ^2 Spearman Squared χ^2 Divergence Clark Mahalanobis	Fidelity Bhattacharyya Hellinger Matusia Squared-Chord	Hamming Levenshtein Swap Interchange Parallel-Interchange	LCS Jaro String N-Grams

Table 1. Similarity or distance measures appearing in this survey, categorized by measures families.

https://arxiv.org/pdf/2408.07706

Nearest Neighbor Classifiers – Distance Metric

Cosine Similarity

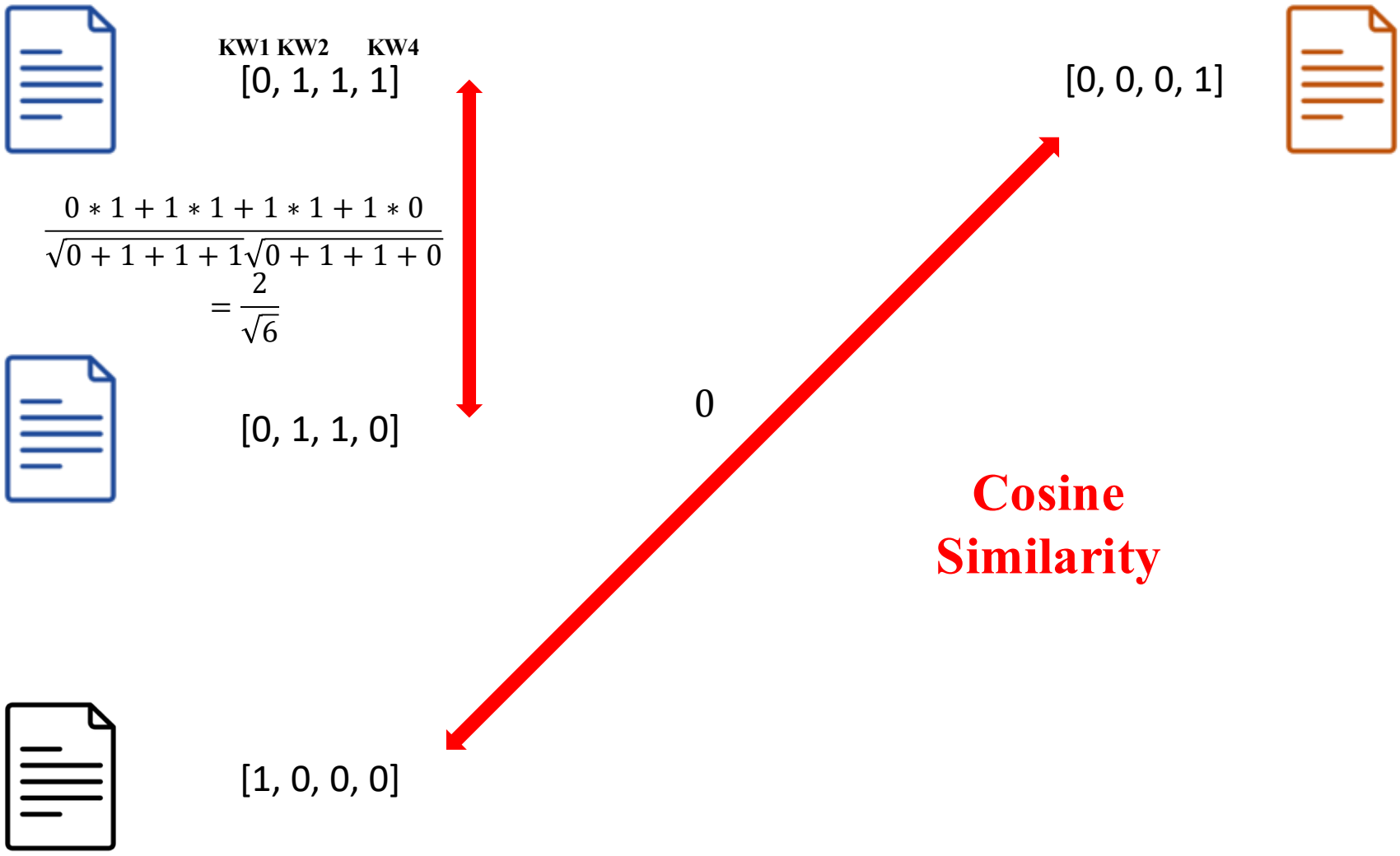
$$sim_{Cos}(P, Q) = \frac{\langle P, Q \rangle}{\|P\| \, \|Q\|}$$

Jaccard Similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

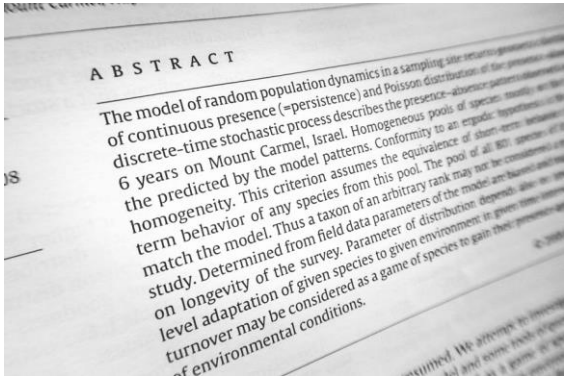
https://arxiv.org/pdf/2408.07706

Nearest Neighbor Classifiers – Paper Classification

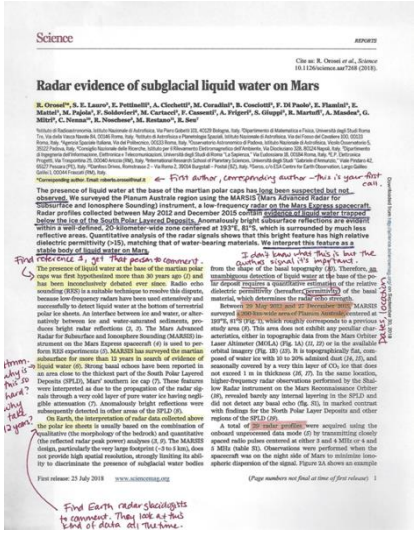


Nearest Neighbor Classifiers – Feature Preprocessing

- **Data Preprocessing is often required**
  - Different attributes have different scales
    - **height of a person may vary from 1.5m to 1.8m**
    - **weight of a person may vary from 90lb to 300lb**
    - **income of a person may vary from $10K to $1M**
  - Different instances may have different background


ABSTRACT

The model of random population dynamics in a sampling site returning continuous presence (=persistence) and Poisson distribution of discrete-time stochastic process describes the presence–absence pattern 6 years on Mount Carmel, Israel. Homogeneous pools of species, the predicted by the model patterns. Conformity to an ergodic hypothesis homogeneity. This criterion assumes the equivalence of short-term behavior of any species from this pool. The pool of all species match the model. Thus a taxon of an arbitrary rank may not be considered study. Determined from field data parameters of the model are based on longevity of the survey. Parameter of distribution depends, also on level adaptation of given species to given environment in given turnover may be considered as a game of species to gain their of environmental conditions.


Science

Radar evidence of subglacial liquid water on Mars

R. Orosei*, S. E. Lauro, E. Pettinelli, A. Cicchetti, M. Coradini, B. Cosciotti, F. Di Paolo, E. Flamini, E. Mattei, M. Pajola, F. Soldovieri, M. Cartacci, F. Cassenti, A. Frigeri, S. Giuppi, R. Martufi, A. Masdea, G. Mitri, C. Nenna, R. Noschese, M. Restano, R. Seu

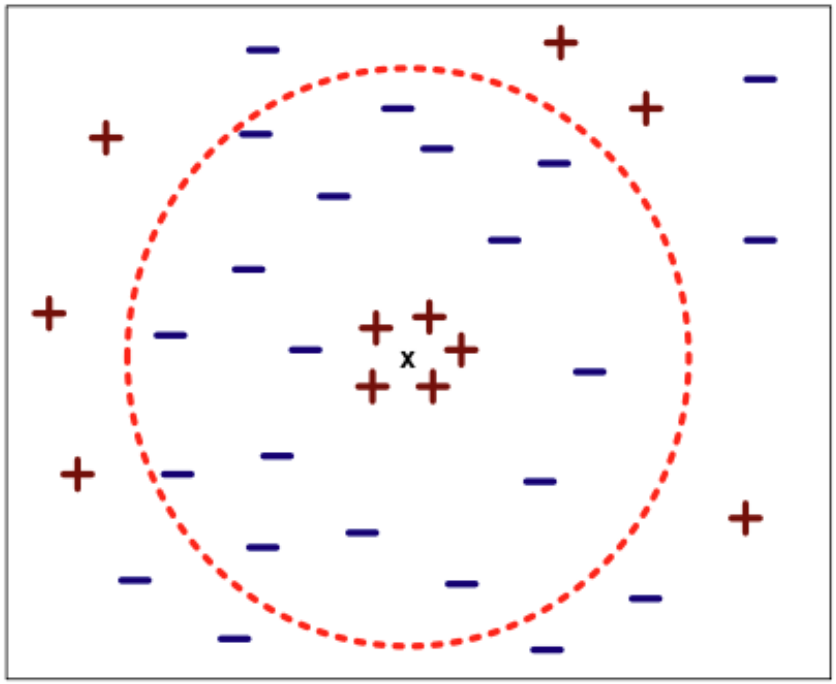
The presence of liquid water at the base of the martian polar caps has long been suspected but not observed. We surveyed the Planum Australe region using the MARSIS (Mars Advanced Radar for Subsurface and Ionosphere Sounding) instrument, a low-frequency radar on the Mars Express spacecraft. Radar profiles collected between May 2012 and December 2015 contain evidence of liquid water trapped below the ice of the South Polar Layered Deposits. Anomalously bright subsurface reflections are evident within a well-defined, 20-kilometer-wide zone centered at 193°E, 81°S, which is surrounded by much less reflective areas. Quantitative analysis of the radar signals shows that this bright feature has high relative dielectric permittivity (>15), matching that of water-bearing materials. We interpret this feature as a stable body of liquid water on Mars.

The presence of liquid water at the base of the martian polar caps was first hypothesized more than 30 years ago (1) and has been inconclusively debated ever since. Radio echo sounding (RES) is a suitable technique to resolve this dispute, because low-frequency radars have been used extensively and successfully to detect liquid water at the bottom of terrestrial polar ice sheets. An interface between ice and water, or alternatively between ice and water-saturated sediments, produces bright radar reflections (2, 3). The Mars Advanced Radar for Subsurface and Ionosphere Sounding (MARSIS) instrument on the Mars Express spacecraft (4) is used to perform RES experiments (5). MARSIS has surveyed the martian subsurface for more than 11 years in search of evidence of liquid water (6). Strong basal echoes have been reported in an area close to the thickest part of the South Polar Layered Deposits (SPLD), Mars' southern ice cap (7). Those features were interpreted as due to the propagation of the radar signals through a very cold layer of pure water ice having negligible attenuation (7). Anomalously bright reflections were subsequently detected in other areas of the SPLD (8).

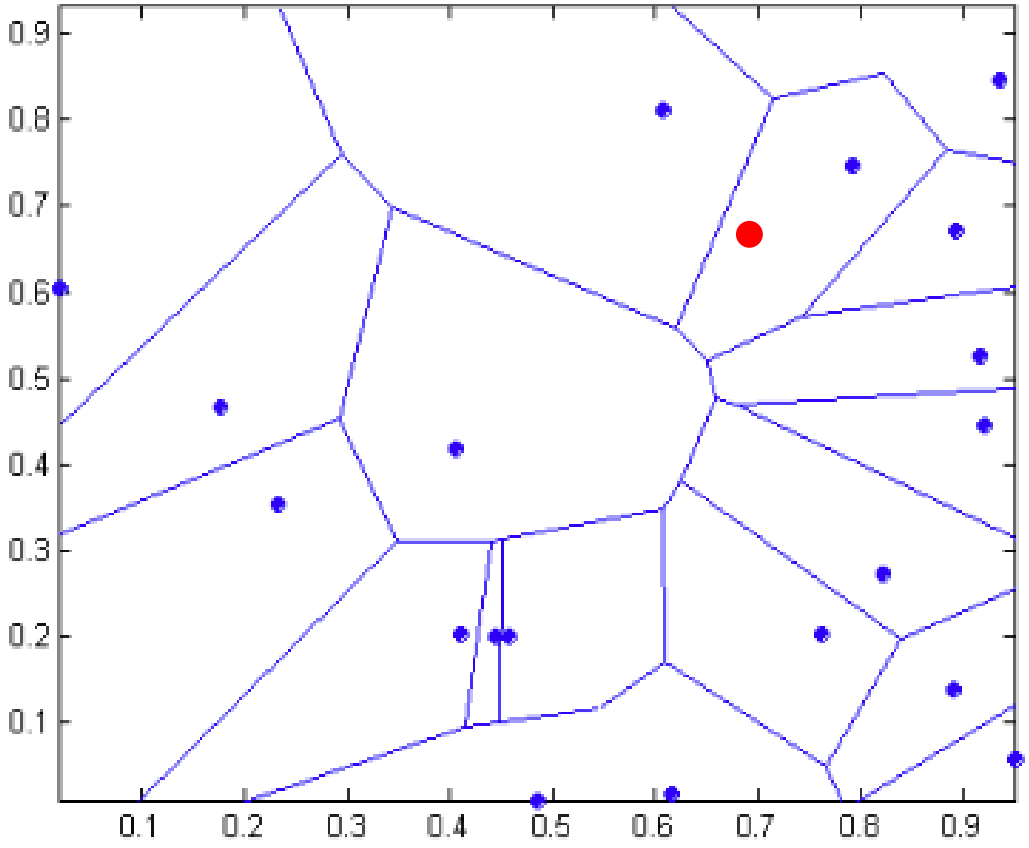
On Earth, the interpretation of radar data collected above the polar ice sheets is usually based on the combination of qualitative (the morphology of the bedrock) and quantitative (the reflected radar peak power) analyses (2, 3). The MARSIS design, particularly the very large footprint (~3 to 4 km), does not provide high spatial resolution, strongly limiting its ability to discriminate the presence of subglacial water bodies

Nearest Neighbor Classifiers – Hyperparameter

- **Choosing the value of k**
- **K too small?**
- **K too large?**



1-nn decision boundary is a Voronoi Diagram



Assuming N Training points

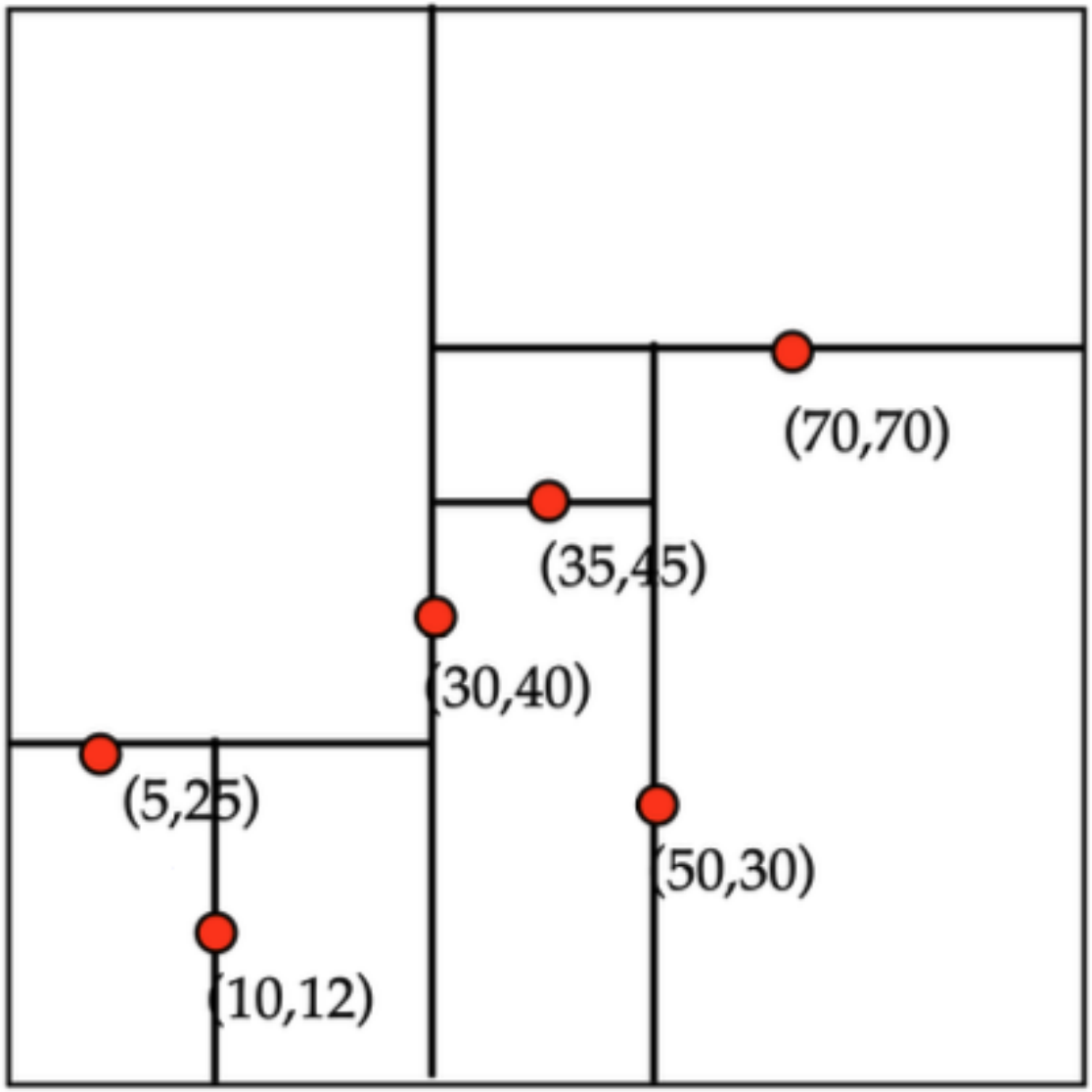
D Dimension

What is the time complexity of classifying that red point?

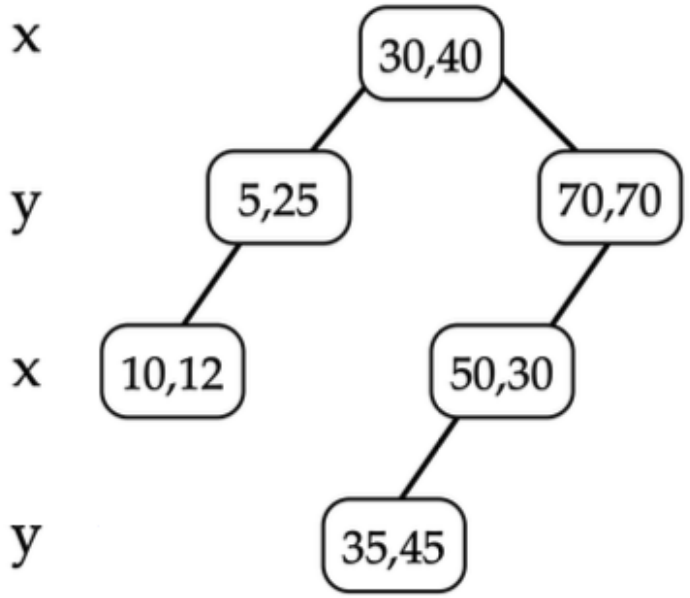
O(ND)

Everytime, we need to compare with every training points

Nearest Neighbor Classifiers – Efficiency

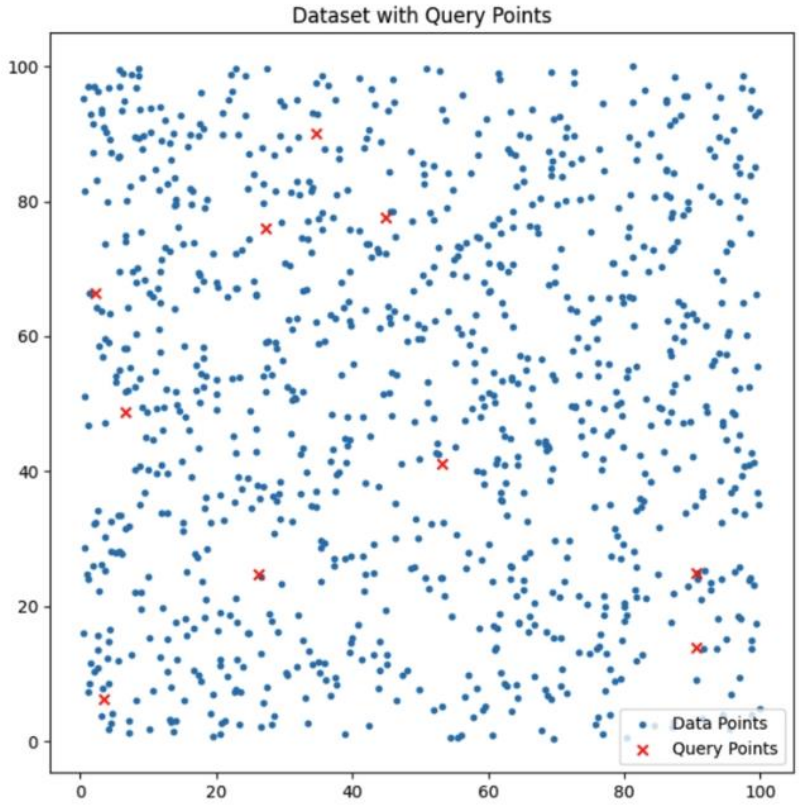


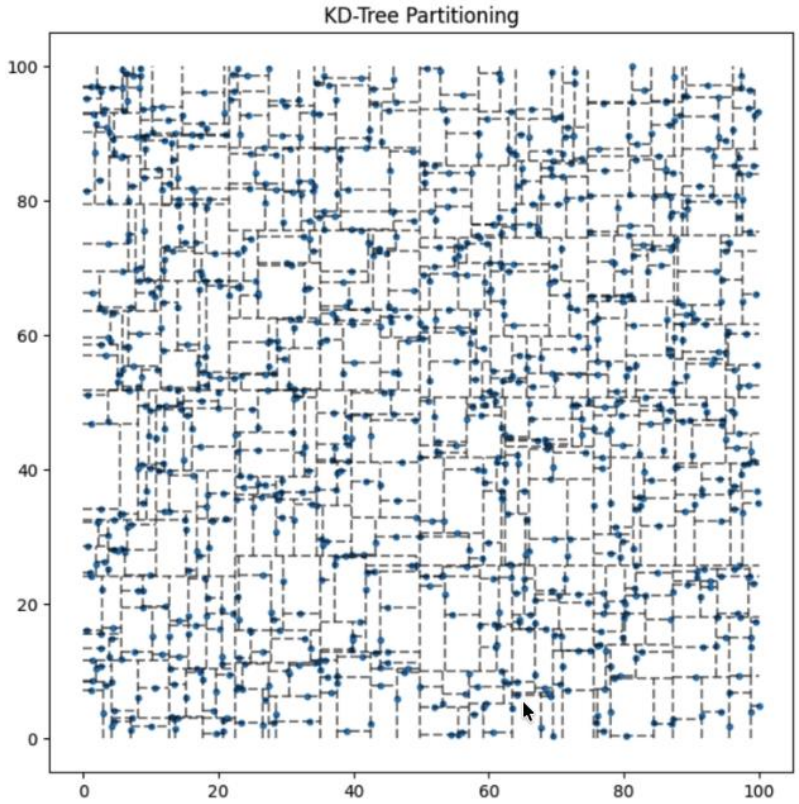
insert: (30,40), (5,25), (10,12), (70,70), (50,30), (35,45)



What is the time complexity of classifying that red point?

Nearest Neighbor Classifiers – Efficiency






```python
# Generate synthetic 2D dataset
np.random.seed(42)
n_samples = 1000  # Number of data points
data = np.random.rand(n_samples, 2) * 100  # 2D points in a 100x100 area

# Generate some random query points
n_queries = 10
queries = np.random.rand(n_queries, 2) * 100
```

```
Brute-force avg time: 0.175 ms
KD-Tree avg time: 0.059 ms
```

Nearest Neighbor Classifiers – Efficiency

Meta Our approach Research Product experiences Llama Blog

TOOLS

Faiss

Faiss (Facebook AI Similarity Search) is a library that allows developers to quickly search for embeddings of multimedia documents that are similar to each other. It solves limitations of traditional query search engines that are optimized for hash-based searches, and provides more scalable similarity search functions.

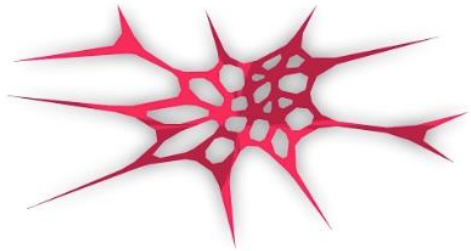
Efficient similarity search

With Faiss, developers can search multimedia documents in ways that are inefficient or impossible with standard database engines (SQL). It includes nearest-neighbor search implementations for million-to-billion-scale datasets that optimize the memory-speed-accuracy tradeoff. Faiss aims to offer state-of-the-art performance for all operating points.

Faiss contains algorithms that search in sets of vectors of any size, and also contains supporting code for evaluation and parameter tuning. Some if its most useful algorithms are implemented on the GPU. Faiss is implemented in C++, with an optional Python interface and GPU support via CUDA.

FAISS

Scalable Search With Facebook AI



```python
# Generate synthetic 2D dataset
np.random.seed(42)
n_samples = 100000
data = np.random.rand(n_samples, 20) * 100

n_queries = 10
queries = np.random.rand(n_queries, 20) * 100
```

```
Brute-force avg time: 8.112 ms
KD-Tree avg time:     4.666 ms
FAISS (FlatL2) avg:   0.253 ms
```