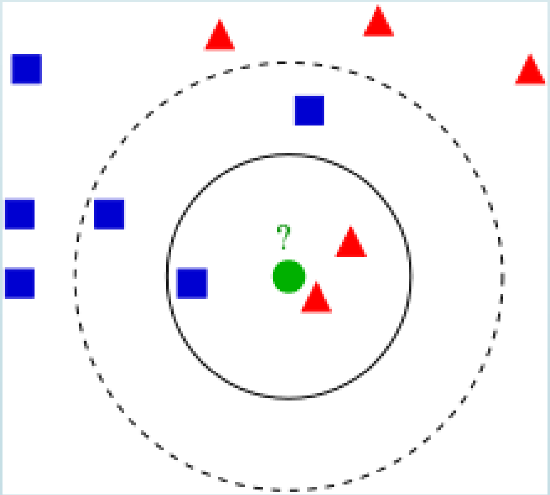
Supervised Learning

Nearest-Neighbor Classifiers

Sadegh Sulaimany
University of Kurdistan
www.bioinformation.ir

Spring 2024



Outline

- Motivation
- Idea
- Similarity measure
- Determination of ***k***?
- Examples
- Irrelevant features
- Scaling problem
- Normalization

Motivation

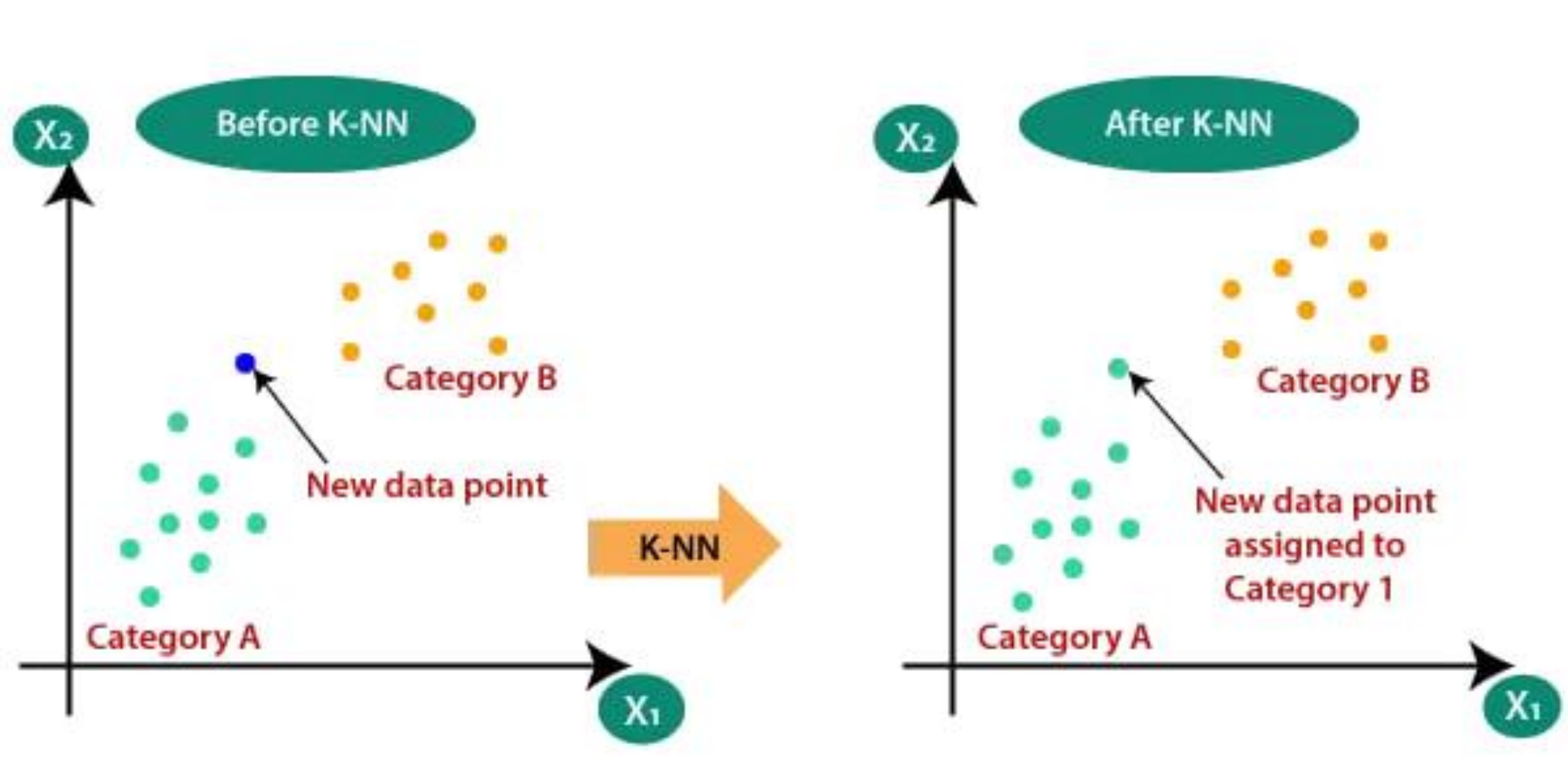

quotelia.com
You 're the average of the five people spend the most time with

Motivation

- Similar objects often belong to the same class
  - Two plants that look very much alike probably represent the same species
  - Patients complaining of similar symptoms suffer from the same disease

Main idea

- When asked to determine the class of object **x**,
  - find the training example most similar to it.
  - Then label **x** with this example's class.



Similarity of Feature Vectors

- How do we establish that an object is more similar to **x** than to **z**?
  - Counting the features in which they **differ**
    - **Less difference**
      - **More similarity**
        - ?

Table 3.1 Counting the numbers of differences between pairs of discrete-attribute vectors

Example	Shape	Crust		Filling		Class	# differences
		Size	Shade	Size	Shade		
x	Square	Thick	Gray	Thin	White	?	–
ex_1	Circle	Thick	Gray	Thick	Dark	pos	3
ex_2	Circle	Thick	White	Thick	Dark	pos	4
ex_3	Triangle	Thick	Dark	Thick	Gray	pos	4
ex_4	Circle	Thin	White	Thin	Dark	pos	4
ex_5	Square	Thick	Dark	Thin	White	pos	1
ex_6	Circle	Thick	White	Thin	Dark	pos	3
ex_7	Circle	Thick	Gray	Thick	White	neg	2
ex_8	Square	Thick	White	Thick	Gray	neg	3
ex_9	Triangle	Thin	Gray	Thin	Dark	neg	3
ex_{10}	Circle	Thick	Dark	Thick	White	neg	3
ex_{11}	Square	Thick	White	Thick	Dark	neg	3
ex_{12}	Triangle	Thick	White	Thick	Gray	neg	4

k-Nearest-Neighbor Rule

The simplest version of the k-NN classifier

Suppose we have a mechanism to evaluate the similarly between attribute vectors. Let $\mathbf{x}$ denote the object whose class we want to determine.

1. Among the training examples, identify the k nearest neighbors of $\mathbf{x}$ (examples most similar to $\mathbf{x}$).
2. Let c_i be the class most frequently found among these k nearest neighbors.
3. Label $\mathbf{x}$ with c_i.

k-Nearest-Neighbor

- Dealing with continuous features
  - each example **a point in an n-dimensional space**
  - calculate the geometric distance between any pair of examples
    - Euclidean distance
  - the closer to each other the examples are in the instance space, the greater their mutual similarity
  - the training example with the smallest distance from **x** in the instance space is x’s nearest neighbor.

From a Single Neighbor to k Neighbors

- In noisy domains,
  - the testimony of the nearest neighbor cannot be trusted.
- A more robust approach identifies several nearest neighbors
  - k-NN classifier, where k is the number of the voting neighbors
    - usually a user-specified parameter

K?

- For Binary classifier,
  - *k* should be an odd number

    Why?

  - Example
    - a 4-NN classifier might face a situation where the number of positive neighbors is the same as the number of negative neighbors.
    - This will not happen to a 5-NN classifier

K?

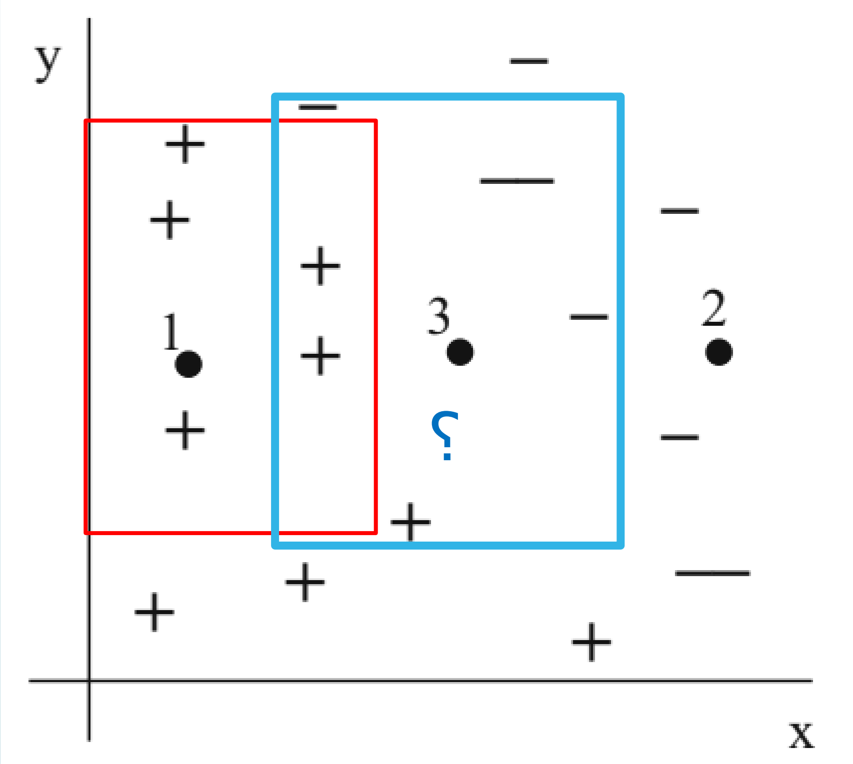
- For a multi-class model,
  - Does using an odd number of nearest neighbors prevent ties?

    Why?

  - Example,
    - the 7-NN classifier can realize that three neighbors belong to class C1, three neighbors belong to class C2, and one neighbor belongs to class C3.
    - The engineer designing the classifier needs to define a mechanism to choose between C1 and C2.
      - **What generative AI may recommend?**

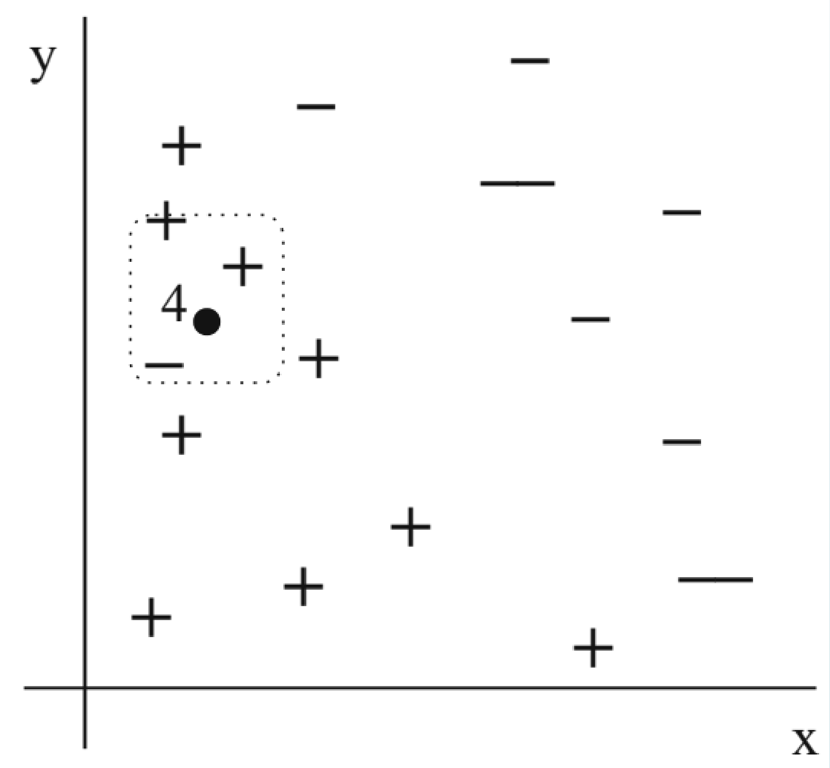
Example

- Border line examples are unreliable
  - Sensitive to noise



Example

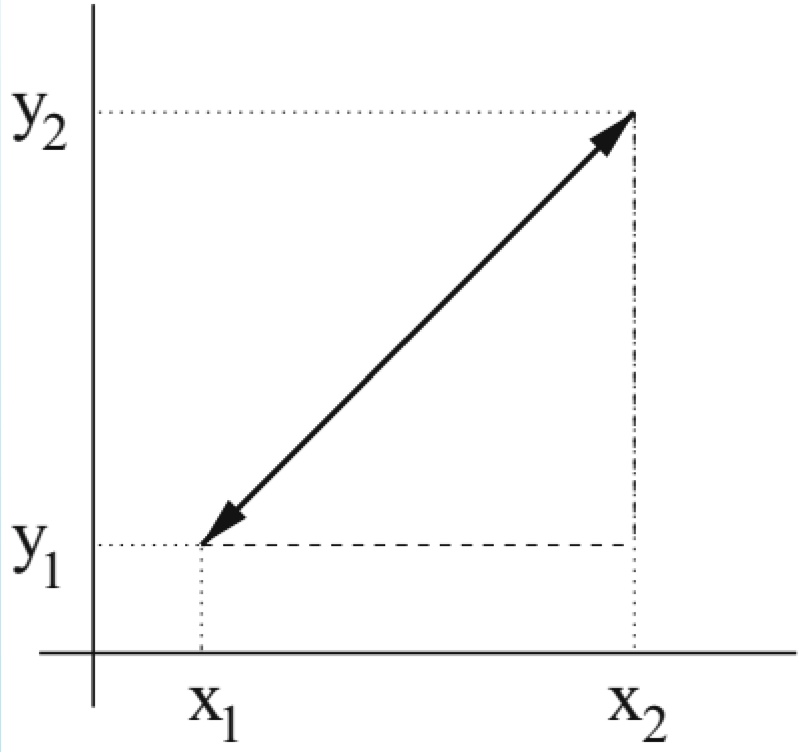
- 1-NN classifier will be affected by mislabeled noisy neighbor
- 3-NN classifier will give the correct answer



Measuring similarity in k-NN

- a natural way to find the nearest neighbors of object **x**
  - is to compare the geometrical distances
    - **Example of two-dimentional space**

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



- **For n continuous features**

$$\mathbf{x} = (x_1, \ldots, x_n) \text{ and } \mathbf{y} = (y_1, \ldots, y_n)$$

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\Sigma_{i=1}^{n} (x_i - y_i)^2}$$

Examples

- Using the nearest-neighbor principle in a 3-dimensional Euclidean space
  - 1-NN classifier ?
  - 3-NN classifier ?

Using the following training set of four examples described by three numeric attributes, determine the class of object $\mathbf{x} = [2, 4, 2]$.

Distance between ex_i and $[2, 4, 2]$	
ex_1	$\{[1, 3, 1], \text{pos}\}$
ex_2	$\{[3, 5, 2], \text{pos}\}$
ex_3	$\{[3, 2, 2], \text{neg}\}$
ex_4	$\{[5, 2, 3], \text{neg}\}$

General simulation for similarity

- A mixed formula

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{\Sigma_{i=1}^{n} d(x_i, y_i)}$$

  - For continuous features

$$d(x_i, y_i) = (x_i - y_i)^2$$

  - For discrete features

$$d(x_i, y_i) = 0 \text{ if } x_i = y_i$$
$$d(x_i, y_i) = 1 \text{ if } x_i \neq y_i$$

General simulation for similarity

- A mixed formula

$$d_M(\mathbf{x},\mathbf{y}) = \sqrt{\Sigma_{i=1}^{n} d(x_i, y_i)}$$

  - If all features are continuous
    - formula is identical to Euclidean distance
  - If all features are discrete
    - formula simply count the differences
    - In purely Boolean domains

      Just calculate the hamming distance

$$\mathbf{x} = (t, t, f, f)$$
$$\mathbf{y} = (t, f, t, f)$$

$$d_H(\mathbf{x},\mathbf{y}) = 2$$

Misleading distances

- feature-to-feature Distances Can Be Misleading
  - be careful not to apply Formula mechanically ignoring the specific aspects of the given domain
  - Example
    - Features:

      [size, price, season]

$$\mathbf{x} = (2, 1.5, \texttt{summer})$$
$$\mathbf{y} = (1, 0.5, \texttt{winter})$$

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(2-1)^2 + (1.5-0.5)^2 + 1} = \sqrt{3}$$

- d(summer, winter) ?= d(fall, winter)

Misleading distances

- feature-to-feature Distances Can Be Misleading
  - be careful not to apply Formula mechanically
  - Example 1
    - Features:

      [size, price, season]
    - Mixing continuous and discrete features can be risky
    - Size 1 = 1, Size 2 = 12
      - can totally dominate the difference between two seasons

Distances in General

- few other formulas have been suggested
  - polar distance, the Minkowski metric, and the Mahalanobis distance

- any distance metric has to satisfy the following requirements:

1. the distance must never be negative;
2. the distance between two identical vectors, $\mathbf{x}$ and $\mathbf{y}$, is zero;
3. the distance from $\mathbf{x}$ to $\mathbf{y}$ is the same as the distance from $\mathbf{y}$ to $\mathbf{x}$;
4. the metric must satisfy the triangular inequality: $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$.

Irrelevant features & Scaling Problems

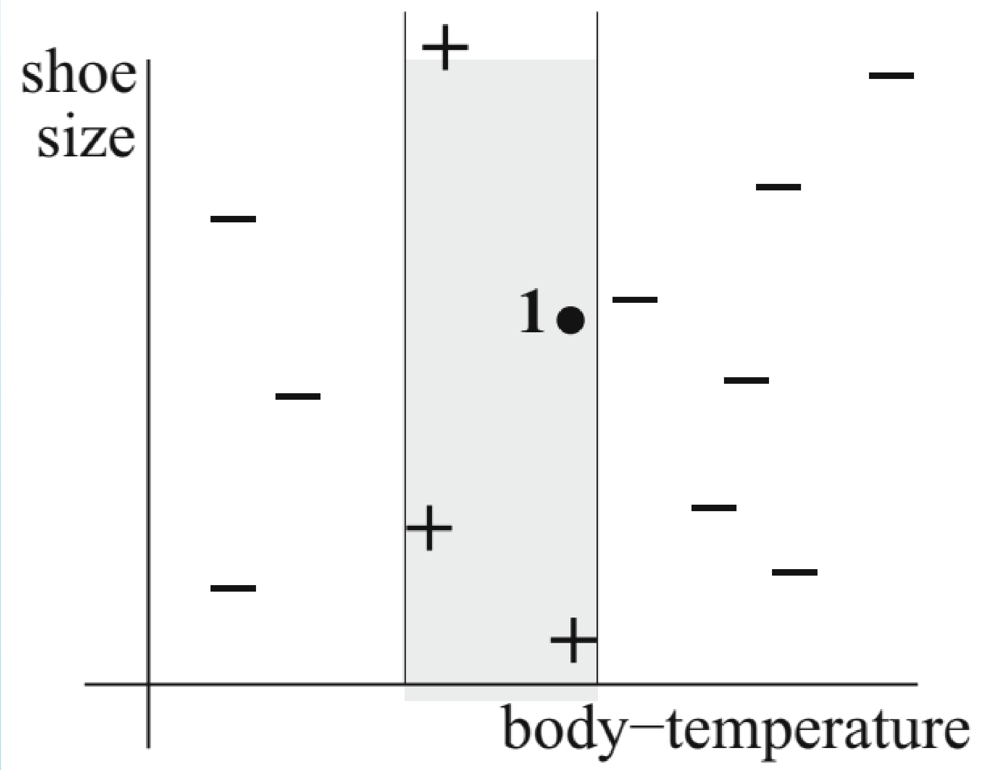
- By now, the reader understands the principles of the k-NN classifier well enough
  - to be able to write a computer program implementing the tool.

    This is not enough

- rock-bottom of the nearest-neighbor paradigm
  - "objects are similar if the geometric distance between the vectors describing them is small."
  - In certain situations, the geometric distance can be misleading.

Irrelevant features

- Not all features are equal !
  - some are irrelevant
  - their values have nothing to do with the given example's class
  - But they do affect the geometric distance between vectors
  - **Example**
    - **The black dot stands for the object that the k-NN classifier is expected to label as healthy (pos) or sick (neg)**



Irrelevant features

- How much damage is caused by irrelevant features?
  - depends on how many of them are used to describe the examples.
- In a domain with hundreds of features, of which only one is irrelevant
  - there is no need to panic
- If the vast majority of the features have nothing to do with the class we want to recognize,
  - then the geometric distance will become almost meaningless

Scales of feature Values

- When a feature dominates the distance
- Example

$$\mathbf{x} = (t, 0.2, 254)$$
$$\mathbf{y} = (f, 0.1, 194)$$

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(1-0)^2 + (0.2-0.1)^2 + (254-194)^2}$$

  - First feature is Boolean
  - Second feature is continuous with values from interval [0,1]
  - Third is continuous with values from interval [0,1000]

- No matter the first and second feature are when having great change in the third feature value!

Normalization

Another example of feature scaling

- Second feature is Temperature
  - 1-NN Classifier

in centigrade

$$\mathrm{ex}_1 = [(10, 10), \mathrm{pos})]$$
$$\mathrm{ex}_2 = [(20, 0), \mathrm{neg})]$$
$$\mathbf{x} = (32, 20)$$

$$d_M(\mathbf{x}, \mathrm{ex}_1) = \sqrt{584}$$
$$d_M(\mathbf{x}, \mathrm{ex}_2) = \sqrt{544}$$

label $\mathbf{x}$ as neg

in farenheit

$$\mathrm{ex}_1 = [(10, 50), \mathrm{pos})]$$
$$\mathrm{ex}_2 = [(20, 32), \mathrm{neg})]$$
$$\mathbf{x} = (32, 68)$$

$$d_M(\mathbf{x}, \mathrm{ex}_1) = \sqrt{808}$$
$$d_M(\mathbf{x}, \mathrm{ex}_2) = \sqrt{1440}$$

classify $\mathbf{x}$ as positive

Normalizing the features

- makes all values fall into the same interval
  - [0,1]
- Simplest formula

$$x = \frac{x - MIN}{MAX - MIN}$$

- Example

$$[7, 4, 25, -5, 10]$$

$$[12, 9, 30, 0, 15]$$

$$[0.4, 0.3, 1, 0, 0.5]$$

Potential Weakness of Normalization

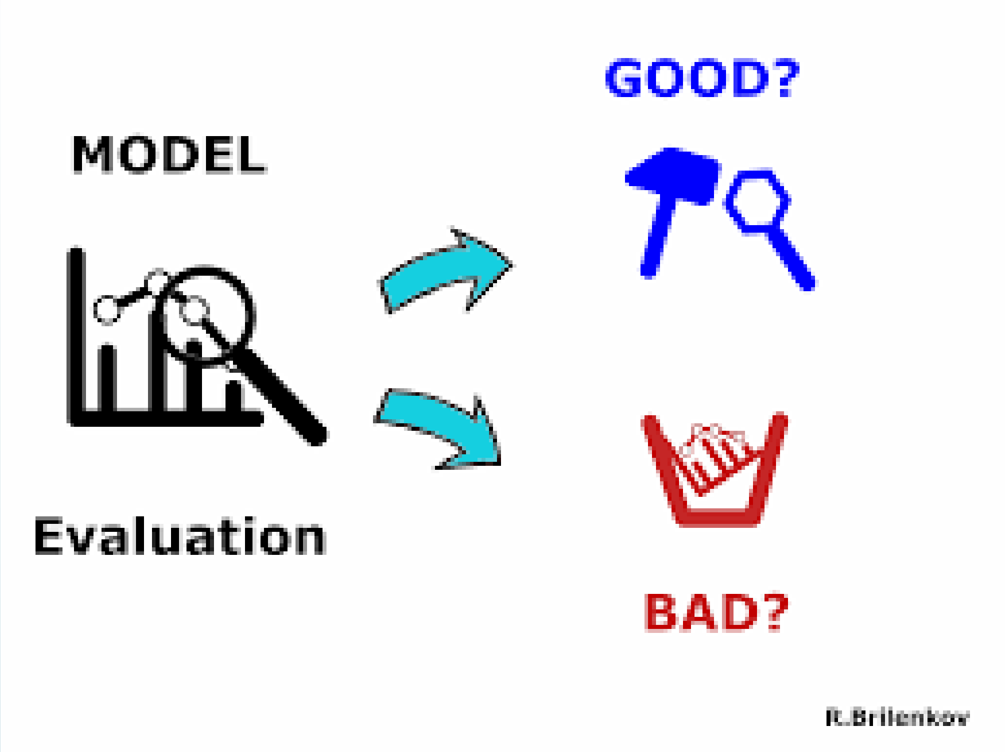
- Loss of originality
  - Sometimes make the results hard to interpret
- Sensitivity to outlier
  - single outlier can skew the normalized values
- Gives equal importance to all features
  - Example:

  difference between summer and fall is 1,

  Seems bigger than the difference between two normalized body temperatures.

Handling these is up to the engineer's common sense—
assisted by his or her experience and perhaps a little experimentation

Evaluating the classifier



Evaluating the classifier

- Confusion matrix
  - A table that is used to define the performance of a classification algorithm

		Actual Class	
		Positive (**P**)	Negative (**N**)
Predicted Class	Positive (**P**)	True Positive (**TP**)	False Positive (**FP**)
	Negative (**N**)	False Negative (**FN**)	True Negative (**TN**)

Evaluating the classifier

- Confusion matrix Example
  - Imagine we have a machine learning model

    that predicts whether an email is spam (positive class) or not spam (negative class).

    After testing it with 100 emails, we get the following results:

    - 40 (emails correctly identified as spam)
      - **TP**
    - 10 (emails incorrectly identified as spam)
      - **FP**
    - 45 (emails correctly identified as not spam)
      - **TN**
    - 5 (emails incorrectly identified as not spam)
      - **FN**

Evaluating the classifier

- Precision (Positive Predictive Value):
  - The ratio of correctly predicted positive observations to the total predicted positives.
    - ***Precision = TP / (TP + FP)***
    - ?
- Recall (Sensitivity or True Positive Rate):
  - The ratio of correctly predicted positive observations to all actual positives.
    - ***Recall = TP / (TP + FN)***
    - ?

Precision-Recall relation?

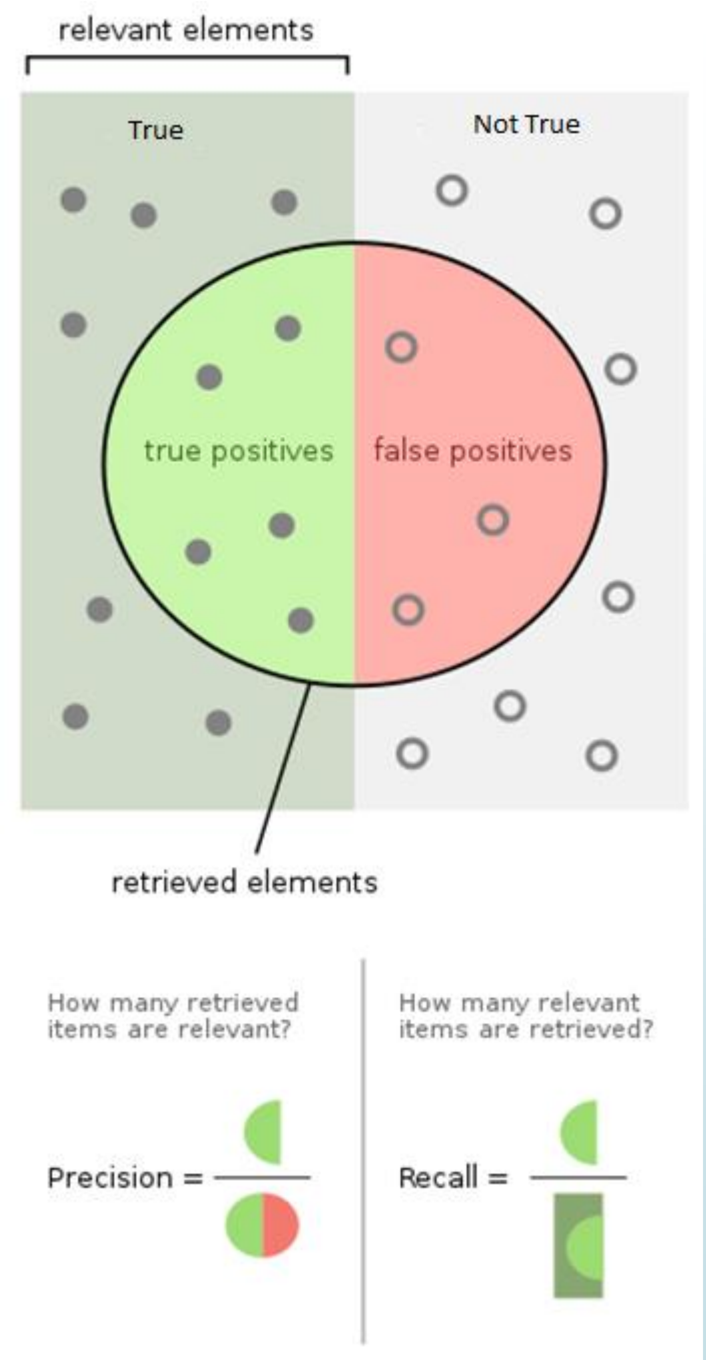
- Precision
  - Of all the instances the model labeled as positive, how many are actually positive?
  - Local view
- Recall
  - Of all the actual positive instances, how many did the model correctly identify?
  - Global view
- *What is the relation between Precision and Recall?*
  - They are inversely related!
  - **precision-recall trade-off**

Precision-Recall Trade-off

- F1 score
  - the harmonic mean of precision and recall. It gives a single score that balances both concerns

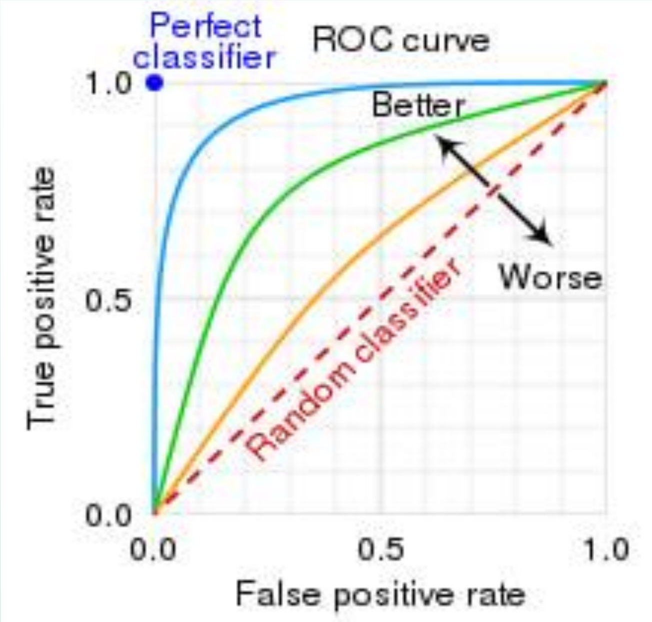
$$F_1 = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

- F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
- ?



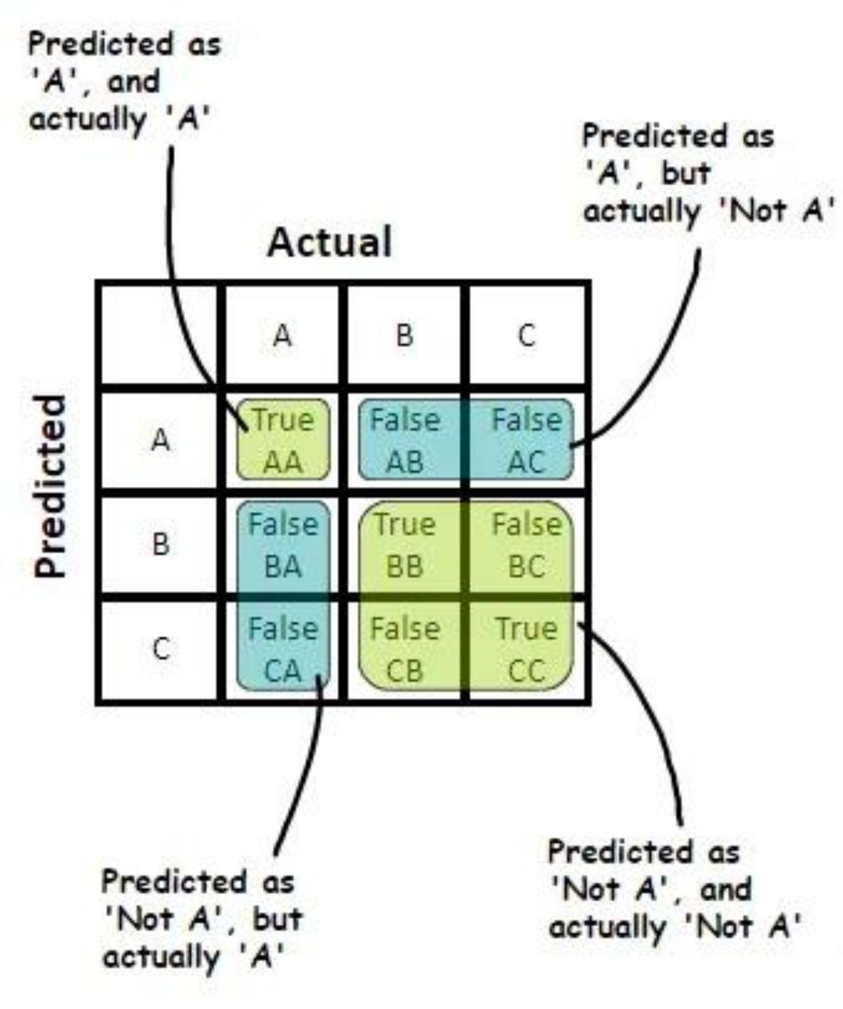
ROC Curve

- ROC Curve
  - (Receiver Operating Characteristic Curve)
  - A graph showing the performance of a classification model
  - This curve plots two parameters:
    - True Positive Rate (Recall)
    - False Positive Rate (FPR)



  - The **Area Under the Curve (AUC)** of the ROC curve
    - quantifies the model's overall performance, with a higher AUC indicating better predictive accuracy.

Future studies?



Thanks