CS 481

Artificial Intelligence Language Understanding

February 21, 2023

Announcements / Reminders

- Please follow the Week 06 To Do List instructions
- PA #01 due on ~~Monday (02/20/23) at 11:59 PM CST~~
  Thursday (02/23/23) at 11:59 PM CST
- Exam dates:
  - Midterm: 03/02/2023 during Thursday lecture time
  - Final: 04/27/2023 during Thursday lecture time

Plan for Today

- **Text classification**
- **Naïve Bayes classifier**

What is Classification?

Definition:

Classification is a process of **categorizing data into distinct classes**. In practice it means **developing a model that maps input data to a discrete set of labels / targets**. Classification can be:

- **binary** - there is only two classes: yes / no, true / false, spam / not spam
- **multi-class** - there are multiple classes available, only one is assigned
- **multi-label** - multiple classes an be assigned

Main Machine Learning Categories

Supervised learning	Unsupervised learning	Reinforcement learning
Supervised learning is one of the most common techniques in machine learning. It is based on **known relationship(s) and patterns within data** (for example: relationship between inputs and outputs). Frequently used types: **regression**, and **classification**.	**Unsupervised learning** involves finding underlying patterns within data. Typically used in **clustering** data points (similar customers, etc.)	Reinforcement learning is inspired by behavioral psychology. It is **based on a rewarding / punishing an algorithm**. Rewards and punishments are based on algorithm's action within its environment.

Supervised Learning

Given a training set of N example input-output (feature-label) pairs

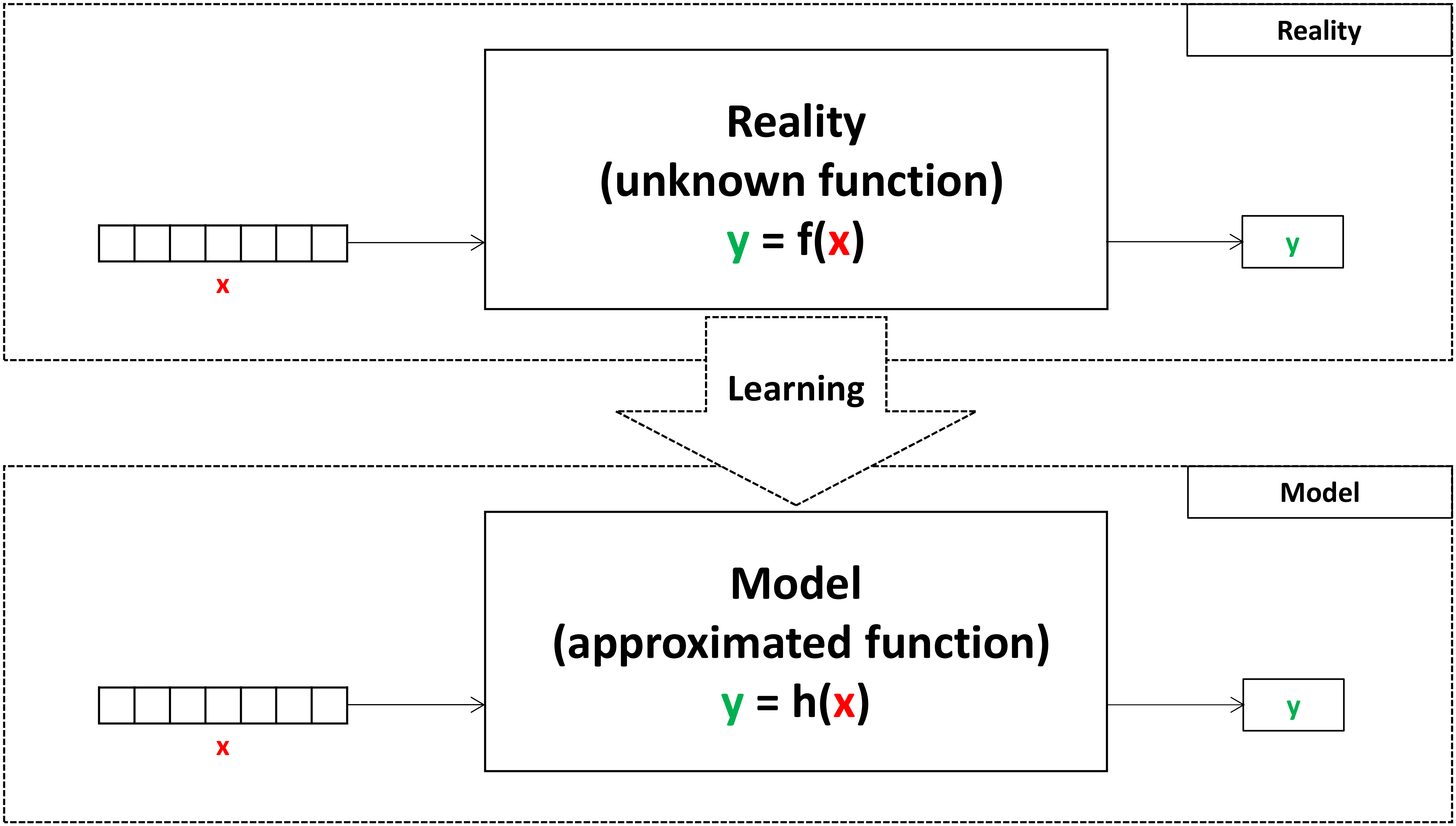
$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by some UNKNOWN function

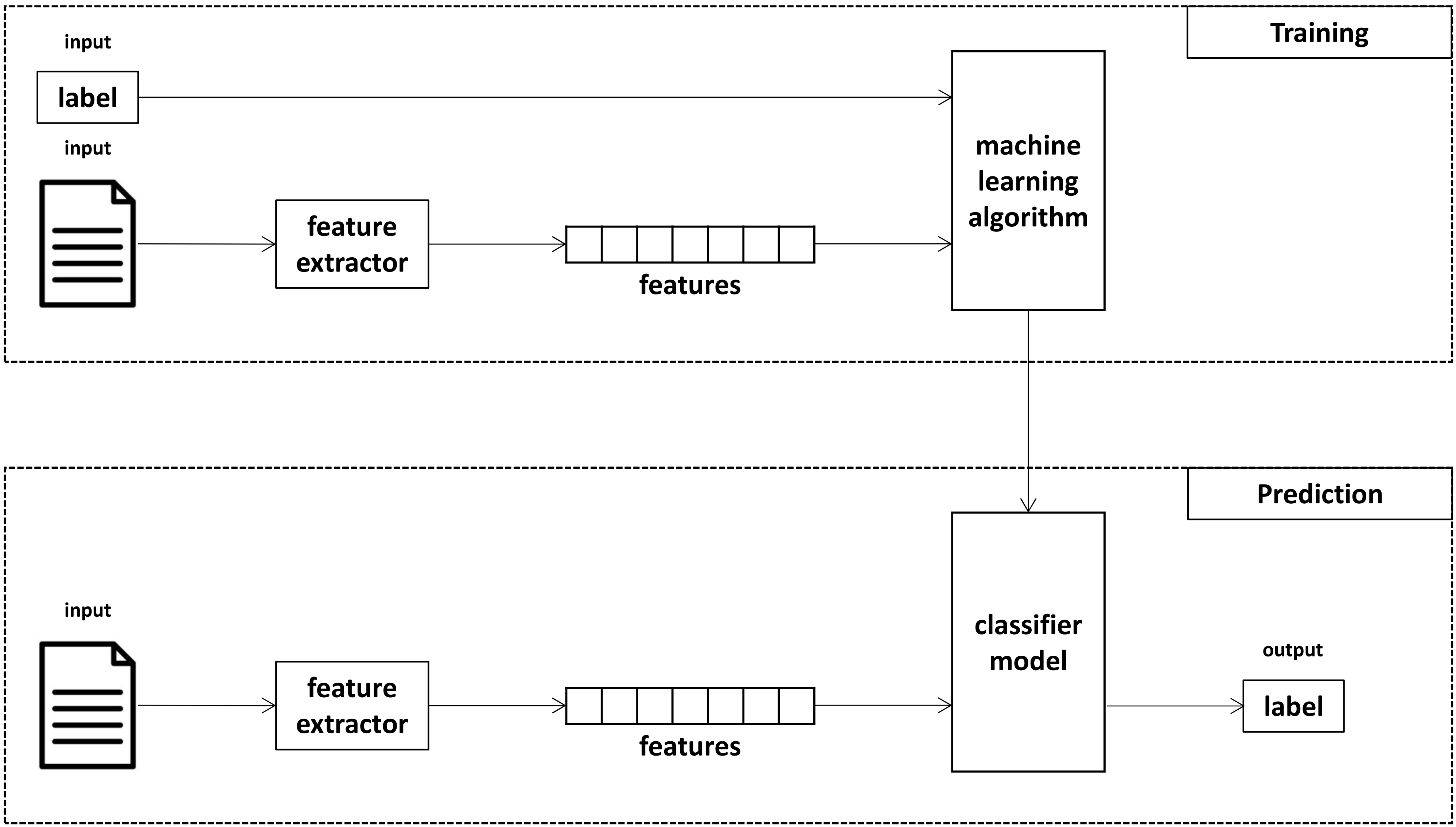
$$y = f(x)$$

discover a function (model) $h(x)$ (hypothesis) that approximates the true function $f(x)$.

Reality versus Model



Supervised Learning with ML



Choosing Hypothesis / Model

Given a training set of N example input-output (feature-label) pairs

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by

$$y = f(x)$$

Ideally, we would like our model $h(x)$ (hypothesis) that approximates the true function $f(x)$ to be:

$h(x) = y = f(x)$ **(consistent hypothesis)**

Choosing Hypothesis / Model

Typically consistent hypothesis is impossible or difficult to achieve:

- **use best-fit model / hypothesis**

Our model needs to be tested on the test set inputs (data the model has not "seen" yet) to see how well it generalizes (how accurately it predicts the outputs of the test set).

Overfitting

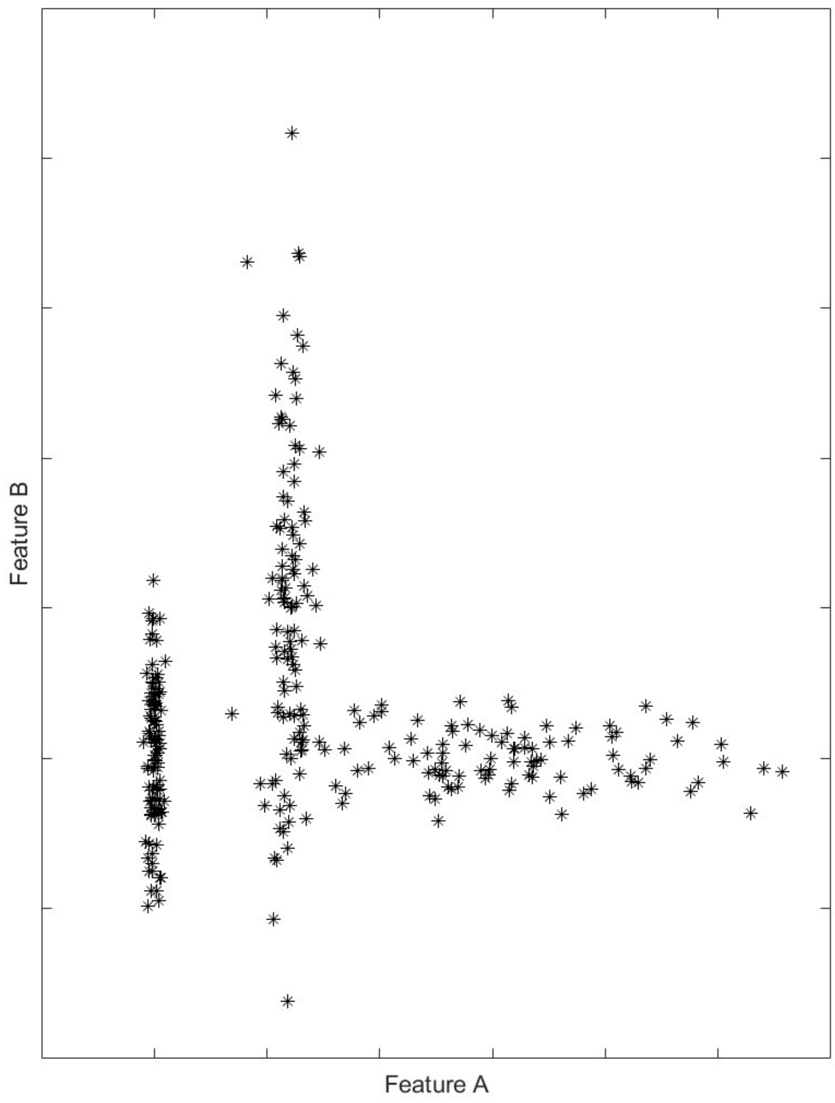
Likely to happen when using relatively small data sets.

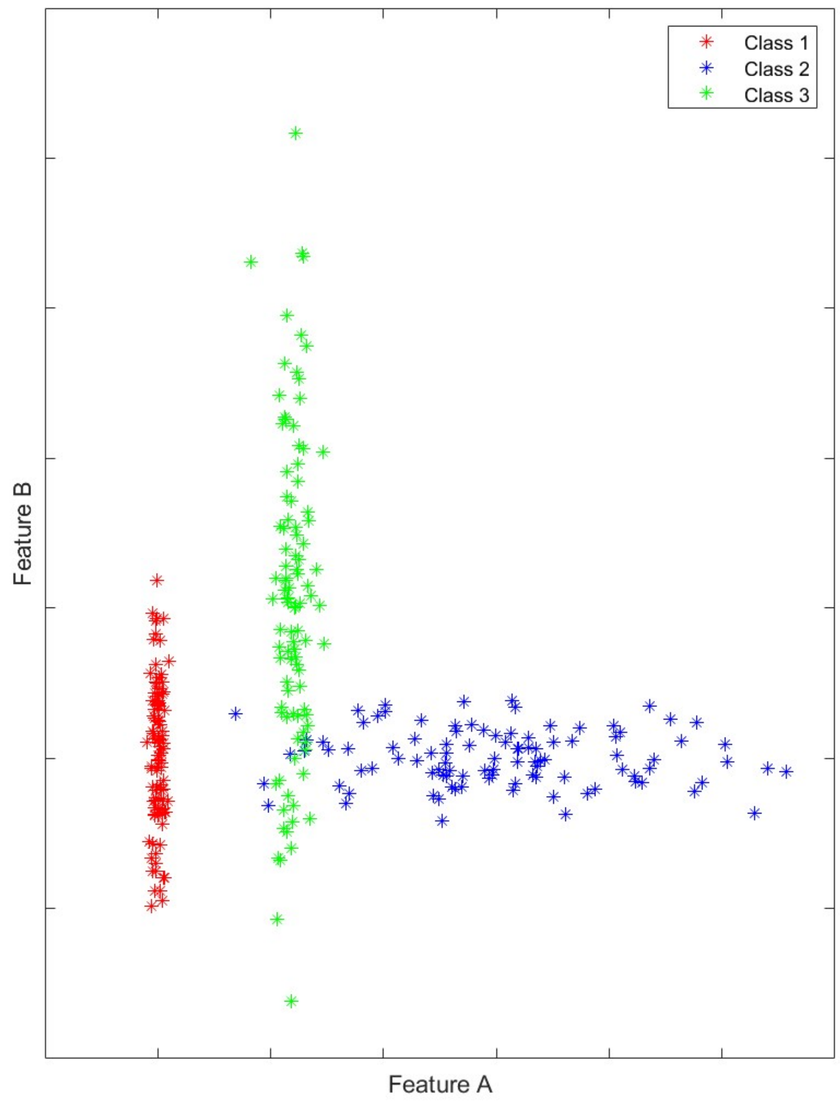
Training / Validation / Test Sets

In order to create the best model possible, given some (relatively large) data set, we should divide it into:

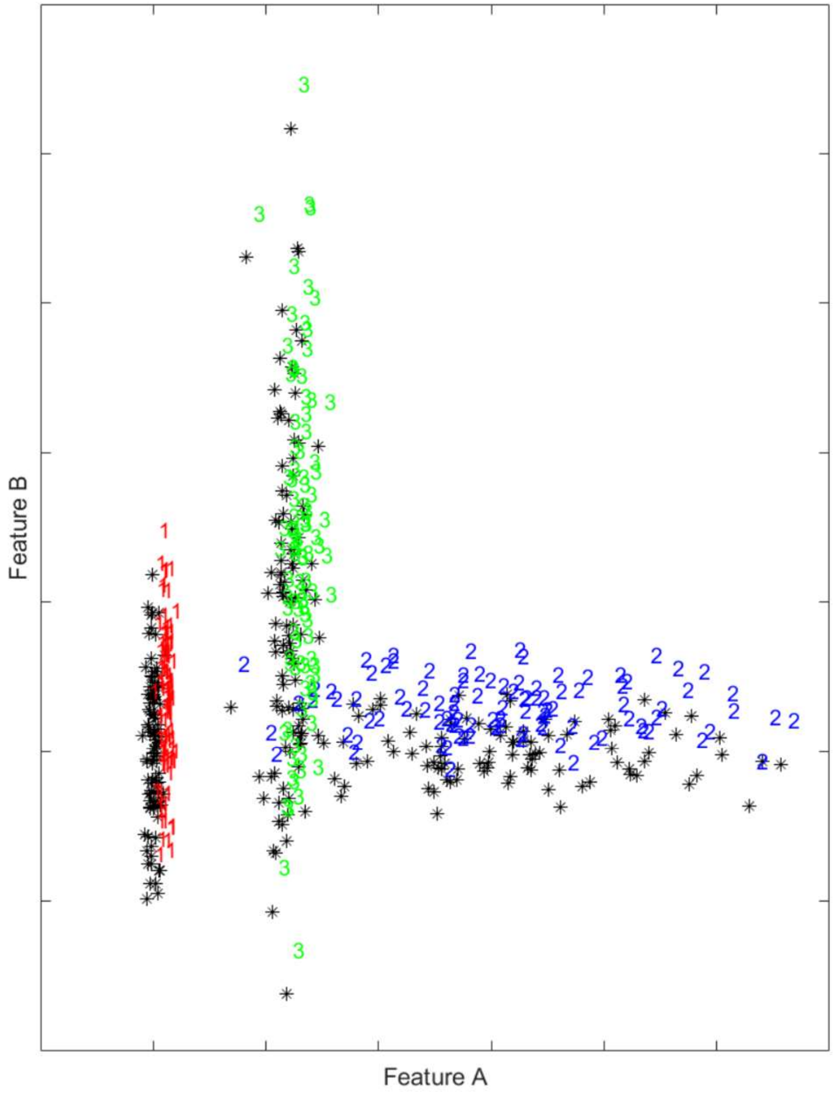
- training set: to train candidate models
- validation set: to evaluate candidate models and pick the best one
- test set: to do the final evaluation of the model

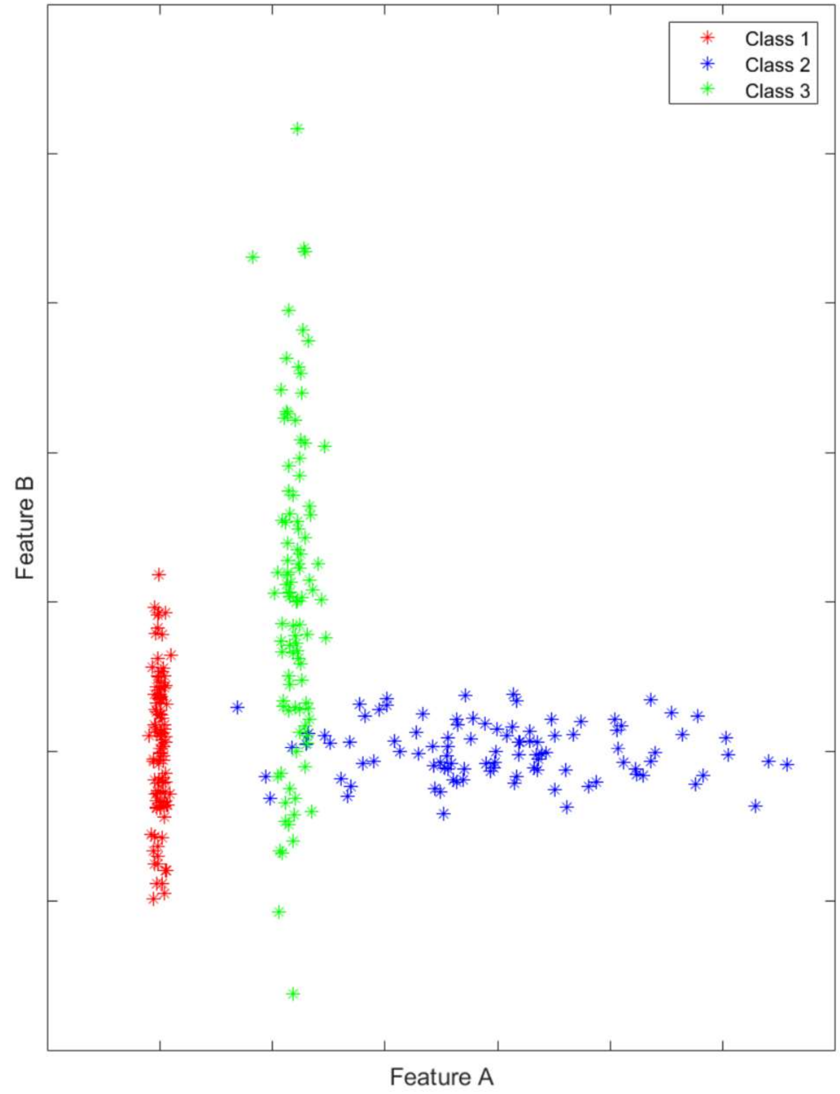
Supervised Learning: Classification



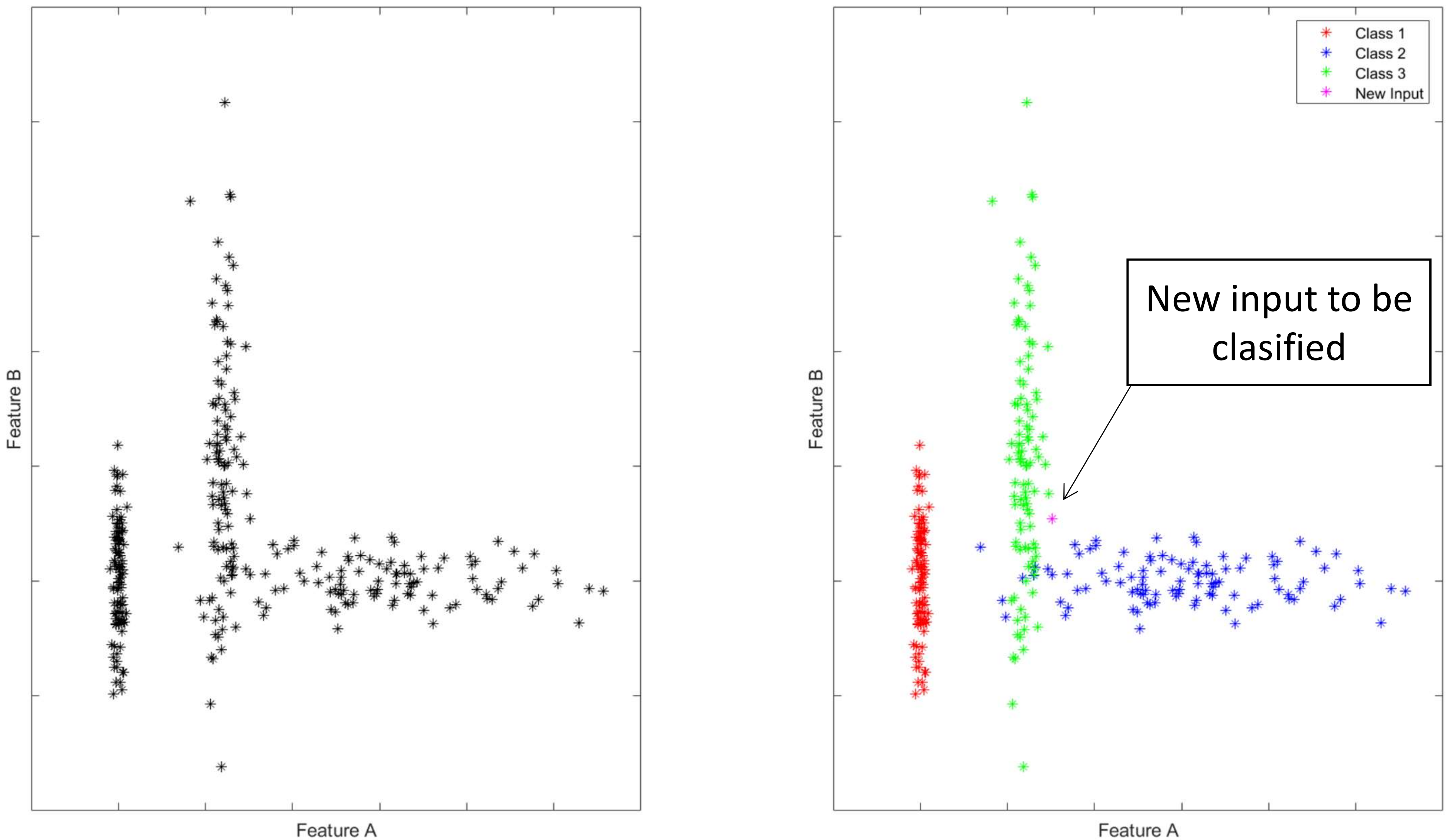


Data Set: Labeled Data

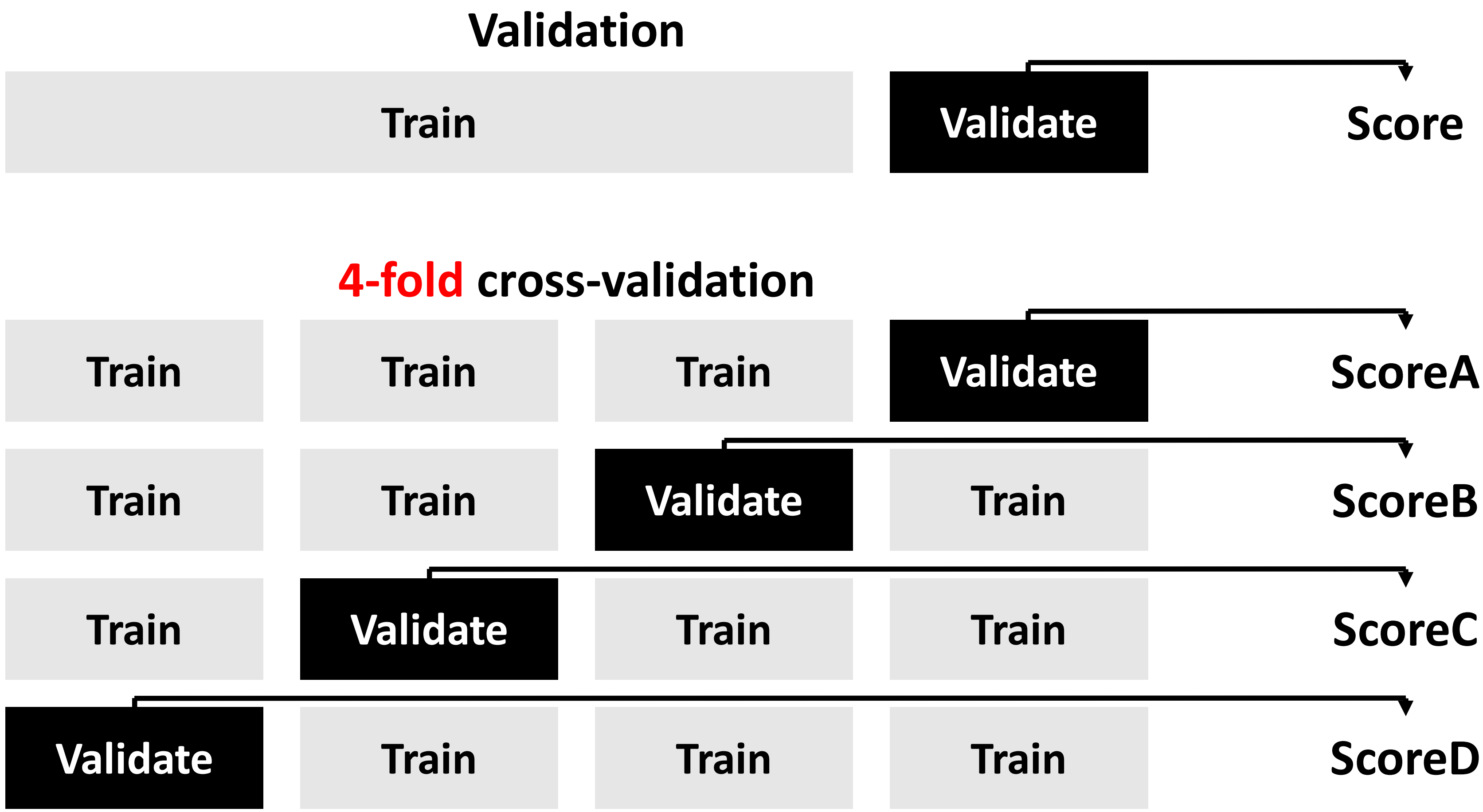




Supervised Learning: New Input



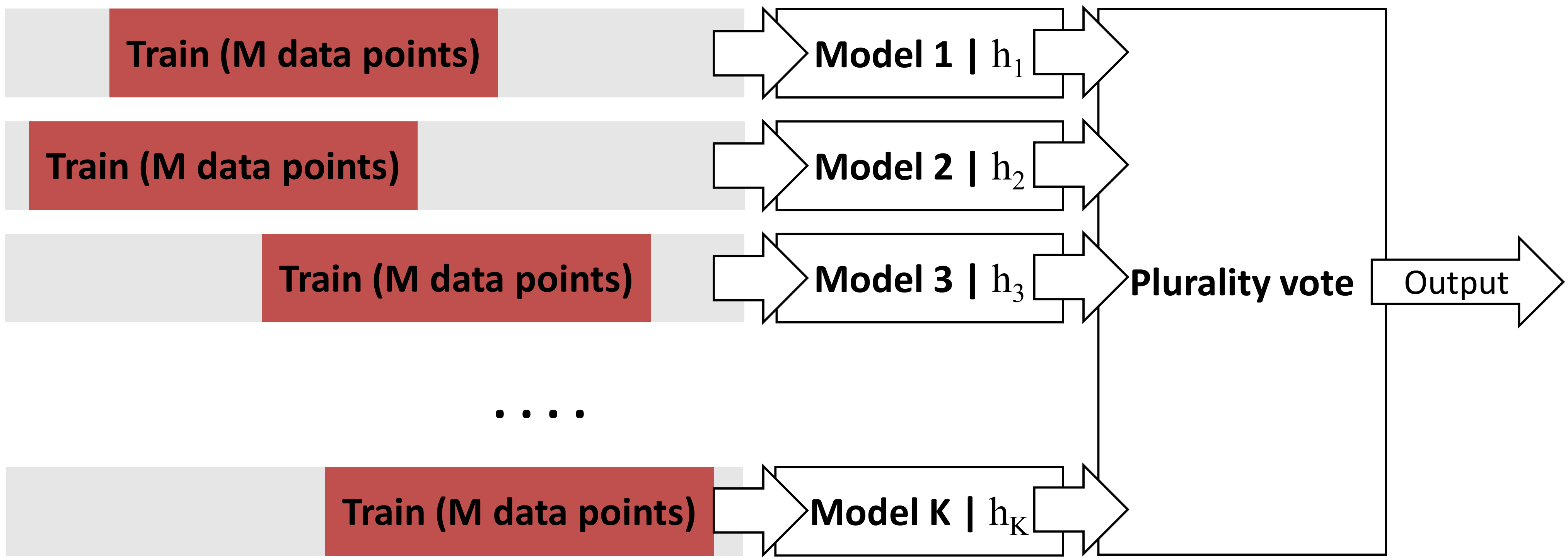
K-Fold Cross-Validation



$$\text{Score} = \frac{\text{ScoreA} + \text{ScoreB} + \text{ScoreC} + \text{ScoreD}}{4}$$

Bagging: Classification

In bagging we generate K training sets by sampling with replacement from the original training set.



Bagging tends to reduce variance and helps with smaller data sets.

Classifier Evaluation: Confusion Matrix

		Predicted class		
		Positive	**Negative**	
Actual class	**Positive**	**True Positive (TP)**	**False Negative (FN)** **Type II Error**	**Sensitivity** $\frac{TP}{TP+FN}$
	Negative	**False Positive (FP)** **Type I Error**	**True Negative (TN)**	**Specificity** $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	**Negative Predictive Value** $\frac{TN}{TN+FN}$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$

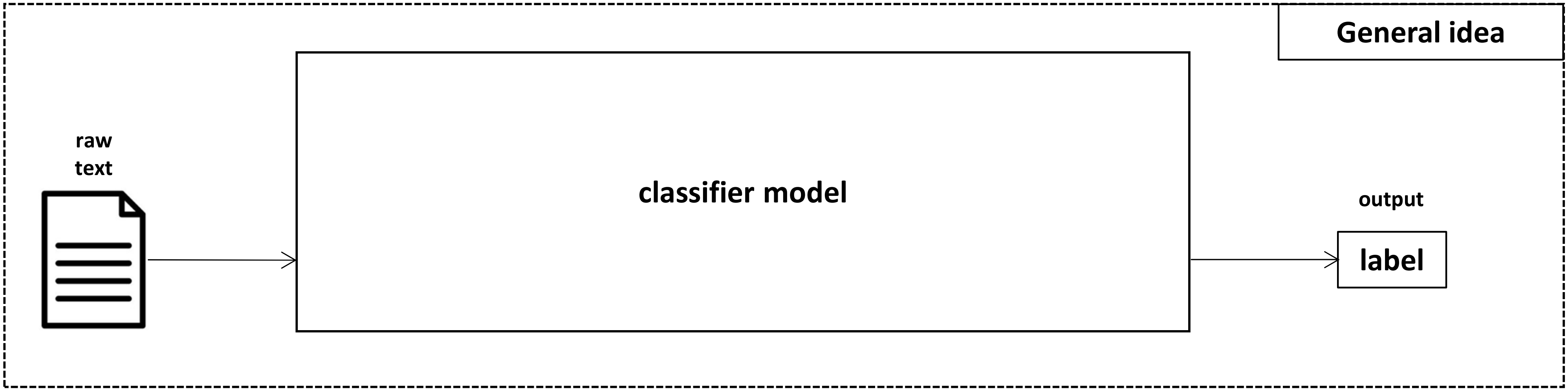
Text Classification: Definition

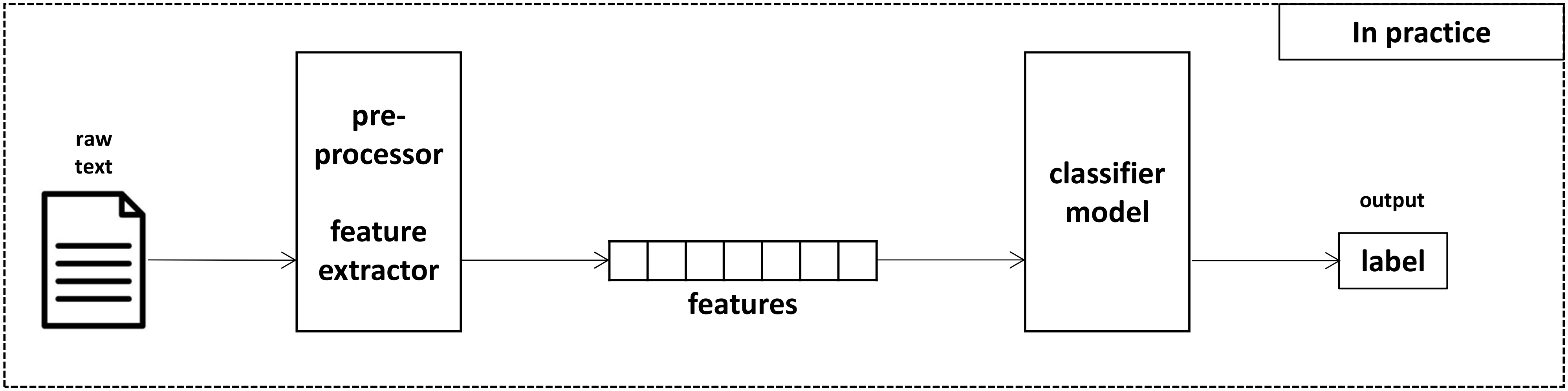
Input:

- a document d
- a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

Output: a predicted class $c \in C$

Text Classification: the Idea

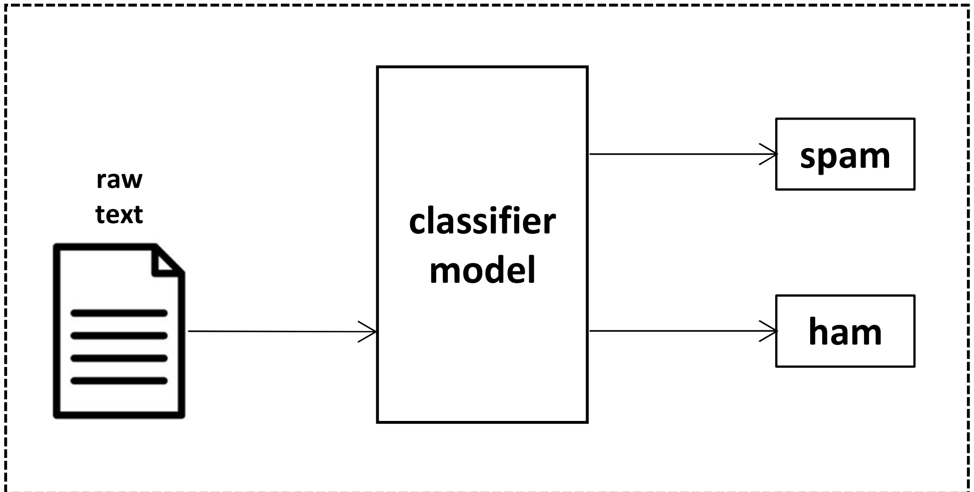




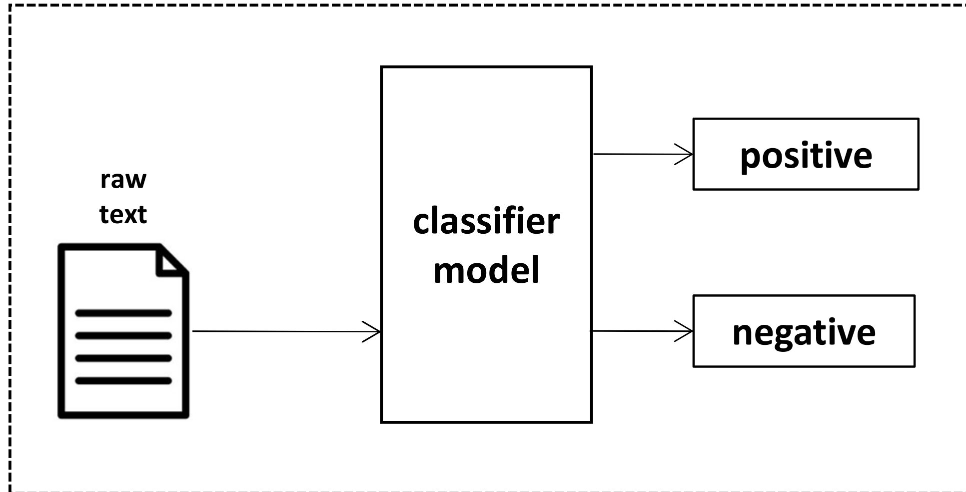
Text Classification: Applications

- Sentiment / opinion analysis
- Spam detection
- Gender identification
- Authorship identification
- Language identification
- Assigning subject categories, topics, or genres
- …

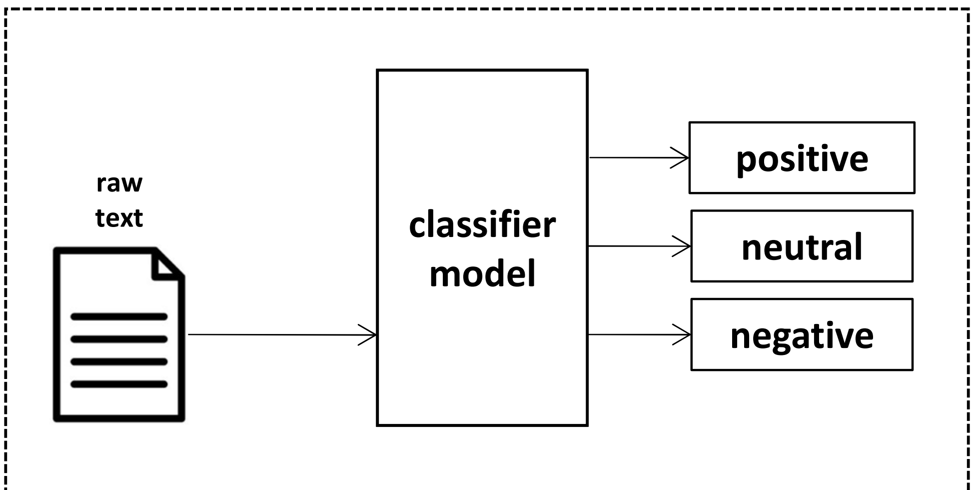
Text Classification: Applications



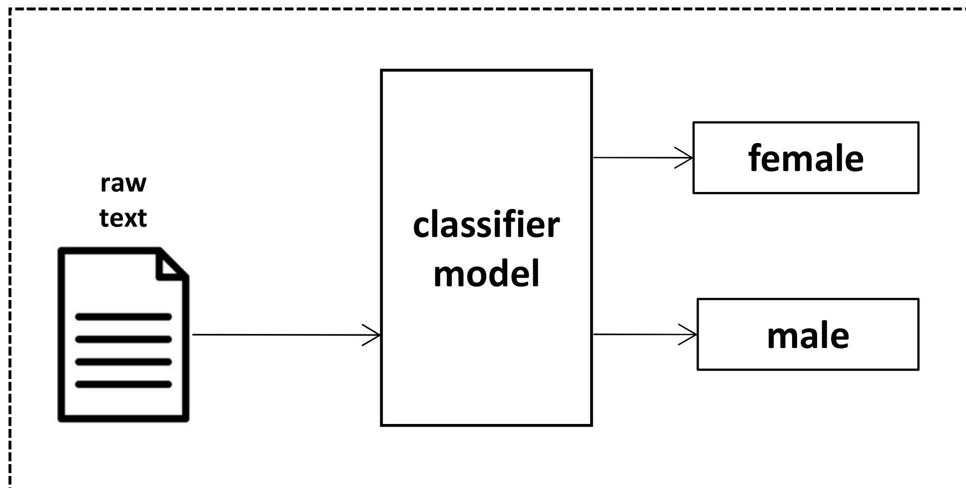
Spam classification



Sentiment and opinion analysis



Sentiment and opinion analysis



Gender identification / Attribution

Text Classification: Rule-Based

- Rules based on combinations of words or other features
  - spam: black-list-address OR (“dollars” AND “you have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Text Classification: Supervised ML

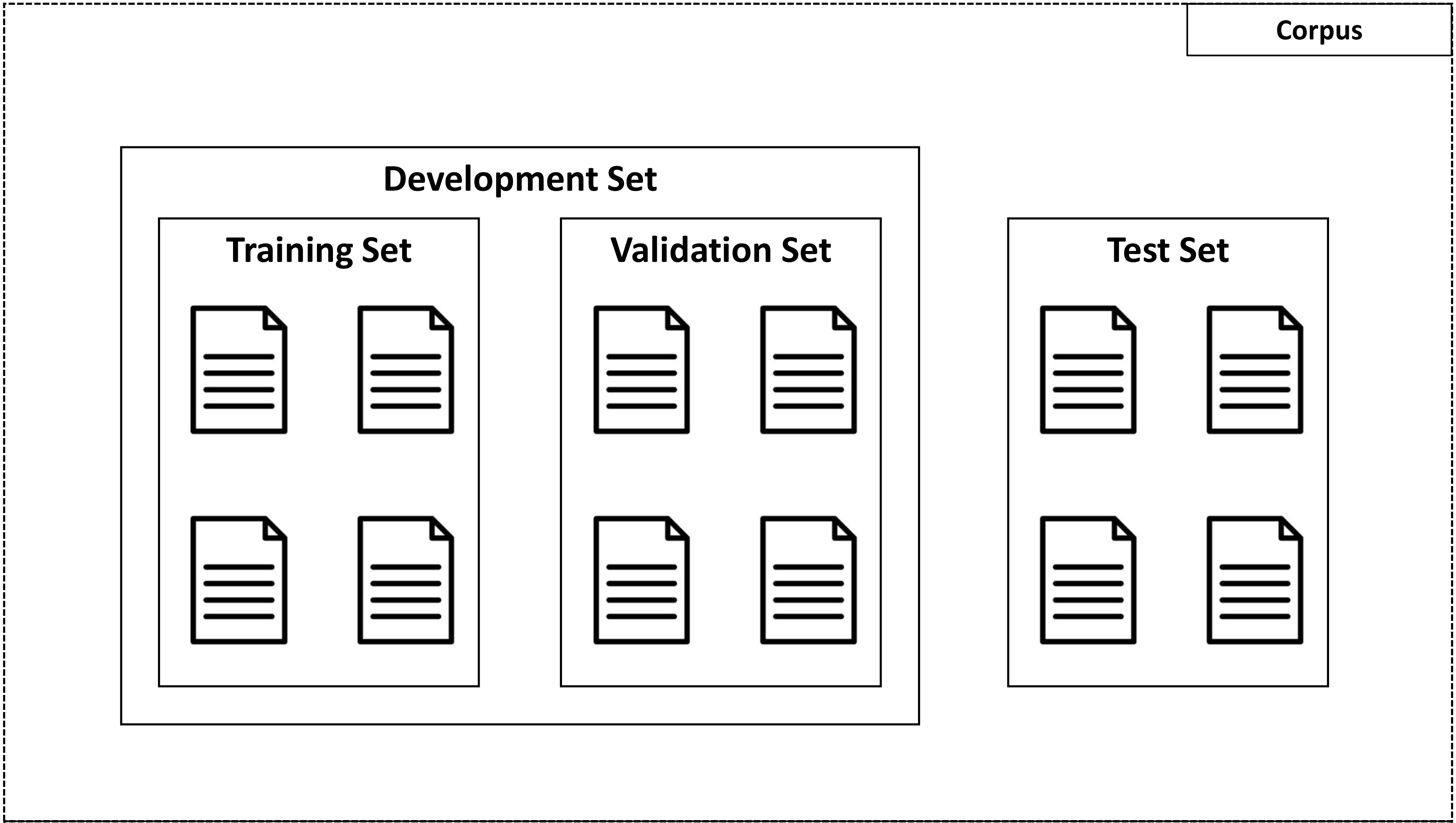
Input:

- a document d
- a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- a training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$

Output:

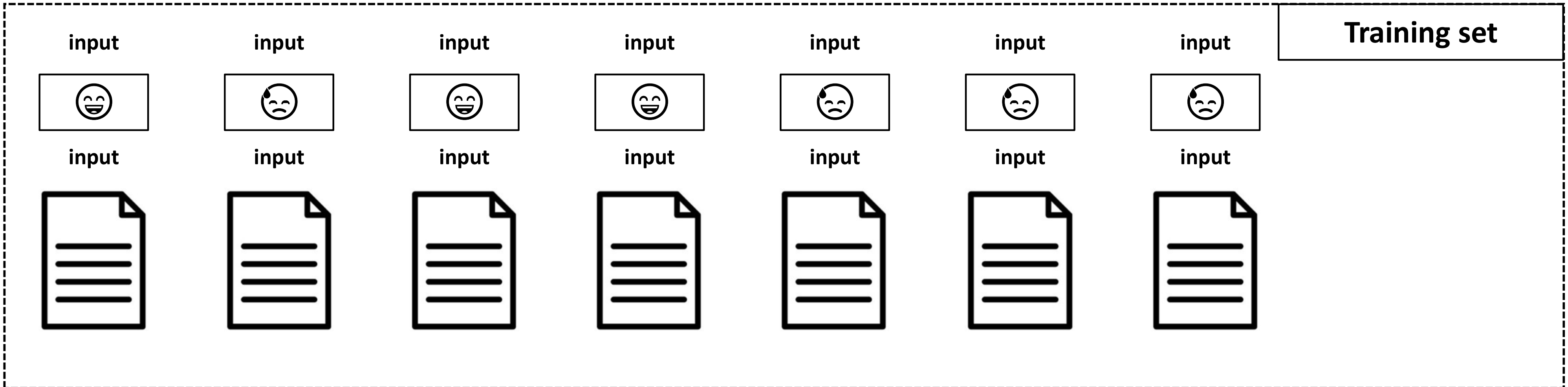
- a learned classifier $\gamma: d \rightarrow c$

Corpus: Training / Validation / Test

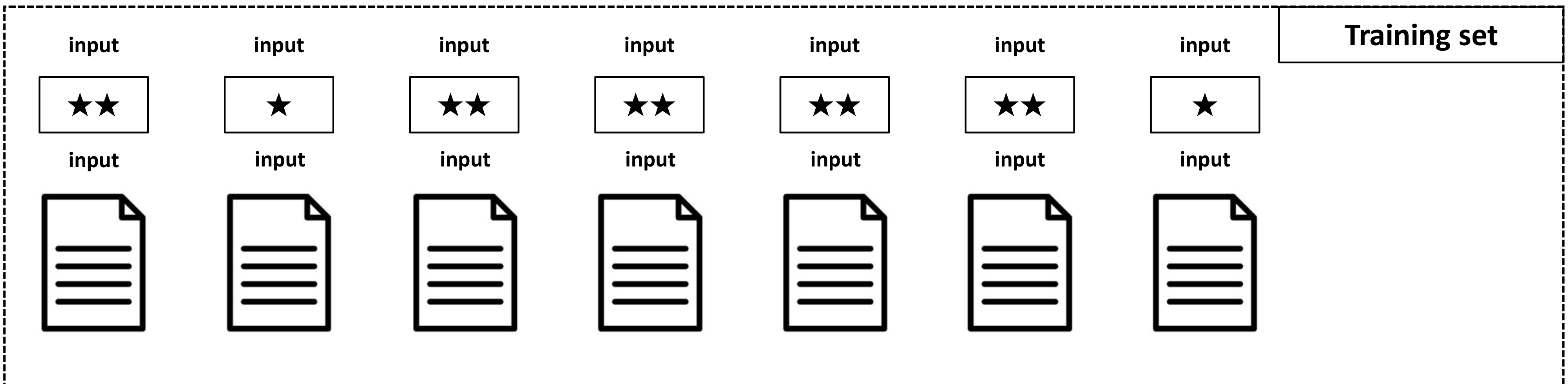


Typical training / validation / test set split for a text corpora

Text Training Set (Auto) Labeling



Look for positive (happy) and negative (sad, angry) emoticons to decide label (Twitter, Facebook, etc.)

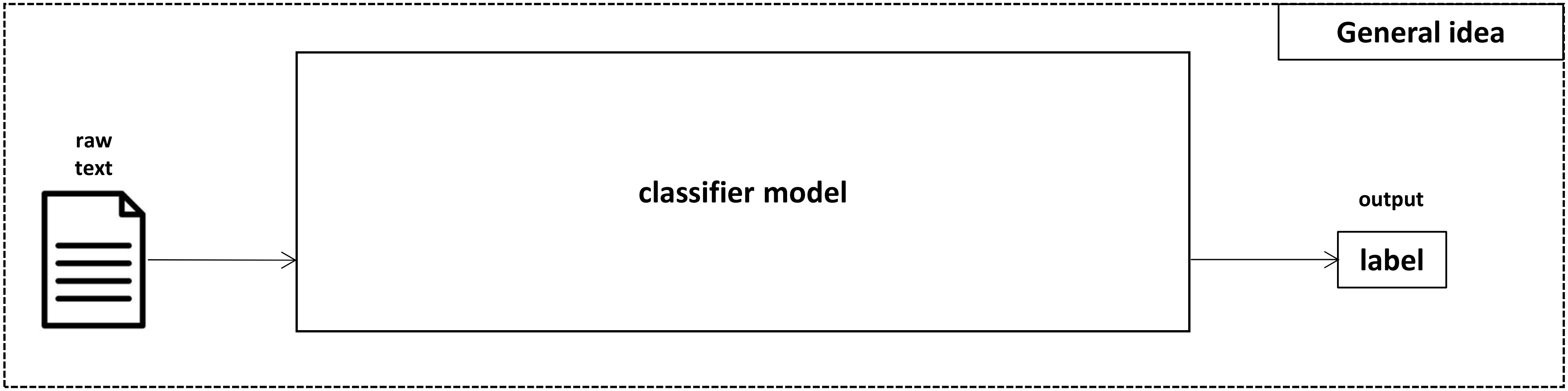


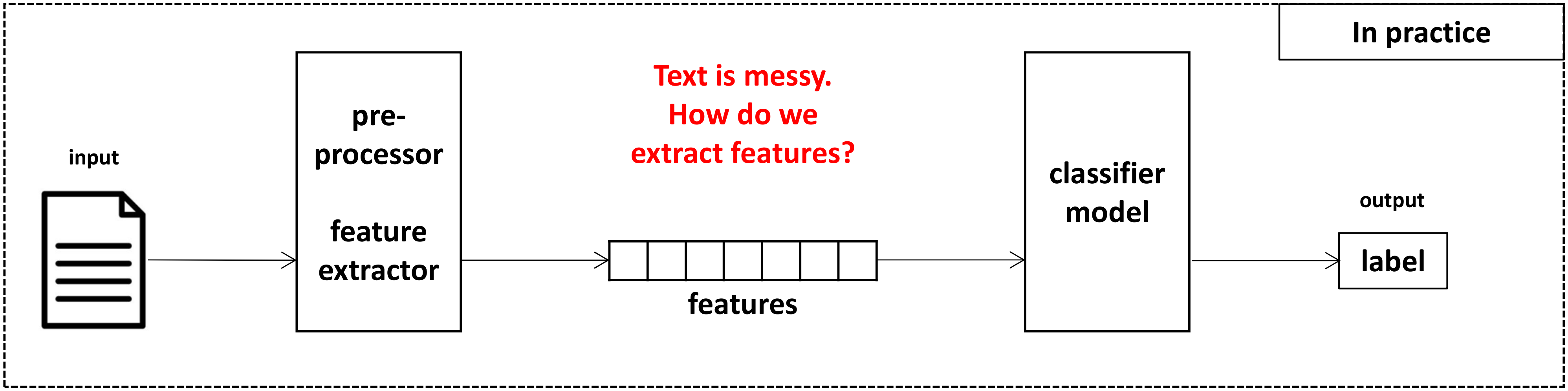
Starred reviews: ★★★★ or more stars → positive | ★★★ or less → negative (Amazon, IMdB, etc.)

Text Classification: Supervised ML

- Various Machine Learning supervised learning classifier approaches can be employed:
  - Naïve Bayes
  - Logistic regression
  - Neural networks
  - k-Nearest Neighbors
  - etc.

Text Classification: Feature Extraction





Bag of Words: the Idea

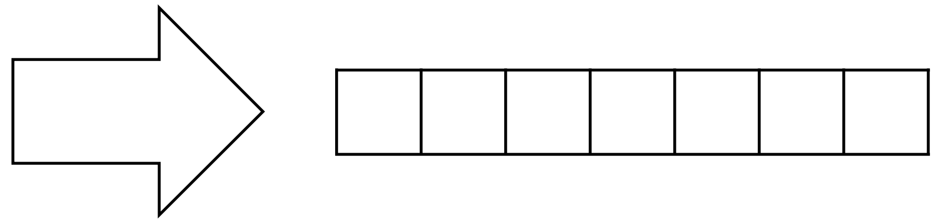
Elon Musk @elonmusk · Feb 18
So amazing that we decompress from a single cell to over 30 trillion cells
14.7K 15.2K 190.2K

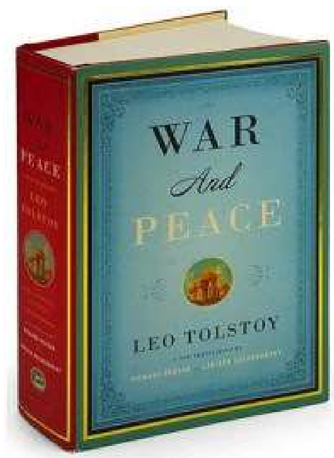
By Amy Bizzarri 1st March 2022

Obtained from the autumnal flowering of the strawberry tree on the island of Sardinia, corbezzolo honey isn't sweet and has a history that dates back more than 2,000 years.

Corbezzolo honey tricks the palate. Instead of the sweetness one would expect, this extremely rare honey, born in the mountains of the Italian island of Sardinia, is surprisingly bitter, with notes of leather, liquorice and smoke. Nomadic beekeepers have been setting up beehives in the region to collect this aromatic treat – derived from the white, bell-shaped flowers of the wild strawberry tree – for more than 2,000 years.

Statesman, lawyer and philosopher Marcus Tullius Cicero (106-43 BCE) mentioned the honey in his defence of a Roman citizen accused of murder in Nora, Sardinia. "*Omne quod Sardinia fert, homines et res, mala est! Etiam mel quod ea insula abundat, amarum est!* (Everything that the island of Sardinia produces, men and things, is bad!)," he exclaimed. "Even the honey, abundant on that island, is bitter!"



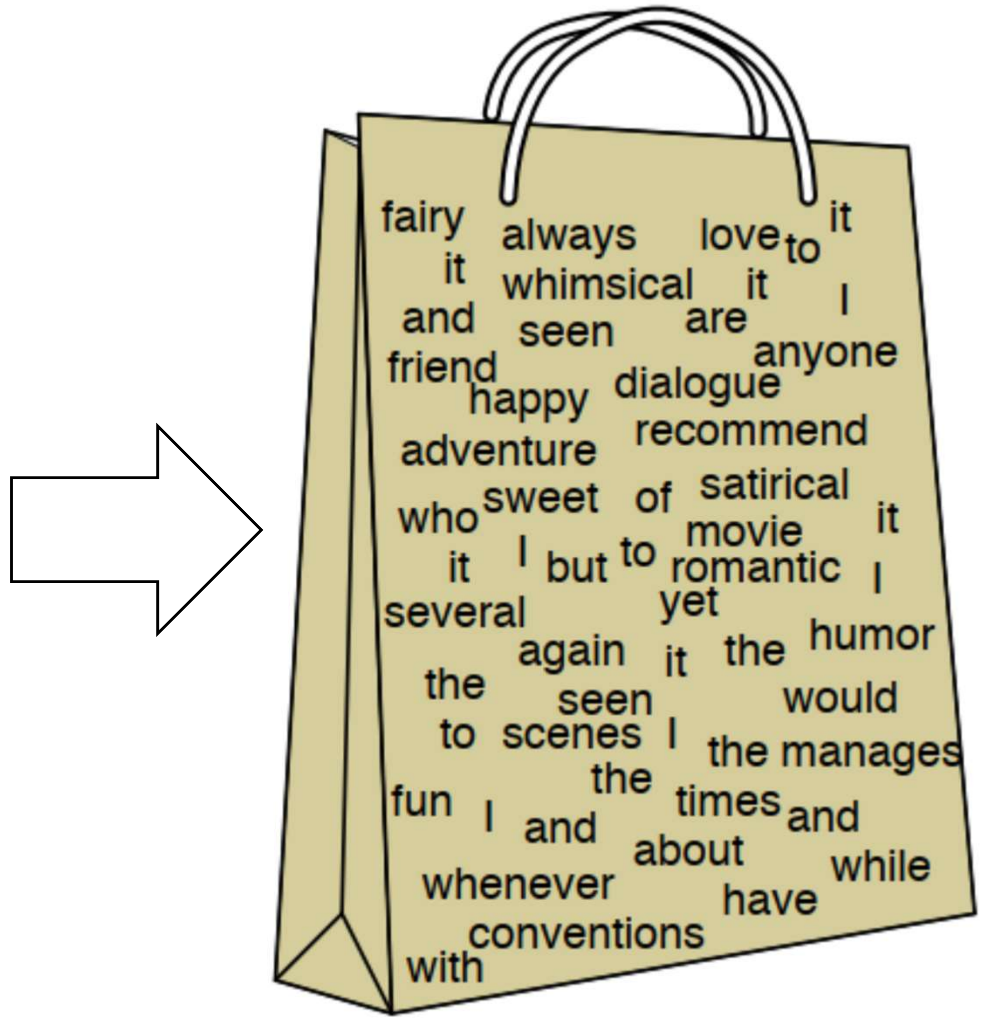


FIXED size

Feature vector

Bag of Words: the Idea

Some document:
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



Bag of words assumption: word/token position does not matter.

Bag of Words: the Idea

Some document:
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

Word:	Frequency:
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
times	1
....	...

Bag of words assumption: word/token position does not matter.

Bag of Words: Document Vector

Some document:

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

Word:	**Frequency:**
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
times	1
....	...

vector

Bag of Words: Document Vector

Pre-defined Vocabulary:							
Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	...	Word N

Document A Binary Vector [0-word absent | 1-word present]:							
1	0	1	1	1	0	. . .	1

Document B Binary Vector [0-word absent | 1-word present]:							
1	1	0	0	1	0	. . .	1

Document C Binary Vector [0-word absent | 1-word present]:							
0	0	1	0	0	1	. . .	0

Document vectors can be used to compare documents.

Bag of Words: Document Vector

Pre-defined Vocabulary:							
Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	...	Word N

Document A Non-binary Vector [0-word absent | >0-word count]:							
6	0	2	3	1	0	. . .	4

Document B Non-binary Vector [0-word absent | >0-word count]:							
4	2	0	0	5	0	. . .	1

Document C Non-binary Vector [0-word absent | >0-word count]:							
0	0	3	0	0	7	. . .	0

Document vectors can be used to compare documents.

Bag of Words: Document Vector

Pre-defined Vocabulary:							
Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	...	Word N

Document A Binary Vector [0-word absent | 1-word present]:							
1	0	1	1	1	0	. . .	1

Document B Binary Vector [0-word absent | 1-word present]:							
1	1	0	0	1	0	. . .	1

Document C Binary Vector [0-word absent | 1-word present]:							
0	0	1	0	0	1	. . .	0

Document vectors can be used to compare documents.

Document Vector = Feature Vector

Pre-defined Features:							
Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	...	Feature N

Document A Binary Vector [0-word absent | 1-word present]:							
1	0	1	1	1	0	. . .	1

Document B Binary Vector [0-word absent | 1-word present]:							
1	1	0	0	1	0	. . .	1

Document C Binary Vector [0-word absent | 1-word present]:							
0	0	1	0	0	1	. . .	0

Document vectors can be used to compare documents.

Bag of Words: Document Vector

Pre-defined Vocabulary:							
she	want	to	walk	drive	fly	there	or

“She wants to walk there today”: Binary Document Vector							
1	1	1	1	0	0	1	0

“She wants to drive there today”: Binary Document Vector							
1	1	1	0	1	0	1	0

“She wants to fly or drive there today”: Binary Document Vector							
1	1	1	0	1	1	1	1

Note: sentences lemmatized and lowercased.

Bag of Bigrams: Document Vector

Pre-defined Bigrams:							
w1, w2	w2, w3	w3, w4	w4, w5	w5, w6	w6,w7	...	wN-1,wN

Document A Binary Vector [0-word absent | 1-word present]:							
1	0	1	1	1	0	. . .	1

Document B Binary Vector [0-word absent | 1-word present]:							
1	1	0	0	1	0	. . .	1

Document C Binary Vector [0-word absent | 1-word present]:							
0	0	1	0	0	1	. . .	0

Document vectors can be used to compare documents.

Bag of Words: Classification

$$\text{category} = \mathbf{h}\left(\begin{array}{|c|} \hline 6 \\ \hline 5 \\ \hline 4 \\ \hline 3 \\ \hline 3 \\ \hline 2 \\ \hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \ldots \\ \hline \end{array} \right)$$

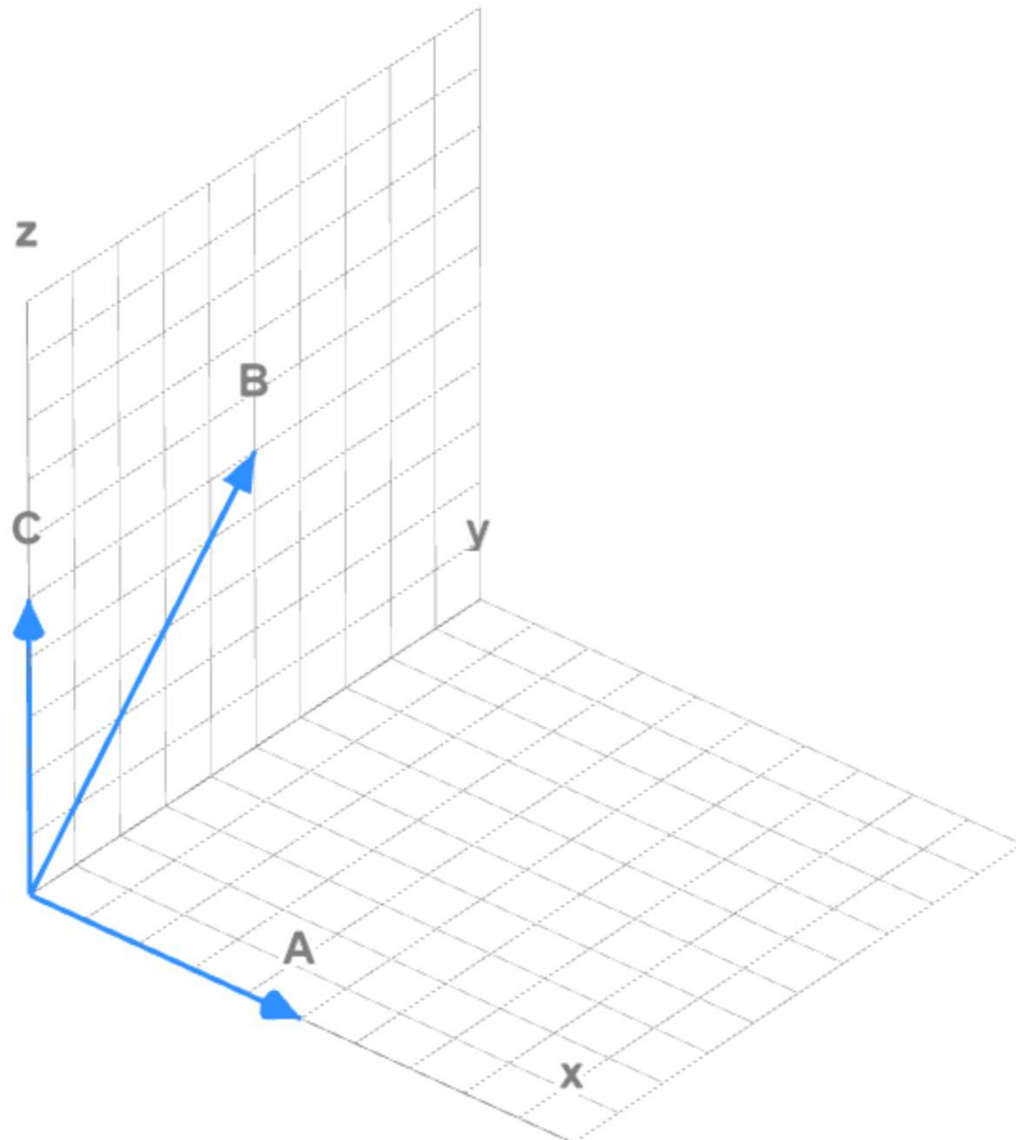
Learned Classifier model (hypothesis)

Similar Documents

=

Similar Structure

Document Vectors in Vector Space



Note: vector space can be N-dimensional (N - feature vector length).

How similar are two documents?

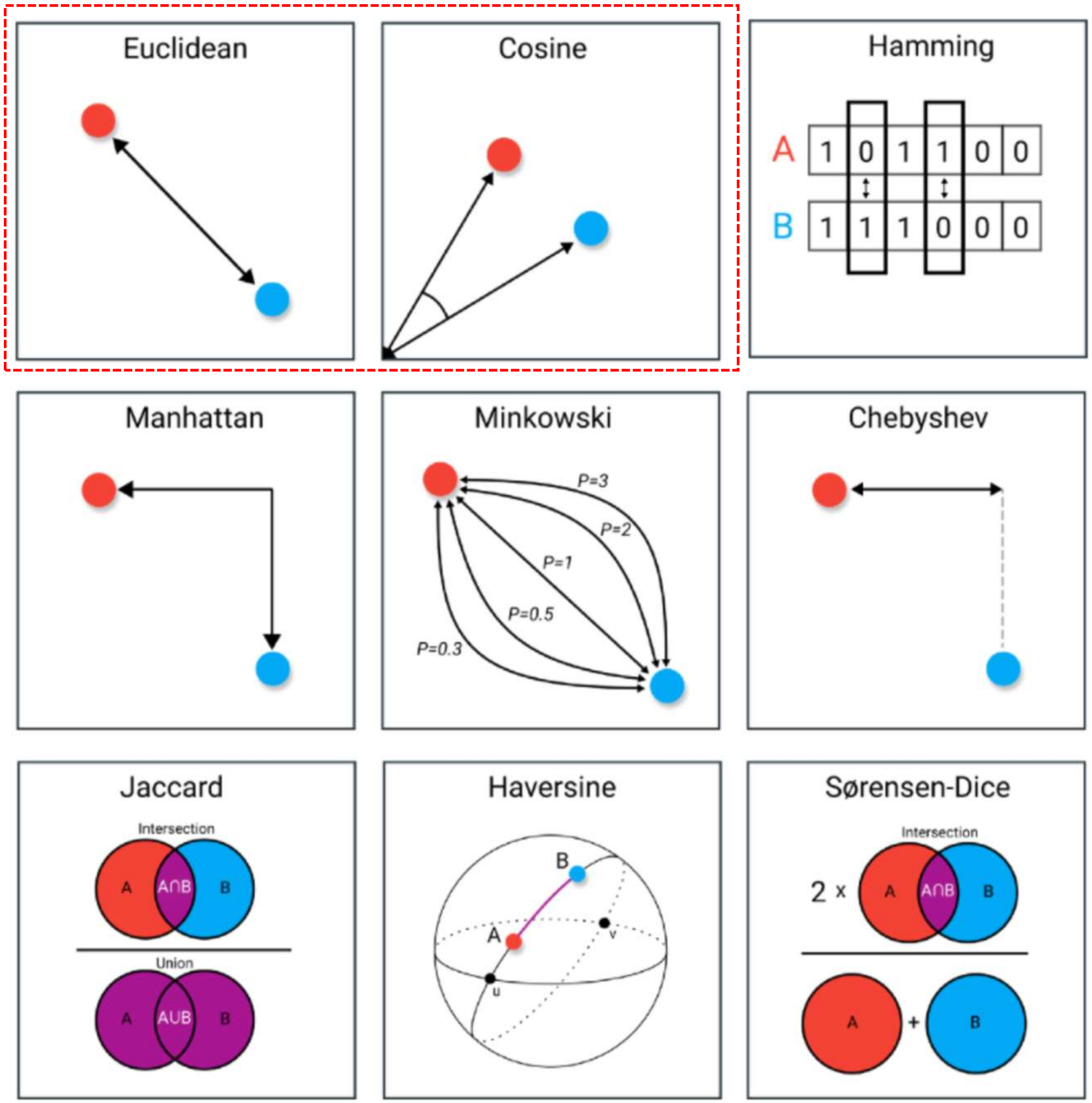
=

How similar are their structures?

=

How close (in a vector space) are points defined by their document vectors

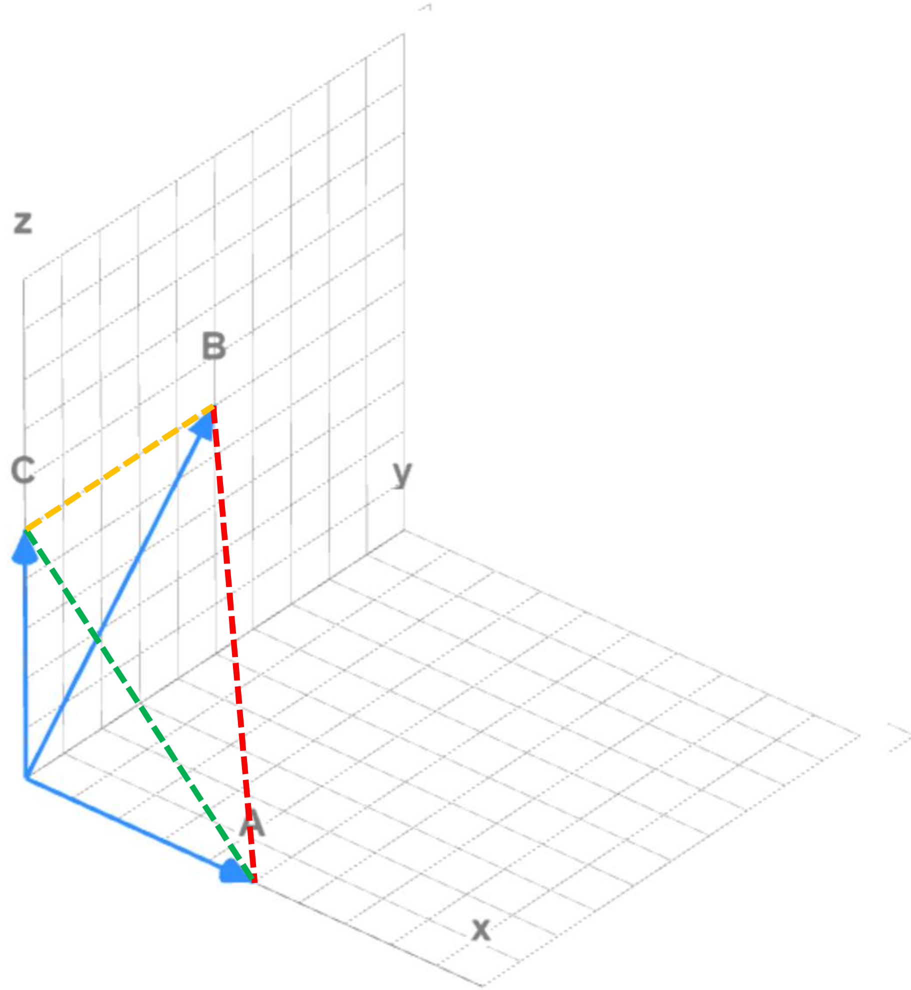
Distance Measures



Source: https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

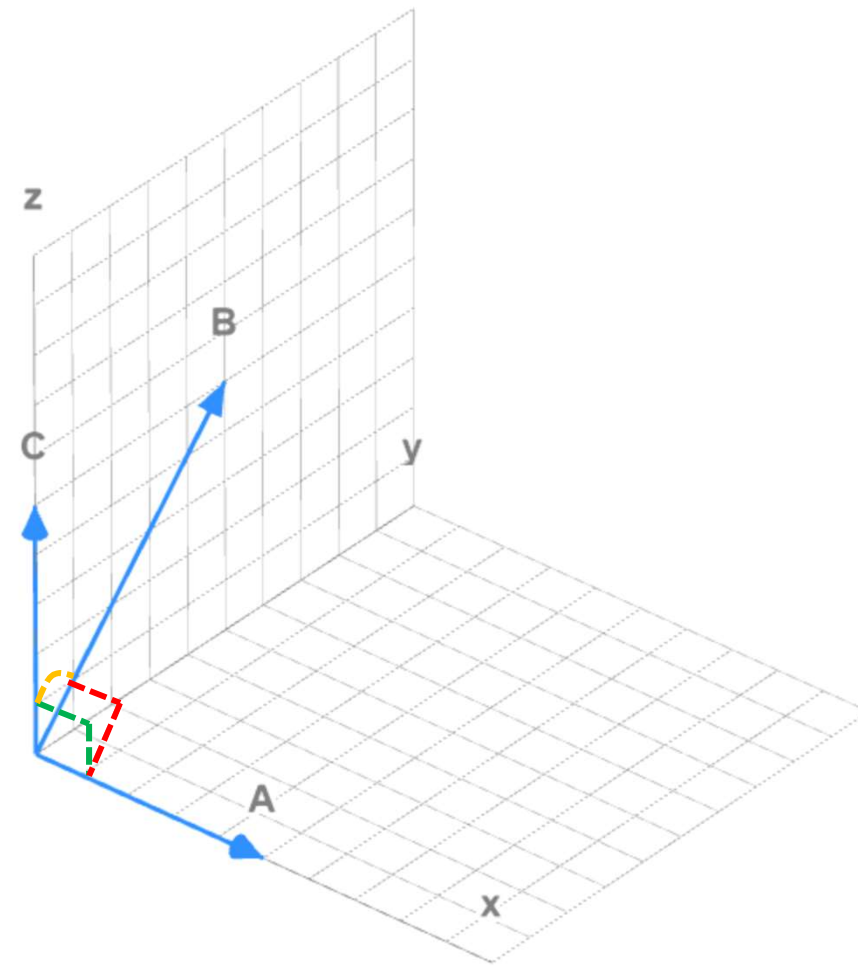
Distance Measures

Euclidean distance



$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2)}$$

Cosine similarity



$$D(x,y) = cos(\theta) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

Bag of Words: Limitations

- Word locations ignored
- Semantics ignored
  - similar / synonymous words could become distinct features

Pre-defined Vocabulary (features):							
Word 1	soccer	Word 3	Word 4	football	Word 6	...	Word N

  - similar sentences will have different vectors

buy	old	desktop	purchase	used	PC	...	Word N
"Buy old desktop" Vector [0-word absent | 1-word present]:							
1	1	1	0	0	0	. . .	0
"Purchase used PC" vector [0-word absent | 1-word present]:							
0	0	0	1	1	1	. . .	0

- New / unknown words | vocabulary range

Text Classification: Definition

Input:

- a document x
- a fixed set of classes $Y = \{y_1, y_2, \ldots, y_J\}$

Output: a predicted class $y \in Y$

Classification: Key Question

Given a document (email, tweet, etc.):

which category / class does it belong to?

Classification: Key Question

Given a document (email, tweet, etc.):

which category / class is the best (predicted) match for this document?

Classification: Key Question

Given a document (email, tweet, etc.):

which category / class is the most probable (= lowest error) for this document?

Classification: Key Question

Given a document (email, tweet, etc.):

which category / class has the highest

$P(y = \text{class} \mid \mathbf{x} =$  $)$**?**

Classification: Key Question

Which category / class has the highest

$P(y = \text{class}_1 \mid \mathbf{x} =$  $) = \ ???$

$P(y = \text{class}_2 \mid \mathbf{x} =$  $) = \ ???$

…

$P(y = \text{class}_j \mid \mid \mathbf{x} =$  $) = \ ???$

Calculate all probabilities ...

Classification: Key Question

Which category / class has the highest

$P(y = \text{class}_1 \mid \mathbf{x} =$  $) = \mathbf{0.1}$

$P(y = \text{class}_2 \mid \mathbf{x} =$  $) = \mathbf{0.3}$

...

$P(y = \text{class}_j \mid \mathbf{x} =$  $) = \mathbf{0.2}$

... and pick the maximum $P()$.

Classification: Key Question

Which category / class has the highest

$P(y = \text{class}_1 \mid \mathbf{x} =$  $) = \mathbf{0.1}$

$P(y = \text{class}_2 \mid \mathbf{x} =$  $) = \mathbf{0.3}$

…

$P(y = \text{class}_j \mid \mathbf{x} =$  $) = \mathbf{0.2}$

Corresponding class $\rightarrow$ most probable.

Classification: Key Question

Which category / class has the highest

$P(y = \text{class}_1 \mid \mathbf{x} =$  $) = ???$

$P(y = \text{class}_2 \mid \mathbf{x} =$  $) = ???$

…

$P(y = \text{class}_j \mid \mid \mathbf{x} =$  $) = ???$

Calculate all probabilities … but how?

Bayes' Rule

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}$$

Bayes' Rule: Another Interpretation

Another way to think about Bayes' rule: it allows us to update the hypothesis H in light of some new data/evidence e.

$$P(H \mid e) = \frac{P(e \mid H) * P(H)}{P(e)}$$

$$P(Hypothesis \mid evidence) = \frac{P(evidence \mid Hypothesis) * P(Hypothesis)}{P(evidence)}$$

where:

- P(H) - **probability of the** Hypothesis H **being true BEFORE we see new data/**evidence e **(<u>prior</u> probability)**
- P(H | e) - **probability of the** Hypothesis H **being true AFTER we see new data/**evidence e **(<u>posterior</u> probability)**
- P(e | H) - **probability of new data/**evidence e **being true under the** Hypothesis H **(<u>likelihood</u>)**
- P(e) - **probability of new data/**evidence e **being true under ANY hypothesis (<u>normalizing constant</u>)**

Bayes' Rule: Another Interpretation

Another way to think about Bayes' rule: it allows us to update the hypothesis H in light of some new data/evidence e.

$$P(H \mid e) = \frac{P(e \mid H) * P(H)}{P(e)}$$

$$P(Hypothesis \mid evidence) = \frac{P(evidence \mid Hypothesis) * P(Hypothesis)}{P(evidence)}$$

$$P(y \mid \boldsymbol{x}) = \frac{P(\boldsymbol{y} \mid y) * P(y)}{P(\boldsymbol{e})}$$

$$P(class \mid document) = \frac{P(document \mid class) * P(class)}{P(document)}$$

$$P(y \mid x_1, x_1, \ldots, x_N) = \frac{P(x_1, x_1, \ldots, x_N \mid y) * P(y)}{P(x_1, x_1, \ldots, x_N)}$$

for example:

$$P(y = y_k \mid x_1 = 1, x_1 = 3, \ldots, x_N = 0) = \frac{P(x_1 = 1, x_1 = 3, \ldots, x_N = 0 \mid y = y_k) * P(y = y_k)}{P(x_1 = 1, x_1 = 3, \ldots, x_N = 0)}$$

Bayes' Rule

$$posterior = \frac{likelihood * prior}{evidence}$$

Bayes' Rule

$$\boldsymbol{P(y \mid x) = \frac{P(x \mid y) * P(y)}{P(x)}}$$

$$\boldsymbol{P(Category \mid Document) = \frac{P(Document \mid Category) * P(Category)}{P(Document)}}$$

$$\boldsymbol{P(Category \mid Instance) = \frac{P(Instance \mid Category) * P(Category)}{P(Instance)}}$$

$$\boldsymbol{P(Category \mid Sample) = \frac{P(Sample \mid Category) * P(Category)}{P(Sample)}}$$

Classification: Conditional Probability

$$\boldsymbol{P}(y \mid \boldsymbol{x}) = \frac{\boldsymbol{P}(\boldsymbol{x} \mid y) * \boldsymbol{P}(y)}{\boldsymbol{P}(\boldsymbol{x})}$$

$\mathbf{x} = x_1, x_2, \ldots, x_N$, **so:**

$$\boldsymbol{P}(y \mid x_1 \wedge x_2 \wedge \ldots \wedge x_N) = \frac{\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y)}{\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N)}$$

Classification: Conditional Probability

$$P(y \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid y) * P(y)}{P(\boldsymbol{x})}$$

$$\mathbf{x} = x_1, x_2, \ldots, x_N, \text{ so:}$$

How to calculate?

$$P(y \mid x_1 \wedge x_2 \wedge \ldots \wedge x_N) = \frac{\boxed{P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y)} * P(y)}{\boxed{P(x_1 \wedge x_2 \wedge \ldots \wedge x_N)}}$$

constant

Classifier

$$y_{MAP} = \underset{y \in Y}{argmax}\left(\boldsymbol{P}(y \mid \boldsymbol{x})\right) = \underset{y \in Y}{argmax}\left(\frac{\boldsymbol{P}(\boldsymbol{x} \mid y) * \boldsymbol{P}(y)}{\boldsymbol{P}(\boldsymbol{x})}\right)$$

$\mathbf{x} = x_1, x_2, \ldots, x_N$, **so:**

$$y_{MAP} = \underset{y \in Y}{argmax}\left(\frac{\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y)}{\boxed{\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N)}}\right)$$

constant | we can drop

$$y_{MAP} \propto \underset{y \in Y}{argmax}\left(\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y)\right)$$

MAP: Maximum a posteriori (corresponds to the most likely class).

Classifier

Prior

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boxed{\boldsymbol{P}(x_1 \wedge x_2 \wedge ... \wedge x_N \mid y)} * \boxed{\boldsymbol{P}(y)} \right)$$

proportional

Likelihood

MAP: Maximum a posteriori (corresponds to the most likely class).

Classifier

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y)\right)$$

How to calculate?

MAP: Maximum a posteriori (corresponds to the most likely class).

Classifier

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boxed{\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y)}\right)$$

How to calculate?

MAP: Maximum a posteriori (corresponds to the most likely class).

Conditional Probability (Product Rule)

$$P(A \wedge B) = P(A \mid B) * P(B)$$

so:

$$P(A \mid B) * P(B) = P(A \wedge B)$$

Conditional Probability (Product Rule)

$$P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * P(y) = P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y)$$

so:

$$P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y) = P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * P(y)$$

Conditional Probability (Product Rule)

$$P(x_1 \wedge x_2 \wedge ... \wedge x_N \mid y) * P(y) = P(\underbrace{x_1}_{A} \wedge \overbrace{x_2 \wedge ... \wedge x_N \wedge y}^{B})$$

and

$$P(A \wedge B) = P(A \mid B) * P(B)$$

so:

$$P(x_1 \wedge x_2 \wedge ... \wedge x_N \wedge y) = P(x_1 \mid x_2 \wedge ... \wedge x_N \wedge y) * P(x_2 \wedge ... \wedge x_N \wedge y)$$

Chain Rule

Conditional probabilities can be used to decompose conjunctions using the chain rule. For any events $f_1, f_2, \ldots, f_n$:

$$P(f_1 \land f_2 \land \ldots \land f_n) =$$
$$P(f_1) *$$
$$P(f_2 \mid f_1) *$$
$$P(f_3 \mid f_1 \land f_2) *$$
$$\ldots$$
$$P(f_n \mid f_1 \land \ldots \land f_{n-1}) =$$
$$= \prod_{i=1}^{n} P(f_i \mid f_1 \land \ldots \land f_{i-1})$$

Expansion

$\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y) =$

$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \wedge \ldots \wedge x_N \wedge y) =$

$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \wedge \ldots \wedge x_N \wedge y) =$

$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \mid x_4 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \wedge \ldots \wedge x_N \wedge y) =$

$\ldots$

$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \ldots * \boldsymbol{P}(x_N \mid y) * \boldsymbol{P}(y)$

Independence

Assume that the knowledge of the truth of one proposition Y, does not affect the agent's belief in another proposition, X, in the context of other propositions Z. We say that X is independent of Y given Z.

Conditional Independence

Random variable X is conditionally independent of random variable Y given Z if for all $x \in Dx$, for all $y \in Dy$, and for all $z \in Dz$, such that

$$P(Y = y \wedge Z = z) > 0 \textbf{ and } P(Y = y' \wedge Z = z) > 0$$

$$P(X = x \mid Y = y \wedge Z = z) = P(X = x \mid Y = y' \wedge Z = z)$$

In other words, given a value of Z, knowing Y's value DOES NOT affect your belief in the value of X.

Conditional Independence

The following four statements are equivalent as long as conditional probabilities:

1. X **is conditionally independent of** Y **given** Z
2. Y **is conditionally independent of** X **given** Z
3. $P(X \mid Y, Z) = P(X \mid Z)$
4. $P(X, Y \mid Z) = P(X \mid Z) * P(Y \mid Z)$

Naive Bayes Assumption

$P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y) =$
$P(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * P(x_2 \wedge \ldots \wedge x_N \wedge y) =$
$P(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * P(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * P(x_3 \wedge \ldots \wedge x_N \wedge y) =$
$P(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * P(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * P(x_3 \mid x_4 \wedge \ldots \wedge x_N \wedge y) * P(x_3 \wedge \ldots \wedge x_N \wedge y) =$
$\ldots$
$P(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * P(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \ldots * P(x_N \mid y) * P(y)$

Now let's assume that all events $x_1, x_2, \ldots, x_N$ are mutually independent (not true in reality) and conditionally independent given y → Naive Bayes assumption.

Under this assumption:

$$P(x_i \mid x_{i+1} \wedge \ldots \wedge x_N \wedge y) = P(x_i \mid y)$$

Naive Bayes Assumption

Under Naive Bayes assumption:

$$\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y) =$$
$$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \wedge \ldots \wedge x_N \wedge y) =$$
$$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \wedge \ldots \wedge x_N \wedge y) =$$
$$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \mid x_4 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_3 \wedge \ldots \wedge x_N \wedge y) =$$
$$\ldots$$
$$\boldsymbol{P}(x_1 \mid x_2 \wedge \ldots \wedge x_N \wedge y) * \boldsymbol{P}(x_2 \mid x_3 \wedge \ldots \wedge x_N \wedge y) * \ldots * \boldsymbol{P}(x_N \mid y) * \boldsymbol{P}(y)$$

becomes:

$$\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \wedge y) =$$
$$\boldsymbol{P}(x_1 \mid y) * \boldsymbol{P}(x_2 \mid y) * \boldsymbol{P}(x_3 \mid y) * \ldots * \boldsymbol{P}(x_{N-1} \mid y) * \boldsymbol{P}(x_N \mid y) * \boldsymbol{P}(y) =$$
$$\boldsymbol{P}(y) * [\boldsymbol{P}(x_1 \mid y) * \boldsymbol{P}(x_2 \mid y) * \boldsymbol{P}(x_3 \mid y) * \ldots * \boldsymbol{P}(x_{N-1} \mid y) * \boldsymbol{P}(x_N \mid y)] =$$
$$\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y)$$

Naive Bayes Classifier

Under Naive Bayes assumption:

$$y_{MAP} \propto \underset{y \in Y}{argmax} (\boldsymbol{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * \boldsymbol{P}(y))$$

becomes:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{P}(x_i \mid y) \right)$$

MAP: Maximum a posteriori (corresponds to the most likely class).

Naive Bayes Classifier

Under Naive Bayes assumption:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boxed{P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y)} * P(y)\right)$$

becomes:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(P(y) * \boxed{\prod_{i=1}^{N} P(x_i \mid y)} \right)$$

MAP: Maximum a posteriori (corresponds to the most likely class).

Naive Bayes Classifier

Under Naive Bayes assumption:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(P(x_1 \wedge x_2 \wedge \ldots \wedge x_N \mid y) * P(y) \right)$$

becomes:

How to calculate?

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(P(y) * \boxed{\prod_{i=1}^{N} P(x_i \mid y)} \right)$$

MAP: Maximum a posteriori (corresponds to the most likely class).

Text Classification: Supervised ML

Input:

- a document $\boldsymbol{x}$
- a fixed set of classes $Y = \{y_1, y_2, \ldots, y_J\}$
- a training set of N hand-labeled documents $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$

Output:

- a learned classifier $h: \boldsymbol{x} \rightarrow y$ ($y = h(\boldsymbol{x})$)

Text Classification: Classifier

$$\text{category/class} = h(\mathbf{document})$$

Learned Classifier model
(hypothesis)

Text Classification: Classifier

$$y = \color{red}{h}(\mathbf{x})$$

Learned Classifier model (hypothesis)

Text Classification: Supervised ML

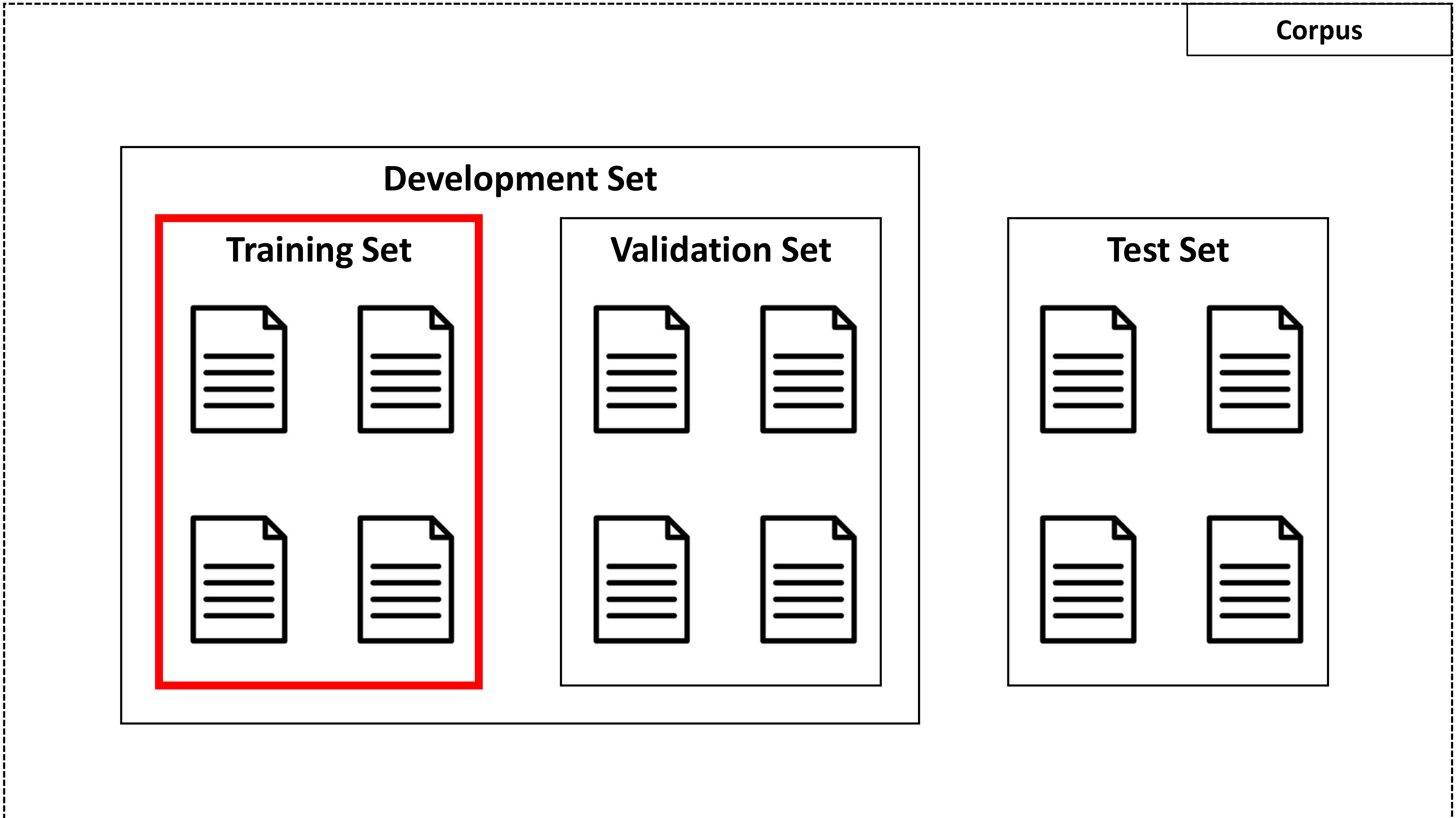
Input:

- a document $\mathbf{x}$
- a fixed set of classes $Y = \{y_1, y_2, \ldots, y_J\}$
- a **training set** of N hand-labeled documents $(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_N,y_N)$

Output:

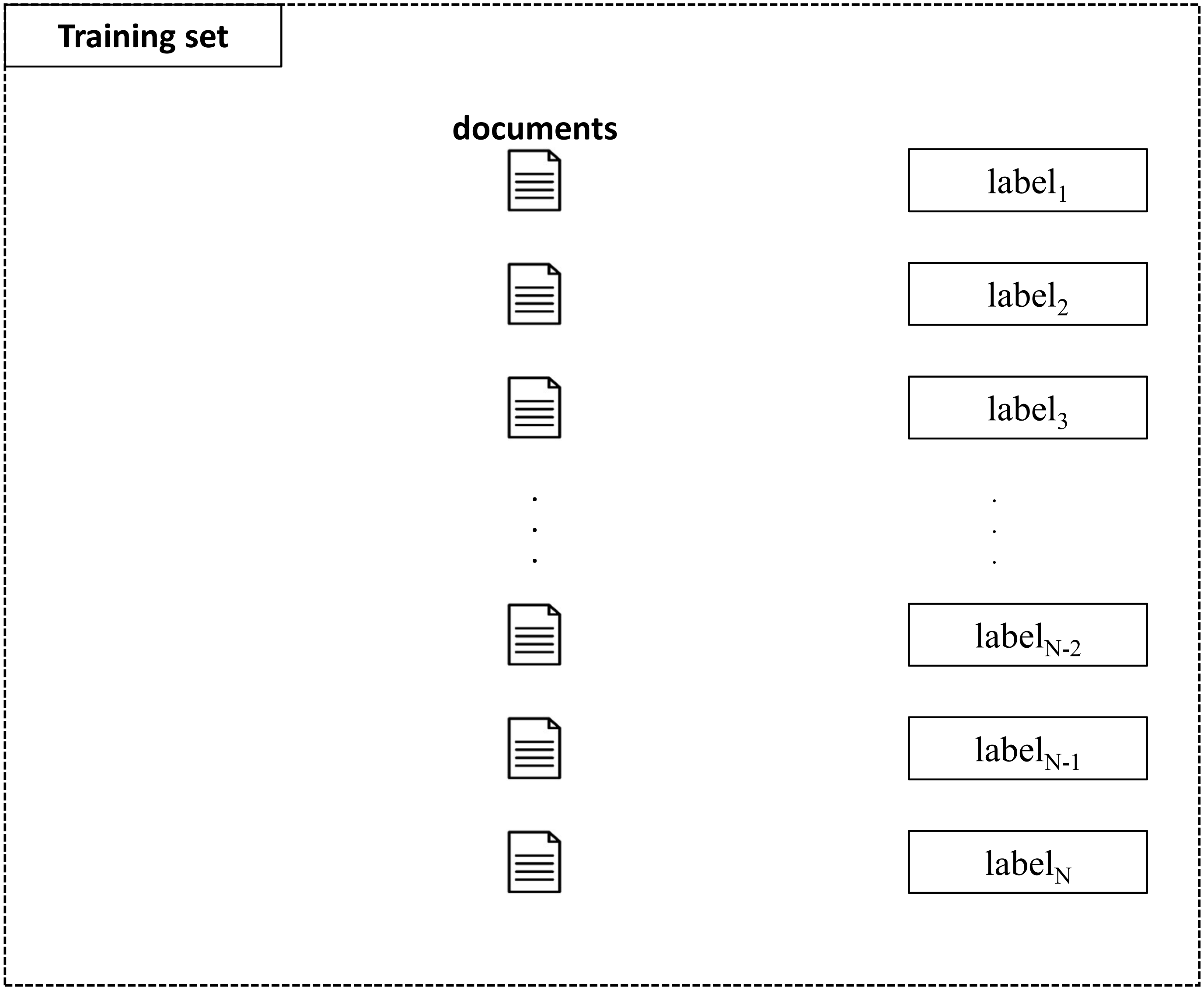
- a learned classifier $h: \mathbf{x} \rightarrow y$ ($y = h(\mathbf{x})$)

Corpus: Training / Validation / Test

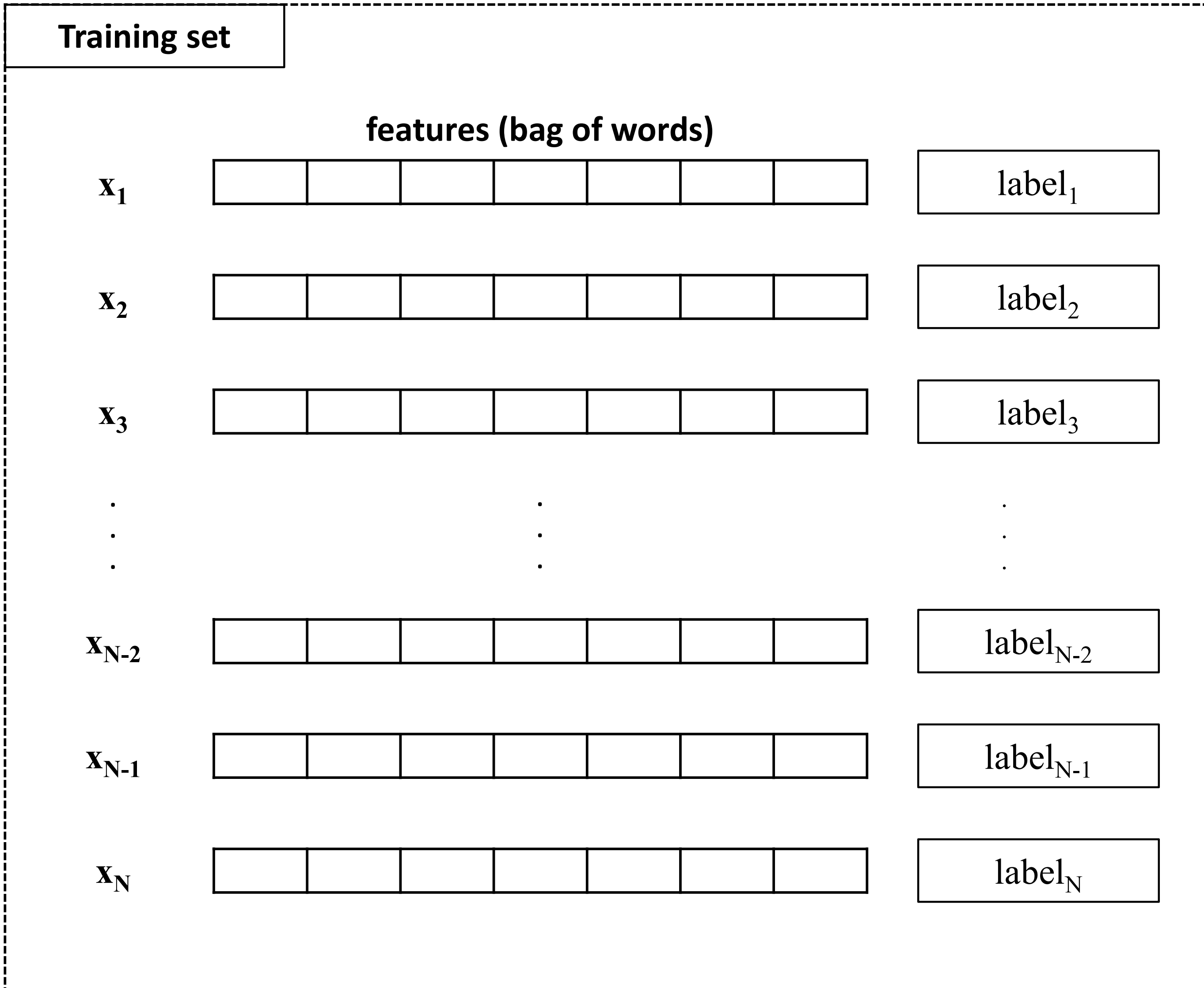


Typical training / validation / test set split for a text corpora

Text Classification: Training Set

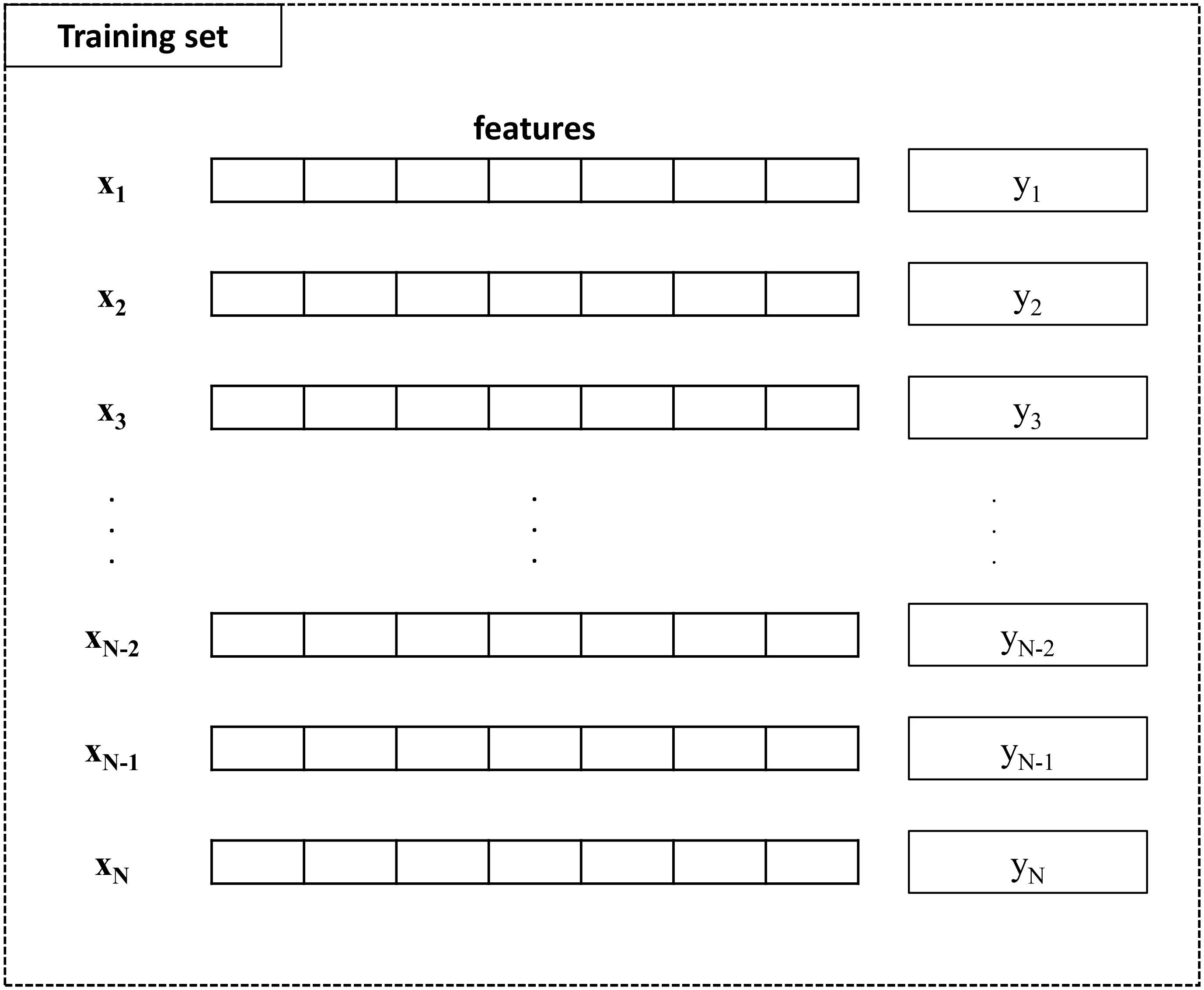


Text Classification: Training Set



$\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, …, $\mathbf{x}_{N-2}$, $\mathbf{x}_{N-1}$, $\mathbf{x}_N$ - feature vectors (in **bold**)

Text Classification: Training Set



$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Spam Detection: Training Set

Training set

Vocabulary **V**

	word1	rolex	word3	replica	word5	word6	word7	
$\mathbf{x}_1$	0	0	1	0	1	1	1	y_1=HAM
$\mathbf{x}_2$	1	0	1	1	0	1	1	y_2=HAM
$\mathbf{x}_3$	0	1	0	1	0	1	1	y_3=SPAM
. . .				. . .				. . .
$\mathbf{x}_{N-2}$	1	1	1	1	0	1	1	y_{N-2}=HAM
$\mathbf{x}_{N-1}$	1	1	0	1	0	0	1	y_{N-1}=SPAM
$\mathbf{x}_N$	1	0	0	1	0	0	1	y_N=HAM

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Text Classification: Bag of Words

Bag of words document representation (feature vector)

$$\text{category/class} = h\left(\begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 3 \\ 2 \\ 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}\right)$$

Learned Classifier model (hypothesis)

Text Classification: Bag of Words

Bag of words **binary** document representation (feature vector)

$$\text{category/class} = h\left(\begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 0 \\ \hline 1 \\ \hline 0 \\ \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline 1 \\ \hline 0 \\ \hline \end{array}\right)$$

Learned Classifier model (hypothesis)

Text Classification: Bag of Words

Bag of words document representation (feature vector)

$$\text{category/class} = h(\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{})$$

Learned Classifier model (hypothesis)

Spam Detection: Learning

Training set

Vocabulary $\mathbf{V}$

	word1	rolex	word3	replica	word5	word6	word7	
$\mathbf{x}_1$	0	0	1	0	1	1	1	y_1=HAM
$\mathbf{x}_2$	1	0	1	1	0	1	1	y_2=HAM
$\mathbf{x}_3$	0	1	0	1	0	1	1	y_3=SPAM
. . .				. . .				. . .
$\mathbf{x}_{N-2}$	1	1	1	1	0	1	1	y_{N-2}=HAM
$\mathbf{x}_{N-1}$	1	1	0	1	0	0	1	y_{N-1}=SPAM
$\mathbf{x}_N$	1	0	0	1	0	0	1	y_N=HAM

Learning

Naive Bayes Classifier:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Probability estimates (Maximium Likelihood estimation):

$$P(y_k) = \frac{N_{samples\ labeled\ y_k}}{N}$$

$$P(x_i \mid y_k) = \frac{count(x_i, y_k)}{\sum_{x \in V} count(x, y_k)}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Spam Detection: Learning

Training set

Vocabulary V

	word1	rolex	word3	replica	word5	word6	word7	
$\mathbf{x}_1$	0	0	1	0	1	1	1	y_1=HAM
$\mathbf{x}_2$	1	0	1	1	0	1	1	y_2=HAM
$\mathbf{x}_3$	0	1	0	1	0	1	1	y_3=SPAM
$\mathbf{x}_4$	1	1	1	1	0	0	0	y_4=HAM
$\mathbf{x}_5$	1	1	1	1	0	1	1	y_5=HAM
$\mathbf{x}_6$	1	1	0	1	0	0	1	y_6=SPAM
$\mathbf{x}_7$	1	0	0	1	0	0	1	y_7=HAM

Learning

Naive Bayes Classifier:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Probability estimates (Maximium Likelihood estimation):

$$P(y = HAM) = \frac{N_{samples\ labeled\ HAM}}{N} = \frac{5}{7}$$

$$P(y = SPAM) = \frac{N_{samples\ labeled\ SPAM}}{N} = \frac{2}{7}$$

$$P(x_i = rolex \mid y = SPAM) = \\ = \frac{count(x_i = rolex, y = SPAM)}{\sum_{x \in V} count(x, y = SPAM)} = \frac{2}{8}$$

and so on...

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Spam Detection: Learning

Training set

Vocabulary $\mathbf{V}$

	word1	rolex	word3	replica	word5	word6	word7	
$\mathbf{x}_1$	0	0	1	0	1	1	1	y_1=HAM
$\mathbf{x}_2$	1	0	1	1	0	1	1	y_2=HAM
$\mathbf{x}_3$	0	1	0	1	0	1	1	y_3=SPAM
⋮				⋮				⋮
$\mathbf{x}_{N-2}$	1	1	1	1	0	1	1	y_{N-2}=HAM
$\mathbf{x}_{N-1}$	1	1	0	1	0	0	1	y_{N-1}=SPAM
$\mathbf{x}_N$	1	0	0	1	0	0	1	y_N=HAM

Learning

Naive Bayes Classifier:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Probability estimates:

$$P(y_k) = \frac{N_{samples\ labeled\ y_k}}{N}$$

or

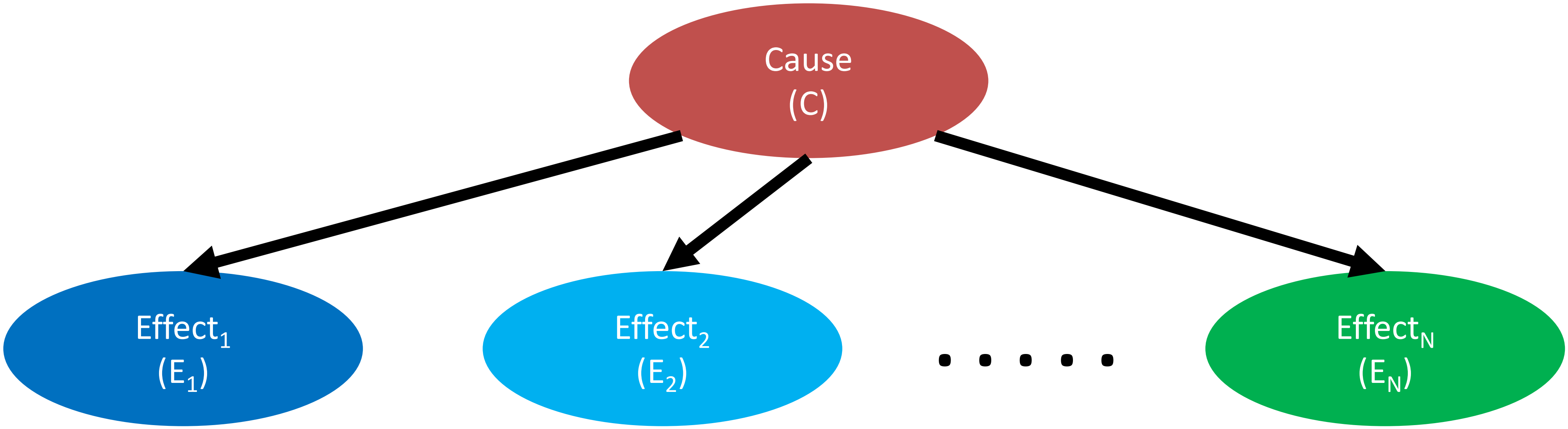
- **equiprobable (all classes have equal probability)**

$$P(y = HAM) = P(y = SPAM) = 0.5$$

- **can be determined by experts in the area**

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Naive Bayes Models



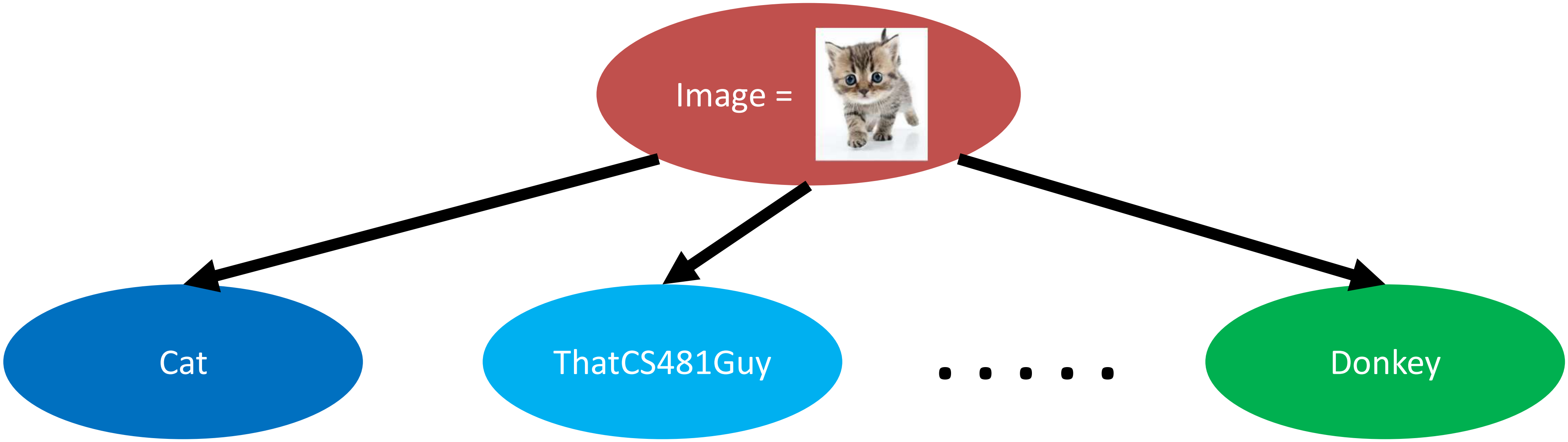
Consider a situation where all effects E_1, E_2, ..., E_N are conditionally independent given the cause. If that's true we can express full joint probability with:

$$\boldsymbol{P}(Cause, Effect_1, \ldots, Effect_N) = \boldsymbol{P}(Cause) * \prod_{\boldsymbol{i}} \boldsymbol{P}(Effect_i \mid Cause)$$

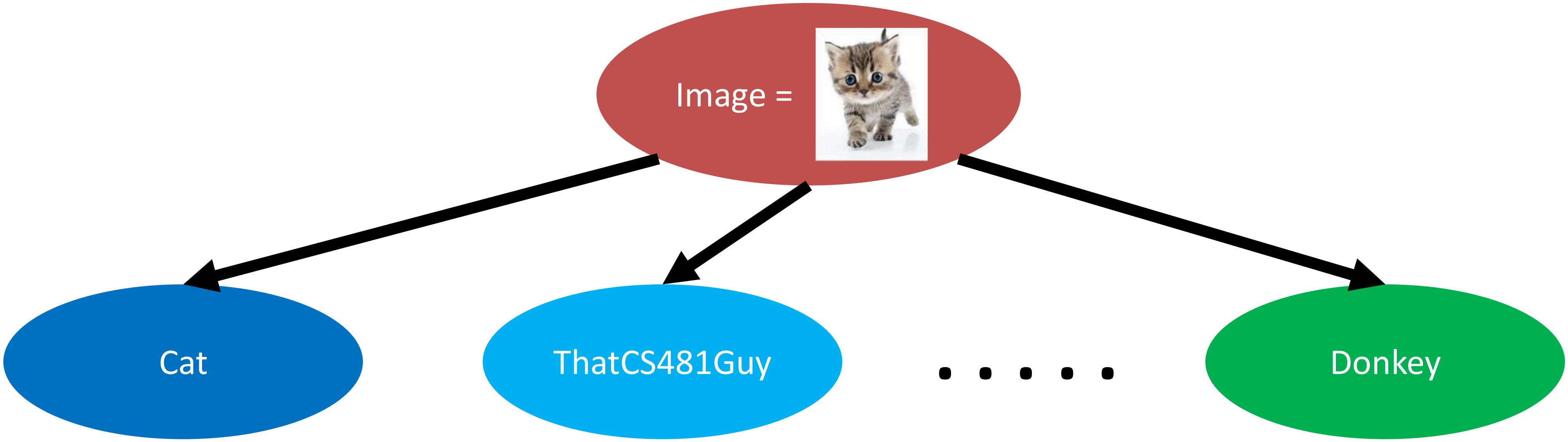
and from that:

$$\boldsymbol{P}(Cause|\boldsymbol{e}) = \alpha * \boldsymbol{P}(Cause) * \prod_{\boldsymbol{j}} \boldsymbol{P}(e_j \mid Cause)$$

Naive Bayes “Classifier”



Naive Bayes "Classifier"



P(Image | Cat) = 0.9

P(Image | ThatCS481Guy) = 0.01

…

P(Image | Donkey) = 0.03