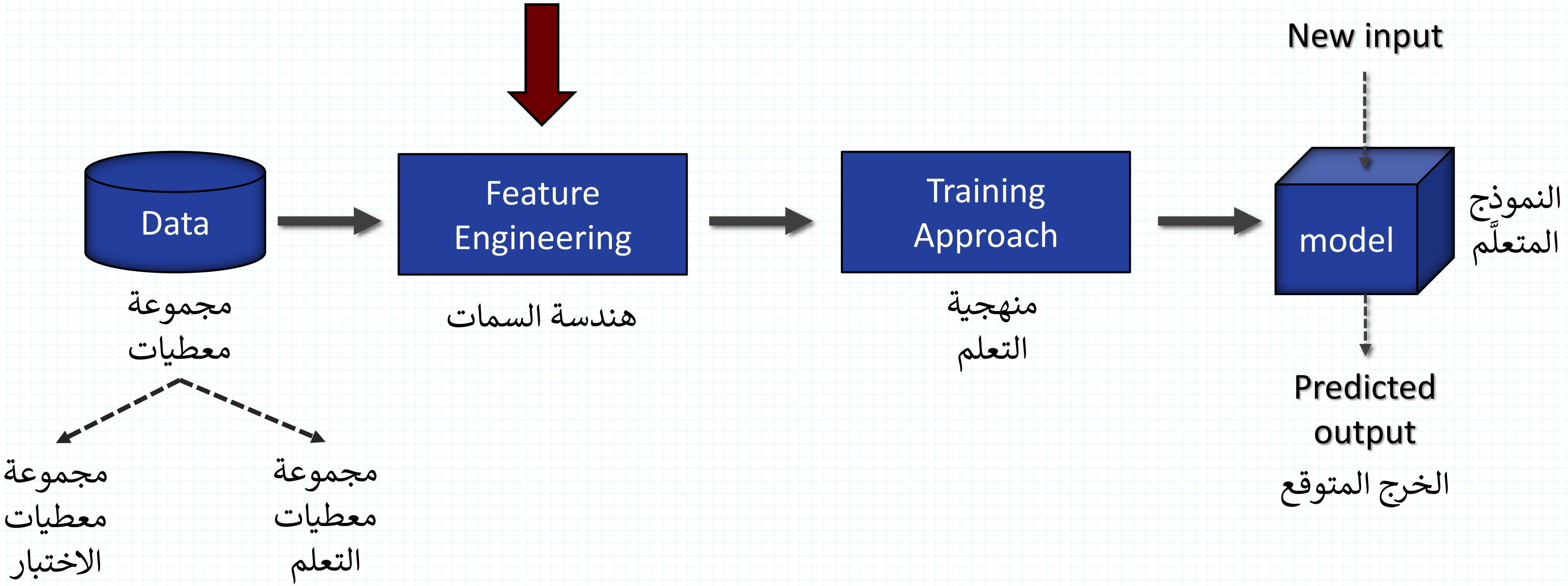
الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

تعلم الآلة | كلية الهندسة المعلوماتية | المحاضرة السابعة

هندسة السمات
Feature Engineering

د. رياض سنبل

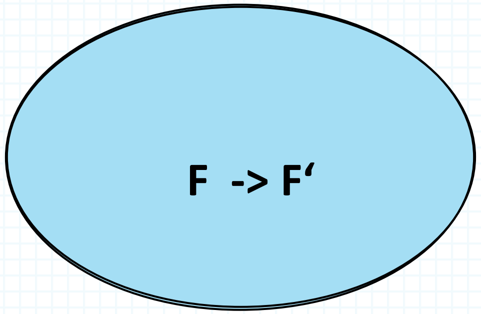
ML Pipeline


New input
Data
Feature Engineering
Training Approach
model
النموذج المتعلَّم
مجموعة معطيات
هندسة السمات
منهجية التعلم
Predicted output
الخرج المتوقع
مجموعة معطيات الاختبار
مجموعة معطيات التعلم

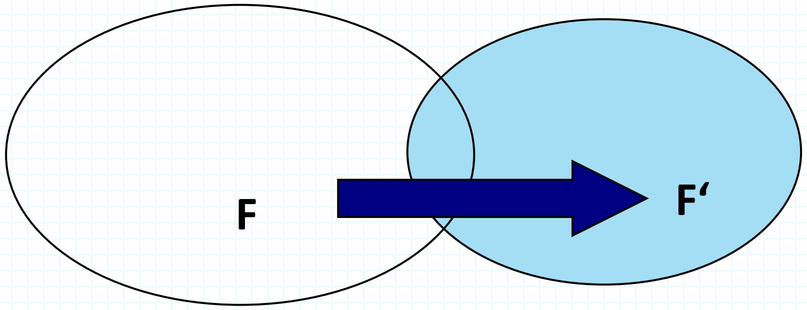
Feature (Preprocessing vs Selection vs Extraction)

- **Feature Preprocessing:** Clean, normalize, transform features the values of specific feature using a defined formula.
- **Feature extraction:** Creates new features (dimensions) defined as functions over all features
- **Feature selection:** Chooses subset of features

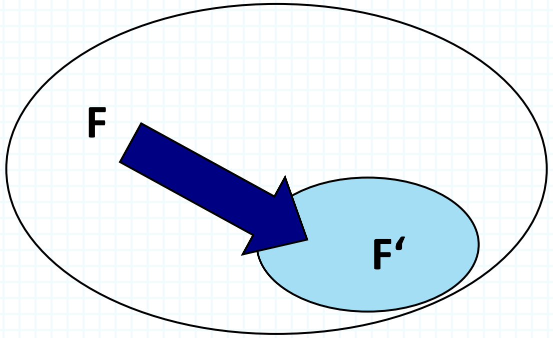
Feature Preprocessing



Feature extraction

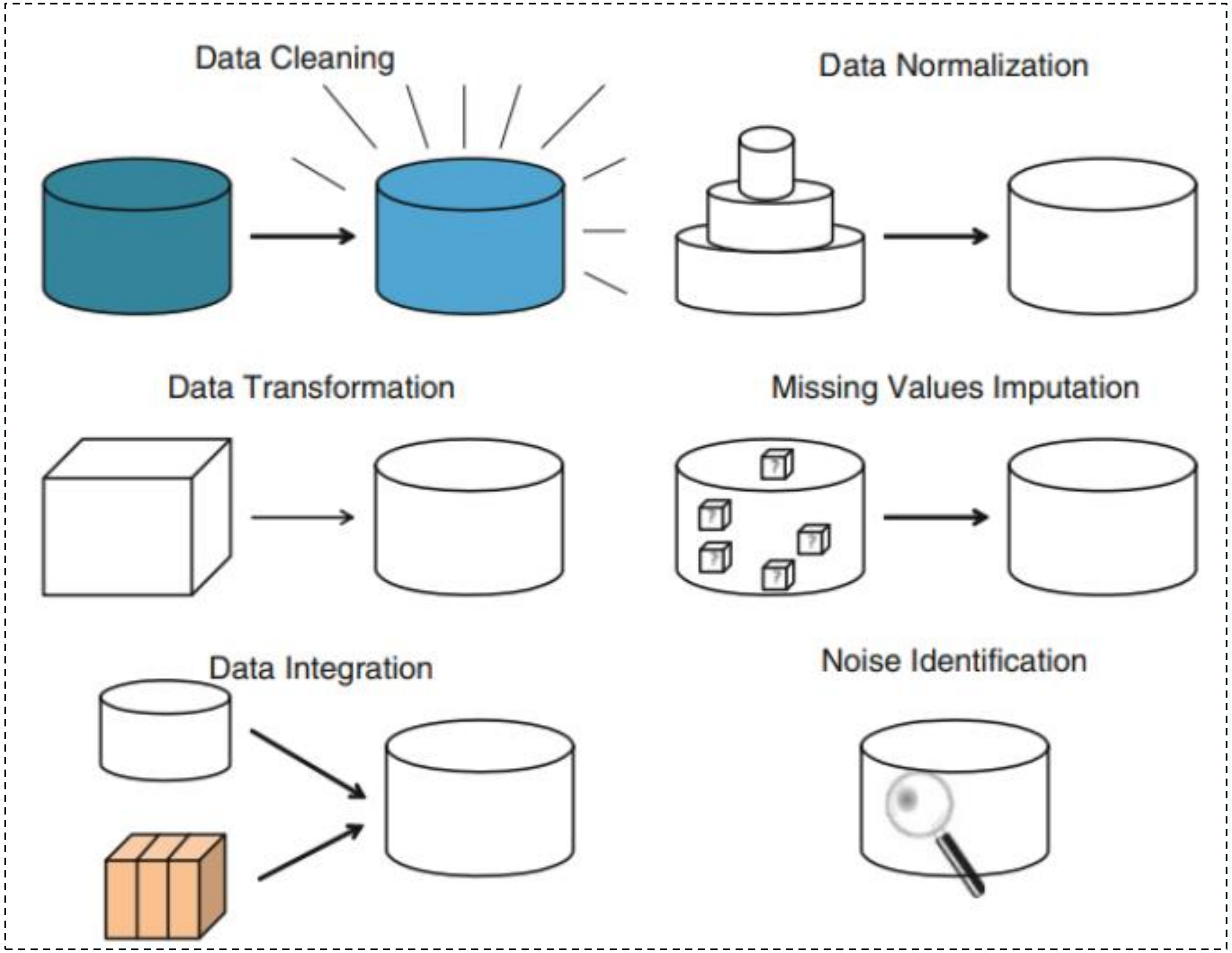


Feature selection



Feature Preprocessing

Feature Preprocessing Tasks



Features Transformation

Numeric Feature => Binary Feature

Length of text + [40] => { 0, 1 }

Single threshold

Numeric Feature => Categorical Feature

Length of text + [20, 40] => { short or medium or long }

Set of thresholds

Categorical Feature => Binary Features

{ short or medium or long } => [1, 0, 0] or [0, 1, 0] or [0, 0, 1]

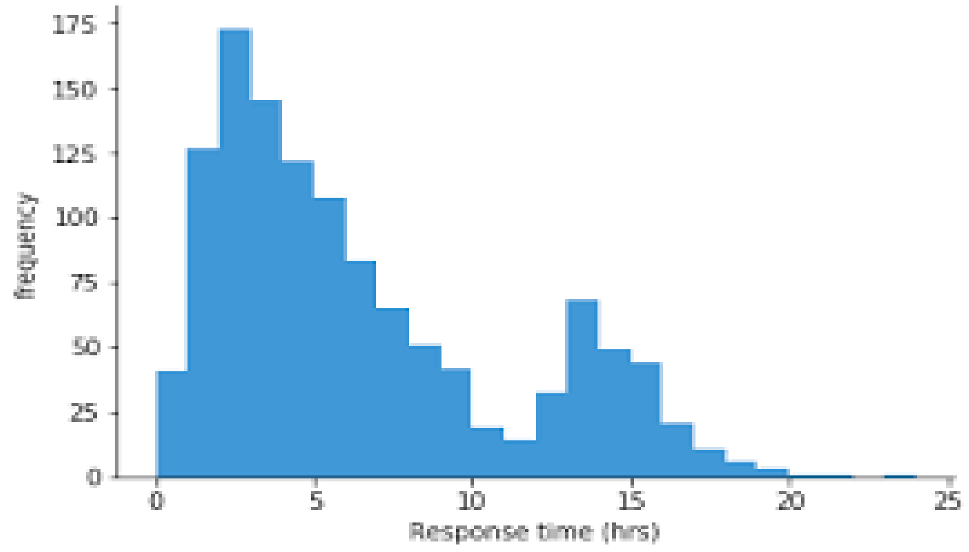
One-hot encoding

Binary Feature => Numeric Feature

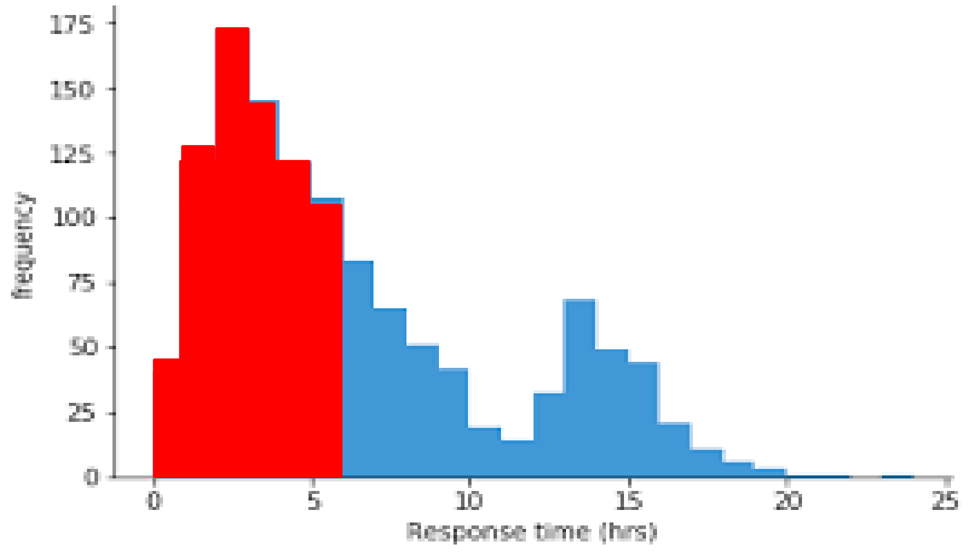
{ 0, 1 } => { 0, 1 }

…

Which threshold is better?



Unsupervised

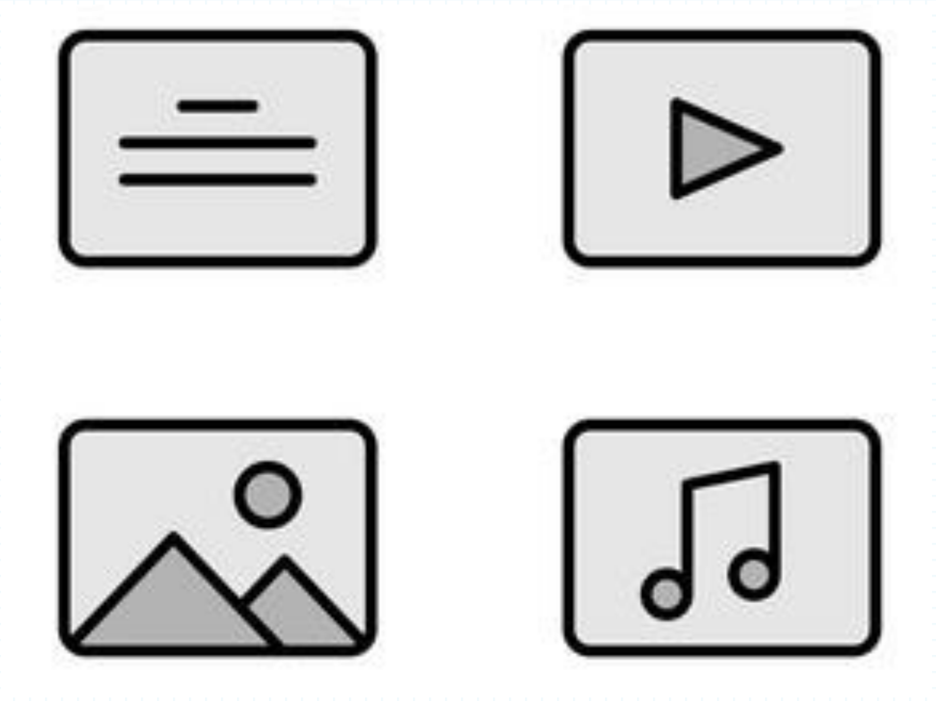


Supervised

Feature Extraction

Feature Extraction

- Feature extraction is a process in which you take raw data, often in the form of complex and high-dimensional variables, and transform it into a reduced and more manageable set of features.



Feature Extraction: SMS Spam

- SMS Message (arbitrary text) -> 5 dimensional array of binary features
  - 1 if message is longer than 40 chars, 0 otherwise
  - 1 if message contains a digit, 0 otherwise
  - 1 if message contains word ‘call’, 0 otherwise
  - 1 if message contains word ‘to’, 0 otherwise
  - 1 if message contains word ‘your’, 0 otherwise

“SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info”

Long?	HasDigit?	ContainsWord(Call)	ContainsWord(to)	ContainsWord(your)

Possible Features

Binary Features

- ContainsWord(call)?
- IsLongSMSMessage?
- Contains(*#)?
- ContainsPunctuation?

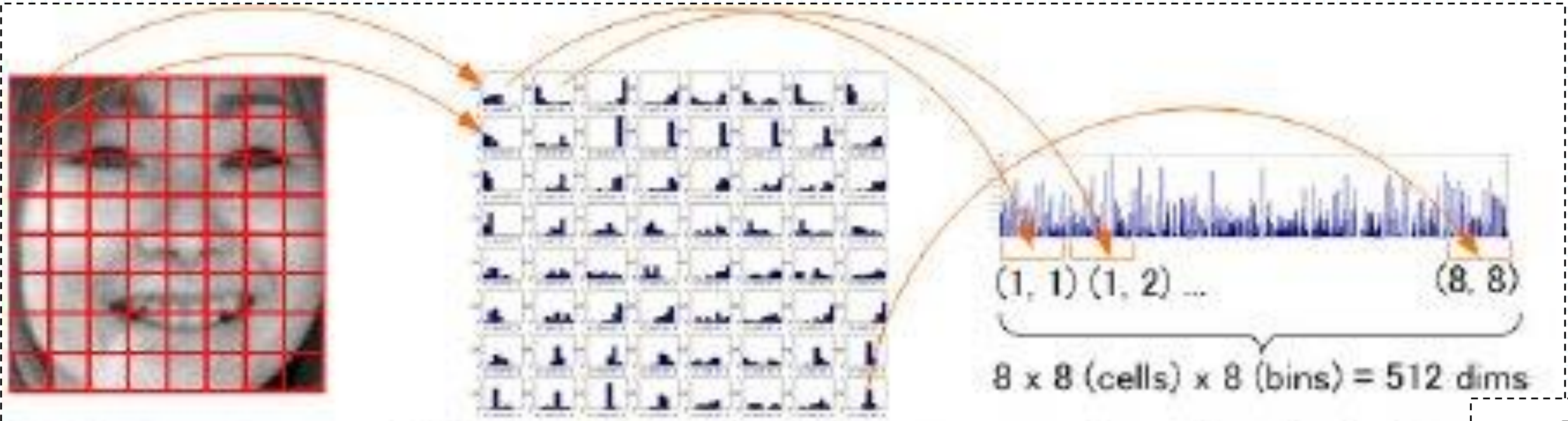
Categorical Features

- FirstWordPOS -> { Verb, Noun, Other }
- MessageLength -> { Short, Medium, Long, VeryLong }
- TokenType -> { Number, URL, Word, Phone#, Unknown }
- GrammarAnalysis -> { Fragment, SimpleSentence, ComplexSentence }

Numeric Features

- CountOfWord(call)
- MessageLength
- FirstNumberInMessage
- WritingGradeLevel

Feature Engineering: Smile Detection



Divide face image into 8x8 cells

Calc. local intensity histogram with 8 bins for each cell as a local feature

Concatenate local features

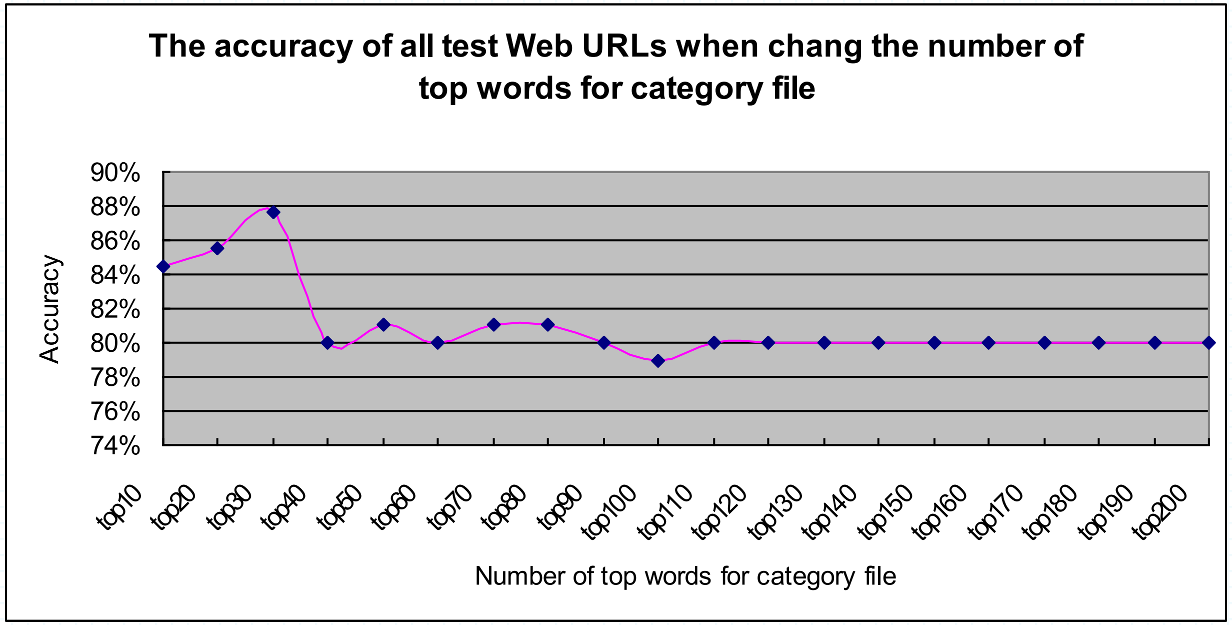
512 numerical features

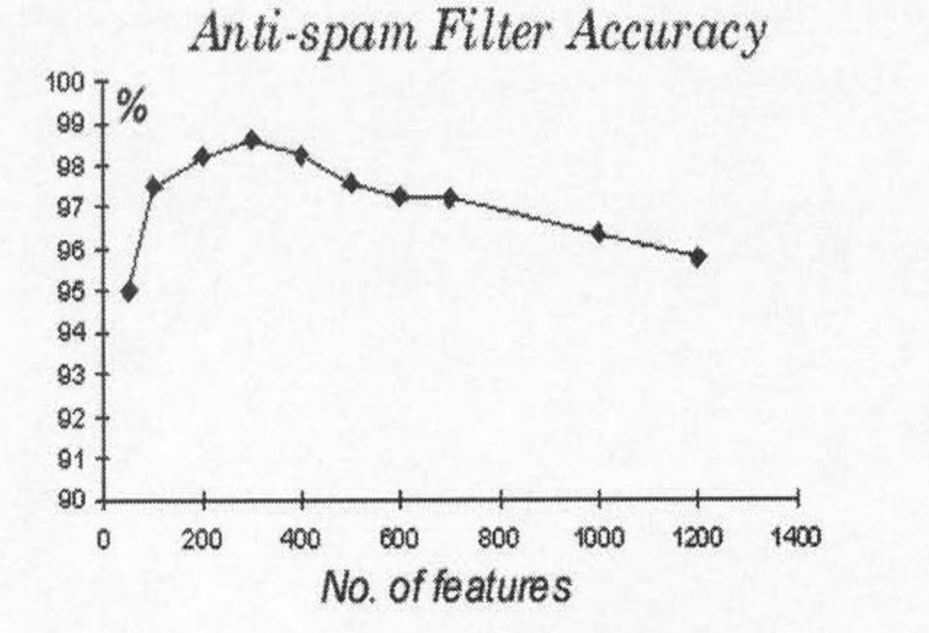
Feature Selection

Feature Selection: What

- You have a data has 100,000 fields (features)
  - Examples?
- You want to use it to build a classifier, so that you can predict something
  - What are the possible problems?
- you need to cut it down to 1,000 fields before you try machine learning. Which 1,000?
  - How to do that => Feature Selection

Feature Selection: Why





Why accuracy reduces

Why accuracy reduces

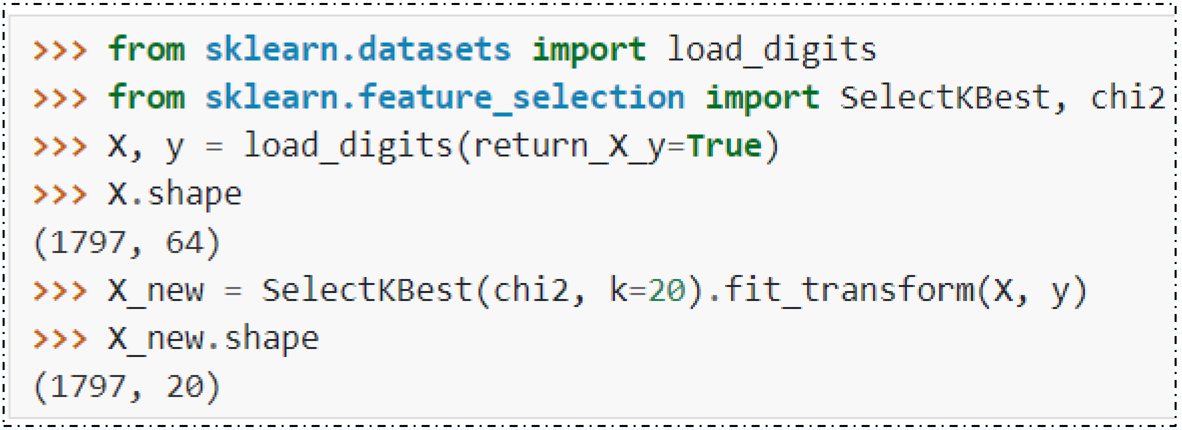
- **Noise:** The additional features typically **add noise**. Machine learning will pick up on **fake correlations**, that might be true in the training set, but not in the test set (**overfitting**).
  - Example: what will happen if you learn ID3 with too many noisy data?

- **Explosion:** For some ML methods, more features means **more parameters to learn** (more NN weights, more decision tree nodes, etc…) – the increased space of possibilities is **more difficult to search**.

Univariate feature selection

- Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator.
  - Example: In scikit, *SelectKBest* removes all but the h highest scoring features based on a scoring function.

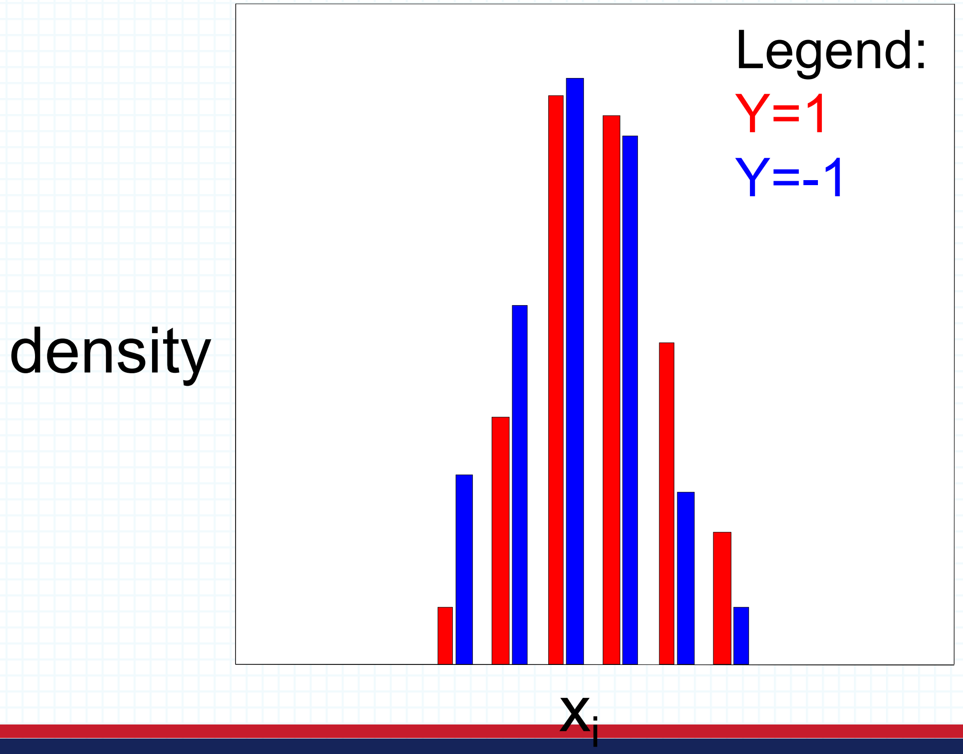
- Methods are used to rank features by importance
  - Pearson correlation coefficient
  - F-score
  - Chi-square
  - Signal to noise ratio
  - And more such as mutual information,

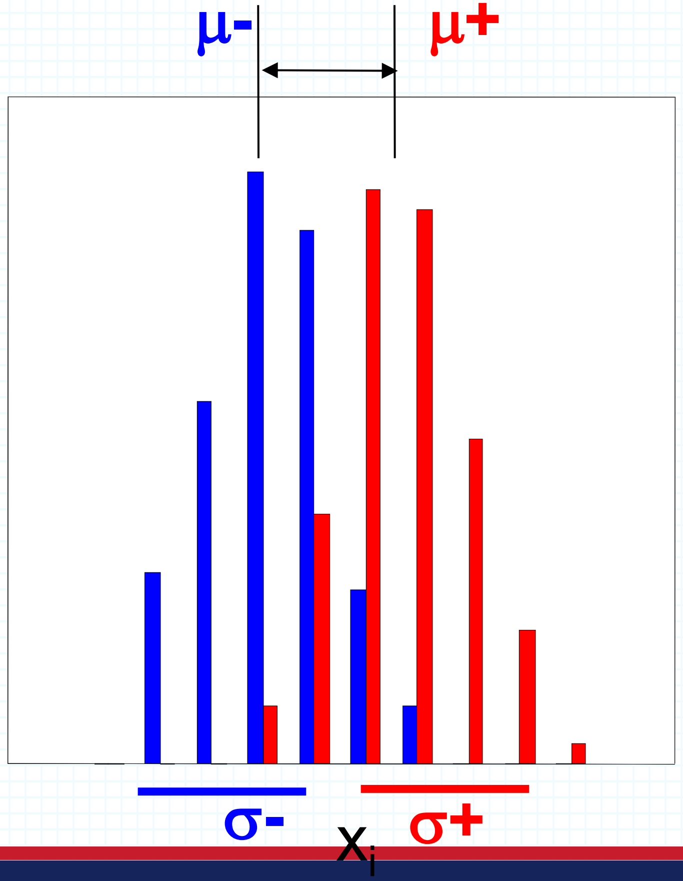


```
>>> from sklearn.datasets import load_digits
>>> from sklearn.feature_selection import SelectKBest, chi2
>>> X, y = load_digits(return_X_y=True)
>>> X.shape
(1797, 64)
>>> X_new = SelectKBest(chi2, k=20).fit_transform(X, y)
>>> X_new.shape
(1797, 20)
```

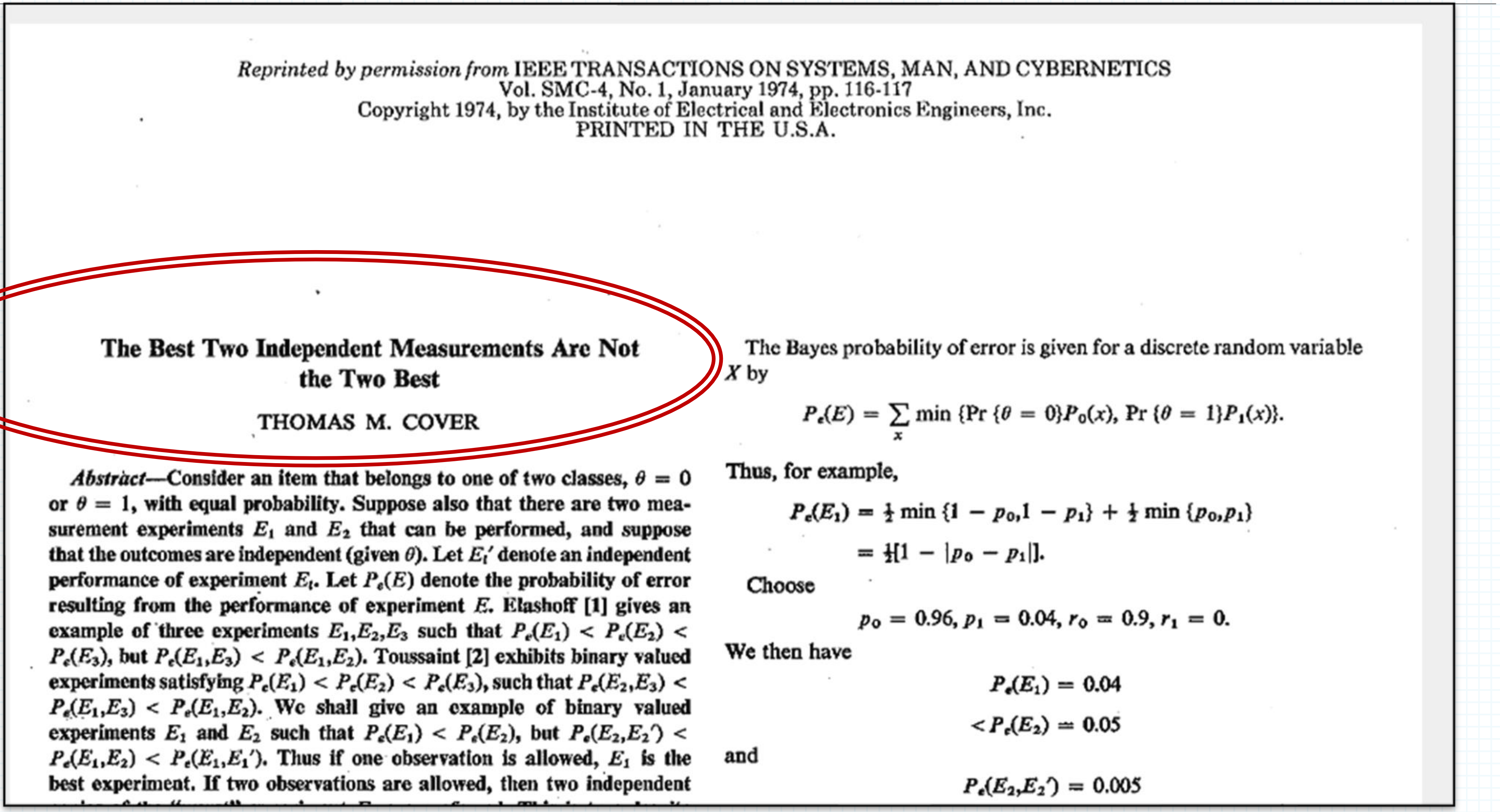
Univariate feature selection: Example

Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator.





But..



Reprinted by permission from IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS
Vol. SMC-4, No. 1, January 1974, pp. 116-117
Copyright 1974, by the Institute of Electrical and Electronics Engineers, Inc.
PRINTED IN THE U.S.A.

The Best Two Independent Measurements Are Not the Two Best

THOMAS M. COVER

Abstract—Consider an item that belongs to one of two classes, $\theta = 0$ or $\theta = 1$, with equal probability. Suppose also that there are two measurement experiments E_1 and E_2 that can be performed, and suppose that the outcomes are independent (given θ). Let E_i' denote an independent performance of experiment E_i. Let $P_e(E)$ denote the probability of error resulting from the performance of experiment E. Elashoff [1] gives an example of three experiments E_1,E_2,E_3 such that $P_e(E_1) < P_e(E_2) < P_e(E_3)$, but $P_e(E_1,E_3) < P_e(E_1,E_2)$. Toussaint [2] exhibits binary valued experiments satisfying $P_e(E_1) < P_e(E_2) < P_e(E_3)$, such that $P_e(E_2,E_3) < P_e(E_1,E_3) < P_e(E_1,E_2)$. We shall give an example of binary valued experiments E_1 and E_2 such that $P_e(E_1) < P_e(E_2)$, but $P_e(E_2,E_2') < P_e(E_1,E_2) < P_e(E_1,E_1')$. Thus if one observation is allowed, E_1 is the best experiment. If two observations are allowed, then two independent

The Bayes probability of error is given for a discrete random variable X by

$$P_e(E) = \sum_x \min\{\Pr\{\theta = 0\}P_0(x), \Pr\{\theta = 1\}P_1(x)\}.$$

Thus, for example,

$$P_e(E_1) = \tfrac{1}{2}\min\{1 - p_0, 1 - p_1\} + \tfrac{1}{2}\min\{p_0, p_1\}$$
$$= \tfrac{1}{2}[1 - |p_0 - p_1|].$$

Choose

$$p_0 = 0.96,\ p_1 = 0.04,\ r_0 = 0.9,\ r_1 = 0.$$

We then have

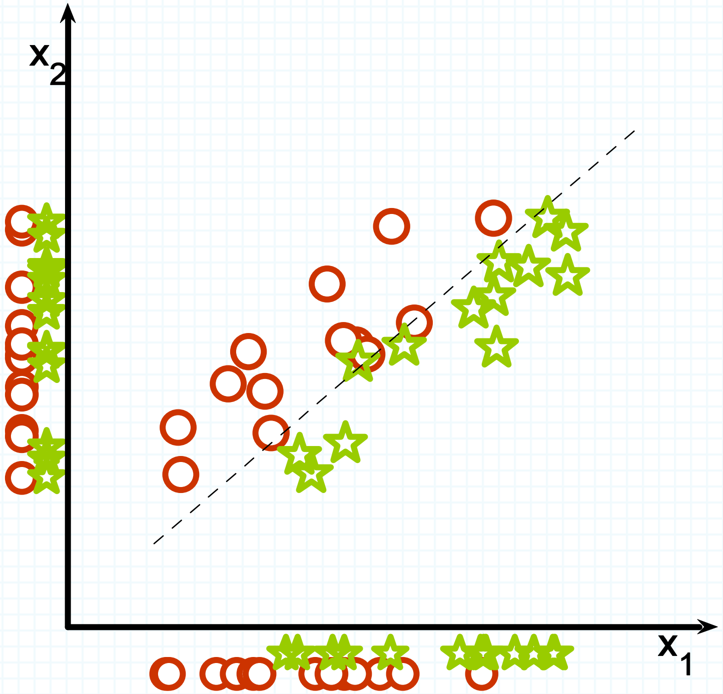
$$P_e(E_1) = 0.04$$
$$< P_e(E_2) = 0.05$$

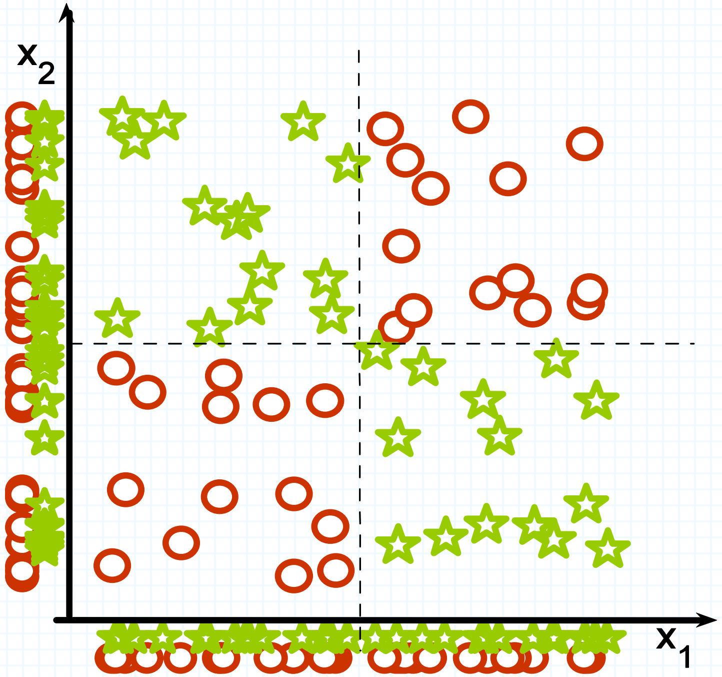
and

$$P_e(E_2,E_2') = 0.005$$

Univariate feature selection

- Look at the projection onto each axis.
- Univariate feature selection could throw away x1 and X2 in both cases.
- X2 alone is irrelevant but together with x1 is good.





Example

- Correlation-based feature ranking?
  - It is actually fine for certain datasets.
  - But bad for many cases
    - WHY?
  - Example:

f1	f2	f3	f4	…	class
0.4	0.6	0.4	0.6		1
0.2	0.4	1.6	-0.6		1
0.5	0.7	1.8	-0.8		1
0.7	0.8	0.2	0.9		2
0.9	0.8	1.8	-0.7		2
0.5	0.5	0.6	0.5		2

Example

f1	f2	f3	f4	…	class
0.4	0.6	0.4	0.6		1
0.2	0.4	1.6	-0.6		1
0.5	0.7	1.8	-0.8		1
0.7	0.8	0.2	0.9		2
0.9	0.8	1.8	-0.7		2
0.5	0.5	0.6	0.5		2

Correlated with the class

Example

f1	f2	f3	f4	…	class
0.4	0.6	0.4	0.6		1
0.2	0.4	1.6	-0.6		1
0.5	0.7	1.8	-0.8		1
0.7	0.8	0.2	0.9		2
0.9	0.8	1.8	-0.7		2
0.5	0.5	0.6	0.5		2

uncorrelated with the class

(Noise?)

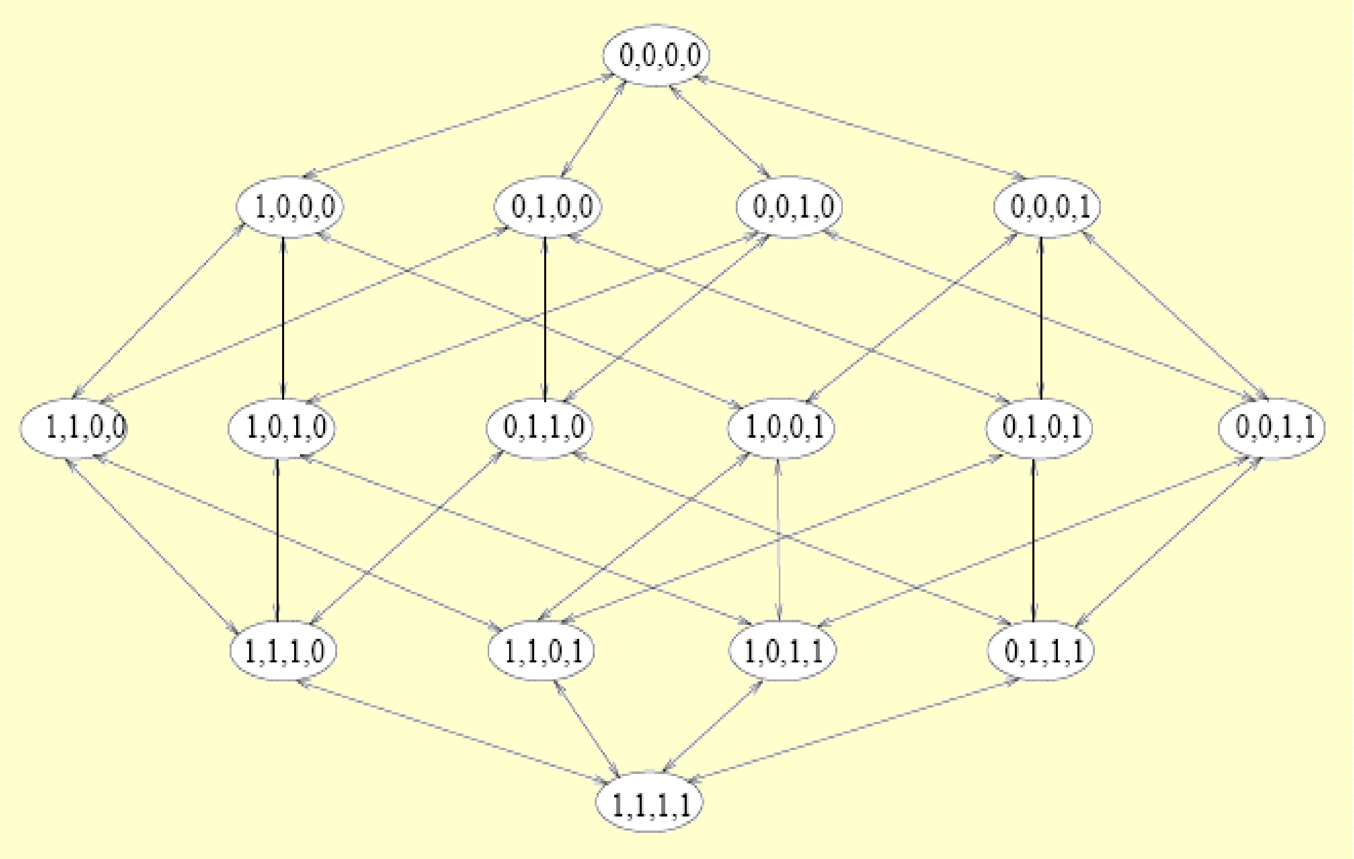
Example

f1	f2	f3	f4	…	class
0.4	0.6	0.4	0.6	1	1
0.2	0.4	1.6	-0.6	1	1
0.5	0.7	1.8	-0.8	1	1
0.7	0.8	0.2	0.9	1.1	2
0.9	0.8	1.8	-0.7	1.1	2
0.5	0.5	0.6	0.5	1.1	2

But, col 5 shows us f3 + f4 – which is perfectly correlated with the class!

Multivariate feature selection

- Multivariate feature selection implies a search in the space of all possible combinations of features.
- For n features, there are 2^n possible subsets of features.
- This yields both to a high computational and statistical complexity.

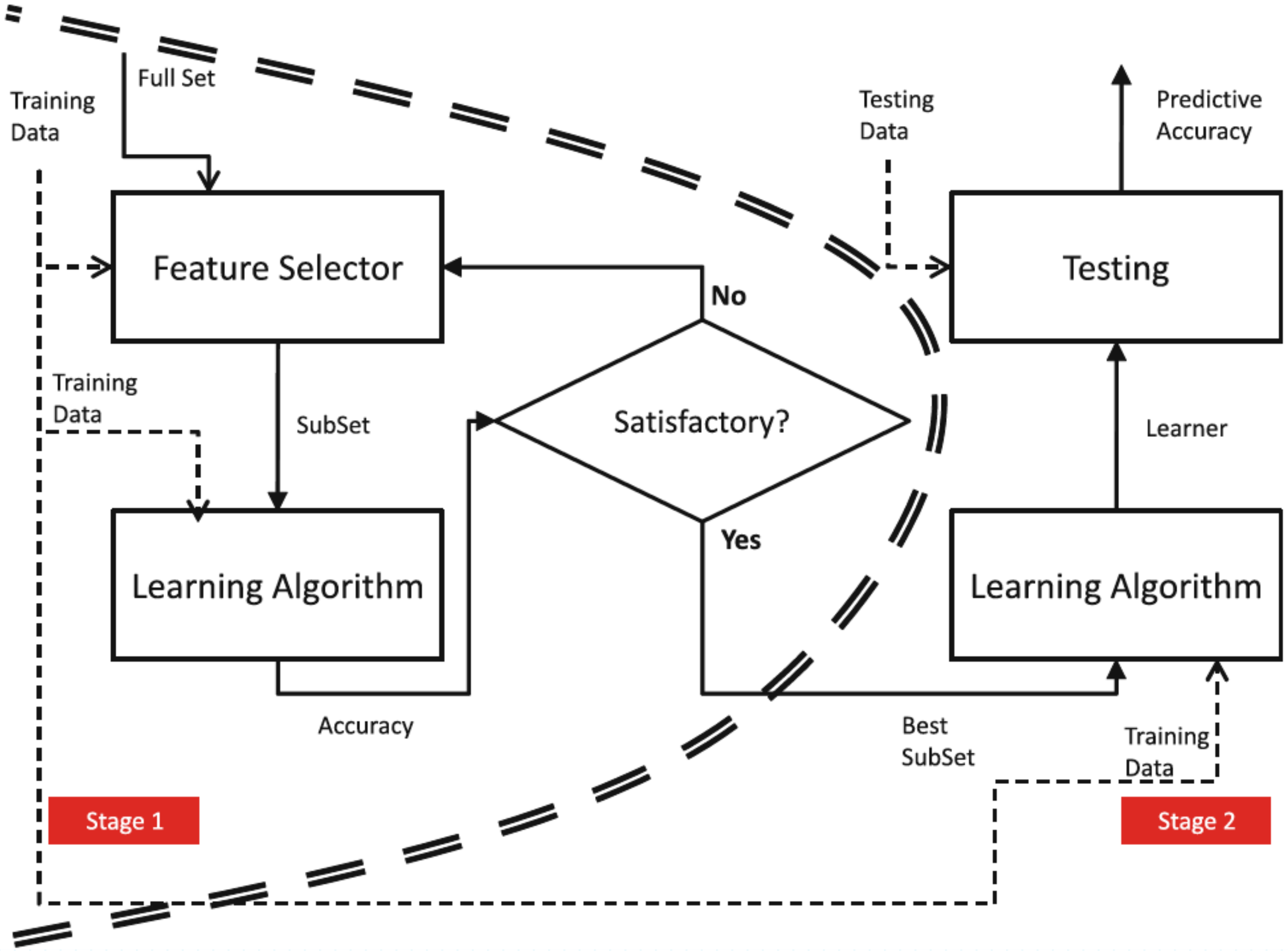


Multivariate feature selection

- How to search the space of all possible variable subsets ?
  - A wide range of heuristic search strategies can be used.
    Two different classes:
    - Forward selection
      (start with empty feature set and add features at each step)
    - Backward elimination
      (start with full feature set and discard features at each step)
- How can we evaluate each subset?

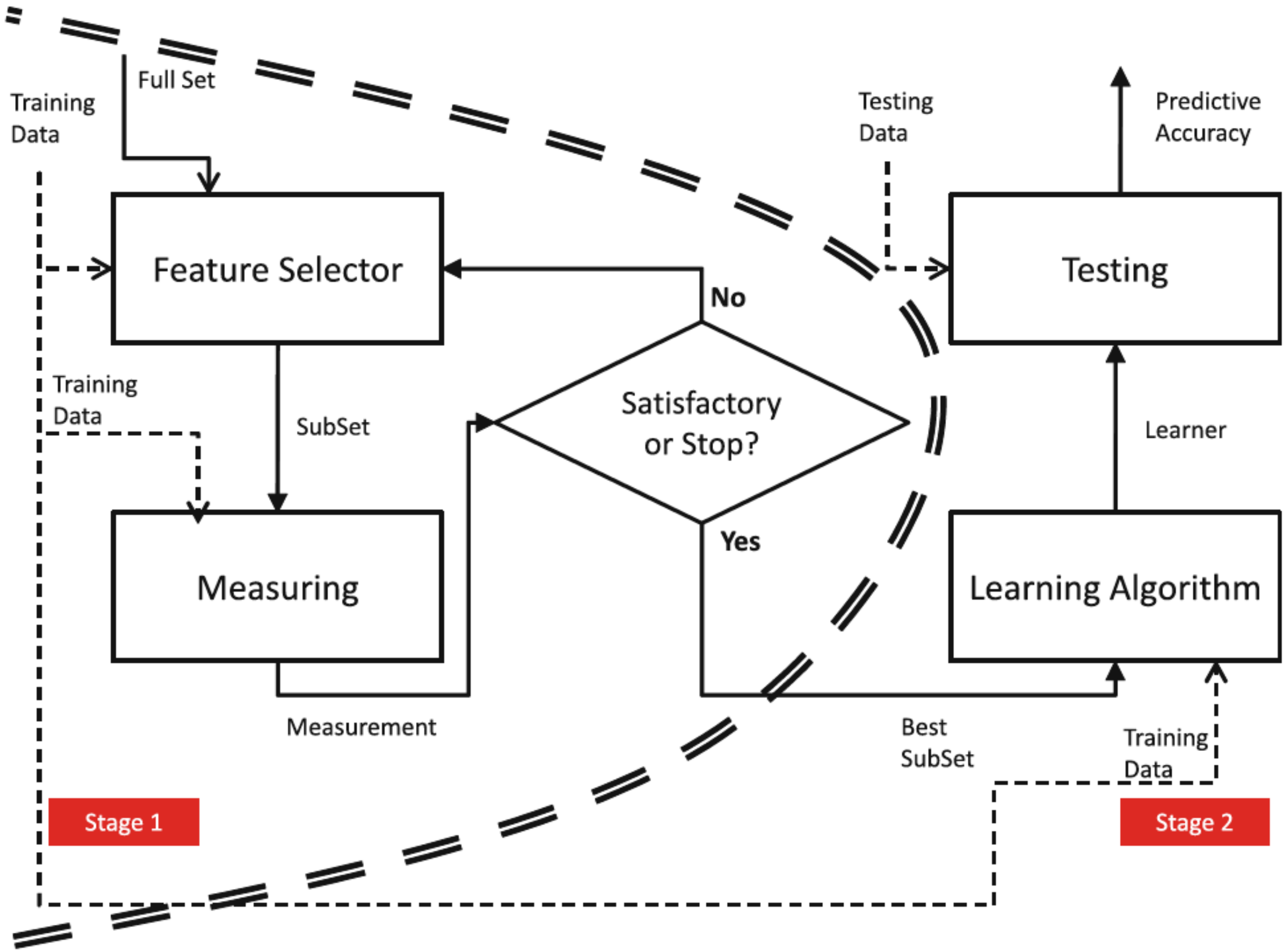
Wrapper Methods

- A Learner is used to score subsets of features according to the predictive power of the learner when using the subsets.
- Results vary for different learners.



Filter Methods

- Filters function analogously to wrappers, but they use in the evaluation function something cheaper to compute than the performance of the target learning machine (e.g. a correlation coefficient or the performance of a naïve machine learning approach).
- Filtering method is much faster but it do not incorporate learning.



Training Data
Full Set
Feature Selector
Training Data
SubSet
Measuring
Measurement
Satisfactory or Stop?
No
Yes
Testing Data
Predictive Accuracy
Testing
Learner
Learning Algorithm
Best SubSet
Training Data
Stage 1
Stage 2