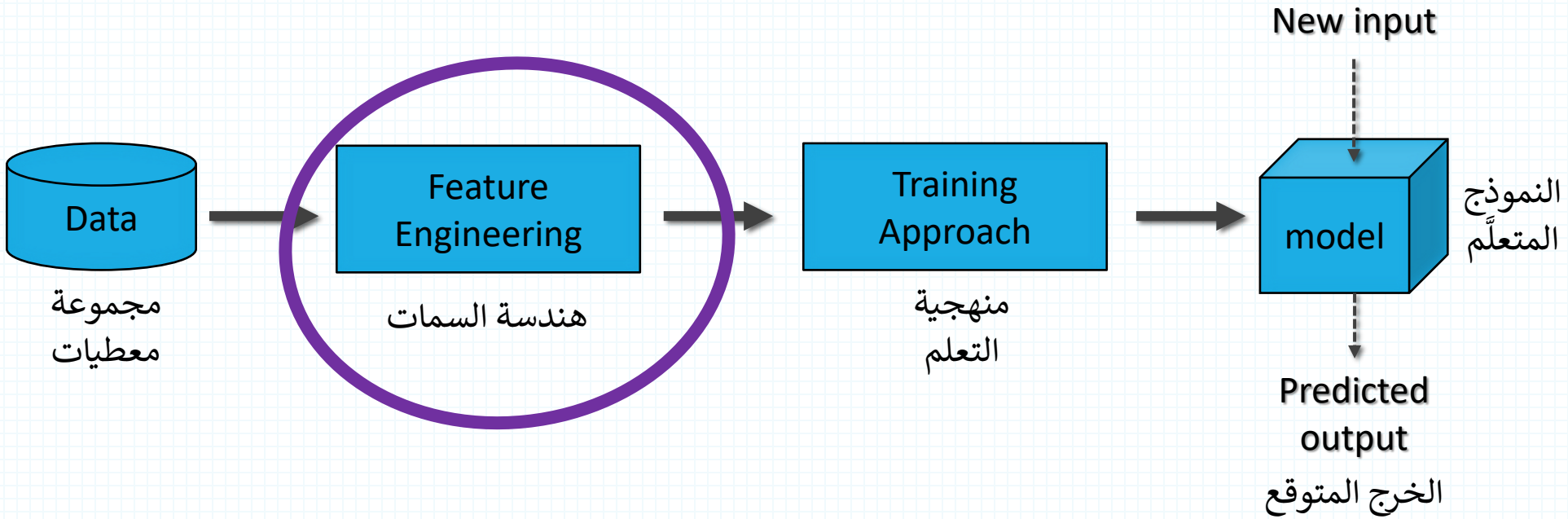
مقرر التعلم التلقائي | السنة الخامسة – هندسة المعلوماتية / الذكاء الصنعي | Week 4

Feature Engineering

د. رياض سنبل

Access Course Materials

Traditional ML Pipeline



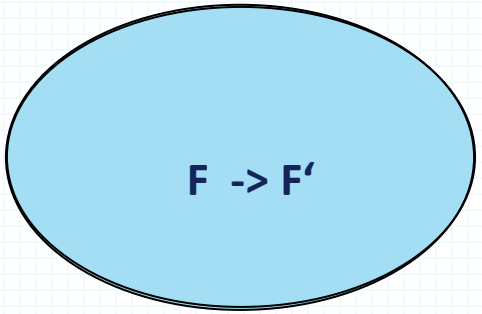
Feature Engineering

- **Feature Engineering** refers to manipulation — addition, deletion, combination, mutation — of your data set to improve machine learning model training, leading to better performance and greater accuracy.
- It includes:
  - Clean, normalize, transform features.
  - Convert ‘context’ -> input to learning algorithm (a representation).
  - Balance number of features, complexity of concept, complexity of model, amount of data.

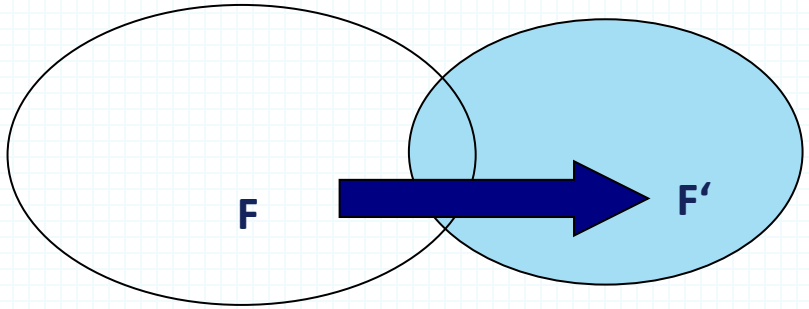
Feature (Preprocessing vs Selection vs Extraction)

- **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

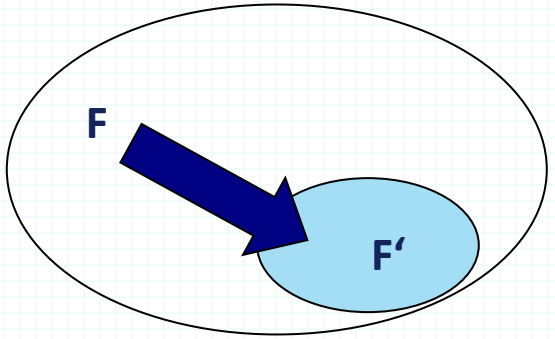
Feature Preprocessing



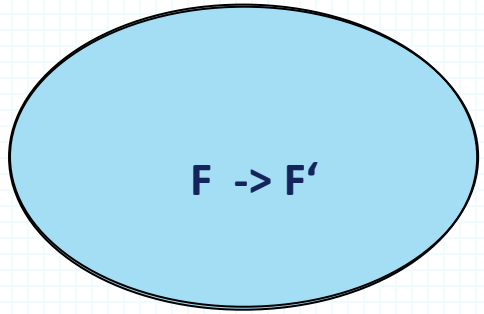
Feature extraction



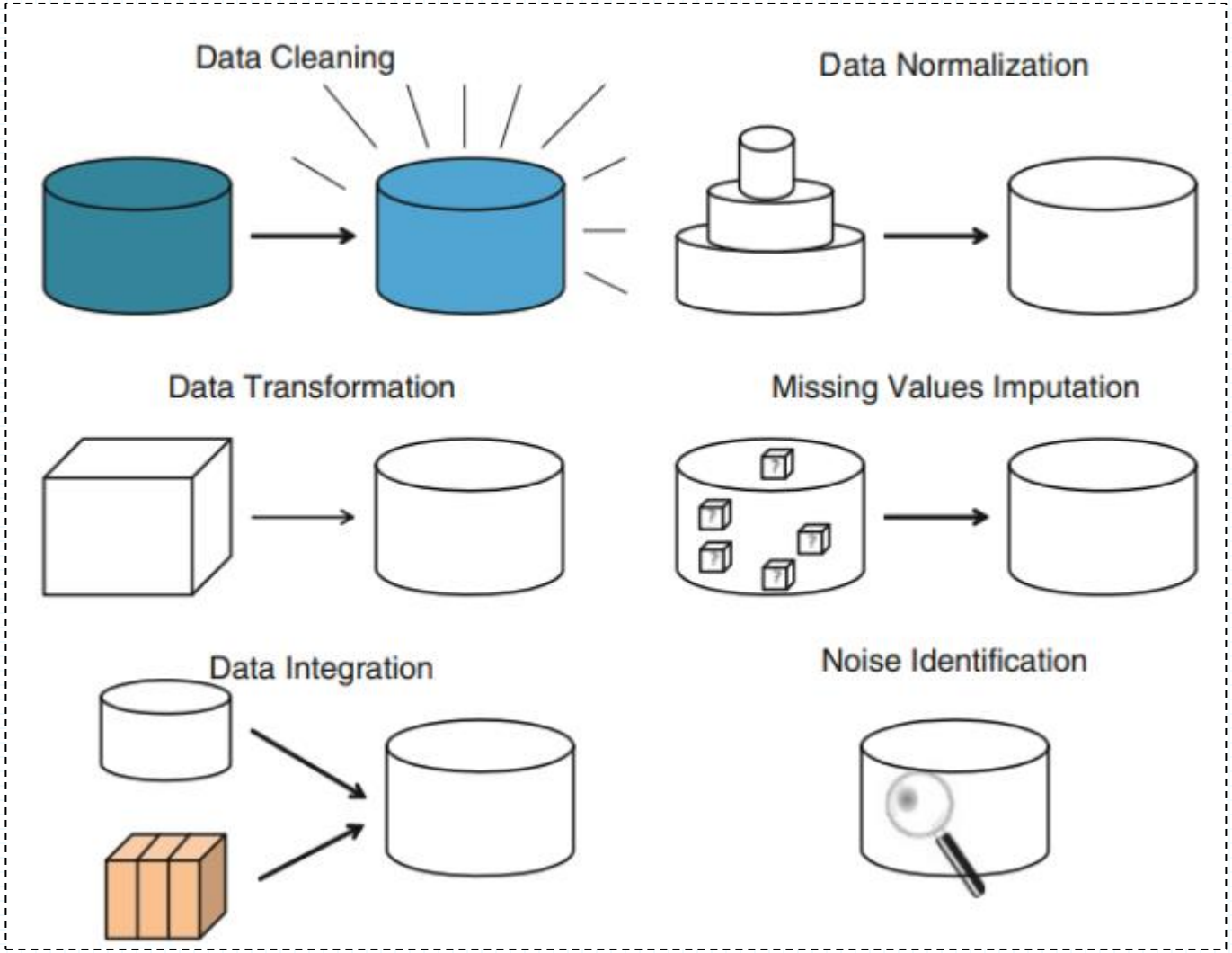
Feature selection



Feature Preprocessing



Feature Preprocessing Tasks



Features Cleaning

- Features cleaning refers to the process of **Identifying and handling:**
  - Inconsistencies,
  - Errors,
  - outliers

  in the input features of a dataset.
- This may involve removing or correcting invalid or inaccurate data points to ensure the quality of the features.

Scenario: *You're working with a dataset of housing prices, and you notice some entries with unrealistic values in the "number of bedrooms" feature, such as houses with 100 bedrooms.*

Features Transformation

- Features transformation involves **modifying the existing features or creating new features** to enhance the performance of a machine learning model.
- This may include operations like scaling, encoding categorical variables, or applying mathematical transformations to make the features more suitable for a particular algorithm.

Scenario: *You're working with a dataset for predicting the energy consumption of buildings. One of the features is "timestamp," representing the time and date when energy readings were recorded. However, the model you're using doesn't effectively capture the temporal patterns with just the raw timestamp..*
Convert it to DAY!

Features Transformation

Numeric Feature => Binary Feature — Single threshold

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

Numeric Feature => Categorical Feature — Set of thresholds

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

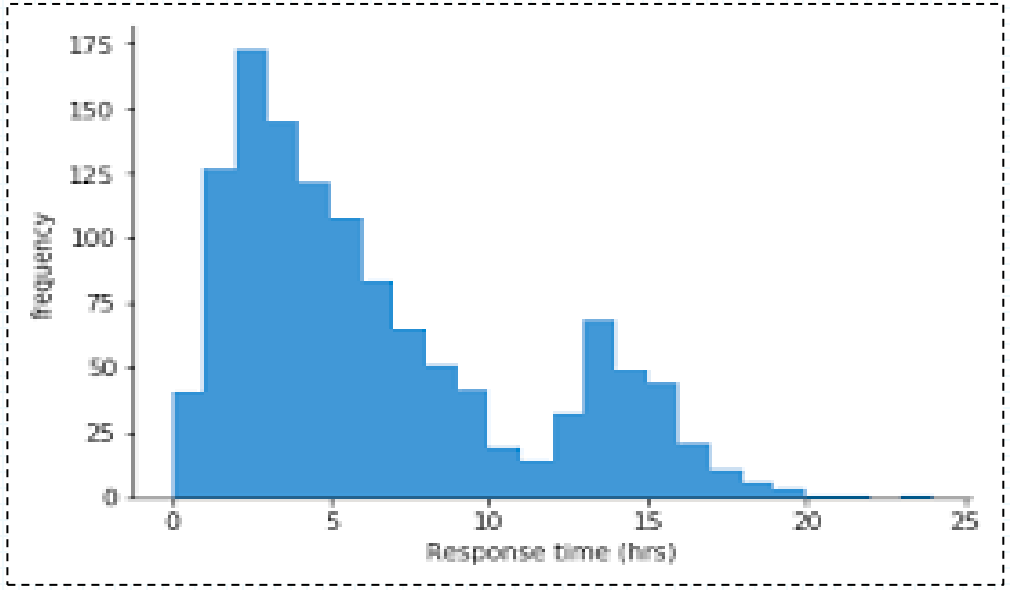
Categorical Feature => Binary Features — One-hot encoding

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

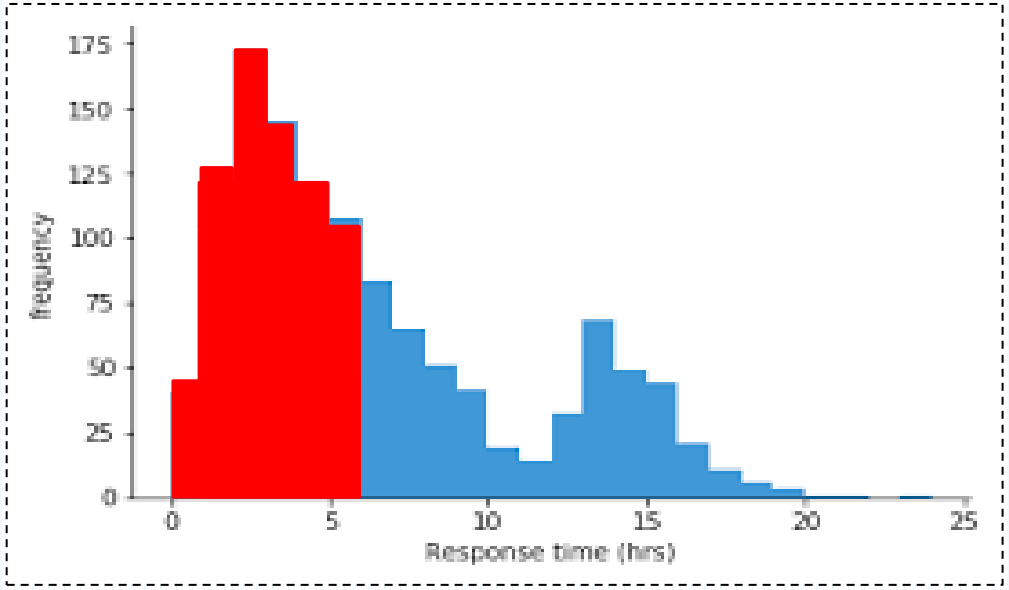
Binary Feature => Numeric Feature — …

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

Which threshold is better?



Unsupervised



Supervised

Features Integration

- Features integration involves **combining multiple features into a single feature** or creating new features by merging information from existing ones.
- This can be useful when certain features individually may not provide enough information, but their combination might reveal meaningful patterns for the machine learning model.

Scenario: *In a social media dataset, you have separate features for the number of "likes" and "shares" for each post. You believe that the combined engagement, i.e., the total interaction, might be a more meaningful feature.*

Normalization

- Normalization is the process of **scaling numerical features** to a standard range, usually between 0 and 1 or -1 and 1.
- This is done to ensure that all features contribute equally to the model training, especially when features have different scales. **Common normalization techniques** include Min-Max scaling and Z-score normalization.

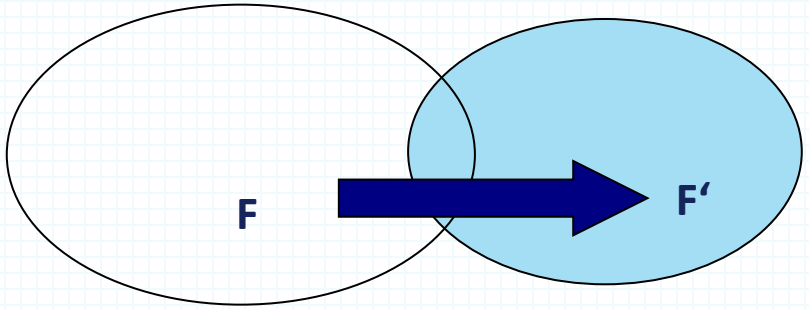
***Scenario**: You are working on a dataset containing both the salary and age of employees. The salary values are in the range of thousands, while age is in the range of tens.*

Missing Values Imputation

- Missing values imputation involves **filling in or estimating missing values** in a dataset.
- Machine learning algorithms often require complete datasets, so dealing with missing values is crucial. Techniques for imputation include using the mean, median, or mode of a feature, or more sophisticated methods like regression imputation.

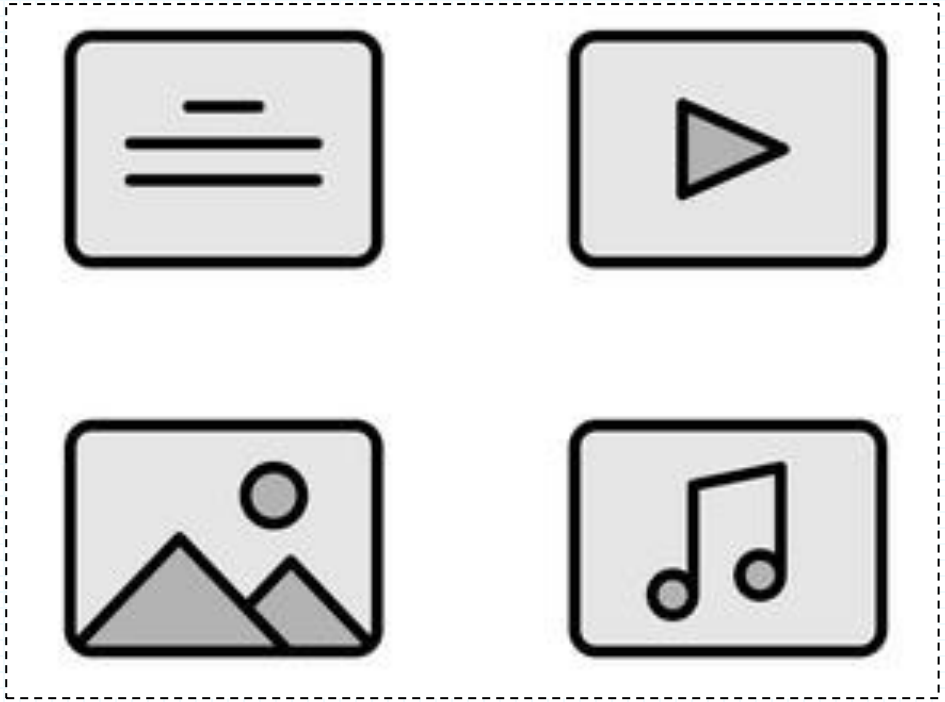
Action: *Instead of removing rows with missing BMI values, you impute the missing values by calculating the mean BMI of the dataset and filling in the gaps with this average value.*

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?

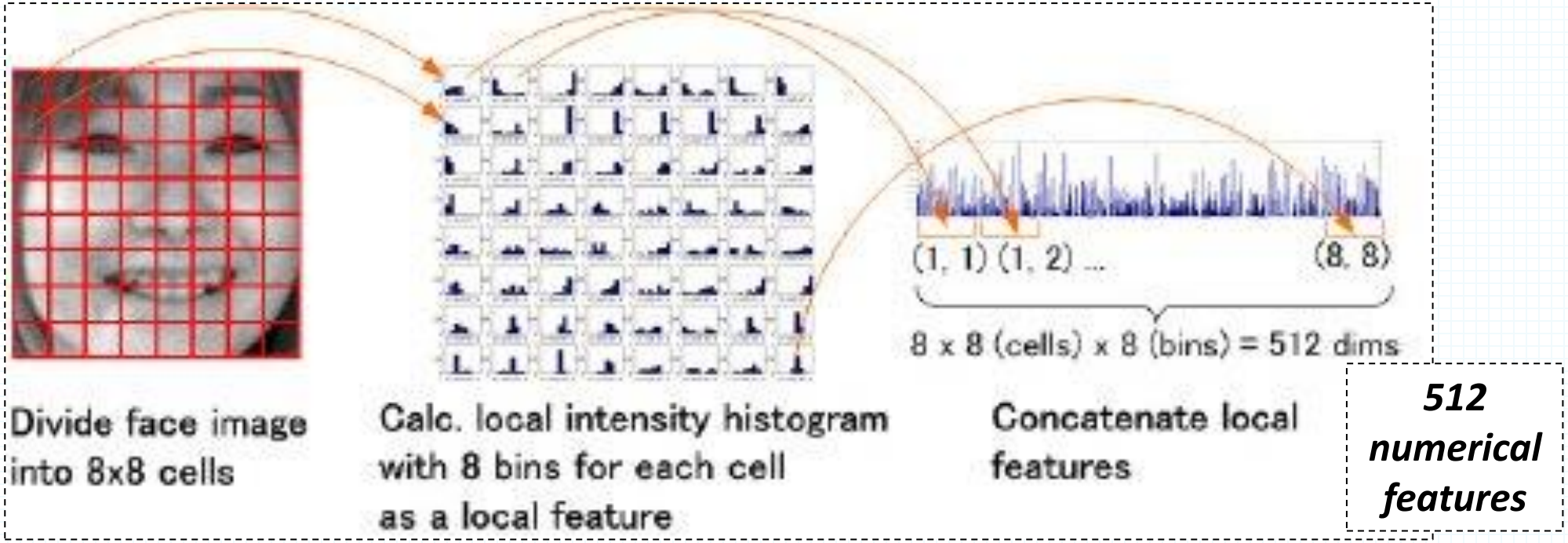
Categorical Features

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

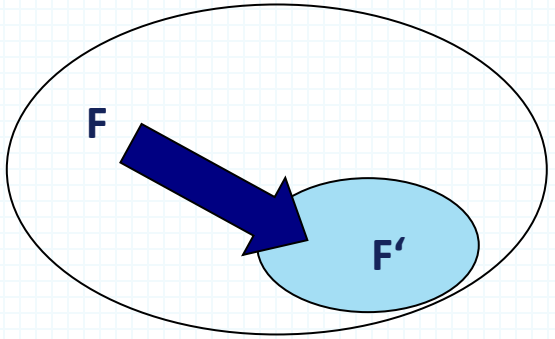
Numeric Features

- CountOfWord(call)
- MessageLength
- FirstNumberInMessage
- WritingGradeLevel

Feature Engineering: Smile Detection



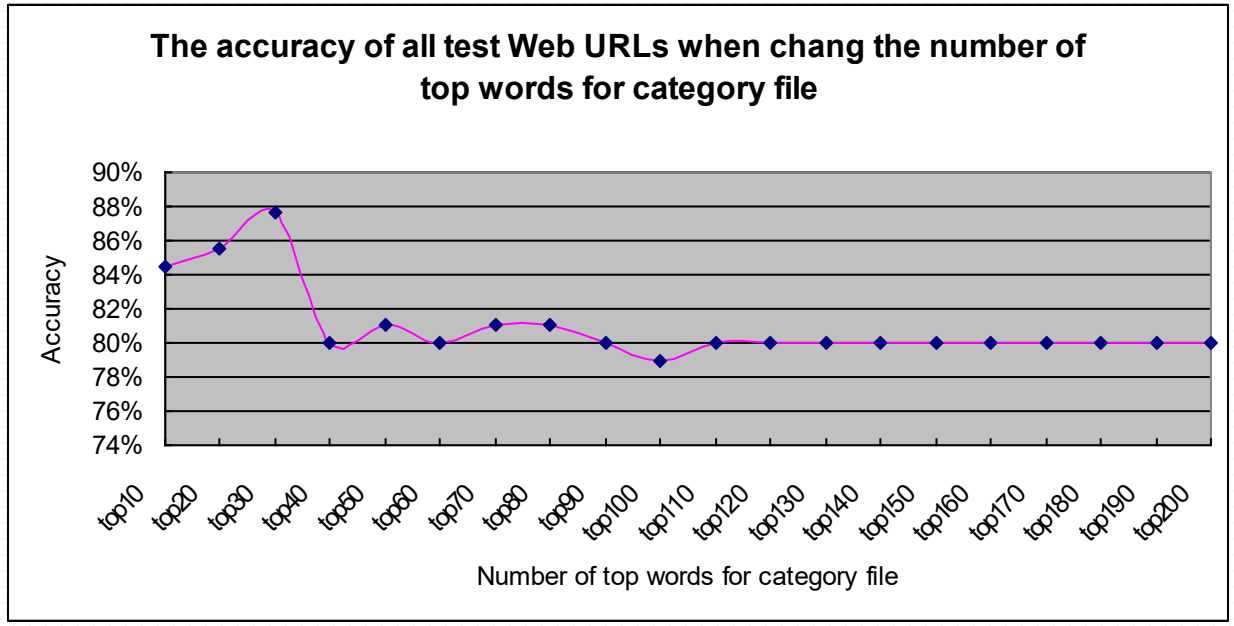
Feature Selection

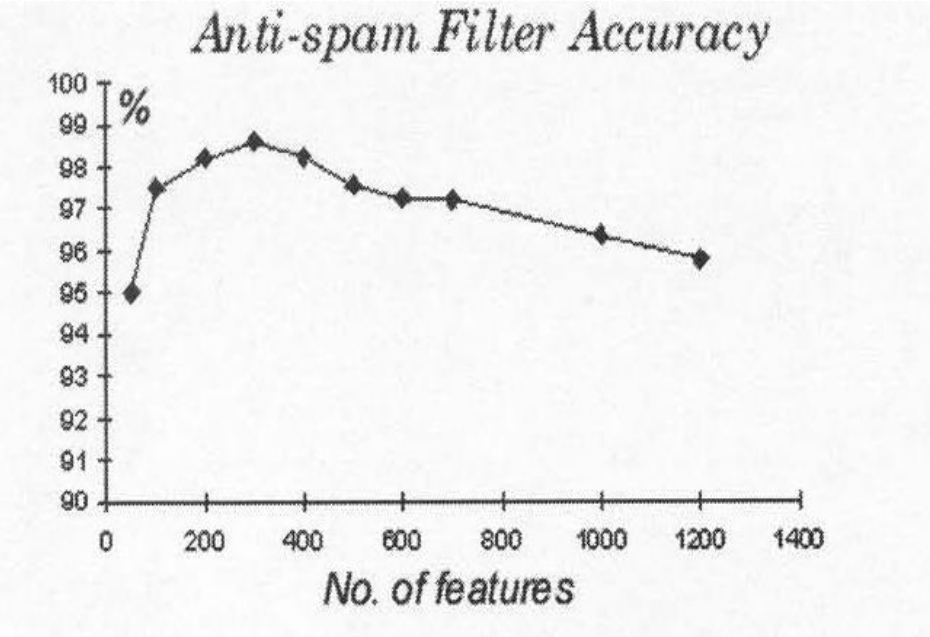


Feature Selection: What

- 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

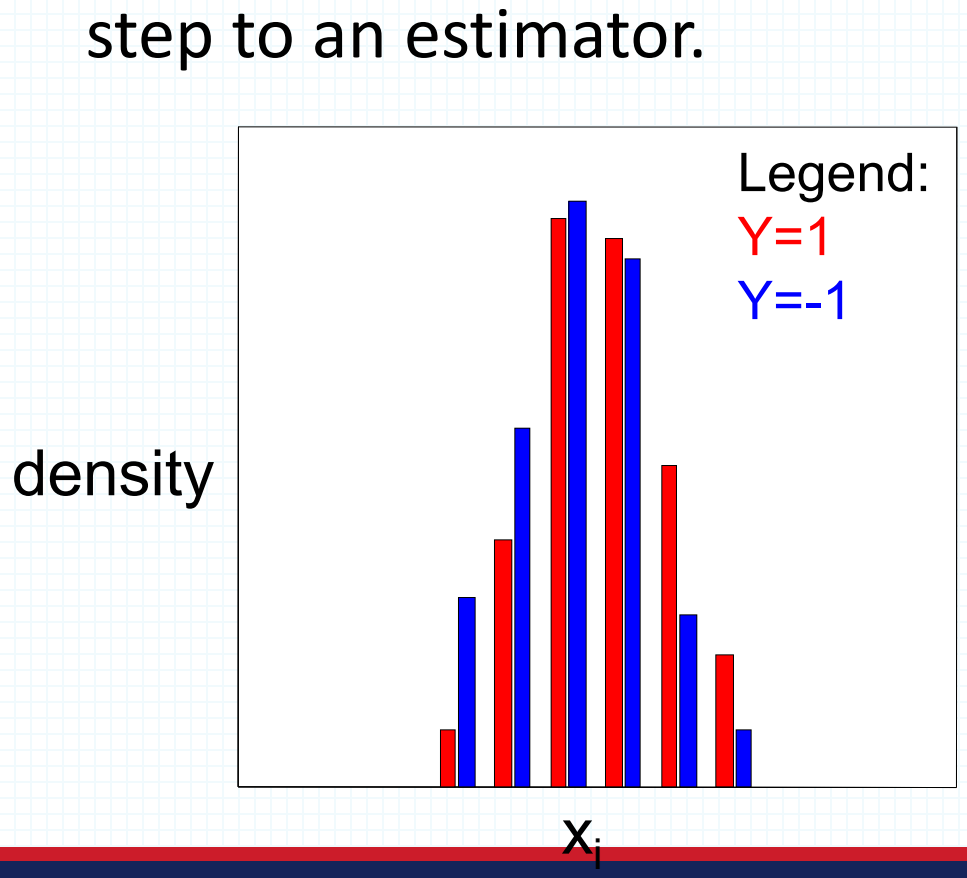
Why accuracy reduces

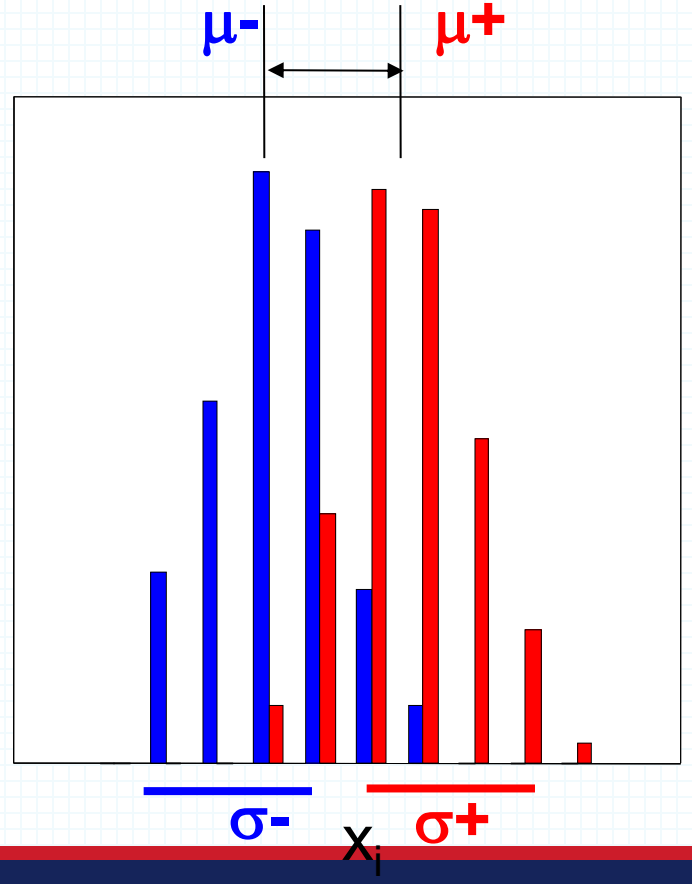
- **Noise:** The additional features typically **add noise**. Machine learning will pick up on **fake correlations**, that might be true in the training set, but not in the test set (**overfitting**).
  - Example: what will happen if you learn ID3 with too many noisy data?

- **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.





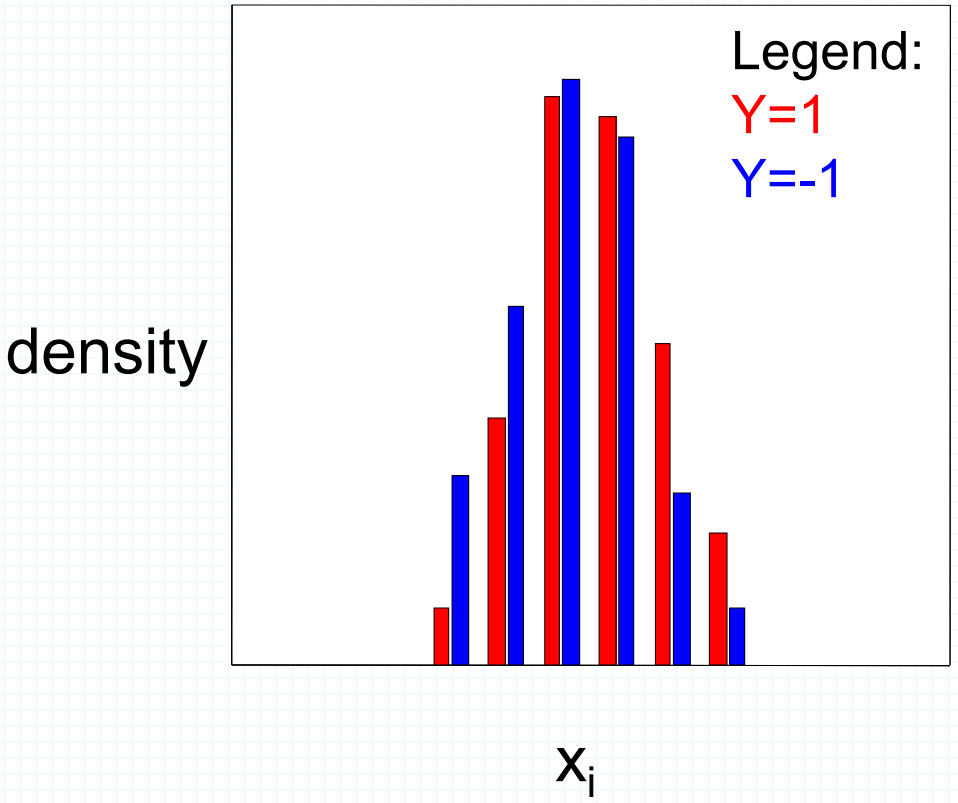
Univariate feature selection

- **Example: Irrelevant Feature**
- the probability of the variable given the target is equal to the prob of the variable. Which means we can ignore it.

$$P(X_i, Y) = P(X_i)\, P(Y)$$

$$P(X_i \mid Y) = P(X_i)$$

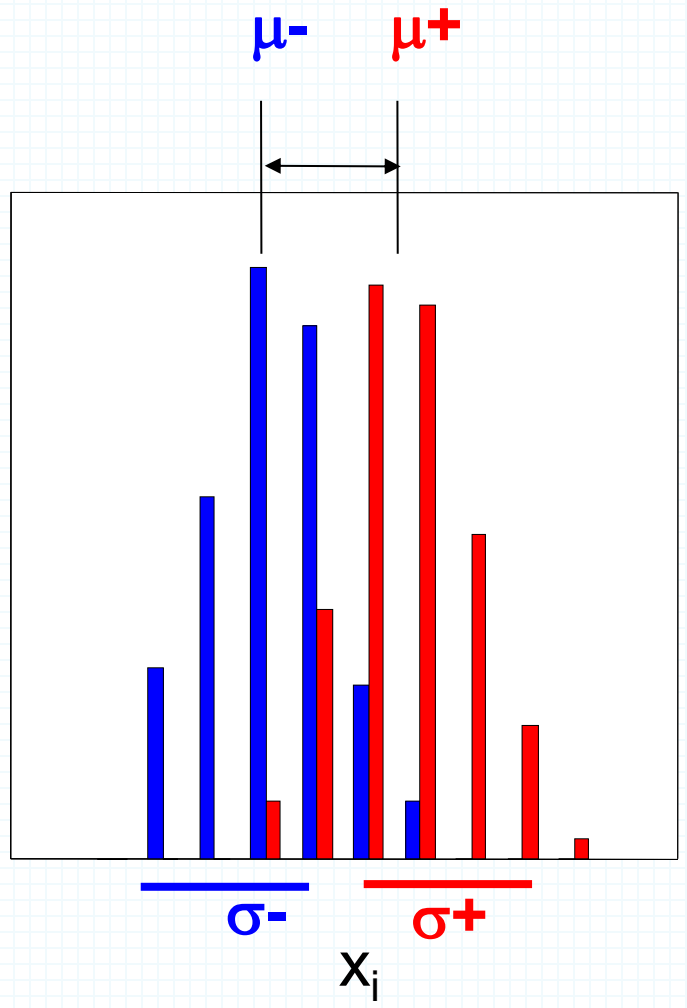
$$P(X_i \mid Y=1) = P(X_i \mid Y=-1)$$



Univariate feature selection

- How do we define an algorithm to determine which features are predictive?

- Simple idea: use the difference between the means (normalized to account for different ranges of course).



Feature selection methods

- 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

Correlation-based feature ranking

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

Correlation-based feature ranking

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?)

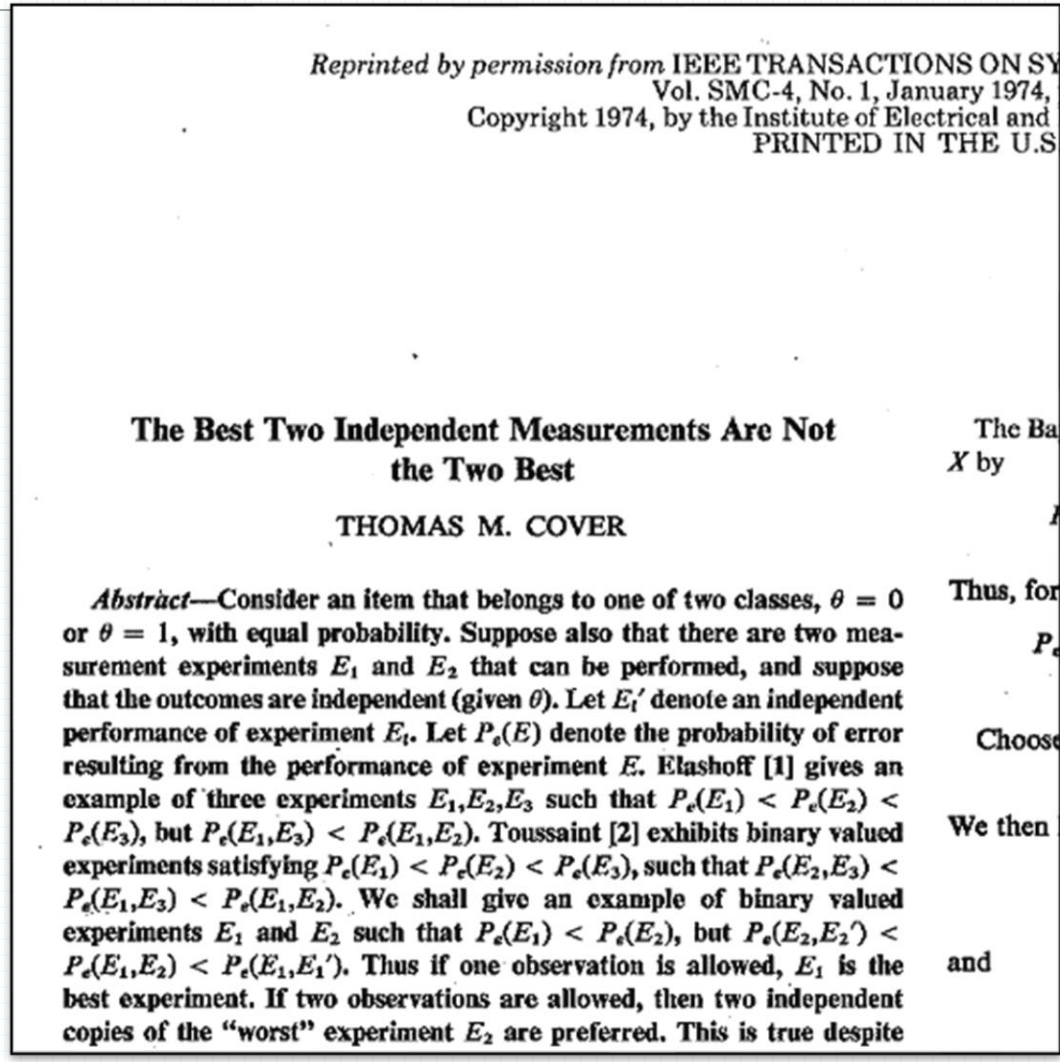
Correlation-based feature ranking

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!

Feature selection methods

- Good FS Methods therefore:
  - Need to consider how well features work **together**
  - As we have noted before, if you take 100 features that are each well correlated with the class, they may simply be correlated strongly with each other, so provide no more information than **just one of them**



Reprinted by permission from IEEE TRANSACTIONS ON SY
Vol. SMC-4, No. 1, January 1974,

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_1' denote an independent performance of experiment E_1. 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 copies of the "worst" experiment E_2 are preferred. This is true despite

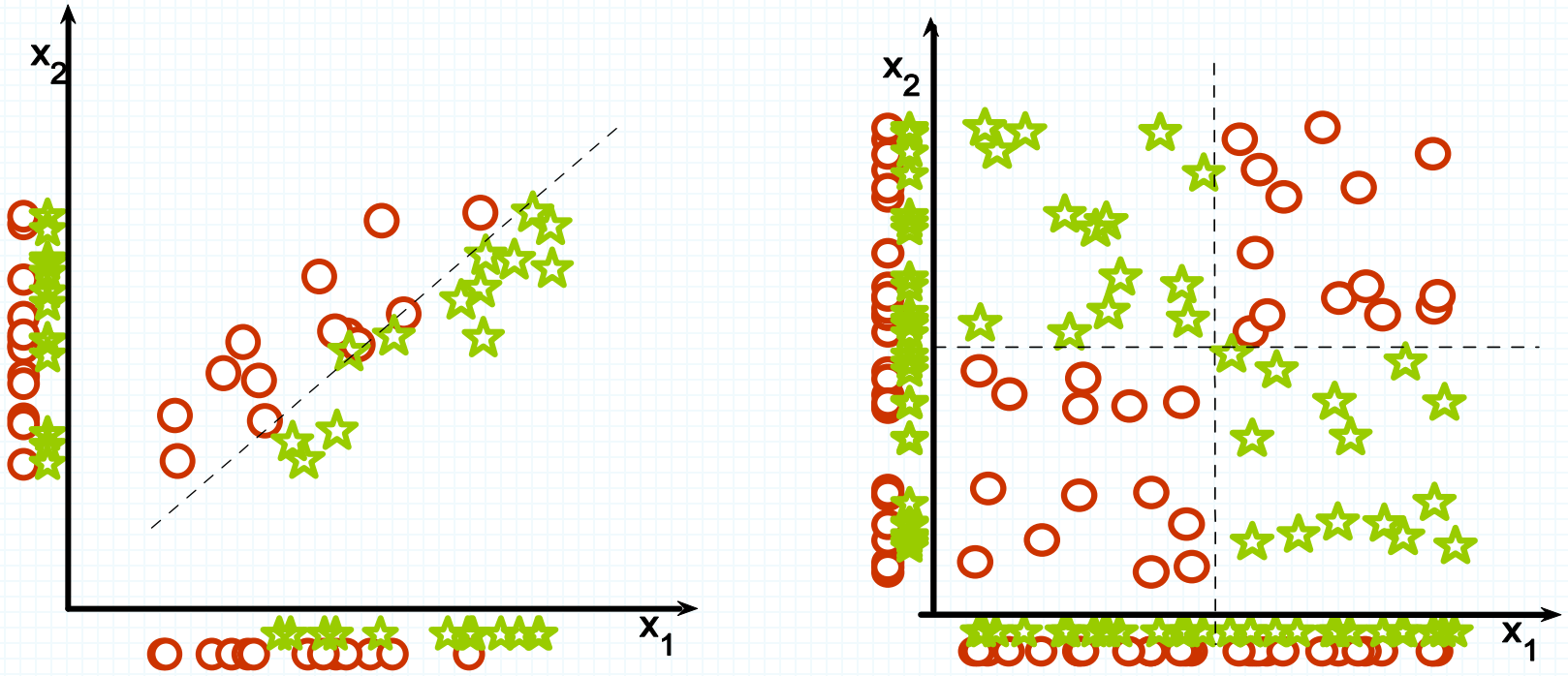
Univariate feature selection

- 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)
```

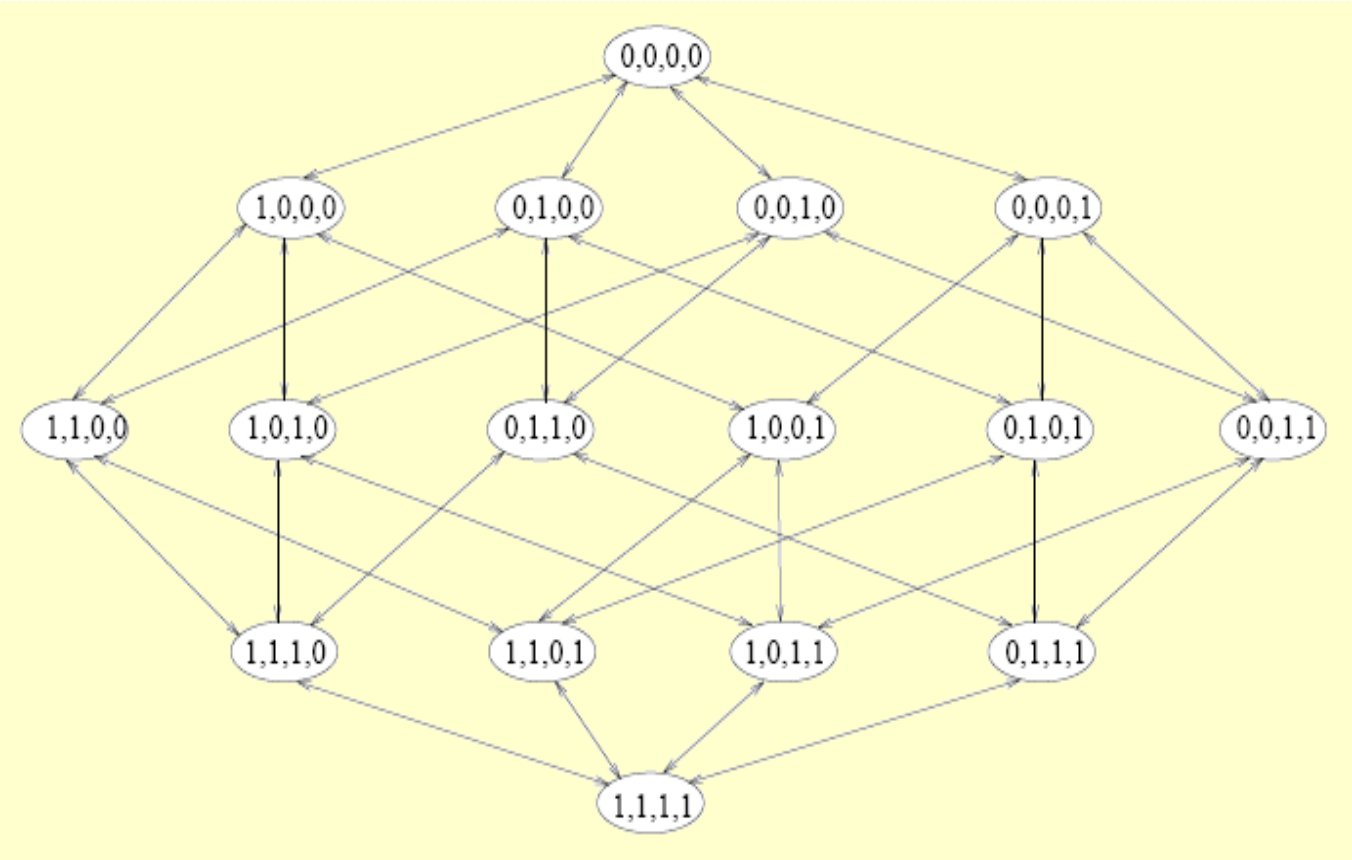
Univariate feature selection

- 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.



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.

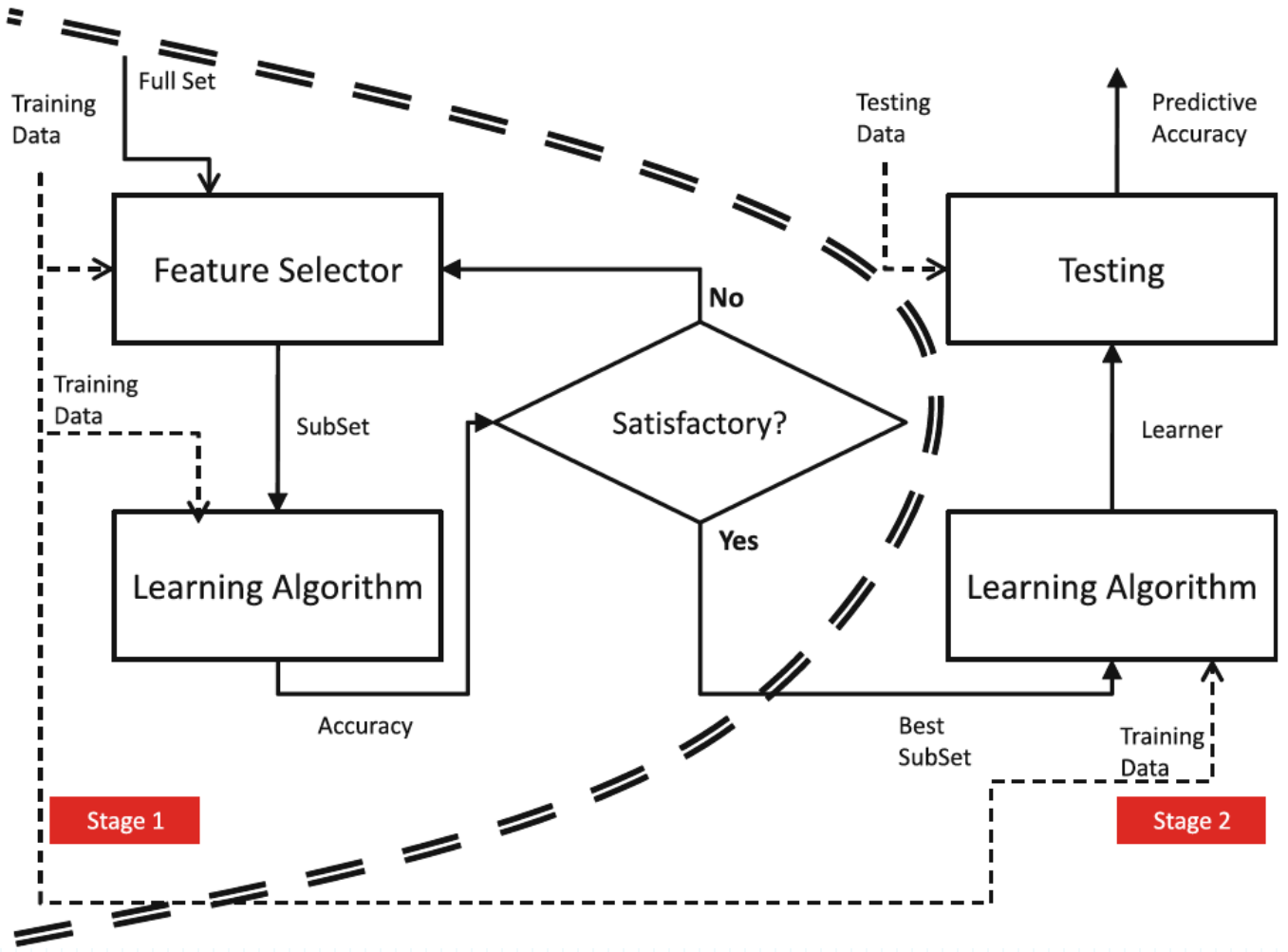


Multivariate feature selection

- 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?

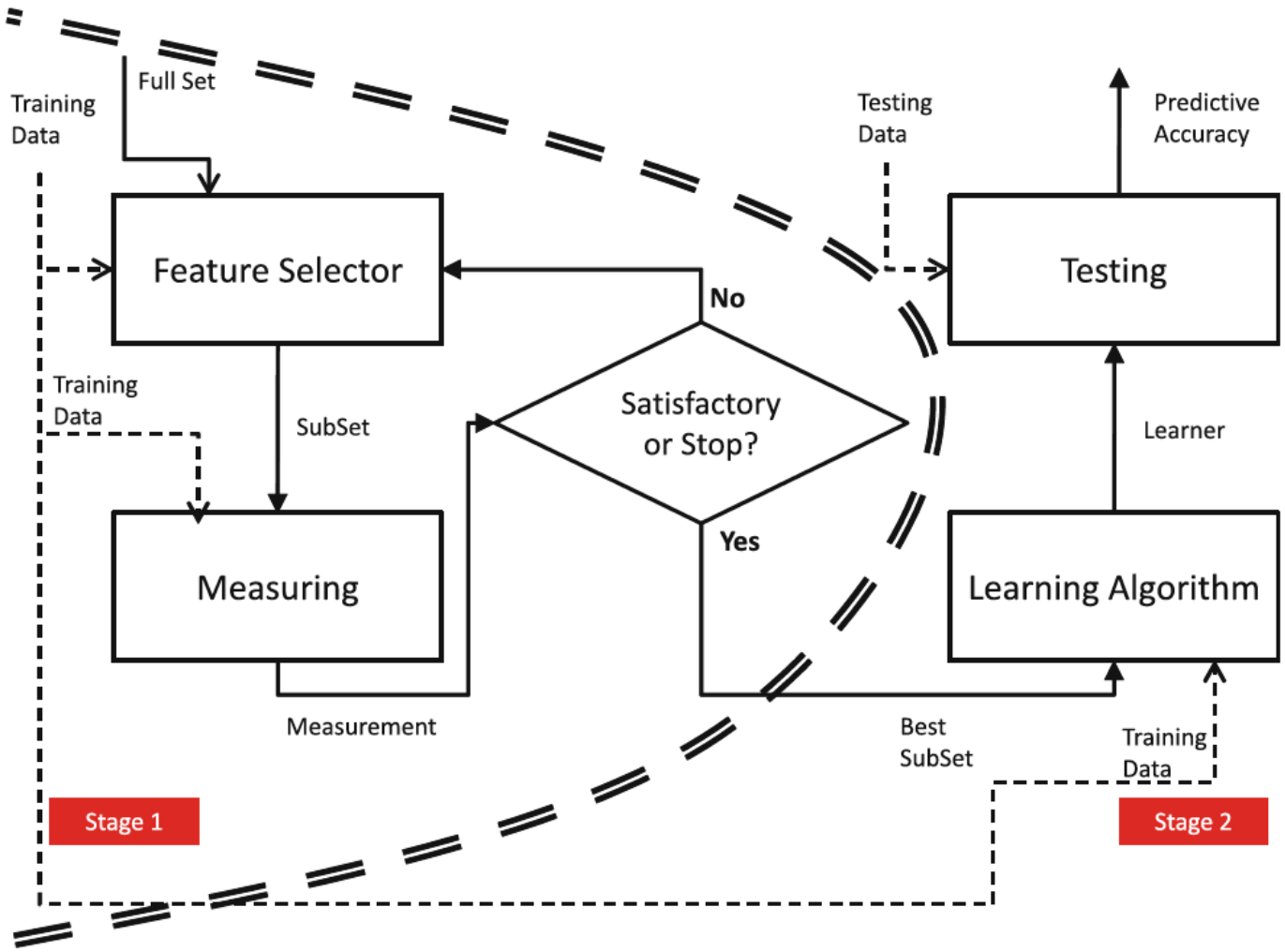
Wrapper Methods

- 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.



New “Hyperparameters”!

- Which feature evaluation technique
- How many K feature you need to select
- Which feature selection technique

Next Lecture!