

April 27, Kyiv

**Data Science & Mathematical  
Modeling Bachelor Program**

**Course “Basics of Machine Learning”**

**Lecture 6: Regression Metrics. Feature  
Engineering**



Oleg CHERTOV

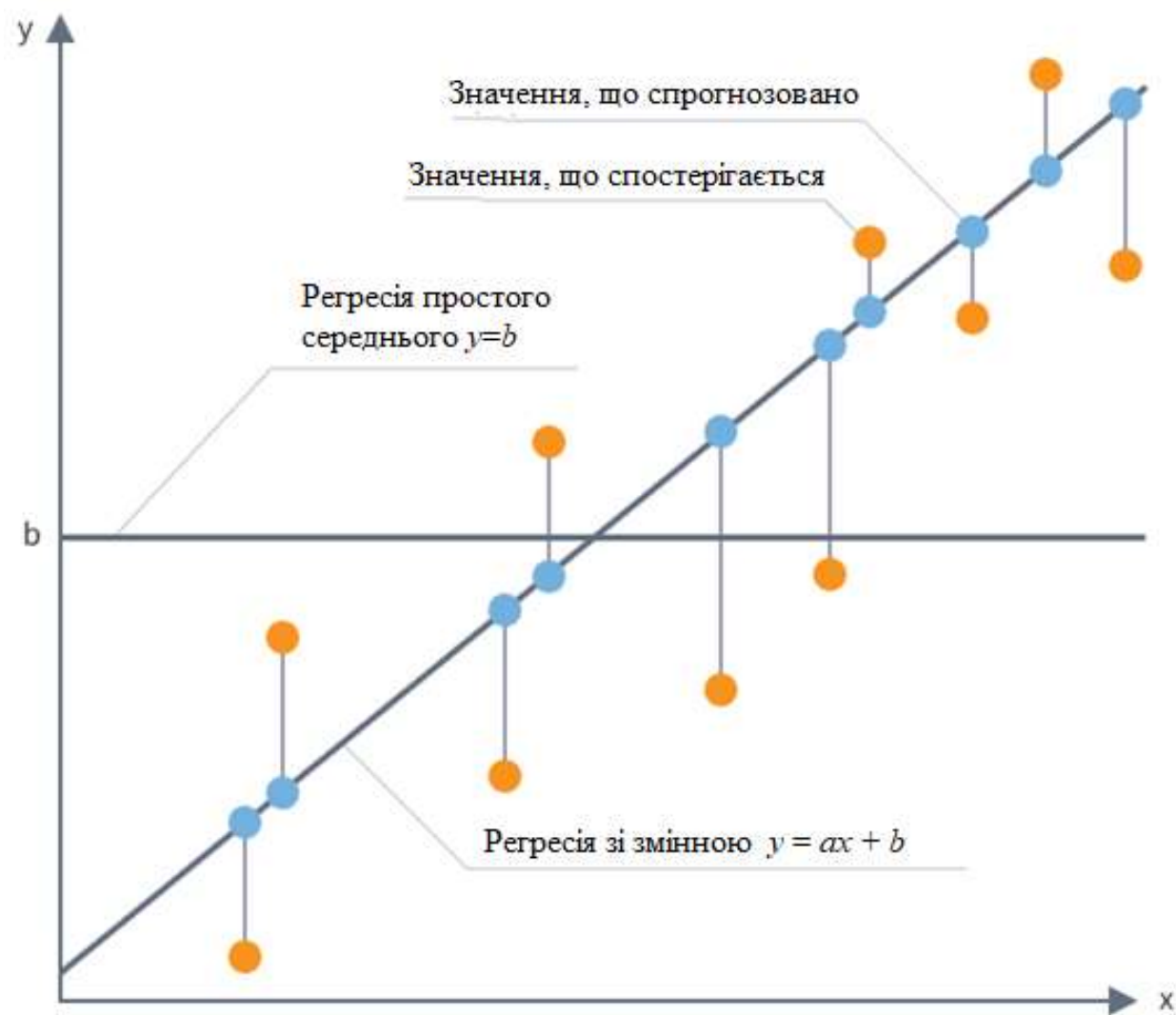
Professor, Sc.D. (Doctor Habilitatus),  
Head of the Applied Mathematics Department



Applied Mathematics Department  
Igor Sikorsky Kyiv Polytechnic Institute  
Ukraine



# Регресійні метрики



# Регресійні метрики

- *Mean Bias Error (МБЕ)*, середня помилка зміщення:

$$MBE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)$$

$\hat{y}_i$  = *predicted value*

$y_i$  = *actual value*

$n$  = *# of observations*

# Регресійні метрики

- ❑ ~~Mean Bias Error (MBE), середня помилка зміщення:~~

$$\del{MBE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)}$$

$\hat{y}_i$  = predicted value

$y_i$  = actual value

$n$  = # of observations

- ❑ Mean Squared Error (MSE), середньоквадратична помилка:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- ❑ Якщо великі помилки для нас справді неприйнятні, то квадратичний штраф за них - дуже корисна властивість (і її навіть можна посилювати, підвищуючи ступінь, до якого ми зводимо помилку на об'єкті)
- ❑ Однак якщо в тестових даних присутні викиди, то буде складно об'єктивно порівняти моделі між собою: помилки на викидах маскуватимуть відмінності в помилках на основній множині об'єктів
- ❑ Отже, якщо ми порівнюватимемо дві моделі за допомогою MSE, у нас виграватиме та модель, у якої менша помилка на об'єктах-викидах, а це, найімовірніше, не те, чого вимагає від нас наша бізнес-задача
- ❑ Складна інтерпретація (із-за піднесення у квадрат), але як функція втрат - завжди диференційовна

# Регресійні метрики

- ~~Mean Bias Error (MBE), середня помилка зміщення:~~

~~$$MBE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)$$~~

$\hat{y}_i$  = predicted value

$y_i$  = actual value

$n$  = # of observations

- Mean Squared Error (MSE), середньоквадратична помилка:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Якщо зашумлені дані, то піднесення до квадрату зробить похибку ще більше, тобто викиди є критичними
- Складна інтерпретація (із-за піднесення у квадрат)
- + Функція завжди диференційовна + МНК

- Root Mean Squared Error (RMSE), корінь із середньоквадратичної помилки:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- ± Викиди не так критичні, як в MSE
- + Проста інтерпретація (розмірність даних)
- ± Функція практично завжди диференційовна
- Складніші обчислення, ніж в MSE

- Mean Absolute Error (MAE), середня абсолютна помилка:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

- ± Викиди не так критичні, як в MSE
- + Проста інтерпретація + дає лінійну оцінку (різниця між 0 і 10 вдвічі більша, ніж між 0 і 5, що не так в RMSE)
- Функція не завжди диференційовна
- + Простіші обчислення, ніж в MSE і RMSE

# Регресійні метрики

- Крамниця 1: прогнозували продати 9 штук, продали: 10,  $MSE=RMSE=MAE=1$
- Крамниця 2: прогнозували продати 999 штук, продали: 1000,  $MSE=RMSE=MAE=1$
- Буває, що відносні помилки важливіші за абсолютні

$\hat{y}_i$  = *predicted value*  
 $y_i$  = *actual value*  
 $n$  = *# of observations*

# Регресійні метрики

- Крамниця 1: прогнозували продати 9 штук, продали: 10,  $MSE=RMSE=MAE=1$
- Крамниця 2: прогнозували продати 999 штук, продали: 1000,  $MSE=RMSE=MAE=1$

□ Буває, що відносні помилки важливіші за абсолютні

□ Середні помилки **у відсотках**:

□ *MAPE (Mean Absolute Percentage Error,*  
середня абсолютна помилка у відсотках):

✓ Для першої крамниці:  $100\%|(10-9)/10| = 10\%$

✓ Для другої крамниці:  $100\%|(1000-999)/1000| = 0,1\%$

□ Ця помилка не має розмірності й дуже проста в інтерпретації. Її можна виражати як у частках, так і у відсотках

□ Якщо вийшло, наприклад, що  $MAPE=10\%$ , то це означає, що помилка склала 10% від фактичного значення

$\hat{y}_i$  = *predicted value*

$y_i$  = *actual value*

$n$  = *# of observations*

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

# Регресійні метрики

- Крамниця 1: прогнозували продати 9 штук, продали: 10,  $MSE=RMSE=MAE=1$
- Крамниця 2: прогнозували продати 999 штук, продали: 1000,  $MSE=RMSE=MAE=1$
- Буває, що відносні помилки важливіші за абсолютні
- Середні помилки у відсотках:

$\hat{y}_i$  = *predicted value*  
 $y_i$  = *actual value*  
 $n$  = *# of observations*

- *MAPE (Mean Absolute Percentage Error)*

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

- *MSPE (Mean Squared Percentage Error)*

$$MSPE = \frac{100\%}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2$$

- *RMSLE (Root Mean Squared Log Error)*

$$\begin{aligned} RMSLE &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2} = \\ &= RMSE(\log(y_i + 1), \log(\hat{y}_i + 1)) = \\ &= \sqrt{MSE(\log(y_i + 1), \log(\hat{y}_i + 1))} \end{aligned}$$



# Регресійні метрики

- Крамниця 1: прогнозували продати 9 штук, продали: 10,  $MSE=RMSE=MAE=1$
- Крамниця 2: прогнозували продати 999 штук, продали: 1000,  $MSE=RMSE=MAE=1$
- Буває, що відносні помилки важливіші за абсолютні
- Середні помилки у відсотках:

➤ *MAPE (Mean Absolute Percentage Error)*

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

➤ *MSPE (Mean Squared Percentage Error)*

$$MSPE = \frac{100\%}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2$$

➤ *RMSLE (Root Mean Squared Log Error)*

$$\begin{aligned} RMSLE &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2} = \\ &= RMSE(\log(y_i + 1), \log(\hat{y}_i + 1)) = \\ &= \sqrt{MSE(\log(y_i + 1), \log(\hat{y}_i + 1))} \end{aligned}$$

$\hat{y}_i$  = *predicted value*

$y_i$  = *actual value*

$n$  = *# of observations*

- *MAPE* дуже чутливе до масштабу: якщо фактичне значення близьке до нуля, то *MAPE* різко збільшується
- *MAPE* несиметричне:
  - $100\% |(10-20)/10| = 100\%$
  - $100\% |(30-20)/30| = 33,3\%$
- На даних невеликого обсягу *MAPE* оманливо погане:
  - $100\% |(1-2)/1| = 100\%$ , хоча помилка лише на 1

## Регресійні метрики. Коефіцієнт детермінації ( $\mathcal{R}^2$ )

- Альтернативний підхід до уникнення недоліків метрик з абсолютними значеннями помилок - коефіцієнт детермінації  $\mathcal{R}^2$

(*coefficient of determination*)

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\bar{y} - y_i)^2}$$

- Фактично, в чисельнику стоїть середньоквадратична помилка регресійної моделі, а в знаменнику – середньоквадратична помилка моделі, яка дає сталу – середньоарифметичне значення цільової змінної
- Іншими словами,  $\mathcal{R}^2$  показує долю дисперсії залежної змінної, яка пояснюється за допомогою регресійної моделі
- Коефіцієнт детермінації – безрозмірна величина в діапазоні  $(-\infty, 1]$

# Регресійні метрики. Коефіцієнт детермінації ( $\mathcal{R}^2$ )

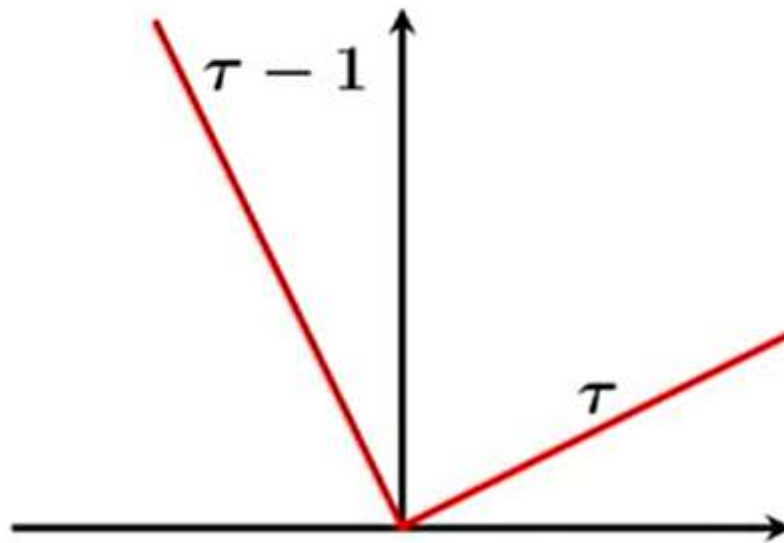
- Значення  $\mathcal{R}^2$  близькі до 1 вказують на високий ступінь відповідності моделі даним
- $\mathcal{R}^2 = 0$  показує, що між незалежною і залежною змінними моделі відсутня функціональна залежність, маємо регресію простого середнього (= моделі з константою)
- Коефіцієнт набуває від'ємних значень (зазвичай невеликих), коли додавання до моделі з константою деякої змінної тільки погіршує її
- Незважаючи на те, що коефіцієнт детермінації показує відсоток дисперсії, поясненої регресійною моделлю, сильно покладатися на нього не варто. Бо коефіцієнт детермінації буде завжди тим більшим, чим більша кількість параметрів (= змінних) у моделі

$$R_{adj}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k}$$

Скоригований (*adjusted*) коефіцієнт детермінації компенсує своє зростання тільки за рахунок збільшення кількості  $k$  незалежних предикторів (змінних моделі) нормуванням, де  $n$  – розмір вибірки

# Регресійні метрики. Квантильна помилка (для несиметричних функцій втрат)

$$\rho_{\tau}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} ((\tau - 1)[y_i < a(x_i)] + \tau[y_i \geq a(x_i)])(y_i - a(x_i))$$



*Alexei Botchkarev (2018).  
Performance Metrics (Error  
Measures) in Machine Learning  
Regression, Forecasting and  
Prognostics:  
проаналізовано 40+ метрик,  
але цієї там не має ☹*

<https://arxiv.org/ftp/arxiv/papers/1809/1809.03006.pdf>

# Перенавчання свідчить, що модель занадто складна

- Якщо модель працює набагато краще на навчальному наборі даних, ніж на тестовому, то це свідчить про перенавчання
- **Перенавчання** означає, що модель апроксимує параметри занадто близько до окремо взятих спостережень у навчальному наборі, але не узагальнюється добре на реальних даних (=модель має високу дисперсію)
- Причина перенавчання полягає в тому, що наша модель занадто складна для наявних навчальних даних (типу інтерполяційного поліному Лагранжа)

# Загальні рішення для підвищення узагальнюючої здатності моделі

- ❖ зібрати більше навчальних даних ==> дорого і не завжди можливо
- ❖ вибрати простішу модель за рахунок підбору гіперпараметрів (підрізання дерев прийняття рішень за рахунок фіксації глибини дерева тощо)
- ❖ ввести штраф за складність на основі регуляризації (розглядали на прикладі лінійних моделей та кластеризації)
- ❖ **знижити розмірність даних** ==> предмет цієї лекції

# Features, Label and Dimension in a Dataset



Position	Experience	Skill	Country	City	Salary (\$)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

- ❑ *Features (attributes) are individual independent variables which acts as the input in the system*
- ❑ *Prediction models uses these features to make predictions. To make it simple, you can consider one column of your dataset to be one feature. The number of features is **dimension***
- ❑ *Labels are the final output or target Output. We obtain labels as output when provided with features as input*



# Які бувають ознаки (атрибути, характеристики, фактори)?

Ознаки можуть бути таких видів:

- ❑ **Бінарні**, які приймають тільки два значення. Наприклад,  $[true, false]$ ,  $[0, 1]$ , ["так", "ні"]
- ❑ **Категоріальні** (або ж **номінальні**). Вони мають кінцеву кількість значень, наприклад, ознака "день тижня" має 7 різних значень: понеділок, вівторок тощо до неділі
- ❑ **Упорядковані**. Певною мірою схожі на категоріальні ознаки. Різниця між ними в тому, що в цьому випадку існує чітке впорядкування категорій. Наприклад, "класи в школі" від 1 до 11. Сюди ж можна віднести «година доби», яка має 24 значення та є впорядкованою
- ❑ **Числові (кількісні)**. Це значення в діапазоні від мінус нескінченності до плюс нескінченності, які не можна віднести до попередніх трьох типів ознак



# Features, Label and Dimension in a Dataset



Position	Experience	Skill	Country	City	Salary (\$)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

- *Features (attributes)* are individual independent variables which acts as the input in the system
- *Prediction models* uses these features to make predictions. To make it simple, you can consider one column of your dataset to be one feature. The number of features is *dimension*
- *Labels* are the final output or target Output. We obtain labels as output when provided with features as input

Але чи завжди ми відразу маємо ознаки? Де їх тоді взяти?

# Features, Label and Dimension in a Dataset

Customer		
Customer ID	Email	Registration date
0	dsjdns@gmail.com	15/07/2021
1	nxneje@gmail.com	15/08/2021

Interactions					
Interaction ID	Customer ID	Date of purchase	Type	Amount	
0	0	16/07/2021 09:21:01	Add to cart	N/A	
1	0	16/07/2021 09:21:56	Purchase	60	
2	0	17/07/2021 17:54:32	Purchase	400	
3	1	16/08/2021 10:32:09	Add to cart	N/A	
4	1	16/08/2021 10:33:03	Purchase	30000	

- ❑ Нехай, у БД інтернет-магазину є таблиця "Покупці", що містить один рядок для кожного клієнта, який відвідав сайт
- ❑ У БД також може бути таблиця "Дії", що містить рядок для кожної взаємодії (кліка або відвідування сторінки), яку клієнт здійснив на сайті. Ця таблиця також містить інформацію про час дії користувача та тип відповідної події ("Покупка", "Пошук" або "Додати в кошик"). Ці дві таблиці пов'язані між собою стовпчиком "Customer ID"
- ❑ А які ознаки можна побудувати, крім "Скільки днів пройшло з дати реєстрації?"
- ❑ Потрібна таблиця "Покупці-Ознаки"

# Features, Label and Dimension in a Dataset

Customer		
Customer ID	Email	Registration date
0	<a href="mailto:dsjdns@gmail.com">dsjdns@gmail.com</a>	15/07/2021
1	<a href="mailto:nxneje@gmail.com">nxneje@gmail.com</a>	15/08/2021

Interactions				
Interaction ID	Customer ID	Date of purchase	Type	Amount
0	0	16/07/2021 09:21:01	Add to cart	N/A
1	0	16/07/2021 09:21:56	Purchase	60
2	0	17/07/2021 17:54:32	Purchase	400
3	1	16/08/2021 10:32:09	Add to cart	N/A
4	1	16/08/2021 10:33:03	Purchase	30000

- ❑ А які ознаки можна побудувати, крім "Скільки днів пройшло з дати реєстрації?"
- ✓ Середній час між покупками
- ✓ Середня сума покупок
- ✓ Максимальна сума покупок
- ✓ Час, що минув з моменту останньої покупки
- ✓ Загальна кількість покупок

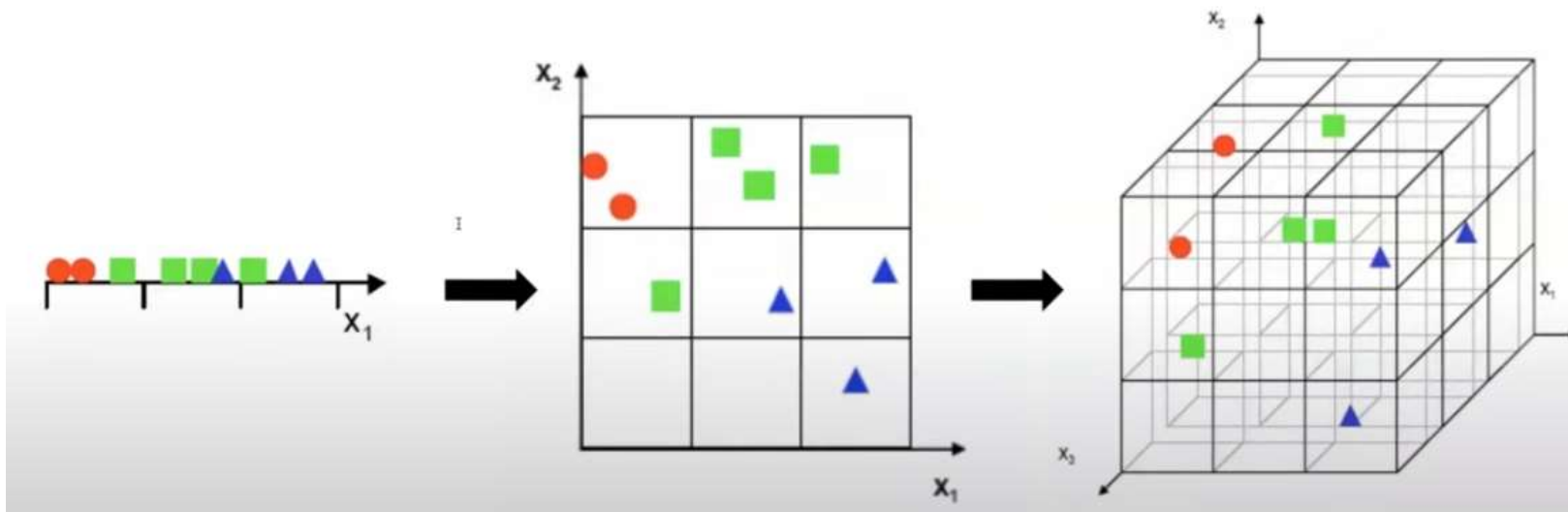
# Додаткова мотивація для зниження розмірності даних - 1

- Прибрати шумові (залишкові) характеристики, які не мають користі, дублюються чи, навіть, погіршують прогноз
- Прибрати малоінформативні характеристики, оскільки вартість обчислення певних характеристик може бути зіставна з вартістю збитків від помилкових прогнозів
- Менше розмірність<sup>1</sup> ==> більш швидке навчання ==> більше варіантів гіперпараметрів можна розглянути за фіксований час ==> кращий кінцевий результат, економія ресурсів

<sup>1</sup> Але метод  $SVM$  працює тим краще, чим вище розмірність

# Додаткова мотивація для зниження розмірності даних - 2

- Візуалізація та наочність: складно дивитися на 100-мірний простір, потрібен 2-3-х вимірний простір
- Розуміння даних
- Прокляття розмірності (*curse of dimensionality*):



3 варіанти

$3 \times 3 = 9$  варіантів











$3 \times 3 \times 4 = 36$  варіантів

- ❖ **Задача діагностики хвороби;**  
є вибірка із 100 хворих
- ❖  $x_1$ : температура (низька, нормальна, висока)
- ❖  $x_2$ : рівень тиску (низький, нормальний, підвищений)
- ❖  $x_3$ : група крові (I, II, III, IV)
- ❖ ...
- ❖  $x_{1000}$ :  $> 3^{1000}$  різних варіантів, всі хворі «розкидані» по вузлах гіперкуба і несхожі одне на одного



# Додаткова мотивація для зниження розмірності даних - 3

<https://archive.ics.uci.edu/ml/datasets.php?format=&task=&att=&area=&numAtt=&numIns=&type=&sort=attDown&view=table>

Machine Learning Repository		View ALL Data Sets					
Center for Machine Learning and Intelligent Systems							
Browse Through: 622 Data Sets		Table View List View					
Default Task	Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
Classification (466) Regression (151) Clustering (121) Other (56)	 URL Reputation	Multivariate, Time-Series	Classification	Integer, Real	2396130	3231961	2009
Attribute Type	 KASANDR	Multivariate	Causal-Discovery	Integer	17764280	2158859	2017
Categorical (38) Numerical (422) Mixed (55)	 Gas sensor arrays in open sampling settings	Multivariate, Time-Series	Classification	Real	18000	1950000	2013
Data Type	 YouTube Multiview Video Games Dataset	Multivariate, Text	Classification, Clustering	Integer, Real	120000	1000000	2013
Multivariate (480) Univariate (30) Sequential (59) Time-Series (126) Text (69) Domain-Theory (23) Other (21)	 Twin gas sensor arrays	Multivariate, Time-Series, Domain-Theory	Classification, Regression	Real	640	480000	2016
Area	 Deepfakes: Medical Image Tamper Detection	Multivariate	Classification	Real	20000	200000	2020
Life Sciences (147) Physical Sciences (57) CS / Engineering (234) Social Sciences (41) Business (45) Game (12) Other (81)	 Gas sensor array exposed to turbulent gas mixtures	Multivariate, Time-Series	Classification, Regression	Real	180	150000	2014
# Attributes	 ElectricityLoadDiagrams20112014	Time-Series	Regression, Clustering	Real	370	140256	2015
Less than 10 (166) 10 to 100 (279) Greater than 100 (110)	 PEMS-SF	Multivariate, Time-Series	Classification	Real	440	138672	2011
# Instances	 Gas sensor array under flow modulation	Multivariate, Time-Series	Classification, Regression	Real	58	120432	2014
Less than 100 (38) 100 to 1000 (210) Greater than 1000 (335)							
Format Type							
Matrix (439) Non-Matrix (183)							

- На практиці часто зустрічаються задачі, коли кількість ознак набагато більше кількості об'єктів. Це типова ситуація для медичних даних
- Другий приклад – набори даних для машинного навчання відомого репозитарія : *UCI Machine Learning Repository: Data Sets*

# Підходи до зниження розмірності даних

- Відбір інформативних характеристик (*feature selection*)

*Data space* → *Data subspace*

Відбір деякої підмножини найбільш змістовних характеристик

- ❖ (*filter methods*, методи фільтрації) – методи попарного відбору характеристик
- ❖ (*wrapper methods*, методи обгортки) – методи послідовного відбору характеристик

- Синтез інформативних характеристик (*feature extraction*)

*Data space* → *Feature space*

Конструюється простір з новими характеристиками

- Методи, вбудовані в модель навчання (*embedded models*), – наприклад: *LASSO* як *feature selection* в лінійній регресії, *ЛДА* як *feature extraction*, *Decision Trees with pruning*

# Відбір інформативних характеристик

- Відбір інформативних характеристик (*feature selection*)

*Data space* → *Data subspace*

Відбір деякої підмножини найбільш змістовних ознак

- ❖ (*filter methods*, методи фільтрації) – методи попарного відбору характеристик:
  - *Correlation feature selection*
  - *Minimum redundancy maximum relevance*
- ❖ (*wrapper methods*, методи обгортки) – методи послідовного відбору характеристик



# Виявлення неінформативних ознак за допомогою фільтрації

- Для знаходження надлишкових або нерелевантних ознак використовуються статистичні методи:

- ❖ коефіцієнт кореляції

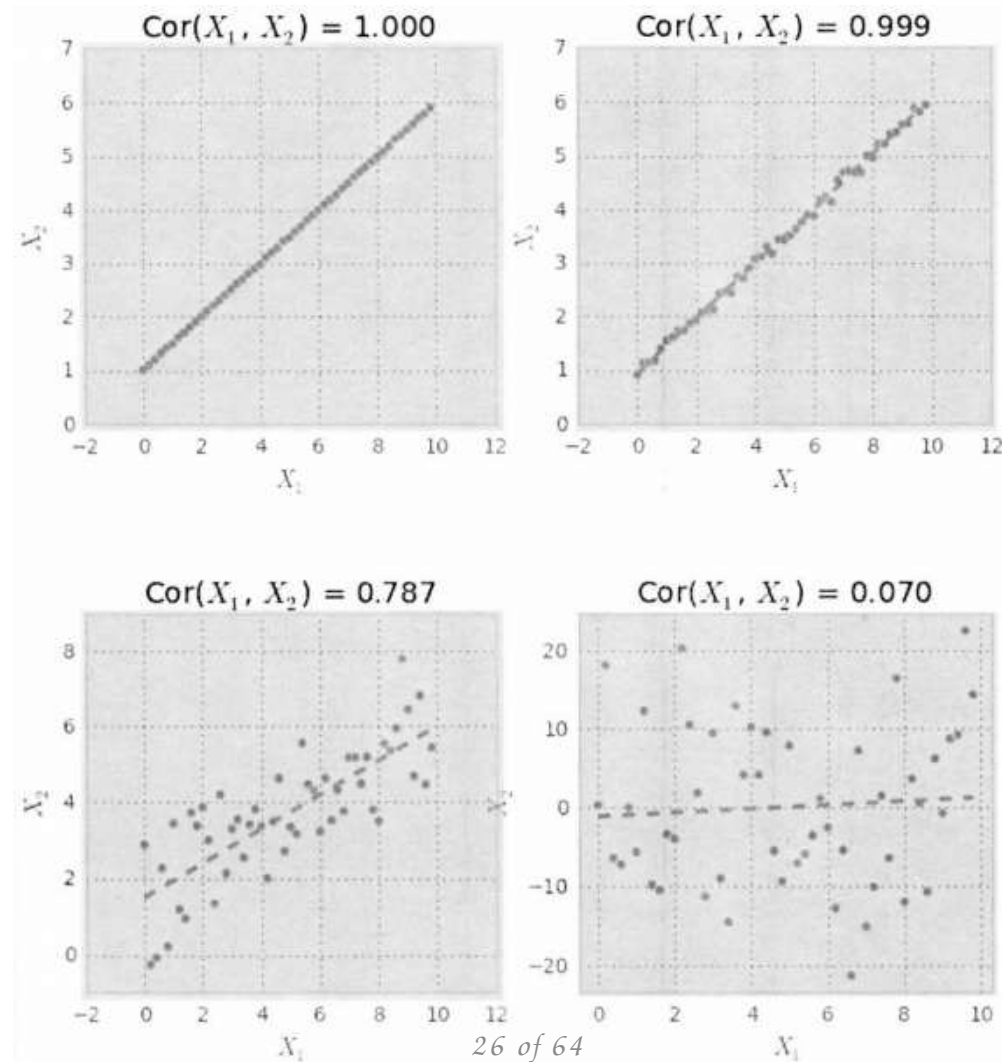
(*Correlation feature selection*): 
$$r(X, Y) = \frac{\sum_x (x - \bar{x}) \sum_y (y - \bar{y})}{\sqrt{\sum_x (x - \bar{x})^2} \sqrt{\sum_y (y - \bar{y})^2}}$$

- ❖ взаємна інформація (*Minimum redundancy maximum relevance, mRMR*):

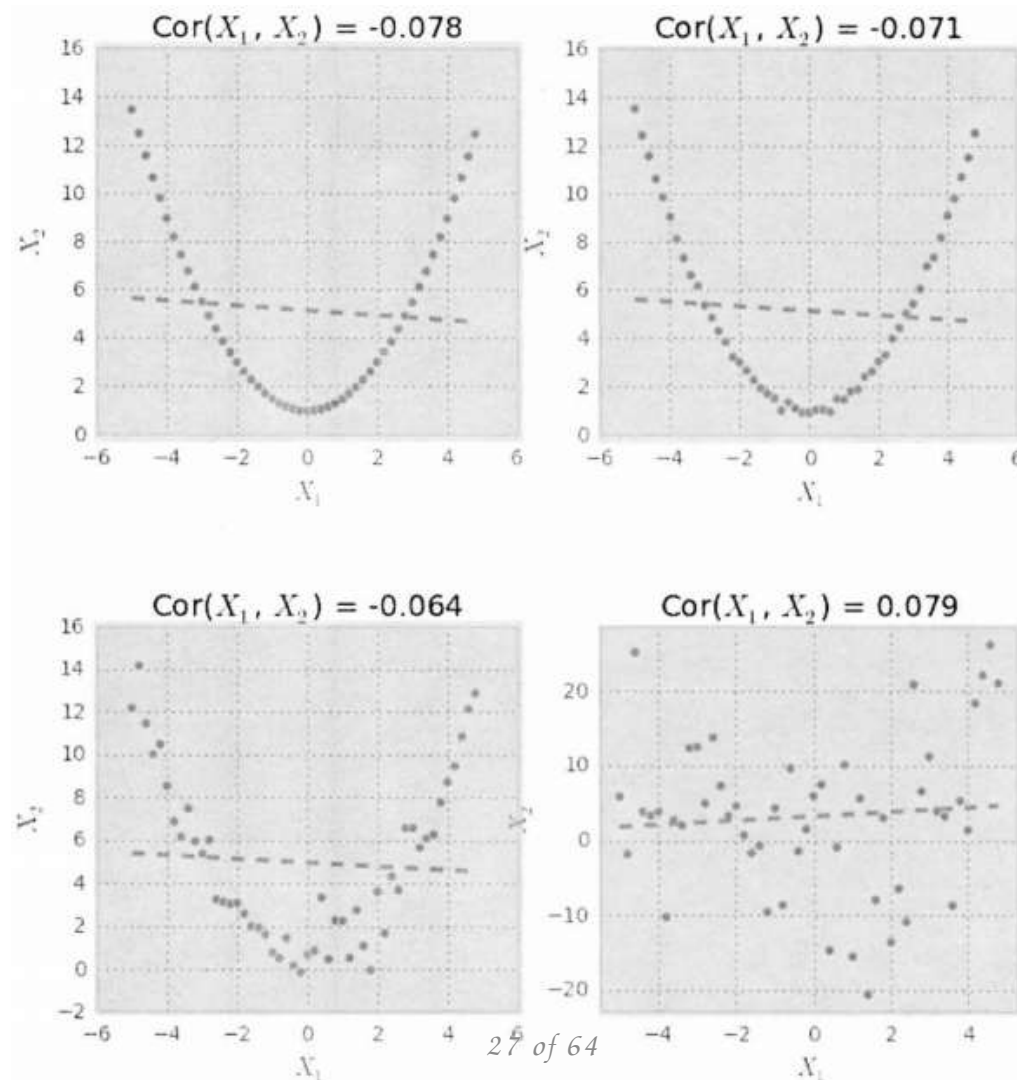
$$I(X, Y) = \sum_x \sum_y p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right)$$

- Якщо знайдені надлишкові ознаки, то з кожної групи ми залишаємо тільки один, а нерелевантні ознаки просто видаляються

# Кореляція **виявляє** лінійні зв'язки між парами ознак



# Кореляція **не виявляє** нелінійні зв'язки між парами ознак



# Взаємна інформація. Ентропія Шеннона

- Говорячи про відбір ознак, ми не повинні зосереджуватися на типі залежності (лінійна, квадратична, ще якась)
- Замість цього потрібно думати про те, скільки інформації привносить ознака (за умови, що вже є інша ознака)
- Поняття взаємної інформації обчислює кількість інформації, загальної для двох ознак, воно базується на понятті **ентропії інформації**:

$$H(X) = \sum_{i=1}^n p(X_i) \log_2 p(X_i)$$

# Взаємна інформація

- Модифікуємо формулу ентропії  $H(X)$ , застосувавши її до двох ознак замість однієї, так щоб вона вимірювала, наскільки зменшується невизначеність ознаки  $X$ , коли ми дізнаємося про значення ознаки  $Y$
- Таким чином, ми зможемо дізнатися, як одна ознака знижує невизначеність іншої
- Наприклад, за відсутності будь-якої інформації про погоду, ми не можемо сказати, йде на вулиці дощ чи ні — повна невизначеність
- Але якщо ми знаємо, що трава мокра, то невизначеність зменшується (правда, потрібно ще перевірити, чи це не активована поливальна система)

# (Нормована) взаємна інформація

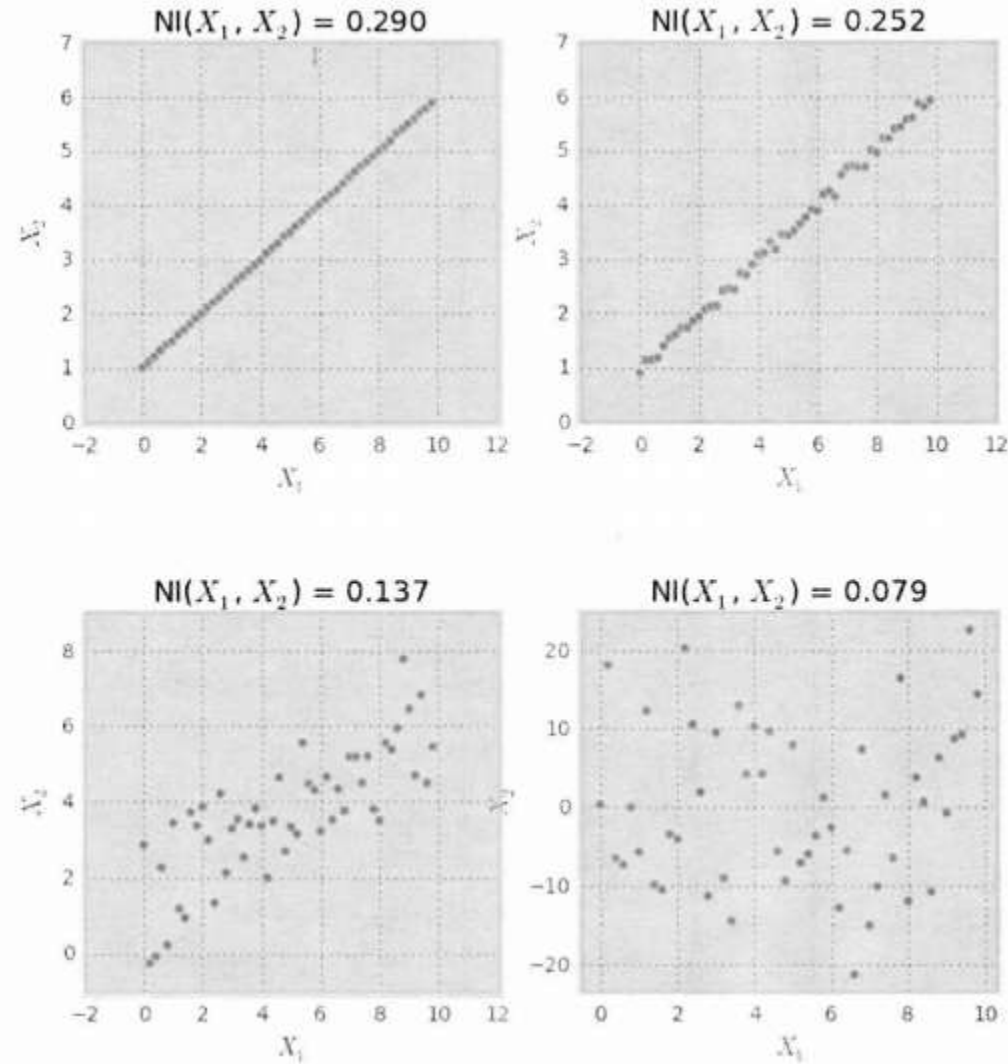
- Формально взаємна інформація визначається наступним чином:

$$I(X;Y) = \sum_{i=1}^m \sum_{j=1}^n P(X_i, Y_j) \log_2 \frac{P(X_i, Y_j)}{P(X_i)P(Y_j)}$$

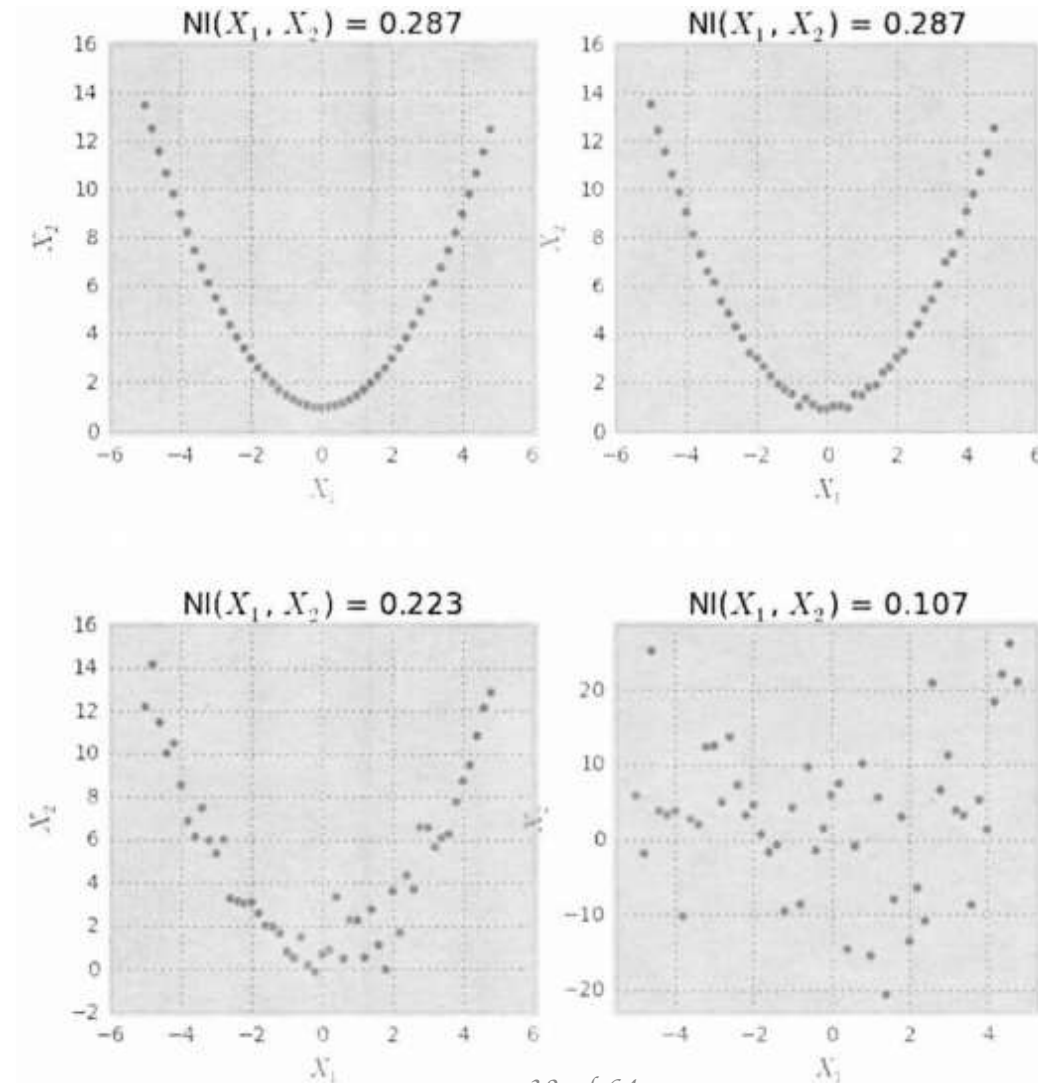
- Щоб привести взаємну інформацію до діапазону  $[0, 1]$ , потрібно розділити її на суму ентропій окремих ознак. В результаті отримуємо нормовану взаємну інформацію:

$$NI(X;Y) = \frac{I(X;Y)}{H(X) + H(Y)}$$

# (Нормована) взаємна інформація (для лінійної залежності)



# Взаємна інформація (для нелінійної залежності)





# Ідея методів фільтрації

- Таким чином, ми повинні обчислити нормовану взаємну інформацію для **всіх** пар ознак
- Якщо для якоїсь пари значення виявилось занадто великим (ще треба визначити, що це означає), то одну із ознак слід відкинути
- У разі регресії ми могли б відкинути також ознаку, для якої дуже мала взаємна інформація з бажаним результуючим значенням

# Перший недолік методів фільтрації

- Цей підхід годиться, коли набір ознак не дуже великий
- Але, починаючи з якогось моменту, процедура стає занадто дорогою, оскільки обсяг обчислень зростає квадратично (обчислюється взаємна інформація для кожної пари ознак)

# Maximum Dependency Feature Selection is Combinatorial!

- *Number of subspaces for space with  $n$  elements*
  - *For given number  $k$  of features: ??*

# Maximum Dependency Feature Selection is Combinatorial!

- *Number of subspaces for space with  $n$  elements*
  - *For given number  $k$  of features:  $C_n^k = \frac{n!}{k!(n-k)!}$*
  - *Total:  $C_n^0 + C_n^1 + C_n^2 + \dots + C_n^n = 2^n$*
- *Example*

$n$	1000	1000	1000	5000
$k$	1	2	3	3
# configurations of selected variables	$10^3$	$\sim 0.50 \times 10^6$	$\sim 1.66 \times 10^8$	$\sim 2.08 \times 10^{10}$

- *Heuristic search algorithms are necessary*
- *Simplest case: the incremental search (тобто поступово (по одному) збільшуємо кількість ознак)*

# Критерій mRMR (мінімальна надмірність максимальна релевантність)

*Peng H. C., Long F., Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2005. — Т. 27, вып. 8.*

- Будемо відбирати ознаки, які мають найбільшу взаємну інформацію з відповідями («максимальна релевантність»)

$$Relevance(F, c) = \frac{1}{|F|} \sum_{f_i \in F} I(f_i, c)$$

- Будемо штрафувати ознаки за надмірність, в контексті вже відібраних ознак («мінімальна надмірність»)

$$Redundancy(F) = \frac{1}{|F|^2} \sum_{f_i, f_j \in F} I(f_i, f_j)$$

- Тоді критерій  $mRMR$  має вигляд:

$$mRMR = \max_F (Relevance(F, c) - Redundancy(F))$$

# Search Algorithm of mRMR

## □ Greedy search algorithm

1) In the set  $S$  of  $n$  factors find the variable  $x_1$  that has the largest  $\text{Relevance}(\cdot, c)$ .

2) Exclude  $x_1$  from  $S$

3) Search  $x_2$  so that it maximizes

$$\text{Relevance}(\cdot, c) - \sum \text{Redundancy}(\cdot, x_1)/n$$

4) Iterate this process (step 3) until an expected number  $k$  of variables have been obtained, or other constraints are satisfied

## □ Complexity $O(n * k)$

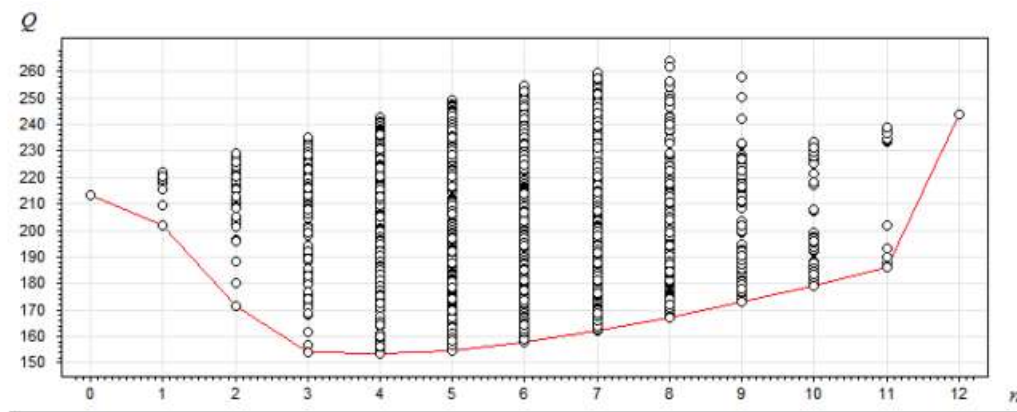
# Другий недолік методів фільтрації

- Ще один істотний недолік фільтрації полягає в відкиданні ознак, які здаються марними окремо, але разом є важливими

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

□ Операція XOR і окремі ознаки

# Методы обгортки. Алгоритм повного перебора - 1

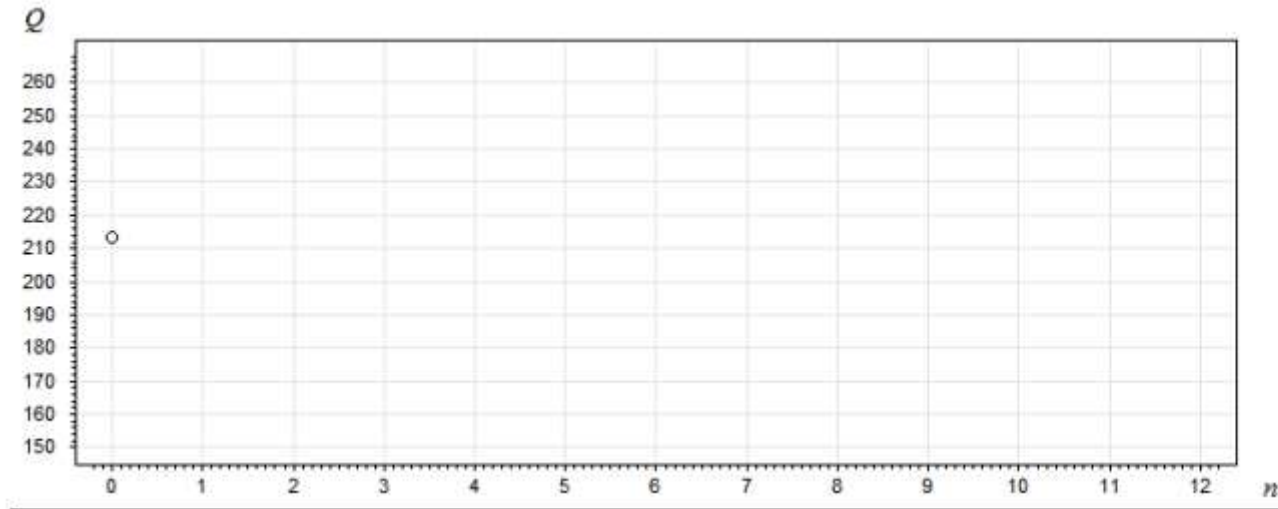


**Вход:** множество  $F$ , критерий  $Q$ , параметр  $d$ ;

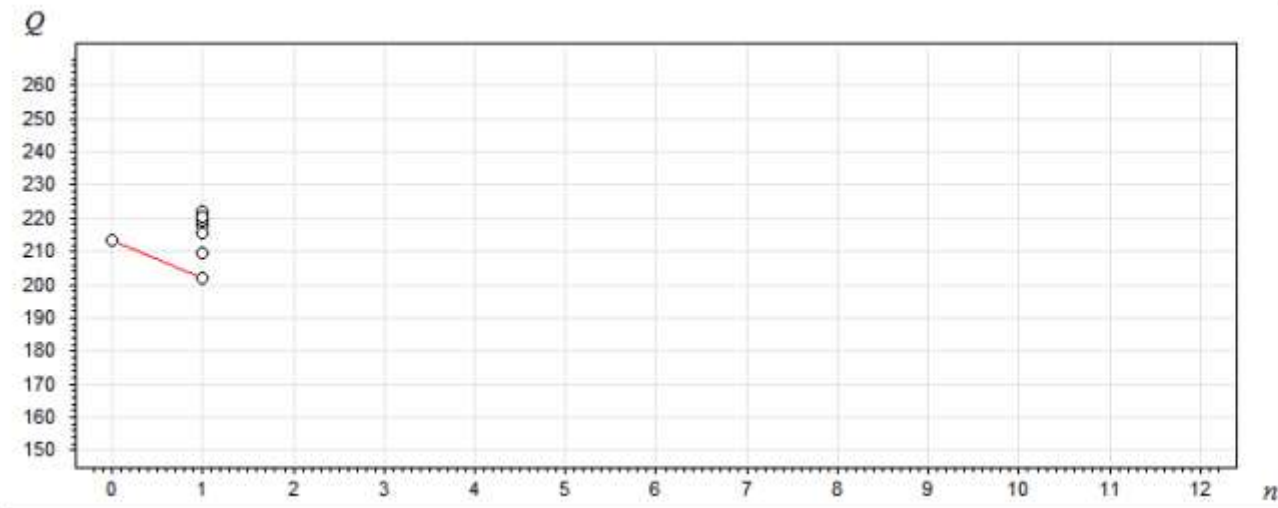
- 1:  $Q^* := Q(\emptyset)$ ; — инициализация;
- 2: **для всех**  $j = 1, \dots, n$ , где  $j$  — сложность наборов:
- 3: найти лучший набор сложности  $j$ :  
 $J_j := \arg \min_{J: |J|=j} Q(J)$ ;
- 4: **если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;
- 5: **если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;



# Методи обгортки. Алгоритм повного перебору - 2

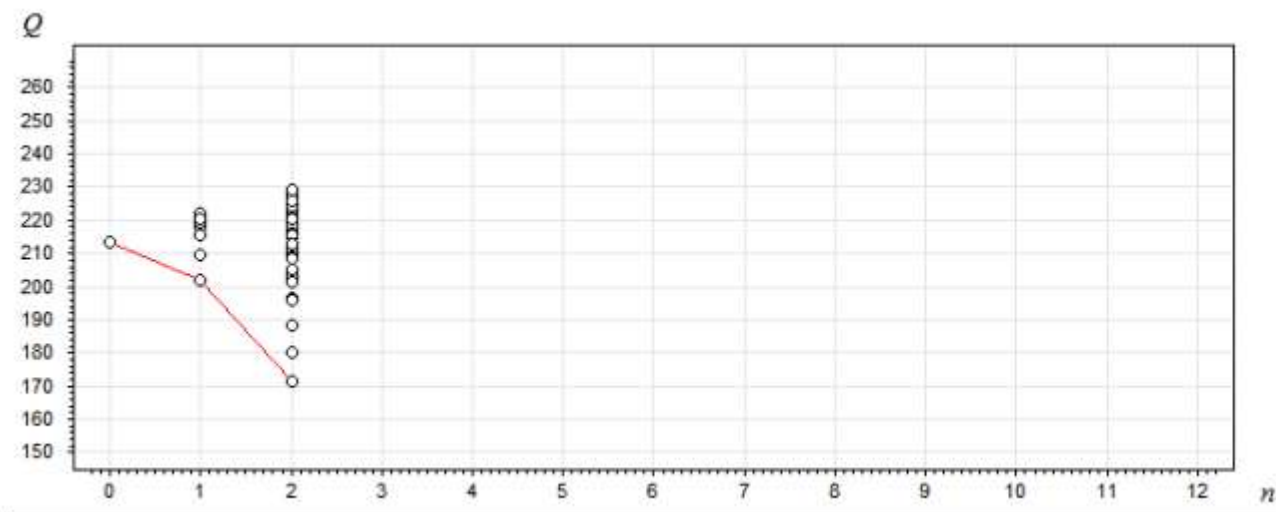


$$d = 3$$
$$j = 0$$



$$d = 3$$
$$j = 1$$
$$j^* = 1$$

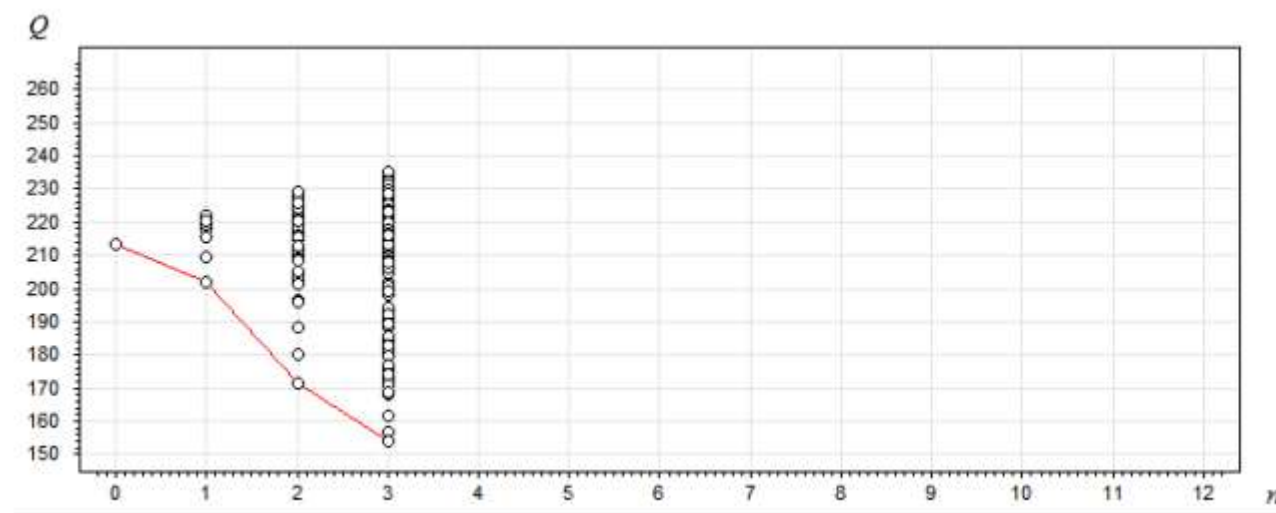
# Методи обгортки. Алгоритм повного перебору - 3



$$d = 3$$

$$j = 2$$

$$j^* = 2$$

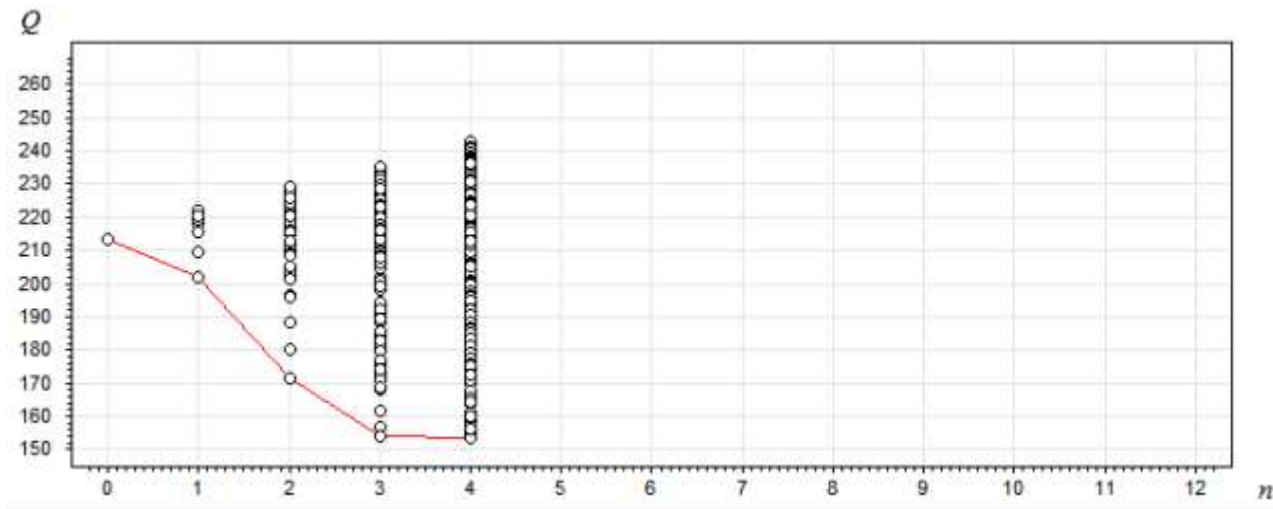


$$d = 3$$

$$j = 3$$

$$j^* = 3$$

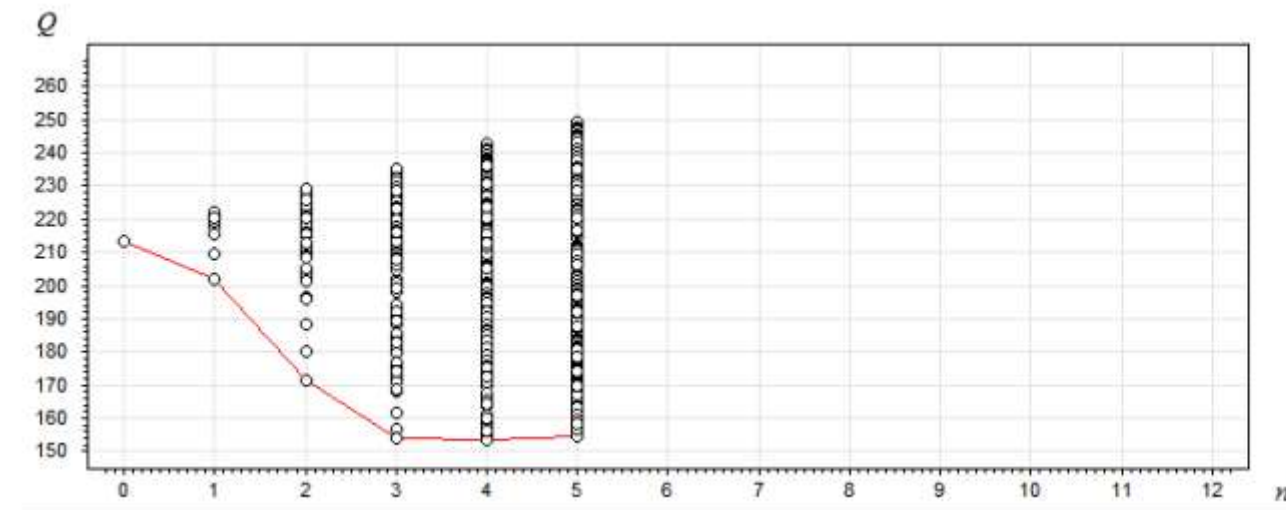
# Методи обгортки. Алгоритм повного перебору - 4



$$d = 3$$

$$j = 4$$

$$j^* = 4$$

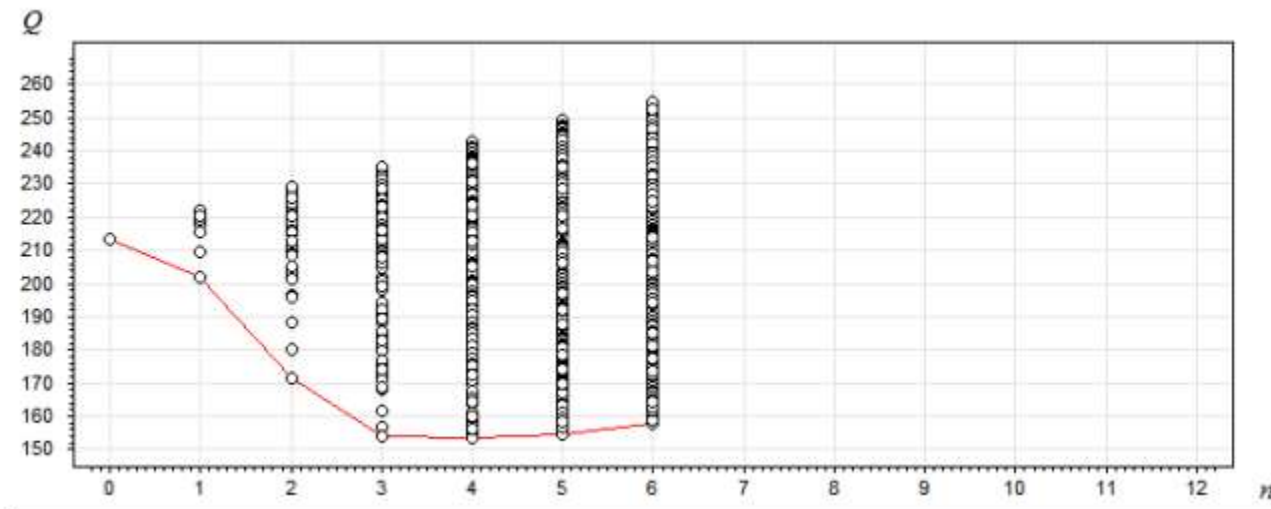


$$d = 3$$

$$j = 5$$

$$j^* = 4$$

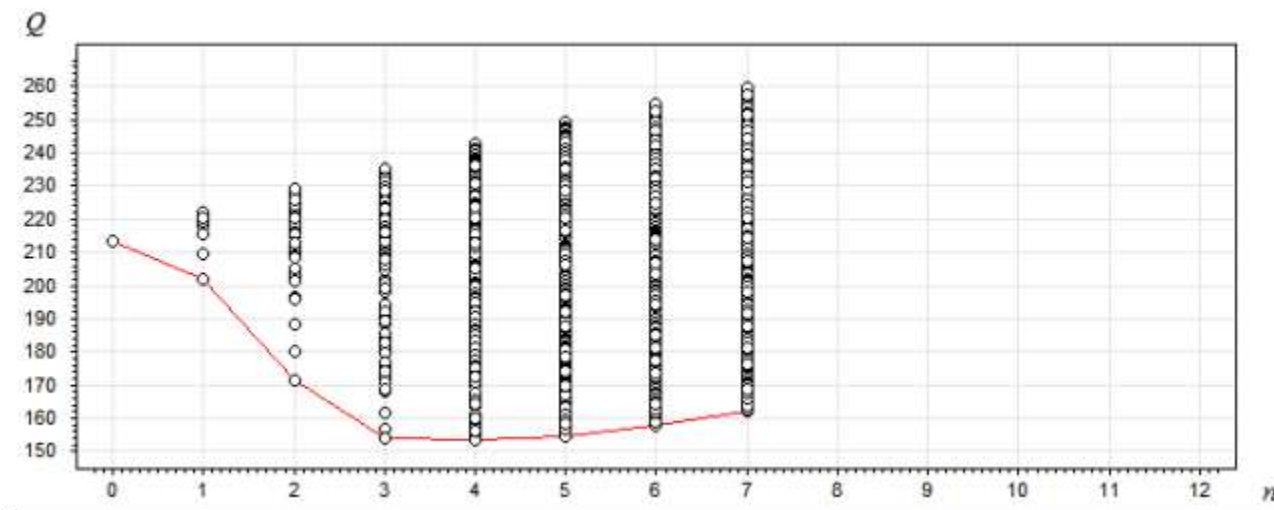
# Методи обгортки. Алгоритм повного перебору - 5



$$d = 3$$

$$j = 6$$

$$j^* = 4$$



$$d = 3$$

$$j = 7$$

$$j^* = 4$$

# Методи обгортки. Алгоритм повного перебору

## ❑ Переваги

- простота реалізації
- **гарантований результат**
- повний перебір є ефективним, коли
  - інформативних ознак небагато (до 5)
  - всього ознак також небагато (до 100)

## ❑ Недоліки:

- в інших випадках – дуже довго  $O(2^n)$
- чим більше варіантів перебирається, тим більше рівень перенавчання

## ❑ Способи зменшення недоліків:

- застосування евристичних методів скорочення перебору

# Use Wrappers to Refine Features

□ *mRMR is a filter approach*

- + Fast
- - Features might not be optimal
- + Independent of the classifier

□ *Wrappers seek to minimize the number of errors directly*

- - Slow
- - Features are less robust
- - Dependent on classifier
- + Better prediction accuracy

□ **Який же метод обрати на практиці?**

# Use Wrappers to Refine Features

□ *mRMR is a filter approach*

- + Fast
- - Features might not be optimal
- + Independent of the classifier

□ *Wrappers seek to minimize the number of errors directly*

- - Slow
- - Features are less robust
- - Dependent on classifier
- + Better prediction accuracy

□ Use mRMR first to generate a short feature pool and use wrappers to get a least redundant feature set with better accuracy



# Підходи до зниження розмірності даних

- Відбір інформативних характеристик (*feature selection*)

*Data space* → *Data subspace*

Відбір деякої підмножини найбільш змістовних характеристик

- ❖ (*filter methods*, методи фільтрації) – методи попарного відбору характеристик
- ❖ (*wrapper methods*, методи обгортки) – методи послідовного відбору характеристик

- **Синтез інформативних характеристик** (*feature extraction*)

*Data space* → *Feature space*

Конструюється простір з новими характеристиками

- **Методи, вбудовані в модель навчання** (*embedded models*), – наприклад: *LASSO* як *feature selection* в лінійній регресії, *ЛДА* як *feature extraction*, *Decision Trees with pruning*

# Синтез інформативних характеристик

- Буває так, що навіть після виключення надлишкових і відкидання нерелевантних ознак залишається занадто велика кількість ознак
- І тоді будь-який метод навчання працює погано, а, з огляду на розмір простору ознак, ми розуміємо, що вдіяти нічого не можна
- Тоді ми приходимо до висновку, що потрібно конструювати (=синтезувати) нові ознаки
- Методи синтезу інформативних ознак змінюють структуру простору ознак, пристосовуючи його до моделі

# Синтез інформативних характеристик

- Синтез (=виділення) інформативних характеристик (feature extraction)

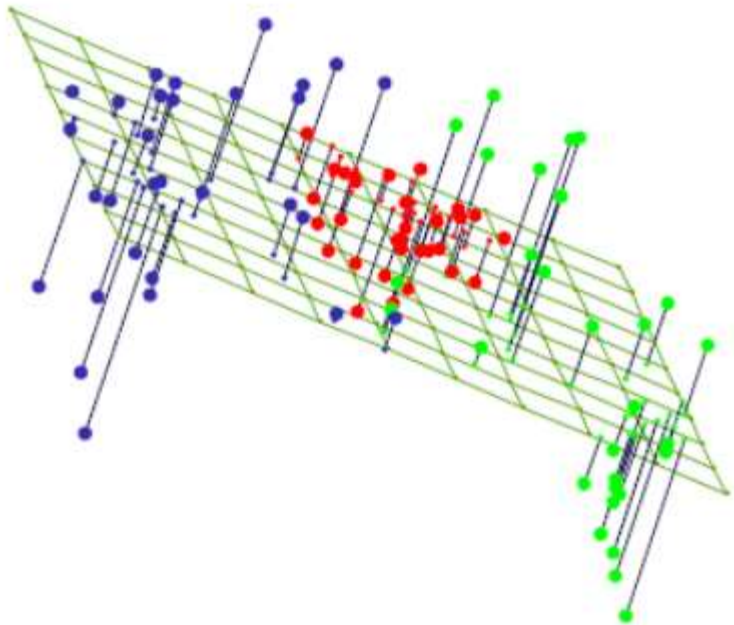
*Data space* → *Feature space*

Конструюється простір з новими характеристиками

- ❖ методи навчання без учителя:
  - метод головних компонент (*principal component analysis, PCA*)
  - метод незалежних компонент (*independent component analysis, ICA*)
- ❖ методи навчання з учителем:
  - лінійний дискримінантний аналіз (*linear discriminant analysis, LDA*)

# Метод головних компонент (principal component analysis, PCA)

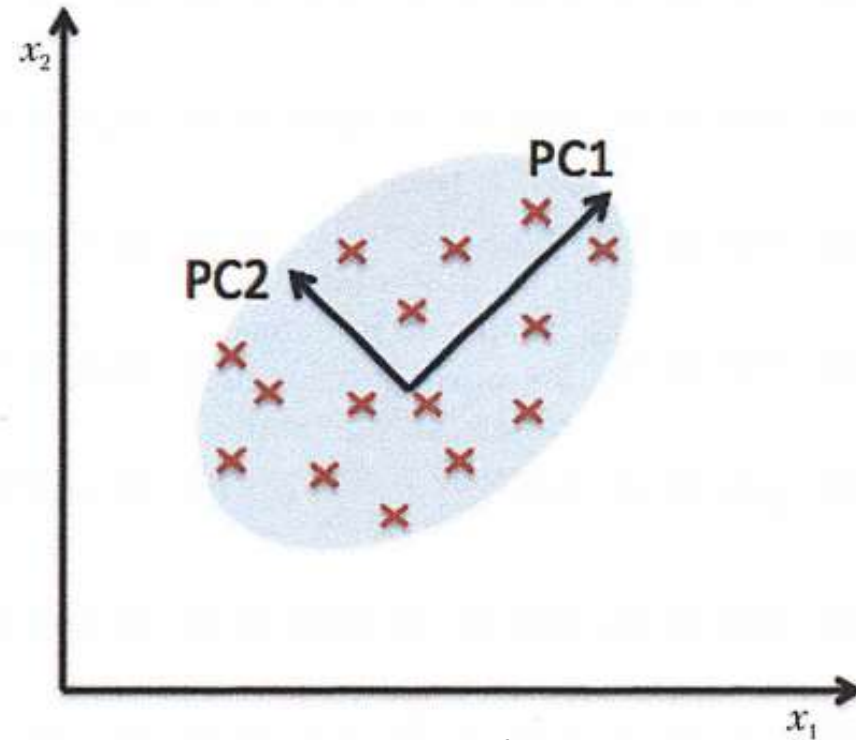
- *РСА (Principal Component Analysis)* – аналіз ГОЛОВНИХ КОМПОНЕНТ
- У теорії інформації відомий також як перетворення Карунена-Лоева



- Суть методу:  
шукаємо гіперплощину заданої розмірності, таку що помилка проектування вибірки на дану гіперплощину була б мінімальною

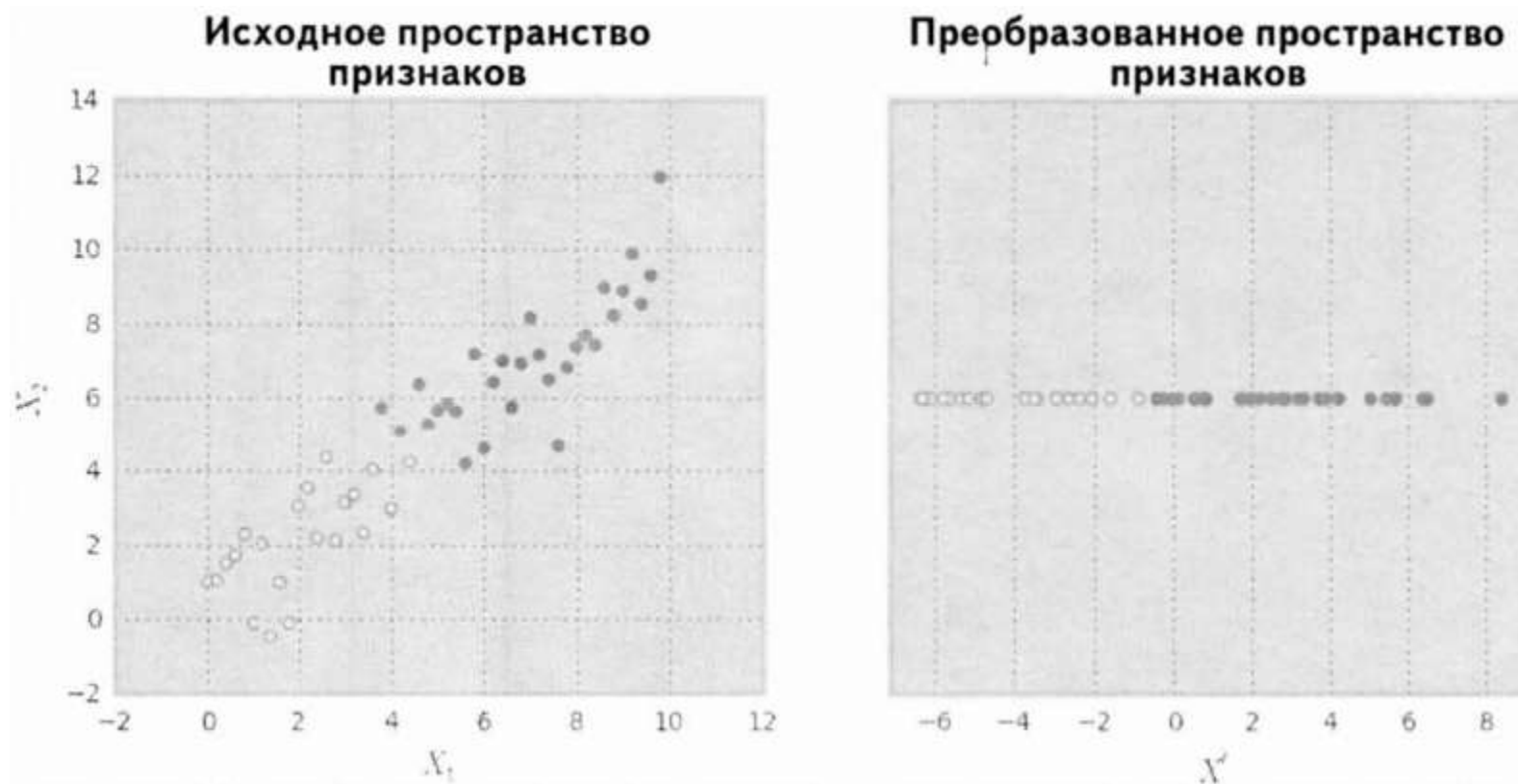
# РСА: вихідні дані та головні компоненти

- $x_1$  і  $x_2$  – це початкові вісі ознак
- PC1 і PC2 – це головні компоненти



# РСА: геометрична ілюстрація

- ПРИКЛАД: розглянемо набір даних з 2 характеристиками, тобто в 2-вимірному просторі



# Алгоритм роботи PCA

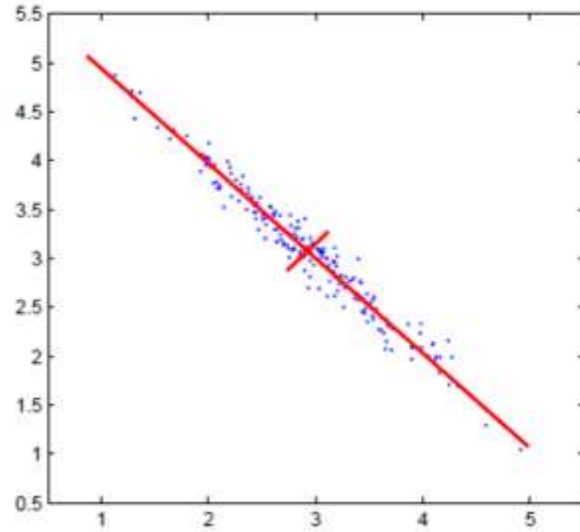
- Алгоритм PCA заснований на лінійній алгебрі:
  1. Центрувати дані, віднявши з кожного елемента середнє
  2. Обчислити коваріаційну матрицю
  3. Обчислити власні вектори і власні числа коваріаційної матриці
  4. Відібрати власні вектори, що відповідають  $K$  найбільшим власним числам (більшим власним числам відповідають вектори з більшою дисперсією)



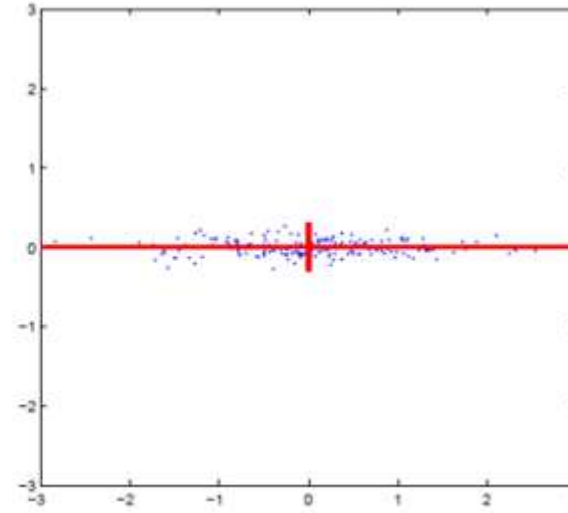
# Алгоритм роботи PCA

- Нехай на початку було  $\mathcal{X} = 1000$  ознак, і ми знаємо, що модель добре працює, тільки якщо ознак не більш 20
- Тоді ми просто вибираємо 20 власних векторів з найбільшими власними значеннями
- Отже, маючи початковий простір ознак, алгоритм PCA знаходить його лінійну проекцію на простір меншої розмірності з такими властивостями:
  - досягається максимум залишкової дисперсії
  - досягається мінімум помилки реконструкції (при спробі повернення від нових ознак до початкових)

# РСА: геометрична ілюстрація



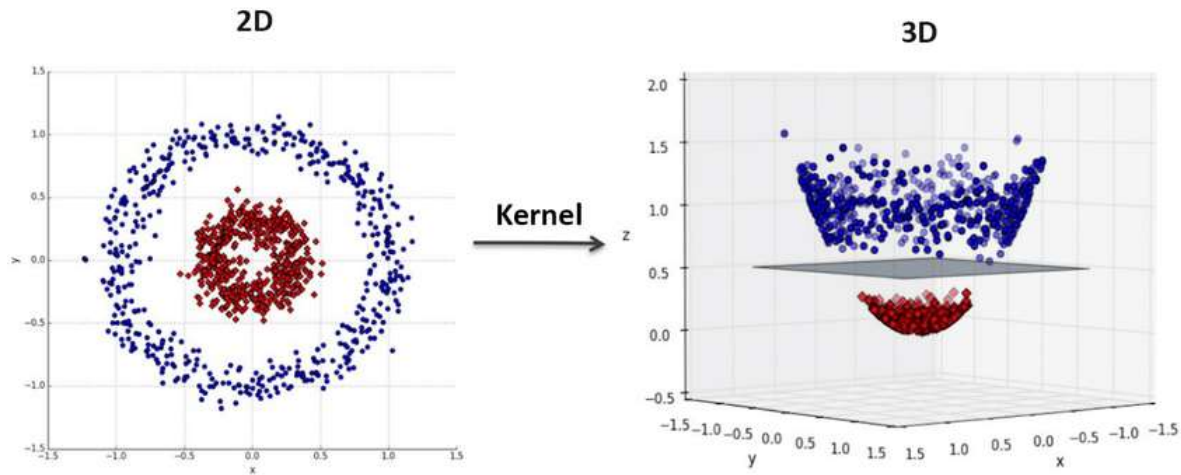
Початковий простір



Остаточний простір

- Пересуваємо початок координат у центр вибірки
- Повертаємо осі так, щоб ознаки не корелювали
- Позбуваємося координат з малою дисперсією

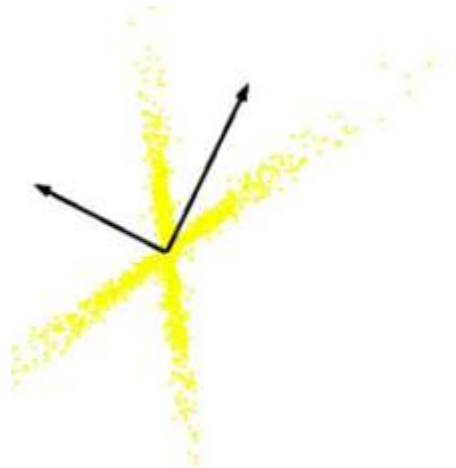
# РСА: переваги та недоліки



- + Алгоритм простий
- + За допомогою "*kernel trick*" адаптується на нелінійний випадок (*Kernel PCA*)
- - Проблема з обчисленням власних векторів коваріаційної матриці в разі великої кількості даних
- - Координати об'єктів у новому просторі визначені неоднозначно

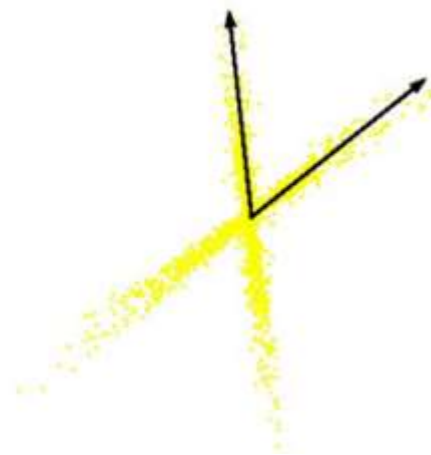
# Independent Component Analysis (метод незалежних компонент)

- Геометрична ілюстрація



PCA

(ортогональні компоненти)

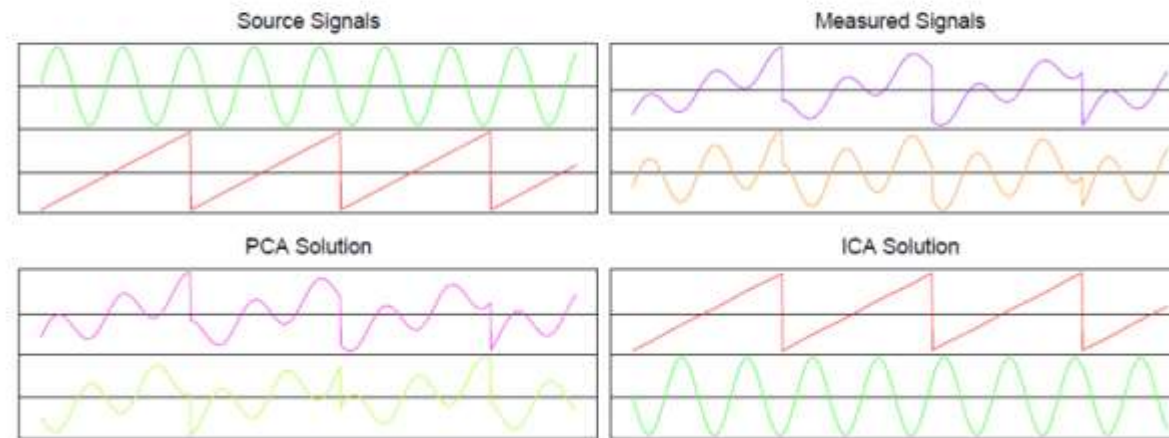


ICA

(неортогональні компоненти)

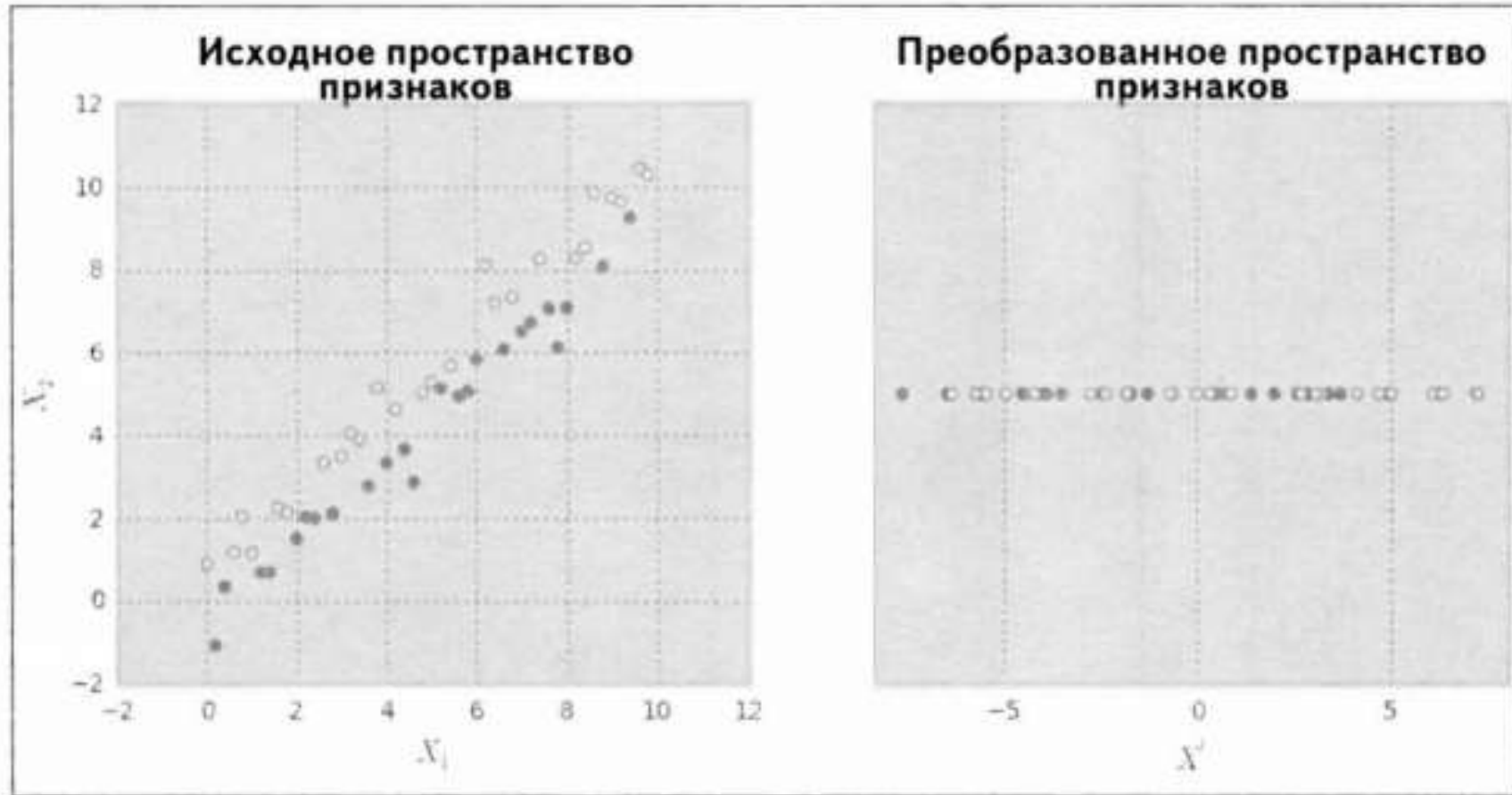
# PCA vs ICA

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$



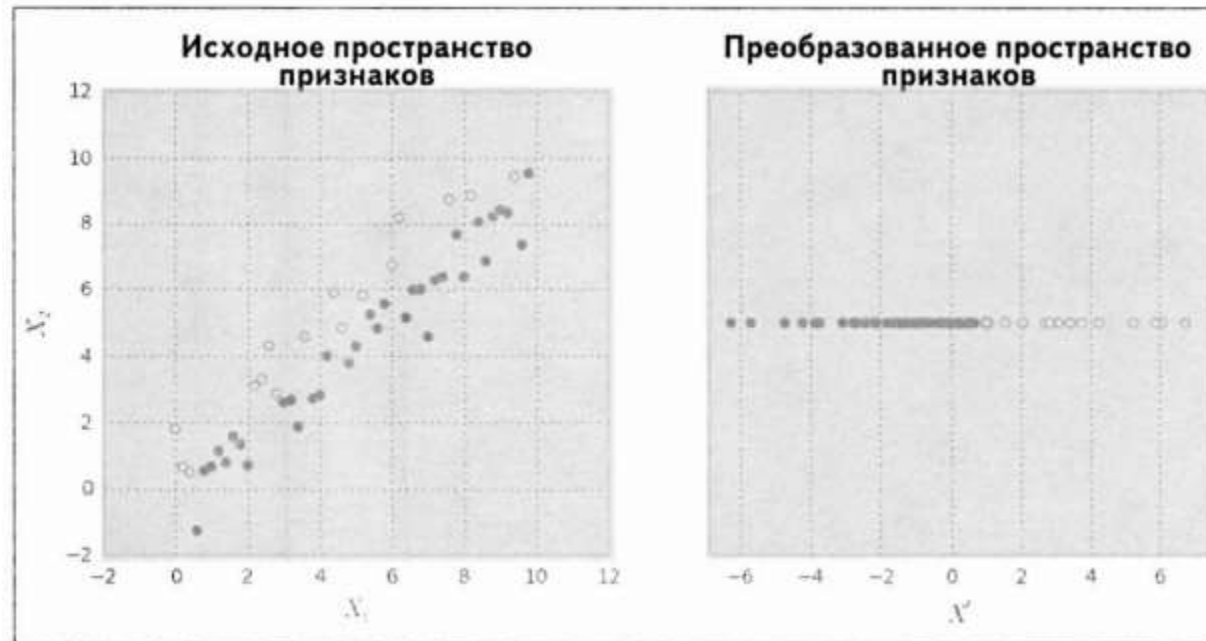
- Порівняння PCA і ICA на штучно згенерованому часовому ряді, змодельованому на 1000 рівномірно розподілених точках

# Проблеми RSA – відсутній учитель



# Перевага LDA – є вчитель!

- Лінійний дискримінантний аналіз (LDA)
- Ідея методу полягає в тому, щоб максимізувати відстані між точками, що належать різним класам, але мінімізувати відстані між точками одного класу





# Навіщо PCA, якщо є LDA?

- Для PCA не потрібен учитель
- Коли число класів зростає, а кількість прикладів кожного класу зменшується, то LDA працює все гірше
- PCA менш чутливий до відмінностей в навчальних даних, ніж LDA

# Підходи до зниження розмірності даних

- Відбір інформативних характеристик (*feature selection*)

*Data space* → *Data subspace*

Відбір деякої підмножини найбільш змістовних характеристик

- ❖ (*filter methods*, методи фільтрації) – методи попарного відбору характеристик
- ❖ (*wrapper methods*, методи обгортки) – методи послідовного відбору характеристик

- Синтез інформативних характеристик (*feature extraction*)

*Data space* → *Feature space*

Конструюється простір з новими характеристиками

- **Методи, вбудовані в модель навчання** (*embedded models*), – наприклад: *LASSO* як *feature selection*, *ЛДА* як *feature extraction*, *Decision Trees with pruning*

# Регуляризація в лінійних моделях

$$\mathcal{L}(X, \vec{y}, \vec{w}) = \frac{1}{2n} \|\vec{y} - X\vec{w}\|_2^2 + \alpha \|\vec{w}\|^p$$

- В загальному випадку **ступінь регуляризатора  $p$**  визначає клас методів оптимізації:
- ❖ при  $p=2$  і гладкому функціоналі  $\mathcal{L}(w)$  можна застосовувати стандартні градієнтні методи мінімізації (**гребенева регуляризація**)
- ❖ при  $p=1$  і опуклому функціоналі  $\mathcal{L}(w)$  маємо задачу опуклого програмування з обмеженнями типу нерівностей; в результаті її розв'язання частина коефіцієнтів  $w_j$  **обнуляються**, що фактично означає відсів неінформативних характеристик (**LASSO**)  
least absolute shrinkage and selection operator
- Якщо є обидва доданки: з  $p=1$  ( $\mathcal{L}1$  штраф) і з  $p=2$  ( $\mathcal{L}2$  штраф), то побудована модель називається **еластичною мережею**