



# Intelligenza Artificiale e al Machine Learning

**Esercizi di classificazione - Soluzioni**

# Pre processing bank data

- Il dataset bank-data.arff

- ✓ 600 istanze
- ✓ Nessun dato missing

Attributo	Descrizione
<b>Id</b>	Identificatore unico
<b>Age</b>	età del cliente in anni (numeric)
<b>Sex</b>	MALE / FEMALE
<b>Region</b>	inner_city/rural/suburban/town
<b>Income</b>	reddito del cliente (numeric)
<b>children</b>	sposato? (YES/NO)
<b>car</b>	possiede un'automobile? (YES/NO)
<b>save_acct</b>	ha un conto di risparmio? (YES/NO)
<b>current_acct</b>	ha un conto corrente? (YES/NO)
<b>Mortgage</b>	ha un mutuo (YES/NO)
<b>pep</b>	<b>ha acquistato un PEP (Personal Equity Plan) dopo l'ultimo invio postale? (YES/NO)</b>



# Pre processing bank data

1. Caricare il file e salvarlo in formato ARFF con il nome “bank data.arff”
2. Effettuare un’analisi manuale dei dati mediante visualizzazione
  - ✓ Istogrammi attributo – attributo (da pagina Visualize)
  - ✓ Distribuzioni attributo – attributo – classe (da pagina Visualize)
    - Commentare la distribuzione Income –Children - PEP
3. Eliminare l’attributo ID e salvare nuovamente con il nome “bank data.arff”
4. Discretizzare l’attributo AGE (Equal- frequency 10 bin) e Children (Manualmente) e salvare il dataset con il nome “bank data1.arff”

# Classificazione bank data

- Utilizzare i seguenti algoritmi per eseguire la classificazione e commentare il risultato
  - ✓ Valutare il risultato con Cross-validation 10 folds / Use training set / Percentage split
  - ✓ Utilizzare sia il data set discretizzato (bank data1.arff), sia quello non discretizzato (bank data.arff)
- 1. J48
  - Rappresentare e discutere i decision boundary
- 2. J48 senza post-pruning (unpruned = True)
  - Visualizzare l'albero di decisione
- 3. IBk con k=1 e k=5 sul data set non discretizzato (BankData.arff)
  - Dopo aver normalizzato i rimanenti attributi numerici
- 4. Ripetere la classificazione utilizzando BankData1.arff dopo aver discretizzato anche l'attributo income (equal frequency 10 bins)
  - Salvare il training set nel file BankData2.arff

# Classificazione bank data

## Bank data1.arff: Discretizzato

J48 – Training set 91.5%

a b <-- classified as  
239 35 | a = YES  
16 310 | b = NO

J48 – Cross val 91.3%

a b <-- classified as  
239 35 | a = YES  
17 309 | b = NO

J48 – Split 88.2%

a b <-- classified as  
81 15 | a = YES  
9 99 | b = NO

## Bank data.arff: Non discretizzato

J48 – Training set 92.3%

a b <-- classified as  
245 29 | a = YES  
17 309 | b = NO

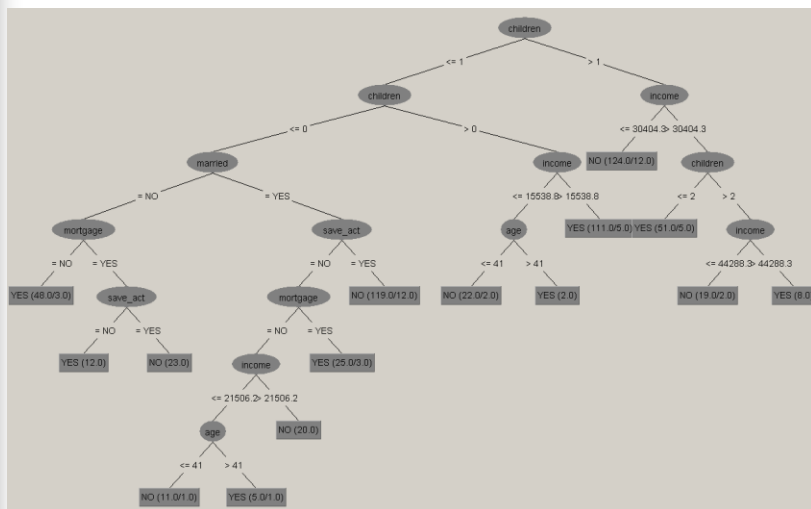
J48 – Cross val 89.8%

a b <-- classified as  
236 38 | a = YES  
23 303 | b = NO

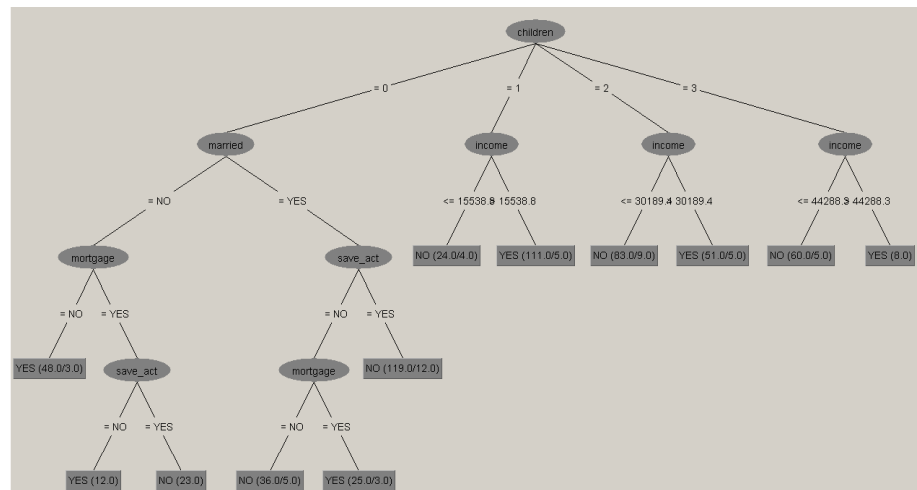
J48 – Split 86.8%

a b <-- classified as  
81 15 | a = YES  
12 96 | b = NO

## Bank data.arff: Non discretizzato + Split



## Bank data1.arff: Discretizzato + Split



# Classificazione bank data

## Bank data1.arff: Discretizzato + No pruning

J48 – Training set 95.2%

J48 – Cross val 89.5%

J48 – Split 85.3%

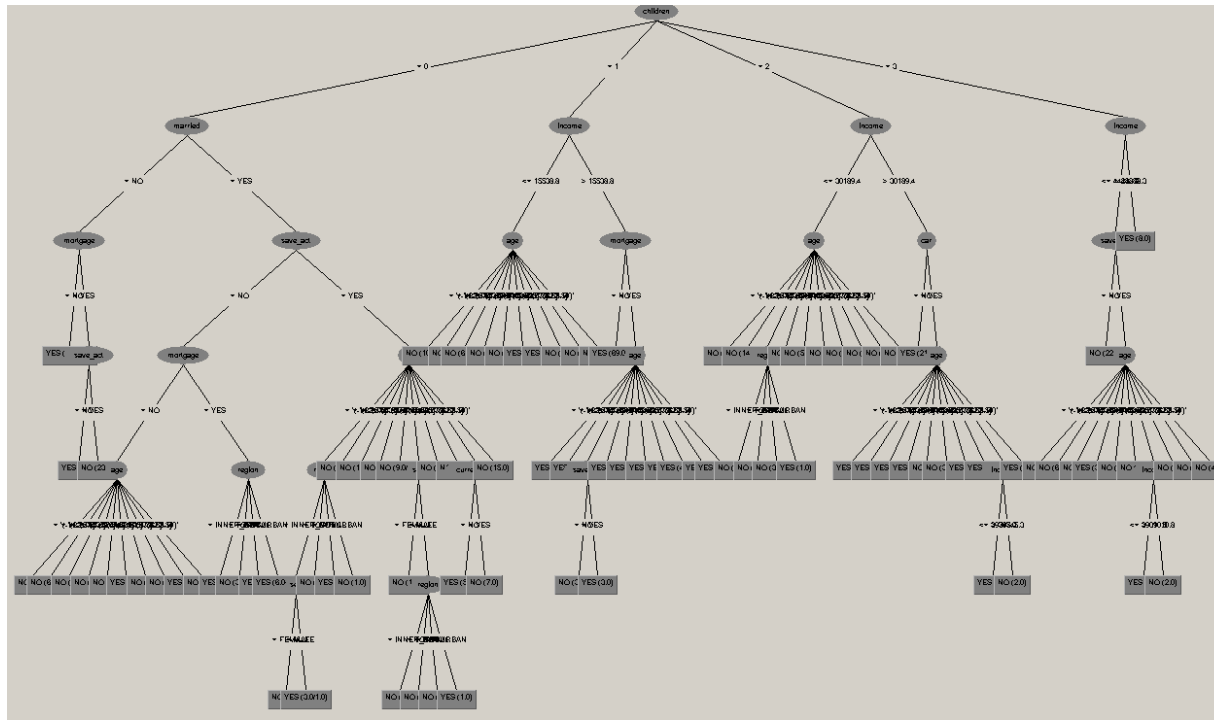
a b <-- classified as  
255 19 | a = YES  
10 316 | b = NO

a b <-- classified as  
241 33 | a = YES  
30 296 | b = NO

a b <-- classified as  
81 15 | a = YES  
15 93 | b = NO

**Migliora la classificazione sul training set, peggiora negli altri casi  
Fatica a generalizzare! → Overfitting**

## Bank data1.arff: Discretizzato + Split + No pruning



# Classificazione bank data

## Bank data.arff2: Normalizzato + Discretizzato

### IB1– Training set 100%

```
a b <-- classified as
274 0 | a = YES
0 326 | b = NO
```

### IB1– Cross val 74.7%

```
a b <-- classified as
209 65 | a = YES
87 239 | b = NO
```

### IB1– Split 70.6%

```
a b <-- classified as
76 20 | a = YES
40 68 | b = NO
```

## Bank data.arff: Normalizzato + Non discretizzato

### IB1– Training set 100%

```
a b <-- classified as
274 0 | a = YES
0 326 | b = NO
```

### IB1– Cross val 64.2%

```
a b <-- classified as
171 103 | a = YES
112 214 | b = NO
```

### IB1– Split 61.3%

```
a b <-- classified as
59 37 | a = YES
42 66 | b = NO
```

## Bank data.arff2: Normalizzato + Discretizzato

### IB5 – Training set 82.3%

```
a b <-- classified as
217 57 | a = YES
46 280 | b = NO
```

### IB5– Cross val 76.3%

```
a b <-- classified as
195 79 | a = YES
63 263 | b = NO
```

### IB5– Split 75.5%

```
a b <-- classified as
73 23 | a = YES
27 81 | b = NO
```

## Bank data.arff: Normalizzato + Non discretizzato

### IB5 – Training set 78.2%

```
a b <-- classified as
203 71 | a = YES
60 266 | b = NO
```

### IB5 – Cross val 65.2%

```
a b <-- classified as
162 112 | a = YES
97 229 | b = NO
```

### IB5 – Split 60.3%

```
a b <-- classified as
51 45 | a = YES
36 72 | b = NO
```

# Bank data: alcune considerazioni

- Le tecniche k-nearest neighbors non sembrano fornire buoni risultati
  - ✓ Rumore nei dati? (l'utilizzo di un  $k=5$  aumenta di poco l'efficacia)
  - ✓ Training set limitato?
  - ✓ Funzioni distanza non appropriate? (l'algoritmo non riesce a sfruttare gli attributi numerici)
  - ✓ **Presenza di attributi ridondanti o non rilevanti?**
- E' possibile fare una verifica...
  - ✓ Selezionando un sottoinsieme di attributi
  - ✓ Eseguendo su di esso la classificazione
    - Sul training set che ha dato i risultati migliori bank data1.arff (senza campo id con children e age discretizzati) si esegue la selezione degli attributi (CfsSubsetEval) che restituisce:  
**income + married + children**
    - Si esegue la classificazione sulla versione normalizzata di questi attributi
    - In alternativa si possono utilizzare i primi attributi scelti dall'albero decisionale che ha fornito i risultati migliori (ossia quello basato bank data1.arff):  
**children+income+married+mortgage+save act**



# Classificazione bank data

## bank data1.arff: income + married + children

IB5– Training set 86.5%

```
a b <-- classified as
223 51 | a = YES
30 296 | b = NO
```

IB5– Cross val 81.5%

```
a b <-- classified as
212 62 | a = YES
49 277 | b = NO
```

IB5– Split 88.9%

```
a b <-- classified as
75 21 | a = YES
18 90 | b = NO
```

## bank data1.arff: children + income + married + mortgage + save act

IB5– Training set 90.3%

```
a b <-- classified as
242 32 | a = YES
26 300 | b = NO
```

IB5– Cross val 87.0%

```
a b <-- classified as
232 42 | a = YES
36 290 | b = NO
```

IB5– Split 84.8%

```
a b <-- classified as
78 18 | a = YES
13 95 | b = NO
```

# Dati di censimento

- Il dataset CensusTraining.arff riporta dati del censimento USA (<http://cps.ipums.org/>)
  - ✓ 1000 istanze per il training
  - ✓ 31561 istanze per la validazione

Attributo	Descrizione
age	Età in anni
workclass	Classe di lavoro
fnlwgt	“Final sampling weight” peso dell’istanza (campione) rispetto alla popolazione
education	Titolo ottenuto
education-num	Numero di anni di studio
marital-status	Stato civile
occupation	Occupazione
relationship	Tipo di relazione con il capo famiglia
race	Razza
sex	Sesso
capital-gain	Utile da capitali (plus valenza)
capital-loss	Perdite da capitali (minus valenza)
hours-per-week	Ore di lavoro settimanali
native-country	Nazionalità
<b>Total Income</b>	<b>L’individuo guadagna più o meno di 50K\$</b>



# Dati di censimento

- Obiettivo dello studio è trovare un modello che permetta di predire quali persone guadagnano più di 50K€
  - ✓ Ricerca di frodi fiscali
- Si proceda utilizzando la metodologia CRISP-DM
  1. Comprensione del dominio applicativo
  2. Comprensione dei dati
  3. Preparazione dei dati
  4. Creazione del modello
  5. Valutazione del modello e dei risultati
  6. *Deployment*



# Dati di censimento

- C'è una forte correlazione tra i campi education ed education num
- Il campo fnlwgt sembra poco rilevante ai fini della determinazione della classe
- L'attributo native country è fortemente sbilanciato sul valore United States
  - ✓ Le selezioni fatte utilizzando questo attributo saranno sempre poco significative
- Alcuni campi presentano valori nulli ('?')
  - ✓ E' necessario valutare se calcolarli sostituendoli con media e moda
- Tra i campi che “a vista” meglio si prestano a fornire informazioni utili alla classificazione
  - ✓ Relationship, Race, Sex, Education – Education num
  - ✓ Il profilo di chi guadagna >50k\$ è maschio, bianco, sposato con una educazione di livello superiore.
  - ✓ Dalla prima analisi non è possibile evincere se tra questi attributi esistano delle correlazioni e quindi debbano essere usati in alternativa



# Dati di censimento

- Si comparino le performance degli algoritmi studiati (J48, JRip e IBK) utilizzando come training set i file
  - ✓ [AdultTraining.arff](#)
  - ✓ [AdultTrainingSmall.arff](#)
- In entrambi i casi si utilizzi come modalità di validazione il test set
  - ✓ [AdultTest.arff](#)
- Si verifichi in seguito come cambiano i risultati ottenuti dai due data set utilizzando le tecniche
  - ✓ [AdaBoost](#)
  - ✓ [Bagging](#)
- Si commentino i risultati

# Dati di censimento

- L'utilità di calcolare i dati mancanti è verificata applicando gli algoritmi ai dati grezzi e ai dati modificati tramite il filtro ReplaceMissingValues

## CensusTraining.arff

J48– Supplied TS 82.7%

```
a  b  <-- classified as
22571 1381 | a = <=50K
4082 3527 | b = >50K
```

JRip– Supplied TS 82.6%

```
a  b  <-- classified as
21990 1962 | a = <=50K
3544 4065 | b = >50K
```

IB5– Supplied TS 80.7%

```
a  b  <-- classified as
21430 2522 | a = <=50K
3575 4034 | b = >50K
```

## CensusTrainingNoMissing.arff

J48– Supplied TS 82.3%

```
a  b  <-- classified as
22664 1288 | a = <=50K
4304 3305 | b = >50K
```

JRip– Supplied TS 83.0%

```
a  b  <-- classified as
21561 2391 | a = <=50K
2960 4649 | b = >50K
```

IB5– Supplied TS 80.1%

```
a  b  <-- classified as
19613 2340 | a = <=50K
3486 3801 | b = >50K
```

- Si decide di mantenere i dati originali

# Dati di censimento

- Le performance degli algoritmi sul training set ridotto sono:

## CensusTrainingSmall.arff

J48– Supplied TS 75.9%

```
a  b  <-- classified as
23952  0 |  a = <=50K
7609   0 |  b = >50K
```

JRip– Supplied TS 75.9%

```
a  b  <-- classified as
23952  0 |  a = <=50K
7609   0 |  b = >50K
```

IB5– Supplied TS 76.7%

```
a  b  <-- classified as
22224 1728 |  a = <=50K
5637  1972 |  b = >50K
```

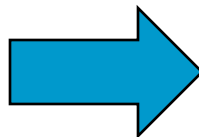
- Riducendo il numero di attributi considerati sulla base del filtro CfsSubsetEval la performance dell'algoritmo di k-nearest neighbor migliora leggermente

✓ Age + education-num + marital-status + relationship + capital gain + capital loss + hours-per-week

## CensusTraining.arff

IB5– Supplied TS 80.7%

```
a  b  <-- classified as
21430 2522 |  a = <=50K
3575  4034 |  b = >50K
```



IB5– Supplied TS 80.9%

```
a  b  <-- classified as
21818 2134 |  a = <=50K
3904  3705 |  b = >50K
```

- Cosa succede discretizzando gli attributi numerici?

# Applicazione: marketing lift

- Una ditta di collocamento possiede una banca dati contenente le informazioni (classe esclusa) relative a 50000 infermiere. Si vogliono contattare tramite posta tutte le infermiere appartenenti alla classe “**priority**”. Tali infermiere saranno tutte inviate ad un insieme di ospedali che hanno fatto richiesta di nuove infermiere.
- L’operazione di invio delle lettere ha un costo fisso di 10,000 € e un costo individuale (per ogni infermiera contattata) pari a 5 €.
- Gli ospedali prenderanno in prova le infermiere selezionate dalla ditta e alla fine del periodo di prova pagheranno alla ditta 10 € per ogni infermiera che risulterà essere effettivamente una infermiera classificabile come appartenente alla classe “priority”. Decidere quale modello, tra quelli studiati permette di ottenere potenzialmente il profitto maggiore e quante infermiere (in percentuale) devono essere contattate.
- Verificare quale delle tecniche di classificazione studiate fornisce il modello che fornisce il **lift** maggiore
  - ✓ Dataset di training: [nurseryTrain.arff](#)
  - ✓ Tecnica di validazione 10 folds cross-validation



# Applicazione: marketing lift

- Il **lift** è un indicatore che permette di comparare più modelli di classificazione favorendo quelli che permettono di individuare un campione distorto della popolazione che massimizzi la probabilità di trovare istanze della classe desiderata  $C_i$

$$\text{Lift}(\text{Modello}X) = \frac{P(C_i | \text{Campione})}{P(C_i | \text{Popolazione})}$$

- Molto utilizzato in ambito marketing per selezionare i clienti su cui operare (campagne focalizzate).
  - ✓ La classe desiderata è quella degli utenti che risponderanno positivamente all'attività di marketing
- Calcolare il lift e il guadagno effettivo per J48 JRIP e IB1

# Applicazione: marketing lift

## ■ J48

a	b	c	d	e	<-- classified as
3333	0	0	0	0	a = not_recom
0	0	2	0	0	b = recommend
0	0	239	89	0	c = very_recom
0	0	63	3785	103	d = priority
0	0	0	54	2332	e = spec_prior

- Error rate = 3.11%
- % TP stimata della popolazione = 39.51%
- % del campione rispetto alla popolazione =  $3,928 / 10,000 = 39.28\%$
- % TP stimata del campione = 96.36%
- $\text{Lift(J48)} = (3,785 / 3,928) / (3,951 / 10,000) = 0.9636 / 0.3951 = 2.4389$
- Costo su popolazione =  $-10,000\text{€} - 50,000 \times 5\text{€} + 50,000 \times 39.51\% \times 10\text{€} = -260,000\text{€} + 197,550\text{€} = -62,450\text{€}$
- Costo su campione =  $-10,000\text{€} - 50,000 \times 39.28\% \times 5\text{€} + 50,000 \times 39.28\% \times 96.36\% \times 10 = -108,200\text{€} + 189,251\text{€} = +81,051\text{€}$

# Applicazione: marketing lift

## ■ JRIP

a	b	c	d	e	<-- classified as
3333	0	0	0	0	a = not_recom
0	0	2	0	0	b = recommend
0	0	214	114	0	c = very_recom
0	1	124	3719	107	d = priority
0	0	0	35	2351	e = spec_prior

- Error rate = 3.81%
- % TP stimata della popolazione = 39.51%
- % del campione rispetto alla popolazione =  $3,868 / 10,000 = 38.68\%$
- % TP stimata del campione = 96.15%
- $\text{Lift(JRIP)} = (3,719 / 3,868) / (3,951 / 10,000) = 0.9615 / 0.3951 = 2.4336$
- Costo su popolazione =  $-10,000\text{€} - 50,000 \times 5\text{€} + 50,000 \times 39.51\% \times 10\text{€} = -260,000\text{€} + 197,550\text{€} = -62,450\text{€}$
- Costo su campione =  $-10,000\text{€} - 50,000 \times 38.68\% \times 5\text{€} + 50,000 \times 38.68\% \times 96.15\% \times 10 = -108,200\text{€} + 185,954\text{€} = +77,754\text{€}$

# Applicazione: marketing lift

## ■ IB1

a	b	c	d	e	<-- classified as
3333	0	0	0	0	a = not_recom
0	0	2	0	0	b = recommend
0	0	194	134	0	c = very_recom
0	0	0	3916	35	d = priority
0	0	0	89	2297	e = spec_prior

- Error rate = 2.60%
- % TP stimata della popolazione = 39.51%
- % del campione rispetto alla popolazione =  $4,139 / 10,000 = 41.39\%$
- % TP stimata del campione = 94.61%
- $\text{Lift}(\text{IB1}) = (3,916 / 4,139) / (3,951 / 10,000) = 0.9461 / 0.3951 = 2.3946$
- Costo su popolazione =  $-10,000\text{€} - 50,000 \times 5\text{€} + 50,000 \times 39.51\% \times 10\text{€} = -260,000\text{€} + 197,550\text{€} = -62,450\text{€}$
- Costo su campione =  $-10,000\text{€} - 50,000 \times 41.39\% \times 5\text{€} + 50,000 \times 41.39\% \times 94.61\% \times 10 = -113,475\text{€} + 195,795\text{€} = +82,320\text{€}$



# Considerazioni

- La tecnica di classificazione più idonea sembra essere quella dei k-mediani
- Le performance degli alberi decisionali non aumentano utilizzando tecniche di boosting
- Il risultato non dipende dalla componente variance del dataset ma piuttosto dalla componente bias
  - ✓ I decision boundary di questo data set sono difficilmente modellabili tramite segmenti paralleli agli assi
- Per il problema del lift non è detto che il classificatore migliore sia quello con accuracy o error rate minimi visto che siamo interessati agli errori commessi per una particolare classe
  - ✓ Classificazione cost based con costi più elevati per gli errori di interesse

# Applicazione: marketing lift

## ■ Cost SensitiveClassifier +J48

Cost Matrix					a	b	c	d	e	<-- classified as	
0	1	1	1	1	3333	0	0	0	0		a = not_recom
1	0	1	1	1	0	0	2	0	0		b = recommend
1	1	0	10	1	0	0	308	20	0		c = very_recom
1	1	1	0	1	0	0	161	3642	148		d = priority
1	1	1	10	0	0	0	0	11	2375		e = spec_prior

- Error rate = 3.42%
- % TP stimata della popolazione = 39.51%
- % del campione rispetto alla popolazione =  $3,673 / 10,000 = 36.73\%$
- % TP stimata del campione = 99.16%
- $\text{Lift}(\text{CSC}+\text{J48}) = (3,642/3,673) / (3,951/10,000) = 0.9916/0.3951 = 2.5096$
- Costo su popolazione =  $-10,000\text{€} - 50,000 \times 5\text{€} + 50,000 \times 39.51\% \times 10\text{€} = -260,000\text{€} + 197,550\text{€} = -62,450\text{€}$
- Costo su campione =  $-10,000\text{€} - 50,000 \times 36.73\% \times 5\text{€} + 50,000 \times 36.73\% \times 99.16\% \times 10 = -101,825\text{€} + 182,107\text{€} = +80,282\text{€}$

**Gli errori si riducono, ma la riduzione del campione controbilancia la maggior precisione**