UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 15 - Qualità dei Dati e Feature Engineering

Qualità dei Dati e Feature Engineering

Dati, dati, dati…

Come detto altre volte, *senza dati non si cantano messe*! L'intelligenza artificiale in particolare è una scienza che richiede un ampio utilizzo di dati.

Il problema diventa ancora più evidente quando parliamo di algoritmi di apprendimento: apprendimento significa esperienza, ma quale esperienza può un algoritmo acquisire senza dati?

> **Ingegneria dei dati:** L'insieme delle tecniche e degli algoritmi che consentono l'estrazione, l'analisi e la preparazione di dati che siano fruibili da altre tecniche o algoritmi di data analytics.

L'ingegneria dei dati è per l'intelligenza artificiale ciò che l'esame di algoritmi è per l'esame di fondamenti di intelligenza artificiale —> se non sapessimo come trattare i dati, non potremmo creare alcuna pipeline di machine learning affidabile!

> **Qualità dei dati:** Descrive l'accuratezza, la completezza e la consistenza dei dati.

Avere dati di qualità è l'unico modo di costruire strumenti di intelligenza artificiale capaci di assistere gli utenti nel processo di decision making (altrimenti, le decisioni sarebbero basate su dati non affidabili).

> **Data governance:** Gestisce la disponibilità, usabilità, integrità e sicurezza dei dati. E' basata su standard interni ad un organizzazione o politiche che ne regolano l'utilizzo.

Qualità dei Dati e Feature Engineering

Dati, dati, dati… non sono solo nella nostra mente!

Ad alcuni di voi potrebbe sembrare che il voler ingegnerizzare qualsiasi cosa, compresi i dati, sia un esercizio oltremodo esagerato e/o puramente accademico.

Sfortunatamente, non è così. (S)Fortunatamente, avere un processo ingegnerizzato evita o, almeno, mitiga alcune criticità che hanno delle conseguenze disastrose!

Per comprendere la praticità del problema, facciamo un esempio.

Supponiamo di voler creare un algoritmo di apprendimento capace di predire se una persona è affetta da polmonite.

Ha la polmonite	Età	Peso	Sesso	…	Ha preso antibiotico
NO	65	100	F		NO
NO	70	130	M		NO
NO	34	88	M		NO
SI	22	99	M		SI
NO	78	58	F		NO

Qualità dei Dati e Feature Engineering

Dati, dati, dati… non sono solo nella nostra mente!

Ad alcuni di voi potrebbe sembrare che il voler ingegnerizzare qualsiasi cosa, compresi i dati, sia un esercizio oltremodo esagerato e/o puramente accademico.

Sfortunatamente, non è così. (S)Fortunatamente, avere un processo ingegnerizzato evita o, almeno, mitiga alcune criticità che hanno delle conseguenze disastrose!

Per comprendere la praticità del problema, facciamo un esempio.

Supponiamo di voler creare un algoritmo di apprendimento capace di predire se una per

Variabili indipendenti, ovvero le caratteristiche che usiamo per la predizione

Variabile dipendente, ovvero ciò che vogliamo predire.

Ha la polmonite	Età	Peso	Sesso		Ha preso antibiotico
		100	F		NO
		130	M		NO
NO	34	88	M	…	NO
SI	22	99	M		SI
NO	78	58	F		NO

Qualità dei Dati e Feature Engineering

Dati, dati, dati… non sono solo nella nostra mente!

Ad alcuni di voi potrebbe sembrare che il voler ingegnerizzare qualsiasi cosa, compresi i dati, sia un esercizio oltremodo esagerato e/o puramente accademico.

Sfortunatamente, non è così. (S)Fortunatamente, avere un processo ingegnerizzato evita o, almeno, mitiga alcune criticità che hanno delle conseguenze disastrose!

Per comprendere la praticità del problema, facciamo un esempio.

Supponiamo di voler creare un algoritmo di apprendimento capace di predire se una persona è affetta da polmonite.

Ha la polmonite	Età	Peso	Sesso	…	Ha preso antibiotico
NO	65	100	F		NO
NO	70	130	M		NO
NO	34	88	M		NO
SI	22	99	M		SI
NO	78	58			

Dov’è il problema?

Qualità dei Dati e Feature Engineering

Dati, dati, dati… non sono solo nella nostra mente!

Ad alcuni di voi potrebbe sembrare che il voler ingegnerizzare qualsiasi cosa, compresi i dati, sia un esercizio oltremodo esagerato e/o puramente accademico.

Sfortunatamente, non è così. (S)Fortunatamente, avere un processo ingegnerizzato evita o, almeno, mitiga alcune criticità che hanno delle conseguenze disastrose!

Per comprendere la praticità del problema, facciamo un esempio.

Supponiamo di voler creare un algoritmo di apprendimento capace di predire se una persona è affetta da polmonite.

Ha la polmonite	Età	Peso	Sesso		Ha preso antibiotico
NO	65				NO
NO	70				NO
NO	34	88	M	…	NO
SI	22	99	M		SI
NO	78	58			

Tipicamente, una persona prende l'antibiotico DOPO che la polmonite sia stata diagnosticata.

Dov'è il problema?

Qualità dei Dati e Feature Engineering

Dati, dati, dati… non sono solo nella nostra mente!

Data leakage: Problema che si presenta quando un modello è capace di lavorare accuratamente in fase di addestramento, ma non in fase di rilascio.

Nel caso precedente, la variabile `'Ha preso antibiotico'` sarà disponibile quando il modello sarà creato, poiché il dato fa riferimento ad uno storico disponibile delle persone affette da polmonite, ma non sarà necessariamente disponibile una volta che il modello sarà chiamato a predire la malattia sulla base di dati nuovi!

L'esempio spiega il problema dei *leaky predictor*, ovvero quelle caratteristiche che ci aiutano sicuramente a caratterizzare il problema, ma che nella pratica non saranno disponibili il più delle volte, potenzialmente causando quindi il fallimento nostro modello —> non possiamo contare su una caratteristica se questa non sarà disponibile!

Questo è uno dei problemi. Esistono altre tipologie di data leakage che vedremo quando parleremo di validazione di un modello di machine learning.

Ecco però l'importanza di *ragionare* e *progettare* bene l'insieme di caratteristiche/metriche da considerare quando si vuole avere una soluzione intelligente basata su machine learning.

Messa in questi termini, sembra una cosa banale. Se stiamo progettando un sistema intelligente, dovremmo essere prima di tutto intelligenti noi… ma non è sempre così banale, ed è per questo che abbiamo bisogno di processi sistematici!

Qualità dei Dati e Feature Engineering

Tipologie di dati

Dato: In termini di machine learning, un dato è un qualsiasi elemento di cui si dispone per formulare un giudizio o risolvere un problema.

Come potreste facilmente pensare, esistono diversi tipi di dati.

Dati strutturati

Dati non strutturati

Dati semi-strutturati

Qualità dei Dati e Feature Engineering

Tipologie di dati

Dato: In termini di machine learning, un dato è un qualsiasi elemento di cui si dispone per formulare un giudizio o risolvere un problema.

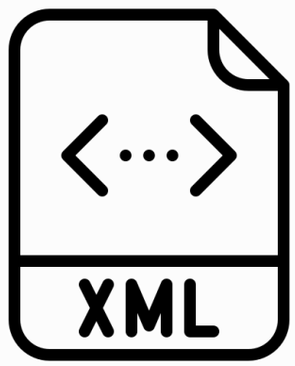
Come potreste facilmente pensare, esistono diversi tipi di dati.

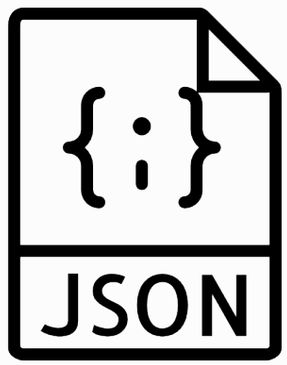
Dati strutturati | Dati non strutturati | Dati semi-strutturati

I dati strutturati sono quei dati tabulari per cui righe e colonne sono ben definite.

Il formato dei dati strutturati è molto stretto, nel senso che per ogni colonna sappiamo con esattezza il significato di un dato e quali sono i tipi di informazione per quel dato.

Spesso questi dati sono memorizzati in basi di dati che rappresentano anche le relazioni tra i dati. In questo caso, i dati possono essere recuperati tramite query.





Qualità dei Dati e Feature Engineering

Tipologie di dati

Dato: In termini di machine learning, un dato è un qualsiasi elemento di cui si dispone per formulare un giudizio o risolvere un problema.

Come potreste facilmente pensare, esistono diversi tipi di dati.

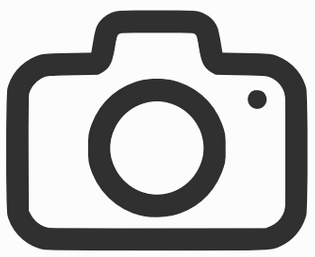
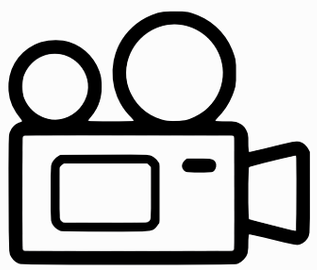
Dati strutturati | Dati non strutturati | Dati semi-strutturati

I dati non strutturati possono essere rappresentati da qualsiasi tipo di file che non ricade nella categoria dei dati strutturati.

Sono sicuramente i più difficili di estrarre poiché richiedono degli strumenti *ad-hoc*, come parser e/o tool di formattazione creati sulla base dello specifico dato.

In questa categoria rientrano i dati testuali, per i quali una branca specifica dell'intelligenza artificiale nota come Natural Language Processing è stata definita.



Qualità dei Dati e Feature Engineering

Tipologie di dati

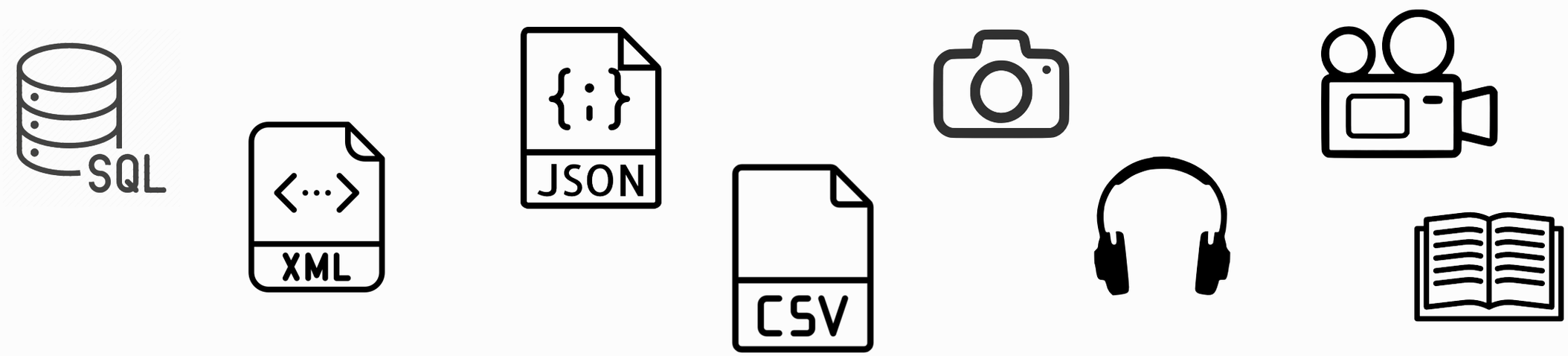
Dato: In termini di machine learning, un dato è un qualsiasi elemento di cui si dispone per formulare un giudizio o risolvere un problema.

Come potreste facilmente pensare, esistono diversi tipi di dati.

Dati strutturati | Dati non strutturati | Dati semi-strutturati

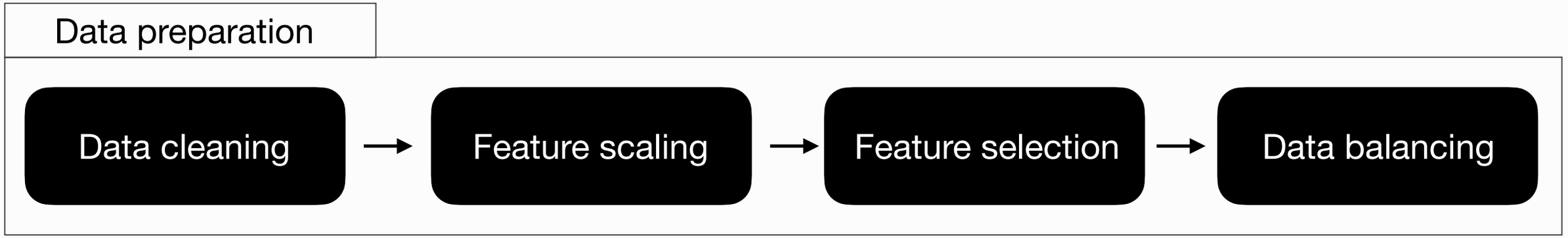
In questo caso, il formato è a metà tra lo strutturato e lo non strutturato. Mentre il formato è fissato, la struttura non ha una definizione stretta. Ad esempio, dati tabulari potrebbero avere dati mancanti o dati espressi tramite formato non strutturato.

I dati semi-strutturali sono generalmente memorizzati come file. Alcuni però potrebbero anche essere presentati all'interno di basi di dati document-oriented.



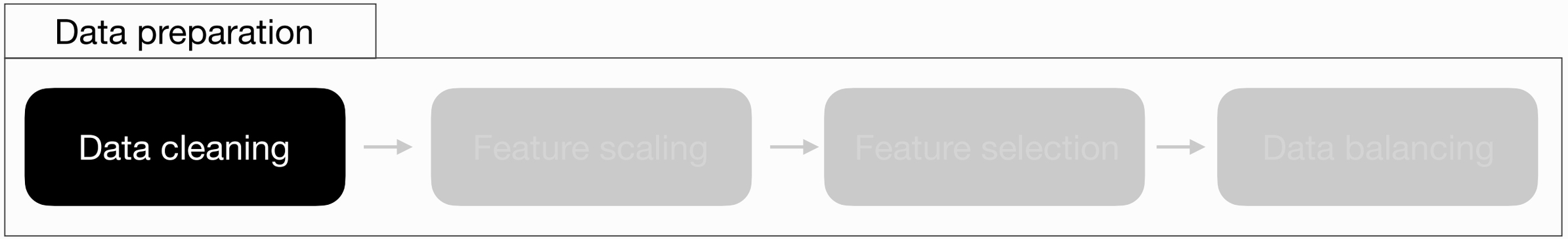
Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Qualità dei Dati e Feature Engineering

Ingegneria dei dati



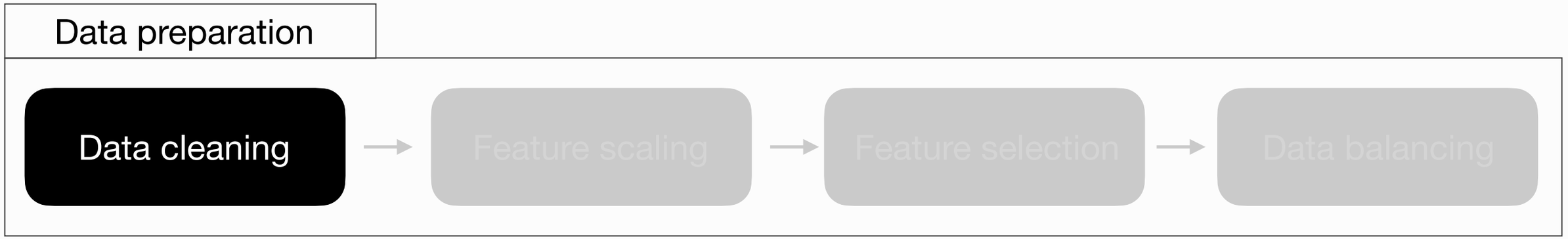
Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Spam	Dominio email	Oggetto	Numero di allegati	Lunghezza del testo	Colore del testo
NO	unisa.it	Esame	1	300	Nero
NO	gmail.com	Esame	-	180	Nero
SI	live.it	Business Interest!	-	222	Nero
SI	spam.it	How to get			-
NO	unisa.it				Nero

Cosa possiamo fare quando ci troviamo ad avere a che fare con dei dati mancanti?

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Data imputation: Insieme di tecniche che possono stimare il valore di dati mancanti sulla base dei dati disponibili oppure mitigare il problema dei dati mancanti.

Due soluzioni al problema dei dati mancanti sono abbastanza banali:

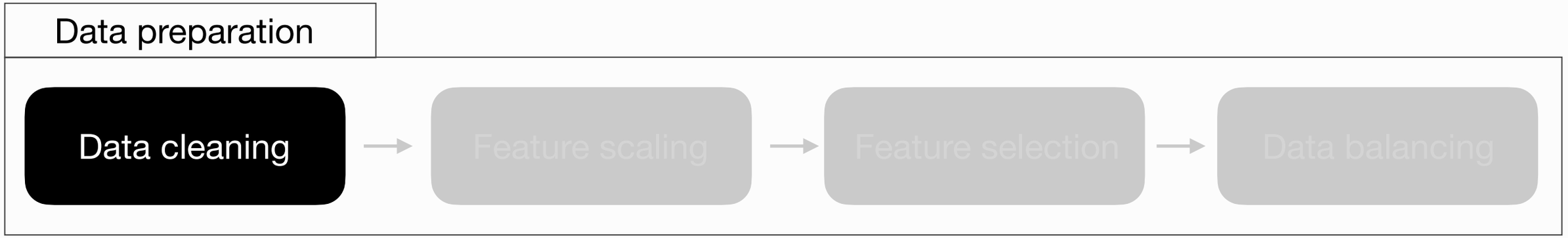
(1) *Scartare le righe* del dataset che presentano dati mancanti: una soluzione facile, ma non sempre applicabile. Se per il problema in esame non abbiamo tante osservazioni, scartare le righe diventa un problema.

(2) *Scartare le colonne* del dataset che presentano dati mancanti: una soluzione altrettanto facile, ma non sempre applicabile o desiderabile. Se la colonna che presenta dati mancanti rappresenta una caratteristica rilevante per il problema in esame, non possiamo scartarla.

Negli altri casi, queste due soluzioni sono assolutamente plausibili e raccomandabili!

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Data imputation: Insieme di tecniche che possono stimare il valore di dati mancanti sulla base dei dati disponibili oppure mitigare il problema dei dati mancanti.

In alternativa, l'imputazione statistica, la quale si basa sull'applicazione di semplici tecniche statistiche per stimare il valore dei dati mancanti.

Ad esempio, la media

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

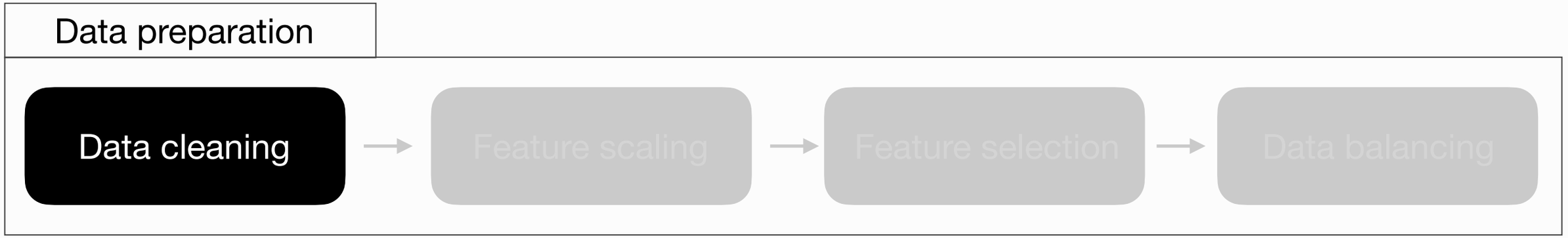
mean()

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

Sicuramente facile da applicare, ha due problemi: (1) Non può essere applicata su dati non-numerici; (2) Non considera l'incertezza quando va ad imputare i dati.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Data imputation: Insieme di tecniche che possono stimare il valore di dati mancanti sulla base dei dati disponibili oppure mitigare il problema dei dati mancanti.

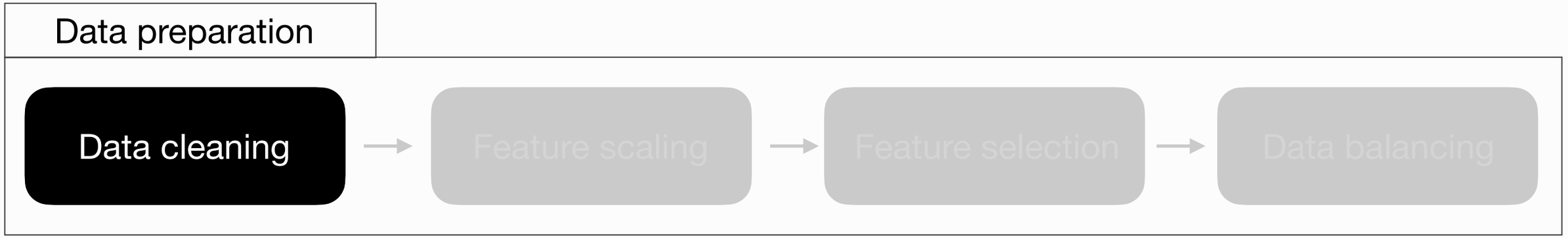
In alternativa, l'imputazione statistica, la quale si basa sull'applicazione di semplici tecniche statistiche per stimare il valore dei dati mancanti.

- Imputazione tramite most frequent imputation: i dati mancanti sono sostituiti dal valore più frequente contenuto in una colonna —> Occhio: quando ci sono tanti dati mancanti, si rischia di *influenzare eccessivamente la distribuzione della variabile*!

- Imputazione deduttiva: Il progettista definisce una regola di imputazione sulla base di una deduzione logica. Ad esempio, consideriamo una variabile che riporta il numero di figli di una famiglia per anno: se abbiamo 2 figli all'anno 1 e 2 figli all'anno 3, è ragionevole pensare che la variabile sarà uguale a 2 per l'anno 2 —> Occhio: non è sempre facile *trovare una regola valida e sicuramente non è un approccio scalabile*!

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Data imputation: Insieme di tecniche che possono stimare il valore di dati mancanti sulla base dei dati disponibili oppure mitigare il problema dei dati mancanti.

In alternativa, l'imputazione statistica, la quale si basa sull'applicazione di semplici tecniche statistiche per stimare il valore dei dati mancanti.

uiti dal
tanti dati
abile!

ulla base
a il
i all'anno
cchio: non
scalabile!

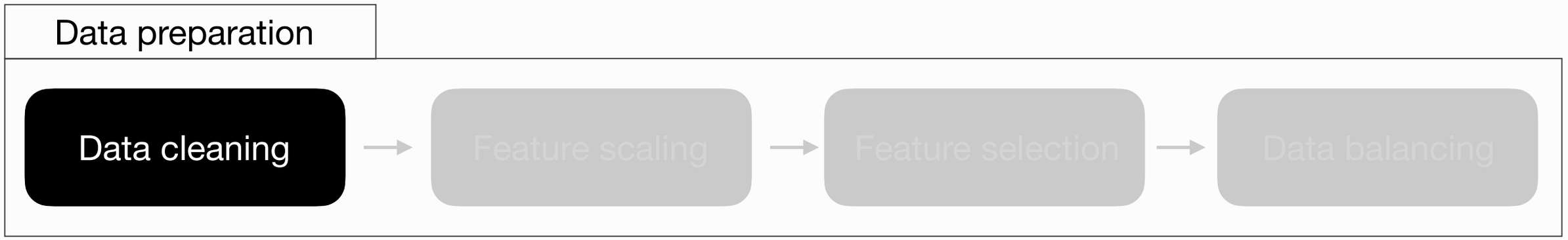
E quindi, se abbiamo un problema con i dati mancanti, cosa usiamo?

Dipende! Dipende da quanto è esteso il problema, da che tipo di dati sono mancanti e dalla facilità di trovare una relazione che consenta di stimare il valore dei dati mancanti.

In alcuni casi, possiamo anche decidere di combinare le tecniche viste in precedenza: di fatti, potrebbe succedere che per qualche variabile un tipo di imputazione sia più adatto degli altri, e viceversa.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

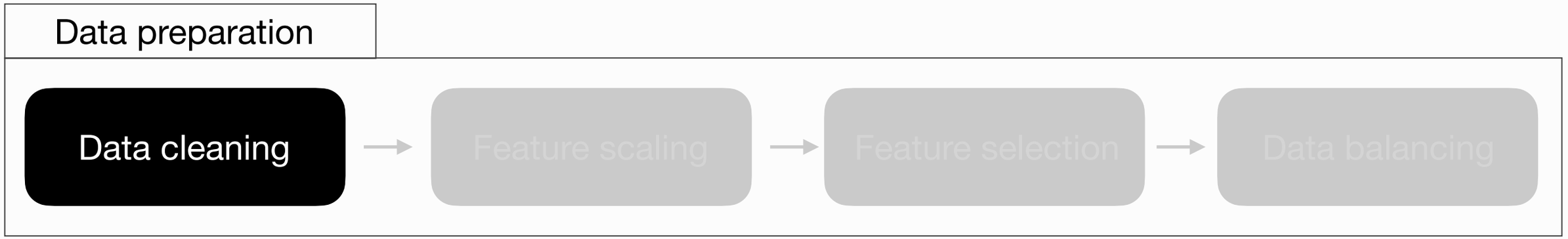
Ma se i dati strutturati sono da considerare talvolta problematici, cosa succede con i dati non strutturati? Il caso tipico è quello di testo scritto in linguaggio naturale.

L’analisi del linguaggio naturale è utilizzata in molti contesti e, non a caso, un intero campo dell’Intelligenza Artificiale è dedicato al Natural Language Processing.

Il linguaggio naturale è estremamente utile: ad esempio, considerate le user review rilasciate dagli utenti sul Google Play Store…

Qualità dei Dati e Feature Engineering

Ingegneria dei dati

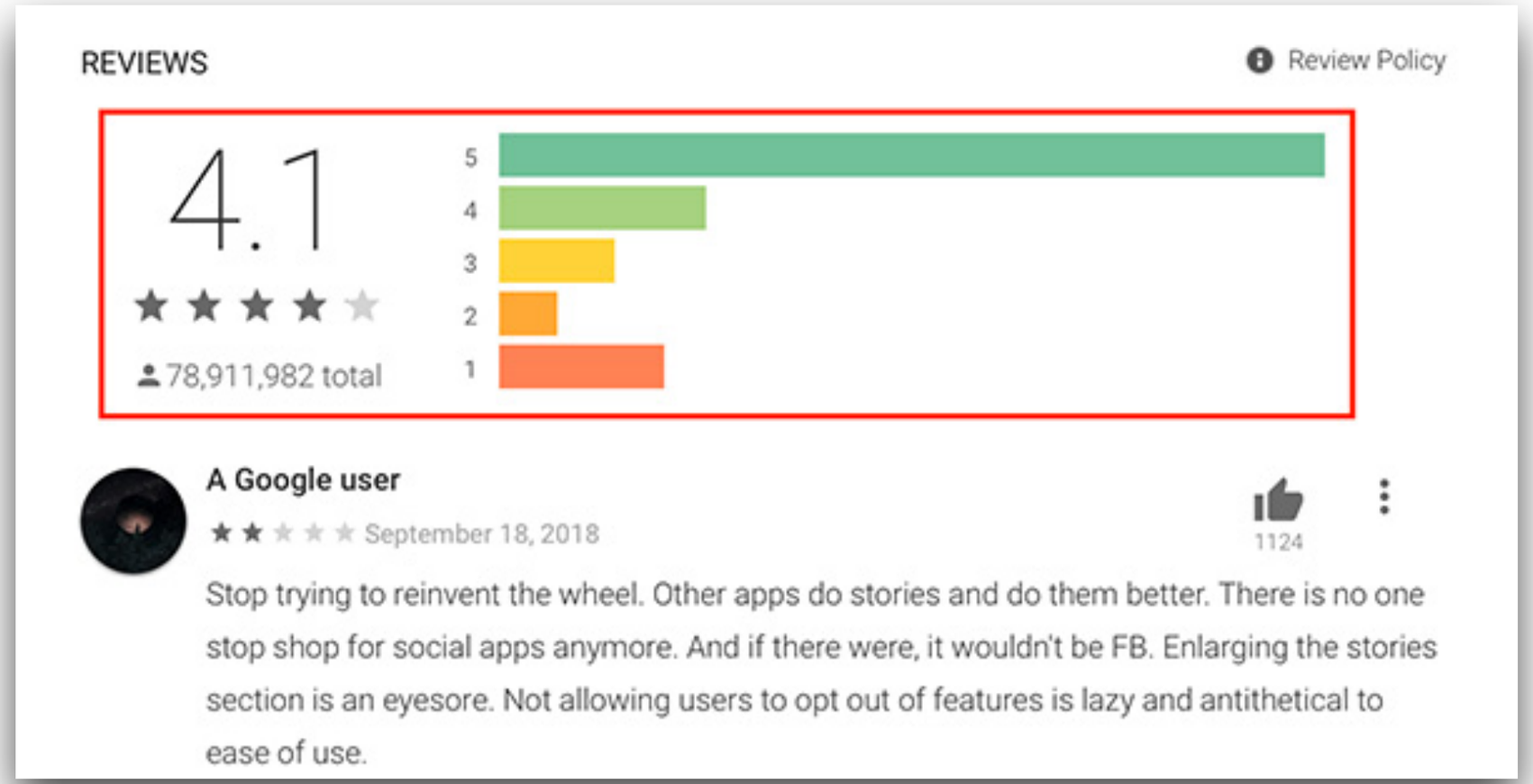


Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

e con i

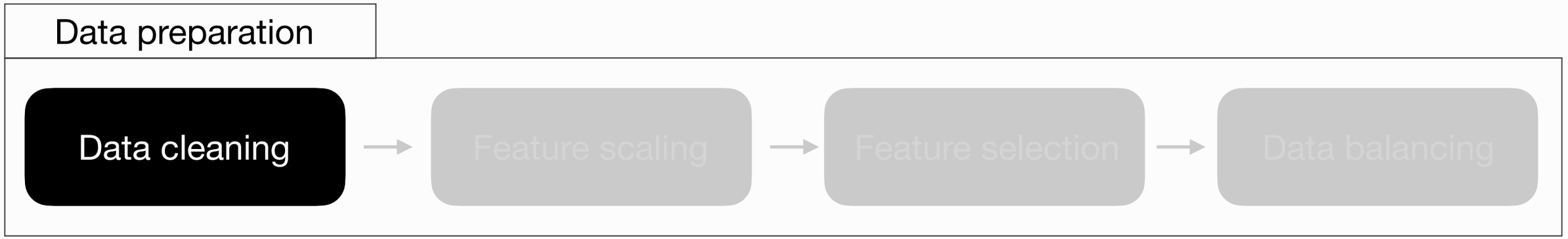
ntero

view



Qualità dei Dati e Feature Engineering

Ingegneria dei dati

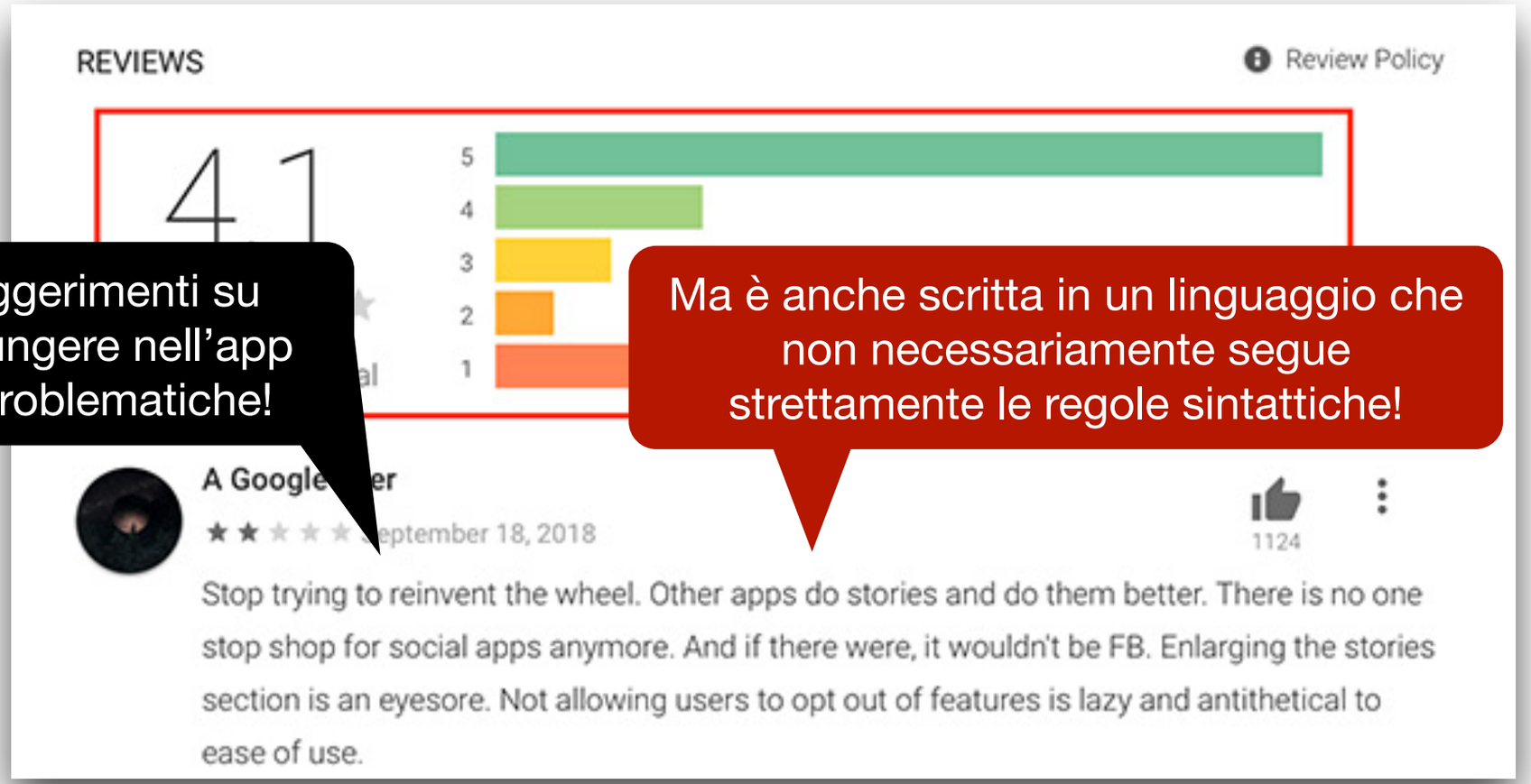


Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

e con i

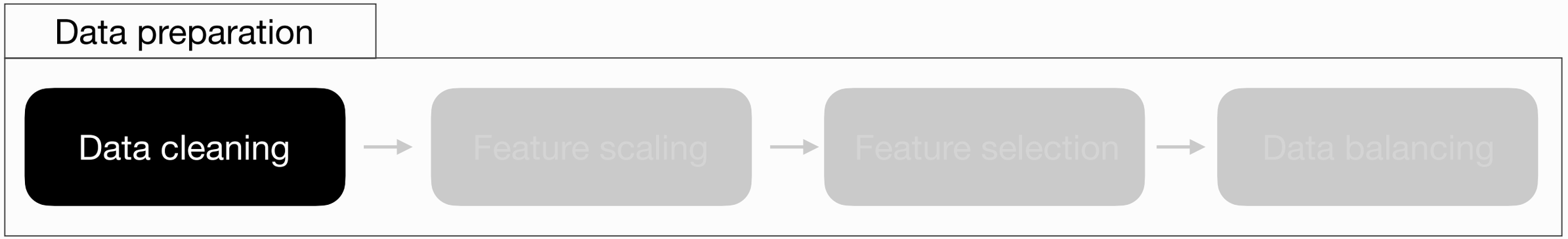
ntero

view



Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

e con i

ntero

view

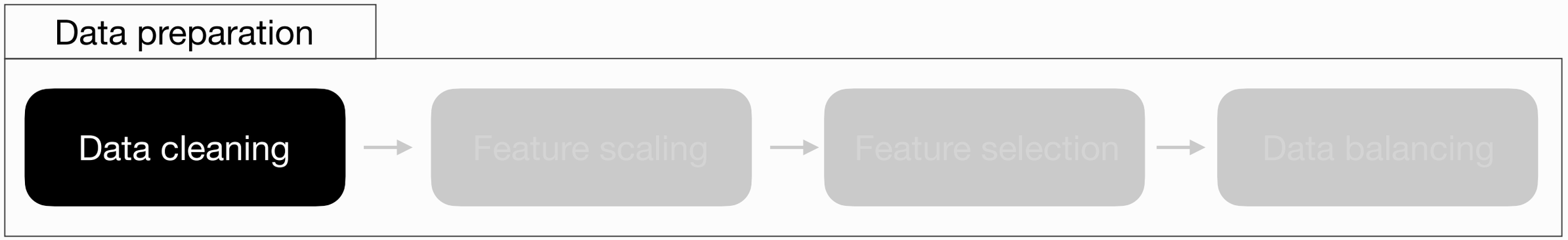
NB: tutto, ma proprio tutto, può essere considerato come testo!

Ogni tipologia di testo ha le sue peculiarità —> il processo di pulizia va sempre adattato al contesto!

```java
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

    connect = DBConnection.getConnection();

    String sql = "INSERT INTO USER"
        + "(login,first_name,last_name,password"
        + ",email,cell,id_parent) " + "VALUES ("
        + pUser.getLogin() + ","
        + pUser.getFirstName() + ","
        + pUser.getLastName() + ","
        + pUser.getPassword() + ","
        + pUser.getEMail() + ","
        + pUser.getCell() + ","
        + pUser.getIdParent() + ")";

    executeOperation(connect, sql);
 }
```

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Ma se i dati strutturati sono da considerare talvolta problematici, cosa succede con i dati non strutturati? Il caso tipico è quello di testo scritto in linguaggio naturale.

L'analisi del linguaggio naturale è utilizzata in molti contesti e, non a caso, un intero campo dell'Intelligenza Artificiale è dedicato al Natural Language Processing.

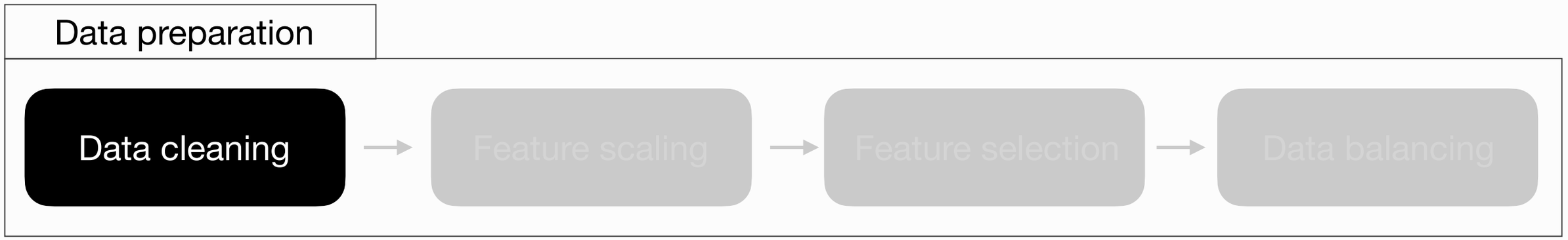
Il linguaggio naturale è estremamente utile: ad esempio, considerate le user review rilasciate dagli utenti sul Google Play Store…

Abbreviazioni, errori di battitura, errori grammaticali, ed altro… fare mining di testi è tutt'altro che facile! Abbiamo perciò bisogno di alcuni step dedicati per *estrarre della semantica da testi dove la semantica è nascosta.*

Ma attenzione: Le tecniche di data cleaning di cui parleremo funzionano discretamente bene su testi scritti in lingua inglese! Per altre lingue, necessiteremo, laddove possibile, di personalizzare questi strumenti.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Pulizia dei dati = Come sopravvivere a dati mancanti o rumorosi!

Innanzitutto, sono tre le sfide principali da affrontare quando si ha a che fare con il testo: (1) difficoltà di estrazione; (2) ambiguità del linguaggio; (3) esistono molti modi di esprimere concetti simili.

Per semplicità, consideriamo questo statement: `connect = DBConnection.getConnection();`

Leggendolo, possiamo dire che l'istruzione consente di stabilire una connessione con un database. Perché?

Utilizza i termini `connect` e `connection`, che richiamano il concetto di "connessione";

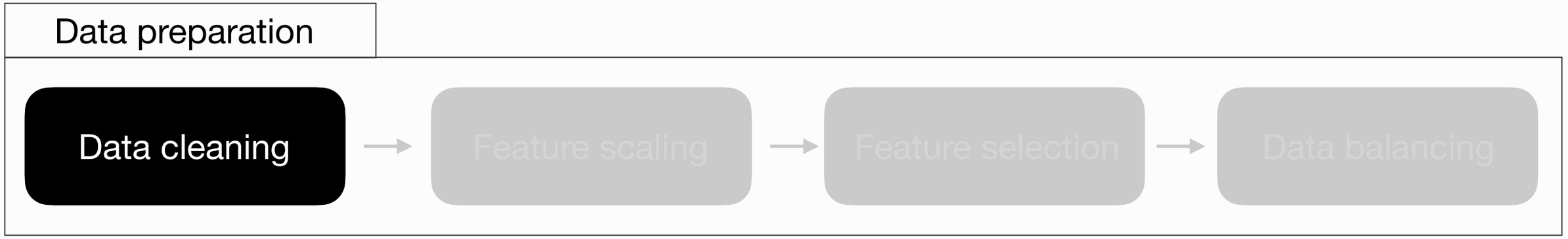
L'istruzione include un riferimento a `DB`, che associamo normalmente ad un database;

L'istruzione usa una notazione che associamo ad un significato: il simbolo `'='` richiama l'assegnazione di un valore ad una variabile, il simbolo `'.'` ci indica che è il metodo `getConnection` della class `DBConnection` a restituire una connessione.

Ma noi siamo essere pensanti, dobbiamo consentire ad una macchina di ragionare!

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect = DBConnection.getConnection();
```

Se ci interessa dare un senso ai simboli, allora dovremo prima di tutto procedere a sostituirli con delle parole.

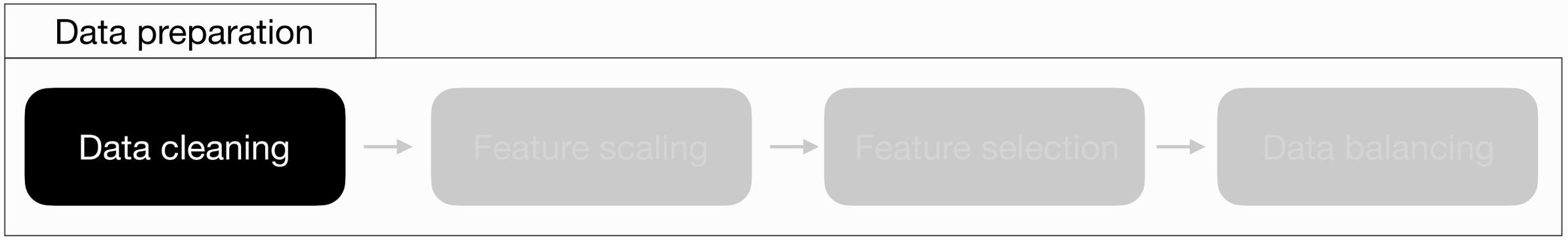
```
connect equals DBConnection calls getConnection();
```

Nell'esempio, non ci interessa sostituire il simbolo `'();'` perché non aggiunge semantica!

La sostituzione è possibile definendo un *dizionario*, ovvero una struttura che restituisce la parola associata ad un simbolo.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect = DBConnection.getConnection();
```

Se ci interessa dare un senso ai simboli, allora dovremo prima di tutto procedere a sostituirli con delle parole.

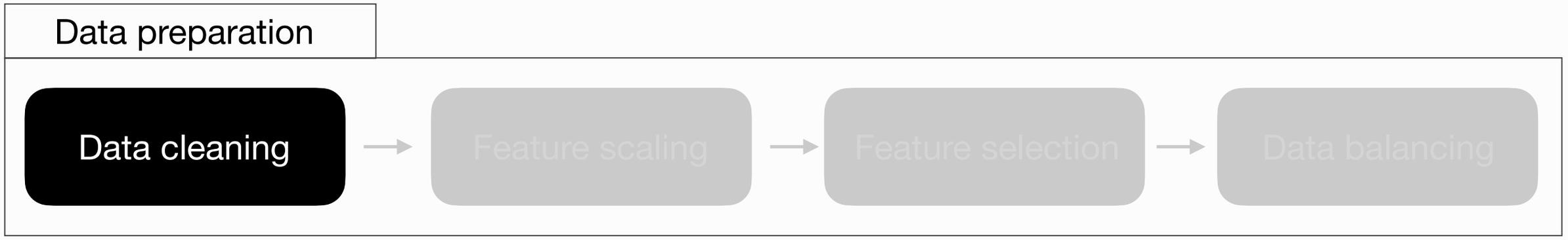
```
connect equals DBConnection calls getConnection();
```

Eliminiamo quindi tutto ciò che può solo creare rumore.

```
connect equals DBConnection calls getConnection~~();~~
```

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect = DBConnection.getConnection();
```

↓ Se ci interessa dare un senso ai simboli, allora dovremo prima di tutto procedere a sostituirli con delle parole.

```
connect equals DBConnection calls getConnection();
```

↓ Eliminiamo quindi tutto ciò che può solo creare rumore.

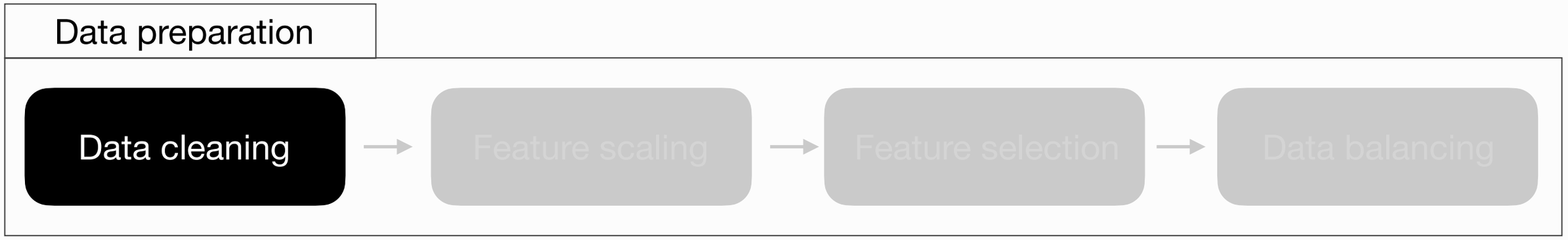
```
connect equals DBConnection calls getConnection();
```

↓ Per consentire ad un algoritmo di “comprendere” il testo, non possiamo avere situazioni in cui più parole siano concatenate.

```
connect equals DB Connection calls getConnection
```

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect = DBConnection.getConnection();
```

Se ci interessa dare un senso ai simboli, allora dovremo prima di tutto procedere a sostituirli con delle parole.

```
connect equals DBConnection calls getConnection();
```

Eliminiamo quindi tutto ciò che può solo creare rumore.

```
connect equals DBConnection calls getConnection();
```

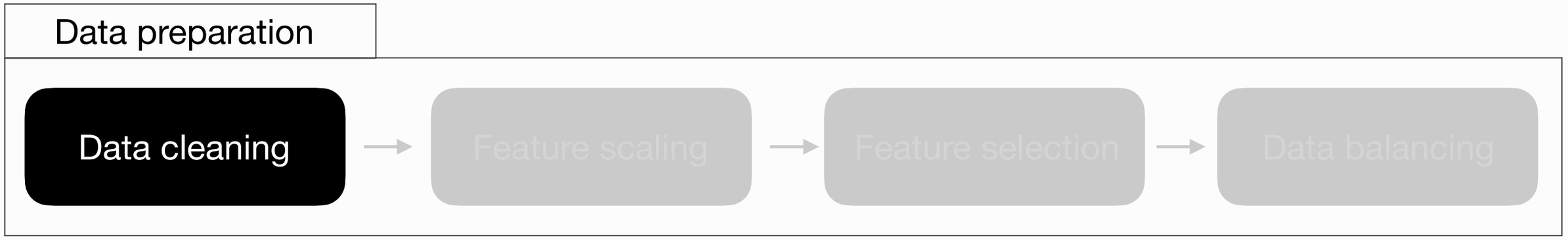
Per consentire ad un algoritmo di “comprendere” il testo, non possiamo avere situazioni in cui più parole siano concatenate.

```
connect equals DB Connection calls get Connection
```

Questa operazione è fatta sulla base si euristiche (camelCase, underscore, ecc.)

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect equals DB Connection calls get Connection
```

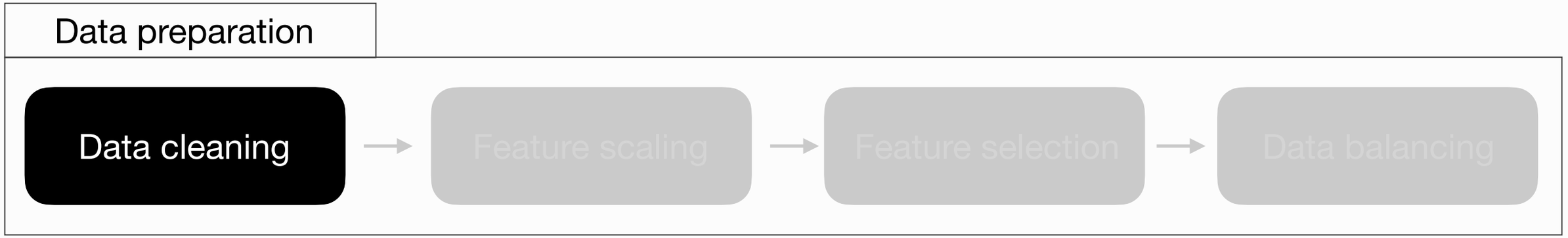
Tramite l'utilizzo di dizionari, possiamo procedere poi a quello che definiamo *contraction expansion*.

```
connect equals database Connection calls get Connection
```

In inglese, la contraction expansion consente spesso di disambiguare il significato delle frasi (ad esempio, l'espansione di "wouldn't" in "would not" consente di definire una *negazione*.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect equals DB Connection calls get Connection
```

↓ Tramite l'utilizzo di dizionari, possiamo procedere poi a quello che definiamo *contraction expansion*.

```
connect equals database Connection calls get Connection
```

↓ Connect e connection fanno riferimento alla stessa semantica, ma sono scritte in maniera diversa!

```
connect equals database Connect calls get Connect
```

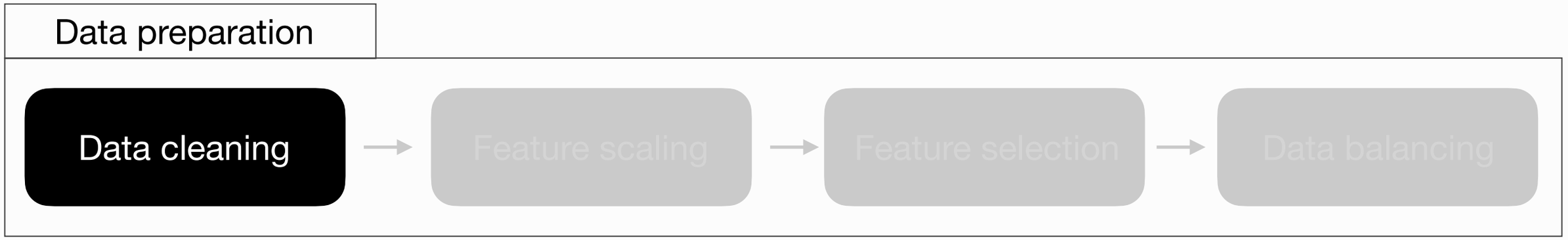
Questo step si definisce *stemming* e consiste nel processo di sostituzione di una parola nella sua radice.

Anche qui, si fa riferimento ad un dizionario.

NB: Nell'esempio specifico, potremmo anche considerare i nomi dei metodi come nomi propri, ma dipende da quello che vogliamo fare!

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect equals DB Connection calls get Connection
```

↓ Tramite l'utilizzo di dizionari, possiamo procedere poi a quello che definiamo *contraction expansion*.

```
connect equals database Connection calls get Connection
```

↓ Connect e connection fanno riferimento alla stessa semantica, ma sono scritte in maniera diversa!

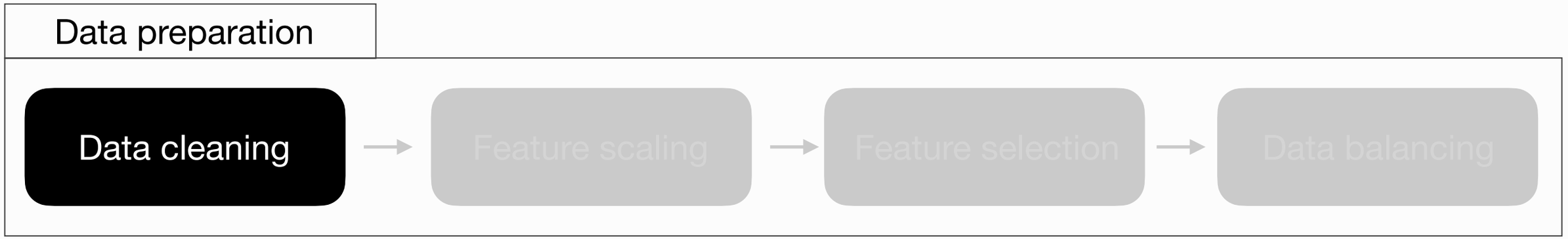
```
connect equals database Connect calls get Connect
```

↓ Connect, Connect e connect sono la stessa cosa? Non proprio! Riduciamo tutto in maiuscolo/minuscolo.

```
connect equals database connect calls get connect
```

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Vediamo quindi un processo standard di *normalizzazione* del testo.

```
connect equals DB Connection calls get Connection
```

↓ Tramite l'utilizzo di dizionari, possiamo procedere poi a quello che definiamo *contraction expansion*.

```
connect equals database Connection calls get Connection
```

↓ Connect e connection fanno riferimento alla stessa semantica, ma sono scritte in maniera diversa!

```
connect equals database Connect calls get Connect
```

↓ Connect, Connect e connect sono la stessa cosa? Non proprio! Riduciamo tutto in maiuscolo/minuscolo.

```
connect equals database connect calls get connect
```

Altri step a volte necessari:

(1) Spelling correction;
(2) Filtrare verbi e sostantivi;
(3) Singolarizzazione;
(4) Rimozione delle ripetizioni;
(5) Rimozione di documenti;
(6) Stopword removal;

Le librerie di NLP implementano già molti di questi step. In alcuni casi, però, vanno personalizzati!

Qualità dei Dati e Feature Engineering

Ingegneria dei dati

Data preparation

Data cleaning → **Feature scaling** → Feature selection → Data balancing

Immaginate adesso di trovarvi nella seguente situazione.

Spam	Dominio email	Oggetto	Numero di allegati	Lunghezza del testo	Colore del testo
NO				300	Nero
NO				180	Nero
SI				222	Nero
SI				492	Nero
NO	unisa.it	Ricevimento	0	300	Nero

Il numero di allegati ha una distribuzione naturalmente minore rispetto alla distribuzione della lunghezza del testo

Nell'esempio, il numero di allegati va da 0 a 3, mentre la lunghezza del testo va da 180 a 492

La diversa distribuzione delle caratteristiche può impattare le prestazione di un machine learner

Qualità dei Dati e Feature Engineering

Ingegneria dei dati

Tutti gli algoritmi di machine learning lavorano sui dati.

Se l'insieme dei valori per una determinata caratteristica è molto diverso rispetto ad un altro, c'è il rischio di "far confondere" l'algoritmo di apprendimento —> l'algoritmo potrebbe sottostimare/sovrastimare l'importanza di una caratteristica poiché questa ha una scala di valori molto inferiore/superiore rispetto alle altre.

Feature scaling: Insieme di tecniche che consentono di normalizzare o scalare l'insieme di valori di una caratteristica.

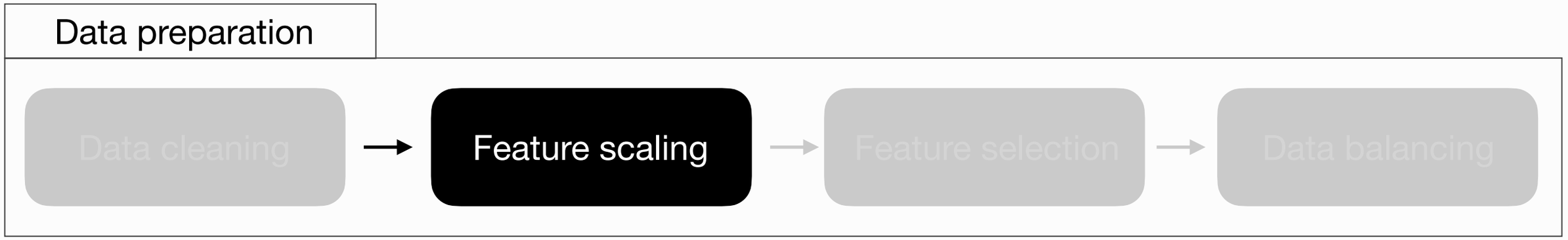
Il metodo più comune per normalizzare è chiamato *min-max normalization*:

$$x' = a + \frac{(x - min(x))(b - a)}{max(x) - mix(x)}$$

Dove *a* e *b* rappresentano i valori minimo e massimo che vogliamo ottenere dalla normalizzazione (ad esempio, 0 e 1 se vogliamo normalizzare nell'intervallo [0, 1].

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Facciamo un esempio…

Lunghezza del testo
300
180
222
492
300

$$x' = a + \frac{(x - min(x))(b - a)}{max(x) - mix(x)}$$

Il minimo della distribuzione è 180; Il massimo è 492. Vogliamo normalizzare in una scala da 0 a 1.

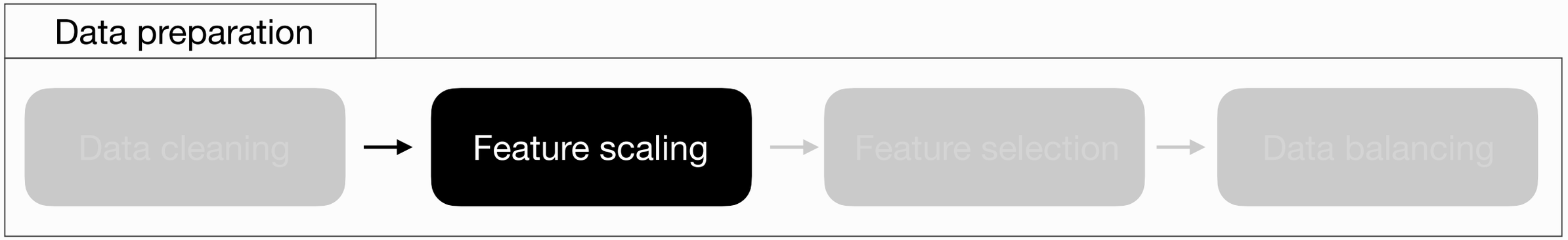
Prendiamo il caso di 300:

$$x' = 0 + \frac{(300 - 180)(1 - 0)}{492 - 180} = \frac{120}{312} = 0{,}38$$

Nella distribuzione normalizzata, 180 sarà uguale a 0, mentre il valore 492 sarà uguale ad 1.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Facciamo un esempio…

Lunghezza del testo
300
180
222
492
300

$$x' = a + \frac{(x - min(x))(b - a)}{max(x) - mix(x)}$$

Il minimo della distribuzione è 180; Il massimo è 492. Vogliamo normalizzare in una scala da 0 a 1.

Prendiamo il caso di 300:

$$x' = 0 + \frac{(300 - 180)(1 - 0)}{492 - 180} = \frac{120}{312} = 0{,}38$$

Nella distribuzione normalizzata, 180 sarà uguale a 0, mentre il valore 492 sarà uguale ad 1.

Lunghezza del testo
0,38
0
0,13
1
0,38

Qualità dei Dati e Feature Engineering

Ingegneria dei dati

Tutti gli algoritmi di machine learning lavorano sui dati.

Se l'insieme dei valori per una determinata caratteristica è molto diverso rispetto ad un altro, c'è il rischio di "far confondere" l'algoritmo di apprendimento —> l'algoritmo potrebbe sottostimare/sovrastimare l'importanza di una caratteristica poiché questa ha una scala di valori molto inferiore/superiore rispetto alle altre.

Feature scaling: Insieme di tecniche che consentono di normalizzare o scalare l'insieme di valori di una caratteristica.

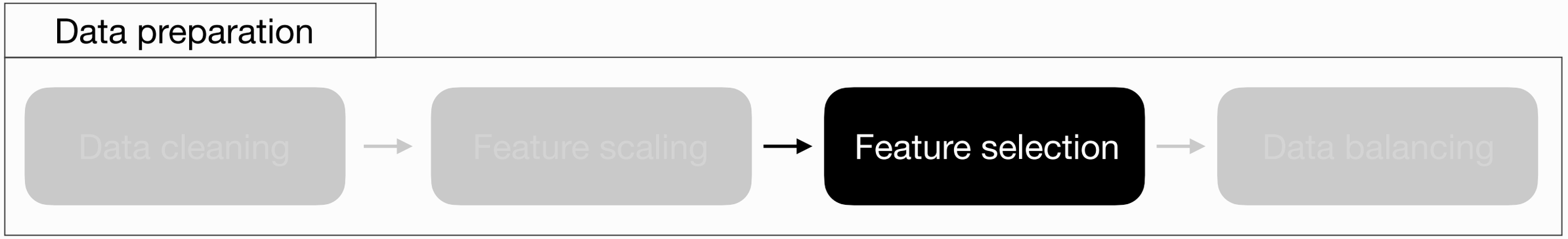
Una alternativa spesso considerata è quella della *z-score normalization*.

$$x' = \frac{(x - \bar{x})}{\sigma}$$

Dove *x* è il valore originale, $\bar{x}$ è la media della distribuzione, σ la deviazione standard.
Questa è la normalizzazione di default implementata da molti dei tool di machine learning.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Engineering: Processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le caratteristiche (feature) dai dati grezzi estraibili tramite tecniche di data mining.

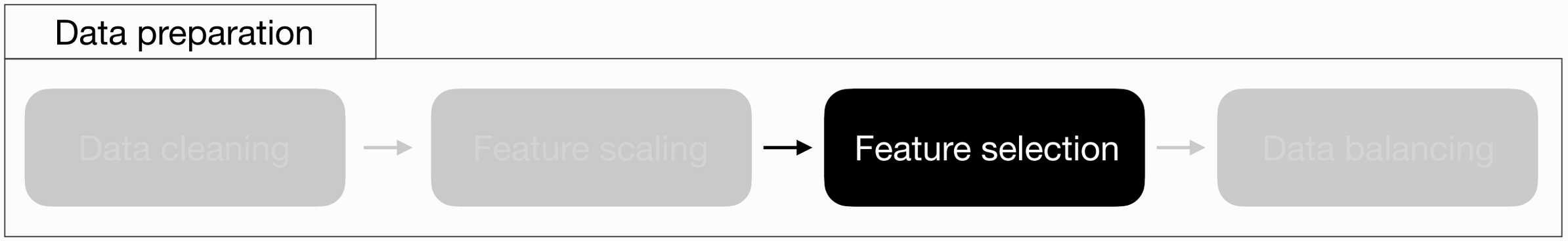
In altri termini, il feature engineering ci consente di “dare un senso” ai dati strutturati, non strutturati o semi-strutturati che possiamo estrarre dalle diverse sorgenti a disposizione.

L’obiettivo finale è quello di definire delle caratteristiche, anche chiamate feature, metriche, o variabili indipendenti che possano caratterizzare gli aspetti principali del problema in esame e, quindi, avere una buona potenza predittiva.

Questo processo è quello più creativo e complesso dell’intera modellazione, poiché dipende dal problema specifico e dall’abilità del progettista di individuare quali sono le caratteristiche che possono influenzare la predizione di un fenomeno.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Engineering: Processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le caratteristiche (feature) dai dati grezzi estraibili tramite tecniche di data mining.

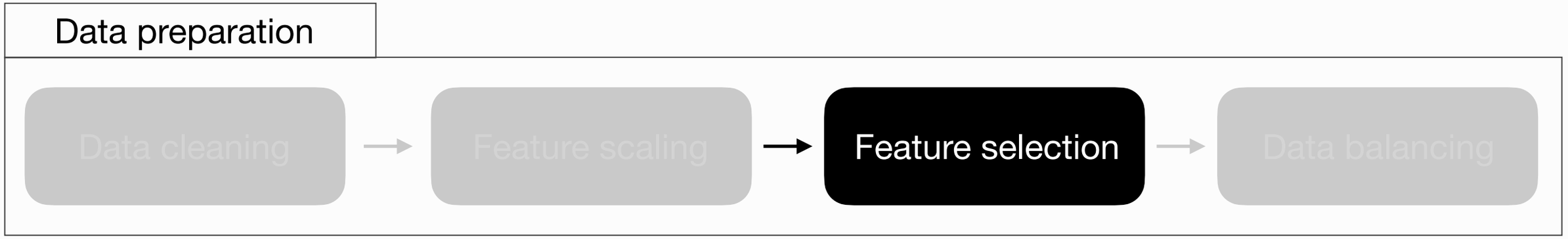
Ad esempio, consideriamo il problema di dover classificare i casi di polmonite preso in esame precedentemente. L'insieme dei dati numerici che fanno riferimento all'età di una persona è un dato, l'età è invece una caratteristica/feature che possiamo considerare per il nostro esercizio predittivo.

Norvig diceva: *"More data beats clever algorithms, but better data beats more data"*. In altri termini, identificare le feature rilevanti è la chiave del successo!

Se nelle fasi precedenti abbiamo potuto estrarre e manipolare una grande quantità di dati, arrivati a questo punto ci interessa selezionare le sole variabili o i soli dati che hanno un effettivo impatto sulla variabile che intendiamo predire.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Engineering: Processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le caratteristiche (feature) dai dati grezzi estraibili tramite tecniche di data mining.

Identificare buone feature porta diversi vantaggi da un punto di vista pratico:

(1) *La flessibilità del modello di machine learning aumenta*: Anche scegliendo un algoritmo di apprendimento sub-ottimo, le prestazione resteranno alte;

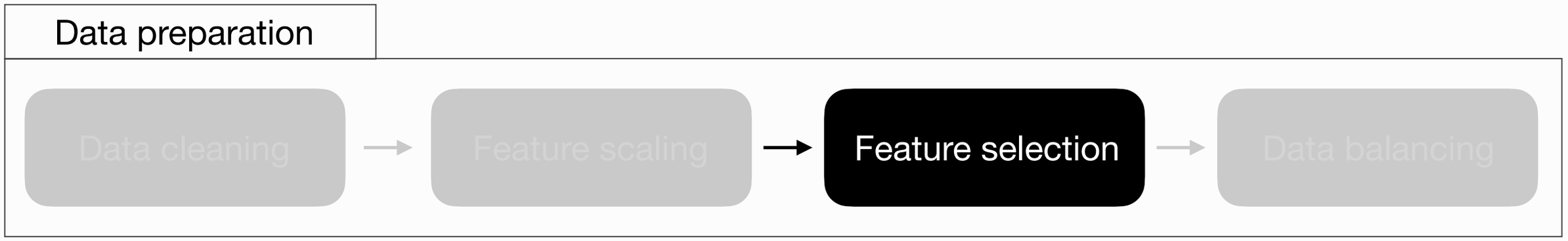
(2) *La semplicità del modello aumenta*: Anche scegliendo una configurazione sub-ottima dell'algoritmo di apprendimento, le prestazione resteranno alte;

(3) *Il livello di explainability del modello aumenta*: Avendo delle buone feature, riuscirò facilmente a capire il *perché* del comportamento del modello;

(4) *Le prestazioni del modello aumentano*: Avendo delle buone feature, è più semplice per il machine learner apprendere e capire come comportarsi su dati ignoti.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Engineering: Processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le caratteristiche (feature) dai dati grezzi estraibili tramite tecniche di data mining.

E quindi, come posso identificare queste feature?

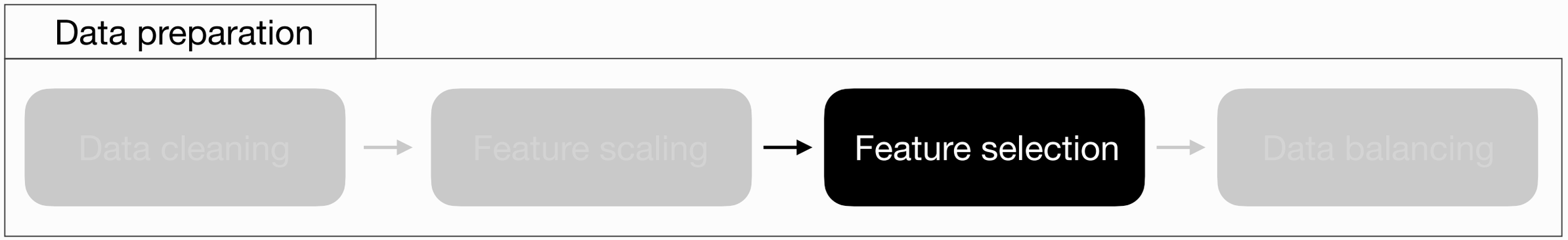
Il feature engineering ha diversi sotto-campi, tra cui la feature extraction (riduzione della dimensionalità), la feature construction e la feature selection.

Feature extraction e construction differiscono poiché la prima punta a generare automaticamente delle feature dai dati, la seconda punta alla creazione di feature da parte del progettista. La prima è spesso menzionata come riduzione della dimensionalità poiché parte da un insieme più grande di dati per identificare quelli più significativi.

Feature extraction e selection differiscono poiché la seconda punta a selezionare le variabili più significative partendo da quelle a disposizione, mentre la feature extraction parte dai dati grezzi e li converte in nuovi tipi di dati.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Selection: Processo tramite il quale vengono selezionate le caratteristiche più correlate al problema in esame, a partire da un insieme di caratteristiche esistenti.

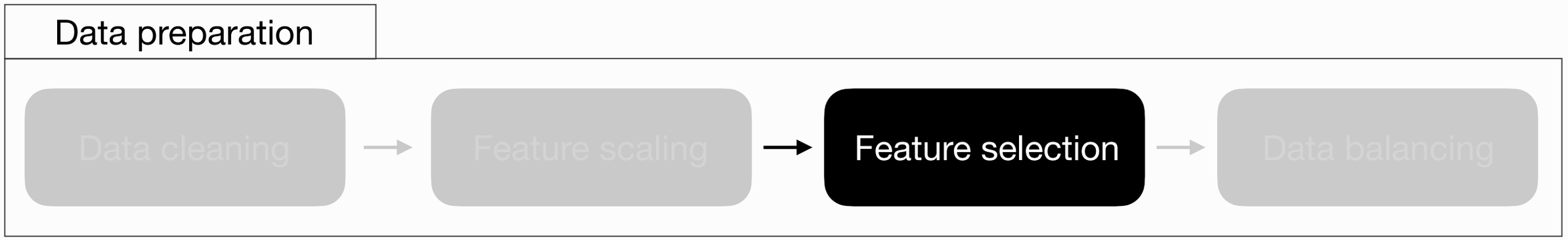
Il modo più semplice per approcciare il problema delle selezione delle feature è quello di usare metodi non supervisionati.

- *Eliminazione di feature con bassa varianza*: Questo metodo prevede l'eliminazione delle caratteristiche che hanno valori simili nel dataset. Il razionale è semplice: se una variabile è simile nell'intero dataset, è probabile che non sia discriminante!

Spam	Dominio email	Oggetto	Numero di allegati	Lunghezza del testo	Colore del testo
NO	unisa.it	Esame	1	300	Nero
NO	gmail.com	Esame	3	180	Nero
SI	live.it	Business Interest!	1	222	Nero
SI	spam.it	How to get money!	3	492	Nero
NO	unisa.it	Ricevimento	0	300	Nero

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Selection: Processo tramite il quale vengono selezionate le caratteristiche più correlate al problema in esame, a partire da un insieme di caratteristiche esistenti.

Il modo più semplice per approcciare il problema delle selezione delle feature è quello di usare metodi non supervisionati.

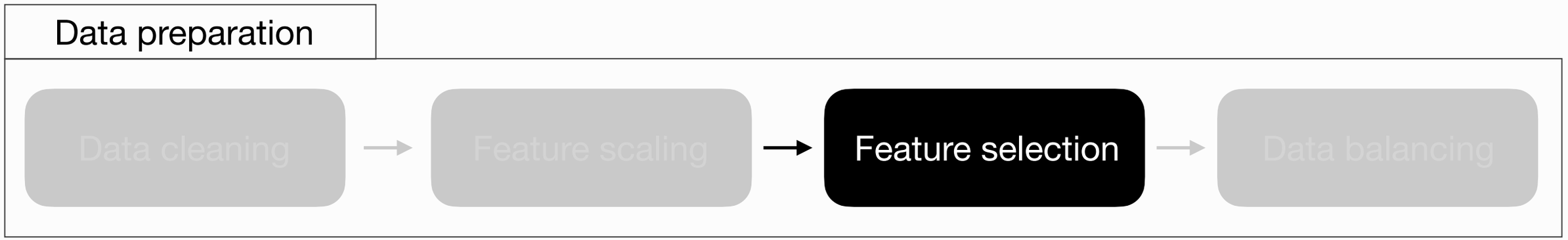
- *Eliminazione di feature con bassa varianza*: Questo metodo prevede l'eliminazione delle caratteristiche che hanno valori simili nel dataset. Il razionale è semplice: se una variabile è simile nell'intero dataset, è probabile che non sia discriminante!

Spam	Dominio email	Oggetto	Numero di allegati	Lunghezza del testo	Colore del testo
NO	unisa.it	Esame	1	300	Nero
NO	gmail.com				Nero
SI	live.it				Nero
SI	spam.it				Nero
NO	unisa.it	Ricevimento	0	300	Nero

Nell'esempio dello spam, il colore del testo è sempre 'Nero'. Sembra perciò naturale non considerare questa una variabile significativa.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Feature Selection: Processo tramite il quale vengono selezionate le caratteristiche più correlate al problema in esame, a partire da un insieme di caratteristiche esistenti.

Il modo più semplice per approcciare il problema delle selezione delle feature è quello di usare metodi non supervisionati.

- *Eliminazione di feature con bassa varianza*: Questo metodo prevede l'eliminazione delle caratteristiche che hanno valori simili nel dataset. Il razionale è semplice: se una variabile è simile nell'intero dataset, è probabile che non sia discriminante!

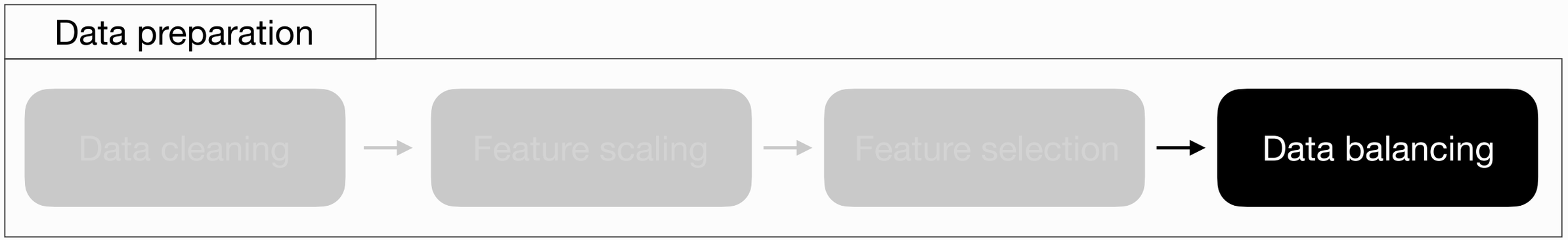
  —> Di default, il metodo prevede l'eliminazione di variabili con varianza zero.

- *Eliminazione univariata di feature*: Questo metodo prevede la selezione delle variabili sulla base di test statistici (ad esempio, il Pearson's χ^2). Ogni variabile indipendente viene correlata con la variabile dipendente, ottenendo quindi una classifica delle variabili basata sulla correlazione. Si sceglieranno solo le *k* migliori variabili.

Più avanti, accenneremo anche alle tecniche supervisionate per la feature selection.

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



La maggior parte dei modelli di machine learning funzionano bene solo quando il numero di esempi di una certa classe (ad esempio, il fatto che una mail sia classifica come ‘spam’) è simile al numero di esempi di un’altra classe (la classe ‘no-spam’).

D’altronde, come può qualcuno apprendere a caratterizzare qualcosa se non conosce abbastanza di quel qualcosa che dovrebbe caratterizzare?

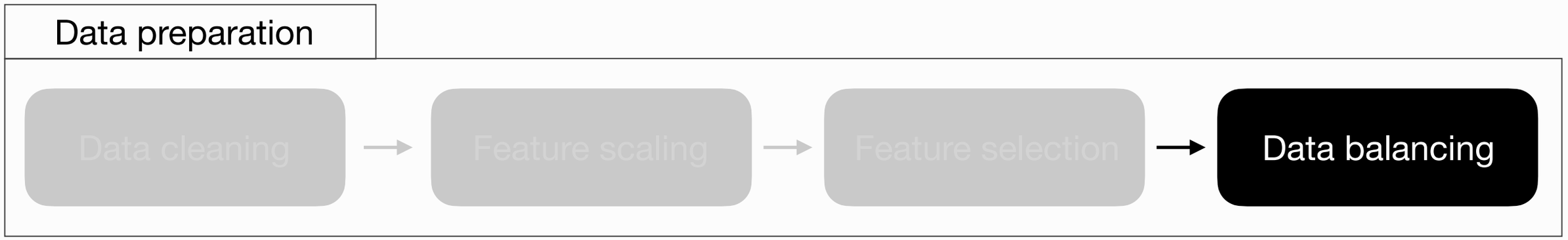
Problema: Molti dei problemi reali sono *sbilanciati*!

Pensate ad esempio a tutti i problemi di carattere medico. Classificare una polmonite, o il COVID-19, o un tumore è tutt’altro che banale. Per questi problemi il numero di pazienti affetti da una malattia è molto molto inferiore al numero di pazienti malati.

Se non considerassimo il problema dello sbilanciamento dei dati, definiremmo molto probabilmente un modello di machine learning capace di caratterizzare bene solo gli esempi della classe più popolosa, che nella maggior parte dei casi è quella meno interessante (ci interessa trovare i malati, non chi sta bene)!

Qualità dei Dati e Feature Engineering

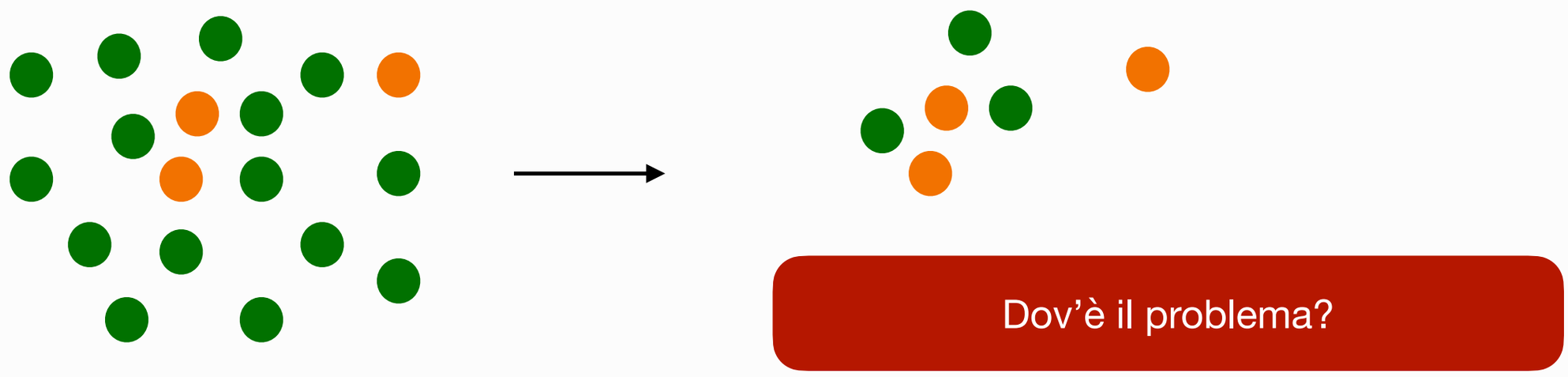
Ingegneria dei dati



Data Balancing: Insieme di tecniche per convertire un dataset sbilanciato in un dataset bilanciato.

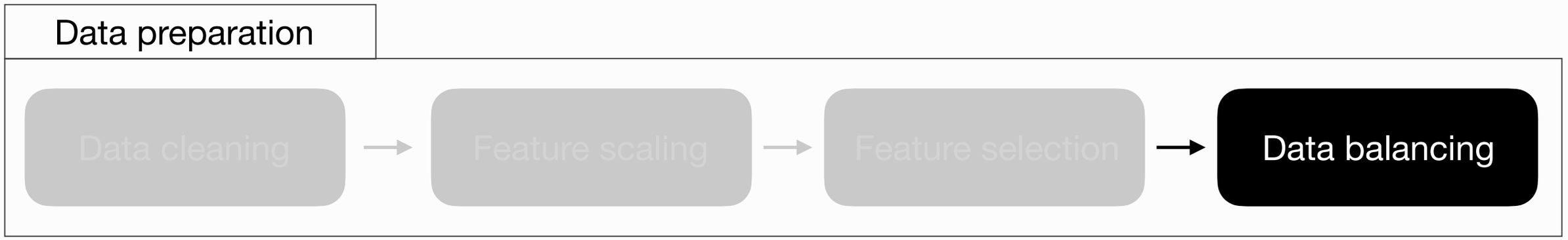
Sono due, principalmente, le tecniche applicabili: undersampling e oversampling.

- *Undersampling*: Metodo tramite il quale vengono casualmente eliminate un numero di istanze (righe) del dataset della classe di maggioranza.



Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Data Balancing: Insieme di tecniche per convertire un dataset sbilanciato in un dataset bilanciato.

Sono due, principalmente, le tecniche applicabili: undersampling e oversampling.

- *Undersampling*: Metodo tramite il quale vengono casualmente eliminate un numero di istanze (righe) del dataset della classe di maggioranza.

—> Se ho un numero eccessivamente basso di istanze della classe di minoranza, i dati non saranno sufficienti per apprendere né la classe di maggioranza originaria né tantomeno quella di minoranza.

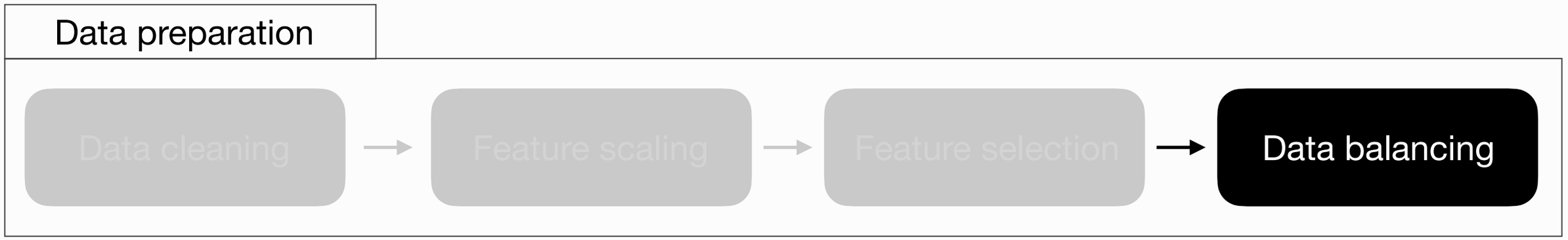
—> Il secondo problema sta nell'approccio. Un undersampling casuale potrebbe portare alla rimozione di istanze particolarmente rilevanti per l'apprendimento del modello.

olema?

—> Possiamo risolvere il secondo problema tramite una tecnica di undersampling basata su clustering. Ma lo vedremo più in là…

Qualità dei Dati e Feature Engineering

Ingegneria dei dati

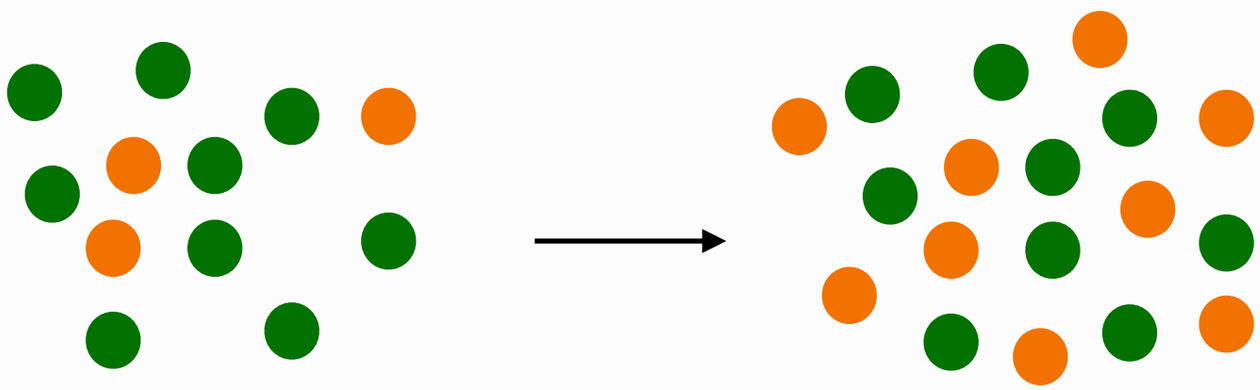


Data Balancing: Insieme di tecniche per convertire un dataset sbilanciato in un dataset bilanciato.

Sono due, principalmente, le tecniche applicabili: undersampling e oversampling.

- *Undersampling*: Metodo tramite il quale vengono casualmente eliminate un numero di istanze (righe) del dataset della classe di maggioranza.

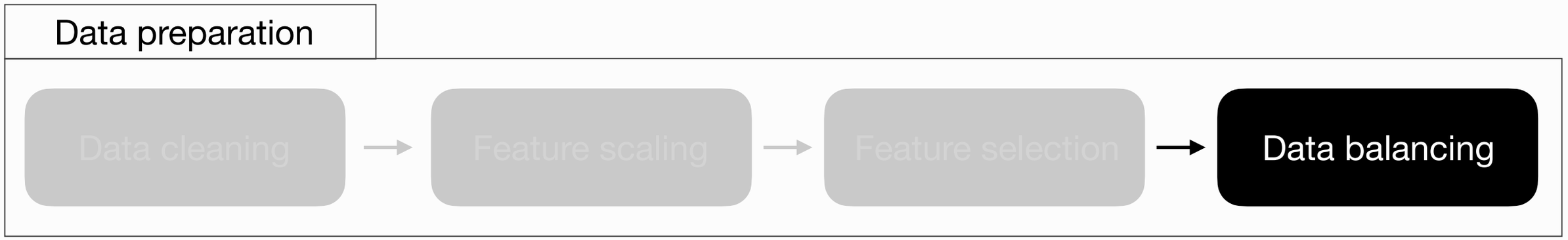
- *Oversampling*: Metodo tramite il quale vengono casualmente aggiunte un numero di istanze (righe) del dataset della classe di minoranza.



Dov'è il problema?

Qualità dei Dati e Feature Engineering

Ingegneria dei dati



Data Balancing: Insieme di tecniche per convertire un dataset sbilanciato in un dataset bilanciato.

Sono due, principalmente, le tecniche applicabili: undersampling e oversampling.

- *Undersampling*: Metodo tramite il quale vengono casualmente eliminate un numero di istanze (righe) del dataset della classe di maggioranza.

- *Oversampling*: Metodo tramite il quale vengono casualmente aggiunte un numero di istanze (righe) del dataset della classe di minoranza.

—> La duplicazione di istanze potrebbe creare overfitting!!! Il modello potrebbe saper imparare “a memoria” quali sono le istanze della classe di minoranza solo perché queste rappresentano delle copie che si ripetono più volte.

oblema?

—> Anche qui, vedremo come poter inventarci qualcosa di meglio tramite l’uso del clustering…

UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 15 - Qualità dei Dati e Feature Engineering