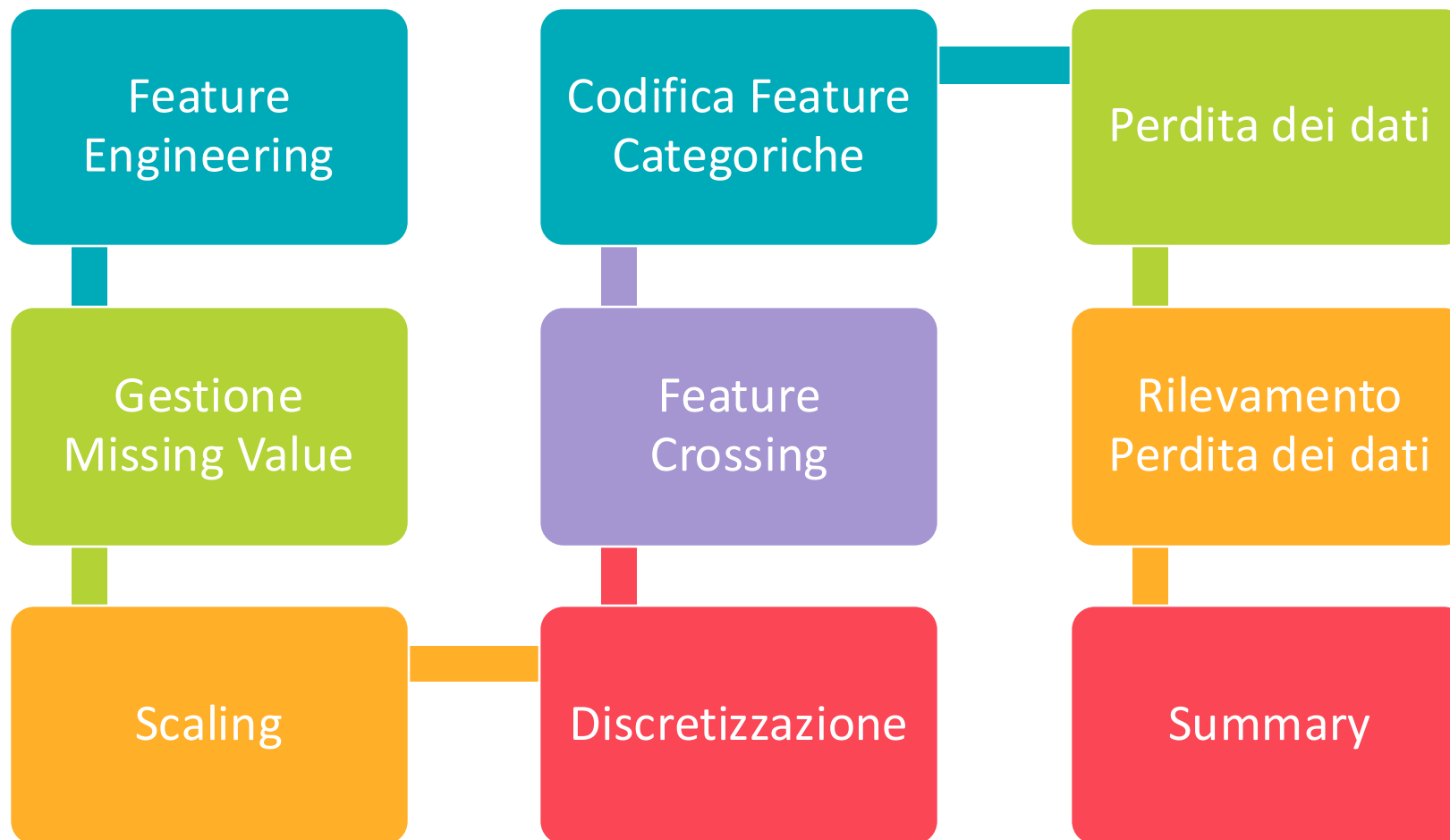


Machine Learning

4 - Feature Engineering



Outline



Feature Engineering

Introduzione Feature Engineering

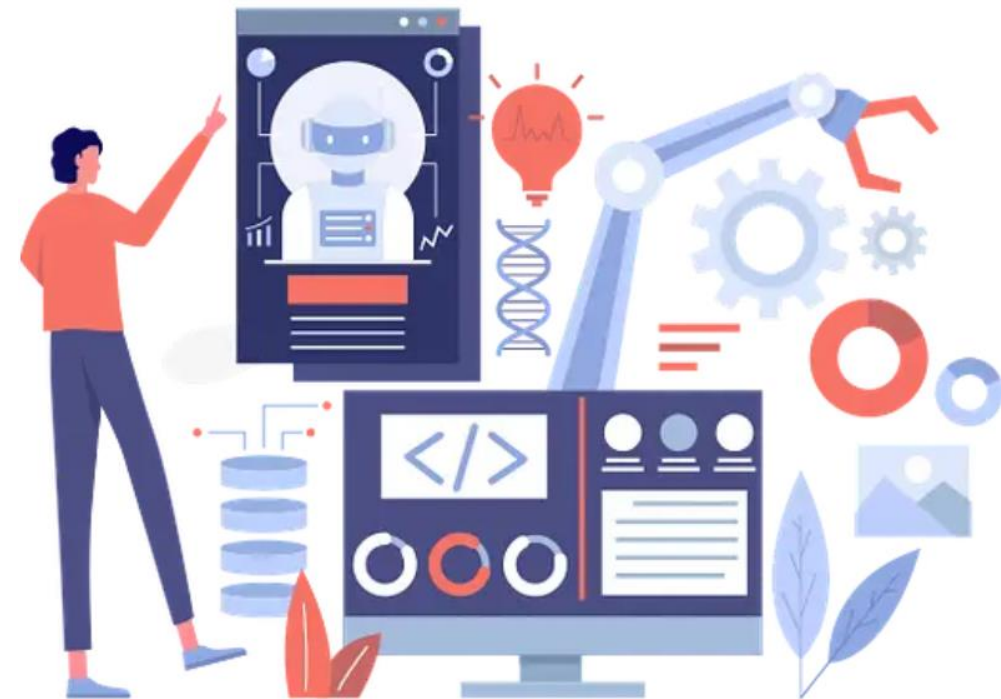
- Con il termine **Feature Engineering (FE)**, o ingegneria delle funzionalità, si intende l'insieme dei processi di **selezione, manipolazione, trasformazione** di variabili dai dati grezzi, in modo da poter utilizzare le funzionalità per l'addestramento e la previsione.
- È una parte **fondamentale** del processo di creazione di un modello di machine learning, il quale può richiedere più tempo rispetto a tutte gli altri processi.



Processi di Feature Engineering (1/2)

Possiamo suddividere il feature engineering in vari processi:

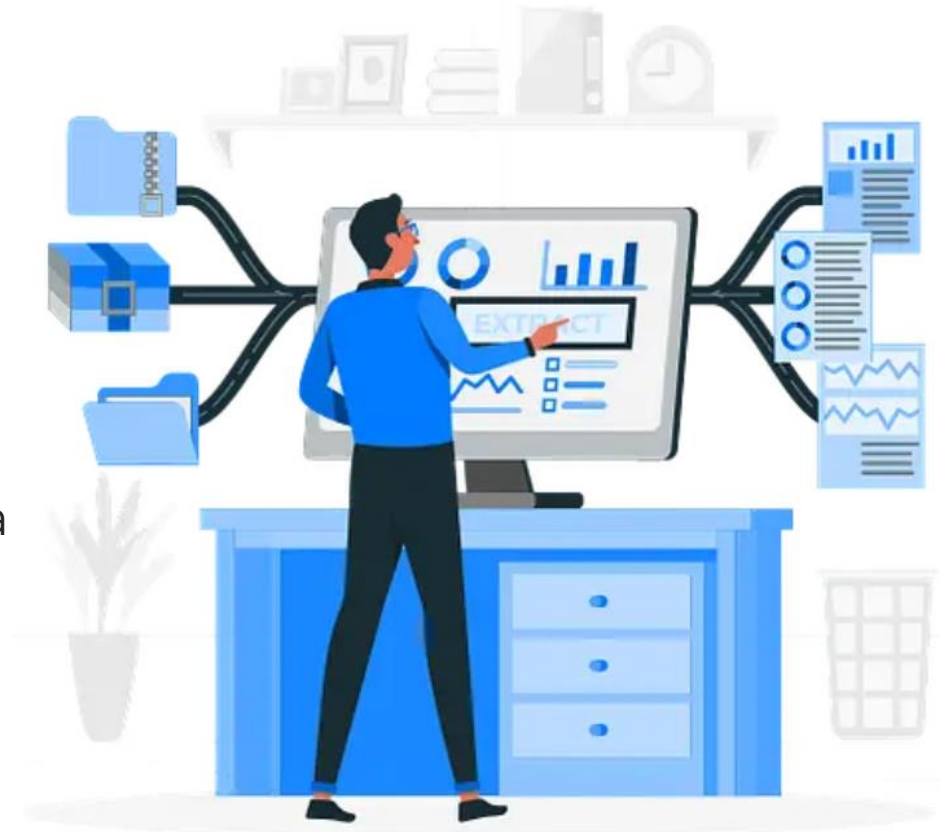
- **Creazione di funzionalità (feature)**
 - Implica la **realizzazione** di nuove **variabili** che saranno utili per il modello di ML.
 - Questo può significare **aggiungere** o **rimuovere** alcune funzionalità.



Processi di Feature Engineering (1/2)

- **Trasformazione**

- È definita come una funzione che **trasforma** le caratteristiche da una **rappresentazione a un'altra**.
- L'obiettivo è tracciare e visualizzare i dati, ed in caso **ridurre** o **incrementare** le funzionalità utilizzate, **accelerando l'addestramento** o facendo in modo da aumentare la precisione di un determinato modello.



Processi di Feature Engineering (2/2)

- **Estrazione delle funzionalità**

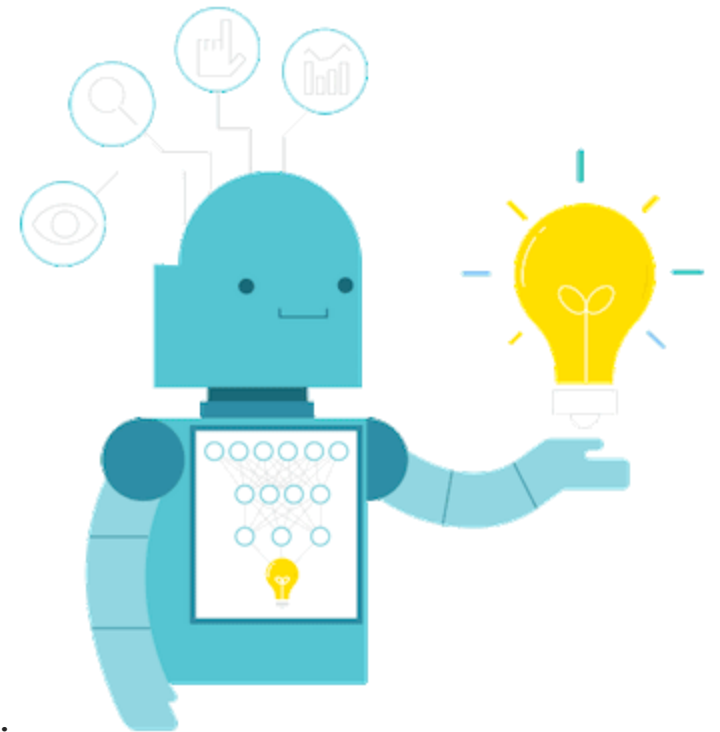
- Si definisce come l'insieme dei processi delle funzionalità che **estraggono feature** da un set di dati per identificare informazioni utili.
- L'obiettivo è fare ciò **senza distorcere** le relazioni originali o le informazioni significative, questo comprime la quantità di dati in quantità gestibili affinché gli algoritmi possano elaborarli.



Processi di Feature Engineering (2/2)

- **Analisi esplorativa dei dati**

- L'analisi esplorativa dei dati (EDA) è uno strumento potente e semplice che può essere utilizzato per **migliorare** la **comprensione** dei dati, con l'uso di rappresentazione grafiche, quali istogrammi, diagrammi, esplorandone le proprietà.
- La tecnica viene spesso applicata quando l'obiettivo è creare **nuove ipotesi** o trovare modelli nei dati.
- Viene spesso utilizzato su **grandi quantità** di dati qualitativi o quantitativi che non sono stati analizzati prima.



Operazioni principali

Operazioni di FE principali

- Le tecniche più usate per la progettazione delle caratteristiche includono:
 - **Gestione valori mancanti**
 - **Scaling o Ridimensionamento**
 - **Discretizzazione**
 - **Feature Crossing**
 - **Codifica delle features categoriche**
 - **Incorporazioni posizionali discrete e continue**

Le tecniche presentate sono solo alcune di quelle presenti in letteratura.

Il loro utilizzo è **strettamente legato al dominio del problema** e al progetto da realizzare.

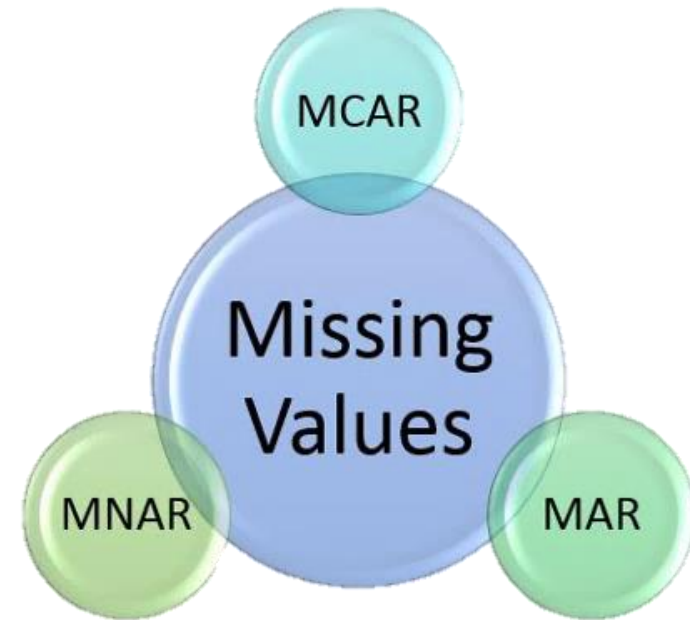
Gestione valori mancanti

- Quando si tratta di **preparare** i dati per l'apprendimento automatico, i valori mancanti, o missing value, sono uno dei problemi più tipici.
- Errori umani, interruzioni del flusso di dati, problemi di privacy e altri fattori potrebbero contribuire alla mancanza di valori.
- I valori mancanti possono **compromettere** le **prestazioni** dei modelli di machine learning.



Gestione valori mancanti

- I valori mancanti però non sono tutti uguali, possono essere distinti in tre categorie:
 - **Mancanti non casuali**
(Missing not at random (**MNAR**))
 - **Mancanti casuali**
(Missing at random (**MAR**))
 - **Mancanti del tutto casuali**
(Missing completely at random (**MCAR**))



Esempio di dati mancanti

- Consideriamo il task di prevedere se qualcuno **acquisterà una casa nei prossimi 12 mesi**, avendo a disposizione i seguenti dati:

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Esempio di dati mancanti

- Consideriamo il task di prevedere se qualcuno **acquisterà una casa nei prossimi 12 mesi**, avendo a disposizione i seguenti dati:

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Come si può notare, sono presenti **molti dati mancanti**. Ma non sono tutti uguali.

Dati Mancanti non casuali (MNAR)

- Il motivo per cui il valore manca varia a seconda **dell'attributo** a cui è **collegato**, e la **spiegazione** è a noi **sconosciuta**.
- Questa è la classe più **complessa** da analizzare.

ID	Gender	Annual income
1	A	150,000
2	B	50,000
3	A	100,000
4	B	
5	B	
6	A	50,000
7	B	60,000
8	B	10,000

- In questo esempio, possiamo notare che alcuni intervistati non hanno rivelato il proprio **reddito**.
- Dall'indagine potrebbe risultare che: gli intervistati che non lo hanno denunciato tende ad essere più alto di quello di chi ha denunciato.
- I valori legati al reddito **mancano** potrebbero mandare per ragioni **legate** al **valore stesso**.

Dati Mancanti casuali (MAR)

- Quando i dati sono MAR, il fatto che manchino i dati è **correlato** ai **dati osservati** ma non a quelli **non osservati**.
- In altre parole, è quando il motivo per cui manca un valore non è dovuto al valore stesso, ma è dovuto ad **un'altra variabile osservata**.

ID	Age	Gender
1		A
2	27	B
3		A
4	40	B
5	35	B
6		A
7	33	B
8	20	B

- **Questa è la classe di dati da cui partono tutte le ipotesi.**
- In questo esempio, potremmo notare che i valori dell'età spesso mancano per gli intervistati del genere "**A**", il che potrebbe essere dovuto al fatto che le persone di genere "**A**" in questo sondaggio non amano rivelare la propria età.

Dati Mancanti del tutto casuali (MCAR)

- Quando i dati sono MCAR, il fatto che manchino i dati è **indipendente dai dati osservati e non osservati**. In altre parole, non esistono differenze sistematiche tra i partecipanti con dati mancanti e quelli con dati completi.

ID	Job	Buy?
1	Engineer	No
2	Teacher	No
3		Yes
4	Engineer	Yes
5	Doctor	Yes
6	Teacher	No
7	Teacher	No
8	Student	No

- In questo esempio, si potrebbe pensare che sia dovuto ad un'altra variabile, in questo caso, bisognerà analizzare le risposte avute dal sondaggio.
- Dalle risposte, si è notato che il **progettista** si è dimenticato di inserire il valore.
- La sua mancanza non ha una ragione particolare.**
- Tuttavia questo tipo di scomparsa è molto rara.

Come rimuoviamo i Missing Values?

- Quando incontri valori mancanti, possiamo applicare due strategie:
 - **Cancellazione**: tecnica che rimuove dal dataset i valori mancanti
 - **Cancellazione di Colonne**
 - Rimozione di un'intera colonna
 - **Cancellazione su Righe**
 - Rimozione di uno o più record
 - **Imputazione**: tecnica che cerca di ricavare il valore da sostituire
 - **Imputazione numerica**
 - **Imputazione categorica**

Cancellazione di Colonne

La strategia più usata è andare ad **eliminare** la **colonna** nella quale sono presenti il maggior numero di Missing Values.

Prendendo in considerazione il precedente esempio

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Cancellazione di Colonne

La strategia più usata è andare ad **eliminare** la **colonna** nella quale sono presenti il maggior numero di Missing Values.

Prendendo in considerazione il precedente esempio



ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No
Missing Values		3/8	2/8	5/8	3/8	1/8	

Cancellazione di Colonne



ID	Marital status
1	
2	
3	Married
4	
5	Single
6	
7	Single
8	

- In questo esempio, manca **più** del **50%** dei valori per la variabile "**Stato civile**", quindi potrebbe essere utile rimuovere la variabile dal modello.
- Lo svantaggio di questo approccio è che potresti rimuovere **informazioni importanti** e **ridurre** la **precisione** del modello.
- Lo stato civile potrebbe essere altamente correlato all'acquisto di case, poiché le coppie sposate hanno molte più probabilità di essere proprietari di case rispetto alle persone single.

Non è una strategia sempre utile, ma varia a seconda della tipologia di variabile.

Cancellazione di Righe

Un ulteriore strategia è andare ad **eliminare il record** nella quale sono presenti il maggior numero di Missing Value.

Prendendo in considerazione il precedente esempio

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Cancellazione di Righe

Un ulteriore strategia è andare ad **eliminare il record** nella quale sono presenti il maggior numero di Missing Value.

Prendendo in considerazione il precedente esempio

	ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?	Missing Values
→	1		A	150,000		1	Engineer	No	2/8
→	2	27	B	50,000			Teacher	No	2/8
→	3		A	100,000	Married	2		Yes	2/8
→	4	40	B			2	Engineer	Yes	2/8
	5	35	B		Single	0	Doctor	Yes	1/8
→	6		A	50,000		0	Teacher	No	2/8
	7	33	B	60,000	Single		Teacher	No	1/8
→	8	20	B	10,000			Student	No	2/8

Cancellazione di Righe

La cancellazione dei record con maggior numero di Missing Value ha portato a **cancellare 6 righe su 8**, causando un ridimensionamento dei dati molto impattante e che porterà risultati disastrosi.

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
5	35	B		Single	0	Doctor	Yes
7	33	B	60,000	Single		Teacher	No

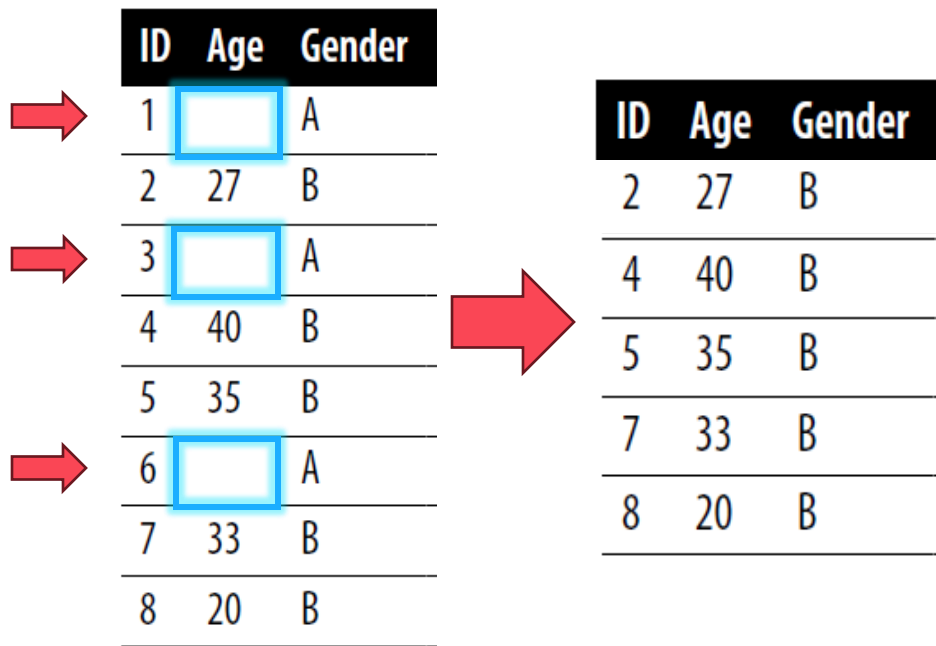
Questo metodo può funzionare quando:

- I valori mancanti **sono completamente casuali** (MCAR)
- Il **numero di record con valori mancanti è piccolo**, ad esempio inferiore allo 0,1%.

I dati e le informazioni cancellate possono essere importanti per l'addestramento del modello.

Cancellazione di Righe

La rimozione di righe di dati può **creare distorsioni** nel modello, soprattutto se i valori mancanti sono casuali (MAR).



ID	Age	Gender
1		A
2	27	B
3		A
4	40	B
5	35	B
6		A
7	33	B
8	20	B

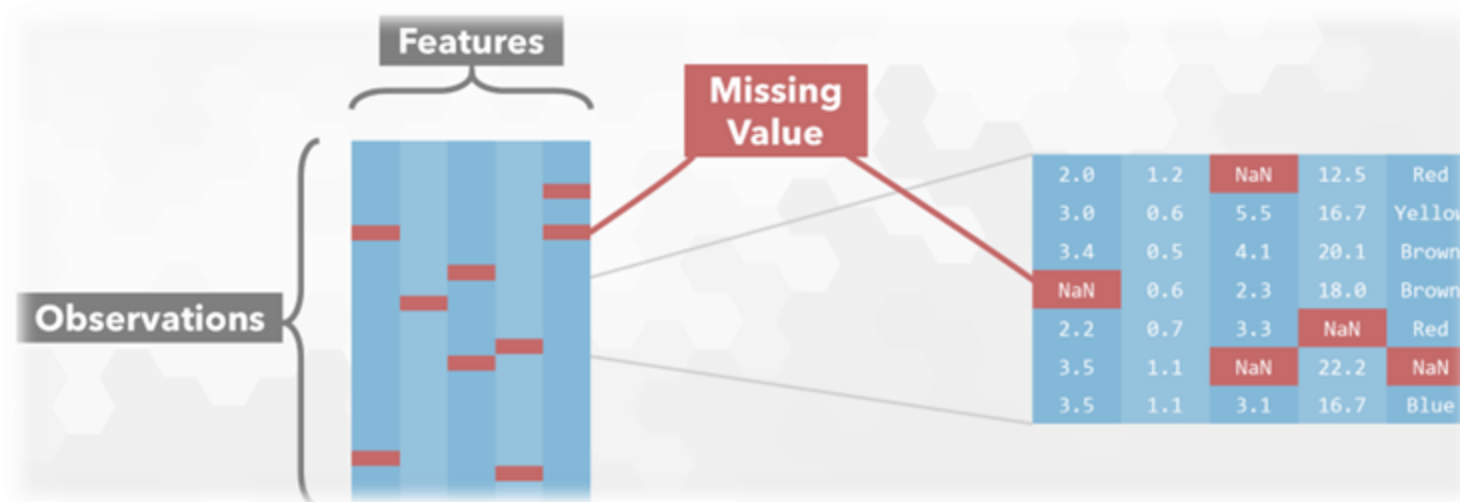
ID	Age	Gender
2	27	B
4	40	B
5	35	B
7	33	B
8	20	B

Ad esempio, se rimuovi tutti gli esempi di valori di età mancanti si andranno a rimuovere tutti gli intervistati di genere **A** dai dati e il modello **non sarà in grado di fare previsioni** valide per gli intervistati di quel genere.

Anche se l'eliminazione è allettante perché è facile da eseguire, l'eliminazione dei dati può portare alla **perdita di informazioni** importanti e introdurre **distorsioni** nel modello.

Imputazione

- L'obiettivo principale è la **gestione** di tutti quei dati **valori mancanti** che potrebbero portare, con la loro cancellazione, a **perdita di informazioni** o a **distorsioni** del modello.
- Essa consiste **nell'assegnazione** di un **valore sostitutivo** al fine di ripristinare la **“completezza”** dei dati.



Metodi di Imputazione

- In linea generale possiamo considerare 2 classi di metodi:
 - **Imputazione numerica**, valori numerici:
 - **Metodi deduttivi**
 - **Metodi deterministici e stocastici**
 - **Imputazione categorica**, valori categorici
- Tutti i metodi di imputazione per i valori mancanti (ad eccezione dei metodi deduttivi) **si basano** implicitamente o esplicitamente sull'assunzione che i dati siano di tipologia **MAR** (Missing at Random).

Imputazione: Metodi Deduttivi

- Si basa sulla possibilità di **sfruttare le informazioni presenti nel data set** in modo da poter **dedurre** il valore da sostituire al dato mancante da una o più variabili ausiliarie.
- L'applicazione del metodo è legata a **valutazioni soggettive** sul fenomeno oggetto di studio e spesso dipende dal grado di conoscenza del data set su cui si sta lavorando.
- Ad esempio se un record contiene una serie di cifre ed il loro totale, ma una delle cifre è mancante.

ID	A	B	C	Tot
1	10	5	Missing	20
2	15	5	10	30

Imputazione: Metodi Deduttivi

- Si basa sulla possibilità di **sfruttare le informazioni presenti nel data set** in modo da poter **dedurre** il valore da sostituire al dato mancante da una o più variabili ausiliarie.
- L'applicazione del metodo è legata a **valutazioni soggettive** sul fenomeno oggetto di studio e spesso dipende dal grado di conoscenza del data set su cui si sta lavorando.
- Ad esempio se un record contiene una serie di cifre ed il loro totale, ma una delle cifre è mancante, **questa può essere dedotta per sottrazione**.

ID	A	B	C	Tot
1	10	5	Missing	20
2	15	5	10	30



$$C = 20 - 10 - 5 = 5$$

Imputazione: Metodi Deduttivi

- Nella tabella sono presenti alcune generalità di utenti. Il missing value è la Patente di guida.

Codice Fiscale	Nome	Cognome	Data di Nascita	Patente B
LMBGLI06A01H703V	Giulio	Lamberti	01/01/2008	Missing
RSSMRC95H13F032C	Marco	Rossi	13/06/1995	SI

- Due strategie:**
 - Dalle le leggi italiane sappiamo che la patente B è fruibile solo dai 18 anni in poi.
 - Il valore da inserire quindi è collegato all'età dell'utente, la quale può essere dedotta:
 - Dal Codice Fiscale
 - Dalla Data di Nascita.

Imputazione: Metodi Deduttivi

- Nella tabella sono presenti alcune generalità di utenti. Il missing value è la Patente di guida.

Due strategie:

- Dalle le leggi italiane sappiamo che la patente B è fruibile solo dai 18 anni in poi.
- Il valore da inserire quindi è collegato all'età dell'utente, la quale può essere dedotta:
 - Dal Codice Fiscale
 - Dalla Data di Nascita.

Codice Fiscale	Nome	Cognome	Data di Nascita	Patente B
LMBGLI06A01H703V	Giulio	Lamberti	01/01/2008	Missing
RSSMRC95H13F032C	Marco	Rossi	13/06/1995	SI

→ $2024 - 2008 = 16$



16 anni = NO

Per una corretta applicazione di questi metodi è necessario avere un buon grado di conoscenza dei dati e delle relazioni esistenti al loro interno.

Imputazione: Metodi Deterministici e Stocastici

- Con questo metodo si sostituiscono tutti i valori mancanti con un **unico valore**.
 - Alcuni esempi sono:
 - **Media**
 - **Moda**
 - **Mediana**
 - **Etc.**
- I dati che prendono parte al processo di imputazione devono avere le **stesse caratteristiche**, non possono quindi essere diversi tra loro sia per unità di misura, non possono riferirsi a variabili diverse, ecc.
- Un'ulteriore strategia, più comune, è quella di inserire come valori mancanti con i valori di **default** di quella variabile

Imputazione Categorica

- Il missing value è sostituito con il **valore** con la **maggiore frequenza**.
- In questo caso, questo inserimento è più complesso, poiché potrebbe portare a **distorsioni** del dataset.
- Una pratica maggiormente usata è la creazione di **un'ulteriore categoria** in cui inserire i missing value.

ID	Job	Buy?
1		No
2	Teacher	No
3		Yes
4	Engineer	Yes
5		Yes
6	Teacher	No
7		No

Imputazione Categorica

- Il missing value è sostituito con il **valore** con la **maggiore frequenza**.
- In questo caso, questo inserimento è più complesso, poiché potrebbe portare a **distorsioni** del dataset.
- Una pratica maggiormente usata è la creazione di **un'ulteriore categoria** in cui inserire i missing value.

ID	Job	Buy?
1		No
2	Teacher	No
3		Yes
4	Engineer	Yes
5		Yes
6	Teacher	No
7		No



ID	Job	Buy?
1	Other	No
2	Teacher	No
3	Other	Yes
4	Engineer	Yes
5	Other	Yes
6	Teacher	No
7	Other	No

- È possibile utilizzare più tecniche contemporaneamente o in sequenza per gestire i valori mancanti per un particolare insieme di dati.

Imputazione

- Indipendentemente dalle tecniche utilizzate:

non esiste un modo perfetto per gestire i valori mancanti.

- Con la **cancellazione** rischi di perdere informazioni importanti o di accentuare pregiudizi.
- Con **l'imputazione**, rischi di iniettare i tuoi pregiudizi e aggiungere rumore ai tuoi dati.

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	NaN	9.0	Sunny
2	1/5/2017	28.0	NaN	Snow
3	1/6/2017	NaN	7.0	NaN
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	NaN	NaN	NaN
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

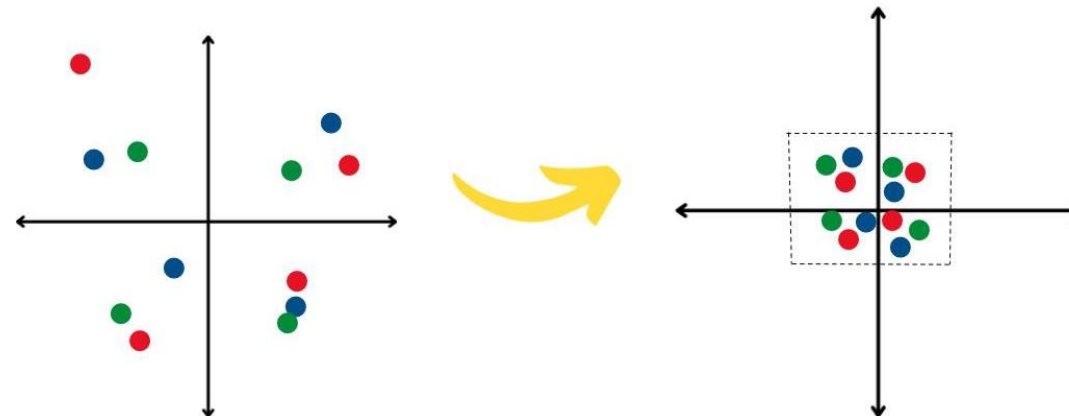


	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	28.0	9.0	Sunny
2	1/5/2017	28.0	7.0	Snow
3	1/6/2017	32.0	7.0	Rain
4	1/7/2017	32.0	8.0	Rain
5	1/8/2017	34.0	8.0	Sunny
6	1/9/2017	34.0	8.0	Cloudy
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

Scaling o ridimensionamento

Il **ridimensionamento delle funzionalità** è uno dei problemi più pervasivi nel ML, ma è una delle cose più importanti da risolvere. Le motivazioni sono varie:

- **Miglioramento delle prestazioni**
 - Diversi **metodi** di apprendimento automatico, inclusi **algoritmi** basati sulla discesa del gradiente, algoritmi basati sulla distanza (come i vicini k-NN) e SVM, **funzionano meglio** o **convergono** più **rapidamente**.



Scaling o ridimensionamento

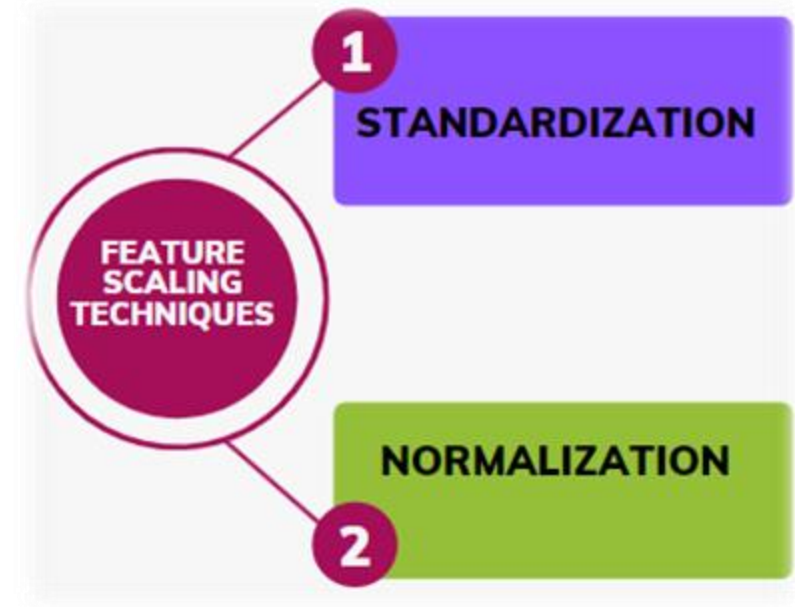
Prevenzione dell'instabilità ed apporto delle features

- La scalabilità garantisce che tutte le funzionalità siano su una **scala comparabile** e abbiano intervalli comparabili.
- Caratteristiche su larga scala possono **dominare** il processo di apprendimento e avere un impatto eccessivo sui risultati.



Scaling o ridimensionamento

- Possiamo evitare questo problema assicurandoci che ciascuna funzionalità **contribuisca equamente** al processo di apprendimento ridimensionando le funzionalità.
- Senza il ridimensionamento e la presenza di elementi con **scale radicalmente diverse** può comportare problemi di **overflow** o **underflow**.
- Diviene quindi necessario attuare queste tecniche.
- I processi di scaling sono noti come:
 - **Normalizzazione** delle funzionalità
 - **Standardizzazione** delle funzionalità



Scaling o ridimensionamento

Tenendo presente l'esempio precedente, notiamo come i valori di «**Età**» nei dati vanno da 20 a 40, mentre i valori della variabile «**Reddito annuo**» vanno da 10.000 a 150.000.

- Quando inseriamo queste due variabili in un modello ML, non capirà che 150.000 e 40 rappresentano cose diverse.
- Li vedrà entrambi semplicemente come numeri e, poiché il numero 150.000 è molto più grande di 40, potrebbe dargli maggiore importanza, indipendentemente da quale variabile sia effettivamente più utile per generare previsioni.

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

Scaling o ridimensionamento

Diviene quindi **indispensabile** fare in modo che tutte le variabili siano poste sullo stesso piano e ridimensionate.

- Sono disponibili vari metodi per il ridimensionamento:
- **Normalizzazione**
 - **Scaling nel range [0,1]**
 - **Scaling in range arbitrari [a,b] (es. [-1,1])**
- **Standardizzazione**

Scaling o ridimensionamento

- **Scaling nel range [0,1]**

- Data una variabile x i suoi valori possono essere ridimensionati per rientrare in nell'intervallo $[0,1]$ con la seguente formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Dove $\left\{ \begin{array}{l} \text{se } x \text{ è il valore massimo dai dati allora } x' \text{ sarà } 1; \\ \text{se } x \text{ è il valore minimo allora } x' \text{ sarà } 0. \end{array} \right.$

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

$$Age' = \frac{27 - 20}{40 - 20} = 0.35$$



$$AI' = \frac{150000 - 10000}{150000 - 10000} = 1$$

ID	Age	Annual income
1		1
2	0.35	0.285714
3		0.642857
4	1	
5	0.75	
6		0.285714
7	0.65	0.357143
8	0	0

Scaling o ridimensionamento

- **Scaling in range arbitrari [a,b] (es. [-1,1])**

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

Dove $\left\{ \begin{array}{l} \text{se } x \text{ è il valore massimo dai dati allora } x' \text{ sarà } b; \\ \text{se } x \text{ è il valore minimo allora } x' \text{ sarà } a. \end{array} \right.$

Scalare ad un intervallo arbitrario funziona bene quando non vuoi fare alcuna ipotesi sulle tue variabili.

Scaling o ridimensionamento

Se ritieni che le tue variabili possano seguire una distribuzione normale, potrebbe essere utile normalizzarle in modo che abbiano media zero e varianza unitaria.

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

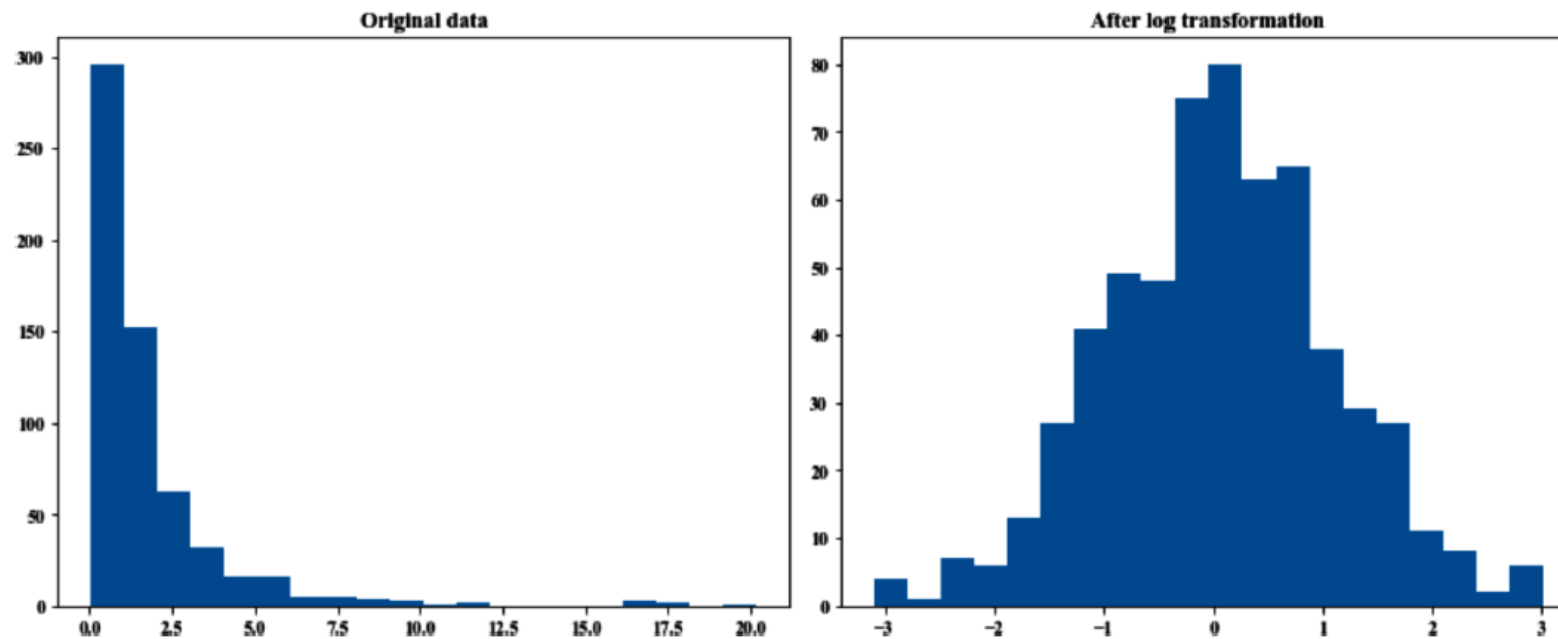
Dove $\begin{cases} \bar{x} & \text{è la media della variabile } x; \\ \sigma & \text{è la sua deviazione standard.} \end{cases}$

Questo processo è chiamato **standardizzazione**

Scaling o ridimensionamento

Nella maggioranza dei casi i modelli di ML si ritrovano ad essere addestrati con dati con una distribuzione distorta, la quale porta ad avere modelli non accurati e non performanti.

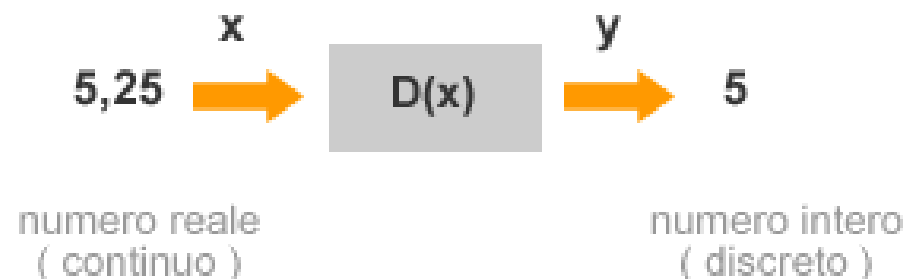
- Per mitigare l'asimmetria dei dati, una tecnica comune è la log transformation, la quale applica la funzione logaritmica (\log) alle tue variabili.



Discretizzazione

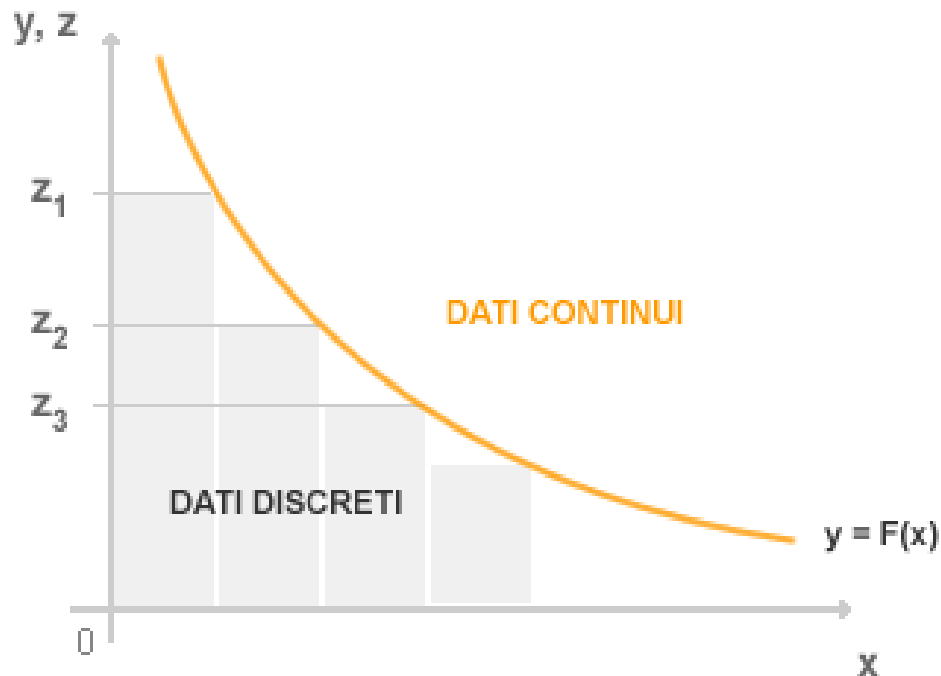
La discretizzazione è il processo di **trasformazione** di una **caratteristica continua** in una caratteristica **discreta**, noto anche come **quantizzazione** o **binning**.

- Si definisce come il processo di **raggruppamento** di **valori continui** di variabili in **intervalli contigui**.
- Questa procedura trasforma le variabili continue in variabili discrete ed è comunemente utilizzata nei processi di ML, nonché per addestrare modelli.



Discretizzazione

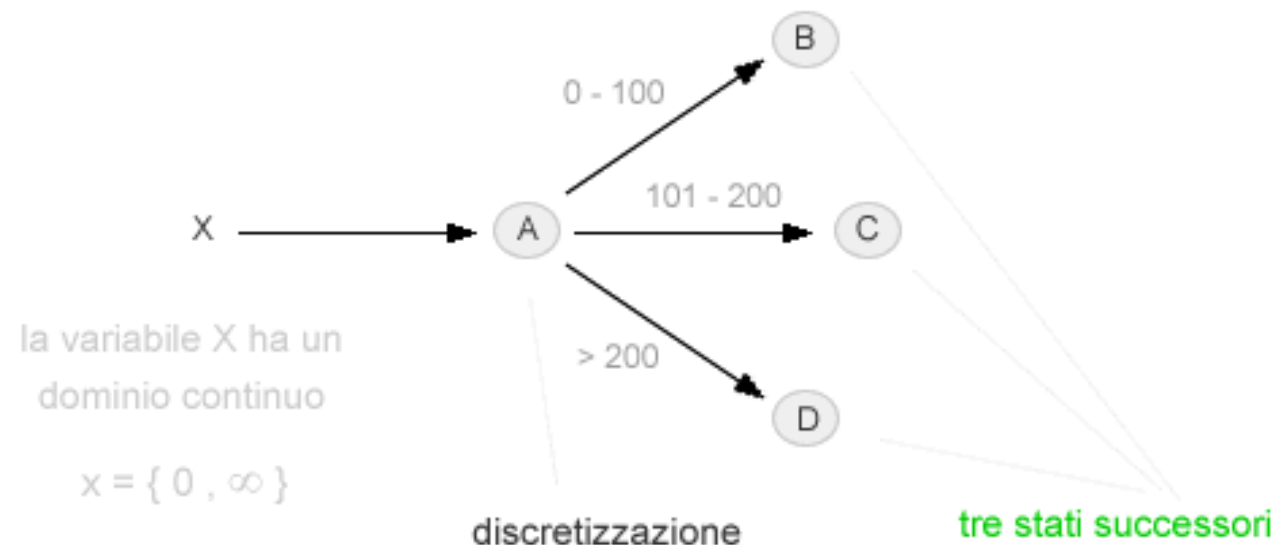
- La trasformazione dei dati continui comporta una **perdita di informazioni**.
- L'obiettivo di un algoritmo di discretizzazione è determinare il **minor numero** di **intervalli** possibile **senza perdere** in modo significativo le **informazioni**.



- Il compito dell'algoritmo diventa quindi quello di **determinare i punti limite** per tali **intervalli**.
- Nonostante la perdita di dati, la discretizzazione apporta **molti vantaggi** nelle fasi di elaborazione dei dati grezzi.

Vantaggi Discretizzazione

- Diversi modelli di **regressione e classificazione**, come gli alberi decisionali e Naive Bayes, **funzionano meglio** con valori discreti.
- La discretizzazione delle funzionalità continue può **accelerare** quindi il processo di **addestramento**.
- La discretizzazione può **ridurre** al **minimo** l'influenza dei valori **anomali** posizionandoli negli intervalli inferiori o superiori insieme ai restanti valori della distribuzione.



Metodi di Discretizzazione

Nel complesso, la discretizzazione delle funzionalità continue rende i dati più semplici, il processo di apprendimento più veloce e può produrre risultati più accurati.

I metodi più popolari sono

- **Discretizzazione ad uguale ampiezza**
- **Discretizzazione ad uguale frequenza**
- **Approcci non supervisionati**
- **Approcci supervisionati**



Metodi di Discretizzazione

Discretizzazione ad uguale ampiezza

Consiste nel **dividere l'intervallo** di valori continui in **k intervalli** di **uguale dimensione**.

- Data una variabile il cui range di valori va da 0 a 100:

$k=2 \rightarrow$ Gruppo 1 (0 – 50) , (50 -100);

$k=5 \rightarrow$ Gruppo 1 (0 – 20) , (20 -40), (40 – 60) , (60 -80), (80-100).

La discretizzazione di uguale ampiezza **non altera in modo drammatico** la distribuzione delle variabili.

Se una variabile è distorta prima della discretizzazione, sarà comunque distorta dopo la discretizzazione.

Possiamo valutare la distribuzione della variabile originale utilizzando gli istogrammi per la variabile continua e dopo la discretizzazione utilizzando i grafici a barre.

Metodi di Discretizzazione

Discretizzazione ad uguale frequenza

- La discretizzazione a frequenza uguale ordina la variabile continua in intervalli con lo **stesso numero** di **osservazioni**. La larghezza dell'intervallo è determinata dai **quantili**.
- La discretizzazione a frequenza uguale è particolarmente utile per le **variabili asimmetriche**, poiché **distribuisce equamente** le osservazioni sui diversi gruppi.

Discretizzazione con approcci non supervisionati

- Si tratta di tecniche **non supervisionate** perché trovano i limiti dell'intervallo in modo **autonomo**, senza l'aiuto del progettista
- Per creare intervalli o contenitori che raggruppano osservazioni simili, possiamo utilizzare **algoritmi** di **clustering**, come il **k-means**.
- Le partizioni sono i **cluster** identificati dall'algoritmo.
- La discretizzazione con k-means richiede un parametro k cioè il numero di cluster.

Metodi di Discretizzazione

Approcci supervisionati

- Ulteriori tecniche, come quelle basate sui **Decision Tree**, sono maggiormente utilizzate poiché possono **determinare automaticamente** i **punti** di **taglio** e il numero ottimale di divisioni.
- I metodi dell'albero decisionale discretizzano attributi continui durante il processo di apprendimento. Un albero decisionale **valuta tutti i possibili valori di una caratteristica** e seleziona il punto di taglio che massimizza la separazione delle classi utilizzando una metrica delle prestazioni come **l'entropia** o l'impurità **Gini**. Quindi ripete il processo per ciascun nodo della prima separazione dei dati e delle successive suddivisioni, **fino** al raggiungimento di un determinato **criterio** di **arresto**.
- Pertanto, gli alberi decisionali possono, in base alla progettazione, trovare l'insieme di punti di taglio che suddividono una variabile in intervalli con una buona coerenza di classe.

Metodi di Discretizzazione

- Tenendo presente l'esempio, notiamo come i valori del reddito annuale, in questo caso specifico sono molto pochi.
- In una situazione reale, un data scientist ha a che fare con centinaia o milioni di dati quando deve addestrare un modello.

**Conviene far imparare al
nostro modello un numero
infinito di possibili valori?**

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

Metodi di Discretizzazione

- Anche se, per definizione, la **discretizzazione** è **destinata** a caratteristiche **continue**, può essere utilizzata anche per caratteristiche **discrete**.
- Una buona pratica è il **raggruppamento**, così da far imparare al modello solo un numero finito di valori, il che è un compito molto più semplice da apprendere.

Un esempio di discretizzazione di variabili discrete:

- **Reddito annuale:**

Reddito inferiore: meno di 35.000

Reddito medio: tra 35.000 e 100.000

Reddito superiore: più di 100.000

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

Metodi di Discretizzazione

Un esempio di discretizzazione di variabili discrete:

- **Età:**

Meno di 18 anni

Tra i 18 e i 30

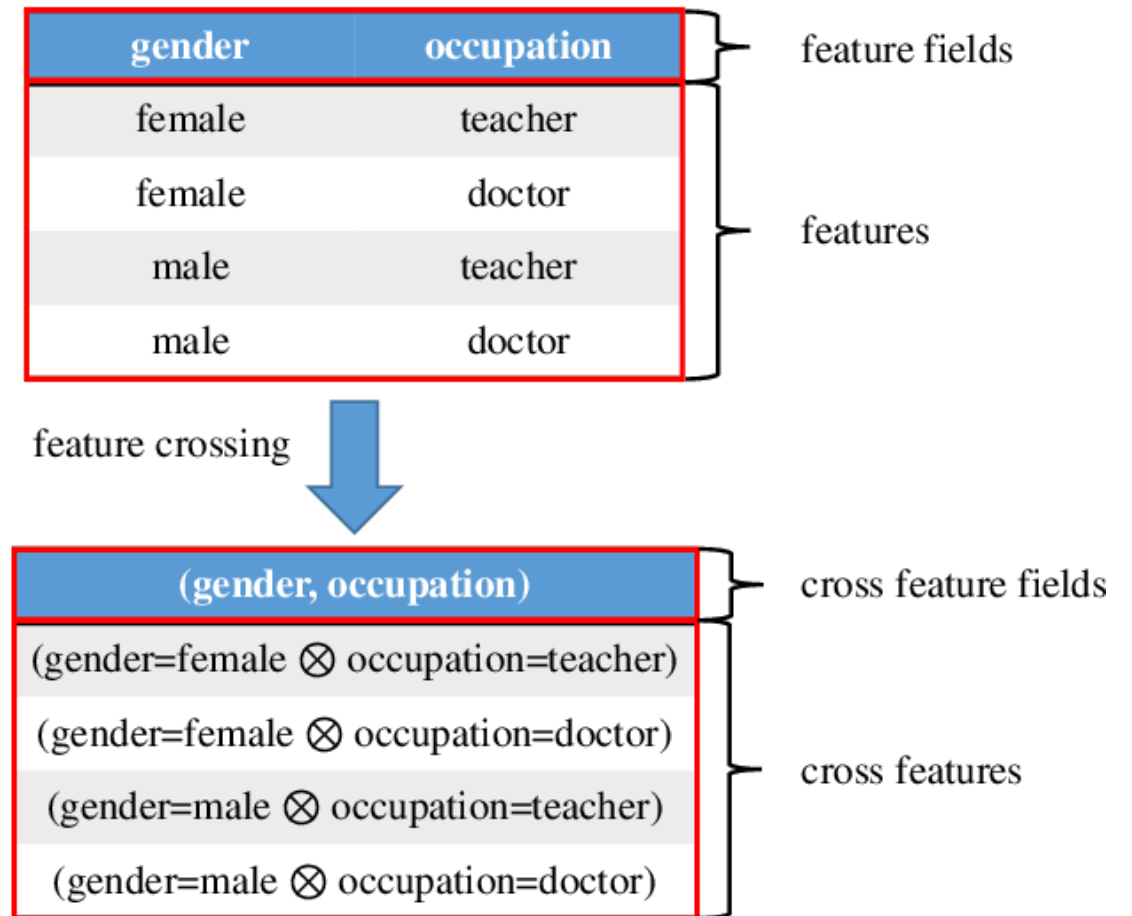
Tra i 30 e i 50

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

- Lo svantaggio è che si **introduce discontinuità** ai confini della categoria.
- **Scegliere i confini** delle categorie potrebbe non essere così facile.
- Puoi provare a tracciare gli **istogrammi** dei valori e scegliere i confini che hanno senso.

Feature Crossing **

- L'incrocio di feature è la tecnica per **combinare** due o più **feature** in modo da **generare nuove caratteristiche**.
- Questa tecnica è utile per modellare le relazioni non lineari tra nuove caratteristiche.



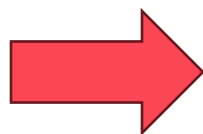
Feature Crossing

Prendiamo in considerazione l'esempio.

- **Sospetti che possa esserci una relazione non lineare tra stato civile e numero di figli, per l'acquisto di una casa?**

Combiniamoli per crearne una nuova feature chiamata «**matrimonio e figli**» da aggiungere.

ID	Marital status	Number of children
1	Married	1
2		
3	Married	2
4	Married	2
5	Single	0
6	Single	0
7	Single	0
8		



Marriage	Single	Married	Single	Married
Children	0	2	1	1
Marriage and children	Single, 0	Married, 2	Single, 1	Married, 1

Feature Crossing

- Poiché l'incrocio delle feature aiuta a modellare le **relazioni non lineari** tra le variabili, è essenziale per tutti i modelli che **non riescono ad apprendere** o sono **non apprendono bene** le **relazioni non lineari**, come la *regressione lineare*, la *regressione logistica* e i *modelli basati su alberi*.





Feature Crossing: Avvertimenti



- Immagina se abbiamo due feature **A** e **B** che ha **100** possibili valori ciascuno;
- Se combiniamo queste feature si avrà come risultato una caratteristica con:
 - $100 \times 100 = \mathbf{10.000}$ **valori possibili**.
- La loro combinazione, invece che migliorare, farà **regredire** il modello.
- Inoltre, se l'incrocio delle feature aumenta il numero di feature utilizzate dal modello, ciò potrebbe far soffrire il modello di **overfitting** o **underfitting**.

Codifica delle features categoriche

Nonostante ciò che si pensa, la codifica delle features categoriche può essere complessa.

- Prendendo in considerazione l'esempio precedente:

- **Reddito annuale:**

Reddito inferiore: meno di 35.000

Reddito medio: tra 35.000 e 100.000

Reddito superiore: più di 100.000

- In questo caso, se le categorie scelte sono state prese sulla

base di una determinata logica, quindi tenendo in

considerazione leggi o convenzioni, sarà difficile che queste

categorie possano cambiare nel tempo

ID	Age	Annual income
1		150,000
2	27	50,000
3		100,000
4	40	
5	35	
6		50,000
7	33	60,000
8	20	10,000

- **Tuttavia, nella realtà, è possibile che i gruppi possano subire modifiche**

Codifica delle features categoriche

- Prendiamo in considerazione la creazione di un sistema di raccomandazione per prevedere quali prodotti gli utenti potrebbero voler acquistare da Amazon
- Tra le feature più importanti c'è il **Brand** dei prodotti (possono essere migliaia).
- Ipotesi:
 - Non essendoci la possibilità, come per il **Reddito**, di suddividere in gruppi, si sceglie di tradurre ogni marchio, in un numero sequenziale, da 1 a n.
- Risultati:
 - Il tuo modello con questa configurazione ottiene massime prestazioni e si conferma come ottimo per il task di previsione degli acquisti

Con il passare del tempo, però, il modello perde di stabilità e di performance, diventando inutilizzabile.

Codifica delle features categoriche

Nuovi **Brand** si aggiungono ogni giorno ad Amazon, e il modello addestrato non li ha mai visti.

- Ipotesi:
 - Non essendoci la possibilità di prevedere i marchi che si aggiungeranno nel corso dei giorni, si decide di aggiungere una ulteriore classe «Sconosciuto», nel quale saranno convogliati tutti i marchi non conosciuti dal modello.
- Risultati:
 - Il modello con questa configurazione non si blocca, ma i venditori dei marchi classificati come «Sconosciuto», riferiscono che non sono trattati come gli altri, poiché scartati dalle previsioni. Il modello non consiglierà mai ad un cliente di comprare un prodotto di quella classe.

Codifica delle features categoriche

- Il modello, riaddestrato, torna a funzionare, ma come prima, le prestazioni iniziano a calare.
- I nuovi **Brand** arrivati sono di lusso, marchi affermati, e di scarsa qualità.
- Il modello però **tratta tutti i marchi allo stesso modo**, senza fare distinzioni, durante l'addestramento.
- **La risoluzione di questo problema è complesso e richiede una conoscenza approfondita del problema.**

Codifica delle features categoriche

- Il modello, riaddestrato, torna a funzionare, ma come prima, le prestazioni iniziano a calare.
- I nuovi **Brand** arrivati sono di lusso, marchi affermati, e di scarsa qualità.
- Il modello però **tratta tutti i marchi allo stesso modo**, senza fare distinzioni, durante l'addestramento.
- **La risoluzione di questo problema è complesso e richiede una conoscenza approfondita del problema.**
- Una strategia per la soluzione è il **Feature hashing**, anche conosciuto come, **hashing trick**.

Codifica delle features categoriche: hashing trick

- Si definisce come un metodo **veloce** ed **efficiente** in termini di spazio per vettorizzare le caratteristiche , ovvero **trasformare caratteristiche** arbitrarie in indici in un **vettore** o **matrice**.
- Funziona applicando una **funzione hash** alle **caratteristiche** e utilizzando i relativi valori **hash** direttamente come indici, anziché cercare gli **indici** in un array associativo.
- **Gli algoritmi di apprendimento automatico, tuttavia, sono generalmente definiti in termini di vettori numerici.**

Codifica delle features categoriche: hashing trick

- L'approccio comune è quello di costruire, al momento dell'apprendimento o prima, una rappresentazione del dizionario del vocabolario del training set e usarla per mappare la features negli indici, cioè il **valore hash** risultante diventerà **l'indice** di quella **categoria**.
- Il problema con questo processo è che tali **dizionari occupano** una **grande quantità** di spazio di archiviazione e aumentano di dimensioni man mano che il set di **training cresce**.

Perdita dei Dati

Perdita dei dati: Cause comuni

La **perdita di dati**, o data leakage, si verifica quando per creare il modello dovrebbero essere utilizzate **informazioni, non presenti**, al set di dati di addestramento.

- Queste informazioni aggiuntive possono consentire al modello di **apprendere** o conoscere qualcosa che altrimenti non saprebbe e, a sua volta, **invalidare** le **prestazioni** stimate.



Perdita dei dati: Cause comuni

La perdita dei dati può verificarsi a nostra **insaputa**, infatti, nella maggioranza dei casi sembra le prestazioni del modello possono sembrarci buone, ma in realtà non possiamo sapere nelle realtà come e cosa ha imparato dai dati.

La perdita è spesso subdola e indiretta, rendendola difficile da rilevare ed eliminare.

La perdita può indurre uno statistico o un modellista a selezionare un **modello non ottimale**, che potrebbe essere superato da un **modello privo di perdite**.



Perdita dei dati: Cause comuni

Le problematiche principali possono essere suddivise in:

- **Suddivisione casuale dei dati**
- **Ridimensionamento**
- **Imputazione dei dati mancanti**
- **Duplicazione di dati**
- **Perdita di gruppo o di generalizzazione**

Suddivisione causale dei dati

Prima di ogni addestramento di un modello di ML, un'operazione fondamentale da fare sui dati è scelta di chi sarà colui che **addestrerà** il modello e chi lo andrà **testare**.

Questo metodo è chiamato **Split dei Dati**, e riguarda la **suddivisione** dei **dati**, sulla base di qualche regola, per la generazione di due nuovi set di dati:

- **Training** (Train) , cioè il set di dati più corposo che addestrerà il modello.
- **Testing** (Test), cioè il set di dati che andrà a testare le prestazioni.



Suddivisione causale dei dati

Il modo più semplice per suddividere il set di dati di in set di training e testing si basa sulla legge di **Pareto 80/20**, ma sono presenti molte varianti, a scelta del progettista.

In base a questa si sceglie di suddividere il dataset **80% in Train e 20% in Test**.



I dati sono **suddivisi casualmente** e messi nei due insiemi senza alcuna precisa regola di suddivisione.

Questo primo passaggio, è una causa comune di perdita di dati.

Suddivisione causale dei dati

Sappiamo che all'interno dei dati possono esserci **correlazioni** basate su **incroci** di **variabili**, o **correlazioni** basate sul **tempo**.

A volte la correlazione tra le variabili è **evidente**, ma nella maggior parte dei casi è il progettista che, **conoscendo** il problema ed i **dati**, deve capire come meglio evitare la **perdita di informazioni**.



Suddivisione causale dei dati

Task di predizione dei prezzi delle azioni settimanali

Una dataset che contenente variabili che descrivono **azioni** significa che in questi dati il **tempo** in cui sono generati **influisce** sulla distribuzione delle etichette di appartenenza.

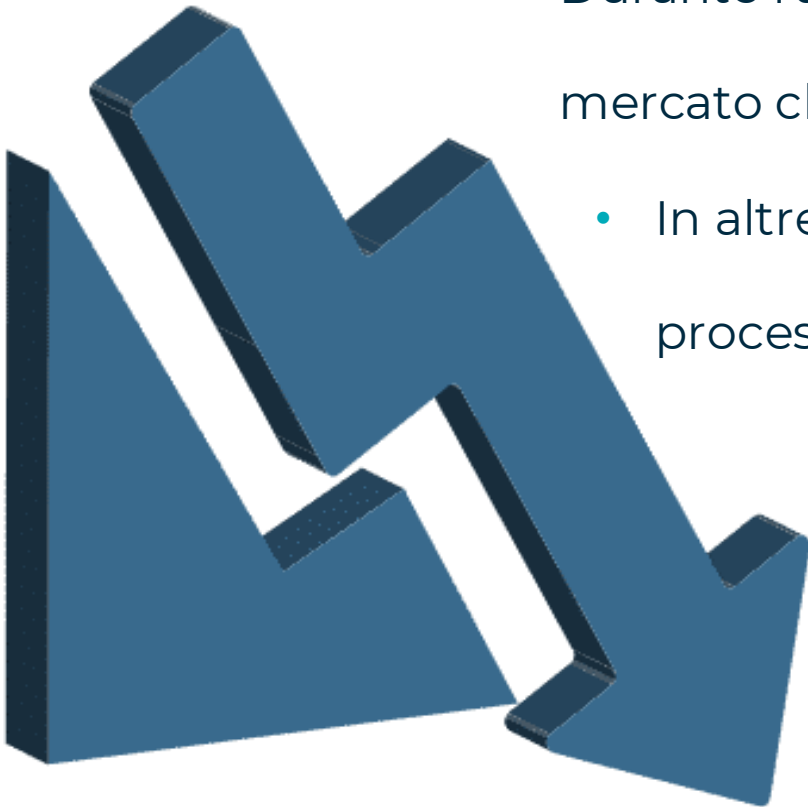
- Quando si creano modelli per prevedere i futuri prezzi, ci si aspetta che si voglia suddividere i dati in base al tempo.
- Se addestriamo il modello sui dati dei primi sei giorni e lo testiamo con i dati del settimo giorni, ci aspettiamo che:
 - Se oggi il 90% dei titoli tecnologici crolla, è molto probabile che scenda anche il restante 10%.



Suddivisione causale dei dati

Task di predizione dei prezzi delle azioni settimanali

- Se suddividiamo in **modo casuale**, i prezzi del settimo giorno verranno sparsi tra train e test.
 - Durante l'addestramento il modello riceverà dati di condizioni di mercato che non riguardano quel giorno.
 - In altre parole, le informazioni dal futuro vengono introdotte nel processo di formazione.
 - Le prestazioni del modello in se potrebbero non essere impattate da questa suddivisione, il problema si **aggraverà** con l'arrivo di nuovi dati e nuove predizioni da effettuare per il mercato.



Suddivisione causale dei dati

Predizione se un utente farà clic su una canzone suggerita

Considera il compito di prevedere se qualcuno farà clic su un suggerimento relativo a una canzone.

Se qualcuno ascolterà una canzone dipende:

- **Dai suoi gusti musicali**
- **Dal trend musicale del giorno**
- ...

Il dataset così fornito conterrà tutti i gusti musicali, le tendenze giornaliere e mensili dei vari artisti.



Suddivisione causale dei dati

Predizione se un utente farà clic su una canzone suggerita

Se un artista muore un giorno o ha uno scandalo, gli utenti saranno molto più curiosi nel capire chi sia e saranno molto più propensi ad ascoltarlo.

- In questo caso, nel set di dati potremmo avere tendenze di artisti o generi, non sulla base dei trend giornalieri, ma sulla base di fatti accaduti nel tempo, o in quel determinato giorno.
- Includendo campioni solo di un determinato giorno nel train, le informazioni sulla tendenza musicale che il modello andrà a predire saranno non veritiere e guidate dalle informazioni su cui è stato addestrato.

Suddivisione causale dei dati

È buona prassi, quindi, studiare i dati, la loro provenienza e come sono stati creati e quali sono le loro correlazioni.

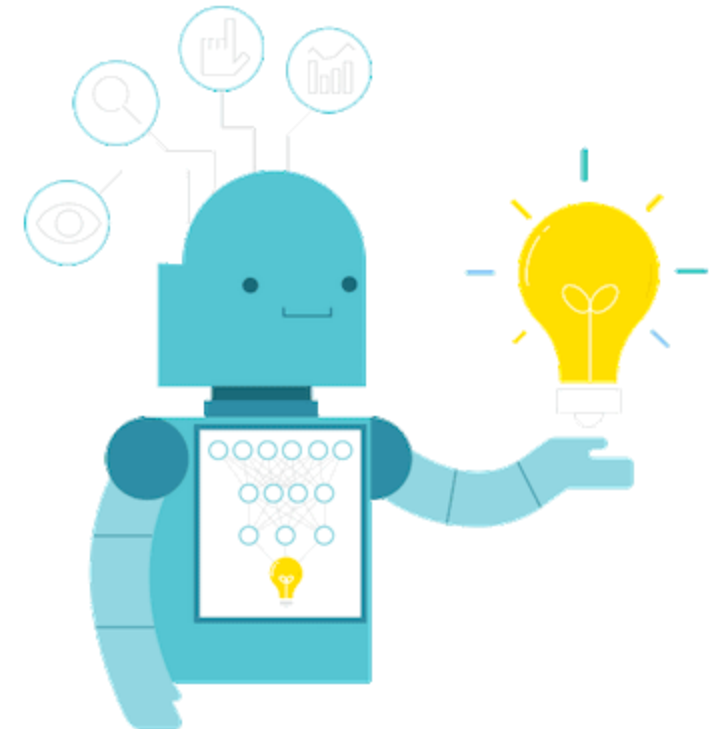
Riproponendo l'esempio precedente, se abbiamo un set di dati per la predizione dei prezzi delle azioni, e i dati sono correlati nel tempo, diviene conveniente, suddividerli tenendo conto delle implicazioni dei prezzi durante le settimane.

Train split					
Week 1	Week 2	Week 3	Week 4	Week 5	Valid split
X11	X21	X31	X41	X51	
X12	X22	X32	X42	X52	
X13	X23	X33	X43	X53	Test split
X14	X24	X34	X44	X54	
...	

Ridimensionamento

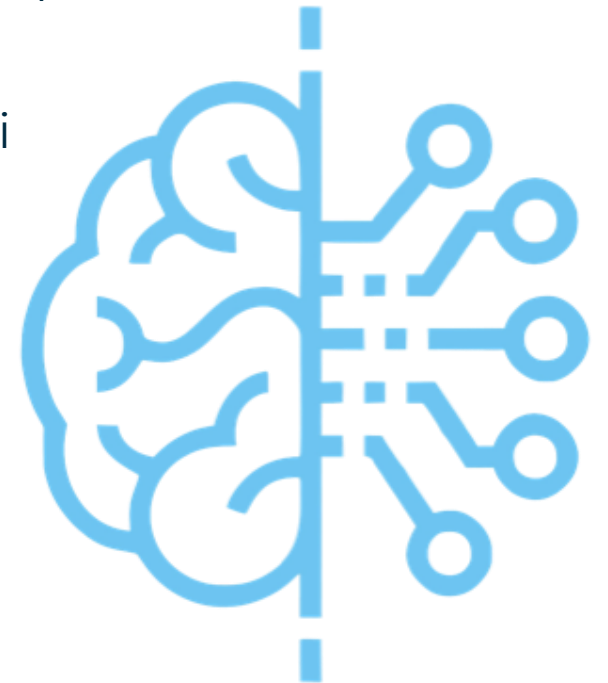
Il ridimensionamento è una parte fondamentale di pre-elaborazione dei dati durante i vari processi di ML.

- Un errore comune ed una causa di perdita di dati è l'applicazione delle tecniche di **scaling** utilizzando **l'intero set** di dati.
- La normalizzazione o la standardizzazione tra le istanze deve essere eseguita dopo aver suddiviso i dati di training e di test, utilizzando solo i dati di **training**.
- Questo perché il **test svolge il ruolo di dati nuovi e invisibili**, quindi non dovrebbe essere accessibile nella fase di training.



Ridimensionamento

- L'utilizzo di qualsiasi informazione proveniente dal set di test prima o durante l'addestramento è un **potenziale pregiudizio** nella valutazione della prestazione.
- Quando si normalizza, poi, il test set, si devono applicare i parametri di scaling precedentemente ottenuti dal training così come sono.
- Non ricalcolarli sul test set, perché sarebbero incoerenti con il modello e ciò produrrebbe previsioni errate.



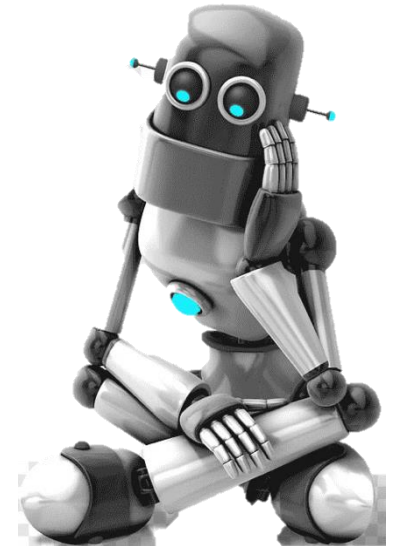
Imputazione dei dati mancanti

- Un modo comune per gestire i valori mancanti di una caratteristica è la **sostituzione** del missing value con la **media** o la **mediana** di tutti i valori presenti.
- Potrebbero **verificarsi perdite** se la media o la mediana vengono calcolate utilizzando **tutti i dati** disponibili.



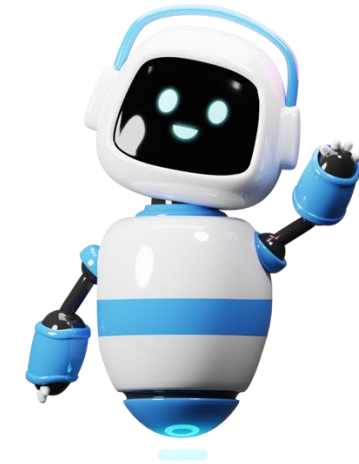
Imputazione dei dati mancanti

- Anche in questo caso, i dati si test, sono sconosciuti al modello, e dovranno essere presi in considerazione sono i dati del train appena creato.
- Questo tipo di perdita è simile al tipo di perdita causata dal ridimensionamento e può essere prevenuta utilizzando solo le statistiche del train per inserire i valori mancanti in tutte le suddivisioni.



Duplicazione di dati

- Se nei dati sono presenti duplicati o quasi duplicati, la mancata rimozione degli stessi prima della suddivisione dei dati potrebbe causare la comparsa degli stessi campioni sia nelle suddivisioni di training che di test.
- **La duplicazione dei dati è abbastanza** comune nel settore ed è stata riscontrata anche in set di dati di ricerca popolari.



Duplicazione di dati

- La duplicazione dei dati può derivare dalla raccolta di dati o dalla fusione di diverse fonti di dati o a causa dell'elaborazione dei dati.

- Una delle tecniche maggiormente impattanti è il

Sovracampionamento.

- Per evitare questa tipologia di problemi è bene controllare sempre il set di dati, producendo analisi e statistiche.



Perdita di gruppo o di generalizzazione

- La perdita di gruppo si verifica quando un gruppo di **osservazioni condividono** una **forte correlazione** tra di loro, e il progettista decide di dividerle **casualmente** tra train e test.
 - Ad esempio, dato un dataset di pazienti con riferimenti alle TAC polmonari di due settimane.
 - In questo caso un paziente potrebbe avere due TAC a una settimana di distanza, che probabilmente hanno le stesse etichette sul fatto che contengano segni di cancro ai polmoni.
 - Data la scelta del progettista, è possibile che le due TAC siano suddivise una in train e l'altra in test.

Perdita di gruppo o di generalizzazione

- Questo tipo di perdita è comune per le attività di object detection che contengono foto dello stesso oggetto scattate a millisecondi di distanza.
- È difficile evitare questo tipo di perdita di dati senza capire come sono i dati siano stati generati.
- **È bene svolgere sempre analisi approfondite sui dati a disposizione, così da capire a fondo il problema e sapere come gestire i dati a disposizione.**



Perdita dal processo di generazione dei dati

- Non esiste un modo infallibile per evitare questo tipo di perdita di dati, ma si può mitigare il rischio tenendo traccia delle fonti dei tuoi dati e comprendendo come vengono raccolti ed elaborati.
- **È buona prassi trattare dati da diverse fonti in modo diverso, anche durante i processi di scaling.**



Rilevamento della perdita di dati

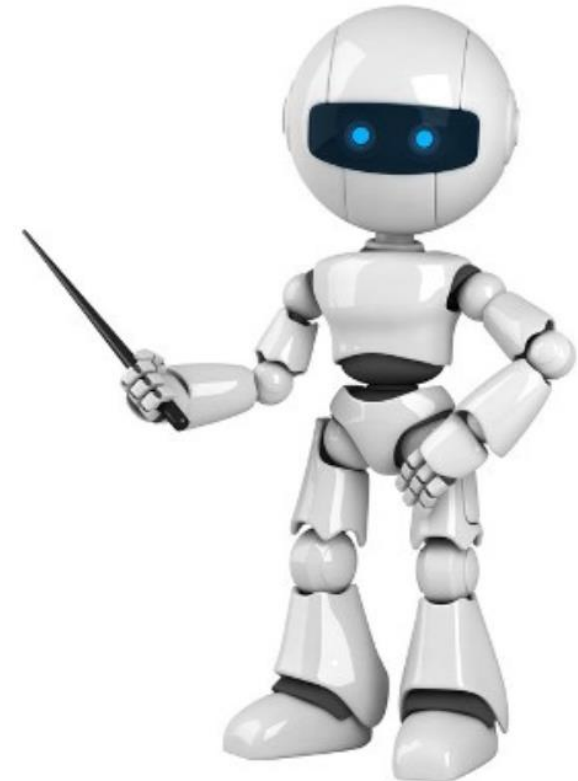
- La perdita di dati è può verificarsi in molte fasi durante le fasi di gestione dei dati:
 - Generazione dei dati
 - Raccolta dei dati
 - Campionamento dei dati
 - Suddivisione e scaling
 - Elaborazione dei dati

È importante monitorare la perdita di dati durante l'intero ciclo di vita di un progetto ML.



Rilevamento della perdita di dati

- Per mitigare la perdita di dati è di vitale importanza condurre **analisi profonde** sui dati che partecipano al problema.
- Il progettista, si ipotizza, sia informato sul **dominio** del **problema**, e che sia a conoscenza della **composizione** dei **dati**, della sorgente da cui provengono.



Rilevamento della perdita di dati

- Nel caso delle features che appartengono ai dati, è bene **analizzarle** in **profondità**, e ipotizzare ogni variazione, positiva e negativa, che queste possono fornire al **modello**.
- Se la **rimozione** di una caratteristica causa un **peggioramento** significativo delle prestazioni, è buona norma indagare sul **motivo** per cui quella caratteristica è così importante.

Rilevamento della perdita di dati

- Se si dispone di una quantità **enorme di feature** potrebbe essere impossibile eseguire studi di ablazione su ogni possibile combinazione di esse.
- Può essere utile eseguire occasionalmente studi di ablazione con un sottoinsieme di feature che si sospetta maggiormente



Buone caratteristiche ingegneristiche

- In genere, l'aggiunta di nuove feature rappresentative nei dati porta, di solito, ad un incremento delle prestazioni dei modelli.
- D'altra parte, però, l'aggiunta comporta un incremento di tempo per l'esecuzione e un aumento di memoria da utilizzare.



Buone caratteristiche ingegneristiche

- Inoltre, avere troppe feature può essere negativo sia durante l'addestramento che durante l'utilizzo del modello per i seguenti motivi:
 - Maggiori sono le feature di cui si dispone, maggiore è la possibilità di ottenere:
 - **Perdita di dati**
 - **Overfitting**
 - **Aumento di risorse richiesto**

**Come valutare l'importanza di
una feature?**

Buone caratteristiche ingegneristiche

- Ci sono notevoli tecniche utili a valutare l'importanza di una caratteristica.
- Sono algoritmi complessi e si differenziano a seconda del problema.
- L'obiettivo è valutare quanto un modello si deteriora se una o più caratteristiche vengono rimosse.
- Un pacchetto disponibile open source è **InterpretML**, in particolare **SHAP** (SHapley Additive exPlanations).
- Esso non solo **misura l'importanza** di una caratteristica per un intero modello, ma misura anche il **contributo** di ciascuna **caratteristica** alla previsione specifica di un modello.

Summary

Summary

La progettazione delle feature è una pratica complessa e dinamica.

Questa varia a seconda del dominio del problema e in base alle scelte del progettista.

Un riepilogo delle migliori pratiche:

- Dividere i dati sulla base della composizione e della correlazione dei dati e non in modo casuale
- Ridimensionare i dati è una pratica che non deve includere i dati di test, poiché saranno sconosciuti al modello che dovrà essere addestrato
- Per la gestione dei valori mancanti è bene tenere conto solo dei dati di training, senza considerare le statistiche sui dati di test.

Summary

Un riepilogo delle migliori pratiche:

- Comprendere come i dati vengono generati, raccolti, elaborati e da quali sorgenti essi provengano
- È buona prassi prendere in considerazione i consigli di un esperto del dominio
- Comprendere le feature migliori e valutare il loro apporto al modello, attraverso la loro combinazione o la loro rimozione

Nella maggior parte dei progetti ML reali, il processo di raccolta dei dati e di progettazione delle feature continua finché i modelli sono in produzione.