Appunti - Machine Learning

Alberto Filosa

30/6/2020

Indice

1 Data

In questo capitolo si tratteranno argomenti riguardanti la tipologia, il modo di esplorare, di valutare e come agire sui Missing Data e come fare pre-processing sui dati. Esistono diversi tipi di dati:

- *Categoriale* (quantitativa):
  - Nominale: i valori sono solo nomi differenti e fornisce informazioni solo sulla distinzione di un livello da un altro;
  - Ordinale: i valori sono ordinati mediante una scala;
- *Numerica* (quantitativa):
  - Intervallare: la differenza tra valori è significativo;
  - Rapporti: sia la differenza che il rapporto sono significativi.

Una variabile può essere:

- *Discreta*, un valore finito o contabile. Essa può essere:
  - Categoriale;
  - Numerica;
  - Binaria;
- *Continua*, un valore appartenente all'insieme dei numeri Reali.

1.1 Exploration

Esistono diversi modi per descrivere le statistiche relative ad un dataset:

- Media;
- Moda;
- Mediana;
- Quantile e Percentile;
- Range e Range Interquartile;
- Varianza e Deviazione Standard;
- Matrice di Varianze-Covarianze;
- Coefficiente di Correlazione Lineare.

Per la visualizzazione grafica si possono utilizzare sia istogrammi che boxplot.

1.2 Missing Replacement

Molteo spesso sono presenti all'interno di un dataset dei *Missing Values*; i motivi possono essere molteplici:

- I valori non sono misurabili;
- I valori non sono rilevanti alla prima collezione dei dati;
- Si hanno degli errori di salvataggio del dataset;
- I valori sono inconsistenti rispetto alle altre variabili.

Bisogna ricordare che è molto importante risolvere questo tipo di problemi in modo da ottenere risultati robusti. Esistono diversi modi per lavorare sui valori mancanti:

- Rimuovere il record se è presente un valore mancante;
- Imputare manualmente il valore mancante (**NON consigliato**);
- Usare una costante globale;
- Usare la moda per sostituire il valore mancante;
- Usare la media;
- Usare la media condizionata, ovvero la media di righe che hanno lo stesso valore di un attributo;
- Inserire il valore più probabile attraverso una semplice regressione lineare oppure utilizzando anche modelli più complessi.

1.3 Pre-Processing

Il *Pre-Processing* è un'area di strategia e di tecniche che consente di lavorare con i dati, per rendere le successive analisi più efficienti.

L'***Aggregazione*** consiste nel combinare due o più record in una singola riga utilizzando la media o la somma dei valori. Se la variabile è categoriale, si aggregano come una tupla. I vantaggi del suo utilizzo sono molteplici:

- Il dataset è più compatto e pesa meno dal punto di vista computazionale;
- Si ha un livello di dettaglio più generale dei dati;
- Riduce la varianza, aumentando l'efficienza, ma diminuendo l'interesse nei dettagli dei dati.

Il ***Campionamento*** consiste nel selezionare diversi record casualmente, utilizzando meno righe. Il campione, però, deve essere rappresentativo, ovvero le stesse proprietà del dataset originale, ad esempio, confrontando la media del campione con quello della popolazione. Esistono due principali modalità di campionamento:

- ***C***ampionamento ***C***asuale ***S***emplice (CCS): ogni riga del dataset ha la stessa probabilità di essere estratta nel campione. Essa si divide in estrazione:
  - Con Reinserimento (CR);
  - Senza Reinserimento (SR).
- ***C***ampionamento ***C***asuale ***St***ratificato: utile per variabili qualitative, si compie un campionamento in base ai livelli della variabile considerata. Il campionamento può essere:
  - Proporzionale, in base alla percentuale di stratificazione di una variabile;
  - Numero equo per ogni stratificazione della variabile.

Una volta selezionata la tipologia di campionamento, si sceglie la grandezza del campione: un ampio campione aumenta la probabilità di rappresentatività, andando ad eliminare però numerosi vantaggi del campionamento.

Molte volte capita di analizzare dataset con molteplici attributi; è possibile e consigliato ridurre la ***Dimensionalità*** del dataset. I principali vantaggi di questa modalità di pre-processing sono molteplici:

- Gli algoritmi lavorano meglio con bassa dimensionalità;
- Aumenta l'interpretabilità del modello;
- La rappresentazione grafica è più facile;
- Lo spazio di memoria ed il tempo di processo sono ridotti.

Aumentando la dimensionalità, l'analisi di dati diventa più complicata, quindi non sempre è efficiente un dataset molto grande. Esistono diverse tecniche di riduzione della dimensionalità, le più importanti sono:

1. ***P***rincipal ***C***omponent ***A***nalysis (PCA): l'obiettivo è quello di trovare nuovi attributi che siano una combinazione lineare degli attributi originali, ortogonali tra di loro e che catturano la massima variabilità dei dati;
2. ***S***ingular ***V***alue ***D***ecomposition (SVD): tecnica alternativa alla PCA, è usata per la riduzione della dimensionalità.

Talvolta è utile trasformare variabili discrete/continue in binarie; il processo è chiamato ***Binarizzazione***: si associa il singolo fattore ad un valore (se ordinale, l'ordine deve essere mantenuto) e si converte il valore ad una variabile dicotoma, come $s = \log_2 k$. Alcune volte è necessario inserire una variabile binaria per ogni valore della variabile categoriale iniziale.

Utilizzata solitamente per analisi di classificazione/associazione, la ***Discretizzazione*** divide in intervalli la variabile iniziale. La discretizzazione può essere:

- *Non supervisionata*: non si prendono in considerazione ulteriori attributi del dataset, ma si specifica autonomamente il numero di intervalli e i punti di divisione. Gli intervalli possono avere stessa frequenza o stessa ampiezza;
- *Supervisionata*: si prendono in considerazione attributi del dataset e si sceglie una funzione obiettivo, come l'Entropia, per valutare il massimo livello di purezza del singolo intervallo (oppure il minimo livello di entropia).

In alcuni casi, le variabili categoriali potrebbero avere troppi livelli. Se la variabile categoriale è ordinale, allora si usa la tecnica simile alle variabili continue per ridurre la dimensionalità. Se nominale, allora si crea una nuova variabile in base al tipo di modalità di raggruppamento; se non è possibile, si usano metodi algoritmici.

Infine, sulle variabili esplicative è possibile applicare delle ***Trasformazioni*** in modo da crearne delle nuove attraverso semplici funzioni matematiche, come la funzione logaritmica, oppure tramite una normalizzazione o standardizzazione della stessa.

2 Classification

I *Modelli di Classificazione* hanno l'obiettivo di prevedere un particolare attributo (per esempio prevedere se un utente rimarrà nella compagnia telefonica) presente in un dataset. Il modello presenta degli ingressi, chiamate variabili di *input* (in questo caso il dataset di partenza) ed una uscita, la previsione della variabile categoriale chiamata *output*. Inoltre, risolve il problema di classificazione per una osservazione ed è utile per:

- Modelli descrittivi, per distinguere oggetti da classi differenti;
- Modelli predittivi, per prevedere la classe di un nuovo record dato un insieme di valori.

La tecnica di classificazione è un approccio sistematico per costruire un modello che classifica una osservazione a partire da un dataset. Si seleziona una parte dei dati per costruire il dataset di **training**, su cui il modello di

classificazione imparerà il metodo di classificazione. Successivamente, si applica il modello di classificazione sul dataset di **test**. L'output del learning model si chiama *inducer* e l'istanza del modello verrà utilizzata per la previsione di classificazione sul dataset di test.

Per misurare le performance del modello si utilizza la matrice di confusione: sulle righe si hanno i valori reali della classe (Yes/No, 1/0, +1/-1, ecc.), mentre sulle colonne il valore previsto dal modello. Esistono 4 possibili combinazioni:

- *True Negative* (TN): il numero di record nei quali l'effettivo valore della classe è negativo ed il modello lo classifica come negativo;
- *False Negative* (FN): il numero di record nei quali l'effettivo valore della classe è negativo, ma il modello lo classifica come positivo;
- *True Positive* (TP): il numero di record nei quali l'effettivo valore della classe è positivo ed il modello lo classifica come positivo;
- *False Positive* (FP): il numero di record nei quali l'effettivo valore della classe è positivo, ma il modello lo classifica come negativo.

La metrica di performance del modello è chiamata **Accuracy**, definita come il rapporto della somma della diagonale principale e la cardinalità del dataset:

$$Accuracy = \frac{TN + TP}{TN + FN + FP + TP}$$

L'errore di performance è definita come il reciproco della accuracy:

$$Error = 1 - Accuracy$$

2.1 Tecniche di Classificazione

Un tecnica di classificazione (*Classifier*) è un approccio sistematico per la costruzione di un modello di classificazione partendo da un dataset. Esse possono essere divise in:

- Euristici (es: Decision Tree, Random Forest, Nearest Neighbor, ecc.);
- Regression Based: si usa un parametro di probabilità (es: Logistic Regression);
- Separation: si partiziona lo spazio delle variabili (es: Support Vector Machine, Artificial Neural Networks, ecc.);
- Probabilistici: si usa la formula di Bayes e si prevedono le posterior (es: Naive Bayes, ecc.).

2.1.1 Modelli Euristici

Un ***Albero Decisionale***, in inglese *Decision Tree*, descrive una struttura ad albero costituito in diversi nodi:

- Nodo radice, senza archi in entrata e 0 o più archi in uscita;
- Nodo interni: con un solo arco in ingresso ed uno o più archi in uscita;
- Nodo terminale (o nodo foglia), conun solo arco in entrata e nessuno in uscita.

Ad ogni nodo è associata una soglia legata ad un attributo di colonna e a seconda del valore si sceglie il nodo di destra o sinistra. L'albero decisionale può essere usato sia per variabili categoriali che numeriche. Per dividere in modo ottimale i nodi si usano diverse misure, come l'*Entropia*, l'*Indice di Impurità di Gini* o l'*Errore di Classificazione*.

Si possono utilizzare anche split multipli, non solo binomiali, o anche in base al tipo di livello dell'attributo. Il cambio di classificazione dove si modifica la linea di decisione è chiamata *Decision Boundary*.

La ***Regressione Logistica*** è un modello statistico nella quale si vuole prevedere la probabilità a posteriori della variabile risposta binomiale partendo da un insieme di variabili esplicative. Indicato con w il vettore dei parametri, si presentano le posterior del modello:

$$P(Y = 0|X = x) = \frac{1}{1 + e^{w \times x}}$$
$$P(Y = 1|X = x) = \frac{e^{w \times x}}{1 + e^{w \times x}}$$

2.1.2 Modelli di Separation

Il ***Support Vector Machine*** è un modello di classificazione robusto basato su algoritmi efficienti. Dato un dataset di m righe dove si misurano due attributi, con $\vec{x}_i \in R^2$ e $y_i \in (-1, +1)$:

$$D = (\vec{x}_1, \vec{y}_1), ..., (\vec{x}_m, \vec{y}_m)$$

si possono separare due (nel caso generare più di due) istanze tramite la seguente equazione:

$$\vec{w}\vec{x} + b = w_1 b_1 + w_2 b_2 + b = 0$$

La retta si può traslare o ruotare, ma per selezionare la miglior decision boundary si utilizza la retta che massimizza il margine tra le due classificazioni. Il modello SVM svolge la seguente funzione:

$$h(x) = \begin{cases} +1 & \text{se retta positiva} \\ -1 & \text{altrimenti} \end{cases}$$

Allenare il margine lineare SVM consiste nel formulare e risolvere la seguente equazione:

$$min_{\vec{w},h} \frac{1}{2} \vec{w} \times \vec{w}^t$$

con il seguente vincolo: $y_i(\vec{w}_i \times \vec{x}_i + b) \geq 1$ e $\forall i = 1, ..., m$

Tutto ciò avviene se l'insieme dei dati è linearmente separabile. Se questo non è possibile, si introduce il concetto di formulazione del *Linear Soft-Margin* della SVM, aggiungendo un termine di penalizzazione alla equazione precedente:

$$min_{\vec{w},b,\xi} \vec{w}\vec{w}^t + \Delta \sum_{i=1}^{m} \xi_i$$

con vincoli:

- $\forall i = 1, ..., m : y_i(\vec{w}\vec{x}_i + b) \geq 1 - \xi_i$;
- $\forall i = 1, ..., m : \xi_i \geq 0$.

Per migliorare la velocità delle SVM, si genera uno spazio, chiamato *Feature Space*, nella quale si mappa lo spazio X in F, trasformando le osservazioni rendendo le istanze linearmente separabili. Così facendo, è possibile ottenere una retta che divide i punti in due classi traslandoli in una quantità pari a Φ_i. La *linear decision boundary* che divide le due istanze nel feature space F ha la seguente equazione:

$$\vec{w}\Phi(\vec{x}) + b = 0$$

Allenare il margine non lineare SVM consiste nel formulare e risolvere la seguente equazione:

$$min_{\vec{w},h} \frac{1}{2} \vec{w}\vec{w}^t$$

con i seguenti vincoli:

- $y_i \times (\vec{w}\Phi(\vec{x}_i) + b) \geq 1$ e $\forall i = 1, ..., m$

La principale differenza tra queste algoritmi di separazione è che la Non linear SVM utilizza una trasformazione di X, ottenendo $\Phi(\vec{X})$. L'algoritmo di apprendimento della Non Linear SVM è il seguente:

$$K(\vec{u}, \vec{v}) = \Phi(\vec{u}) \times \Phi(\vec{v})$$

con K una funzione di similarità ottenuta nello spazio delle variabili X e definita come funzione kernel.

Le ***Neural Network*** sono dei modelli molto avanzati ispirati dai neuroni biologici che costituiscono il cervello animale. Un ***M***ulti-***L***ayer ***P***erceptron (MLP) consiste in un diverso numero neuroni artificiali che comunicano in modo unidirezionale, dalle variabili di input X all'attributo di classe. In generale, si calcola come la combinazione lineare tra le variabili di input meno la soglia (treshold):

$$y_j = f(\sum_{i=1}^{n} w_{ij} \times x_i - \theta_j) = f(z_j - \theta_j)$$

Possibili funzioni di trasferimento sono:

- Trasformazione iperbolica: $f(z) = \frac{e^{(z)} - e^{(-z)}}{e^{(z)} + e^{(-z)}}$
- Trasformazione logistica: $f(z) = \frac{1}{1+e^{(-z)}}$

I MLP sono costituiti da diversi neuroni:

- Neuroni di input, associato alle covariate;
- Neuroni nascosti;
- Neuroni di output, associato all'attributo di classe.

Ogni neurone di input è connesso in modo unidirezionale ai neuroni nascosti, propagando il segnale dal layer di input a quello nascosto. Quando tutti i neuroni del layer nascosto ricevono il segnale dai layer di input, il segnale è mandato a quello di output. Tuttavia, è possibile anche avere più di un layer nascosto: il primo layer manda tutti i segnali al secondo che si attiveranno. Alla fine, quest'ultimo manda dei segnali al neurone di output.

L'obiettivo principale del MLP è quello di identificare il numero di layer nascosti ed i relativi neuroni nascosti. MLP è un modello molto complesso ed esistono diverse misure di ottimizzazione, ma non è possibile determinare il minimo assoluto per ogni layer.

Esistono diverse architetture del MLP, con un numero diverso di layer nascosti e numero diverso di neuroni nascosti. Bisogna ricordare che non è possibile avere degli archi che vanno da uno strato più basso ad uno più alto.

2.1.3 Modelli Probabilistici

I modelli probabilistici usano la formula di Bayes per prevedere le posterior.

Il ***Bayes Classifier*** è un classificatore probabilistico che risolve il problema di classificazione calcolando la probabilità condizionata $P(Y|\vec{X}) = \frac{P(\vec{X}|Y)P(Y)}{P(\vec{X})}$ dove:

- $P(Y)$ è la probabilità a priori dell'attributo della classe;
- $P(\vec{X}|Y)$ è la verosimiglianza delle covariate dato l'attributo della classe;
- $P(\vec{X})$ è la probabilità dell'evidenza (osservazione di una prova);
- $P(Y|\vec{X})$ è la posterior dell'attributo della classe date le covariate.

Dopo aver calcolato le posterior, bisogna decidere a quale label appartiene l'osservazione. Si utilizza la decisione di massima probabilità, nella quale si associa la label al massimo valore della posterior della singola osservazione. Assumendo n covariate binarie e una risposta binaria, bisogna tenere in considerazione che:

- $\theta_{ki} = P(\vec{X} = x_k|Y = y_i)$;
- $k \in (1, ..., 2^n)$;
- $y_i \in \{-1, +1\}$.

Il ***Naive Bayes*** è un modello probabilistico che dati tre attributi X,Y,Z è possibile affermare che X è condizionalmente indipendente da Y dato Z se e solo se la probabilità di X è indipendente dal valore dell'attributo Y quando il valore di Z è noto:

$$P(X = x_i|Y = y_j, Z = z_k) = P(X = x_i|Z = z_k)$$

L'indipendenza condizionata di n covariate dato l'attributo di classe riduce il numero di parametri da svolgere. Il modello Naive Bayes assume questa indipendenza ed è possibile calcolare le posterior dell'attributo di classe date le covariate secondo la seguente formula:

$$P(X_1, X_2, ..., X_n|Y) = \prod_{i=1}^{n} P(X_i|Y))$$

Il numero di parametri k passa da $k(k^n - 1)$ a kn. Il classificatore Naive Bayes calcola le posterior dell'attributo di classe secondo la formula:

$$P(Y = y_k|X_1, ..., X_n) = \frac{P(Y = y_k) \prod_i^n P(X_i|Y = y_k)}{\sum_j^k P(Y = y_k) \prod_i^n P(X_i|Y = y_j)}$$

Il record $(X_1, X_2, ..., X_n)$ è etichettato come il valore della classe che massimizza le posterior:

$$\max_{y_k} P(Y = y_k|X_1, X_2, ..., X_n)$$

Bisogna considerare che il classificatore Naive Bayes è utilizzato anche con attributi numerici: ogni attributo numerico è associato ad una classe di probabilità condizionale. Ogni attributo numerico è associato alla probabilità di classe condizionata, calcolata con la seguente formula:

$$P(X_i = x_i|Y = y_k) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \times e^{-\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}}$$

con parametri:

- $\mu_{ik} = E[X_i|Y = y_k]$;
- $\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2|Y = y_k]$.

Il classificatore Naive Bayes è generalizzato dal ***Bayesian Network*** che rende meno drastico l'assunzione d'indipendenza incondizionata, servendosi della *sparsity*. L'attributo di classe è ancora associato al nodo di root Y, mentre gli attributi esplicativi sono associati non solo alla Y, ma anche agli stessi attributi di classe. Bisogna considerare, però, che non sono ammessi cicli.

Questa generalizzazione mantiene le proprie dipendenze tra le covariate che non assumono più l'indipendenza dato l'attributo di classe Y. A livello di classificazione, il vantaggio è che funziona perfettamente anche con NA di input.

Un particolare modello è il ***Tree-Augmented Naive Bayes***, che permette di avere al più un altro nodo genitore rimuovendo il nodo di classe. Inoltre, effettua model selection calcolando gli attributi più interessanti.

3 Performance Evaluation

L'*Accuracy* del modello non sempre è una modalità sufficiente per confrontare l'efficienza del modello. Gli errori compiuti su un modello di classificazione possono essere:

- *Errore di Training*: il numero di record del training classificate in modo errato;
- *Errore di Generalizzazione*: errore previsto nel dataset di test.

Un buon modello di classificazione non solo deve classificare bene il dataset di training, ma anche quello di test; in particolare, devono avere sia un basso training error che un basso errore di generalizzazione.

Un modello che classifica troppo bene il dataset di training può avere un basso errore di generalizzazione rispetto ad un alto errore di Training. Questo fenomeno è

chiamato ***overfitting*** ed avviene quando un modello ha una performance troppo elevata nel dataset di training, ma generalizza male quello di Test. Il fenomeno opposto è detto ***underfitting***, nella quale un modello commette un alto errore di classificazione sia sui dati di training che sui dati di test. Molto probabilmente il modello è poco complesso rispetto ai dati e non classifica bene le osservazioni.

3.1 Misure di performance

L'analisi di classificazione consiste nell'utilizzare diversi modelli di classificazione associati a parametri valutati e successivamente nel confrontarli per osservare quale tra quelli utilizzati è il più efficiente. I modelli di classificazione sono confrontati in diversi termini.

L'***Accuratezza*** misura la capacità del modello di classificazione di predire in modo affidabile nuovi record. Inoltre, permette di selezionare e garantire le performance migliori sui nuovi dati. Successivamente verranno considerate le seguenti notazioni:

- D_T, dati di training (con t righe);
- D_{Ts}, dati di test (con v righe).

Un buon indicatore di accuracy è misurato dalla percentuale di righe classificate in modo corretto da D_{Ts}. Sia y_i il valore associato all'istanza $x \in D_{Ts}$, definito come il valore della classe predetto dal modello. Si introduce la funzione di perdita, chiamata in inglese *Loss function*:

$$L(y_i, f(\vec{x}_i)) = \begin{cases} 0 & y_i = f(\vec{x}_i) \\ 1 & y_i \neq f(\vec{x}_i) \end{cases}$$

L'accuracy è calcolata nel nel seguente modo:

$$acc(D_{Ts}) = 1 - \frac{1}{v}\sum_{n=1}^{v} L(y_i, f(\vec{x}_i))$$

In alcuni casi è preferibile utilizzare l'errore dell'accuracy:

$$Err(D_{Ts}) = 1 - acc(D_{Ts}) = \frac{1}{v}\sum_{n=1}^{v} L(y_i, f(\vec{x}_i)$$

Gli algoritmi di classificazione differiscono dal tempo di apprendimento e dallo spazio di memoria. Quando un classificatore richiede un alto tempo di apprendimento e/o uno spazio di memoria elevato è consigliato compiere un campionamento del dataset originale.

Un modello è considerato ***robusto*** nel caso in cui non sono presenti outliers, valori mancanti ed alcuna variazione tra i dati di test e di training.

Un modello è considerato ***scalabile*** quando il modello è capace di apprendere enormi quantità di dati. La scalabilità è strettamente legata alla velocità di apprendimento).

Un modello è definito ***interpretabile*** quando il problema di classificazione è risolto e non si limita ad avere un buon livello di accuracy. Il modello di classificazione deve essere semplice e chiaro nella comprensione di un esperto di dominio, che molte volte non conosce la statistica.

Esistono diverse procedure per la valutazione delle performance di un modello di classificazione:

La procedura ***Holdout*** divide il dataset originale con una semplice procedura di CCS, suggerendo la divisione in 2/3 per il dataset di training, per far apprendere il modello di classificazione ed ottenere un inducer, e 1/3 per il dataset di test, su cui verrà misurata l'accuracy del modello. Essa consiste nel limitare l'uso dei dati per far apprendere il classificatore. L'accuratezza stimata dipende dalla scelta del dataset di test, per non incorrere in fenomeni di over/under-fitting.

Una stima più robusta alla comune Holdout è chiamata ***Iterated Holdout***, che consiste nell'iterare R volte il metodo descritto in precedenza. Per ogni interazione si estrae un campione D_{Tr} composto da t righe, ottenendo $D_{Ts_r} = D - D_{Tr}$. L'accuratezza del classificatore è stimata dalla media dei valori di accuracy di ogni campionamento:

$$acc = \frac{1}{R} \times \sum_{n=1}^{R} acc(D_{Ts_r})$$

In questo modo, è possibile ridurre la varianza associata alla stima, ma non permette di controllare il numero di volte in cui il dato record è contenuto nel dataset di training e di test.

La ***Cross Validation*** è un altro metodo robusto di valutazione delle performance del modello. Il dataset D viene diviso in k sotto-categorie, contenendo diversi piccoli dataset $D_1, D_2, ..., D_k$. Successivamente, si compiono k volte apprendimenti sul dataset di test; alla k-esima iterazione, si ottiene il seguente dataset di training:

$$D_{T_k} = \{D_1, ..., D_{k-1}, D_{k+1}, ..., D_k\}$$

Il suo complemento D_k è usato come dataset di test. L'algoritmo di classificazione procede in k fasi di training. Per ottenere l'accuracy, si compie la media di ogni k:

$$acc = \frac{1}{K} \times \sum_{n=1}^{K} acc(D_k)$$

Si possono scegliere diversi valori del parametro K (di solito sono 3,5,10). Comunque, è necessario tenere in considerazione che ogni partizione del dataset deve contenere la stessa proporzione dei possibili valori dell'attributo di classe. Se differiscono molto tra di loro, è consigliato utilizzare un campionamento stratificato.

3.2 Comparing Classifiers

Dopo aver costruito diversi modelli di classificazione, bisogna stimare quale sia il migliore tra tutti; il concetto non è facile da definire, ma un confronto iniziale è comparare le accuracy dei modelli. Molte volte è utile comparare le performance dei modelli per determinare quale sia il migliore, bisogna ricordare che il calcolo della accuracy considerando differenza dei record non è statisticamente rilevante.

Si presentano perciò due problemi:

1. Intervalli di Confidenza dell'Accuracy del modello;
2. Differenza di Accuracy dei modelli.

3.2.1 Intervallo di Confidenza

Si considera il Test set D_N con N il numero di record. Sia $X \sim Bi(Np, Np(1-p))$ il numero di previsioni corrette e p la vera accuratezza del modello. L'accuratezza empirica è calcolata come $acc = \frac{X}{N}$, con $X \sim Bi(p, \frac{p(1-p)}{N})$. La distribuzione Binomiale è usata per stimare l'IC per l'accuracy approssimandola ad una N, calcolata nel seguente modo:

$$P(-z_{1-\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < z_{1-\alpha/2}) = 1 - \alpha$$

Questo procedimento è possibile solamente ripetendo l'esperimento n volte e tra loro indipendenti.

L'IC della accuracy sarà:

$$\frac{acc + \frac{z^2_{1-\alpha/2}}{2N} \pm z_{1-\alpha/2} \times \sqrt{\frac{acc}{N} - \frac{acc}{N^2} + \frac{z^2_{1-\alpha/2}}{4N^2}}}{1 + z^2_{1-\alpha/2/N}}$$

All'aumentare della dimensione del campione, l'ampiezza dell'IC diminuirà e quindi la stima sarà più precisa.

3.2.2 Test Set Differenti

Si considerano due modelli di classificazione indipendenti:

- M_1 con un test set D_1 contenente n_1 osservazioni ed un errore e_1;
- M_2 con un test set D_2 contenente n_2 osservazioni ed un errore e_2.

La differenza in termini di errore, $d = e_1 - e_2$, è statisticamente significativo? Si considerano n_1 e n_2 sufficientemente grandi e quindi che e_1, e_2 siano statisticamente distribuiti come una N; perciò, anche $d \sim N(d_t, \sigma_d^2 = \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2})$. L'IC sarà:

$$IC_{d_t} = (d - z_{1-\alpha/2} \times \sigma_d, d + z_{1-\alpha/2} \times \sigma_d)$$

Si prendono in considerazione tre possibilità:

- Se IC contiene 0, allora i due classificatori non sono significativamente differenti in termini di errore con livello di confidenza α;
- Se l'estremo superiore è negativo, si preferisce il modello M_1 rispetto a M_2 a livello di classificazione;
- Se l'estremo inferiore è positivo, si preferisce il modello M_2 rispetto a M_1 a livello di classificazione.

3.2.3 Test Set Uguali

Si considera la situazione in cui si comparano i due modelli usando la procedura di cross-validazione K-fold, perciò il dataset D è diviso in k sottogruppi, creando $D_1, D_2, ..., D_k$ dataset. Applicando la tecnica di classificazione per costruire M_1 e M_2 dalle $k-1$ partizioni e testarli sulla rimanente, si applica questo step k volte, usando ogni volta partizioni differenti del test set. Si considerano i seguenti modelli:

- M_{1k}, il modello inducer del modello M_1 ottenuto alla k-esima iterazione;
- M_{2k}, il modello inducer del modello M_2 ottenuto alla k-esima iterazione.

Su ogni partizione dei dati, si conduce il test sullo stesso test set. Si considerano i seguenti errori:

- e_{1k} l'errore di M_{1k};
- e_{2k} l'errore di M_{2k}.

La differenza $d_k = e_{1k} - e_{2k}$ alla k-esima iterazione è distribuita come $N(d_t^{cv}, \sigma_{cv}^2)$. La varianza totale è stimata usando la seguente formula:

$$\hat{\sigma}^2_{d_{cv}} = \frac{\sum_{k=1}^{K}(d_k - \overline{d})^2}{K(K-1)}$$

con $\overline{d} = \frac{1}{K}\sum_{k=1}^{K} d_k$. Si utilizza la distribuzione t-Student per calcolare l'IC del valore medio di d_t^{cv}:

$$IC = (\overline{d} - t^{k-1}_{1-\alpha/2} \times \hat{\sigma}_{d_{cv}}, \overline{d} + t^{k-1}_{1-\alpha/2} \times \hat{\sigma}_{d_{cv}})$$

Si compiono le considerazioni precedentemente svolte.

3.3 Regressione lineare

La variabile dipendente (o di risposta) in un modello di regressione lineare è una variabile continua. Il modello spiega una variabile dall'insieme degli attributi del dataset. Si cerca una funzione:

$$f : R^k \to R$$
$$\widehat{Y} = f(\underline{X}) + \epsilon$$

a cui si aggiunge un errore (o residuo) ϵ che rappresenta lo scarto del modello, dovuto a incertezza o ignoranza della formula reale.
Un modello di regressione lineare è una funzione lineare di una matrice (del disegno) di valori $\underline{X}$ pesata con un vettore $\underline{w}$. Se la matrice del disegno è composta da una sola variabile indipendente, il modello è detto *modello lineare semplice*: non ha funzioni pratiche ma è interessante dal punto di vista teorico perché presenta le stesse problematiche dei modelli multivariati, polinomiali e con componenti rettangolari.
L'obiettivo è minimizzare l'errore standard commesso nel prevedere y tramite la funzione f:

$$\begin{aligned} SSE &= \sum_{i=1}^{m} e_i^2 \\ &= \sum_{i=1}^{m} [y_i - f(\underline{X})]^2 \\ &= \sum_{i=1}^{m} [y_i - x_i w_i - b]^2 \end{aligned}$$

Aggiungere variabili indipendenti non modifica il ragionamento né la formulazione matematica del problema. Si possono aggiungere altre variabili dall'elevamento a potenza di quelle presenti o dalla loro moltiplicazione: i parametri sono detti *gradi di libertà* del modello. Aggiungere troppi gradi di libertà però può provocare overfitting dei dati, oltre a ridurre l'interpretabilità del modello.

3.3.1 Critica

Il modello lineare, pur essendo molto semplice, si basa su delle assunzioni molto forti.

3.3.1.1 Assunzioni relative ai residui. I residui devono avere media nulla e devono essere indipendenti (e quindi anche incorrelati) per tutti i valori di $\underline{X}$. Inoltre la varianza dei residui deve essere costante (devono cioè essere *omoschedastici*).

$$\underline{\epsilon} = N(\underline{\mu}, \sigma I)$$

Per verificare queste ipotesi, non esiste un test considerato valido, tuttavia si usano strumenti grafici e test statistici, parametrici o non parametrici.
Per verificare la distribuzione del vettore dei residui, si usano i test Kolmogorov-Smirnoff o Shapiro-Wilk, mentre Durbin-Watson è considerato valido per calcolare l'interdipendenza dei residui. La distanza di Cook, inoltre, stabilisce se un'osservazione è particolarmente influente per la stima del modello.
È possibile ridurre l'eteroschedasticità del modello operando sul logaritmo della variabile.

3.3.1.2 Significatività dei coefficienti. Un coefficiente può anche non essere particolarmente significativo nel modello (ha cioè peso pari a 0). Bisogna dunque calcolare quali coefficienti sono significativamente non nulli; tuttavia, avendo a disposizione solamente la stima, sono necessari test statistici: basta calcolare un intervallo di confidenza ed escludere, con α pari a quello dell'intervallo, i coefficienti per cui il valore 0 è interno all'intervallo. Altro approccio è effettuare un test statistico (t-test) sulla significatività del singolo coefficiente ed escludere con p-value $> \alpha$.

3.3.2 Valutazione

Il coefficiente R^2 di determinazione rappresenta la percentuale di varianza totale spiegata dal modello:

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

Si può comunque verificare l'overfitting: si usa il coefficiente aggiustato.

$$\tilde{R}^2 = 1 - (1 - R^2)\frac{m-1}{m-k-1}$$

che generalmente assume valori minori di R^2. Se la differenza tra i due valori non è particolarmente alta, (probabilmente) non si verifica l'overfitting.

3.3.2.1 Intervalli di confidenza e di previsione. In un modello lineare semplice, si distinguono l'intervallo di confidenza, che contiene il valore medio $E[\underline{y}|\underline{X}]$, e l'intervallo di previsione, che contiene la singola realizzazione y.

3.3.2.2 Selezione variabili. Sono possibili una selezione *forward* e una selezione *backward*. La prima parte dal modello vuoto e inserisce variabili a ogni passo del ciclo; si usa con un grande numero di variabili esplicative per semplificare il modello. L'approccio backward invece parte dal modello pieno ed esclude variabili a ogni ciclo; si usa con poche variabili per eliminare quelle poco significative.

4 Class Imbalance Problem, Feature Selection and Non Binary Class

4.1 Class Imbalance Problem

Si considera il caso in cui una variabile categoriale è particolarmente sbilanciata, con valori di un livello nettamente più alti rispetto ad un altro. In una classificazione binaria, la classe rara è quella con poche osservazioni e classificata come la classe positiva (+1), mentre la rimanente classe maggioritaria viene classificata come negativa (-1). Prendendo in considerazione la matrice di confusione, si considerano i seguenti rapporti:

- Il *TNR* (True Negative Rate), detta anche *Specificity*, è il rapporto tra le osservazioni negative predette correttamente dal modello di classificazione:
$$TNR = \frac{TN}{TN + FP}$$
- Il *TPR* (True Positive Rate), detta anche *Sensitivity*, è il rapporto tra le osservazioni positive predette correttamente dal modello di classificazione:
$$TPR = \frac{TP}{TP + FN}$$
- Il *FPR* (False Positive Rate) è il rapporto tra le osservazioni negative predette come la classe positiva del modello di classificazione:
$$FPR = \frac{FP}{TN + FP}$$
- Il *FNR* (False Negative Rate) è il rapporto tra le osservazioni positive predette come la classe negativa del modello di classificazione:
$$FNR = \frac{FN}{TP + FN}$$

La *precision* e *recall* sono due metriche di misura per determinare quale delle due classi è considerata più importante rispetto all'altra. La **Precision** è definita come il rapporto tra le osservazioni positive correttamente predette e il gruppo che il modello ha classificato come positivo:

$$p = \frac{TP}{TP + FP}$$

Più è alta la precision, meno sarà il numero di FP e quindi errori commessi dal modello. La **Recall**, invece, misura il rapporto tra le osservazioni positive correttamente predette:

$$r = \frac{TP}{TP + FN}$$

Un'alta recall implica poche osservazioni positive classificate come classe negativa. Essa è equivalente al calcolo del TPR. Tipicamente, precision e recall sono combinati tra loro formando la metrica chiamata F_1, calcolata come media armonica tra le due metriche:

$$F_1 = \frac{2rp}{r + p}$$

Un alto valore di F_1 implica elevati valori sia della precision che della recall. Questa metrica è generalizzabile calcolando una nuova metrica, F_β per esaminare la rilevanza tra le misure:

$$F_\beta = \frac{(\beta^2 + 1)rp}{r + \beta^2 p} \in [0, \infty)$$

Osservazioni:

- F_β con $\beta = 0$ è la precision;
- F_β con $\beta = \infty$ è la recall.

4.2 Counting the Cost

Dopo aver diviso il dataset in Training e Test e dopo aver evidenziato il miglior modello, si confronta la matrice di confusione rispetto al matrice di confusione del modello standard. Bisogna anche controllare il **costo** del processo del modello, ricercato sempre dalle aziende. Il costo è calcolato nel seguente modo:

$$Cost = C_{--}TN + C_{-+}FP + C_{+-}FN + C_{++}TP$$

Se i costi sono simmetrici, è possibile concludere che i costi sono proporzionali alla accuracy. In questo caso, la numerosità sarà $N = TP + TN + FP + FN$, l'accuracy $acc = \frac{TP+TN}{N}$ e il costo sarà:

$$\begin{aligned} Cost &= p(TP + TN) + q(FN + FP) = && (1)\\ &= p(TP + TN) + q(N - TP - TN) = && (2)\\ &= qN - (q - p)(TP + TN) = && (3)\\ &= N[q - (q - p)(TP + TN)] && (4) \end{aligned}$$

4.2.1 Cumulative Gains

Dato un modello di classificazione che abbia come output la probabilità prevista per la classe positiva, bisogna trovare un sottoinsieme nella quale le osservazioni hanno alta proporzione delle osservazioni positive ed avere un valore più alto del dataset iniziale. Si applica il modello ad un numero definito di osservazioni e si calcolano le probabilità, ordinandole in modo decrescente. Si sceglie un sottoinsieme con la massima proporzione possibile. Si costruiscono le ***Lift***, osservando quanto sbaglia il modello. Le Lift sono calcolate come rapporto tra la percentuale di record positivi (+1) su quelli negativi (-1) e il numero di osservazioni del campione in percentuale sul dataset originale. Le lift possono essere anche rappresentate in termini grafici e permette di scegliere il miglior campione che massimizza la percentuale di record positivi.

4.2.2 Lift Chart

Le Cumulative Gains possono essere rappresentate anche con le Lift Chart, che presenta sulle ascisse la percentuale del campione sul dataset originale e sulle ordinate il valore delle Lift. Perciò, data una percentuale del campione, è possibile osservare il valore dei record classificati positivamente.

4.2.3 ROC Curve

Le Lift Chart sono sempre messe in relazione con una tecnica grafica per valutare i modelli di classificazione, chiamata **R**eceiver **O**perating **C**haracteristic (ROC). Essa rappresenta la percentuale di Falsi Positivi sull'asse delle ascisse e la percentuale di Veri Positivi sull'asse delle ordinate. Questo grafico illustra le performance dei modelli di classificazione senza considerare la distribuzione della classe o errori di costo. Molte volte capita che la rappresentazione grafica della ROC sia seghettata: questo dipende dalla numerosità dei dati. Non sempre è possibile osservare il miglior modello in termini di classificazione, poiché spesso un modello classifica meglio una parte dei dati rispetto ad un'altra.

4.3 Feature Selection

La Feature Selection è una operazione che permette di scegliere il numero di variabili esplicative del modello per osservare quali di queste sono *ridondanti*, contenenti informazioni già presenti in altre covariate, oppure *irrilevanti*, non contenendo alcuna informazione significativa per la variabile risposta. Sono possibili diversi approcci:

- *Forza bruta*: si applicano tutti i possibili campionamenti di variabili disponibili come variabili di input nel modello di classificazione. Per calcolare il numero possibile di campioni, si calcola una combinazione:

$$\sum_{n=1}^{k} \binom{k}{n}$$

All'aumentare del numero di variabili esplicative aumentano esponenzialmente i possibili campioni;

- *Embedded*: si selezionano le covariate in base alla modalità di apprendimento di alcuni classificatori, come Bayesian Network, Alberi Decisionali, ecc;
- *Filter*: la selezione avviene prima di ogni apprendimento di un modello. La scelta delle coviariate avviene dato un campione di attributi alla funzione obiettivo, che darà un riscontro, in inglese *feedback*. Il processo termina all'ottenimento del campione di covariate ottimali che verrà utilizzato successivamente per il modello di classificazione;
- *Wrapper*: il classificatore è utilizzato per cercare il miglior campione di covariate possibili. La scelta delle covariate avviene tramite un campione di esse al classificatore che darà un feedback. Il processo termina all'ottenimento del campione ottimale che verrà utilizzato per il modello di classificazione.

La differenza principale tra la selezione Filter e Wrapper è che il primo seleziona le covariate in base ad una funzione obiettivo, mentre la seconda direttamente dal modello di classificazione. La modalità di selezione con l'approccio Filter può essere di tipo:

- *Univariato*: si sceglie una misura di associazione tra l'attributo possibile ordinata in base alla misura di associazione e la risposta e si selezionano i migliori attributi per il modello di classificazione. Vengono eliminati le covariate non significative, ma non quelle ridondanti (t-test, ANOVA, permutazione);
- *Multivariato*: si identificano solamente le covariate strettamente correlate alla risposta, eliminando le variabili ridondanti e non significative, con l'operazione chiamata *Correlation Feature Selection*.

I principali vantaggi di utilizzare la feature selection è la riduzione dei costi nella collezione dei dati ottenendo modelli di classificazione con alta Accuracy e alta interpretabilità, ma soprattutto riducendo l'overfitting del modello.

4.4 Train, Validation, Test and Feature Selection

Alcuni modelli statistici permettono di avere parametri, come la regolarizzazione nelle Neural Network. Si presenta quindi un problema di ottimizzazione globale per cui non è possibile risolvere la seguente equazione:

$$E(w, \lambda) = \frac{1}{m} \sum_{i=1}^{m} (y_i - y_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{K} w_i^2$$

con λ parametro di iper-regolarizzazione (o di penalizzazione) e K il numero di parametri liberi delle NN.

Non utilizzare il dataset di Training per regolarizzare il parametro λ, perché si potrebbe ottenere overfitting; invece, si divide ulteriormente il Train in una parte in Validation (1/3 del Training). In questo modo, si regolarizza il parametro per ottenere nel Test una stima imparziale della misura di performance.

Utilizzando il metodo Filter è consigliato dividere il dataset di partenza in Train e Test, in quanto la rilevanza e la ridondanza delle variabili sono stimate tramite il Train, per poi applicare l'apprendimento del classificatore. Infine, si stimano le performance applicando il classificatore al Test.

Utilizzando il metodo Wrapper è consigliato dividere il dataset in Train + Validation e Test. Il Validation è utilizzato per ottimizzare le performance quando diverse covariate vengono utilizzate dal modello di classificazione. Il Train + Validation è utilizzato per l'apprendimento del classificatore. Infine, si stimano le performance applicando il classificatore al Test.

4.5 Non Binary Classification

Una classificazione non binaria è una classificazione nella quale una variabile categoriale ha più di due livelli. Esistono due tipi di classificazioni non binarie:

- Classificazione *Multi-Class* (classi ordinate);
- Classificazione *Multi-Label* (Ranking problem nella quale si considera l'ordinamento dei livelli).

Il problema di classificazione non binaria è tipicamente risolta utilizzando la trasformazione *Uno vs Tutti* nella quale per ogni livello si identifica una opzione in modo da ottenere un set di classificatori binari. Si sviluppa una tipologia di classificazione per ogni valore della classe (*è della classe o no?*). Per ognuna delle classi si calcola il relativo modello di classificazione. Il valore con la probabilità più alta la si classifica alla corrispondente classe. Nel caso di multi-label, si sceglie una soglia per la quale si accetta la categorizzazione della classe; è possibile che si verifichino i seguenti problemi:

- Nessuna soglia superata, quindi soggetto non etichettato;
- Più di un livello supera la soglia, quindi più etichette.

5 Clustering

La Cluster Analysis è una particolare analisi che ha lo scopo di raggruppare soggetti basati sulle descrizioni e sulle loro relazioni. In particolare, si basa su due concetti fondamentali:

- Le istanze all'interno di un cluster possono essere **simili**, quindi relazionati ad un altro (o viceversa);
- Se due istanze sono simili, allora sono **omogenee**, altrimenti **eterogenee**.

La cluster analysis è applicata a due finalità:

- *Understanding*, nella quale si classificano gruppi di oggetti che hanno delle caratteristiche comuni, con ambiti di applicazione in biologia, business, ecc.;
- *Utility*, nella quale si riassumono le principali caratteristiche di un cluster. L'obiettivo è quello di identificare il cluster prototipo più rappresentativo.

Bisogna tenere in considerazione che la definizione di cluster è imprecisa e la miglior definizione dipende dalla natura del dato e dal risultato finale che si vuole ottenere.

5.1 Tipologie di Cluster

Esistono molte tecniche di clustering:

- Partizionale *vs* Gerarchico;
- Esclusivo *vs* Overlapping *vs* Fuzzy;
- Completo *vs* Parziale.

Il metodo di clustering ***Partizionale*** è una divisione dei dati in modo da ottenere dei cluster senza che ci sia una altro cluster superiore. Il metodo ***Gerarchico***, invece, permette di avere dei sub-cluster in un cluster organizzati come un albero, *ma* diverso da un dendrogramma.

Il metodo di clustering ***Esclusivo*** è una particolare tecnica nella quale tutti i cluster sono esclusivi, ovvero ogni osservazione è assegnata ad un singolo cluster. Il metodo ***Overlapping***, invece, permette il contrario: le osservazioni possono essere divisi in più cluster. Infine, il metodo ***Fuzzy*** permette ad un singolo dato di appartenere ad ogni cluster con un peso di appartenenza specifico che varia da 0 ad 1.

Il metodo di clustering ***Completo*** permette di assegnare ad ogni osservazione presente nel dataset ad un cluster. Il metodo ***Parziale***, invece, non necessita di assegnare ogni osservazione ad un cluster: alcune osservazioni possono non appartenere ad un gruppo ben definito.

Bisogna tenere in considerazione che esistono diverse concezioni di prove di clustering:

- *Cluster ben separati*, nella quale ogni oggetto nei cluster è omogeneo al loro interno ed eterogeneo all'esterno. Questa è la definizione ideale di cluster; in realtà, sarà molto difficile dividere eccellentemente due cluster;
- *Cluster Prototype-Based*, nella quale ogni oggetto è simile ai prototipi che definiscono i cluster. Molte volte questo prototipo è il centroide. Se non è rappresentativo, si utilizza;
- *Cluster Density-Based*, nella quale un cluster è una regione densa di oggetti circondata da un altro cluster di bassa densità;
- *Cluster Graph-Based*, nella quale i dati sono rappresentati tramite un grafo. Un cluster è definito come una componente connessa costituita da nodi ed archi, perciò un cluster è definito come un insieme di componenti connesse.

Da un dataset, si compie feature selection, nella quale si selezionano le variabili più significative, o feature extraction. Successivamente si seleziona l'algoritmo che misura la prossimità tra i cluster e si costruisce un criterio di selezione. Dopo l'apprendimento, si validano le divisioni dei cluster, prendendo in considerazioni gli obiettivi dello studio. Infine, si traggono le conclusioni per migliorare la conoscenza dei dati per studi statistici successivi.

5.2 Prossimità

La Cluster Analysis prende in considerazione i concetti di similarità e dissimilarità tra le osservazioni:

- La ***similarità*** (s) tra due record è una misura numerica del grado per cui sono simili. Valori elevati tra similarità implica la somiglianza tra i due oggetti: $s \in [0, 1]$;
- La ***dissimilarità*** (d) tra due oggetti è una misura numerica del grado per cui sono differenti. Di solito $d \in [0, 1]$, ma possono anche assumere valori $\in [0, \infty)$.

Spesso si applicano delle trasformazioni lineari per converitre una similarità ad una dissimilarità (o viceversa) comprese tra i valori 0 ed 1:

$$s' = \frac{s - min_s}{max_s - min_s} d' \quad = \frac{d - min_d}{max_d - min_d}$$

Se la prossimità ha valori compresi tra 0 ed infinito, si compie una trasformazione non lineare, ma i valori della trasformazione non hanno la stessa relazione con l'altra in una nuova scala, perché distorti:

$$d' = \frac{d}{1 + d}$$

Alti valori della dissimilarità originale sono compressi intorno ad 1 usando la nuova trasformazione. Si può passare dalla similarità alla dissimilarità (o viceversa) con il complementare: $d = 1 - s$ e $s = 1 - d$. Un altro approccio utile è definire la similarità come il negativo della dissimilarità (o viceversa):$s = -d$

La prossimità tra due record è una funzione della prossimità tra gli attributi corrispondenti dei due record. Si descrive la prossimità tra due record avendo un singolo attributo:

- *Nominali*: $s = 1$ se $x = y$, dove x ed y sono due record, altrimenti 0 (sia binari che categoriali);

$$d = \begin{cases} 0 & \text{se } x = y \\ 1 & \text{altrimenti} \end{cases}$$

- *Ordinali*: $d = \frac{|x-y|}{n-1}$, dove n è il numero di variabili ordinali, quindi $s = 1 - d$. In questo caso si stanno assumendo intervalli equi tra i livelli, considerando una scala lineare;
- *Numerici*: si considera la dissimilarità e similarità introdotte ad inizio capitolo: $d = |x - y|$.

5.2.1 Misure di Prossimità

Esistono diversi modi per calcolare la prossimità tra due record.

Le distanze sono considerate come dissimilarità con certe caratteristiche. Le distanze più importanti appartengono alla famiglia chiamata ***Distanza di Minkowski***:

$$d(x,y) = \sqrt[r]{\sum_{k=1}^{n} |x_k - y_k|^r}$$

- se $r = 1$, allora si ottiene la distanza di Manhattan;
- se $r = 2$, allora si ottiene la distanza Euclidea;
- se $r = \infty$, allora si ottiene la distanza Suprema.

Le distanze di Minkowski presentano le seguenti proprietà:

- Non Negatività: $d(x,y) \geq 0$, $d(x,y) = 0$ se $x = y$;
- Simmetria: $d(x,y) = d(y,x)$;
- Diseguaglianza Triangolare: $d(x,z) \leq d(x,y) + d(y,z)$.

La similarità soddisfa le prime due proprietà, ma non la terza.

Si presentano altre misure di prossimità. Esistono diversi motivi per cui un coefficiente è migliore di un altro in diversi contesti: il ***S**imple **M**atching **C**oefficient* (SMC) viene utilizzato quando tutti gli attributi binari sono simmetrici, ovvero tutti i valori 0 ed 1 sono equamente distribuiti.

$$SMC(x,y) = \frac{\#attributi\ veri}{\#attributi} \in [0,1]$$

Il ***Coefficiente di Jaccard*** si usa quando gli attributi non sono simmetrici (solo attributi binari).

$$J(x,y) = \frac{\#presenzacongiunte}{\#attributitranne00} = \frac{f_{11}}{f_{11} + f_{10} + f_{01}}$$

Esiste anche una versione estesa, chiamata Extended Jaccard Coefficient, utilizzata spesso per l'analisi di documenti:

$$EJ(x,y) = \frac{x \times y}{||x^2|| + ||y^2|| - x \times y}$$

La ***Similarità del Coseno*** è una misura molto simile al Coefficiente di Jaccard, ma è possibile anche introdurre attributi non binari. Questa misura è estremamente utile per comparare poche osservazioni, utilizzata nella Information Retrieval dove i documenti sono rappresentati come vettori;

$$cos(x,y) = \frac{x \times y}{||x|| \times ||y||} \in [0,1]$$

La ***Correlazione*** (non ho nulla da aggiungere, so tutto):

$$corr(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y} \in [-1,+1]$$

6 Clustering Evaluation

Esistono molti algoritmi con diversi parametri per compiere analisi di clustering, ma la valutazione dei cluster non è una operazione ben sviluppata perchè poco utilizzata nella Cluster Analysis, anche se molto importante.

Un algoritmo di clustering ha l'obiettivo di trovare dei cluster all'interno dei dati anche se non esistono delle divisioni naturali. Per valutare i risultati dell'algoritmo bisogna determinare la tendenza di costruzione di cluster del dataset ed il corretto numero di cluster (anche se non esiste un vero e proprio numero). Per risolvere questi problemi si utilizzano delle misure di valutazione, anche chiamati *Indici*, che possono essere di tre tipi.

6.1 Misure di Valutazione

6.1.1 External Measure

L'indice Esterno, chiamato in inglese ***External Measure***, misura se l'estensione della struttura dei cluster corrisponde a qualche struttura esterna. Si prendono in considerazione una partizione $P = P_1, ..., P_R$ di un datasaet di m righe divise in R categorie, e una partizione $C = C_1, ..., C_K$ ottenuto da un algoritmo di clustering che partiziona i dati in K cluster. L'indice supervisionato compara le partizioni P e C considerando 4 casi:

a. x e y appartengono alla stesso cluster di C e alla stessa categoria di P;
b. x e y appartengono alla stesso cluster di C, ma ad un diversa categoria di P;

c. x e y appartengono ad un cluster diverso di C, ma alla stessa categoria di P;
d. x e y appartengono ad un cluster diverso di C e ad una categoria diversa di P;

Il numero totale di coppie è pari a $M = \frac{m(m-1)}{2} = a + b + c + d$. Si prendono in considerazione diversi indici:

- Rand: $R = \frac{a+d}{M} \in [0, 1]$
- Jaccard: $J = \frac{a}{a+b+c} \in [0, 1]$
- Fowleks and Mallows: $FM = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}} \in [0, 1]$
- Γ Statistics: $\Gamma = \frac{Ma-(a+b)(a+c)}{\sqrt{(a+b)(a+c)(M-a-b)(M-a-c)}} \in [-1, +1]$

Più i valori di questi indici si avvicinano ad 1, più le partizioni C e P saranno simili.

6.1.2 Internal Measure

L'indice Interno, chiamato in inglese ***Internal Measure***, misura la bontà della struttura dei cluster rispetto alle informazioni esterne. Questi indici tengono in considerazione anche di misure di *coesione*, che determina quanto sono relazionate i record all'interno dei cluster, e *separazione*, che determina quanto sono distinti, perciò ben separati, i cluster. In generale, si considera l'insieme dei K cluster $C = C_1, ..., C_K$. La validità generale dei cluster è calcolata come una somma pesata dei singoli cluster:

$$overall_validity = \sum_{i=1}^{K} w_i \times validity(C_i)$$

con la funzione di validità considerata come coesione (c), separazione (s) od una combinazione delle due. Il valore dei pesi dipende dalla misura di valutazione dei cluster. Considerando come funzione di validità la coesione, valori alti implicano un'alta coesione, mentre per la separazione valori più bassi.

Per i cluster *Graph-Based*, la coesione di un di un cluster è definito come somma dei pesi che collegano i punti all'interno di un cluster:

$$c(C_i) = \sum_{x,y \in C_i} p(x, y) = \sum_{x,y \in C_i} s(x, y)$$

La coesione e la similarità sono massimizzate quando la dissimilarità è minimizzata.

La separazione tra due cluster può essere misurata come la somma dei pesi che collegano i punti di un cluster ad un altro:

$$s(C_i, C_j) = \sum_{x \in C_i, y \in C_j} p(x, y) = \sum_{x \in C_i, y \in C_j} s(x, y)$$

La separazione e la similarità sono minimizzate quando la dissimilarità è massimizzata.

Per i cluster *Prootype-Based*, la coesione di un clsuetr è definito come la somma delle prossimità rispetto al centroide (o medoide) del cluster:

$$c(C_i) = \sum_{x \in C_i} p(x, c_i) = \sum_{x \in C_i} s(x, c_i)$$

La separazione tra due cluster è definita come la prossimità tra i centroidi (o medoidi) dei due cluster:

$$s(C_i, C_j) = p(c_i, c_j) = (c_i, c_j)s$$

Un'altra modalità di calcolo della misura è possibile utilizzando il centroide complessivo (in questo caso c):

$$s(C_i) = p(c_i, c) = s(c_i, c)$$

Si possono utilizzare diversi pesi per il calcolo della validità di un cluster, in base anche alla grandezza di un cluster. Per esempio, se si utilizza la coesione nei Graph-Based il peso da utilizzare è $1/m_i$, mentre per i Prototype-Based si utilizza il peso 1.

Per migliorare la qualità dei cluster, si possono clusterizzare i record rispetto a uno specifico valore della validità del cluster, tramite la coesione o la separazione. È possibile valutare la validità dei punti all'interno di un cluster: i record più interni contribuiscono di più alla coesione e separazione dei cluster rispetto a quelli esterni. Per questa valutazione si considera la *Silhouette Coefficient*, combinando le misure di separazione e coesione per l'i-esima riga:

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)} \in [-1, +1]$$

- a_i indica la distanza medie rispetto alle altre righe all'interno del cluster;
- b_i indica la minima distanza degli altri cluster.

Un valore negativo implica una distanza media dei punti nel cluster (a_i) maggiore della minima distanza media dei punti degli altri cluster (b_i). Una misura di validità di clusterizzazione è l' ***Average Silhouette Coefficient***, definita come la media dei singoli coefficienti dei punti appartenti al cluster.

Un'altra metrica, utilizzata soprattuto nei clustering gerarchici, è la ***Cophenetic Corelation Coefficient***, che misura il grado di similarità tra la matrice di prossimità P e la matrice Cophenetica Q, definita nell'intervallo $[-1, +1]$, dove 1 indica la similarità tra le due matrici ed un buon fit di gerarchizzazione dei dati.

6.1.3 Relative Measure

L'indice relativo, chiamato in inglese ***Relative Measure***, compara diversi cluster in termini supervisionati o non supervisionati. Gli indici interni ed esterni richiedono test statistici che possono portare ad un alto peso computazionale. Essa elimina questo requisito e si concentra sul confronto dei risultati della clusterizzazione, utilizzando diversi algoritmi o lo stesso con diversi parametri di input.

In questo caso si introduce il *Problema fondamentale* della validità di un cluster: **determinare il numero ideale di cluster**. Per risolvere questo problema si possono proiettare i dati in 2/3 dimensioni Euclidee utilizzando tecniche di visualizzazione che possono fornire il numero ideale. Questo metodo è utilizzato solo per piccoli scopi di applicazione. Esistono altri indici per risolvere il problema:

- Calinski e Harabasz, per la quale il massimo valore di K corrispondente è preso dal numero ottimale di cluster;
- Dunn, per la quale il massimo valore di K corrispondente è preso dal numero ottimale di cluster. Un alto valore indica cluster compatti e ben separati;
- Davies-Bouldin, per la quale il minimo valore di K corrispondente è preso dal numero ottimale di cluster.

Per modelli di mistura di probabilità si presentano i principali indici:

- ***A***kaike ***I***nformation ***C***riterion (AIC), per la quale il minimo valore di K corrispondente è preso dal numero ottimale di cluster;
- ***M***inimum ***D***escription ***L***ength (MDL), per la quale il minimo valore di K corrispondente è preso dal numero ottimale di cluster;
- ***B***ayesian ***I***nformation ***C***riterion (BIC), per la quale il minimo valore di K corrispondente è preso dal numero ottimale di cluster.

Per un algoritmo di clusterizzazione che richiede il numero dei K cluster dall'utente, il numero ottimale di cluster è ottenuto compiendo un algoritmo r volte dove K è compreso tra un valore minimo e massimo scelto dall'utente. Inanzitutto bisogna scegliere l'algoritmo di clusterizzazione ed un indice di validità; successivamente si compie un ciclo nella quale il valore $K = K_{min}$ e si fa andare l'algoritmo di clustering, si calcola il valore dell'indice e $q(i) = 1$. Dopo aver concluso i due cicli si sceglie il valore migliore di $\vec{q}$ tra i valori proposti e lo si setta pari a $Q(k) = \vec{q}$. Nel caso di strutture gerarchiche, gli indici di definiscono come *stopping rule*, che affermano il livello ottimale per il taglio del dendrogramma.

6.2 Validity Paradigm

Sia le misure interne che esterne sono strettamente correlate con metodi statistici e test d'ipotesi. Il paradigma di validità dei cluster si basa sulla seguente ipotesi nulla: "Non esiste alcuna struttura nel dataset".

Il paradigma procede nelle seguenti fasi:

1. Si identifica la struttura dei dati tramite un algoritmo di clusterizzazione ed il tipo di valutazione, interna od esterna;
2. Si determina l'indice di validità da utilizzare;
3. Si definisce l'ipotesi nulla della struttura nulla. Si possono utilizzare tre tipi di ipotesi: la *Random Position Hypotesis*, in cui tutte le posizioni dei dati di una specifica regione dello spazio sono equamente probabili (usata per dati numerici); la *Random Graph Hypotesis*, nella quale il ranghi delle matrici di prossimità sono equamente probabili (usata per prossimità ordinali); la *Random Label Hypotesis*, nella quale le etichette dei dati sono equamente probabili (usata per tutti i tipi di dati);
4. Si stabilisce la distribuzione di fondo sotto la condizione d'ipotesi nulla, utilizzando metodi computazionali quali l'analisi di Montecarlo e il Bootstrapping;
5. Si calcola l'indice associato alla soluzione di clustering utilizzata;
6. Si testa l'ipotesi della struttura nulla comparando il valore dell'indice precedente sulla distribuzione di fondo dell'ipotesi nulla con un livello di confidenza pari ad α.

7 Association Analysis

L'analisi delle associazioni analizza le transazioni effettuate, ovvero insiemi di elementi di lunghezza variabile che corrispondono ad operazioni unitarie effettuate dai clienti. Un esempio è la lista della spesa degli utenti di

un supermercato. L'obiettivo è trovare rapporti di conseguenza, *antecedenti* e *conseguenti* che offrono dei vantaggi di strategia. I dati sono rappresentati come variabili *booleane* nella quale ogni riga rappresenta una transazione e le collone ogni *item* (in questo caso i prodotti acquistati) nella transazione.

Sia $I = i_1, i_2, ..., i_k$ l'insieme degli item e con $T = t_1, t_2, ..., t_k$ l'insieme delle transazioni. Un ***itemset*** è un insieme di item; se contiente k item differenti, si chiama *k-itemset*. Una transazione contiene più itemset di dimensioni diverse, avendo la possibilità di eleminare o aggiungere elementi. La larghezza della transazione rappresenta il numero di item della transazione.

Il *support count* $\sigma(X) = |\{t_i : X \subseteq t_i, t_i \in T\}|$ conta il numero di k-itemset per determinare regole di associazione $A \to B$, a patto che $A \cap B = \emptyset$. Si definisce *confidenza* il numero di volte in cui l'assocazione si ripete rispetto al numero di volte in cui appare l'antecedente, in percentuale:

$$c(A \to B) = \frac{\sigma(A \cup B)}{\sigma(A)}$$

Invece, il *supporto* è definito come il numero in cui una regola è applicata all'intero dataset, in percentuale:

$$s(A \to B) = \frac{\sigma(A \cup B)}{N}$$

L'idea è trovare regole con un grande support, poichè facili da trovare, ed alta confidenza, perchè attendibili.

Il supporto elimina anche regole che occorrono per puro caso, o regole poco applicabili e perciò non interessanti nel mondo reale, riducendo quindi il numero di regole considerate. Per trovare il numero ideae è sconsigliato il metodo a *forza bruta*, in quanto l'elevato numero di regole segue una legge esponenziale, quindi ad alto peso computazionale. Il problema si risolve eliminando tutte le regole che si è sicuri non soddisfano una soglia minima di frequenza, poichè poco presenti nell'itemset e considerare le regole solo per gli elementi rimanenti.

7.1 Valutazione

La formazione di regole non deve essere troppo semplice e deve interessare gli esperti di dominio. Esistono misure oggettive, indipendenti dal dominio, per calcolare la qualità delle regole formulate: si costruisce una matrice di confusione (come quella per i modelli di classificazione). Si calcolano quindi le frequenze marginali della tabella per verificare che la regola aggiunga informazione: la confidenza è calcolata in relazione al supporto del conseguente. Infatti, l'antecedente è gia verificato dall'algoritmo di formulazione, ma solamente regole con un conseguente frequente possono aggiungere informazione. Si calcola quindi il *fattore di interesse*:

$$Lift = \frac{c(A \to B)}{s(B)}$$

oppure

$$I(A, B) = \frac{s(A, B)}{s(A) \cdot s(B)} = N \frac{f_{1.1}}{f_{1.\cdot} \cdot f_{\cdot.1}}$$

che assume il valore 1 in caso di indipendenza tra i due item. Si calola anche il *Coefficiente di Correlazione*:

$$Phi = \frac{f_{1.1} \cdot f_{0.0} - f_{0.1} \cdot f_{1.0}}{\sqrt{f_{1.\cdot} \cdot f_{\cdot.1} \cdot f_{0.\cdot} \cdot f_{\cdot.0}}}$$

o l'*IS Measure*:

$$IS(A, B) = \sqrt{I(A, B) \cdot s(A, B)} = \frac{S(A, B)}{\sqrt{s(A) \cdot s(B)}}$$

che è una buona misura per verificare l'associazione di parole all'interno di testi; è semplificabile $\sqrt{s(A) \cdot s(B)}$ come in caso di indipendenza tra le due variabili.

Una misura si dice *simmetrica* se scambiando antecedente e conseguente il suo valore non cambia, *asimmetrica* altrimenti. Esistono comunque una quarantina di misure diverse per calcolare il valore di una regola, e la letteratura suggerisce anche delle particolari misure per ogni dominio. Alcuni attributi sono più adatti a misurare le prestazioni in caso di regole simmetriche o asimmetriche.