Politecnico di Torino

Pre-Lab 5

Business Intelligence per Big Data

Classificazione | **Data preparation** | **Modelling**

Decision Tree | **Evaluation** | **Cross Validation**

Eleonora Poeta, Meryem Ennadi

Obiettivo 1

Classificazione introduzione

Business understanding

Classificazione – definizione e applicazione

Cos'è la classificazione?

La classificazione è un compito di apprendimento supervisionato: dato un insieme di esempi etichettati (training set), si costruisce un modello in grado di assegnare una classe a nuovi esempi non etichettati.

Due fasi:

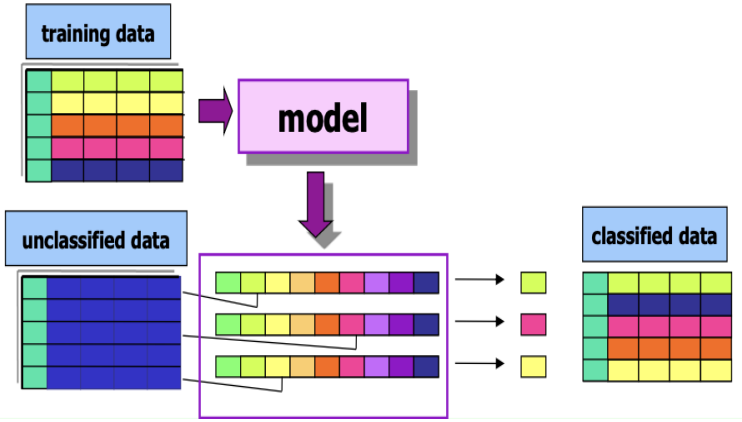
- Training: il modello 'impara' dai dati etichettati (Users.xls)
- Prediction: il modello predice la classe di nuovi dati (Prospects.xls)

Differenza da clustering:

- Clustering: non supervisionato, nessuna etichetta nota
- Classificazione: supervisionato, etichette note nel training

Obiettivi:

- **Predizione:** Assegnare un'etichetta di classe a dati futuri o sconosciuti.
- **Interpretabilità:** Fornire un modello comprensibile (es. **Decision Tree**) che spieghi il fenomeno analizzato.



Nel Lab: useremo I dataset Users.xls e Prospects.xls

Classificazione – Metriche di valutazione e componenti

Metriche di valutazione

- **Accuracy (Accuratezza):** la qualità della predizione.
- **Interpretability (Interpretabilità):** la capacità del modello di fornire spiegazioni comprensibili.
- **Efficiency (Efficienza):** in termini di tempo e risorse.
- **Scalability (Scalabilità):** la capacità di mantenere prestazioni elevate all'aumentare del dataset.
- **Robustness (Robustezza):** la capacità di gestire correttamente valori mancanti o eccezioni senza compromettere la classificazione.
- **Incremental (Incrementalità):** indica se il modello può essere aggiornato con nuovi dati senza dover essere ricostruito interamente da zero.

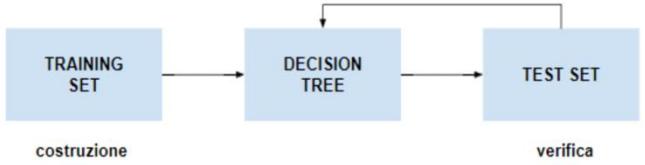
Componenti

Training Set

È una collezione di oggetti **labeled** (etichettati con una classe nota) utilizzati per "insegnare" al modello come effettuare la classificazione, permettendogli di apprendere le relazioni tra attributi e target.

Test Set

È una collezione di dati **labeled** (etichettati) che non sono stati utilizzati durante la fase di addestramento. Lo scopo è quello di **validare** la capacità di generalizzazione del sistema.



Nel Lab: pipeline "Read Excel" -> "Set Role" impostare **attribute name:"Response"** e **target role: "label"**

Obiettivo 1

Decision Tree

Tecniche di classificazione

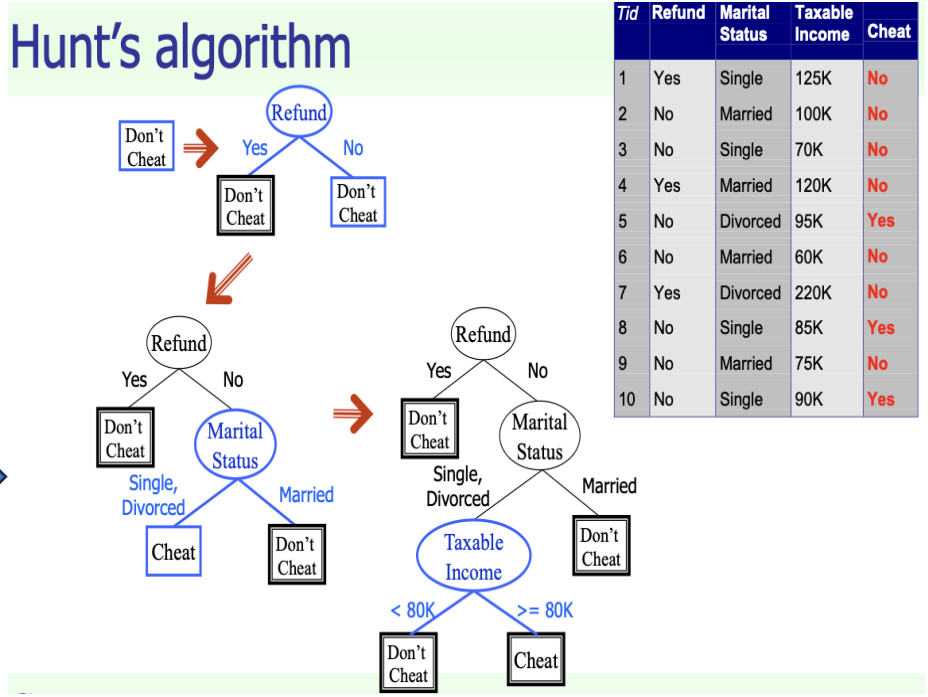
Decision Tree

Che cos'è un Decision Tree?

Un **Decision Tree** è un modello di classificazione.
Permette di rappresentare graficamente una serie di regole decisionali che portano alla **classificazione** di un dato.

Funzioni e caratteristiche principali:

- **Struttura Predittiva e Interpretabile**
- **Suddivisione dei Dati (Splitting):** Serve a dividere ricorsivamente il dataset in sotto-gruppi sempre più omogenei rispetto alla classe di appartenenza. Per fare ciò, utilizza algoritmi (come quello di **Hunt**) che scelgono ad ogni passo l'attributo "migliore" per separare i dati.
- **Gestione di Diversi Tipi di Dati:** Può gestire attributi nominali o continui applicando differenti criteri di suddivisione (split binari o multi-via).
- **Efficienza**



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Nel Lab: aggiungere alla pipeline "Read Excel" -> "Set Role" -> "Decision Tree"

Pipeline Completa – Training e Prediction

Fase 1 – Costruire e salvare il modello (Users.xls)

1. Read Excel	2. Set Role	3. Decision Tree	4. Store
Caricare Users.xls	Response → label	Costruisce il modello	Salva il modello

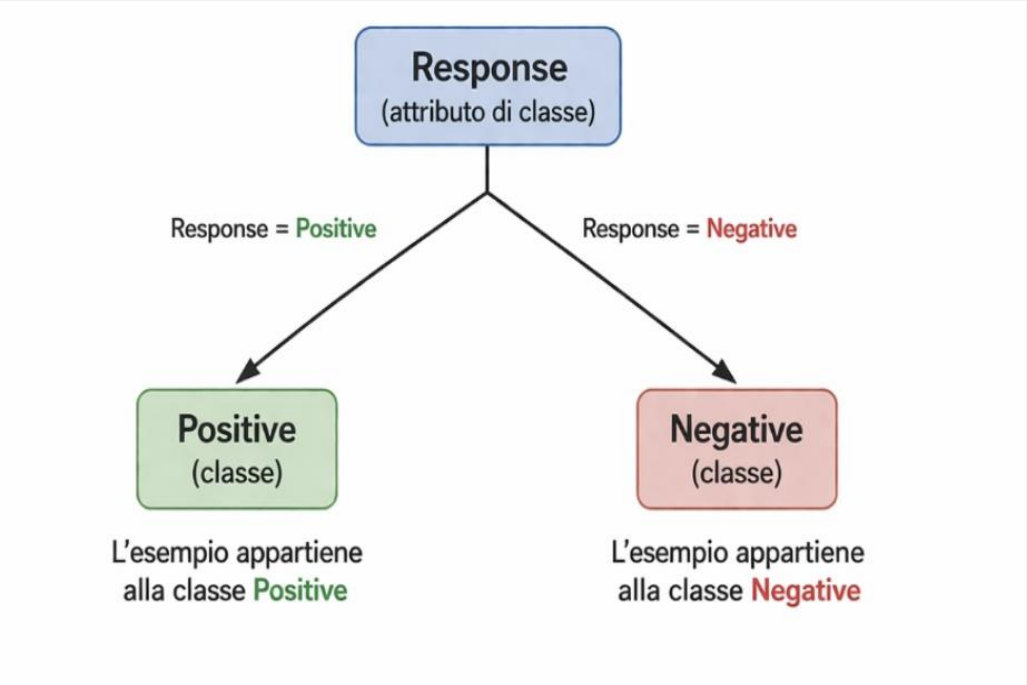
Fase 2 – Applicare il modello (Prospects.xls)

1. Read Excel	2. Retrieve	3. Apply Model	4. Output
Caricare Prospects.xls (senza Response)	Caricare il modello salvato	Predice la classe per ogni prospect	prediction(Response) + probabilità P/N

📋 Nel Lab: Processo 1 → Store modello. Processo 2 → Retrieve + Apply Model su Prospects. Output: attributo prediction(Response).

Applicazione del Decision Tree -Part 1

- **Training**: analizza i 1000 record del dataset Users.xls di cui conosciamo già l'esito (se si sono iscritti o meno) per "imparare" le regole che portano a una risposta positiva o negativa.
- **Identificazione dei criteri di scelta**: Serve a capire quali attributi (vedi slide precedente) sono i più **selettivi** e influenzano la decisione finale dell'utente.
- **Base per la previsione futura**: Questo modello "addestrato" è indispensabile per il passaggio successivo, ovvero essere applicato ai 30.000 nuovi contatti (Prospects.xls).



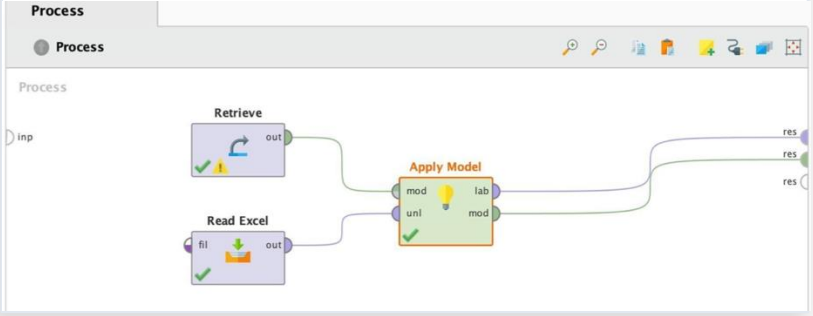
Nel Lab: impostare dentro l'operatore "Decision Tree" I parametri e aggiungere alla pipeline l'operatore "Store"

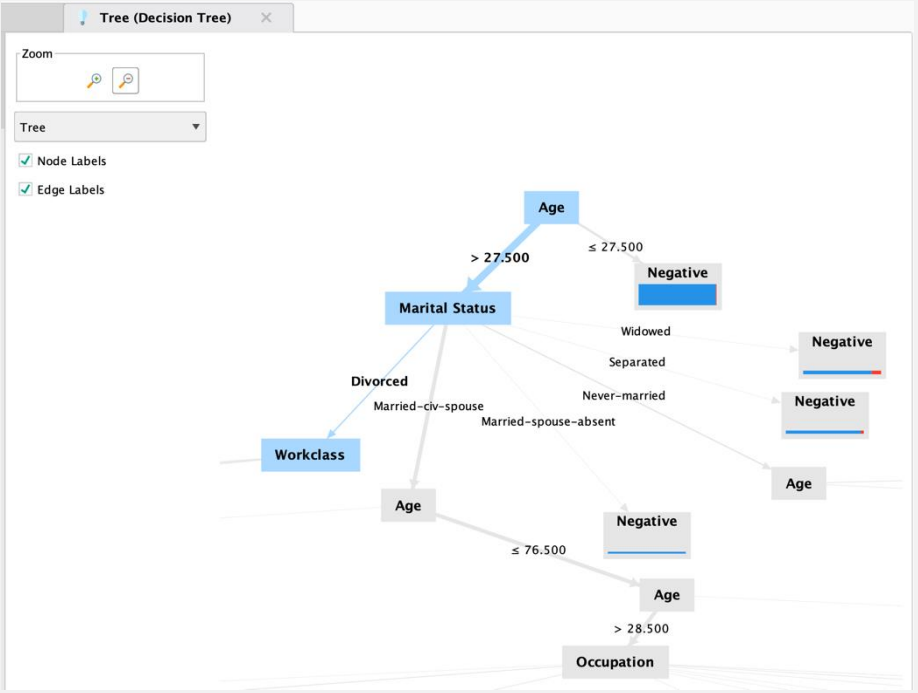
Applicazione del Decision Tree –Part 2

A cosa ci serve?

Una volta che abbiamo creato e addestrato il modello
lo applichiamo ai dati del secondo dataset:" Prospects.xls"
per classificare i nuovi utenti in una delle due classi:

1. Classe Positive (Response= Positive)
2. Clsse Negative (Response= Negative)





Nel Lab: usare l'operatore "Store" per salvare il modello e una volta salvato usare l'operatore "Retrieve" per caricarlo e usarlo sul dataset "Prospects.xls"

Obiettivo 2

Validazione dei modelli di classificazione

Tecniche di validazione

Validazione dei modelli di classificazione

Cos'è la validazione?

Validare un modello significa misurare la sua capacità di fare previsioni corrette su dati nuovi, che non sono stati utilizzati durante la fase di training. Questo processo avviene applicando il modello a un Test set per confrontare le predizioni con la realtà.

Le componenti della validazione

- **Test Set(labeled):** dati non utilizzati nell'addestramento. Essendo etichettati permettono di calcolare l'errore confrontando la classe reale con quella predetta.
- **Cross-Validation:** tecnica che divide i dati in più blocchi per garantire una valutazione statica e non influenzata da una singola divisione fortunata.

Nel Lab: pipeline "Read Excel" -> "Set Role" -> "Cross Validation"

Metriche di Classificazione – Accuracy, Precision, Recall

Matrice di Confusione

		Predetto → Pos	Predetto → Neg
Reale ↓	Pos	**TP** True Positive	**FN** False Negative
	Neg	**FP** False Positive	**TN** True Negative

TP: predetto Pos, reale Pos ✓

TN: predetto Neg, reale Neg ✓

FP: predetto Pos, reale Neg ✗ (falso allarme)

FN: predetto Neg, reale Pos ✗ (mancato)

Formule

Accuracy — $(TP + TN) / (TP+TN+FP+FN)$

% di predizioni corrette. Fuorviante con classi sbilanciate.

Precision — $TP / (TP + FP)$

Tra chi ho predetto Positivo, quanti lo erano davvero? Misura il costo dei falsi allarmi.

Recall (Sensitivity) — $TP / (TP + FN)$

Tra chi era davvero Positivo, quanti ho trovato? Misura il costo dei casi mancati.

F1-Score — $2 \times (P \times R) / (P + R)$

Media armonica di Precision e Recall. Equilibra i due obiettivi.

📋 Nel Lab: analizzare matrice di confusione. Accuracy è affidabile? Quale classe ha precision/recall più alta?

Dataset Sbilanciato – Quando l'Accuracy non Basta

Problema: se il 90% dei dati è classe Negativa, un classificatore che predice sempre 'Negativo' ha 90% di accuracy – ma è inutile!

Esempio pratico

Dataset Users.xls:
~75% Negative, ~25% Positive

Un classificatore pigro che predice sempre 'Negative' avrebbe accuracy ~75%... ma non trova nessun cliente!

Cosa guardare invece:

- Recall sulla classe Positive: quanti clienti interessati troviamo?
- Precision sulla classe Positive: tra i contattati, quanti erano davvero interessati?

Strategie per dataset sbilanciati

Metriche migliori
Usare F1-Score, AUC-ROC o weighted metrics invece di sola accuracy.

Oversampling (SMOTE)
Generare esempi sintetici della classe minoritaria per bilanciare il dataset.

Undersampling
Ridurre gli esempi della classe maggioritaria.

Cost-sensitive learning
Penalizzare maggiormente gli errori sulla classe rara durante il training.

Nel Lab: L'accuracy è affidabile? Analizzare recall e precision su ciascuna classe. Su quale classe performa meglio il modello?

Obiettivo 2

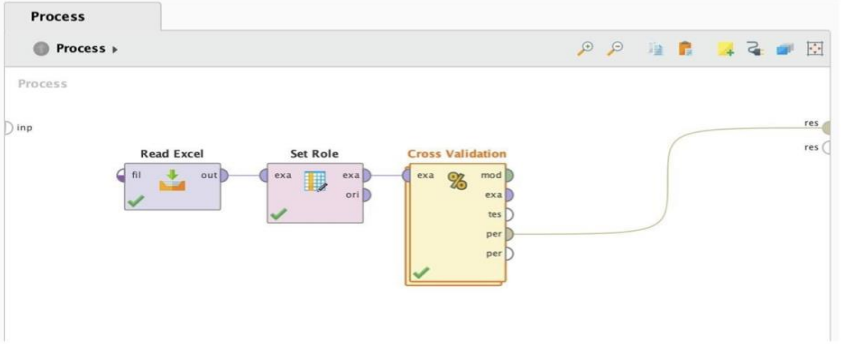
Cross Validation

Tecniche di validazione

Validazione dei modelli di classificazione

Cross Validation: A cosa ci serve?

Un modello può sembrare perfetto sui dati che già conosce, ma fallire totalmente con nuovi dati e nel mondo reale. La validazione serve a impedire l'**overfitting** fenomeno per cui il **Decision Tree** impara a "memoria" i dati di **"Users.xls"** perdendo la capacità di generalizzare ed eventualmente fallire nel classificare i dati del dataset **"Prospects.xls"**.



Nel Lab: pipeline "Read Excel" -> "Set Role" -> "Cross Validation"

Cross Validation

Come funziona?

La **Cross-Validation** è un operatore **complesso** che divide internamente il dataset in k parti (**fold**). Il processo è ciclico: a turno, **k–1** parti sono usate come **Training Set** e la restante come **Test Set**.

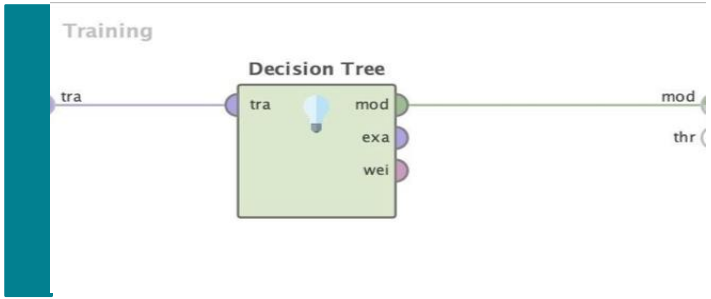
Parametri e Scelte Tecniche:

- **Numero di Fold: P**iù fold (es. 10) aumentano l'affidabilità statistica del risultato, ma richiedono tempi di calcolo più lunghi.

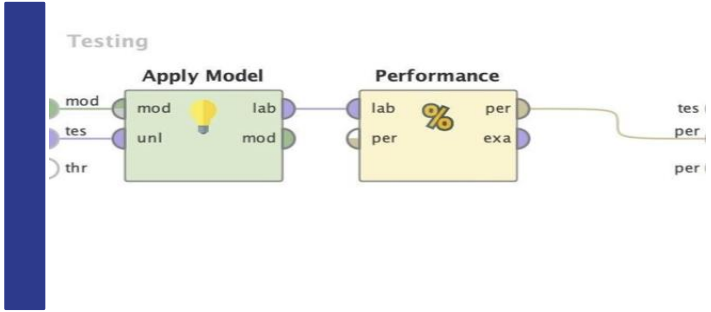
- **Sampling Type:** Definisce come i dati vengono mescolati (es. *stratified sampling* per mantenere le proporzioni delle classi).

- **Main Criterion:** Nel Lab impostiamo l'**Accuracy** come criterio principale per valutare quanto il classificatore "funge" bene.

Training



Testing

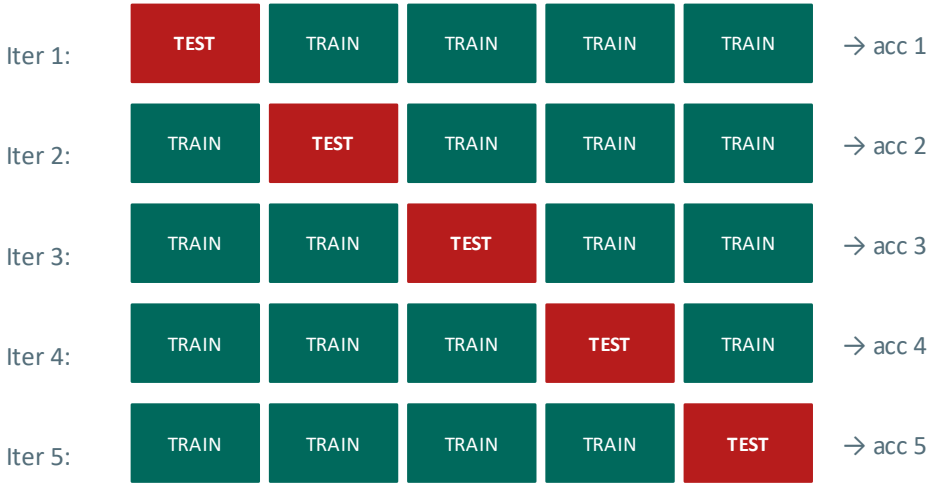


Nel Lab: pipeline dentro l'operatore "Cross Validation" -> "Decision Tree" + "Apply Model" + "Performance"

Cross-Validation – Validare Senza Dati di Test Separati

Problema: non possiamo usare Prospects.xls per validare (la classe reale non è nota). Usiamo K-Fold Cross-Validation sui dati di training.

K-Fold Cross-Validation (es. K=5)



Media delle 5 accuracy = stima robusta

Come funziona in RapidMiner

Operatore Cross Validation:

- Sottoprocesso Training: inserire l'algoritmo (Decision Tree)
- Sottoprocesso Testing: Apply Model → Performance (Classification)

Parametri:

- Number of folds (es. 10): più fold = stima più affidabile ma più lenta
- Leave-one-out: K=N, usato solo con dataset molto piccoli

Output:

- Uscita 'per': accuracy, precision, recall, matrice di confusione media sui K fold

Nel Lab: Cross Validation → Training (Decision Tree) → Testing (Apply Model + Performance). Analizzare matrice di confusione.

Riepilogo del Lab

① Import dati

② Apply Model

③ Classificazione e costruzione del Modello

④ Decision Tree

⑤ Cross Validation

⑥ Valutazione

Buon laboratorio!