Politecnico di Torino

Pre-Lab 1

Business Intelligence per Big Data

Preprocessing di dati strutturati e testuali con RapidMiner/Python

Import dati

Missing values

Outlier detection

Discretizzazione

Normalizzazione

Text Mining

Eleonora Poeta

Agenda del Lab

1

Import dei dati strutturati

Operatore Read Excel – UsersSmall.xls

2

Gestione dei valori mancanti

Declare Missing Values + Replace Missing Values

3

Outlier Detection (univariata)

Istogramma, Scatter plot, Filter Examples

4

Discretizzazione

By binning / frequency / size + confronto

5

Normalizzazione

Normalize – quando e perché

6

Preprocessing testuale

Tokenize, Stopwords, Stemming, TF-IDF

Obiettivo 1

Import dei Dati

Caricare dati strutturati in RapidMiner

Import dei Dati – Concetti Chiave

Cos'è il preprocessing?

Il preprocessing è la fase preliminare di qualsiasi analisi dati. Prima di applicare algoritmi di data mining è necessario:

- Importare i dati
- Comprendere la struttura e la semantica degli attribute
- Identificare il ruolo di ciascuna variabile (feature vs label)
- Rilevare anomalie, errori, valori mancanti

Dataset UsersSmall.xls

- Age (integer)
- Workclass
- Education
- Marital Status
- Occupation
- Relationship
- Race
- Sex
- Native Country
- **Response ← label**

Nel Lab: Usare Read Excel + Import Configuration Wizard per caricare UsersSmall.xls e analizzare gli attributi.

Obiettivo 2

Gestione dei Valori Mancanti

Identificare e rimediare ai missing values

Missing Values – Cosa, Perché, Come

Cosa sono?

- Valori assenti in uno o più attributi di un record.
- Possono essere rappresentati come:
  - ❖ celle vuote;
  - ❖ simboli speciali ('?', 'N/A', etc);
  - ❖ valori numerici sentinella.
- Nel dataset UsersSmall il simbolo '?' denota un valore mancante.

Perché trattarli?

- Molti algoritmi di ML non accettano valori mancanti.
- Possono introdurre bias o ridurre la qualità del modello.
- Ignorarli riduce la dimensione del dataset utile.

Tecniche di gestione

- *Eliminazione*:
  - rimuovere i record
  - rimuovere gli attributi incompleti.
- *Imputazione*
  - con media/mediana (numerici) o moda (categoriali).
  - con modelli predittivi (es. KNN, regressione).
- *Dichiarazione* esplicita: marcare come NULL e gestire a valle.

📋 Nel Lab: Declare Missing Value (simbolo '?') → Replace Missing Values (valore più frequente in media).

Obiettivo 3

Outlier Detection

Identificare e gestire valori anomali

Outlier Detection – Concetti e Tecniche

Cosa sono gli outlier?

Osservazioni che si discostano significativamente dalla distribuzione attesa dei dati.

Cause comuni:

- Errori di misurazione o inserimento dati
- Errori di trasmissione o acquisizione
- Fenomeni reali ma rari (frodi, eventi estremi)

Impatto:

- Distorcono media e deviazione standard
- Degradano le performance dei modelli predittivi

Approcci di rilevamento

Univariato

Analisi di un singolo attributo: istogramma, boxplot, regola ±3σ.

Bivariato

Scatter plot tra coppie di attributi – anomalie nella distribuzione congiunta.

Multivariato

Analisi delle distanze tra punti (es. distanza euclidea)
Individuazione di osservazioni lontane dal centro dei dati

Nel Lab: Histogram (10 bins) sull'attributo Age → Scatter plot vs altri attributi → Filter Examples per rimuovere outlier.

Obiettivo 4

Discretizzazione

Trasformare attributi continui in categorie

Discretizzazione – Perché e Come

Perché discretizzare?

Alcuni algoritmi (es. Naive Bayes, regole associative) richiedono *attributi categorici*. La discretizzazione converte valori continui in intervalli, riducendo il rumore e rendendo il modello più interpretabile e/o allineato al contesto applicativo.

Discretize by Binning

Divide il range dei valori *in k intervalli* di ampiezza uguale.

✓ Pro: Semplice, rapida

✗ Contro: Sensibile agli outlier

Discretize by Frequency

Divide i dati *in k classi* contenenti lo **stesso numero** di **osservazioni**

✓ Pro: Distribuzione uniforme

✗ Contro: Intervalli di ampiezza variabile

Discretize by Size

Crea intervalli di *dimensione* (numero di esempi) *fissa* definita dall'utente (variante di discretizzazione per frequenza)

✓ Pro: Controllo granulare

✗ Contro: Numero di bin variabile

Altri approcci: discretizzazione supervisionata e discretizzazione non supervisionata via clustering.

📋 Nel Lab: Applicare le 3 tecniche sull'attributo Age. Confrontare con number_of_bins = 5. Bonus: Generate Sales Data + Correlation Matrix + Normalize.

Obiettivo 4 Bonus

Correlazione & Normalizzazione

Analisi delle relazioni tra variabili e scaling dei dati

Correlazione e Normalizzazione

Correlazione tra attributi

Il coefficiente di **correlazione** di **Pearson r** misura la relazione lineare tra due variabili (-1 a +1).

- r ≈ +1 → correlazione positiva forte
- r ≈ 0 → assenza di correlazione lineare
- r ≈ −1 → correlazione negativa forte

Attributi altamente correlati sono ridondanti e possono peggiorare alcuni modelli (es. regressione).

💡 Possono essere **rimossi** o **combinati** con PCA.

Operatore: Correlation Matrix

Normalizzazione

Min-Max [0,1]

x' = (x−min)/(max−min). Scala al range [0,1]. Sensibile agli outlier.

Z-score (Standardizzazione)

x' = (x−μ)/σ. Media 0 e dev.std 1. Robusto, usato da SVM e KNN.

Normalizzazione L2

Scala i vettori a norma 1. Utile per dati testuali e cosine similarity.

📋 Nel Lab: Generate Sales Data → Select Attributes (numerici) → Correlation Matrix → Normalize. Discutere quando è necessario normalizzare.

Obiettivo 5

Preprocessing Testuale

Trasformare documenti in rappresentazioni numeriche

Text Preprocessing – Pipeline NLP

1. Tokenize

Divide il testo in unità atomiche (parole/token).

L'ordine non viene mantenuto → Bag of Words.

2. Transform Cases

Converte in minuscolo (o maiuscolo) per uniformare:

'Cibo' e 'cibo' diventano lo stesso token.

3. Filter Stopwords

Elimina parole non informative:

articoli, preposizioni, congiunzioni (es. 'il', 'di', 'the').

4. Stem (Snowball)

Riduce ogni parola alla propria radice morfologica.

'correndo' → 'corr'.
Aumenta il recall.

TF-IDF (Term Frequency – Inverse Document Frequency)

tfidf(i,j) = tf(i,j) × log(N / df_i) dove: tf = n. occorrenze del termine i nel documento j | df = n. documenti contenenti i | N = n. totale documenti.

📋 Nel Lab: Process Documents From Files (Wikipedia) → sottoprocesso con Tokenize → Transform Cases → Stem → Filter Stopwords → matrice documenti×termini TF-IDF.

Riepilogo del Lab

① Import dati

Read Excel → analisi semantica attributi

② Missing Values

Declare (?) → Replace (moda/media)

③ Outlier Detection

Histogram → Scatter → Filter Examples

④ Discretizzazione

Binning / Frequency / Size – confronto k=5

⑤ Corr. & Normalizz.

Correlation Matrix → Normalize

⑥ Text Preprocessing

Tokenize → Cases → Stopwords → Stem → TF-IDF

Buon laboratorio! 🚀