Titolo:
Integrazione di Dati Automobilistici da Sorgenti Eterogenee tramite Record Linkage, Blocking e Modelli di Machine Learning.

Corso:
Ingegneria dei Dati 2025/2026 Homework: 6

Autori:
Alessandro di Mario 547297, Pietro Barone 579635, Guido Petrolito 629688, Douglas Ruffini 482379.

Introduzione

Quando più dataset descrivono entità reali simili (in questo caso automobili usate), ma con schemi differenti, dati rumorosi e rappresentazioni inconsistenti, diventa necessario progettare una pipeline che permetta:

1. La caratterizzazione delle sorgenti
2. La definizione di uno schema mediato
3. L'allineamento delle sorgenti allo schema mediato
4. L'identificazione dei record che si riferiscono alla stessa entità reale (*record linkage*)

In questo progetto sono state utilizzate due grandi sorgenti pubbliche:

- Craigslist Cars & Trucks
- DatasetUS Used Cars Dataset

L'obiettivo è stato costruire una pipeline sperimentale completa per confrontare diverse strategie di record linkage, valutandole in termini di **Precision**, **Recall**, **F1-measure**, **tempi di training** e **tempi di inferenza**.

Punto 1 - Caratterizzazione delle sorgenti

Obiettivo
Per ciascuna sorgente analizzare la percentuale di valori nulli e di valori unici di ciascuno attributo.

Implementazione
E' stata eseguita un'analisi statistica considerando l'intero dataset, andando ad eseguire il conteggio dei valori nulli e distinti tramite caricamento completo in memoria.

Conclusioni
L'analisi ha evidenziato che attributi come `MANUFACTURER/MAKE_BRAND` e `YEAR`, hanno una densità informativa elevata con bassa percentuale di nulli. Al contrario, attributi descrittivi come `CONDITION` o `CYLINDERS` presentano in Craigslist un'elevata percentuale di valori mancanti [>30%], suggerendo cautela nel loro utilizzo per il blocco o il confronto rigido.

Poiché il numero di attributi per ciascuno dataset è considerevole, riportiamo esclusivamente le statistiche degli attributi utilizzati nelle fasi successive per lo schema mediato.

Source	Null %	Valori Unici
VIN	37.73	118264
manufacturer	4.13	42
model	1.24	28576
year	0.28	114
price	0.00	15655
odometer	1.03	104870
fuel	0.71	5
transmission	0.60	3
state	0.00	51
region	0.00	404
description	0.02	N/A (Text)

Sorgente: Craigslist Cars & Trucks Data

Source	Null %	Valori Unici
vin	0.00	3000000
make_name	0.00	100
model_name	0.00	1428
year	0.00	98
price	0.00	88861
mileage	4.81	197577
fuel_type	2.76	8
transmission	2.14	4
description	2.60	N/A (Text)

Sorgente: US Used Cars Dataset

Punto 2 - Definizione dello Schema Mediato

E' stato definito uno schema logico comune:

```
MEDIATED_SCHEMA = ["vin", "make", "model", "year",
"price", "mileage", "fuel", "transmission",
"description", "state", "region"]
```

Mentre il software farebbe Automatic Schema Matching, l'essere umano esegue un matching manuale. Questo processo si divide in due approcci teorici a seconda di come viene costruito lo schema:

- *Bottom-up:* Parti dalle sorgenti e costruisci lo schema mediato che le comprenda tutte.
- *Top-down:* Definisci prima lo schema ideale (mediato) e poi cerchi di capire come "incastrare" le sorgenti esistenti al suo interno.

Nel nostro caso la strategia adottata è stata **Top-down**.

Punto 3 - Allineamento delle sorgenti

Craigslist	UsedCars	Mediato
vin	vin	**vin**
manufacturer	make_name	**make**
model	model_name	**model**
year	year	**year**
price	price	**price**
odometer	mileage	**mileage**
fuel	fuel_type	**fuel**
transmission	transmission	**transmission**
state	-	**state**
region	-	**region**
description	description	**description**

Implementazione

Riportiamo le risultare della fase di allineamento delle sorgenti seguendo un codice cromatico: in **arancione** sono evidenziati i campi identici in entrambi i dataset, in **verde** le informazioni aggiuntive estratte da uno dei due dataset, e in **blu** gli attributi eterogenei che hanno richiesto una ridefinizione dello schema globale.

Per uniformare i dati, sono state eseguite diverse operazioni strutturali e di pulizia: **rinomina** delle **colonne**, **integrazione** degli **attributi mancanti**, **conversioni** dei **tipi numerici**, **pre-processing testuale** e **normalizzazione** del codice **VIN**.

Punto 4.A - Costruzione della Ground Truth tramite VIN

Problema
Il `VIN` (**Vehicle Identification Number**) è un identificatore univoco dell'automobile; nei dataset sono rumorosi (spazi, simboli, errori).

Soluzione
La funzione `clean_vin()`:
- ✓ rimuove caratteri non alfanumerici
- ✓ converte in maiuscolo
- ✓ mantiene solo VIN di 17 caratteri

Viene fatto un merge interno sui `VIN` puliti per ottenere coppie di record che sicuramente si riferiscono alla stessa auto. Questa parte permette di avere una ground truth automatica e affidabile, addestrare modelli ML, e valutare le performance senza annotazione manuale.

Intersezione: Sono stati identificati 3.959 VIN comuni tra le due sorgenti.

Punto 4.B, 4.C - Preparazione Dataset

Rimozione Vin: Dopo la costruzione della ground truth il VIN viene rimosso dai dati usati per il record linkage. Questo perché se il modello vedesse il VIN, apprenderebbe banalmente l'uguaglianza $VIN_A = VIN_B$, ottenendo prestazioni perfette ma irrealistiche (**over fitting** su un attributo identificativo)

Split: La Ground Truth totale (9.003 coppie incluse le duplicazioni interne o varianti) è stata divisa in:

- Train: 60%
- Validation: 20%
- Test: 20%

Punto 4.D - Strategie di Blocking

Per ridurre lo spazio di ricerca poiché il prodotto cartesiano risulta troppo oneroso, sono state definite due strategie:

- B1 (**Standard Blocking**): Si utilizza il blocco esatto sull'attributo `MAKE`.

  Analisi: Ha generato X coppie candidate. Recall Max: 1.00 (cattura tutte le vere corrispondenze).
- B2 (**Sorted Neighborhood**): Ordinamento e finestra scorrevole (±1 anno) sull'attributo `YEAR`.

  Analisi: Ha generato Y coppie candidate. Recall Max: 1.00.

Entrambe le strategie sono eccellenti in termini di copertura (Recall), ma lasciano passare molti falsi positivi che il modello di matching deve filtrare.

Punto 4.E - Record Linkage Rule-Based

Le Regole di Confronto (Similarity) Abbiamo scelto l'algoritmo migliore per ogni tipo di dato:

- **Marca:** *Jaro-Winkler* (Soglia: 0.9) - *Ottimo per intercettare errori di battitura.*
- **Modello:** *Jaro-Winkler* (Soglia: 0.8) - *Soglia più bassa per tollerare le sigle degli allestimenti.*
- **Anno:** *Levenshtein* (Soglia: 0.9) - *Confronto rigido sui caratteri.*
- **Prezzo:** *Gaussiano* (Tolleranza: 0.2) - *Curva a campana per ammettere fluttuazioni di prezzo minime.*

La Decisione Finale (Classificazione)

Una coppia viene promossa a **Match Definitivo (1)** solo se soddisfa almeno 3 di questi 4 criteri.

Condizione di Match: $\Sigma \geq 3$

Punto 4.F - Record Linkage con Machine Learning - Dedupe

1. **Automazione tramite Active Learning Simulato:** Per sfruttare la libreria Dedupe il modello è stato addestrato iniettando direttamente un sottoinsieme della *Ground Truth* (~50 coppie).

2. **Costruzione e Bilanciamento del Training Set:** Affinché il classificatore potesse apprendere correttamente la frontiera di decisione, la Ground Truth (composta da soli **Positivi** estratti per intersezione esatta) è stata integrata dinamicamente con esempi controfattuali tramite:
   - **Random Negative Sampling:** Generazione iterativa di coppie casuali cross-dataset per simulare i *Non-Match*.
   - **Validazione Deterministica:** Applicazione di un controllo rigoroso *(vin1 != vin2)* su ogni coppia negativa generata.
   - **Bilanciamento delle Classi:** Raggiungimento di un rapporto **1:1** (50% Match, 50% Non-Match) essenziale per prevenire fenomeni di *Bias* e *Overfitting* sulla classe maggioritaria.

4.G - Record Linkage con Deep Learning - Ditto

1. Architettura e Vantaggio Semantico

Utilizzo di un approccio basato su reti neurali profonde e architettura **Transformer** (BERT): il modello non calcola semplici distanze tra stringhe, ma apprende il **significato semantico** e il contesto, gestendo nativamente sinonimi, acronimi e dati eterogenei.

2. Serializzazione dei Record

Trasformazione dei dati tabulari in sequenze testuali continue; con l'utilizzo di token speciali (COL e VAL) per preservare la struttura dello schema (es. `COL make VAL ford COL model VAL fiesta`).

3. Classificazione e Fine-Tuning

Il problema è stato trattato come classificazione binaria di sequenze (1:*Match*, 0:*Non-Match*). Il modello è stato poi specializzato tramite *Fine-Tuning* su dataset bilanciati al 50%, garantendo un'ottima generalizzazione e prevenendo l'overfitting.

Punto 4.H - Valutazione prestazioni Record Linkage, Dedupe e Ditto [Campionato]

Record Linkage	Precision	Recall	F1	Time
Blocking[Make]	0.06	0.95	0.11	2.765s
Blocking[Year]	0.06	0.95	0.12	2.390s

Dedupe	Precision	Recall	F1	Training	Inference
Blocking[Make]	0.15	0.70	0.24	2.765s	234.584s
Blocking[Year]	0.15	0.71	0.24	2.390s	206.089s

Ditto	Precision	Recall	F1	Training	Inference
Blocking[Make]	0.973	0.881	0.9251	10m	94.0s
Blocking[Year]	0.973	0.881	0.9251	10m	94.0s

Punto 4.H - Valutazione prestazioni Record Linkage, Dedupe e Ditto [Non Campionato]

Record Linkage	Precision	Recall	F1	Time
Blocking[Make]	0.04	0.94	0.07	9.204s
Blocking[Year]	0.04	0.94	0.07	9.084s

Dedupe	Precision	Recall	F1	Training	Inference
Blocking[Make]	0.12	0.57	0.20	8.80s	134.556s
Blocking[Year]	0.12	0.57	0.20	8.80s	151.238s

Ditto	Precision	Recall	F1	Training	Inference
Blocking[Make]	0.973	0.881	0.9251	10m	94.0s
Blocking[Year]	0.973	0.881	0.9251	10m	94.0s

Conclusioni: L'Efficacia di Ditto nel Record Linkage

Il Vantaggio Semantico (Deep Learning vs. Regole Classiche)
Sfruttando un Language Model, Ditto comprende il contesto e la semantica dei dati (es. acronimi, varianti negli allestimenti), superando i limiti strutturali delle metriche di distanza classiche come Jaro-Winkler.

Sinergia tra Modello e Qualità del Dato
Le prestazioni del modello sono supportate da una rigorosa pipeline di data preparation: applicazione di *Smart Filtering* (rimozione rumore e nulli) e *Random Negative Sampling* bilanciato per un addestramento privo di bias.

Valutazione Finale In scenari complessi caratterizzati da dati eterogenei (User Generated Content di Craigslist vs. dati strutturati), Ditto garantisce metriche di livello industriale (F1 > 0.94), confermandosi la soluzione più robusta per l'integrazione dati in ambito automotive.