

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 2 Homework - 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 1 Homework - 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 **MAKE** e **YEAR**, hanno una densità informativa elevata con pochi valori nulli. Al contrario, attributi come **MODEL**, possono essere poco informativi, si evidenzia il numero di valori di valori unici elevato, che fa pensare ad ambiguità nell'inserimento.

HomeWork 6

Ingegneria dei Dati

Attributo	Nulli (%)	Valori Unici
VIN	37.73	118264
make	4.13	42
year	0.28	114
model	1.24	28576
price	0.00	15655
mileage	1.03	104870
fuel	0.71	5
transmission	0.00	51
state	99.393	3
region	0.00	404
description	0.02	N/A

Sorgete: Craigslist Cars & Trucks Data

Attributo	Nulli (%)	Valori Unici
VIN	0.00	3000000
make	0.00	100
year	0.00	98
model	0.00	1428
price	0.00	88861
mileage	4.81	197577
fuel	2.76	8
transmission	2.14	4
state	-	-
region	-	-
description	2.60	N/A

Sorgete: US Used Cars Dataset

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

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

(Punto 3 Homework) - 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

Abbiamo eseguito un allineamento delle sorgenti, in arancio sono evidenziati gli attributi identici dai due dataset, in verde invece gli attributi del dataset A che hanno fornito informazioni aggiuntive non presente nel B. Il blu sono rimasti gli attributi differenti per cui è stato necessario ridefinire lo schema.
rinomina colonne, aggiunta colonne mancanti, conversioni numeriche, preprocessing testuale, pulizia del VIN

Costruzione della Ground Truth tramite VIN (Punto 4.A)

Il VIN (Vehicle Identification Number) è un identificatore univoco dell'automobile.

Problema

VIN 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.

Preparazione Dataset (Punto 4.B, 4.C)

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 (overfitting 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%

Strategie di Blocking (Punto 4.D)

Per ridurre lo spazio di ricerca (prodotto cartesiano troppo oneroso), sono state definite due strategie:

- B1 (Standard Blocking): Blocco esatto sull'attributo make.
 - Analisi: Ha generato X coppie candidate. Recall Max: 1.00 (cattura tutte le vere corrispondenze).
- B2 (Sorted Neighborhood): Finestra di dimensione 1 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.

Record Linkage Rule-Based (Punto 4.E)

Sono state implementate e confrontate due strategie mediante una regola deterministica basata su:

Strategie di Campionamento e Hard Negatives

Campionamento Stratificato

Hard Negatives (Campionamento Critico)

Bilanciamento delle Classi

Record Linkage non campionario

RL-B1 | P=0.04 R=0.94 F1=0.07 Time=9.204s

RL-B2 | P=0.04 R=0.94 F1=0.07 Time=9.084s

Record Linkage campionario

RL-B1 | P=0.06 R=0.95 F1=0.11 Time=2.765s

RL-B2 | P=0.06 R=0.95 F1=0.12 Time=2.390s

Record Linkage con Machine Learning — Dedupe (Punto 4.F)

È stata utilizzata la libreria Dedupe, che implementa un apprendimento attivo. Il sistema propone all'utente (o in questo caso, simula tramite la GT) coppie ambigue da etichettare come "match", "non-match" o "uncertain".

Automazione del Training: Invece di etichettare manualmente le coppie (processo lento), il codice utilizza la Ground Truth generata al punto 4 per alimentare automaticamente il training set di Dedupe. Il modello è stato addestrato su un sottoinsieme della Ground Truth (circa 50 coppie) per apprendere i pesi relativi dei campi.

Vantaggio: Il modello apprende autonomamente che, ad esempio, una discrepanza nel campo price è meno grave di una discrepanza nel campo model, assegnando pesi ottimali alle feature.

Record Linkage non campionario

Dedupe-B1 | P=0.12 R=0.57 F1=0.20 Train=8.80s Inf=134.556s

Dedupe-B2 | P=0.12 R=0.57 F1=0.20 Train=8.80s Inf=151.238s

Record Linkage campionario

Dedupe-B1 | P=0.15 R=0.70 F1=0.24 Train=14.10s Inf=234.584s

Dedupe-B2 | P=0.15 R=0.71 F1=0.24 Train=14.10s Inf=206.089s

Implementazione e Sfide Tecniche (Ditto in Colab)

L'implementazione di Ditto in ambiente Colab ha presentato significative sfide tecniche, risolte come segue:

1. Parsing e Formattazione: Il dataset conteneva tabulazioni extra che corrompevano il parser di Ditto. È stato necessario applicare una patch al file dataset.py per rendere il caricamento più robusto.
2. Incompatibilità Librerie (NVIDIA Apex): La dipendenza da NVIDIA Apex (spesso problematica su Colab) è stata rimossa, adattando il codice per utilizzare l'ottimizzazione nativa di PyTorch o precisione standard.
3. Gestione Memoria GPU (CUDA OOM): Durante il fine-tuning di DistilBERT, si sono verificati errori di Out Of Memory. Il problema è stato risolto riducendo il batch_size da 32 a 16.
4. Checkpoint Loading: Lo script di inferenza originale falliva nel caricare i pesi dal checkpoint salvato (che conteneva l'intero stato dell'optimizer). È stato modificato lo script matcher.py per estrarre correttamente solo il dizionario dei pesi del modello (model.load_state_dict).

Valutazione Sperimentale

La valutazione è stata condotta su un Test Set separato, garantendo che i modelli non venissero valutati sugli stessi dati usati per il training o la calibrazione.

Eperimento 1: Caricamento campionato Impatto del Blocking (B1 vs B2)

Pipeline	Precision	Recall	F1-Measure	Training Time	Inference Time
Ditto (B1/B2 indip.)*	0.973	0.881	0.9251	~10 min	~94.0s

Eperimento 2: Caricamento totale Impatto del Blocking (B1 vs B2)

Pipeline	Precision	Recall	F1-Measure	Training Time	Inference Time
Ditto (B1/B2 indip.)*	0.973	0.881	0.9251	~10 min	~94.0s

Analisi dei Risultati

Ditto (FAIR-DA4ER):

Grazie all'uso di un Language Model, Ditto è riuscito a capire il contesto (es. differenze negli allestimenti o nelle descrizioni) che sfuggono alle metriche di distanza classiche (Jaro-Winkler).

Sebbene richieda GPU per il training, il tempo di inferenza (~94 secondi) è accettabile considerata l'altissima qualità del risultato.

Conclusioni modello campionato

Il progetto ha dimostrato che, in scenari di integrazione dati complessi come quello automotive, dove mancano identificatori forti e i dati sono eterogenei (UGC vs strutturati), l'adozione di modelli basati su Deep Learning (Ditto), supportata da una strategia rigorosa di generazione della Ground Truth (Smart Filtering e Hard Negative Mining), ha permesso di raggiungere prestazioni di livello industriale ($F1 > 0.94$). Nonostante la maggiore complessità implementativa e computazionale, Ditto rappresenta l'unica soluzione viabile per garantire l'alta qualità del dato integrato in questo contesto.