



Trasite!

PROGRESSO IN DIREZIONE ALLA MIGRAZIONE



Fondatori di Trasite!



Mario Cosenza



Valentina Ferrentino



Giovanni Semioli

Indice della presentazione



1

Problema

4

Feature

2

Stato Dell'Arte

5

Risultati

3

Dataset

6

Demo

Problema in esame



L'Integrità del
Mercato Digitale

1

Sebbene il mercato degli affitti brevi si basi sulla reputazione delle recensioni online, questo ecosistema è costantemente minacciato dalla diffusione di **recensioni ingannevoli** generate per manipolare la percezione degli utenti.

Limiti dei Sistemi
di Valutazione
Tradizionali

2

Distinguere tra un'**opinione autentica e una falsa** è diventata una sfida complessa poiché le semplici valutazioni numeriche non sono più sufficienti; è necessario analizzare la **coerenza informativa** e le sfumature semantiche profonde dei testi.

Difficoltà di
Scelta per
l'Utente

3

Le recensioni false confondono chi deve prenotare; non serve più sapere solo quanto una struttura è apprezzata, ma è fondamentale capire **quanto quel giudizio sia effettivamente degno di fiducia**.

Stato dell'arte (1/2)



Le **recensioni online** sono dati complessi, caratterizzati da elevata **eterogeneità semantica** e dalla presenza di **contenuti manipolativi**.



Una quota significativa di recensioni **sospette** compromette l'affidabilità dei **sistemi di reputazione** e le decisioni degli utenti.



La ricerca si è spostata dai **punteggi aggregati** all'analisi dei **dati sottostanti**, includendo testo e **metadati contestuali** (tempo, frequenza, distribuzione).

Stato dell'arte (2/2)



L'analisi combina **pre-elaborazione**, **Machine Learning** e studio dei **metadati temporali e spaziali** per individuare anomalie e rischio reputazionale.



La **visualizzazione dei risultati** (mappe, strumenti interattivi) è essenziale per migliorare **interpretabilità** e accessibilità.



Emerge la necessità di **soluzioni integrate** che uniscano **analisi testuale**, **valutazione automatica dell'affidabilità** e **visualizzazione dei dati** a supporto di decisioni più consapevoli.

Cos'è Trasite!



Trasite! è un **sistema integrato** nato per analizzare l'**affidabilità dei feedback** nel mercato degli affitti brevi. Unisce linguistica computazionale, statistica multivariata e **Deep Learning** per identificare con precisione le **recensioni ingannevoli**. L'obiettivo finale è garantire **sicurezza e consapevolezza**, rivelando quanto un giudizio sia realmente degno di fiducia.



Servizi che Trasite! offre



1

Intelligenza Artificiale

Implementazione di un motore di analisi che esegue l'estrazione di oltre 90 feature linguistiche. Il sistema valuta indicatori di **bot-likeness**, **sentiment** e **strutture sintattiche**, sfruttando modelli avanzati come **DeBERTa-v3** per catturare le semantiche.

2

REST API

Tramite **FastAPI**, che funge da interfaccia tra i modelli di analisi e l'utente finale, garantendo un accesso strutturato ai dati memorizzati in **MongoDB**.

3

Frontend

Sviluppo di un'interfaccia intuitiva in **Flutter** che integra una **visualizzazione geospaziale interattiva**.

Dataset di Benchmark



Yelp NYC



Un dataset che rispecchia le **dinamiche reali** delle piattaforme web, dove le recensioni false rappresentano una minoranza.

È servito come banco di prova per testare la **robustezza dei modelli** e l'efficacia delle tecniche di **campionamento**.

Deceptive



Definito come il "**gold standard**" della letteratura, contiene 1.600 recensioni di hotel perfettamente **bilanciate** (800 autentiche e 800 fraudolente).

È stato utilizzato per isolare i **marker linguistici puri**, eliminando ogni distorsione statistica durante l'analisi iniziale.

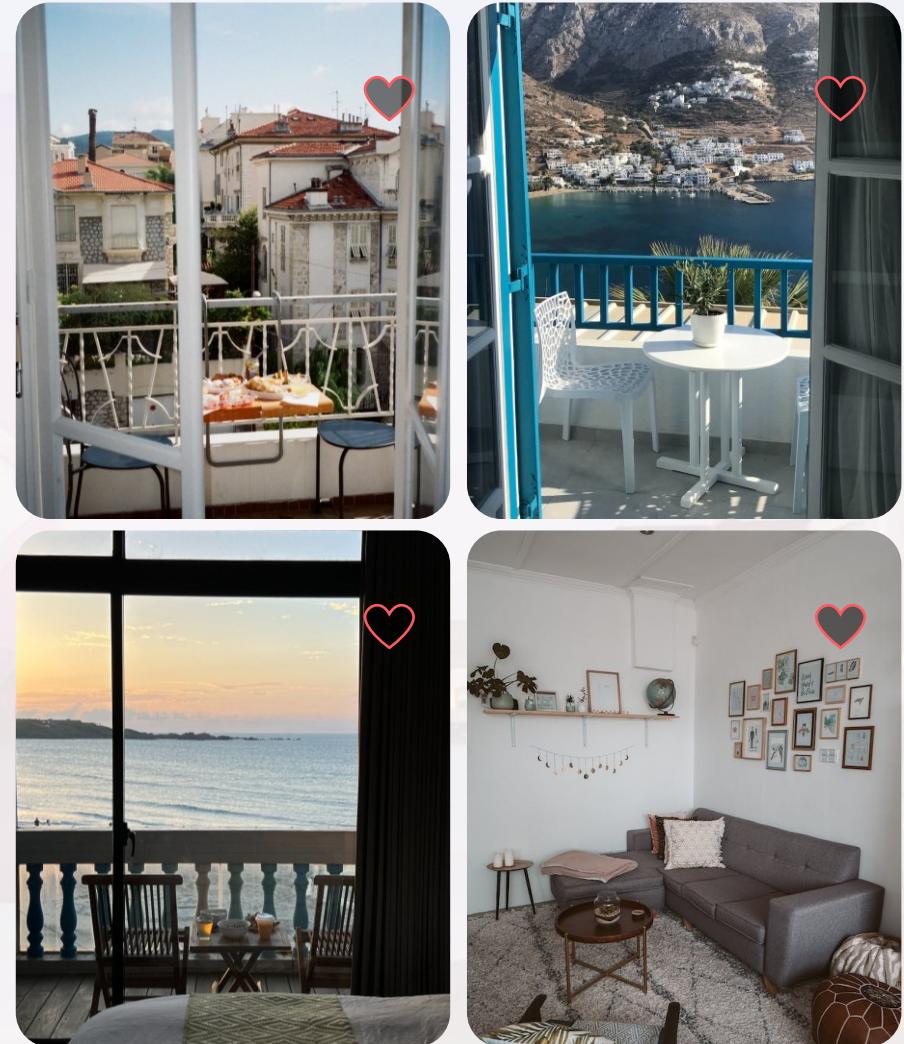
Applicazione reale: Inside AirBnB



Abbiamo scelto [Inside Airbnb](#) come fonte operativa per applicare i nostri modelli al mercato reale degli [affitti brevi a Napoli](#).

Per garantire la massima precisione semantica, l'analisi si è focalizzata esclusivamente su [recensioni in lingua inglese](#), elaborando un database massiccio di ben [212.153](#).

L'uso di metadati come Listing ID, Coordinate (Lat/Long) e Rating permette di proiettare i risultati su una [mappa interattiva](#), rendendo il rischio reputazionale immediatamente visibile all'utente.

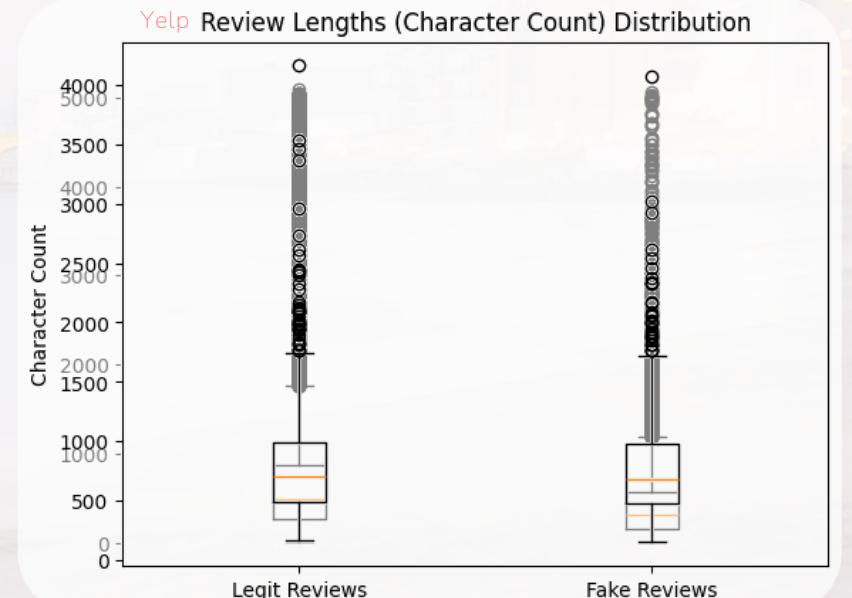
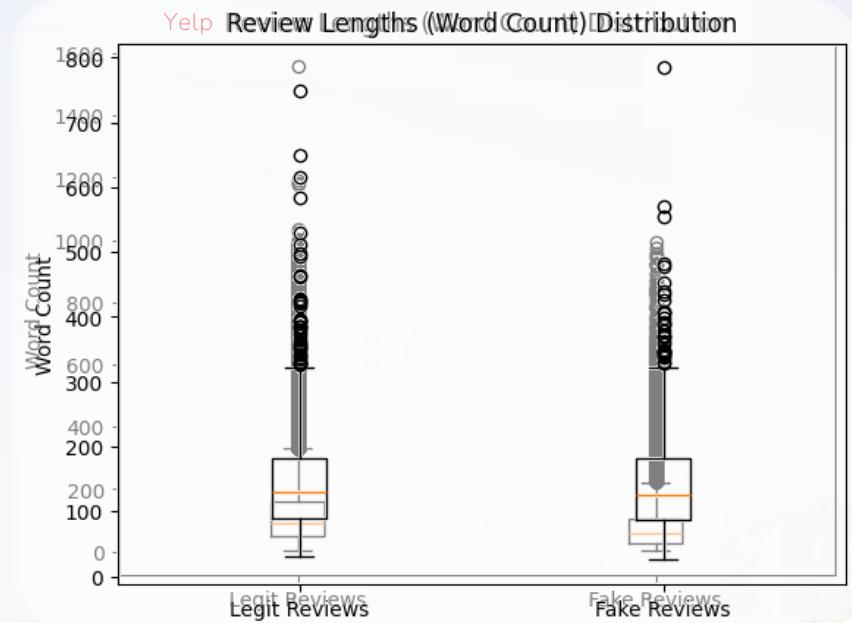


Feature linguistiche (1/2)



In linea con il paper di riferimento, la prima famiglia di segnali analizzata riguarda le feature di “[Writing Style / Linguistic Cues](#)”:

- Lunghezza e densità (proxy di informatività): `char_len`, `word_count`, `avg_word_length`.
- Struttura e punteggiatura: conteggi e rapporti di `punct_*` (esclamativi/interrogativi), densità per frase e normalizzazioni per token.
- Feature di leggibilità (`readability`): indici di complessità (Flesch/Fog e affini) coerenti con l’idea del paper che la complessità percepita e la scorrevolezza possano cambiare tra review spontanee e review costruite.



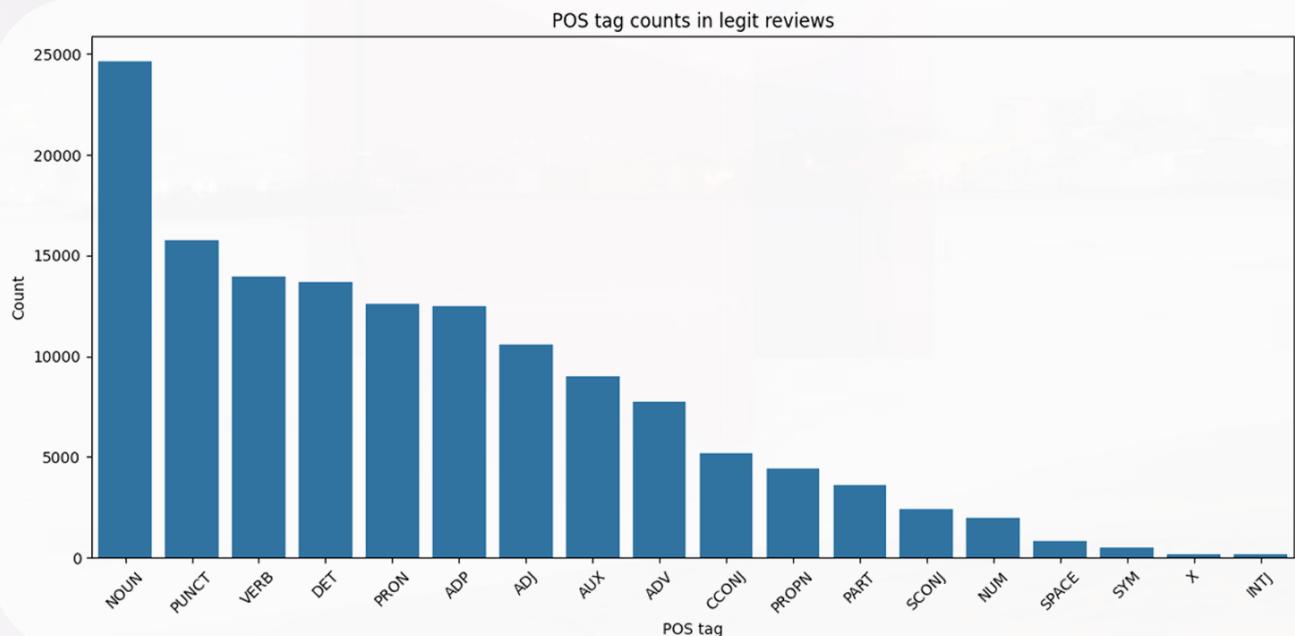
Feature linguistiche (2/2)



Le **feature linguistiche** sono state calcolate con una pipeline uniforme sui due dataset, per rendere confrontabili le distribuzioni tra **Fraud** e **Legit** senza introdurre **bias di preprocessing**.

Parsing linguistico con spaCy

Tokenizzazione e **analisi morfosintattica** (POS tagging) per ricavare, per ogni review, la sequenza di **tag POS** e le **frequenze delle principali categorie** (NOUN/VERB/ADJ/ADV, ecc.), successivamente aggregate per classe.



Risultati Analisi linguistica



Lunghezza



Yelp NY evidenzia un pattern robusto, le **review legit** risultano mediamente più **lunghe** delle **fake**.

Leggibilità



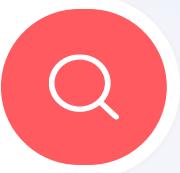
Le **metriche strutturali** mostrano spesso segnali più stabili rispetto a **feature lessicali** rare (es. money/assent).

Conclusioni



Su stile e profilo **morfosintattico** si osservano **scostamenti più sottili** ma coerenti, utili per caratterizzare i pattern linguistici delle due classi.

Analisi dei sentimenti



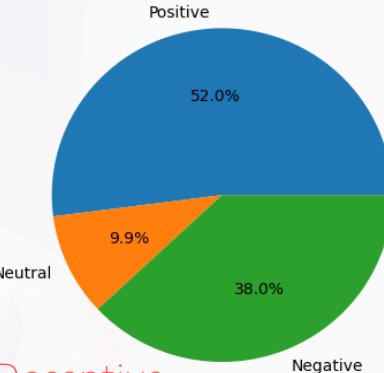
+8%



Misurare differenze di valenza emotiva tra fake e legit con un approccio modellistico.

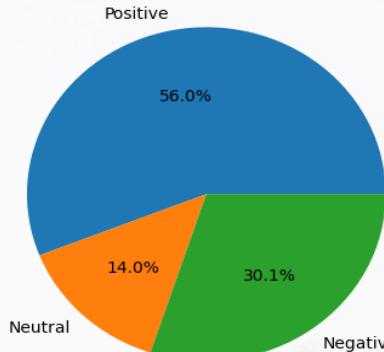
- negative/neutral/positive come distribuzione continua, non etichette discrete.
- subjectivity come proxy di “opinione vs descrizione”.
- spam/no_spam e bot/no_bot come segnali complementari di anomalia/innaturalità

Sentiment Distribution in Fake Reviews

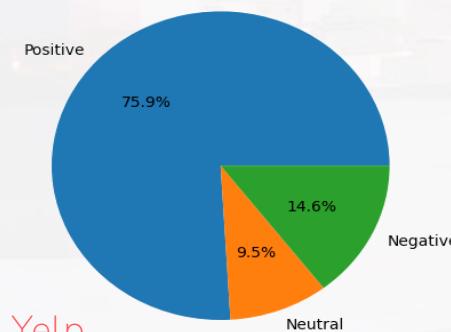


Deceptive

Sentiment Distribution in Legit Reviews

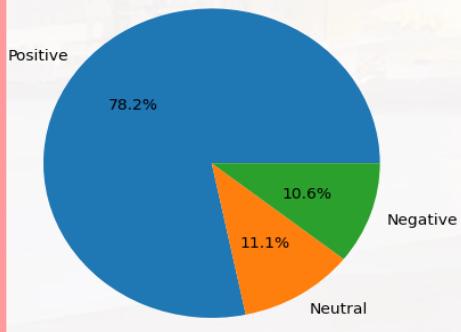


Sentiment Distribution in Fake Reviews

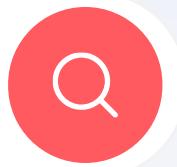


Yelp

Sentiment Distribution in Legit Reviews



Architettura Analisi Sentimenti



Moduli principali

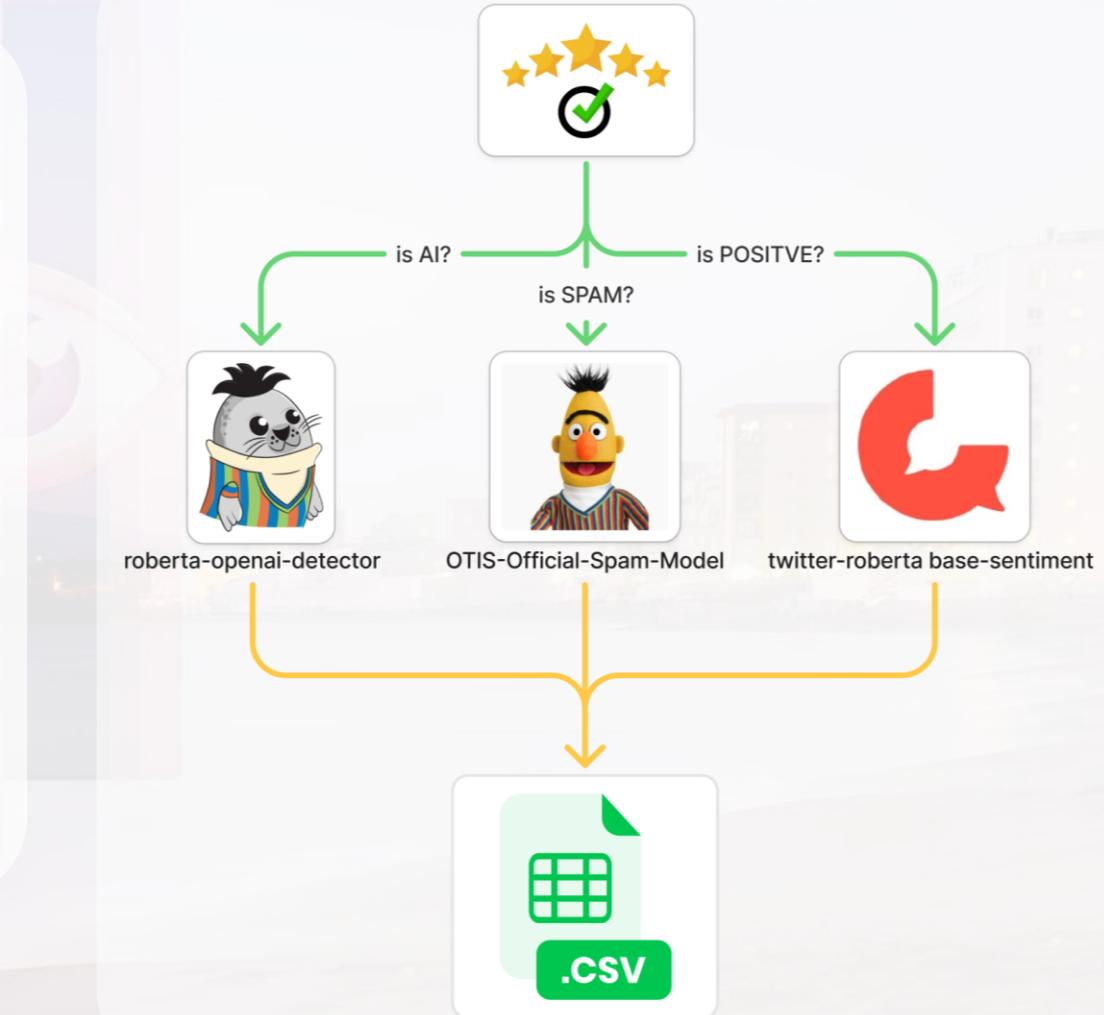
- *openai-community/roberta-base-openai-detector*
→ bot/no_bot
- *cardiffnlp/twitter-roberta-base-sentiment*
→ negative/neutral/positive
- *OTIS spam model* → spam/no_spam

Scelte tecniche

Truncation a 512 token, inferenza in batch.

Ottimizzazione batch

Ordinamento dei testi per lunghezza per migliorare efficienza del padding.



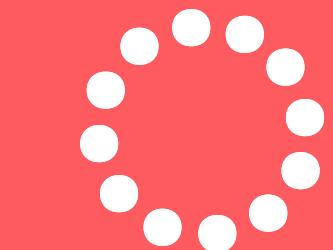
Risultati Analisi Sentimenti

Q



Trend Atteso

Differenze nella polarizzazione (fake più “spinte” su positivo o negativo, dataset-dependent)



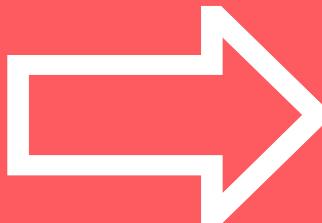
Subjectivity

Tende ad essere una feature debole/moderata: spesso distribuzioni sovrapposte, utile come segnale secondario.



Bot e Spam

Possono evidenziare cluster anomali (outlier e testi estremamente brevi/templati)



Implicazione

Sentiment e subjectivity non risolvono da soli il problema allora serve training dedicato

PCA Analysis (1/3)



T

Tecnica

La Principal Component Analysis (PCA) è una tecnica statistica multivariata che permette di analizzare più variabili contemporaneamente.

Relazioni

Studia le [relazioni reciproche](#) tra le variabili per identificare schemi e correlazioni nascoste.

Riduzione

Il suo obiettivo principale è la [riduzione della dimensionalità](#), cioè semplificare i dati mantenendo l'informazione più rilevante.

S

Sintesi

In questo modo, è possibile descrivere dataset complessi usando un [numero inferiore di variabili](#) senza perdere informazioni significative.

PCA Analysis (2/3)



1

Trasformazione

La PCA trasforma le variabili originali in un nuovo insieme di **componenti principali**, ottenute come combinazioni lineari delle variabili iniziali.

2

Indipendenza

Le componenti principali sono **ortogonal**i, cioè statisticamente indipendenti tra loro.

3

Varianza

Vengono ordinate in base alla **varianza spiegata**, che misura quanta informazione ciascuna componente cattura dal dataset.

4

Informazione

Le prime componenti contengono la maggior parte delle informazioni, mentre le successive descrivono variazioni sempre meno rilevanti.

PCA Analysis (3/3)



5

Esplorazione

La PCA è utile per l'[esplorazione dei dati](#), aiutando a individuare strutture latenti e pattern nascosti.

6

Pre-elaborazione

Nella [pre-elaborazione](#), riduce la ridondanza e il rumore informativo, rendendo i dati più puliti e interpretabili.

7

Visualizzazione

Per la [visualizzazione](#), consente di rappresentare dataset complessi in spazi a bassa dimensionalità.

8

Efficienza

Migliora l'[efficienza delle analisi successive](#), facilitando la comprensione e l'interpretazione dei dati.

Modelli Addestrati PCA



1. Obiettivo e razionale

Affiancare ai modelli testuali un approccio classico basato su **feature numeriche**, riducendo dimensionalità e ridondanza e definendo un **baseline quantitativo** indipendente dal testo.

3. Modelli addestrati

Addestramento di una **Logistic Regression** e di un **MLPClassifier** nello spazio latente, integrati con PowerTransformer per stabilizzare le distribuzioni.

2. Preprocessing e PCA

Costruzione di un dataset ML-ready tramite selezione delle sole feature numeriche, imputazione con **mediana**, **standardizzazione** e applicazione della PCA (95% di **varianza spiegata**), ottenendo componenti **PC1...PC175**.

4. Sbilanciamento e ottimizzazione

Gestione dello **sbilanciamento** con strategie original / undersample / oversample e ottimizzazione degli iperparametri tramite **RandomizedSearchCV**, usando **F1 macro** come metrica.

Risultati Analisi PCA (1/2)



L'integrazione della **Principal Component Analysis** è finalizzata alla costruzione di un dataset **ML-ready**, in grado di bilanciare ricchezza informativa e sintesi computazionale.

Il processo inizia con la selezione delle sole **feature numeriche**, escludendo identificativi e testi grezzi, così da concentrare l'analisi sulla struttura quantitativa dei dati.

Prima della decomposizione, viene svolta una fase di **data preparation**, in cui i valori mancanti sono imputati tramite **mediana** e le feature vengono successivamente **standardizzate**.

La **standardizzazione** rappresenta un requisito essenziale per la PCA, poiché garantisce che tutte le variabili contribuiscano alla varianza totale su una scala comune.

Risultati Analisi PCA (2/2)

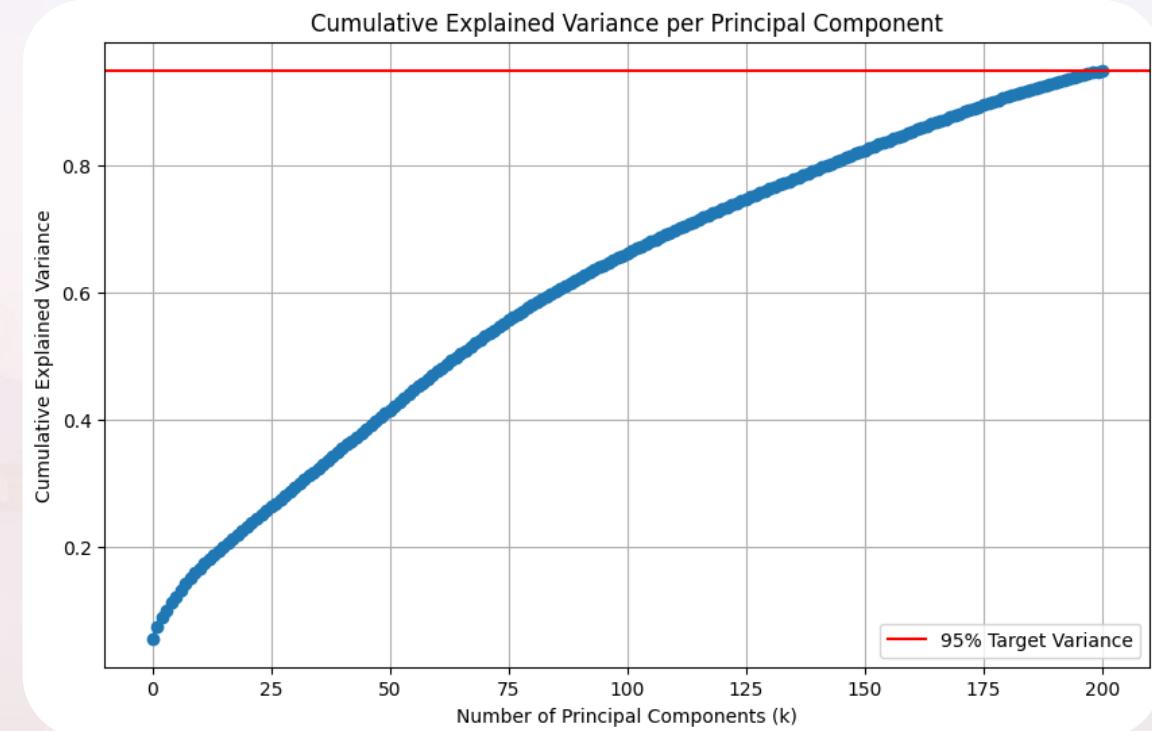


L'informazione risulta **distribuita su molte variabili**, senza pochi indicatori dominanti.

È stata adottata una soglia del **95% di varianza spiegata** come compromesso tra sintesi e completezza.

Tale soglia richiede il mantenimento di circa **175–200 componenti principali**, riducendo la dimensionalità senza perdita significativa di informazione.

Lo spazio latente di componenti **ortogonali** riduce la **complessità computazionale**, elimina la **multicollinearità** e migliora la stabilità dei modelli di Machine Learning.

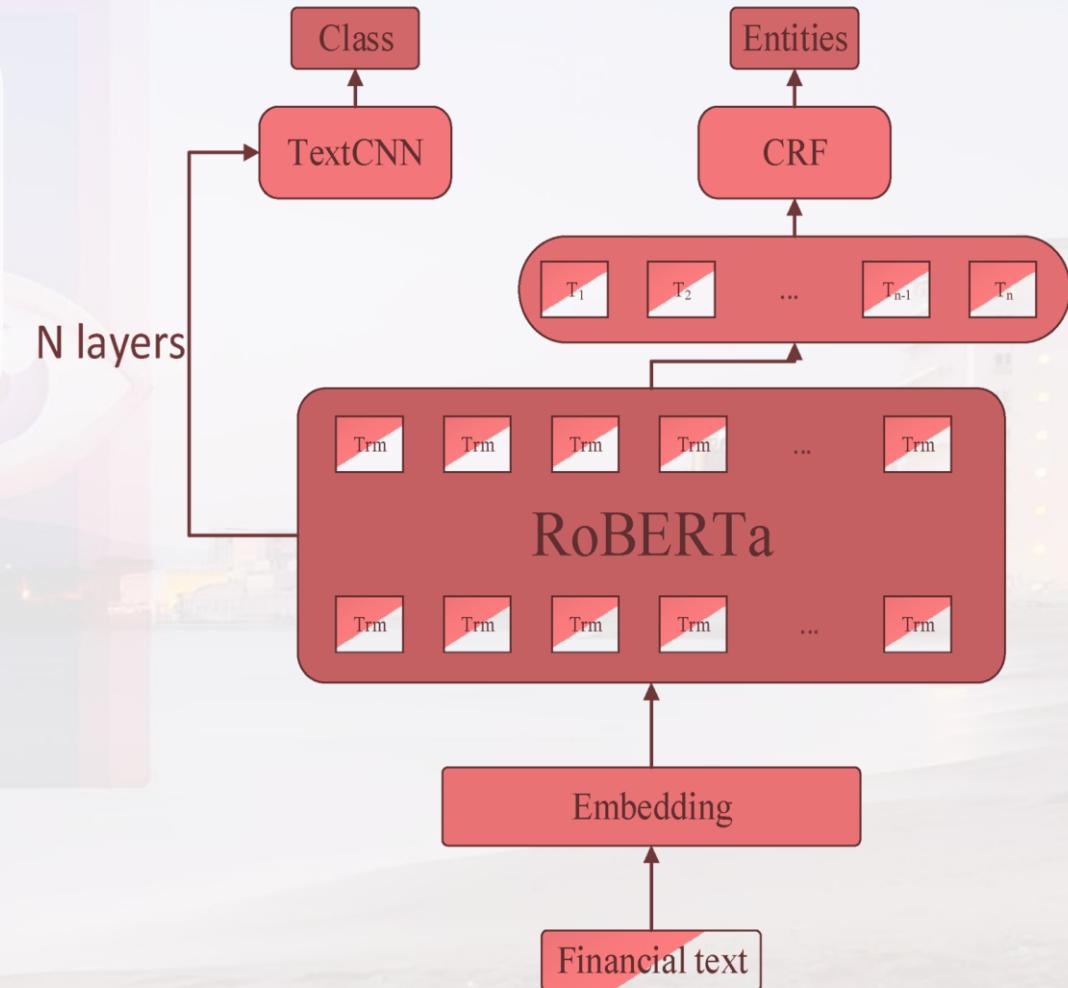
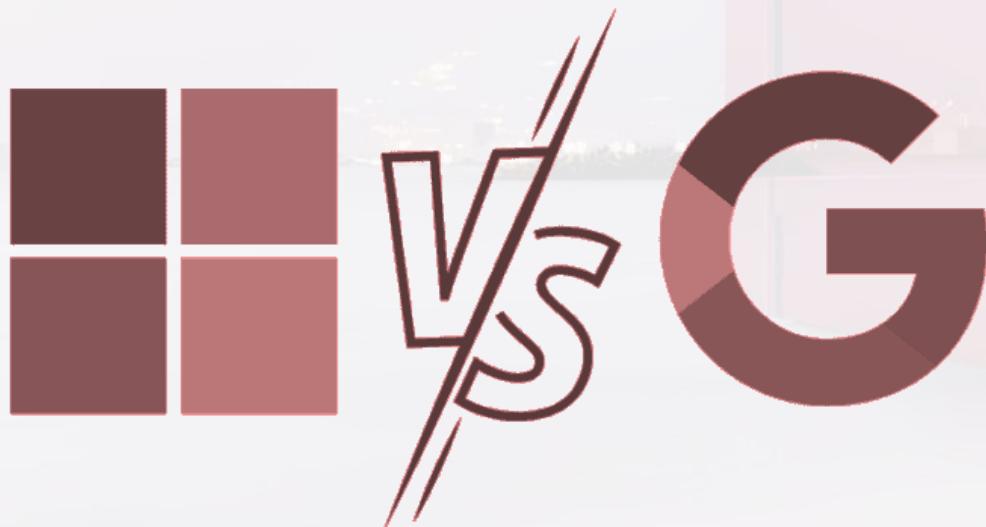


Architettura Transformer (1/2)



DistilRoBERTa

Versione compressa di RoBERTa: mantiene la stessa logica encoder-only Transformer ma con meno strati e parametri, quindi è più **veloce e leggera**, può perdere capacità di modellare segnali rari o dipendenze più sottili nel testo.

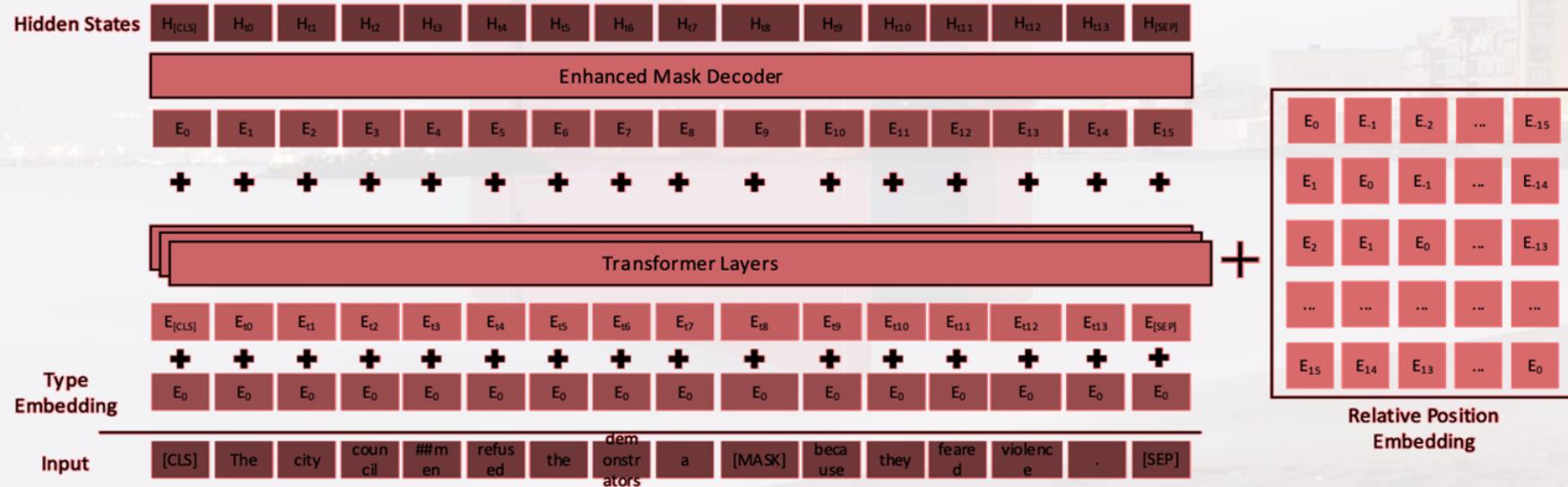


Architettura Transformer (2/2)



DeBERTa-v3-xsmall

Introduce una rappresentazione più ricca grazie alla separazione tra **content**, **embeddings** e position embeddings e a un meccanismo di attenzione “**disentangled**”. Questo design, tende a migliorare la qualità delle rappresentazioni contestuali e la sensibilità a indizi linguistici più fini, che in un task come **fake review detection** possono essere decisivi per differenziare testi “plausibili” ma costruiti.



Feature testuali



Nel training dei modelli **Transformer** la feature principale è il testo stesso: non si costruiscono manualmente vettori linguistici, ma si lascia al **backbone** il compito di apprendere pattern discriminanti tramite rappresentazioni contestuali.

La pipeline parte da review pulite e coerenti con la **rimozione di record incompleti** successivamente la tokenizzazione trasforma ogni review in sequenze di token subword, con **troncamento** a 512.

In output, oltre alla classe predetta, viene conservato un valore di **confidence** (score del classificatore) che è utile per analisi successive (es. ordinamento, scelta di soglie operative), soprattutto quando il sistema verrà applicato a un dataset “reale” come **Inside Airbnb Napoli**.

Quanno 'o modello vede
50 recensioni “Perfetto, vista mare,
tutto fantastico” a Napoli...



Problema del Sampling



Baseline

1

Accuracy alta, ma il modello predice quasi sempre Legit: Recall Fraud molto basso (molte fake non intercettate).

Undersampling

2

Riduce Legit nel train (es. 2:1, 1:1): aumenta Recall Fraud, ma può perdere informazione sulla classe maggioritaria.

Oversampling

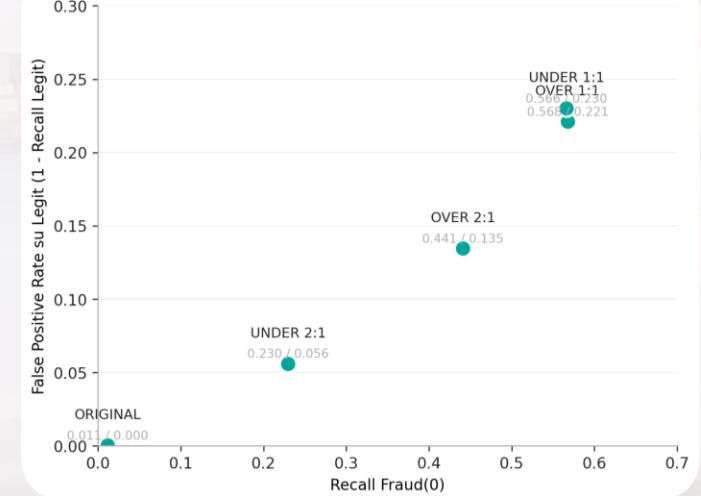
3

Aumenta Fraud nel train cresce la sensibilità alle fake, ma cresce il rischio di falsi positivi e overfitting.

SLIDE 10 — Imbalance classi (Yelp test set)
64.317
(89.8%)



SLIDE 10 — Trade-off Sampling: più Recall Fraud \leftrightarrow più FP



Risultati Deberta (1/5)



FIGURA R1 — DeBERTa: Recall per classe vs strategia di sampling

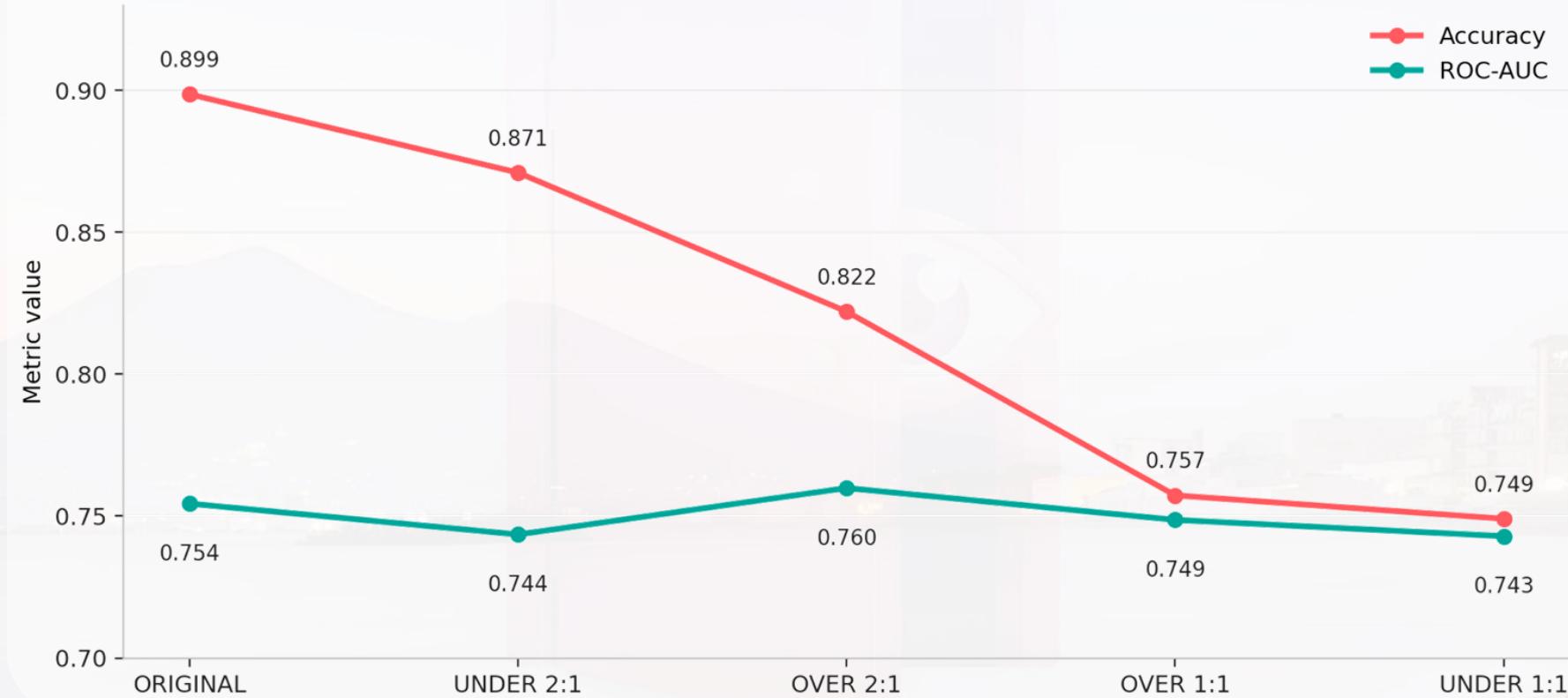


Recall Fraud cresce da ~0.01 (original) a valori > 0.56 (1:1), ma cala l'accuracy cambiando soprattutto l'operating point (FP/FN).

Risultati Deberta (2/5)



FIGURA R2 — DeBERTa: Accuracy e ROC-AUC vs strategia di sampling



DeBERTa mostra ROC-AUC relativamente stabile tra configurazioni mentre l'accuratezza sulla classe maggioritaria peggiora se vengono applicate tecniche di over/under sampling

Risultati Deberta (3/5)



Le **matrici di confusione** evidenziano lo spostamento da “quasi tutto Legit” a maggiore sensibilità Fraud. L’incremento di recall Fraud implica **aumento di falsi positivi** è necessario quindi **scegliere il modello in base al contesto applicativo**.

FIGURA R3 — DeBERTa: Confusion matrix normalizzata (ORIGINAL)

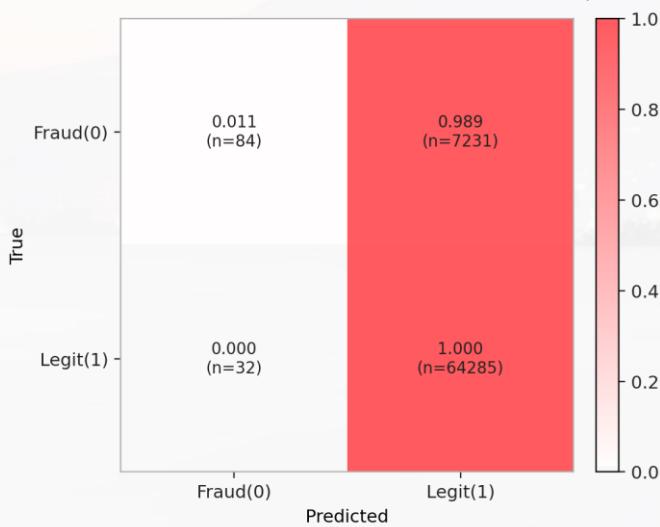


FIGURA R3 — DeBERTa: Confusion matrix normalizzata (OVER 1:1)

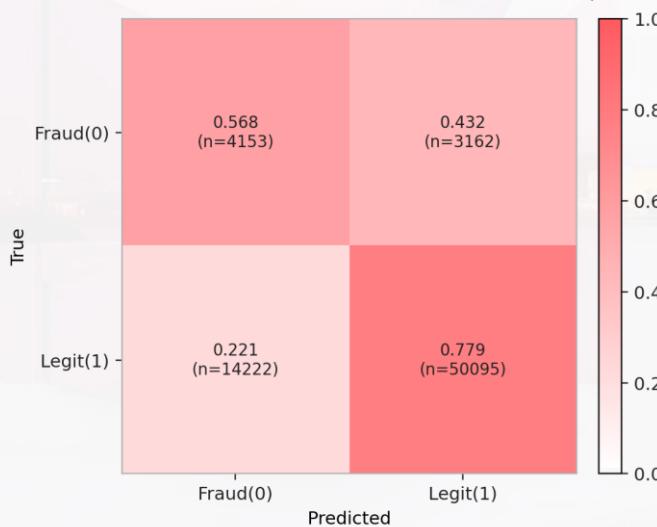
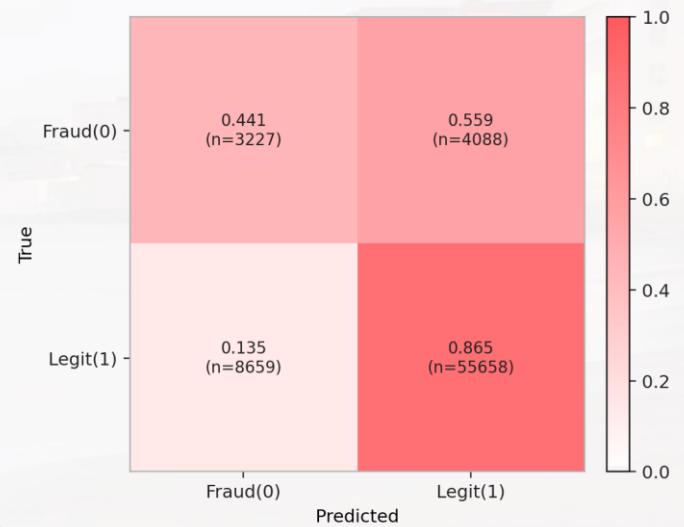


FIGURA R3 — DeBERTa: Confusion matrix normalizzata (OVER 2:1)



Risultati Deberta (4/5)



FIGURA F2 — F1 sulla classe Fraud(0) vs strategia di sampling



F1 Fake: utile per sintetizzare il trade-off; da leggere insieme a F1 Legit per evitare “ottimismo” sulla classe maggioritaria.

Risultati Deberta (5/5)



FIGURA F3 — F1 sulla classe Legit(1) vs strategia di sampling



F1 Legit: utile per sintetizzare il trade-off; da leggere insieme a F1 Fake per evitare “ottimismo” sulla classe maggioritaria.

MongoDB



Ruolo nel Sistema

Database NoSQL orientato ai documenti, utilizzato per l'archiviazione dei risultati della pipeline di IA e dei metadati.

Vantaggi

Flessibilità: Gestisce dati semi-strutturati (punteggi di affidabilità, coordinate geografiche, testi) senza schemi rigidi.

Eterogeneità: Adatto alla variabilità dei dati provenienti da fonti come Inside Airbnb.

Funzione nel sistema

Supporta la gestione dinamica delle informazioni che devono essere visualizzate sulla mappa del frontend.





Ruolo nel Sistema

Framework Python per lo sviluppo di REST API ad alte prestazioni.

Vantaggi

Supporto nativo alle **operazioni asincrone**.

Validazione automatica dei dati.

Integrazione fluida con l'ecosistema Python (PyTorch, Hugging Face).

Funzione nel sistema

Agisce da ponte tra il database MongoDB e il frontend in Flutter, esponendo i livelli di affidabilità calcolati e i metadati geografici in modo sicuro e strutturato.



Flutter



L'interfaccia è sviluppata con il framework [Flutter](#) per garantire una navigazione fluida e intuitiva.

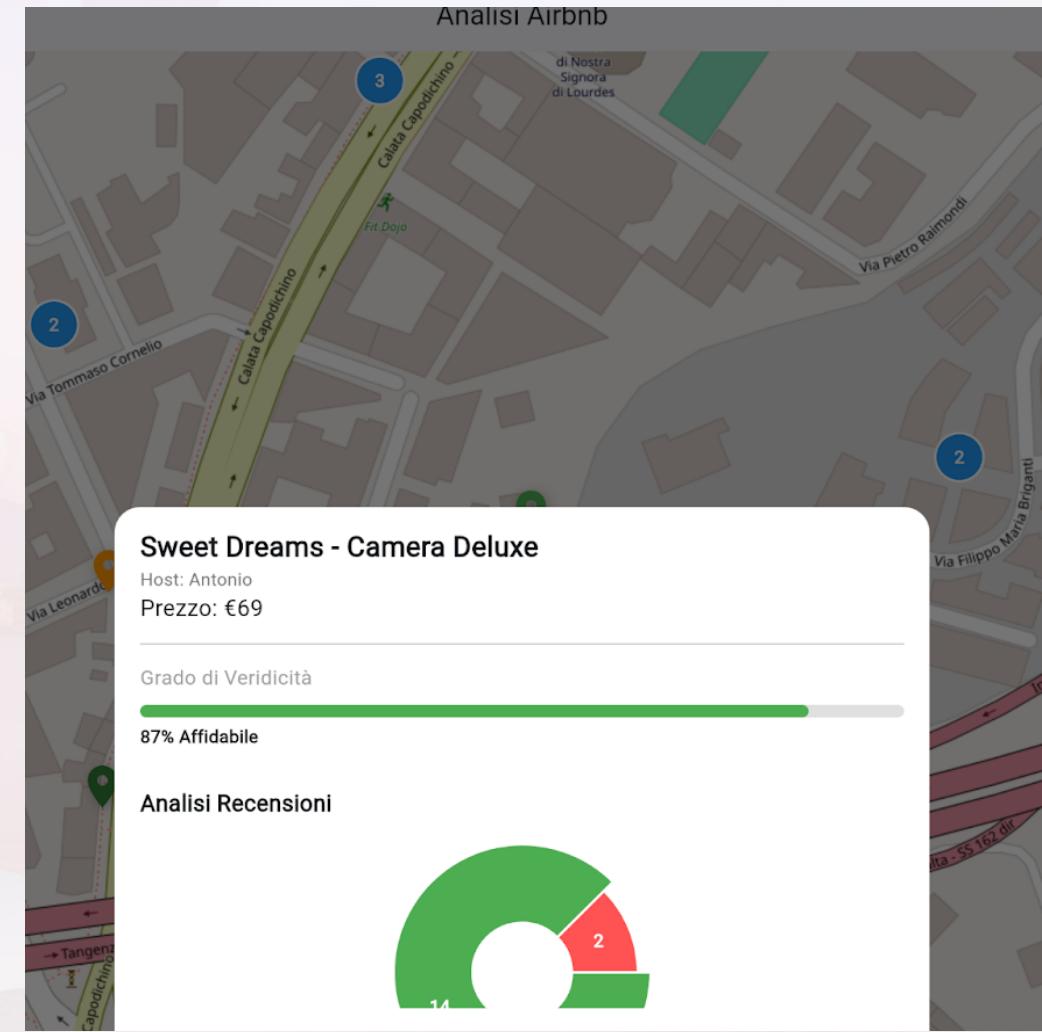
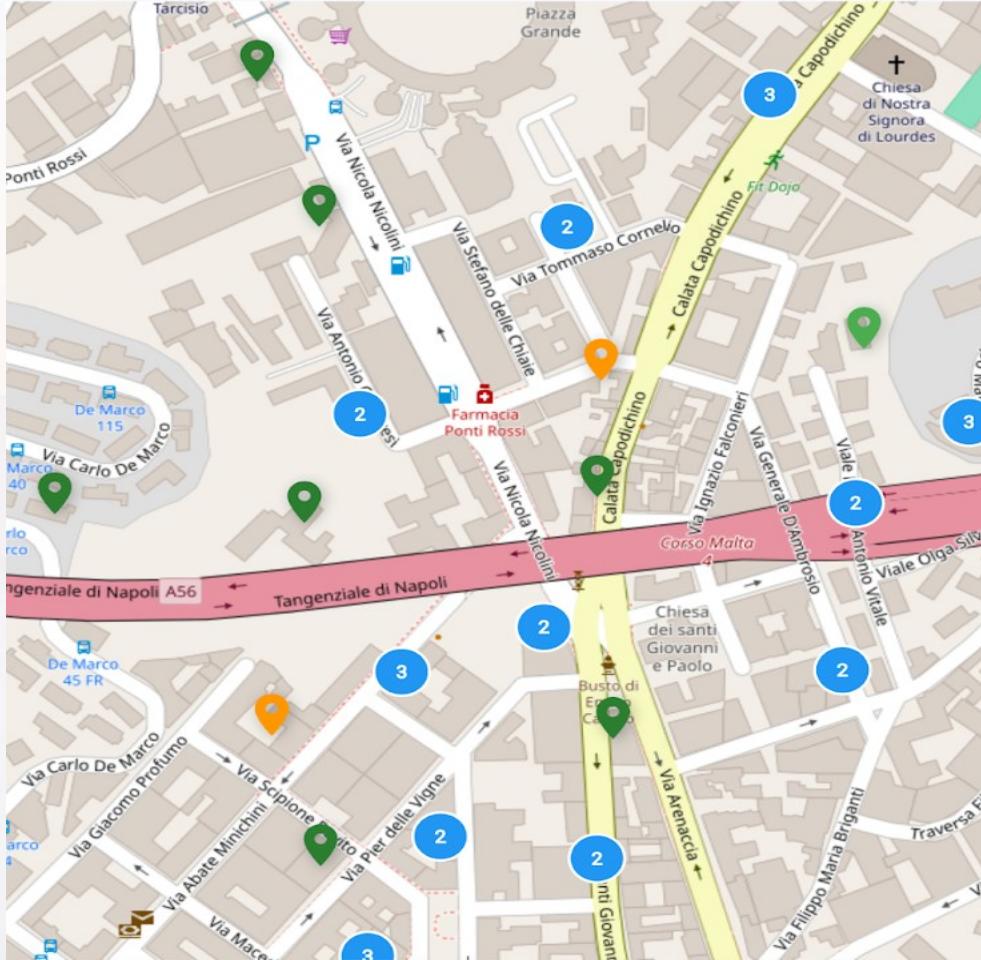
Il sistema comunica in modo asincrono con il backend via [FastAPI](#), trasformando analisi complesse in informazioni accessibili



Demo



Analisi Airbnb



Insegnamenti appresi



1

Abbiamo imparato che per trovare le recensioni false non basta un solo strumento, ma serve unire **statistica, intelligenza artificiale e mappe geografiche**. I modelli moderni come **DeBERTa** sono i più bravi a capire il significato delle parole.

2

Abbiamo capito che la sfida tecnica più grande è lo **sbilanciamento dei dati**. Se ci sono troppe recensioni vere, l'AI impara male; per questo è fondamentale usare tecniche per "aiutarla" a riconoscere quelle false.

3

Abbiamo appreso che un'analisi dei dati è utile solo se chiunque può capirla. Usare una **mappa interattiva** trasforma calcoli molto complessi in un aiuto concreto e immediato per l'utente.



Grazie!



https://github.com/EugenioRusso/IDA_Trasite

