

TRASITE!



UNIVERSITÀ DEGLI STUDI DI SALERNO

CORSO: INTERNET DATA ANALYSIS

2025/2026

DOCENTE:

DELFINA MALANDRINO

GRUPPO:

MARIO COSENZA

VALENTINA FERRENTINO

GIOVANNI SEMIOLI

INDICE

INTRODUZIONE.....	3
STATO DELL'ARTE.....	5
DATASET INDIVIDUATI.....	6
FEATURE DATASET.....	8
FEATURE LINGUISTICHE ESTRATTE DAL DATASET.....	9
ELENCO FEATURE ANALIZZATE.....	10
ANALISI FEATURE LINGUISTICHE.....	15
RISULTATI ANALISI FEATURE LINGUISTICHE.....	17
PCA ANALYSIS.....	25
MODELLI ADDESTRATI SU PCA.....	28
ARCHITETTURA MODELLI FAKE REVIEW DETECTION.....	31
FEATURE PER IL TRAINING DI DEBERTA.....	31
PROBLEMA DEL SAMPLING.....	32
RISULTATI TRAINING DEBERTA.....	33
DESIGN DI SISTEMA.....	39
MONGODB.....	39
FASTAPI.....	40
FRONTEND.....	40
DEMO FRONTEND.....	41
CONCLUSIONI.....	43

INTRODUZIONE

Il sistema di reputazione basato sulle recensioni online rappresenta oggi il pilastro fondamentale su cui poggia il mercato degli affitti brevi.

Tuttavia, l'integrità di questo ecosistema è costantemente minacciata dal fenomeno della **Deception Opinion Spam**: la diffusione di recensioni ingannevoli, generate artificialmente o finalizzate a manipolare fraudolentemente la percezione degli utenti. In questo contesto, distinguere tra un'opinione autentica e una "fake review" è diventata una sfida complessa che richiede strumenti di analisi che vadano oltre la semplice valutazione numerica.

Il progetto **Trasite!** si propone di affrontare questa problematica sviluppando una pipeline di analisi dei dati end-to-end, capace di valutare l'affidabilità dei feedback nel settore dell'ospitalità attraverso un approccio multidisciplinare che combina linguistica computazionale, statistica multivariata e Deep Learning.

L'**architettura del progetto** è stata strutturata per affrontare il problema attraverso tre fasi analitiche distinte e integrate:

1. Ingegnerizzazione delle Feature e Analisi Linguistica:

Il cuore del sistema risiede nell'estrazione di oltre 90 feature descrittive. Partendo dal testo grezzo, vengono isolati indicatori di **bot-likeness**, polarità del **sentiment**, soggettività e metriche strutturali.

Questo approccio permette di mappare lo "stile" della recensione, confrontandolo con i pattern psicometrici riportati come ricorrenti nel linguaggio ingannevole evidenziati dalla letteratura scientifica.

2. Ottimizzazione e Classificazione:

Data l'elevata dimensionalità dei dati linguistici, il progetto utilizza la **Principal Component Analysis (PCA)** per ridurre il rumore e la ridondanza, permettendo l'addestramento di modelli di Machine Learning (come MLP e Logistic Regression) su componenti ottimizzate.

Parallelamente, il sistema sfrutta le potenzialità dei Transformer, in particolare **DeBERTa-v3**, per catturare le sfumature semantiche più profonde, affrontando esplicitamente il problema del campionamento (sampling) per gestire lo sbilanciamento tra recensioni autentiche e fraudolente.

3. Visualizzazione Geospaziale e Fruibilità:

L'analisi non resta confinata all'ambiente di calcolo, ma viene resa operativa attraverso un sistema integrato che vede **MongoDB** come base dati, **FastAPI** come motore di comunicazione e un frontend in **Flutter**.

Quest'ultimo permette di proiettare i livelli di affidabilità calcolati su una mappa interattiva, offrendo all'utente una rappresentazione spaziale immediata del rischio reputazionale basata su coordinate geografiche reali (dati Inside Airbnb).

L'obiettivo finale è fornire un supporto decisionale trasparente, capace di evidenziare non solo "quanto" una struttura è apprezzata, ma "quanto" quel giudizio sia degno di fiducia. Attraverso l'integrazione tra analisi testuale avanzata e visualizzazione cartografica, il progetto mira a restituire sicurezza e consapevolezza agli utenti che navigano nel mercato digitale degli affitti brevi.

STATO DELL'ARTE

La letteratura scientifica recente riconosce le **recensioni online** come una forma di dato digitale complessa, caratterizzata da un'elevata eterogeneità semantica e da una crescente difficoltà nel distinguere **contenuti autentici** da contributi manipolativi, come evidenziato nella rassegna *Recent state-of-the-art of fake review detection*.

Studi consolidati mostrano come una percentuale non trascurabile delle recensioni pubblicate sulle principali piattaforme di e-commerce e di servizi turistici presenti caratteristiche sospette, tali da compromettere l'**affidabilità dei sistemi di reputazione** e, di conseguenza, i processi decisionali degli utenti, fenomeno approfondito anche in *High performance fake review detection using pretrained DeBERTa optimized with Monarch Butterfly paradigm*.

In questo contesto, l'attenzione della ricerca si è progressivamente spostata dall'analisi aggregata dei punteggi verso l'esame dei **dati sottostanti**, includendo il testo delle recensioni e i **metadati associati**, come sintetizzato nello studio di stato dell'arte sulla fake review detection. Un elemento centrale emerso dalla letteratura riguarda l'importanza delle fasi di **pre-elaborazione dei dati online**, considerate condizioni necessarie per ridurre il rumore informativo e rendere confrontabili recensioni provenienti da contesti differenti.

A partire da dati opportunamente strutturati, numerosi contributi propongono l'impiego di tecniche automatiche per individuare pattern ricorrenti e anomalie potenzialmente indicative di recensioni non autentiche, trattando il problema come una forma di analisi della **coerenza informativa**.

In tale ambito, i metodi di **Machine Learning** vengono utilizzati come strumenti di supporto all'analisi dei dati, sia attraverso approcci supervisionati sia non supervisionati, con l'obiettivo di evidenziare discrepanze tra il contenuto linguistico della recensione e il punteggio assegnato, aspetto approfondito anche in *Decoding deception in the online marketplace: enhancing fake review detection with psycholinguistics and transformer models*.

Parallelamente, diversi studi sottolineano il valore informativo dei **metadati contestuali**, quali la distribuzione temporale delle recensioni e la loro frequenza di pubblicazione, che consentono di osservare il fenomeno su scala spaziale e di individuare concentrazioni anomale di rischio reputazionale. Le ricerche più recenti evidenziano inoltre come la **visualizzazione dei risultati**, in particolare mediante mappe e strumenti interattivi, rappresenti un passaggio fondamentale per rendere l'analisi dei dati più accessibile e interpretabile.

In questo quadro emerge l'esigenza di soluzioni che integrino analisi testuale, valutazione automatica dell'affidabilità e visualizzazione dei risultati, andando oltre la semplice classificazione delle recensioni e orientandosi verso un supporto informativo più consapevole per l'utente finale.

DATASET INDIVIDUATI

La robustezza di un sistema di *Fake Review Detection* dipende strettamente dalla qualità e dalla varietà dei dati utilizzati nella fase di addestramento. Per questo progetto, la strategia di reperimento dei dati è stata divisa in due macro-fasi: l'utilizzo di dataset certificati per il training supervisionato e l'acquisizione di dati reali per la fase operativa.

Per addestrare e confrontare i classificatori (Machine Learning classico su PCA e Transformer DeBERTa), utilizzando metriche standard come **Accuratezza**, **F1-Score**, **Recall** e **Precision**, è stato indispensabile utilizzare dataset che presentassero una classificazione binaria esplicita (**Legit vs Fake**), dove la distinzione tra recensione autentica e fraudolenta è assunta come ground truth operativa del dataset, pur riconoscendo che le etichette possono contenere rumore, soprattutto su dataset real-world. Abbiamo individuato due dataset complementari per caratteristiche e complessità, disponibili sulla piattaforma **Kaggle**:

- **Deceptive Opinion Spam Corpus:** Un dataset "gold standard" composto da 1.600 recensioni di hotel. La sua natura perfettamente bilanciata (800 *legit*, 800 *fake*) lo rende ideale per isolare i marker linguistici puri, senza le distorsioni tipiche degli sbilanciamenti statistici.
- **Yelp New York City (YelpNYC):** A differenza del precedente, questo dataset riflette le dinamiche reali delle piattaforme web. Presenta un forte sbilanciamento delle classi (la classe *fraud* è minoritaria), rappresentando il terreno di prova fondamentale per testare la robustezza dei modelli e l'efficacia delle tecniche di *resampling* e campionamento.

Una volta completato il **fine-tuning** dei Transformer e l'addestramento del classificatore finale (downstream classifier), il sistema Trasite! viene applicato a dati reali estratti dal contesto degli affitti brevi a **Napoli**.

Inside Airbnb:

Abbiamo selezionato questa fonte (<https://insideairbnb.com/get-the-data/>) per la ricchezza di metadati associati alle recensioni reali. I dati raccolti non contengono solo il testo, ma informazioni contestuali critiche come:

- **Listing ID:** Per collegare le recensioni a una struttura specifica.
- **Coordinate Geografiche (Lat/Long):** Essenziali per la visualizzazione su mappa nel frontend Flutter.
- **Rating Numerico:** Per l'analisi di coerenza tra il testo e il punteggio assegnato.

La scelta di questi specifici dataset non è stata casuale, ma strategicamente finalizzata alla successiva fase di **Feature Engineering**.

La varietà linguistica e strutturale di queste fonti permette alla pipeline di **Trasite!** di estrarre un set di variabili eterogenee, capaci di catturare sia le sottili anomalie psicométriche del linguaggio ingannevole, sia pattern statistici riportati come ricorrenti nei contenuti generati artificialmente. Questa base informativa costituisce il presupposto necessario per la definizione delle feature che verranno dettagliate nella sezione seguente.

FEATURE DATASET

La fase di **Feature Engineering** rappresenta il cuore analitico del progetto. Per trasformare il testo grezzo in dati interpretabili dai modelli, abbiamo estratto un set di **oltre 90 feature**, progettate per catturare segnali potenzialmente associati a contenuti non autentici. Le feature sono state raggruppate in cinque categorie macro-aree:

1. Indicatori di Anomalia e "Bot-likeness"

Il primo livello di analisi si è concentrato sulla stima di segnali di anomalia nella scrittura, potenzialmente compatibili con generazione artificiale o template. Attraverso l'uso di modelli pre-addestrati, abbiamo estratto indicatori di **bot-likeness** e **spam-likeness**.

L'idea alla base è che una recensione generata da un algoritmo o copiata massivamente da un template di spam può presentare regolarità statistiche diverse da un commento scritto di getto da un utente reale.

Queste probabilità forniscono un segnale preliminare per individuare contenuti che tendono a discostarsi dai pattern medi, pur con ampia sovrapposizione tra classi.

2. Dimensioni Affettive e Soggettività (Sentiment Analysis)

Ci siamo poi spostati sull'analisi del **Sentiment** e della **Soggettività**. Nella letteratura sulla *deception detection*, diversi studi riportano che, in media, contenuti ingannevoli possono mostrare una maggiore polarizzazione emotiva; tuttavia l'effetto è tipicamente debole e dipendente dal dominio.

Attraverso metriche che misurano la polarità e il grado di opinione personale, abbiamo cercato di capire se la classe delle *fake reviews* tendesse a mostrare una carica emotiva più instabile o meno ancorata a descrizioni oggettive.

3. Feature Strutturali

Un'ampia porzione del nostro dataset riguarda la **stilometria** e la **struttura sintattica**. Qui l'analisi si fa granulare: abbiamo contato i verbi, i pronomi e i segni di punteggiatura, normalizzando spesso questi valori "per frase". Questo ci permette di caratterizzare, ad esempio, la differenza tra una narrazione ricca di azioni e una descrizione vaga o eccessivamente enfatica. Abbiamo anche incluso "proxy" di categorie psicométriche come le menzioni di denaro e le parole di assenso, per verificare se la focalizzazione sul prezzo o l'uso di conferme enfatiche siano potenzialmente associati a un tentativo di persuasione non genuino.

Questo insieme eterogeneo di variabili, che spazia dalla probabilità di "essere un bot" alla lunghezza media delle sillabe, riduce il rischio che il sistema **Trasite!** sia un semplice lettore di parole chiave. Al contrario, è in grado di analizzare il "ritmo" e la "consistenza" di ogni feedback, fornendo ai modelli di Machine Learning una base informativa ricca e multidimensionale per produrre una stima di affidabilità.

FEATURE LINGUISTICHE ESTRATTE DAL DATASET

Nel progetto le **feature linguistiche** sono state selezionate con un duplice obiettivo:

- I. mantenere comparabilità concettuale con il paper "*Decoding deception in the online marketplace: enhancing fake review detection with psycholinguistics and transformer models*", che applica **LIWC su 93** categorie e poi riduce la ridondanza rimuovendo variabili altamente correlate;
- II. garantire riproducibilità su dataset **Kaggle** utilizzando modelli pre-addestrati (sentiment/spam/bot-likeness) e su analisi linguistica automatica (spaCy + textdescriptives), evitando categorie LIWC non replicabili senza dizionari proprietari (es. *analytic, clout, tone, dic, function*).

Ne consegue un set di feature che copre: segnali "esterni" (probabilità di testo "AI-like/bot-like" e "spam-like"), dimensioni affettive, misure strutturali di lunghezza/complessità, e descrittori stilistico-sintattici (punteggiatura, pronomi, verbi e tempi verbali, menzioni di denaro), con un'espansione sistematica in colonne multiple per POS tag, simboli di punteggiatura e tempi verbali.

ELENCO FEATURE ANALIZZATE

- **bot**: probabilità continua derivata dal detector [openai-community/roberta-base-openai-detector](#), interpretata come proxy di “testo non genuino / bot-like”. In analisi esplorativa è stata usata per verificare se le recensioni etichettate fake tendono a mostrare punteggi più alti rispetto alle *legit*, e per individuare outlier.
- **no_bot**: complemento probabilistico di **bot** (misura di “realness” nel detector). È utile perché rende immediata l’interpretazione come stima di genuinità e permette confronti simmetrici con altre probabilità.
- **spam**: probabilità continua ottenuta dal modello [Titeiiko/OTIS-Official-Spam-Model](#), trasformata in modo che valori alti indichino maggiore somiglianza a spam. È stata impiegata per valutare in che misura il concetto di ‘spam’ del modello sia coerente con la label fake del dataset.
- **no_spam**: complemento probabilistico di **spam**
- **negative**: componente di sentiment (RoBERTa sentiment), utilizzata come proxy quantitativo del contenuto negativo. È concettualmente confrontabile con il cluster LIWC “negative emotions” che nel paper risulta più alto nelle fake, pur essendo ottenuta con un approccio modellistico e non dizionario.
- **neutral**: componente continua di sentiment, utile come “massa residua” per distinguere testi polarizzati (molto positivi o molto negativi) da testi più descrittivi/oggettivi. Serve anche come controllo: differenze marcate su **neutral** possono indicare che la classe fake tende a essere più polarizzata.
- **positive**: componente continua di sentiment, proxy operativo della positività.
- **subjectivity**: punteggio di soggettività (TextBlob) che approssima quanto il testo sia opinionistico vs descrittivo. Questa feature è stata inclusa perché, a livello teorico, le recensioni ingannevoli possono enfatizzare giudizi e intensificatori; inoltre è una metrica semplice e interpretabile, complementare ai punteggi di sentiment.
- **rating**: valutazione numerica associata alla recensione. È una feature di contesto utile per verificare eventuali bias di campionamento (es. fake concentrate su rating estremi).
- **char_len**: lunghezza in caratteri della recensione. È un proxy diretto della LIWC *len* e viene impiegato sia come indicatore di “sforzo” testuale (real tendenzialmente più lunghe nel paper) sia come variabile di controllo per normalizzazioni e confronti tra dataset.

- **word_count**: numero di parole della recensione. È vicino alla LIWC *wc*, che nel paper viene rimossa per multicollinearità.
- **avg_word_length**: rapporto caratteri/parole (proxy di complessità lessicale, concettualmente collegabile a *sixltr*). È stata introdotta esplicitamente come feature aggiuntiva per catturare differenze di densità informativa senza ricorrere a dizionari.
- **unique_duplicated_fake_texts**: numero di testi distinti duplicati nella classe fake (diagnostica di copy/paste e template). Rilevante per valutare la qualità del dataset e il rischio che un classificatore impari pattern di duplicazione invece di indicatori linguistici generali.
- **unique_duplicated_legit_texts**: analogo del conteggio duplicati, ma sulla classe legit; serve a capire se il fenomeno è bilanciato o concentrato in una sola classe (che sarebbe una scorciatoia predittiva).
- **verb_count**: conteggio di verbi per recensione estratto via spaCy. È confrontabile con LIWC *verb* (che nel paper risulta più alto nelle fake) e, nel progetto, è stato usato sia in forma grezza sia normalizzata per frase per separare l'effetto lunghezza dall'effettiva “densità” verbale.
- **verb_ratio_sent**: densità media di verbi per frase (media su frasi). Questa normalizzazione è importante perché due testi con la stessa lunghezza possono avere struttura sintattica diversa; inoltre, la metrica è concettualmente vicina all'idea di “stile narrativo vs descrittivo”.
- **verb_tense_{TENSE}_count**: conteggio, per ciascun tempo verbale osservato (es. Past/Pres), estratto dalla morfologia spaCy. È un proxy operativo delle categorie di focus temporale LIWC (*focuspast*, *focuspresent*, *focusfuture*), che nel paper mostrano differenze tra fake e real.
- **verb_tense_{TENSE}_ratio_sent**: densità per frase dei tempi verbali; serve a controllare l'effetto “lunghezza” e a confrontare l'orientamento temporale tra classi e dataset.
- **pronoun_count**: conteggio pronomi per recensione. È concettualmente collegabile a LIWC *ipron* (impersonal pronouns) e, più in generale, alle famiglie *pronoun/ppron*, motivo per cui viene interpretata come feature utile ma potenzialmente ridondante in modelli multivariati.
- **pronoun_ratio_sent**: densità media di pronomi per frase. A parità di parole, un aumento della densità pronomiale può indicare maggiore vaghezza o minor specificità, ma nel progetto viene trattata come ipotesi da verificare empiricamente sui due dataset.

- **money_count**: conteggio menzioni di denaro (pattern su token numerici con simbolo “\$” o stringa “dollar”). È il proxy diretto della categoria LIWC *money* (nel paper più alta nelle fake) e viene usato per testare se anche nei dataset Kaggle le fake enfatizzano prezzo/valore economico.
- **money_ratio_sent**: densità media di menzioni di denaro per frase. È utile perché isola il “tema denaro” dalla semplice lunghezza della recensione e riduce l’impatto di poche menzioni concentrate in recensioni molto lunghe.
- **money_mentions_percentage**: percentuale di token “money” sul totale token (normalizzazione per token). Serve come metrika alternativa alla normalizzazione per frase e rende confrontabili recensioni con diversa segmentazione in frasi.
- **agreement_words_count**: conteggio di parole di assenso/accordo (lista lessicale). È un proxy della categoria LIWC *assent* (nel paper più alta nelle fake) e nel progetto è stato analizzato come possibile indicatore di stile persuasivo (“sì”, “assolutamente”, ecc.), pur sapendo che l’effetto può essere dominio-dipendente.
- **punctuation_marks_per_review**: numero totale di segni di punteggiatura per recensione. È confrontabile con LIWC *all_punc* e serve a verificare se le fake presentano “enfasi” stilistica come suggerito dal paper.
- **punct_{SYM}_count**: conteggio per recensione di ciascun simbolo di punteggiatura osservato (es. “.”, “!”, “;”, “?”). Questa granularità permette un confronto più diretto con LIWC *period* ed *exclam*, che nel paper risultano statisticamente più alti nelle fake.
- **punct_{SYM}_ratio_sent**: densità media per frase del simbolo di punteggiatura. È particolarmente utile per separare l’uso “strutturale” (virgolette in frasi complesse) dall’uso “enfatico” (esclamativi ripetuti).
- **pos_{TAG}_count**: conteggio per recensione di ciascun POS tag spaCy (NOUN, VERB, ADJ, ADV, PRON, PROPN, ecc.).
- **pos_{TAG}_ratio_sent**: densità media per frase dei POS tag. La normalizzazione per frase riduce il bias dovuto alla lunghezza e rende comparabili recensioni brevi con recensioni articolate, mantenendo interpretabilità linguistica.
- **token_length_mean**: lunghezza media dei token (caratteri per token), indicatore di complessità lessicale generale; è concettualmente affine al rationale di *sixltr* (parole lunghe) ma in forma continua.

- **token_length_median**: mediana della lunghezza token, più robusta agli outlier (es. un token lunghissimo dovuto a URL o stringhe errate).
- **token_length_std**: deviazione standard della lunghezza token, utile per cogliere eterogeneità lessicale (mix di parole molto brevi e molto lunghe).
- **sentence_length_mean**: lunghezza media delle frasi (in token), proxy diretto del concetto di wps (words per sentence) usato nel paper.
- **sentence_length_median**: mediana della lunghezza delle frasi, robusta alla presenza di frasi eccezionalmente lunghe (tipiche di alcune recensioni dettagliate).
- **sentence_length_std**: variabilità della lunghezza delle frasi, utile per distinguere testi “uniformi” (stile semplice) da testi con alternanza di frasi brevi e lunghe (stile più naturale).
- **syllables_per_token_mean**: media delle sillabe per token, indicatore di complessità fonologica/lessicale (parole mediamente più complesse).
- **syllables_per_token_median**: mediana delle sillabe per token; rende la misura meno sensibile a termini tecnici rari.
- **syllables_per_token_std**: variabilità delle sillabe per token, che può riflettere diversità del vocabolario.
- **n_tokens**: numero totale di token secondo spaCy/textdescriptives; è una misura di lunghezza alternativa a word_count (può differire per tokenizzazione) e viene usata come base per alcune normalizzazioni.
- **n_unique_tokens**: numero di token distinti; misura di ricchezza lessicale grezza, sensibile alla ripetizione.
- **proportion_unique_tokens**: rapporto illustrativo tra token distinti e token totali; cattura la diversità lessicale in modo normalizzato e aiuta a diagnosticare testi ripetitivi o templati.
- **n_characters**: numero di caratteri.
- **n_sentences**: numero di frasi; è la base per confronti “per frase” e può indicare differenze di struttura discorsiva (molte frasi brevi vs poche frasi lunghe).
- **flesch_reading_ease**: indice di leggibilità che aumenta con testi più semplici; è usato per verificare se le fake sono scritte in modo più elementare o più artificiosamente elaborato.
- **flesch_kincaid_grade**: stima del “grado scolastico” necessario per comprendere il testo; utile per confrontare livelli di complessità attesa tra fake e real.

- **smog**: indice focalizzato sulla presenza di parole polisillabiche; segnala testi con lessico più complesso e viene interpretato in continuità con il tema “word complexity” del paper.
- **gunning_fog**: indice di leggibilità basato su lunghezza frasi e parole complesse; spesso sensibile a stili verbosi e artificiosi.
- **automated_readability_index**: indice che enfatizza caratteri per parola e parole per frase; utile per differenze sottili tra testi brevi e lunghi.
- **coleman_liau_index**: indice centrato su lettere per parola; complementare agli indici basati su sillabe e utile quando il calcolo delle sillabe è rumoroso.
- **lix**: indice di leggibilità diffuso in contesti europei; sensibile a parole lunghe e frasi lunghe, quindi comparabile al razionale *wps/sixltr*.
- **rix**: variante semplificata che enfatizza parole lunghe; utile per intercettare recensioni con vocabolario “poco naturale” o eccessivamente tecnico.

Macro-area	LIWC nel paper (sigla)	Feature nel progetto (proxy/colonne)
Lunghezza e complessità	<i>len</i> , <i>wps</i> , <i>sixltr</i>	char_len, word_count, avg_word_length, sentence_length_mean, token_length_mean (+ indici leggibilità)
Punteggiatura	<i>all_punc</i> , <i>period</i> , <i>exclam</i>	punctuation_marks_per_review, punct_[SYM]_count, punct_[SYM]_ratio_sent
Pronomi	<i>ipron</i> (e famiglie pronoun/ppron rimosse)	pronoun_count, pronoun_ratio_sent
Verbi e orientamento temporale	<i>verb</i> , <i>auxverb</i> , <i>focuspast/focuspresent/focusfuture</i>	verb_count, verb_ratio_sent, verb_tense_{TENSE}/*
Denaro e numeri	<i>money</i> , <i>number</i>	money_* , money_mentions_percentage (e, indirettamente, token numerici nei POS)
Assenso/accordo	<i>assent</i>	agreement_words_count
Affective / sentiment	<i>negemo</i> , <i>anx</i> , <i>anger</i> , <i>sad</i> (e posemo rimosso)	negative, neutral, positive
Categorie LIWC “non replicate”	<i>analytic</i> , <i>clout</i> , <i>tone</i> , <i>dic</i> , <i>function</i> , ecc.	—

ANALISI FEATURE LINGUISTICHE

L'analisi delle feature linguistiche è stata impostata come un'**esplorazione comparativa** tra due contesti sperimentali eterogenei:

- I. il dataset *Deceptive* (bilanciato, 800 review *legit* e 800 *fake*)
- II. il dataset *Yelp New York* (fortemente sbilanciato, con una netta prevalenza di review *legit* rispetto alle *fake*).

Tale differenza strutturale è stata trattata come un vincolo metodologico rilevante: nel caso *Deceptive* la lettura dei segnali è meno influenzata dalla distribuzione delle classi, mentre su *Yelp* la valutazione di qualunque feature richiede maggiore cautela interpretativa (rischio di effetti "di massa", outlier e fenomeni di duplicazione più marcati). In linea con il paper, che imposta una prima fase di **analisi descrittiva** (t-test su 93 categorie LIWC) prima di passare a modelli predittivi (logistic regression), anche nel progetto l'analisi è stata organizzata in modo da:

- A. descrivere le distribuzioni delle feature per classe;
- B. individuare pattern potenzialmente discriminanti e possibili variabili confondenti (es. lunghezza, rating);
- C. motivare eventuali scelte di feature selection successive (riduzione di ridondanza e controllo della multicollinearità).

È importante sottolineare che, anche quando una feature risulta statisticamente differente tra classi, **le distribuzioni tendono a essere ampiamente sovrapposte**.

Di conseguenza, la separazione tra *fake* e *legit* emerge soprattutto **dalla combinazione multivariata** di segnali deboli e non da singoli marker univoci.

Sul piano operativo, l'EDA è stata condotta combinando **statistiche descrittive** (ad es. `describe()` per stimare range, quartili e dispersione) e **visualizzazioni mirate** per catturare differenze di forma (asimmetria, code, concentrazione su valori estremi).

In particolare, per le variabili probabilistiche derivate da modelli (ad es. *bot/no_bot*, *spam/no_spam*, *sentiment*) sono state valutate sia le statistiche riassuntive sia rappresentazioni di composizione (grafici a torta per la distribuzione media di *positive/neutral/negative*), mentre per le feature strutturali legate alla lunghezza sono stati utilizzati boxplot con outlier, così da isolare rapidamente l'effetto di review estremamente corte o estremamente lunghe.

Un punto metodologico centrale, coerente con le evidenze del paper, è stata la distinzione tra **feature “di contenuto affettivo”** e **feature “di struttura/stile”**.

Il paper mostra che le review *fake* tendono ad avere segnali più elevati, pur con effetti spesso contenuti e sovrapposizione tra classi, su categorie legate a emotività negativa e a marcatori stilistici (es. *negemo*, *anx*, *anger*, *sad*; e anche *all_punc*, *period*, *exclam*), mentre le review *real* risultano più lunghe e più “complesse” (*len*, *wps*, *sixltr*) e più ricche di indici di stile/autorità.

Nel progetto queste dimensioni sono state “ricalibrate” su proxy riproducibili: da un lato il sentiment modellistico (negative/neutral/positive) e la soggettività (subjectivity) come approssimazioni della componente affettiva, dall’altro metriche di lunghezza/leggibilità e distribuzioni grammaticali.

Per ridurre l’effetto confondente della sola lunghezza, l’analisi ha privilegiato non soltanto conteggi assoluti ma anche **misure normalizzate**.

In concreto, per famiglie di feature particolarmente sensibili alla dimensione del testo (verbi, pronomi, punteggiatura, menzioni di denaro) sono state adottate varianti “per frase” (*ratio per sentence*), in modo da distinguere un aumento “meccanico” dovuto a testi più lunghi da un aumento “strutturale” dovuto a uno stile realmente diverso.

Un ulteriore aspetto analitico, particolarmente importante nei dataset Kaggle, è stato il controllo di **qualità del dato testuale** e il rischio di scorciatoie predittive.

In *Deceptive* la duplicazione nelle *fake* risulta assente, mentre in *Yelp* emergono duplicati sia nelle *fake* sia nelle *legit*, inclusi casi di testi molto corti ripetuti molte volte (es. stringhe minime o placeholder), con implicazioni dirette su feature come lunghezza, punteggiatura e segnali “bot-like”.

Questo controllo è stato trattato come prerequisito interpretativo: se una quota non trascurabile di review è duplicata, parte della separabilità tra classi può dipendere da artefatti di compilazione piuttosto che da marcatori psicologico-linguistici generalizzabili.

Infine, l'analisi è stata esplicitamente impostata come confronto tra **segnali attesi dal paper e segnali empiricamente osservabili** con le feature implementate nel progetto. Ad esempio:

- I. l'**aumento in fake di categorie come money** del paper viene verificato tramite proxy basati su pattern lessicali (menzioni di “\$ / dollar”)
- II. la **maggior incidenza di ipron/verb/auxverb/negate** viene esplorata tramite statistiche e distribuzioni di POS/tempi verbali e densità per frase;
- III. le differenze su **all_punc/period/exclam** vengono analizzate con conteggi di punteggiatura e distribuzioni per tipo di simbolo.

In modo complementare, per categorie LIWC non replicabili senza dizionari dedicati (*analytic, clout, tone, dic, function*) l'analisi si è spostata su misure strutturali e di leggibilità (token/sentence length, indici di readability), mantenendo l'obiettivo sostanziale del paper: caratterizzare tendenze tra *fake* e *real* in termini di complessità e stile, prima di qualunque fase di classificazione.

RISULTATI ANALISI FEATURE LINGUISTICHE

I risultati dell'analisi evidenziano che una parte non trascurabile delle differenze osservate tra recensioni *fake* e *legit* è guidata da feature **strutturali** (lunghezza e complessità) e da alcuni proxy **comportamentali/di anomalia** (bot-likeness), mentre le feature più strettamente "psicometriche" o lessicali (es. *money*, *assent*) mostrano segnali più deboli o fortemente dipendenti dal dataset.

Sul piano affettivo, i punteggi di sentiment modellistico mostrano una tendenza media in cui le review *fake* risultano mediamente meno positive e più negative rispetto alle *legit*, con un effetto più marcato nel dataset *Deceptive* rispetto a *Yelp New York*. Questa evidenza è coerente con una maggiore incidenza di emozioni negative nelle recensioni ingannevoli, compatibile con l'ipotesi che in alcuni contesti possa emergere una maggiore polarizzazione emotiva.

GRAFICO 1: Distribuzione media del sentiment per classe – Deceptive

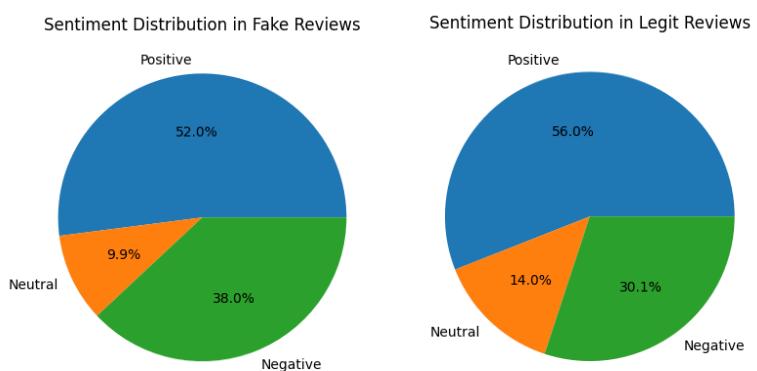
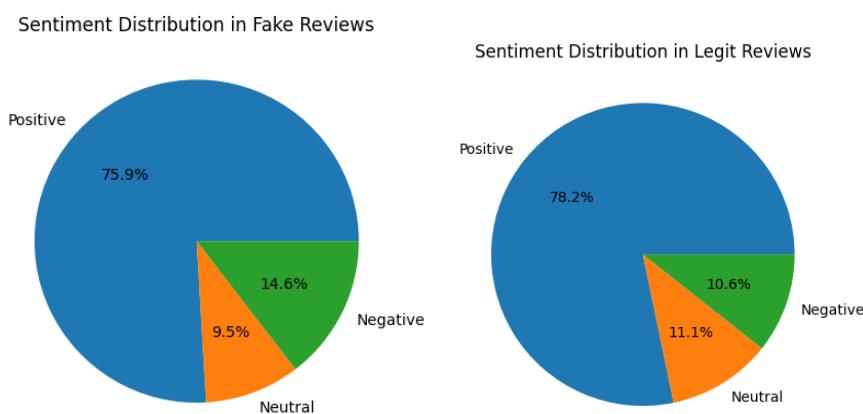


GRAFICO 2: Distribuzione media del sentiment per classe – Yelp New York



Per quanto riguarda le **feature strutturali**, la lunghezza della recensione (in parole e caratteri) emerge come un fattore informativo nel dataset *Yelp New York*, dove le fake risultano in media più brevi.

Va però evidenziato che le distribuzioni risultano ampiamente sovrapposte, e il segnale è trainato soprattutto da code/outlier e micro-review.

Nel dataset *Deceptive*, invece, tale differenza è più contenuta, coerentemente con la natura “controllata” del corpus (stesso dominio, lunghezze più omogenee).

Questo risultato è compatibile con il paper, che osserva recensioni reali mediamente più lunghe e con maggiore complessità (*len*, *wps*, *sixltr*), interpretandole come prodotto di esperienze autentiche più ricche di dettagli.

Nel nostro caso, l’effetto su *Yelp* è amplificato anche dalla presenza di **micro-review** e testi estremamente brevi che possono influenzare fortemente i quantili, suggerendo che parte del segnale sia legata alla distribuzione delle lunghezze nel dataset e alla sua qualità testuale.

GRAFICO 3: Boxplot word_count per classe – Deceptive

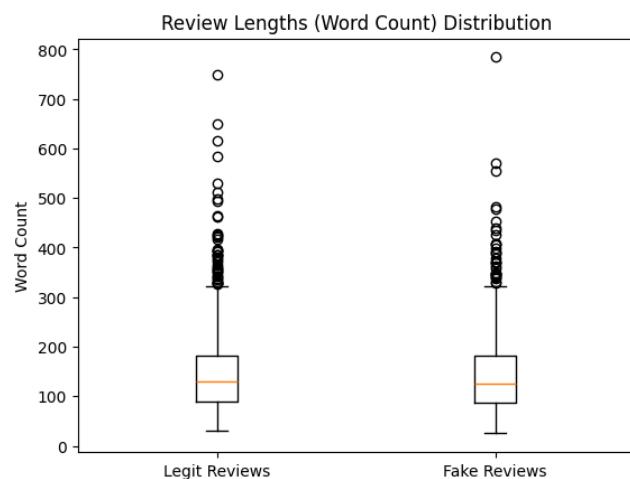


GRAFICO 4: Boxplot char_len per classe – Deceptive

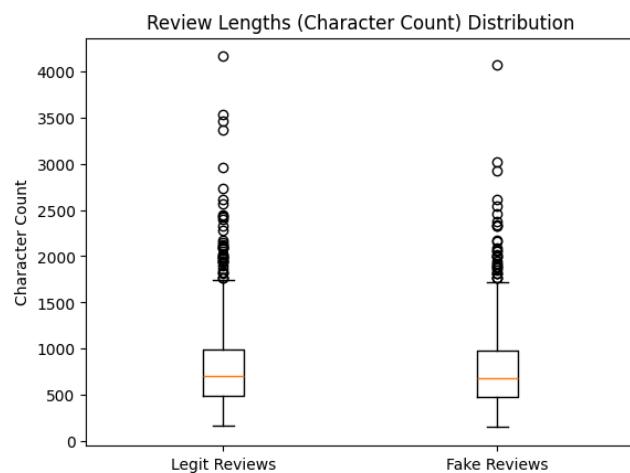


GRAFICO 5: Boxplot word_count per classe – Yelp New York

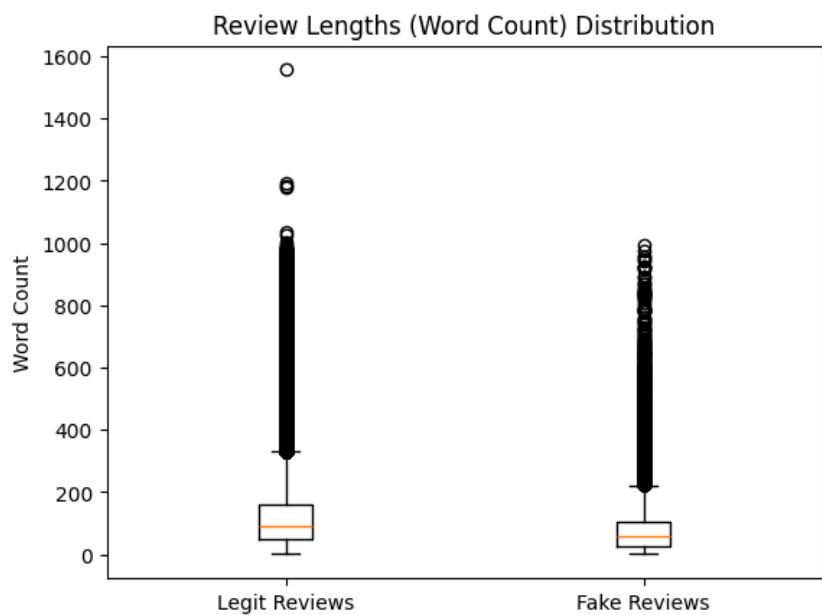
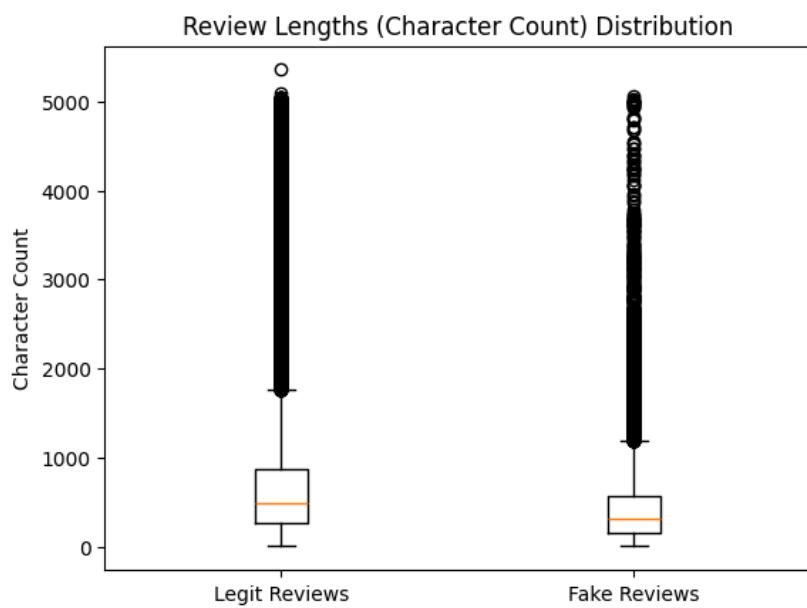


GRAFICO 6: Boxplot char_len per classe – Yelp New York



Tra i proxy comportamentali, il punteggio **bot** mostra una tendenza ricorrente a valori più elevati nelle recensioni etichettate *fake* in entrambi i dataset, suggerendo che una parte del contenuto falso mostra segnali compatibili con ‘non genuinità’ secondo il detector.

Per le feature stilistico/sintattiche, i risultati suggeriscono un comportamento non sempre concorde con il paper, evidenziando la sensibilità al contesto.

La **punteggiatura totale** per review e la sua distribuzione nel paper tende ad aumentare nelle *fake*, mentre nei nostri dataset si osserva frequentemente una maggiore punteggiatura nelle *legit*, plausibilmente come conseguenza indiretta della maggiore lunghezza e complessità sintattica delle recensioni autentiche. Analogamente, la densità verbale per frase mostra **segnali dataset-specifici**: nel Deceptive le *fake* tendono a una maggiore densità, in linea con l’idea che i testi ingannevoli contengano più azione e narrazione, mentre nel Yelp la differenza si riduce o può invertirsi.

GRAFICO 7: KDE verbi per frase (%) per classe – Deceptive

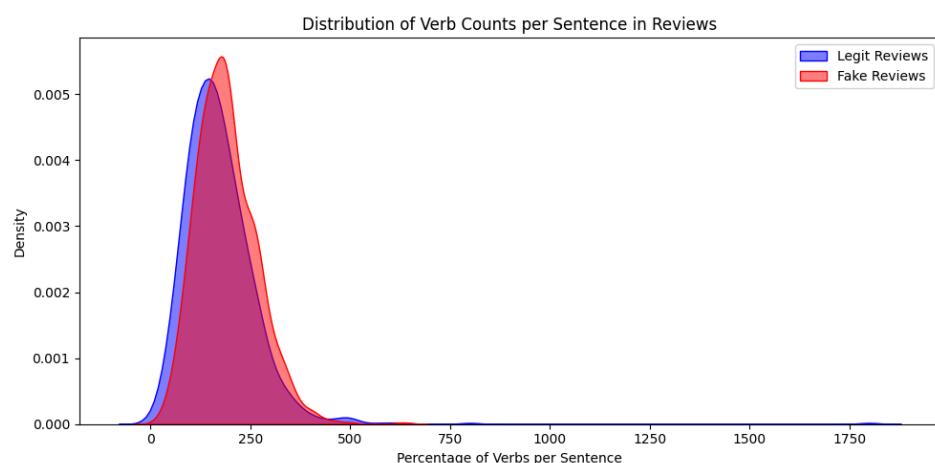
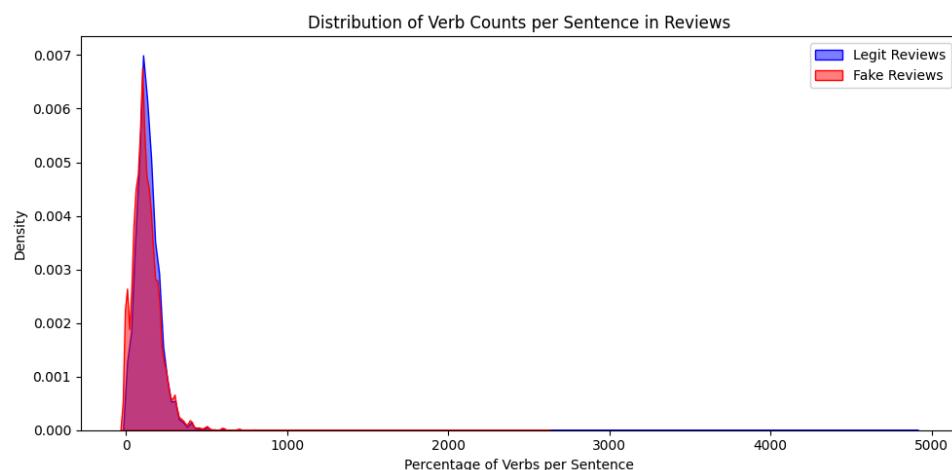


GRAFICO 8: KDE verbi per frase (%) per classe – Yelp New York



Infine, le feature lessicali mirate collegate a categorie LIWC specifiche mostrano nel nostro caso un impatto limitato o non sistematico: le **menzioni di denaro** sono rare e spesso concentrate in poche review, mentre l'assenso risulta poco frequente e con differenze marginali tra classi.

GRAFICO 9: Money mentions (%) per classe – Deceptive

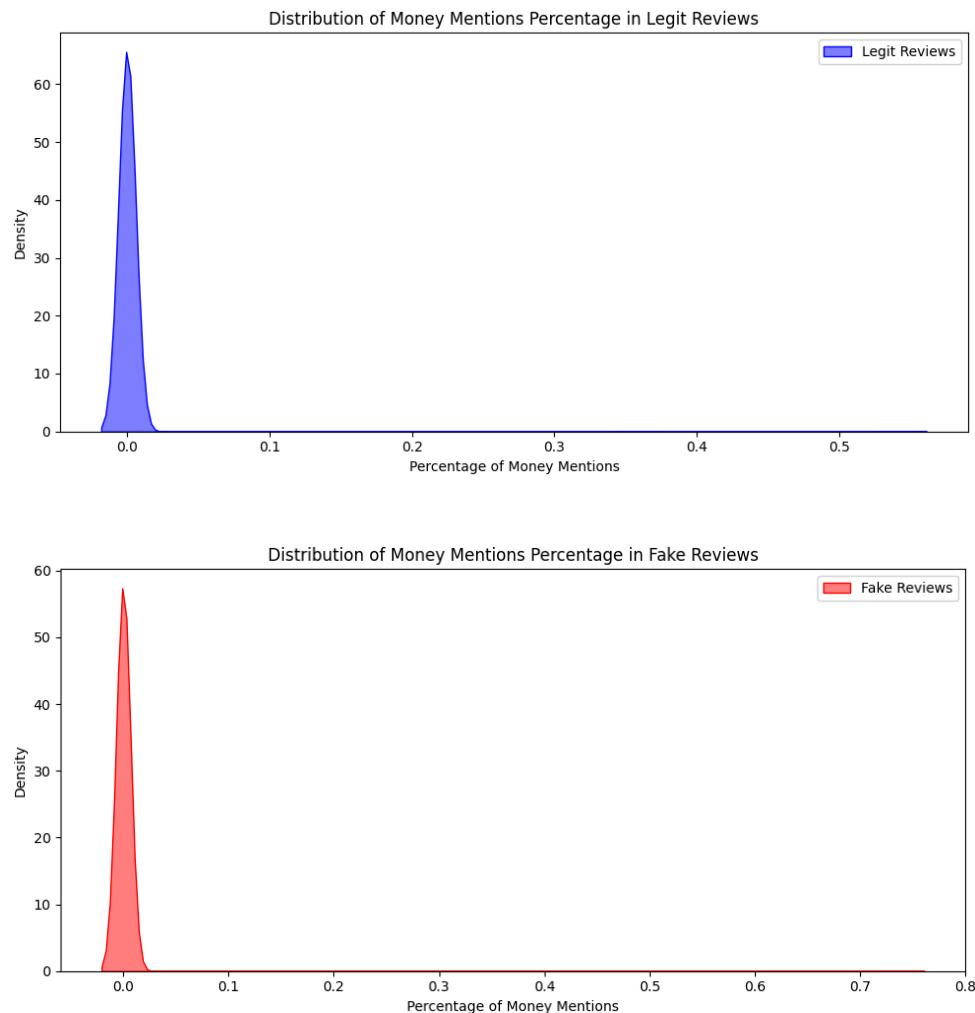


GRAFICO 10: Money mentions (%) per classe – Yelp New York

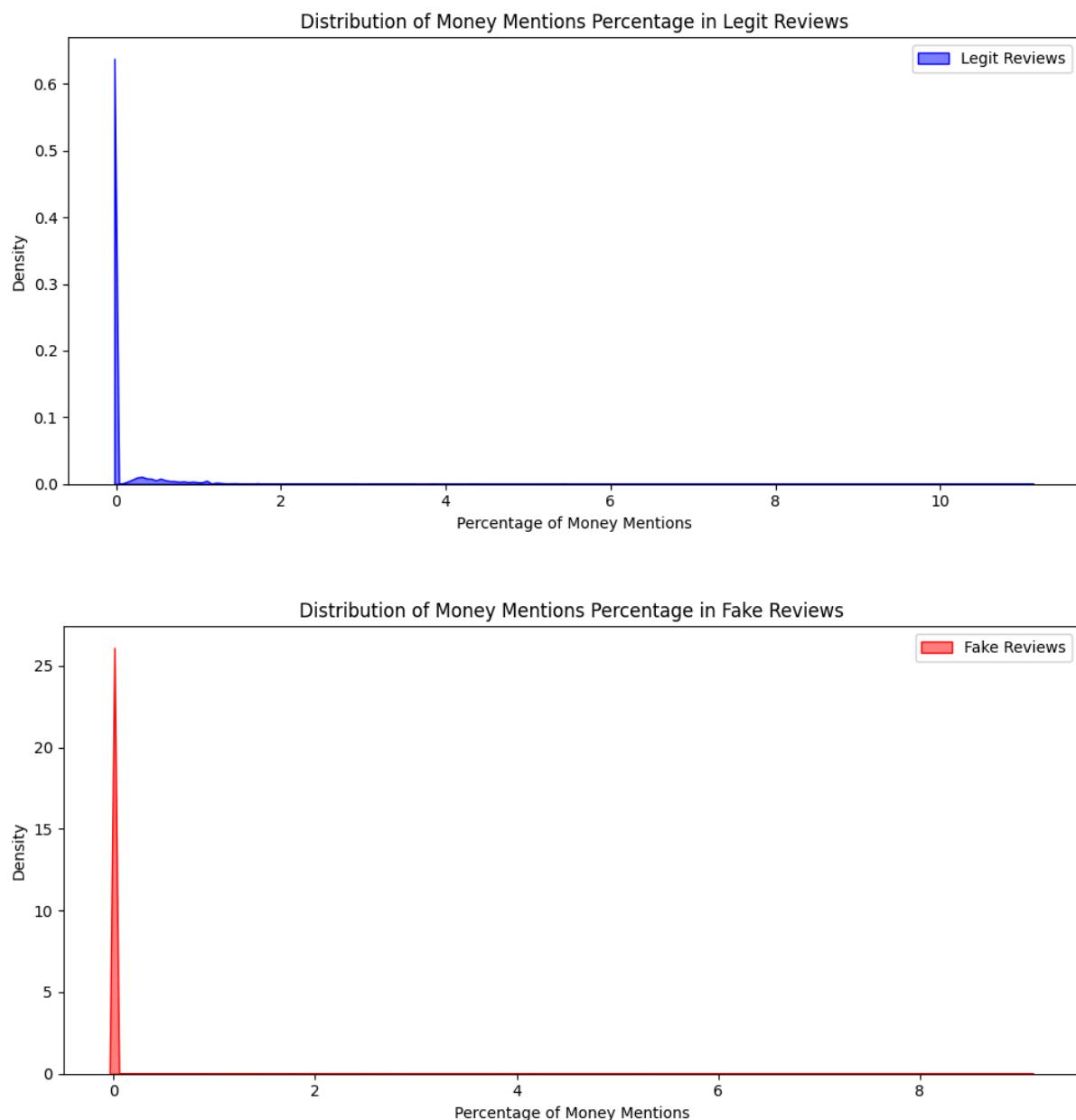


GRAFICO 11: Agreement words count per classe – Deceptive

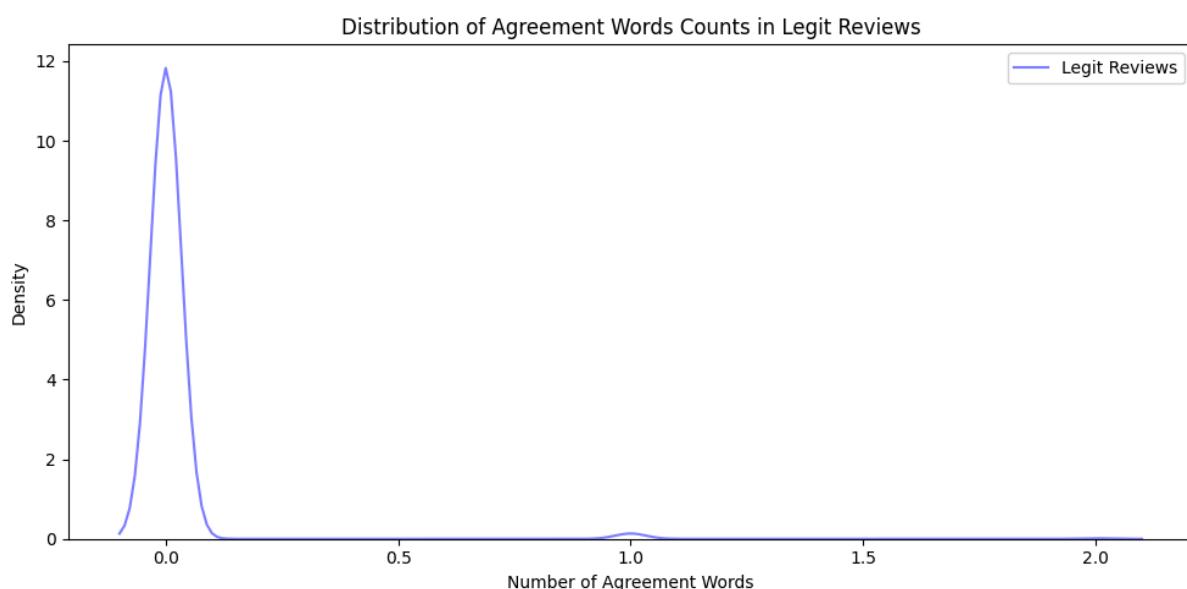
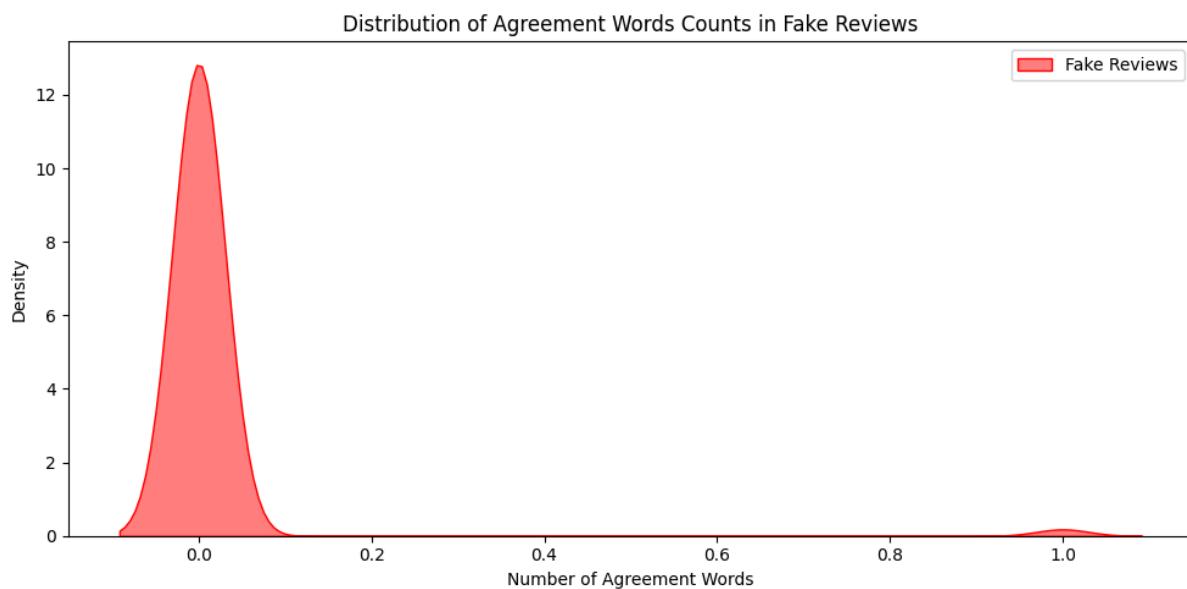
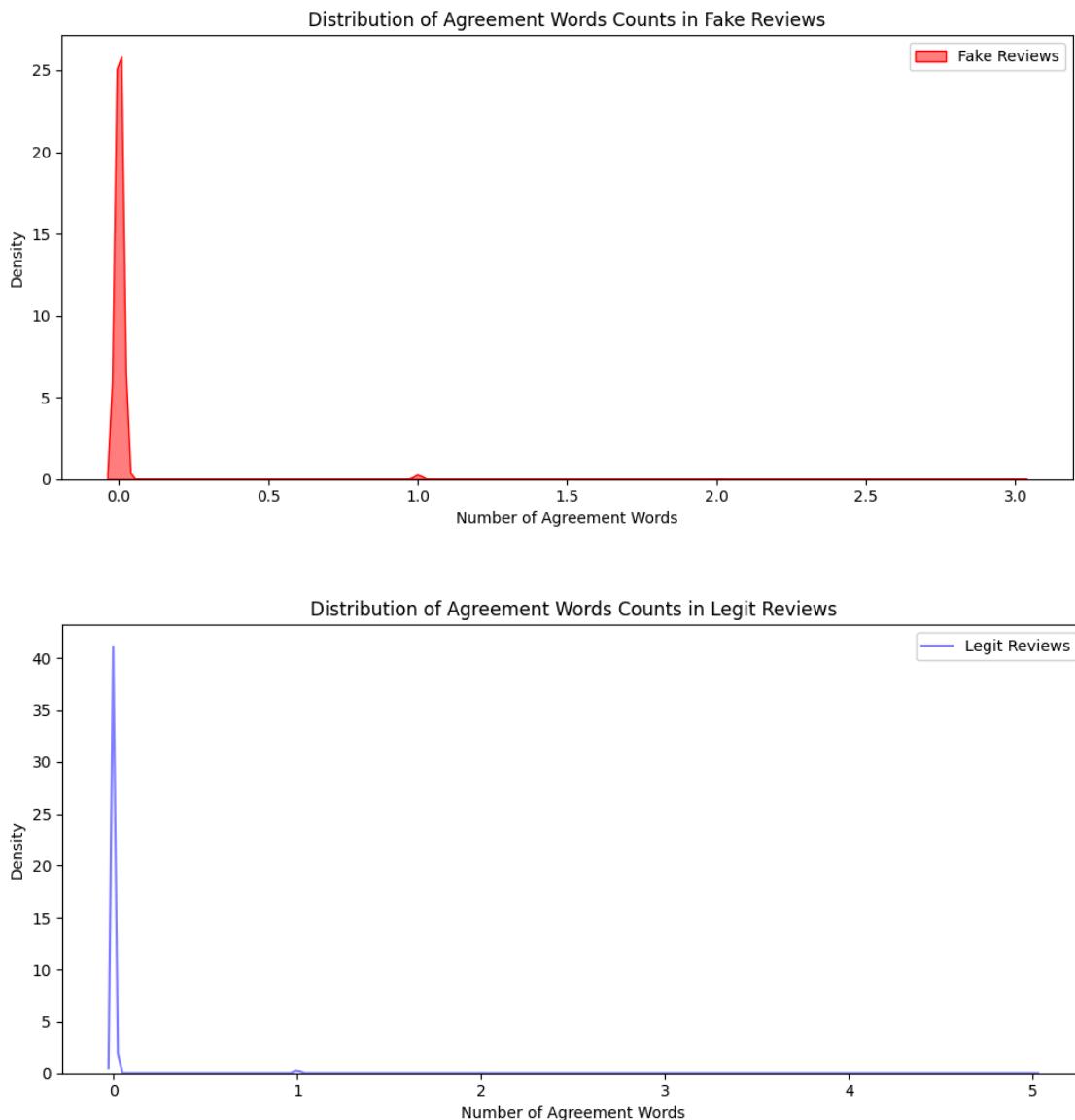


GRAFICO 12: Agreement words count per classe – Yelp New York



Nel complesso, i risultati confermano l'utilità di un set di **feature linguistiche interpretabili come baseline**, con particolare enfasi su lunghezza/leggibilità e su alcuni indicatori di anomalia (bot-likeness), ma indicano anche che la generalizzazione di singoli marker “classici” della deception detection non è garantita e va valutata empiricamente per ciascun dominio, anche perché i segnali univariati presentano tipicamente ampia sovrapposizione tra classi.

PCA ANALYSIS

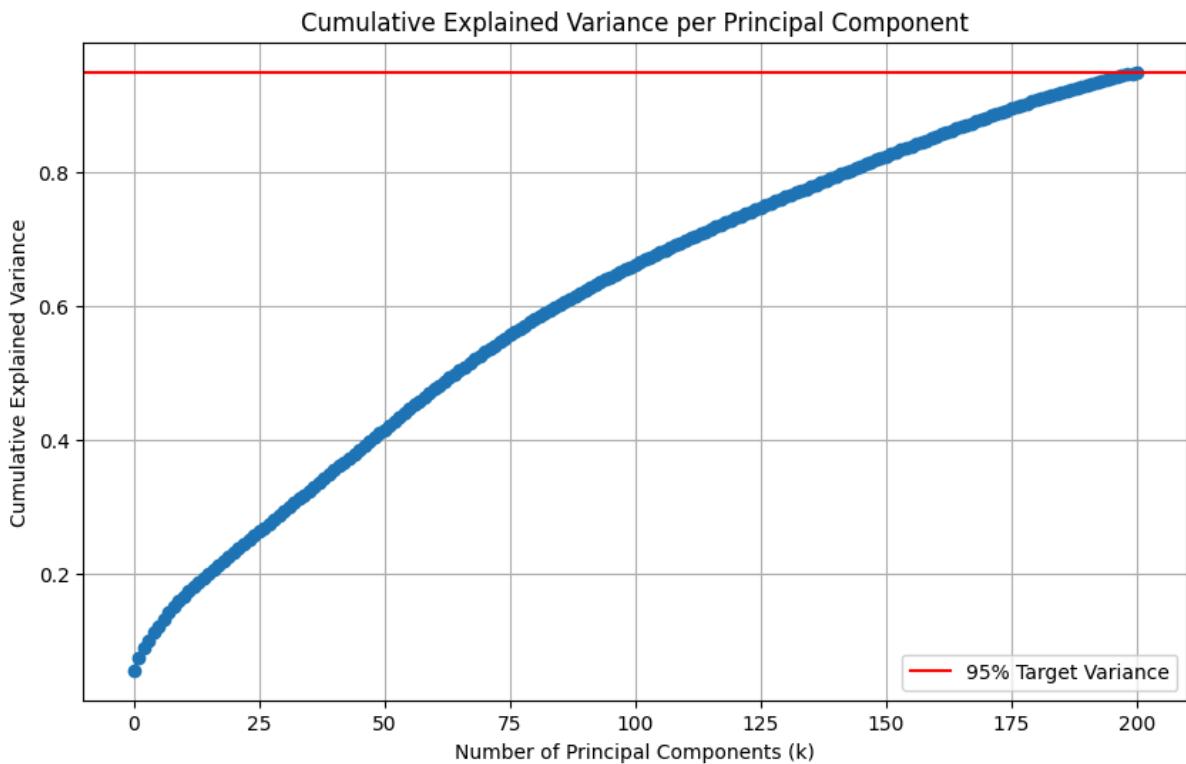
La **Principal Component Analysis (PCA)** è una tecnica di **analisi statistica multivariata**, ovvero un metodo che consente di analizzare simultaneamente più variabili osservate, studiandone le relazioni reciproche. Il suo obiettivo principale è la **riduzione della dimensionalità**, intesa come diminuzione del numero di variabili necessarie a descrivere un insieme di dati, senza perdere una quantità significativa di informazione.

A tal fine, la PCA trasforma le variabili originali in un nuovo insieme di **componenti principali**, che sono nuove variabili ottenute come **combinazioni lineari** delle variabili di partenza. Queste componenti sono tra loro **ortogonali**, cioè incorrelate (un-correlated), e vengono ordinate in base alla **varianza spiegata**, che rappresenta la quota di variabilità totale dei dati catturata da ciascuna componente. Le prime componenti concentrano quindi la maggior parte dell'informazione contenuta nel dataset, mentre le successive descrivono variazioni via via meno rilevanti. Grazie a queste proprietà, la PCA è ampiamente utilizzata nelle fasi di **esplorazione dei dati**, per individuare strutture latenti, nella **pre-elaborazione**, per ridurre la ridondanza e il **rumore** informativo, e nella **visualizzazione**, facilitando la rappresentazione di dati complessi in spazi a bassa dimensionalità e migliorando l'efficienza delle analisi successive.

RISULTATI PCA ANALYSIS

L'integrazione della PCA nel workflow di *Trasite!* è stata finalizzata alla creazione di un dataset "ML-ready" capace di bilanciare ricchezza informativa e sintesi computazionale. Il processo ha avuto inizio con l'isolamento delle sole feature numeriche (interi e floating point), escludendo identificativi e testi grezzi per focalizzare l'analisi sulla struttura quantitativa dei dati. Prima della decomposizione, è stata eseguita una fase critica di **data preparation**: i valori mancanti sono stati gestiti tramite imputazione basata sulla mediana e, successivamente, l'intero spazio delle feature è stato normalizzato utilizzando uno **StandardScaler**.

Questa standardizzazione è un requisito tecnico imprescindibile per la PCA, poiché garantisce che ogni variabile contribuisca alla varianza totale su una scala comune (media 0 e deviazione standard 1), evitando che metriche con ordini di grandezza elevati oscurino indicatori probabilistici più contenuti (come bot_score).



Il grafico illustra l'andamento della **Varianza Spiegata Cumulata** in funzione del numero di componenti principali k .

- **Asse delle Ascisse (k):** Rappresenta il numero di componenti estratte dal set di feature originali.
- **Asse delle Ordinate:** Indica la percentuale di informazione totale (varianza) catturata dalle prime k componenti.
- **Analisi della Curva:** La curva mostra una crescita costante ma gradualmente decrescente. L'assenza di un "gomito" (elbow) pronunciato nelle prime fasi indica che l'informazione è distribuita su un numero elevato di feature linguistiche e stilistiche, piuttosto che concentrata in pochi indicatori dominanti.
- **Soglia di Taglio (Linea Rossa):** È stata impostata una soglia critica del **95% della varianza totale**. Come si osserva dall'intersezione tra la curva blu e la linea rossa, sono necessari circa **175 componenti principali** per soddisfare questo requisito di conservazione dell'informazione.

Una volta identificata la soglia scelta (95%) di varianza spiegata, il sistema ha applicato la trasformazione finale, proiettando le feature originali, quali le densità di POS tag, i punteggi di leggibilità e le metriche di sentiment, in un nuovo spazio latente di componenti ortogonali (PC_1, PC_2, \dots, PC_n). Il passaggio da un elevato numero di feature grezze a circa **175 componenti** permette di ridurre sensibilmente la complessità del modello computazionale mantenendo la maggior parte dell'informazione, favorendo che i successivi algoritmi di Machine Learning lavorino su uno spazio vettoriale dove ogni coordinata è statisticamente incorrelata con le altre. Questo approccio ha permesso di ridurre in modo significativo la multicollinearità tra le feature linguistiche, fenomeno frequente tra metriche simili come `word_count` e `char_len`, migliorando la stabilità del processo di apprendimento e fornendo ai modelli di classificazione (**Logistic Regression** e **MLP**) input più stabili e meno ridondanti, su cui la separazione tra classi deriva dalla combinazione di componenti. Il risultato di tale procedura è stato infine consolidato nel file `yelp_reviews_ml_ready_pca.csv`, che integra i metadati essenziali con le nuove componenti sintetizzate.

MODELLO ADDESTRATO SU PCA

Per affiancare ai modelli basati su testo un approccio più “classico”, è stata predisposta una pipeline di classificazione supervisionata che opera su feature numeriche compattate tramite **Principal Component Analysis (PCA)**.

Il rationale è duplice: da un lato ridurre drasticamente la dimensionalità e la ridondanza delle variabili originali, dall’altro fornire un baseline che sfrutta esclusivamente segnali quantitativi già ingegnerizzati, evitando l’uso diretto del contenuto linguistico.

Il preprocessing costruisce un dataset “ML-ready” selezionando le sole colonne numeriche (escludendo identificativi e testo), imputando i valori mancanti con la mediana, standardizzando le feature e applicando PCA con soglia di varianza spiegata (95%), ottenendo un vettore compatto di componenti $PC_1 \dots PC_k$ (nel setup operativo utilizzato per il training, $PC_1 \dots PC_{175}$).

Su questo spazio latente sono stati addestrati due modelli: una **Logistic Regression** (solver saga) e una **MLPClassifier** (rete neurale feed-forward), entrambi integrati in una pipeline con **PowerTransformer** per stabilizzare distribuzioni e rendere più regolare la geometria delle componenti.

La procedura include inoltre la gestione esplicita dello sbilanciamento con tre strategie alternative: *original*, *undersample*, *oversample*, dove l’obiettivo è riportare il rapporto minoranza/maggioranza verso 0.5 (equivalente a un 2:1), applicando il campionamento solo quando la distribuzione del train è più sbilanciata del target.

La scelta degli iperparametri è stata effettuata tramite **RandomizedSearchCV** con split stratificato interno e ottimizzazione su **F1 macro**, in modo da non favorire la sola classe maggioritaria.

RISULTATI MODELLI ADDESTRATI SU PCA

I risultati dei modelli addestrati su PCA vengono interpretati principalmente come una baseline quantitativa: l'uso di **F1 macro** come metrica obiettivo e di selezione è coerente con la necessità di valutare in modo equilibrato entrambe le classi, soprattutto in presenza di forte sbilanciamento. In generale, le differenze tra strategie di bilanciamento si manifestano come un compromesso tra capacità di intercettare la classe minoritaria e contenimento dei falsi positivi sulla classe maggioritaria: *undersampling* tende ad aumentare la sensibilità verso la minoranza riducendo la dominanza della classe maggioritaria in training, mentre *oversampling* può incrementare la copertura della minoranza senza scartare esempi ma con il rischio di introdurre ridondanza informativa (repliche) e quindi overfitting.

La **Logistic Regression**, per costruzione, fornisce un punto di partenza più stabile e interpretabile nello spazio PCA, mentre la **MLP** può catturare relazioni non lineari tra componenti, risultando potenzialmente più performante a fronte di maggiore variabilità e sensibilità agli iperparametri. In ogni caso, l'adozione di un tuning leggero (random search) e di un singolo split train/test stratificato consente di ottenere una stima comparativa rapida delle configurazioni (modello \times strategia), producendo un leaderboard finale e identificando un “champion” sulla base del miglior F1 macro su test; tale esito costituisce un riferimento utile per stabilire quanto informazione discriminante sia effettivamente contenuta nelle sole feature numeriche compattate, prima di passare a modelli testuali più espressivi.

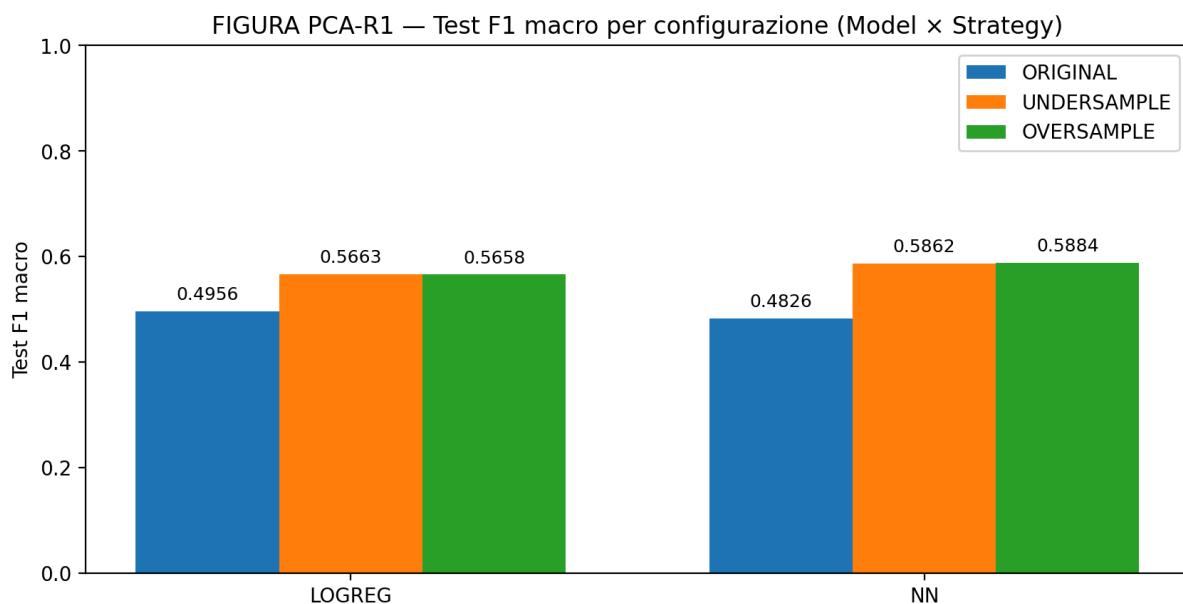


FIGURA PCA-R2 — Leaderboard Test F1 macro (ordinato per performance)

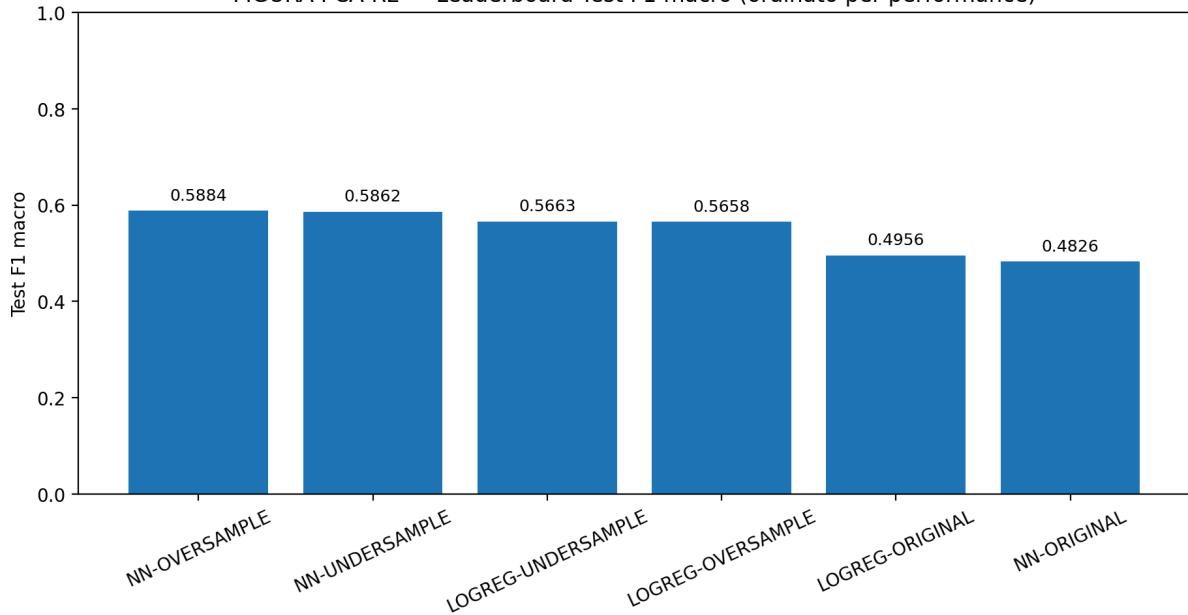


FIGURA PCA-R3 — Effetto del bilanciamento ($\Delta F1$ macro rispetto a ORIGINAL)

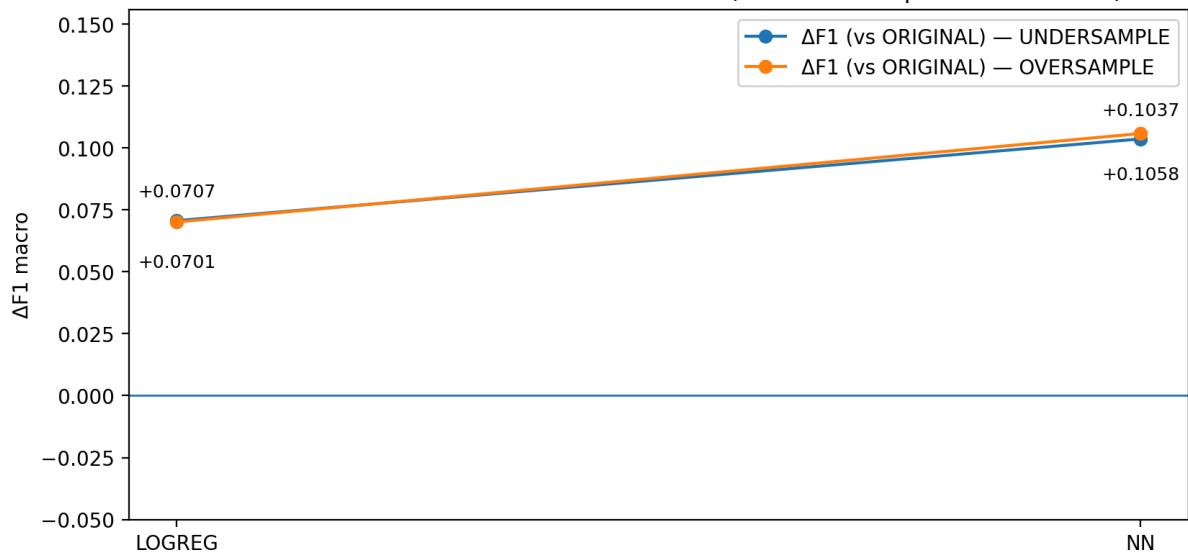
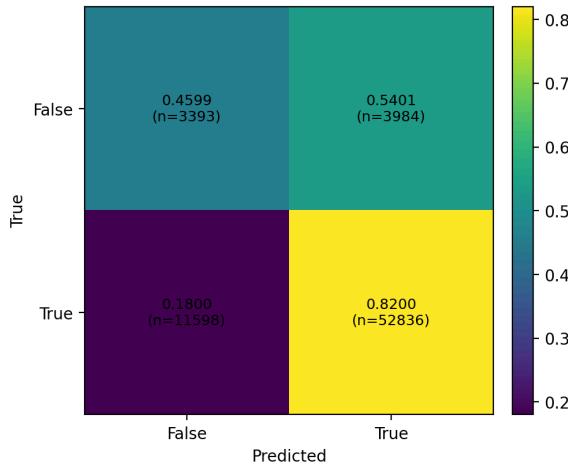


FIGURA PCA-R4 — Confusion matrix normalizzata (NN + OVERSAMPLE, Test set)



ARCHITETTURA MODELLI FAKE REVIEW DETECTION

L'architettura dei modelli impiegati per la "fake review detection" nel progetto può essere letta come un passaggio da approcci **feature-based** (psico-linguistici/stilistici) a modelli **Transformer** addestrati end-to-end sul testo. Nel paper LIWC, le 93 categorie psicométriche vengono prima confrontate con t-test e poi impiegate in modelli di regressione logistica; tuttavia, nonostante numerosi predittori significativi, l'accuratezza risulta relativamente modesta e viene evidenziata la necessità di rappresentazioni linguistiche più avanzate.

In questa cornice si collocano i **Transformer**: DistilRoBERTa rappresenta una variante più compatta ed efficiente, utile come baseline, mentre DeBERTa è progettata per potenziare la comprensione contestuale mediante **disentangled attention**, separando l'informazione di contenuto e posizione, così da catturare pattern sottili che possono caratterizzare il linguaggio ingannevole.

Nel progetto, la comparazione DistilRoBERTa vs DeBERTa-v3-xsmall viene interpretata soprattutto alla luce del problema dominante del dominio Yelp: la **classe Fraud (fake)** è minoritaria e richiede un'attenzione esplicita a recall, falsi negativi e scelte di sampling.

FEATURE PER IL TRAINING DI DEBERTA

Nel training di DeBERTa, le "feature" non coincidono con variabili ingegnerizzate, ma sono costituite dalla **sequenza testuale tokenizzata** e dalle **rappresentazioni contestuali** apprese dal backbone.

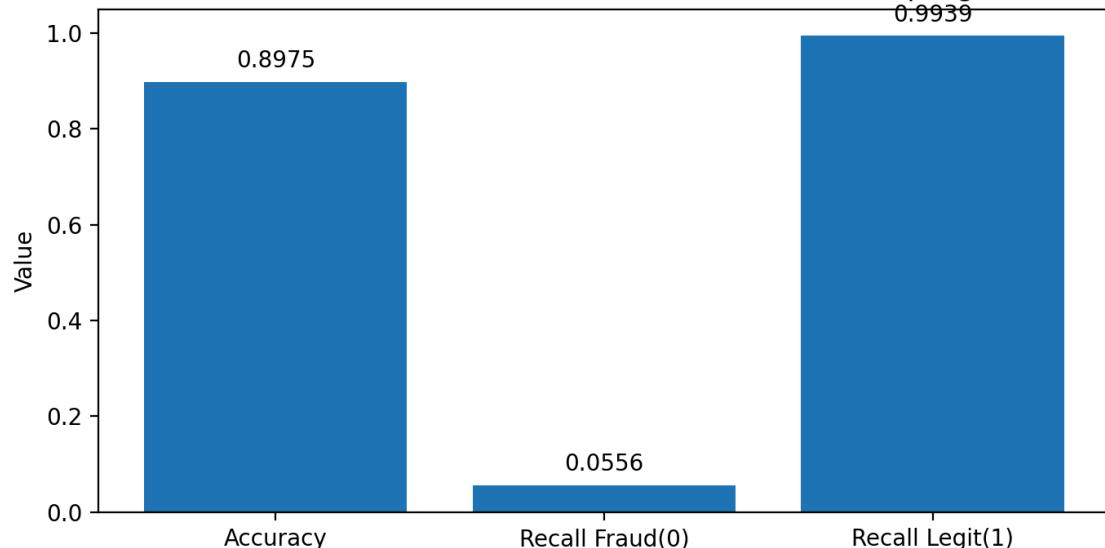
Operativamente, nel progetto l'input è il campo *review* tokenizzato con troncamento a **512 token** e padding dinamico, mentre il target è una label binaria *labels* derivata dal campo *legit*, dopo rimozione dei duplicati testuali per ridurre leakage. La pipeline adotta split stratificato **70/10/20**, early stopping e selezione del best model in base a F1; tali scelte rispondono all'esigenza di valutare non solo la performance media, ma anche la stabilità del training e la generalizzazione in presenza di variabilità, tema enfatizzato anche nei lavori recenti che propongono ulteriori livelli di ottimizzazione e test di robustezza.

PROBLEMA DEL SAMPLING

Il problema sperimentale centrale sui dati Yelp è lo **sbilanciamento di classe**: ottimizzare metriche globali (accuracy, F1 complessiva) può produrre un classificatore che predice quasi sempre *Legit*, ottenendo apparentemente buoni risultati ma fallendo l'obiettivo primario della fake review detection, cioè intercettare la classe Fraud. Questa dinamica è chiaramente evidenziata dal baseline **senza resampling**: metriche aggregate elevate possono coesistere con un recall estremamente basso per Fraud, generando un numero elevato di falsi negativi (fake non rilevate). Per questo motivo, nel progetto il sampling è trattato come variabile sperimentale esplicita: viene applicato **solo al training set** (OVER/UNDER) con rapporto controllato, preservando validation e test nella distribuzione originale per evitare stime ottimistiche.

In fase di lettura dei risultati, la valutazione è quindi centrata su metriche per classe (in particolare **Fraud recall**) e confusion matrix normalizzate, perché rendono esplicito il compromesso operativo tra falsi negativi e falsi positivi.

FIGURA A3 — Effetto dello sbilanciamento (baseline senza resampling: DistilRoBERTa)



RISULTATI TRAINING DEBERTA

I risultati sperimentali confermano un trade-off netto tra performance “globale” (dominata dalla classe maggioritaria) e capacità di individuare Fraud. Considerando due baseline senza resampling, sia DistilRoBERTa sia DeBERTa-v3-xsmall mostrano il comportamento “degenerato” tipico dello sbilanciamento: accuracy elevata (0.8975 e 0.8986) e F1 molto alta sulla classe Legit, ma Fraud recall estremamente basso. In particolare, DistilRoBERTa in modalità originale raggiunge Fraud recall = 0.0556, mentre DeBERTa-v3-xsmall scende ulteriormente a 0.0115, indicando che il classificatore tende quasi sempre a predire Legit (nel caso DeBERTa la confusion matrix evidenzia solo 84 Fraud correttamente riconosciute su 7315). Questi baseline sono metodologicamente cruciali perché dimostrano che l’ottimizzazione della sola accuracy (e, più in generale, di metriche dominate dalla classe maggioritaria) non è adeguata al task di fake review detection: il modello può apparire “ottimo” in termini aggregati pur fallendo quasi completamente sull’obiettivo operativo di intercettare le recensioni fraudolente.

L’introduzione del resampling modifica significativamente il punto di lavoro e rende esplicito il compromesso FP/FN. Per DeBERTa-v3-xsmall, passando dal baseline originale (Fraud recall 0.0115) a UNDER 2:1 (0.2297) e poi a configurazioni più aggressive (OVER 2:1, OVER/UNDER 1:1), il Fraud recall cresce in modo marcato (fino a ~0.566–0.568), ma contestualmente diminuiscono recall e precision sulla classe Legit, con una riduzione dell’accuracy complessiva (fino a ~0.75).

Un andamento analogo si osserva anche per DistilRoBERTa, dove l’undersampling/oversampling incrementa la sensibilità verso Fraud (da 0.0556 a 0.2535–0.5036 a seconda del rapporto), a fronte però di un aumento dei falsi positivi. In termini applicativi, ciò equivale a rendere il sistema più “prudente” (intercetta più fake) accettando il costo di segnalare come sospette alcune review genuine. È rilevante notare che per DeBERTa la ROC-AUC rimane complessivamente nell’intervallo ~0.74–0.76, suggerendo che la capacità discriminativa di ranking del modello resta relativamente stabile; il resampling agisce soprattutto sulla configurazione della decision boundary indotta in training (e quindi sul compromesso operativo tra errori di tipo FP e FN), più che sulla separabilità intrinseca tra le classi.

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

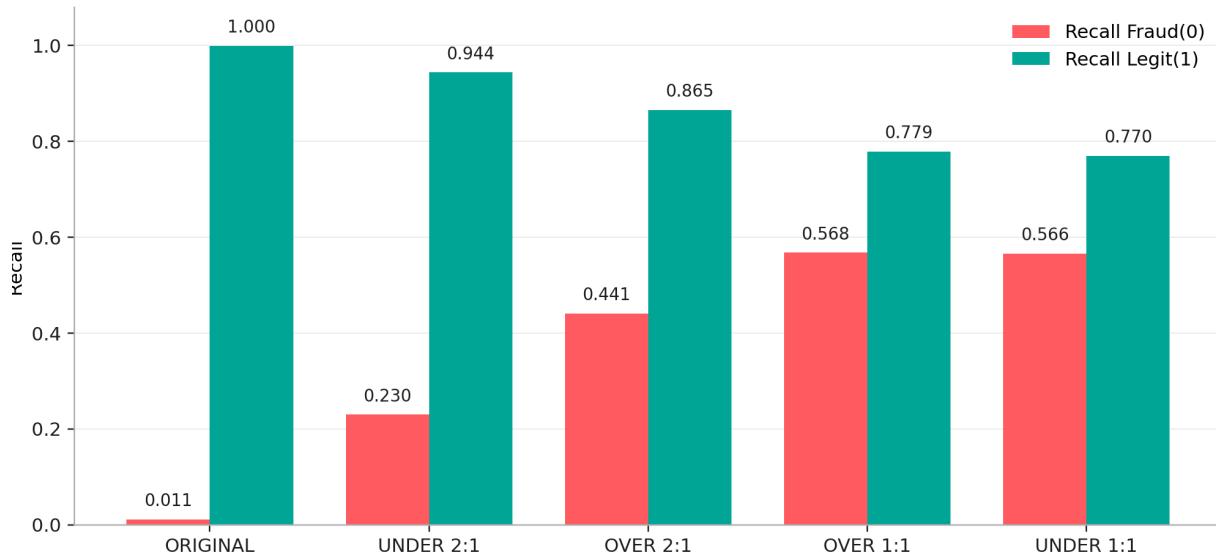


FIGURA R2 — DeBERTa: Accuracy e ROC-AUC vs strategia di sampling

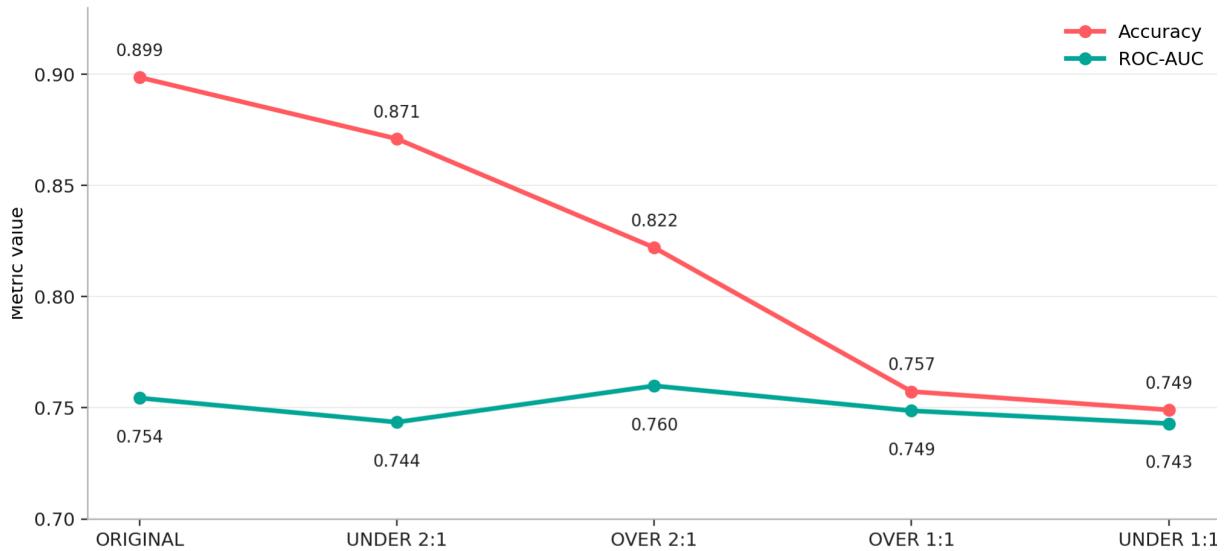


FIGURA R3 — DeBERTa: Confusion matrix normalizzata (ORIGINAL) FIGURA R3 — DeBERTa: Confusion matrix normalizzata (UNDER 2:1)

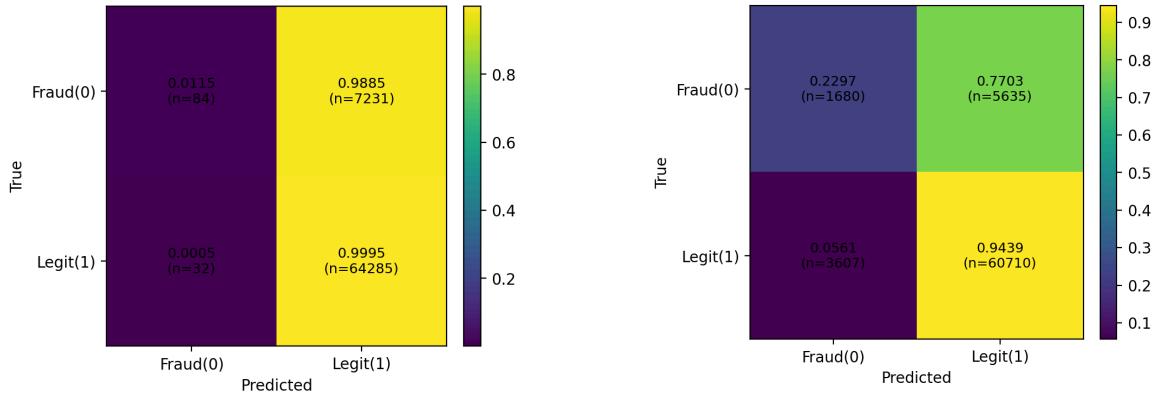


FIGURA R3 — DeBERTa: Confusion matrix normalizzata (OVER 1:1)

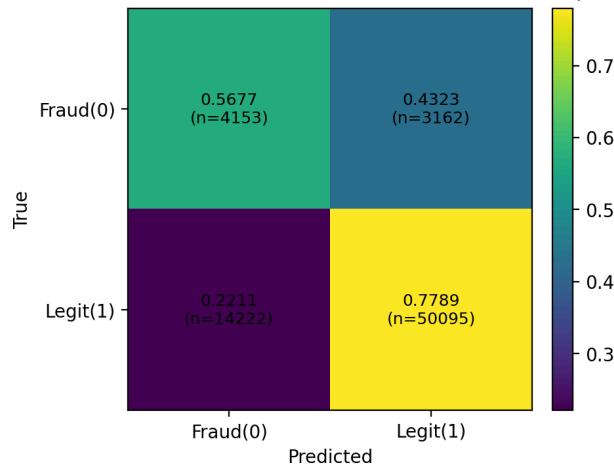


FIGURA R3 — DeBERTa: Confusion matrix normalizzata (OVER 2:1) FIGURA R3 — DeBERTa: Confusion matrix normalizzata (UNDER 1:1)

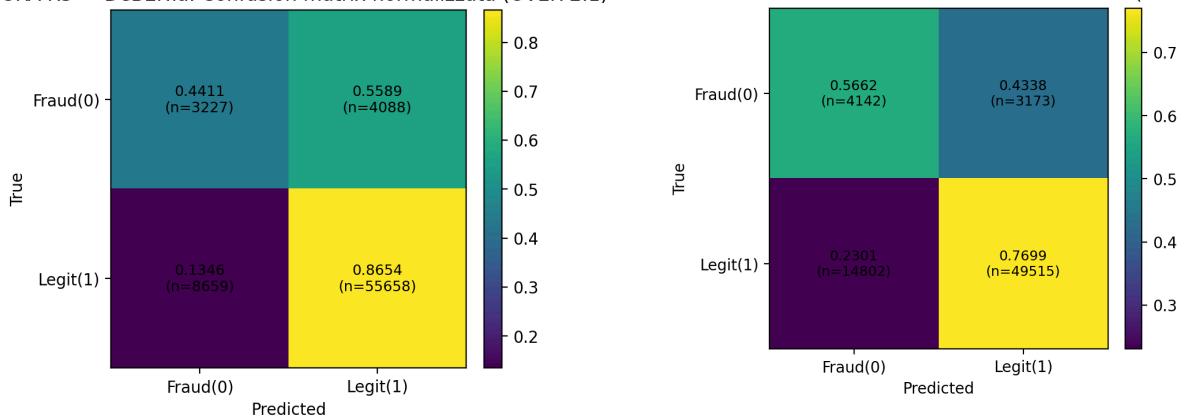


FIGURA R4 — DeBERTa: punti Precision-Recall (classe Fraud)

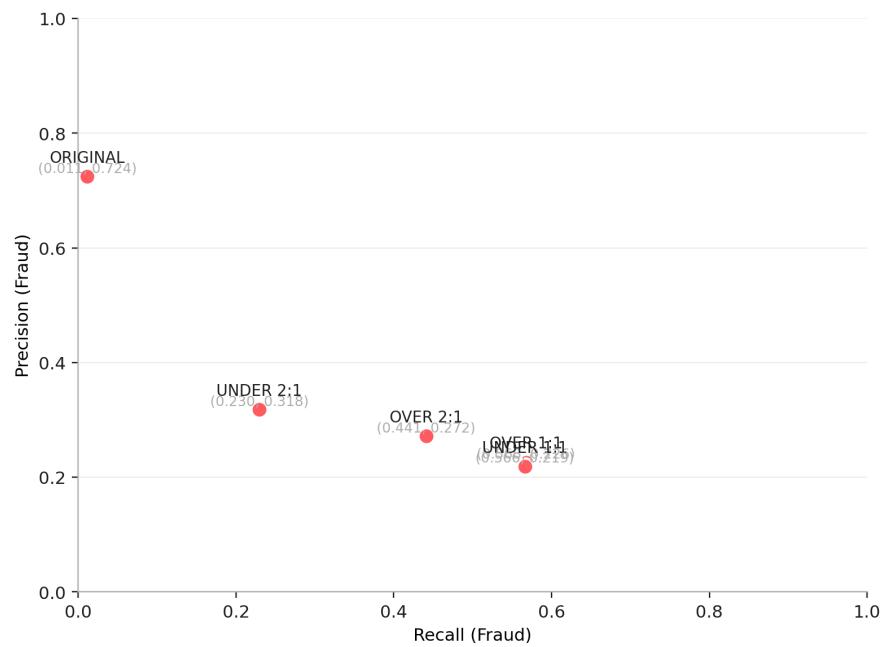


FIGURA R5 — Confronto Recall Fraud(0): DistilRoBERTa vs DeBERTa

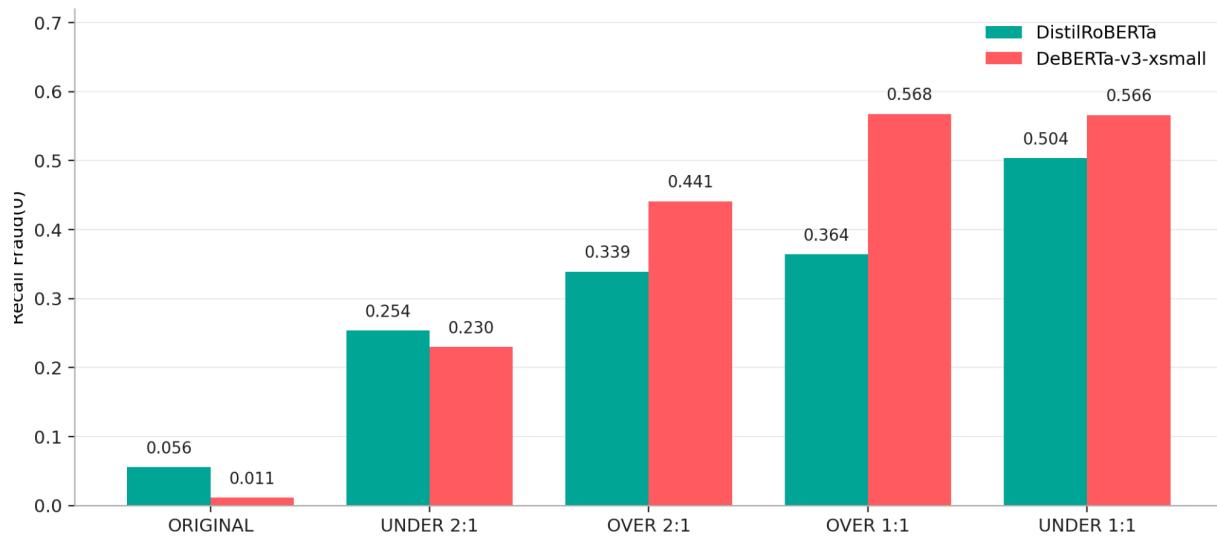


FIGURA F1-1 — DeBERTa: F1-score Legit(1) per strategia di sampling

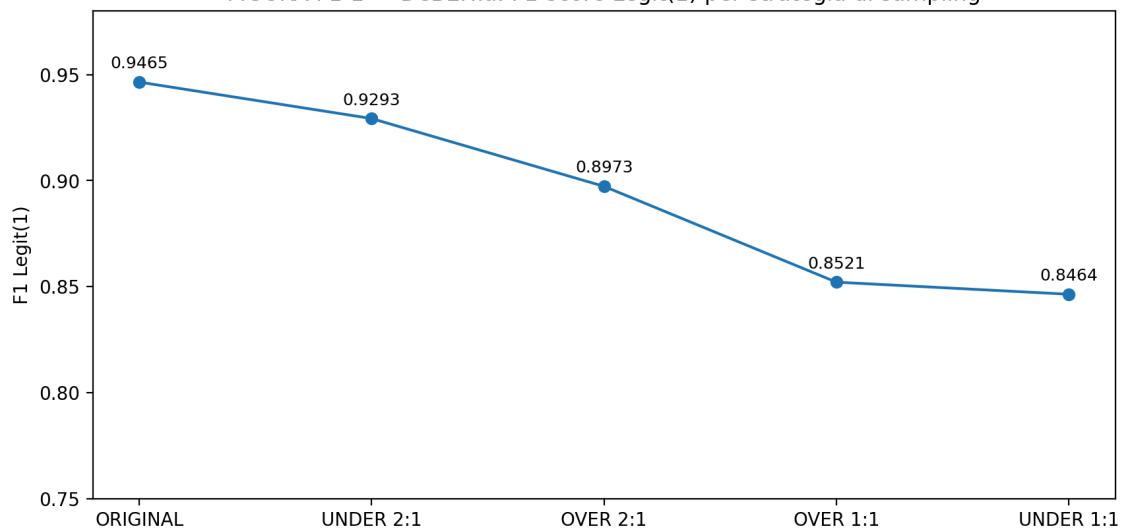


FIGURA F1-2 — DeBERTa: F1-score Fraud(0) per strategia di sampling

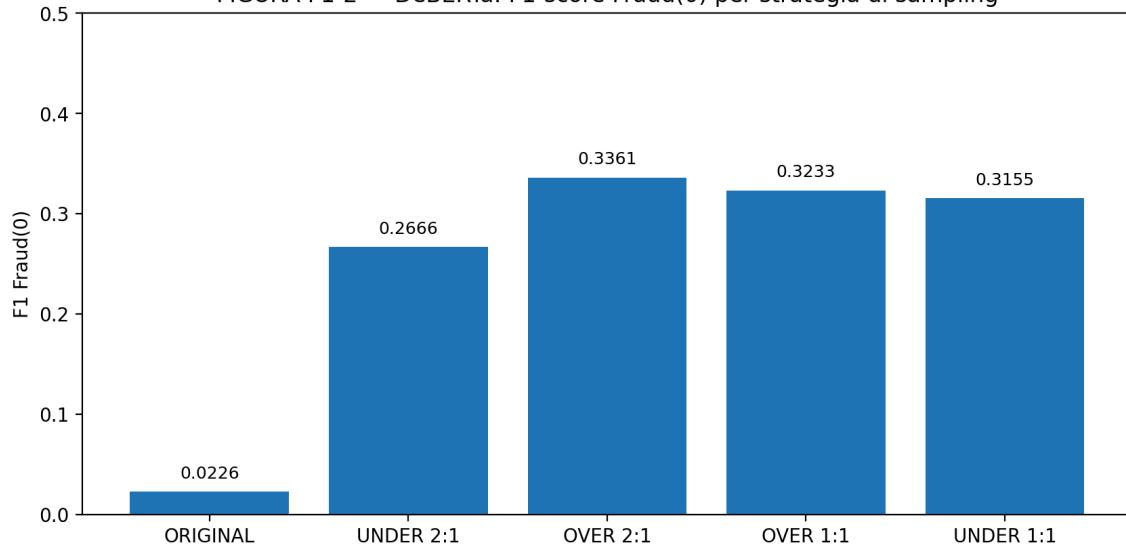


FIGURA F1-3 — DistilRoBERTa: F1-score per classe

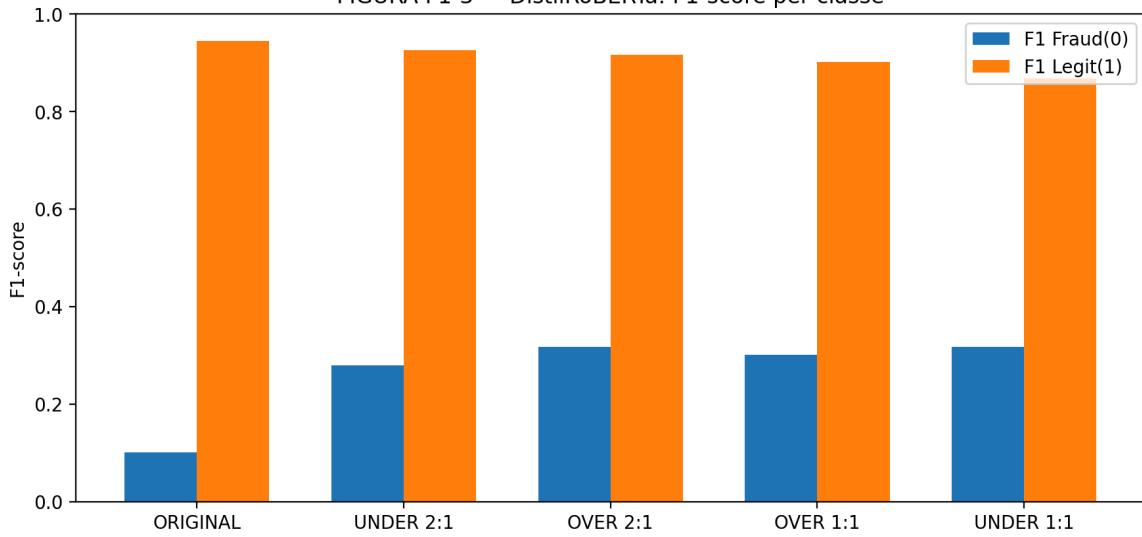


FIGURA F1-4 — Confronto F1 Fraud(0): DistilRoBERTa vs DeBERTa

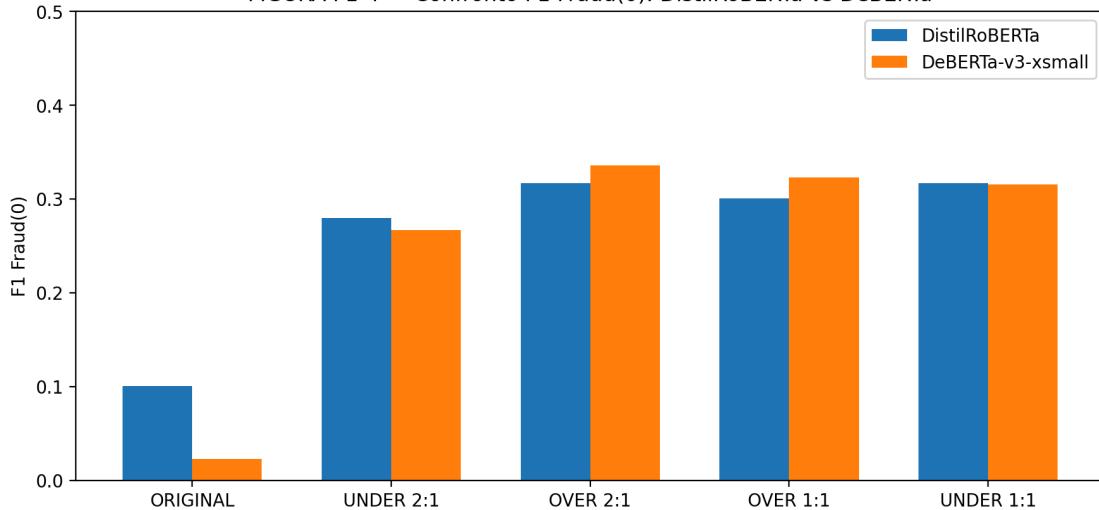
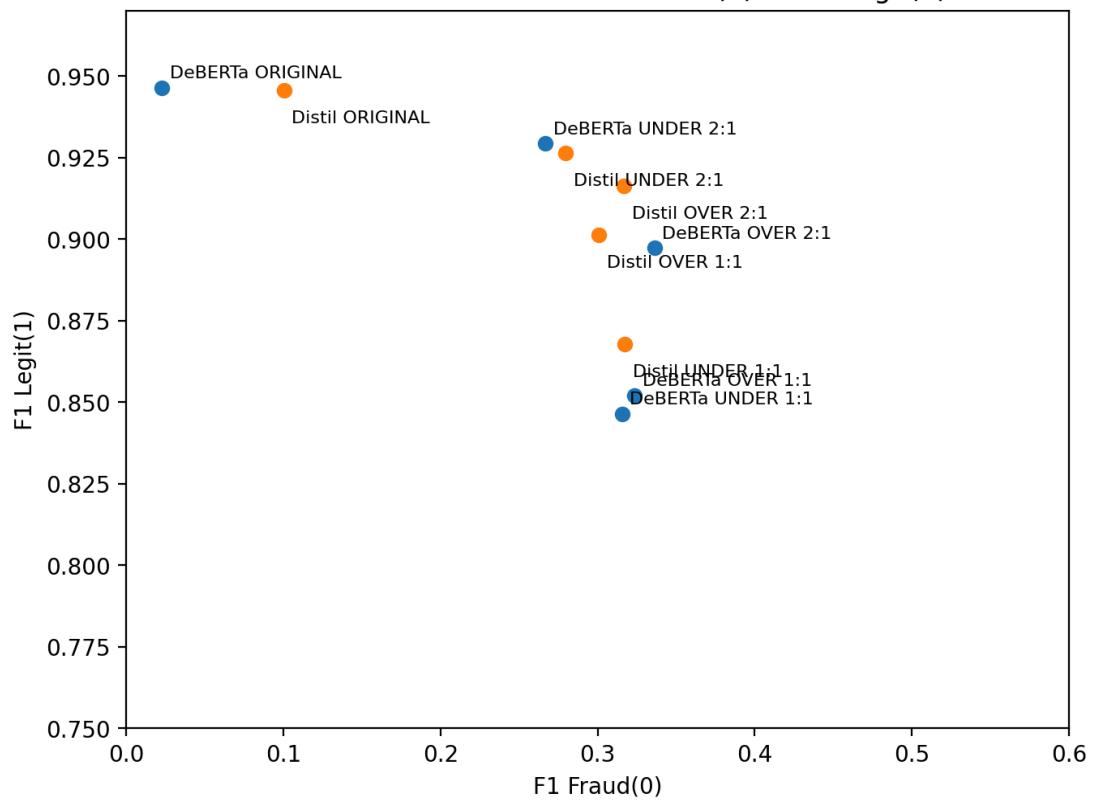


FIGURA F1-5 — Trade-off: F1 Fraud(0) vs F1 Legit(1)



DESIGN DI SISTEMA

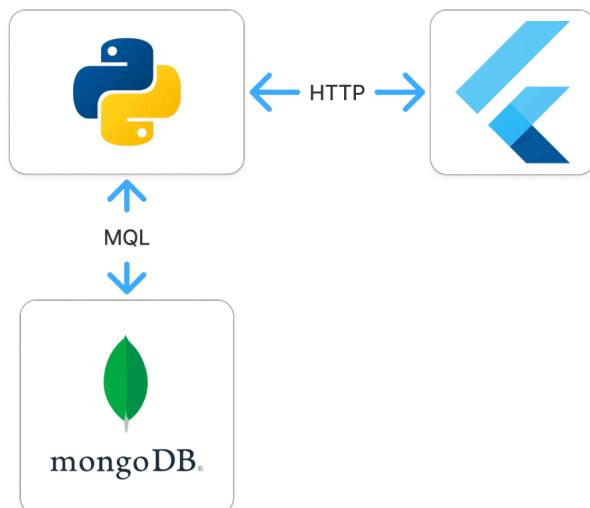


Fig. 2

Il sottosistema di Intelligenza Artificiale è implementato in Python, sfruttando gli ecosistemi **PyTorch** e **Hugging Face Transformers**.

I risultati prodotti dalla pipeline di classificazione e i metadati relativi alle strutture ricettive vengono archiviati in un database NoSQL **MongoDB**. L'accesso e l'interrogazione di tali dati sono mediati da una **REST API**, sviluppata in Python mediante il framework **FastAPI**.

Il frontend multiplattaforma è realizzato tramite **Flutter** e presenta una visualizzazione cartografica delle strutture, segmentate visivamente in funzione del grado di affidabilità delle recensioni associate.

MONGODB

MongoDB è un database NoSQL orientato ai **documenti**, progettato per gestire **dati semi-strutturati** e ad alta variabilità. La sua struttura flessibile consente di gestire efficacemente **dati eterogenei**, come punteggi di affidabilità, testi analizzati e **coordinate geografiche**, senza imporre schemi rigidi.

Nel progetto, MongoDB supporta una gestione **scalabile** ed efficiente dei dati raccolti, facilitando le operazioni di **interrogazione** necessarie alla valutazione dell'**affidabilità informativa**.

FASTAPI

FastAPI è un framework Python per lo sviluppo di **REST API** ad alte prestazioni, utilizzato come livello di **interfaccia applicativa** tra il sistema di analisi e il frontend. Il supporto nativo alla **validazione dei dati**, alle **operazioni asincrone** e all'integrazione con pipeline di **analisi automatica** lo rende adatto alla gestione di flussi informativi dinamici.

Attraverso FastAPI, l'accesso ai dati memorizzati in MongoDB avviene in modo **strutturato e sicuro**, consentendo l'esposizione dei **risultati analitici**, dei **metadati geografici** e dei **livelli di affidabilità** calcolati.

FRONTEND

L'interfaccia utente di **Trasite!** costituisce il modulo di visualizzazione della pipeline analitica e ha l'obiettivo di rendere accessibili i risultati dei modelli di classificazione e le elaborazioni statistiche prodotte dal backend. Lo sviluppo è stato realizzato tramite il framework **Flutter**, scelto per la capacità di gestire nativamente la compilazione multiplataforma e per l'efficienza nell'integrazione di servizi cartografici interattivi.

Il modulo principale dell'interfaccia è basato sulla componente cartografica. Attraverso l'elaborazione delle coordinate geografiche estratte dai dataset di Airbnb, l'applicazione posiziona ogni struttura ricettiva nello spazio urbano di riferimento. A ogni struttura è associato un indicatore cromatico derivato dal calcolo della probabilità stimata di autenticità delle recensioni.

Questa modalità di rappresentazione permette di evidenziare possibili cluster di annunci con bassi livelli di affidabilità o aree soggette a possibili campagne di recensioni non genuine.

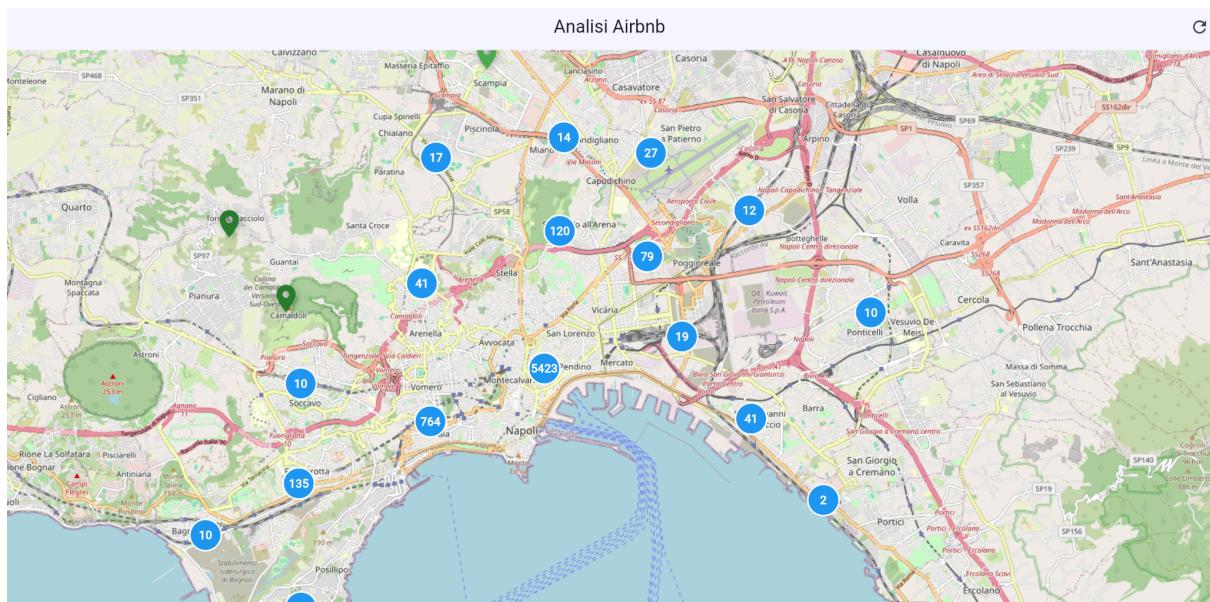
L'utente può così distinguere tra la valutazione media assegnata dalla piattaforma e l'indice di affidabilità stimato dai modelli DeBERTa e dai classificatori basati su feature linguistiche.

L'applicazione interagisce con il backend tramite richieste asincrone verso le REST API sviluppate in **FastAPI**. Per ogni singola struttura selezionata sulla mappa, il frontend interroga il database **MongoDB** per estrarre e visualizzare:

- **Indice di Affidabilità:** Un valore percentuale che esprime la confidenza del modello sulla genuinità dei feedback.
- **Ripartizione del Sentiment:** La distribuzione della polarità (positiva, negativa, neutra) calcolata sui testi analizzati.
- **Indicatori di Anomalia:** Segnalazioni specifiche relative alla presenza di pattern legati a spam o contenuti classificati dal detector come bot-like.

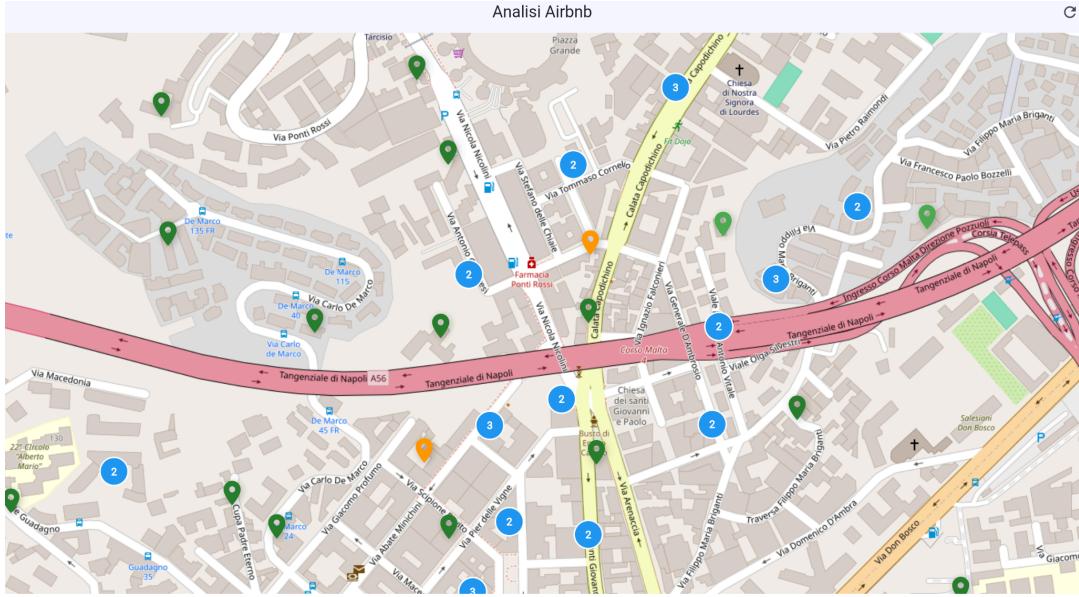
L'obiettivo tecnico dell'interfaccia è fornire una sintesi operativa di dati complessi, permettendo una consultazione immediata dei parametri di rischio reputazionale associati agli annunci degli affitti brevi.

DEMO FRONTEND



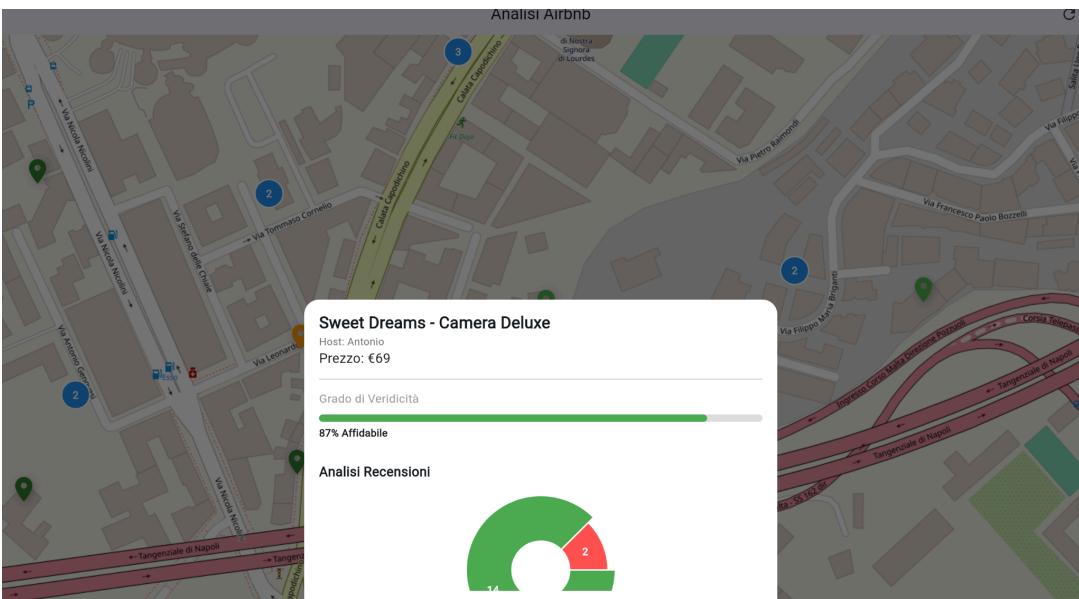
Rappresentazione geospaziale delle strutture ricettive.

L'immagine illustra l'interfaccia principale del frontend, dove i dati estratti da Inside Airbnb vengono proiettati sul territorio. Ogni marker rappresenta un B&B.



La codifica cromatica (dal verde al rosso) non indica il rating medio della piattaforma, ma il **livello di affidabilità calcolato** dalla pipeline di Trasite!

Questa visualizzazione permette di osservare rapidamente la distribuzione del rischio reputazionale in diverse aree della città, evidenziando zone potenzialmente soggette a fenomeni di spam o recensioni non genuine.



Dashboard analitica della singola struttura.

Selezionando un annuncio specifico, il sistema espone i risultati granulari dell'analisi linguistica e modellistica. La schermata mostra la sintesi del **Sentiment Analysis** (ripartizione tra polarità positiva, negativa e neutra) e i segnali di allerta derivati dai modelli di **Bot-Detection**.

CONCLUSIONI

Il progetto **Trasite!** ha mostrato risultati promettenti in uno scenario in cui i segnali tra classi risultano spesso sottili e parzialmente sovrapposti.

L'integrazione di analisi statistica multivariata, modelli Transformer e visualizzazione geospaziale ha permesso di trasformare dati testuali grezzi in informazioni strutturate e utili alla valutazione del rischio reputazionale.

Dalla fase sperimentale sono emersi tre punti fondamentali:

- 1. Rilevanza delle Feature Linguistiche:** L'analisi ha confermato che indicatori come la densità verbale, la soggettività e la complessità sintattica forniscono una baseline interpretabile per caratterizzare anomalie strutturali, sebbene la loro efficacia vari a seconda del dataset.
- 2. Prestazioni dei Modelli:** Il confronto tra l'approccio classico basato su **PCA** e quello end-to-end tramite **DeBERTa-v3** ha evidenziato prestazioni migliori nei nostri esperimenti nella cattura delle sfumature semantiche. Tuttavia, i test hanno anche sottolineato l'importanza critica della gestione dello sbilanciamento delle classi: senza tecniche di *resampling* adeguate, i modelli tendono a privilegiare la classe maggioritaria, fallendo l'obiettivo operativo di intercettare le frodi.
- 3. Utilità del Sistema Integrato:** La combinazione di un back end ad alte prestazioni (FastAPI e MongoDB) con un front end intuitivo (Flutter) ha dimostrato che la complessità della *Data Analysis* può essere mediata con successo per l'utente finale, rendendo agevolmente percepibili i livelli di affidabilità territoriale.

Nonostante i risultati positivi, il sistema presenta margini di evoluzione. Una possibile estensione riguarda l'implementazione di analisi temporali per individuare "picchi" sospetti di recensioni in brevi periodi, potenzialmente indicativi di attacchi coordinati. Inoltre, l'integrazione di tecniche di **Explainable AI (XAI)** potrebbe permettere di esplicitare all'utente i motivi specifici per cui una determinata recensione è stata classificata come sospetta, aumentando ulteriormente la trasparenza del sistema.