LA *QUESTIO DE AQUA ET TERRA*

Nuove indagini e prospettive

A cura di Alberto Casadei e Paolo Pontari

PISA
UNIVERSITY
PRESS

SAGGI E STUDI

La Questio de aqua et terra : nuove indagini e prospettive / a cura di A. Casadei e P. Pontari. - Pisa : Pisa university press, 2025. - (Saggi e studi)

855.1 (23.)
I. Casadei, Alberto (1963-) II. Pontari, Paolo 1. Alighieri, Dante. Quaestio de aqua et terra

CIP a cura del Sistema bibliotecario dell'Università di Pisa

UPI
UNIVERSITY PRESS ITALIANE

Membro Coordinamento
University Press Italiane

Pisa University Press
Polo editoriale - Centro per l'innovazione e la diffusione della cultura
Università di Pisa
Piazza Torricelli 4 - 56126 Pisa
P. IVA 00286820501 · Codice Fiscale 80003670504
Tel. +39 050 2212056 · Fax +39 050 2212945
E-mail press@unipi.it · PEC cidic@pec.unipi.it
www.pisauniversitypress.it

ISBN 979-12-5608-206-3

In copertina: Questio ... de duobus elementis aquae et terrae, Venezia 1508 (*editio princeps*). Particolare della c. B3v.

Indice

Sull'autorialità dantesca della *Questio de aqua et terra*: uno studio computazionale

Martina Leocata, Alejandro Moreo*, Fabrizio Sebastiani*, Marco Signori***

1. Introduzione

Il presente lavoro illustra i risultati di un'analisi computazionale di autorialità, o *computational authorship identification* (CAI), applicata alla *Questio de aqua et terra*, e ha lo scopo di fornire un ulteriore tassello agli studi che si propongono di determinarne l'autenticità o meno. A differenza degli approcci tipici dell'analisi filologica, basati sull'interpretazione qualitativa di elementi linguistici, contenutistici, e storico-letterari, questo studio adotta un approccio quantitativo, focalizzato sull'analisi di micro-tratti stilistici del testo, in ciò facendo propri il "paradigma evidenziale" evocato da Ginzburg (1989) e le proposte della stilometria (Savoy 2020).

La metodologia qui utilizzata si basa sull'impiego di tecniche di apprendimento automatico (*machine learning*), un settore dell'in-

* Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, IT.

** Scuola IMT Alti Studi Lucca, 55100 Lucca, IT.

formatica che sviluppa algoritmi in grado di addestrare, mediante l'uso di dati annotati ("dati di addestramento"), altri algoritmi a eseguire determinati compiti. Dovendo indagare sull'autorialità di un testo per il quale abbiamo, di fatto, un unico autore candidato, siamo di fronte, nella terminologia della CAI, a un problema di *authorship verification* (AV), i.e., un problema in cui l'algoritmo (un "classificatore binario") deve essere addestrato a riconoscere la presenza o meno della mano autoriale del candidato, e quindi a classificare il testo in oggetto in modo binario, i.e., come scritto dal candidato o meno. Il metodo qui seguito è quello tipico della AV, e consiste innanzitutto nell'assemblare un *corpus* in cui siano presenti sia testi dell'autore candidato (Dante Alighieri, nel nostro caso) che testi "di contrasto", i.e., scritti da altri autori coevi e vicini, per stile, genere testuale, e temi trattati, ai testi dell'autore candidato. I testi del *corpus* avranno una doppia funzione, ovvero (1) fungere da dati di addestramento, dai quali cioè il classificatore apprenderà, mediante un'analisi comparativa eseguita con tecniche statistiche, i micro-tratti stilistici che differenziano la mano di Dante da altre, e (2) fungere da dati di test, i.e., dati che chiederemo al classificatore già addestrato di classificare come danteschi o meno in modo da poterne misurare l'accuratezza, ovvero la capacità di predire correttamente la paternità dantesca o meno di un testo di autore incerto. In tal modo, l'accuratezza che il classificatore dimostra quando applicato a testi di autore certo sarà una misura, una volta che il classificatore sarà chiamato a pronunciarsi sull'autorialità di un testo di paternità ignota o controversa, della credibilità del suo giudizio.

Nel capitolo successivo forniamo una breve introduzione all'apprendimento automatico e alla *computational authorship identification*. Nei Capitoli §3 e §4 introduciamo il nostro lavoro sulla *Questio*, descrivendo la fase di preparazione dei dati, illustrando i criteri da noi seguiti per la loro raccolta e cura, e introducendo (i) gli elementi stilistici che analizziamo e (ii) le tecniche che adottiamo per rimediare alla inerente scarsità di testi di addestramento danteschi. Nei Capitoli §5, §6, §7 descriviamo invece la fase di addestramento e validazione del classificatore, e la sua applicazione alla determinazione di autenticità della *Questio*. Nel Capitolo §8 infine presentiamo i risultati di analisi aggiuntive, volte a identificare i fattori che

si rivelano più importanti per le determinazioni di autorialità del nostro algoritmo.

2. Apprendimento automatico e analisi computazionale di autorialità

Quando si indaga sulla paternità di un testo di autore sconosciuto o controverso, l'analisi filologica tradizionale viene condotta combinando elementi esterni al testo (e cioè le caratteristiche del contesto storico-letterario del testo in esame, il sistema di idee dell'autore candidato, le eventuali testimonianze rinvenute in altri testi, ecc.), ed elementi interni al testo (e cioè i contenuti – quindi: l'argomento trattato, le idee esposte, i riferimenti ad altre opere, ecc. – o gli specifici tratti linguistici presenti nel testo – quindi: la struttura dei periodi, l'utilizzo di espressioni linguistiche idiosincratiche, ecc.). In entrambi i casi, il carattere dell'analisi è prettamente *qualitativo*.

La CAI prevede invece un'analisi *quantitativa* dei (micro)tratti stilistici di un testo, identificati e studiati mediante strumenti computazionali. Affiancata agli strumenti di analisi filologica tradizionali, essa è dunque potenzialmente in grado di offrire nuove prospettive nell'attribuzione di testi incerti. Per limitarsi all'ambito della lingua latina, fra i lavori che utilizzano strumenti CAI si possono ricordare lo studio su Hildegard von Bingen di Kestemont et al. (2015), quello sui testi di Giulio Cesare ancora di Kestemont et al. (2016), quello sulla Lettera di Plinio il Giovane a Traiano (Tuccinardi, 2017), e quello sull'*Epistola a Cangrande* (Corbara et al., 2019; Corbara et al., 2022).

I compiti che la CAI affronta sono molteplici. Di questi, due sono importanti ai fini della nostra ricerca (il primo ne è il cuore, mentre al secondo faremo ricorso marginalmente):

- *Authorship verification* (AV): dato un testo x e un autore candidato a, determinare se a sia o meno l'autore di x;
- *Authorship attribution* (AA): dato un testo x e un insieme $\boldsymbol{A}$ di autori candidati, identificare chi, fra gli autori in $\boldsymbol{A}$, è l'autore più probabile di x.

L'ipotesi che soggiace a entrambi i compiti, e alla CAI in generale, è che ogni autore sia caratterizzato da un proprio *stiloma* (van Halteren et al., 2005), e che lasci quindi sui testi che scrive la propria *impronta digitale stilistica*, frutto di uno stile idiosincratico e distinguibile da quello altrui. Qui, il concetto di "stile" è visto come la somma di un insieme di caratteristiche linguistiche frutto della scelta arbitraria dell'utilizzatore della lingua (Crystal, 2008), e rilevabile a partire da un insieme di micro-tratti linguistici che possono essere quantificati ed estratti automaticamente.

2.1 CAI e classificazione automatica di testi

Per risolvere computazionalmente un problema di analisi di autorialità di un testo, un approccio diventato ubiquo negli ultimi due decenni è quello di inquadrare il problema come uno di classificazione automatica di testi, che può essere risolto tramite l'utilizzo di metodi di apprendimento automatico.

Nello specifico, per risolvere un compito di classificazione si addestra un classificatore **M** esponendolo a un insieme di esempi **E** ciascuno associato a una specifica classe appartenente a un insieme predefinito **L**, e si misura l'accuratezza con cui esso capacità misurabile adottando una misura di performance **P** (Mitchell, 1997). I campi di applicazione (di possibili task **T** ed esempi **E**) sono numerosi, così come il numero dei possibili modelli **M** e misure **P** adottabili. Se l'insieme di esempi consiste in un insieme di dati testuali sarà quindi un compito di classificazione automatica di testi. Nel caso dell'*authorship identification*, l'obiettivo del compito sarà l'identificazione dello specifico autore di un certo testo, ma in altri contesti le etichette possibili possono riguardare vari aspetti, ad esempio il *sentiment*, l'argomento trattato, la lingua madre dell'autore ecc.

Tra i task di classificazione, si può distinguere tra:

- *classificazione binaria*: se l'obiettivo è imparare a distinguere tra due possibili etichette, ad esempio {*spam, nonSpam*};
- *classificazione multiclasse*, se invece il set di possibili etichette contiene più di due elementi, ad esempio {*positivo, negativo, neutro*};

Nel caso dell'AI, quindi, parleremo di una classificazione di tipo binario se l'obiettivo è risolvere un compito di AV, dove i set di etichette consistono in {<autore>, *non*<autore>}, oppure di un compito di tipo multiclasse se l'obiettivo è identificare l'autore più probabile a partire da una lista di candidati, come {*Dante Alighieri*, *Giovanni Boccaccio*, *Pier della Vigna*}.

Un'altra caratteristica da considerare, comune a tutti i task di *apprendimento automatico* di tipo supervisionato, riguarda le modalità di apprendimento del modello, definite e adoperate in fase di *model building*, e quelle di valutazione delle prestazioni, condotte in fase di *model evaluation*: per ottenere risultati ottimali e affidabili, quindi, verrà diviso il *corpus* in due insiemi disgiunti, ovvero il *training set*, utile per l'addestramento, e il *test set*, necessario per valutare la sua capacità di generalizzare. Valutare il modello sugli stessi dati adoperati per l'addestramento, infatti, comporterebbe un elevato rischio di *overfitting* (Haykin, 2009). Le strategie più comuni di divisione, o *partitioning*, dei dati sono principalmente la *hold-out validation* e la *k-fold cross-validation*. La prima consiste nella semplice divisione dei dati nei due insiemi disgiunti, sottoponendo il modello a un solo ciclo di addestramento e test; la seconda, invece, consiste nel dividere l'insieme di dati in k parti, quindi addestrare il modello sulla base delle k-1 parti e valutarlo sulla k-esima, ripetendo il processo per k volte. A seconda delle risorse disponibili, temporali e computazionali, quindi, si potrà decidere il valore di k: quelli tipicamente adottati sono 5 o 10, poiché offrono un buon compromesso tra costo computazionale e affidabilità dei risultati. Tuttavia, se l'obiettivo è massimizzare la quantità di informazione estraibile dai dati, indipendentemente dal costo computazionale dell'operazione, si effettuerà una *leave-one-out validation* (LOO). Questa può essere considerata come una istanza particolare della *k-fold cross-validation* in cui il valore di k è fissato a N, cioè coincide con il numero totale di esempi. In questo modo, quindi, il modello sarà addestrato su tutti gli esempi eccetto uno, utilizzato come unico esempio per la valutazione, ripetendo il processo N volte (James et al., 2013).

Durante la fase di valutazione, come descritto sopra, è necessario adottare una certa misura di performance **P**. Nel contesto dei task di classificazione, le più comuni sono la cosiddetta *vanilla accuracy* e la F_1, calcolabili a partire da una *matrice di confusione*, mostrata in (Figura 1).

		Etichetta reale	
		P	N
Etichetta predetta	P	TP	FP
	N	FN	TN

Fig. 1 - Tabella della matrice di confusione, che distingue le decisioni del modello in base all'etichetta reale, generando gli insiemi TP, FP, FN e TN. Da queste classi si potranno calcolare diverse misure riassuntive come la precision, la recall o la vanilla accuracy.

Una volta ottenute le predizioni del modello, infatti, la matrice di confusione permetterà di distinguere tra 4 possibili risultati, ovvero i *true positives* (TP), *false positives* (FP), *false negatives* (FN) e *true negatives* (TN). In un contesto di classificazione binaria avremo una sola matrice di confusione, mentre in uno multiclasse calcoleremo $\boldsymbol{L}$ matrici di confusione, ovvero un numero di matrici pari al numero di etichette. Una volta calcolata la matrice, si possono calcolare le diverse misure di performance: la *vanilla accuracy* consiste nel rapporto tra predizioni corrette (i TP e i TN) rispetto al totale delle predizioni, espresso nella formula

$$a = \frac{TP + TN}{TP+TN+FP+FN}$$

Le misure della *precision* e della *recall*, invece, rappresentano la capacità del modello di catturare tutti gli esempi della classe positiva e vengono calcolate come

$$\pi = \frac{TP}{TP+FP}$$

$$\rho = \frac{TP}{TP+FN}$$

Nel caso della *precision*, quindi, le prestazioni saranno valutate premiando quei modelli che tendono a predire positivamente un dato di test solo quando questo appartiene effettivamente alla classe reale, registrando di conseguenza pochi falsi positivi e un maggior numero di falsi negativi; nel caso della *recall*, al contrario, verranno premiati quei modelli che riescono a catturare la maggior parte degli esempi positivi presenti nei dati, anche a costo di includere anche alcuni falsi positivi fra le predizioni.

A partire dalle misure di *precision* e *recall* si calcola il valore di F_1, che consiste nella media armonica tra le due e viene espressa nella formula

$$F_1 = 2\frac{\pi\rho}{\pi+\rho} = \frac{2TP}{2TP+FP+FN}$$

Una volta ottenute tali misure di performance, a seconda del contesto di analisi si valuterà se il modello ottenuto sia già, di per sé, ottimale, oppure se sarà necessario rivedere la fase di sviluppo del modello, allo scopo, ad esempio, di massimizzare la *recall*, quindi la capacità di catturare tutti gli esempi positivi tra i dati di training; oppure di evitare situazioni di *overfitting*, ovvero di eccessivo adattamento del modello ai dati di training, che intacca così la sua capacità di generalizzare; o di *underfitting*, quando il modello non riesce a estrarre informazione rilevante dai dati di addestramento, dimostrandosi, anche in questo caso, incapace di generalizzare.

Allo scopo di ottenere prestazioni migliori o più adatte alle specifiche esigenze di analisi, quindi, è possibile ricorrere a diverse soluzioni, che possono riguardare la fase di *data preparation* – ad esempio, aumentare il numero dei dati (*data augmentation*), operazioni di pulizia dei dati (*data cleaning*) o rimozione delle informazioni irrilevanti (*feature reduction*) – oppure di *model building* – come la strategia di *partitioning*, il modello scelto o i suoi iperparametri e la strategia per ricercarli – o, ancora, di *model evaluation*, adottando una diversa misura per valutare le prestazioni. È possibile infine che un determinato modello non si dimostri particolarmente adatto per la risoluzione di un determinato problema, e che quindi sia necessario

condurre test anche con modelli differenti. Nel caso della classificazione testuale, ad esempio, negli anni si è visto come il regressore logistico (LR), le *Support Vector Machines* (SVM) e il classificatore *Naïve Bayes* (NBC) si siano dimostrati modelli particolarmente adatti per la risoluzione di questo tipo di problema, anche a dispetto di modelli più costosi e complessi.

In conclusione, abbiamo visto come il quadro della *classificazione automatica di testi* sia particolarmente adatto ad affrontare un problema di analisi computazionale dell'autorialità, in tutte le sue possibili declinazioni. In questo contesto, infatti, l'obiettivo del task di classificazione testuale sarà determinare l'etichetta discreta da associare a uno specifico testo, che ne indicherà l'autore. Si costruirà quindi un *corpus* di addestramento che sarà composto da un insieme di testi simili dal punto di vista stilistico al documento indagato, appartenenti agli autori sospettati ma anche ad autori simili. Successivamente si effettueranno eventuali operazioni di pulizia dei testi, quindi si estrarranno le caratteristiche stilistiche di interesse tramite metodi computazionali. Una volta preparati i dati, li si somministrerà a un modello per il suo addestramento e valutazione. Ottenuto il classificatore ottimale, infine, gli si somministrerà il documento di test, ricavando la predizione sull'autorialità. Per ottenere delle informazioni rilevanti e affidabili, comunque, sarà necessaria la collaborazione tra gli esperti del campo della classificazione automatica di testi, la cui competenza sarà necessaria in fase di estrazione delle caratteristiche stilistiche e dell'addestramento del modello, e quelli del dominio a cui appartiene il testo *target*, per la valutazione delle prestazioni e soprattutto per la costruzione del *corpus* di addestramento, il quale dovrà essere quanto più omogeneo e rappresentativo, come vedremo nel dettaglio nel capitolo successivo.

3. Il *corpus* di riferimento

La composizione del *corpus* di riferimento è un passaggio fondamentale quando si cerca di risolvere un problema di CAI tramite metodi di classificazione automatica di testi: in questo contesto,

infatti, esso rappresenta la base da cui vengono estratti i dati per l'addestramento del modello di classificazione. Dati insufficienti o di scarsa qualità, perché non rappresentativi, rischiano di compromettere la qualità dei risultati, indipendentemente dalla complessità del modello adoperato o dal rigore del protocollo sperimentale adottato.

Nel caso della AV, è necessario raccogliere un numero sufficiente di testi appartenenti alla classe positiva (l'autore candidato) con caratteristiche quanto più simili al documento di test, ad esempio per genere, lingua o datazione. I documenti della classe negativa dovranno essere altrettanto simili, evitando al tempo stesso scenari di eccessivo sbilanciamento. Per la *authorship attribution*, invece, occorre selezionare un numero ragionevole di autori candidati, somministrando per ciascuno un numero adeguato di documenti (Kabala, 2020).

La fase di costruzione del *corpus*, così come quella della preparazione dei dati e all'addestramento del classificatore, è dunque un processo dinamico e iterativo. A seconda del contesto, infatti, potrebbe essere necessario ampliare il *corpus* o, al contrario, ridurre il numero di esempi per alcune classi o tipologie testuali. Attraverso una fase di sperimentazione, quindi, al fine di garantire un'analisi robusta e affidabile si cercherà di ottimizzare i vari parametri, ad esempio il numero di autori, di documenti o di *token*, ovvero di unità minime di testo (come parole o simboli). Un *corpus* ben bilanciato non solo permetterà così di ottenere migliori prestazioni in fase di validazione del modello, ma aumenterà anche l'affidabilità dei risultati sul documento di test, riducendo il rischio di *overfitting*.

Nel caso specifico della *Questio de aqua et terra*, l'indagine condotta mirava a indagare sull'autorialità di Dante (*authorship verification*). Il nostro obiettivo in questa fase è stato dunque quello di raccogliere il maggior numero possibile di testi in lingua latina scritti da Dante e da autori potenzialmente affini per stile, genere e periodo storico-letterario. Tale processo ha portato all'acquisizione di 43 testi per un totale di circa 447.000 token. Il *corpus*, in particolare, consiste in opere redatte nel periodo tra il XIII e il XIV secolo e appartenenti prevalentemente al genere del trattato o a quello delle *questiones* di natura cosmologica. A questi scritti sono stati ag-

giunti i due trattati di Dante in lingua latina, il *De vulgari eloquentia* e la *Monarchia*, per un totale di circa 33.000 token.

Una volta reperiti i testi, abbiamo eseguito operazioni di revisione e *data cleaning* per rimuovere elementi non originari o non riconducibili allo stile dell'autore, come titoli, numeri di pagina e di riga, figure, didascalie e tabelle. Inoltre, sono state applicate trasformazioni grafiche specifiche, come la sostituzione di "v" con "u" e di "j" con "i", e la chiusura di citazioni esplicite – riconducibili quindi con sicurezza ad autori diversi da quello del testo preso in esame – tra parentesi graffe ({}). Le integrazioni editoriali, volte a ricostruire il testo originale, sono state invece mantenute, eliminando le parentesi che le racchiudevano.

In fasi successive dell'analisi abbiamo ampliato il nostro *corpus*, integrando a quello appena descritto il *corpus* originariamente assemblato da Corbara et al. (2022) per lo studio sull'autorialità dell'*Epistola a Cangrande*. Questo *corpus*, dal quale abbiamo rimosso soltanto i testi senza indicazione specifica dell'autore, consiste di 285 testi per un totale di circa 856.000 token. In particolare, le opere che lo compongono sono state scritte tra il XIII e il XV secolo e appartengono principalmente al genere dell'epistola, del commento a opere letterarie e del trattato. In questo caso, non è stata effettuata alcuna operazione di *cleaning*, dal momento che i testi erano già stati curati per la realizzazione della precedente indagine.

Abbiamo infine incluso anche le due *Egloge* dantesche, che, sebbene in versi, dagli esperimenti condotti mostravano anch'esse di migliorare le prestazioni del classificatore.

Oltre alle operazioni di *cleaning* sopra citate, in fase di esecuzione della *pipeline* abbiamo sottoposto tutti i testi a normalizzazione, convertendo tutti i caratteri in minuscolo ed eliminando eventuali spazi extra. Abbiamo inoltre rimosso tutte le citazioni precedentemente marcate.

Il *corpus* finale utilizzato per addestrare e valutare il modello, quindi, è composto da 330 documenti: 16 della classe {*Dante*} e 314 della classe {*nonDante*}, per un totale di circa 2.000.000 token, di cui 41.000 provenienti da testi della classe positiva. Tale raccolta, comunque, presenta alcune criticità:

- Il numero limitato e la lunghezza variabile dei testi: in task di *classificazione automatica di testi*, infatti, il numero di documenti adoperati per l'addestramento e la valutazione del modello è generalmente molto alto e si attesta sulle decine (se non centinaia) di migliaia, tipicamente di lunghezza uniforme;
- il rischio di *topic bias* verso la classe negativa: molti dei documenti negativi, infatti, si presentano più simili, sia per temi che per genere, alla *Questio* rispetto a quelli di Dante. Tale aspetto potrebbe indurre il classificatore a classificare il documento di test come appartenente alla classe negativa;
- sbilanciamento a favore della classe negativa: il 95% del *corpus*, infatti, è costituito da documenti non danteschi. Anche in questo caso, il rischio è quindi di indurre il modello a classificare negativamente il documento di test.

Per mitigare questi problemi sono state adottate diverse contromisure. Una prima operazione è stata quella della *segmentazione*, al fine di aumentare il numero di esempi. Nello specifico, a partire dai testi interi abbiamo estratto quanti più segmenti possibile, ciascuno della lunghezza minima di 400 token, che si è rivelata essere la soglia ottimale per il nostro contesto. Per evitare che i segmenti risultassero composti da frasi incomplete, inoltre, per ogni testo abbiamo dapprima estratto le frasi, e le abbiamo quindi concatenate fino a raggiungere la lunghezza stabilita. Nel caso in cui la concatenazione non avesse consentito di raggiungere la soglia prefissata, le frasi rimanenti sarebbero state accorpate al segmento precedente. In questo modo, siamo quindi riusciti ad aumentare il numero di esempi di training, per un totale di 5.392.

Tale operazione, comunque, non è sufficiente per risolvere i diversi problemi sopra ricordati. Per questo motivo, nella fase di estrazione delle *feature* sono state impiegate ulteriori contromisure, che saranno analizzate nel dettaglio nel capitolo successivo (§4).

4. Tratti linguistici analizzati

Quando si affronta un problema di *classificazione automatica di testi* tramite metodi di *apprendimento automatico* i dati testuali non vengono somministrati al classificatore in forma "grezza". Indipendentemente dal modello adottato, infatti, sarà necessario prima sottoporli a specifiche operazioni di *feature extraction* per renderli leggibili alla macchina e utilizzabili per l'addestramento. Questo processo, infatti, trasforma i dati testuali "non strutturati" o "grezzi" in rappresentazioni vettoriali, cioè in sequenze numeriche, dove ogni numero rappresenta la frequenza – o una funzione della frequenza – di una determinata *feature*. Ogni *feature* rappresenta una determinata caratteristica dei dati: nel caso della CAI, quindi, rappresenterà la frequenza di occorrenza di una determinata caratteristica stilistica.

Modelli basati su reti neurali o *Transformers* estraggono le informazioni dai testi in maniera completamente automatizzata, calcolando i cosiddetti *embeddings*[3] durante il processo di addestramento. Per i modelli tradizionali, invece, l'estrazione delle *feature* impone delle scelte in fase di progettazione, le quali saranno influenzate dal contesto specifico e dall'obiettivo dell'analisi. Essa è quindi un'operazione delicata che richiede una lunga fase di sperimentazione per ottenere un insieme ottimale: rappresentazioni vettoriali poco informative, infatti, comprometterebbero le prestazioni del classificatore e porterebbero a risultati poco significativi.

Sebbene l'approccio basato su modelli tradizionali richieda quindi maggiore competenza di dominio, anche da un punto di vista linguistico, e maggiore impiego di risorse, allo stesso tempo permette un maggiore controllo sulle caratteristiche stilistiche da tenere in considerazione, rendendo i risultati del classificatore più umanamente comprensibili.

3. Termine che indica una rappresentazione vettoriale densa, che cattura il significato semantico e le relazioni contestuali dell'elemento che rappresenta (tipicamente le parole o i *token*).

In entrambi i casi, comunque, il processo di selezione e valutazione delle *feature* è iterativo. A mano a mano che la sperimentazione procede e le prestazioni del modello vengono valutate, è possibile esplorare nuovi tratti linguistici, modificare alcuni iperparametri o scartare *feature* che si sono rivelate poco informative.

Una volta definiti i tratti linguistici, l'estrazione automatica avviene tramite tecniche di *Natural Language Processing* (NLP), la cui complessità varia a seconda del livello linguistico che si intende catturare. Estrattori di *feature* che analizzano caratteristiche superficiali del testo, come i *character n-grams*, che si limitano a calcolare le frequenze di sequenze di caratteri di lunghezza *n*, presentano una complessità computazionale ridotta e maggiore accuratezza rispetto a metodi più complessi. Infatti, gli estrattori che mirano a catturare tratti linguistici di livello più alto, come i *parser* morfo-sintattici o semantici, richiedono un'infrastruttura più complessa: tecniche come il *Part-of-Speech (POS) Tagging*, l'estrazione delle dipendenze sintattiche o la *Named-Entity Recognition (NER)* si basano su *pipelines* articolate, che includono passaggi come la "tokenizzazione" e la lemmatizzazione, ma anche risorse aggiuntive di informazione, come ad esempio risorse lessicali annotate.

Nel contesto della nostra indagine, l'obiettivo di questa fase consisteva nell'estrarre tratti linguistici rivelatori dello stile dell'autore dei testi. Al tempo stesso, però, era necessario minimizzare il rischio di introdurre *feature* che potessero veicolare un *bias* legato al contenuto o al genere delle opere o, più in generale, a elementi non direttamente riconducibili allo stile personale dell'autore. Quest'ultimo aspetto, infatti, era particolarmente importante dato il rischio di *topic bias* che presentava il nostro *corpus* (§3).

A tale scopo, durante la sperimentazione abbiamo adottato diversi insiemi di tratti stilistici, molti ampiamente consolidati in letteratura. Per ciascun insieme, abbiamo calcolato dapprima la frequenza di ogni tratto, quindi abbiamo normalizzato la matrice successivamente ottenuta tramite l'applicazione del TF-IDF (*Term Frequency-Inverse Document Frequency*). Quest'ultimo è un metodo ereditato dall'*Information Retrieval* che assegna un peso maggiore

ai tratti frequenti in un documento ma rari nel *corpus*, riducendo così l'influenza di quelli comuni.

In particolare, abbiamo estratto i seguenti gruppi di *feature*:

- *feature* linguistiche superficiali: tra queste, i *character n-grams*, le lunghezze delle parole e le lunghezze delle frasi, queste ultime calcolate come numero di caratteri;
- *feature* basate su tecniche di *mascheramento* dei documenti, come *DV-MA e DV-EX*. Queste mascherano il contenuto tematico dei testi sostituendo i caratteri di ogni parola con degli asterischi (*), eccetto le parole di una lista specificata dal programmatore (nel nostro caso, la lista delle parole funzione[4] latine). Una volta trasformato così il testo, vengono calcolate le frequenze delle parole (Stamatatos, 2018);
- *feature* linguistiche di alto livello: tra queste, le frequenze delle sole parole funzione, i POS *n*-grams e le *n*-gram di dipendenze sintattiche. Tra i POS *grams*, inoltre, estraiamo anche un set specifico di *feature* basato sulla frequenza dei vari suffissi verbali latini, supponendo che tali tratti sintattici potessero riflettere le preferenze dell'autore per certi modi e tempi verbali;
- *feature* specifiche per il latino: in particolare, nel nostro lavoro estraiamo le quantità sillabiche, il cui calcolo è preceduto da una operazione di pre-elaborazione dedicata per l'estrazione dei pattern ritmici.

Per ogni documento, abbiamo ottenuto una rappresentazione vettoriale differente per ogni gruppo di *feature*, la cui dimensione dipendeva dalla tipologia, dalle caratteristiche del *corpus* o da entrambi i fattori. Ad esempio, nel caso delle parole funzione o dei suffissi verbali, il numero massimo di *feature* corrispondeva alla

4. L'insieme delle parole funzione racchiude tutti i termini che appartengono a classi di parole chiuse, come articoli, preposizioni o congiunzioni. Tali termini sono tipicamente adottati nel campo dell'analisi dell'autorialità in quanto *topic-agnostic*: a differenza dei termini che appartengono a classi di parole aperte, infatti, non hanno un reale contenuto semantico ma hanno la funzione di mantenere una frase coesa e coerente.

cardinalità dell'insieme di parole funzione latine o suffissi verbali latini. Nel caso delle lunghezze delle frasi, invece, la dimensione variava in funzione della distribuzione delle lunghezze nel *corpus*. Per gli *n-grams*, infine, la dimensione dipendeva sia dalla varianza testuale del *corpus* sia dalle scelte di implementazione, come il valore o l'intervallo di *n*.

La scelta di estrarre insiemi di *feature* che catturavano diversi aspetti stilistici, in contrasto con approcci che prevedono di sfruttare singoli gruppi, si è rivelata efficace. Concatenando orizzontalmente le varie rappresentazioni, infatti, in fase di sperimentazione abbiamo osservato miglioramenti nelle prestazioni del classificatore, che quindi beneficiava dei molteplici "punti di vista" forniti per ogni documento, dimostrandosi capace di catturare le informazioni più significative.

Tuttavia, allo scopo di arricchire ulteriormente le rappresentazioni vettoriali e, soprattutto, risolvere un grosso limite del nostro *corpus*, ovvero lo sbilanciamento del *corpus* a favore della classe negativa (§3), abbiamo effettuato una operazione di *oversampling*. Questa è una tecnica di *data augmentation* comunemente adottata in ambito *data science* per la risoluzione di problemi binari in contesti in cui i dati sono sbilanciati. Nello specifico, abbiamo adoperato il *Distributional Random Oversampling* (DRO), un metodo sviluppato da Moreo et al. (2016), progettato appositamente per i dati testuali. Il funzionamento del DRO, infatti, è basato sull'ipotesi distribuzionale, secondo cui il significato di una parola è determinato dai contesti in cui essa compare (Harris, 1954). Partendo da questa premessa, quindi, il metodo genera nuovi esempi sintetici della classe minoritaria, mantenendo le stesse proprietà semantiche dei documenti originali. In particolare, per ogni documento della classe minoritaria il metodo analizza come le parole del documento si distribuiscono nell'intero *corpus* e, sulla base di questa, tramite una funzione probabilistica genera automaticamente nuove combinazioni di parole. Questa funzione viene utilizzata più volte per ogni documento, producendo così diverse varianti che preservano le caratteristiche semantiche dell'originale. In questo modo, quindi, il metodo viene applicato in maniera uniforme a tutti i documenti della classe minoritaria, generando lo stesso numero di varianti per ciascuno di essi, fino a raggiungere il livello di ribilanciamento desiderato.

Applicando il DRO, quindi, abbiamo ottenuto un gran numero di nuove rappresentazioni vettoriali di documenti sintetici positivi che racchiudono le stesse caratteristiche distribuzionali di quelli originali. Sebbene, data la natura probabilistica del metodo, l'introduzione di elementi di casualità nella generazione dei nuovi documenti abbia richiesto più prove sperimentali per valutare in maniera rigorosa le prestazioni del modello, siamo così riusciti a bilanciare il corpus senza alterarne la coerenza distribuzionale. Allo stesso tempo, inoltre, abbiamo ottenuto una rappresentazione vettoriale più informativa dei documenti originali sfruttando l'altro aspetto innovativo del metodo, che non si limita a generare nuovi esempi ma arricchisce anche i vettori originali aumentandone la dimensionalità.

Una volta definiti ed estratti i tratti stilistici di interesse, abbiamo testato diverse strategie di riduzione del numero delle *feature*, allo scopo di diminuire la complessità delle rappresentazioni vettoriali e scartare *feature*, o insiemi di *feature*, non informative. Al tal fine, quindi, abbiamo adoperato una *filter strategy*. In particolare, in fase di estrazione di ogni insieme, abbiamo ordinato le *feature* estratte in base alla loro informatività rispetto all'etichetta *target* tramite il χ^2, una misura statistica[5]. Successivamente, abbiamo condotto degli esperimenti in cui abbiamo fissato una determinata soglia percentuale per selezionare il sottoinsieme più informativo: nello specifico, testiamo valori dal 25% al 75%, con incrementi del 5% in ogni iterazione.

Parallelamente, abbiamo condotto esperimenti mirati a identificare ed eventualmente scartare interi insiemi di *feature*. A tal fine, abbiamo adottato una strategia di *ablation* su alcuni documenti di addestramento particolarmente difficili da classificare correttamente (Savoy, 2020). Nello specifico, dato un sovrainsieme di *n* insiemi di *feature*, il nostro metodo consisteva nel misurare le prestazioni del modello somministrando l'intero sovrainsieme. Successivamente, confrontava le prestazioni rimuovendo uno

5. Il test del χ^2 misura l'indipendenza tra una *feature* e la classe di riferimento, confrontando la frequenza osservata delle occorrenze con la frequenza attesa sotto l'ipotesi di indipendenza.

alla volta ciascun gruppo di *feature*, creando così sovrainsiemi di (*n-1*) gruppi. Se uno dei sovrainsiemi ottenuto registrava prestazioni migliori, procedeva con i sovrainsiemi di (*n-2*) *feature*, e così fino ad arrivare a (*n=1*), altrimenti si fermava e restituiva l'insieme di *feature* con le prestazioni migliori.

Una volta effettuati diversi test preliminari, abbiamo deciso di non applicare alcuna *filter strategy* per il modello finale, non avendo osservato miglioramenti nelle prestazioni. Se l'insieme di *feature* si rivela informativo, infatti, anche le sue componenti meno importanti saranno utili per la classificazione. Gli esperimenti di *ablation*, invece, ci hanno portato a scartare diversi insiemi di tratti stilistici. I gruppi di *feature* mantenuti per la classificazione finale sono riportati in Tabella 1.

Tab. 1 - Gruppi di feature e numero di feature estratti per la classificazione finale.

Gruppo di *feature*	Numero di *feature*
Character n-grams (unigrammi, bigrammi, trigrammi)	8299
Lunghezza delle frasi	998
Lunghezza delle parole	18
Parole funzione	74
POS n-grams (unigrammi, bigrammi, trigrammi)	3333
Totale	**12722**

Al termine della fase di preparazione dei dati, dunque, otteniamo un insieme di 6.588 esempi di addestramento, di cui 1.317 appartenenti alla classe positiva {*Dante*} e 5.271 alla classe negativa {*nonDante*}, ciascuno descritto da un totale di 28.898 *feature*. In Tabella 2 riassumiamo la composizione del *corpus* al termine delle varie fasi.

Tab. 2 - Composizione del *corpus* al termine delle fasi di assemblaggio, di data augmentation, e di oversampling.

	Corpus iniziale	Dopo segmentazione	Dopo oversampling
Dante	16	121	1317
nonDante	314	5271	5271
Totale	330	5392	6588

Una volta preparati i dati, siamo quindi passati alla fase di addestramento del modello, che descriveremo nel capitolo successivo.

5. Il protocollo sperimentale

Prima di procedere con l'addestramento effettivo del modello, un passaggio fondamentale è stabilire il protocollo sperimentale da seguire, allo scopo di ottenere risultati quanto più affidabili e robusti. A seconda del contesto, infatti, possono essere adottate diverse strategie di partizionamento dei dati, metriche di valutazione, modelli da testare e iperparametri da ricercare. Tale processo, comunque, non va inteso come sequenziale ma, anche in questo caso, iterativo, caratterizzato cioè da esperimenti condotti su protocolli sempre più complessi e specifici, adattati all'insieme di dati utilizzato e all'obiettivo di analisi.

Durante i primi esperimenti, abbiamo adottato delle strategie di *partitioning* piuttosto semplici, ovvero la *hold-out validation* e la 5-*fold cross-validation*, fino ad arrivare, in fasi avanzate, alla *LOO validation* (§2). Nelle prime fasi, inoltre, impieghiamo diversi modelli: abbiamo testato sia modelli comunemente impiegati nella classificazione testuale per le loro buone prestazioni, tra cui SVM con *kernel* lineare e la regressione logistica, ma anche un modello più complesso, ovvero *AdaBoost*, appartenente alla categoria dei modelli di tipo *boosting*. Nello specifico, esso è un meta-classificatore che addestra iterativamente più classificatori base. Questi vengono addestrati sugli stessi dati, quindi a ogni iterazione vengono aumentati i pesi delle istanze che si rivelano più difficili da classificare, in modo che i classificatori successivi si concentrino maggiormente su

queste ultime. Successivamente, il meta-classificatore combina le prestazioni di ogni classificatore pesandole in base all'accuratezza registrata, quindi produce la predizione finale (Freund e Schapire, 1995). I classificatori base sono tipicamente dei *weak learners*, ovvero dei classificatori le cui prestazioni sono di poco superiori a classificatori che predicono in maniera casuale (*random classifiers*). Nel nostro caso, invece, abbiamo testato un modello che adotta come classificatori base dei regressori logistici.

Una volta definiti i modelli, viene impostata una strategia per cercare gli iperparametri ottimali, necessaria per adattare il modello al contesto specifico, massimizzando così la capacità di generalizzare e riducendo il rischio di *overfitting* o *underfitting*. A tale scopo, si possono adottare diverse strategie di ricerca in base alle risorse, temporali e computazionali, disponibili.

Nel nostro caso, abbiamo adottato la cosiddetta strategia *grid search*, una strategia di tipo esaustivo che esplora tutte le combinazioni di iperparametri in uno spazio di ricerca definito. In particolare, per tutti i modelli testati ottimizziamo il valore dell'iperparametro *C* (dei modelli base nel caso di *Adaboost*). Per il modello di *boosting*, inoltre, ricerchiamo anche il numero ottimale dei classificatori base.

Come metrica di valutazione, infine, adottiamo misure comunemente adottate in ambito classificazione, ovvero la semplice *vanilla accuracy* durante la ricerca degli iperparametri e la F_1 in fase di valutazione, descritte in (§2). Di quest'ultima, inoltre, adottiamo una variante più raffinata, ovvero la *soft* F_1: questa, infatti, non calcola la matrice di confusione sulla base delle predizioni del modello ma in base alle probabilità *a posteriori*. In altre parole, dato un esempio di addestramento, assegna a ogni classe della matrice di confusione la massa di probabilità restituita dal classificatore per quella classe.

Dopo vari esperimenti preliminari, il modello da noi selezionato è stato il regressore logistico, che si è rivelato particolarmente adatto al contesto di analisi. Pur nella sua semplicità concettuale, infatti, tale modello, in ambito di classificazione automatica di testi è particolarmente vantaggioso in quanto, oltre alle buone prestazioni registrate, combina efficienza computazionale ed interpretabilità dei risultati. In quanto classificatore lineare, infatti, il regressore logistico risulta computazionalmente poco costoso, poiché opera attraverso trasformazioni matematiche relativamen-

te semplici dei dati di input. Esso, inoltre, è vantaggioso anche a livello interpretativo: rispetto ai modelli *black box*, come le reti neurali profonde, consente infatti di comprendere meglio il processo decisionale attraverso l'analisi dei coefficienti assegnati alle caratteristiche testuali. In questo modo, infatti, si possono analizzare quali caratteristiche testuali abbiano maggiormente influenzato le decisioni del modello, ottenendo così informazioni utili per l'interpretazione filologica.

Un ulteriore punto di forza del regressore logistico è la sua capacità di fornire, oltre alla classe predetta, anche un punteggio di confidenza (*confidence score*). Tale punteggio rappresenta una probabilità a posteriori *ben calibrata*, ovvero una stima affidabile della probabilità che un testo appartenga a una determinata classe.

Il funzionamento del regressore logistico si articola in due fasi: dato un vettore di input, nella prima fase combina in modo lineare le *feature*, pesandole secondo l'importanza appresa durante la fase di addestramento; nella seconda, applica la funzione logistica alla combinazione lineare ottenuta, trasformando il risultato in una probabilità compresa tra 0 e 1 (James et al, 2013). Tale trasformazione, pur mantenendo la linearità del modello nella sua struttura di base, genera una curva caratteristica a «forma di S» quando visualizzata in uno spazio geometrico, come visualizzabile in Figura 2.

Una volta scelto il modello finale abbiamo adottato la *LOO validation* ed effettuato numerosi test con svariate configurazioni allo scopo di trovare quello ottimale. Dati i nostri 330 testi originali, quindi, abbiamo eseguito 330 cicli di addestramento su 329 di essi e test sul 330º lasciato fuori, eseguendo l'intera *pipeline* per ogni iterazione (quindi l'estrazione dei segmenti, il calcolo dei vettori, il ribilanciamento dei dati e la ricerca degli iperparametri dell'algoritmo di apprendimento ottimali). Abbiamo quindi esplorato diverse configurazioni, tra cui la lunghezza minima dei segmenti, il valore ottimale di *n* nei gruppi di *feature*, e la percentuale di esempi positivi da generare tramite DRO (§3) (§4).

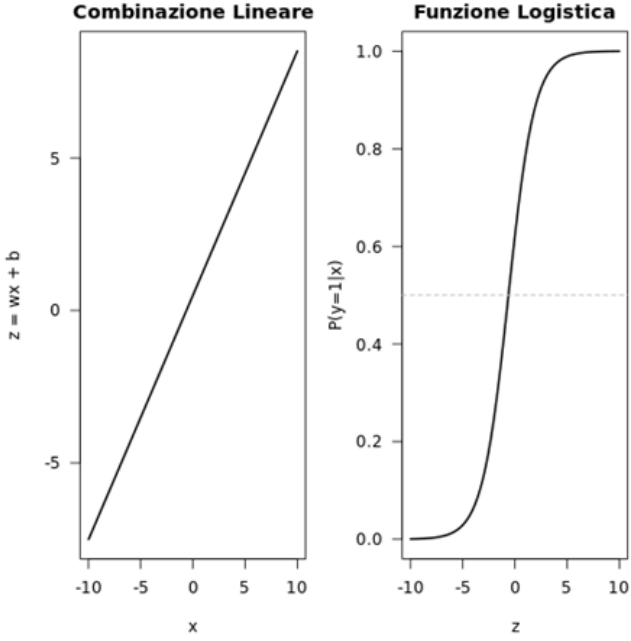

Fig. 2 - Visualizzazione geometrica della modello di regressione logistica: la prima operazione consiste nel combinare linearmente le feature, la seconda applica la funzione logistica..

Una volta ottenute le predizioni su tutti i documenti, misuriamo il punteggio di F_1 e *soft*-F_1. Nel capitolo successivo riportiamo i risultati della configurazione più accurata, che adotteremo quindi per il test finale sulla *Questio*.

6. Verifica di autorialità

Una volta effettuati numerosi esperimenti con diverse configurazioni, il migliore da noi trovato consisteva in un modello di regressione logistica, addestrato su dati estratti a partire da 330 testi di autorialità certa. Su questi veniva applicata una *pipeline* composta da una fase di *segmentazione*, di estrazione di 5 gruppi di tratti stilistici e di *oversampling* della classe minoritaria, come descritto nei capitoli precedenti (§3) (§4).

Per valutare l'affidabilità del modello, abbiamo eseguito una strategia di *LOO*, misurando le metriche di F_1 e *soft*-F_1. I risultati ottenuti hanno dato conferma della robustezza di questa configurazione: effettuando 330 test, infatti, otteniamo un valore di F_1 pari a 0,970 (risultante dall'aver classificato correttamente 329 testi su 330) e uno di *soft*-F_1 pari a 0,900. Il sistema, infatti, predice correttamente tutti i 16 documenti di Dante, commettendo un solo errore tra gli esempi negativi, e cioè l'*Epistola* XXIII di Giovanni Boccaccio, erroneamente classificata come dantesca.

Analizzando le probabilità a posteriori, inoltre, osserviamo che la stragrande maggioranza delle predizioni è supportata da confidenze superiori a 0,90, indice ulteriore della robustezza del sistema. In particolare, solo in 8 casi il classificatore mostra livelli di confidenza più bassi, come riportato in Tabella 3.

Tab. 3 - Testi più difficili da classificare: di questi solo il primo è classificato erroneamente, mentre gli altri, pur essendo classificati correttamente, riportano gradi di confidenza più bassi rispetto a tutti gli altri documenti del *corpus*.

Autore	Testo	Predizione	Confidenza
Giovanni Boccaccio	*Epistola* XXIII	Dante	0,030
Giovanni Boccaccio	*Epistola* XVI	nonDante	0,575
Giovanni Boccaccio	*Epistola* XII	nonDante	0,659
Giovanni Boccaccio	*Epistola* VI	nonDante	0,683
Giovanni Boccaccio	*Epistola* XIV	nonDante	0,685
Giovanni Boccaccio	*Epistola* XVII	nonDante	0,734
Dante Alighieri	*Monarchia*	Dante	0,761
Giovanni Boccaccio	*Epistola* I	nonDante	0,837
Giovanni Boccaccio	*Epistola* II	nonDante	0,893

Dati i risultati solidi ottenuti in fase di validazione, abbiamo quindi impiegato il sistema per il test finale sulla *Questio*. La classe

assegnata è stata *Dante*, con una confidenza estremamente alta, pari a 0,999999967. In altre parole, secondo il nostro classificatore, considerati i dati di addestramento, ci sono 33 possibilità su un miliardo che la *Questio* non sia dantesca. Tale risultato è particolarmente significativo se si considera il rischio di *bias* a favore della classe negativa introdotto dal *corpus* di addestramento, come evidenziato in (§3).

Per verificare se il sistema potesse indicare un altro autore del *corpus* come potenziale autore della *Questio*, abbiamo successivamente effettuato un test di *authorship attribution*. Tuttavia, a differenza del task di AV, non ricerchiamo il sistema di AA migliore ma adottiamo la stessa configurazione utilizzata nel task principale, a eccezione del DRO, non adoperato in quanto, essendo un contesto multiclasse, il grado di sbilanciamento era meno marcato.

Anche per il sistema di AA, comunque, abbiamo effettuato prima un test di tipo *LOO*, per verificare l'affidabilità del sistema. In questo caso, però, consideriamo solo autori di cui si dispone di almeno 2 testi, così da garantire per ogni iterazione almeno un documento positivo per autore tra i dati di addestramento. Effettuiamo, quindi, 307 test su documenti appartenenti a 15 differenti autori, ottenendo anche in questo caso delle buone prestazioni: il sistema, infatti, predice correttamente 285 documenti, con una F_1 pari a 0,628.

Tale risultato, infatti, va rapportato con quelli che otterrebbero dei classificatori *baseline*, ovvero modelli di riferimento usati per valutare la bontà di un sistema. Nel nostro caso, data la distribuzione degli autori nel *corpus*, un classificatore casuale raggiungerebbe un'accuratezza media di circa il 0,31 e un F_1 inferiore a 0,1, mentre un classificatore *dummy*, ovvero che assegna sempre la classe più frequente, otterrebbe un'accuratezza dello 0,475 (in quanto 146 dei 309 documenti appartengono a Pier della Vigna) e un F_1 molto basso, anche in questo caso inferiore a 0,1. Il sistema proposto, quindi, registra delle prestazioni nettamente superiori ai *baseline* considerati dimostrando così la sua robustezza. Nella seguente figura (Figura 3) riportiamo la matrice di confusione.

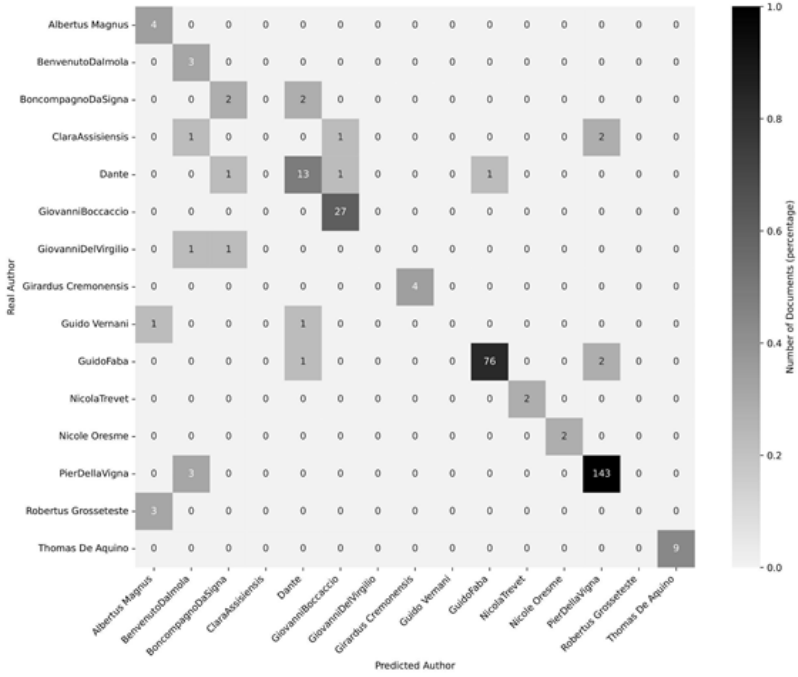

Fig. 3 - Matrice di confusione ottenuta dopo il test di leave-one-out del sistema di AA.

Una volta misurata l'affidabilità del sistema, lo abbiamo quindi testato sulla *Questio*, addestrando il classificatore su tutti i testi del *corpus*. Anche in questo caso la classe predetta è stata {*Dante*}, con una confidenza dello 0,737. La restante massa di probabilità, pari a 0,263, è invece distribuita tra i rimanenti 37 autori, tra cui quelli con confidenza maggiore risultano essere Antonio Pelacani da Parma, con una confidenza pari a 0,067 (pari quindi a quella che otterrebbe con un classificatore casuale), e Pietro Alighieri, pari a 0,049.

Nel capitolo successivo presenteremo ulteriori analisi per approfondire i fattori che hanno contribuito al risultato del sistema di AV, sia dal punto di vista implementativo che dei dati utilizzati.

7. Esperimenti ulteriori

Una volta testato il nostro sistema sulla *Questio*, abbiamo condotto ulteriori analisi per identificare i fattori che hanno contribuito al risultato ottenuto. In particolare, ci siamo focalizzati sia sugli aspetti implementativi, come la scelta dei gruppi di *feature* e l'impatto del processo di *data augmentation*, sia sulle peculiarità dei dati, come la loro distribuzione, la rappresentatività delle opere incluse nel *corpus* e il grado si similarità tra il testo target e quelli di addestramento.

Il primo fattore indagato è stato l'impatto di ciascun insieme di *feature* sulle decisioni del modello. A tale scopo, abbiamo adottato la strategia dell'*ablation*, già adoperata in fase di *feature reduction* (§4), testando solo i sovrainsiemi composti da *(n-1)* insiemi di *feature*: in questo contesto, infatti, l'obiettivo era misurare le prestazioni del modello rimuovendo, uno alla volta, un gruppo di *feature* o non operando il ribilanciamento dei dati tramite DRO. Abbiamo quindi eseguito sei test di *LOO*, ottenendo i risultati riportati in Tabella 4.

Tab. 4 - Risultati dei test di LOO dei vari sistemi ottenuti tramite ablation.

Setup	FP (su 314 testi)	FN (su 16 testi)	F_1	soft F_1
Completo	1	0	0,970	0,900
Senza DRO	0	12	0,400	0,480
Senza Character n-grams	5	1	0,833	0,834
Senza lunghezze delle frasi	5	0	0,865	0,810
Senza lunghezze delle parole	4	1	0,857	0,834
Senza parole funzione	6	0	0,842	0,822
Senza POS n-grams	1	11	0,455	0,431

Come è possibile notare dalla Tabella 4, i risultati ottenuti indicano che la configurazione che non adotta il DRO, sebbene raggiunga la massima precisione registrando 0 FP, risente del forte sbilanciamento dei dati e fallisce la corretta classificazione di 12 dei 16 documenti danteschi. D'altra parte, il sistema privo delle *feature* basate sulle lunghezze delle parole o sulle parole funzione eguagliano la configurazione completa in termini di FN, ma perdono comunque informazione e non riescono a classificare correttamente alcuni documenti non danteschi. I peggiori risultati si ottengono rimuovendo il DRO o le *feature* dei *character n-grams*: infatti, entrambi registrano valori di F_1 e *soft*-F_1 inferiori a 0,5, segnando un drastico calo rispetto alle altre configurazioni, che invece superano lo 0,8.

La strategia di *oversampling* e il gruppo di *character n-grams*, dunque, risultano essere, dal punto di vista implementativo, i due fattori determinanti per le buone prestazioni del sistema. La prima, infatti, contribuisce a mitigare gli effetti dello sbilanciamento del *corpus*, diminuendo il rischio che il modello sviluppi un *bias* verso la classe negativa; la seconda, d'altro canto, si rivela particolarmente efficace nel catturare elementi stilistici e strutturali tipici dei testi danteschi.

L'importanza dei *character n-grams* è ulteriormente confermata dal test condotto sulla *Questio* utilizzando la configurazione priva di tale gruppo: mentre tutti gli altri classificano il testo come positivo con confidenze superiori allo 0,99 (a eccezione di quello senza DRO, che registra una confidenza dello 0,83), questo lo classifica come non dantesco, con una confidenza intorno allo 0,66. Tale risultato, quindi, evidenzia il ruolo cruciale di questo gruppo di *feature* nell'attribuzione della *Questio* come opera dantesca.

Un altro aspetto indagato in questa fase riguarda la composizione del *corpus*. In particolare, ci siamo chiesti quali rappresentazioni vettoriali fossero più simili alla *Questio*, ipotizzando che i testi più simili potessero essere anche i più influenti nella classificazione finale. A tal fine, quindi, abbiamo calcolato la similarità del coseno tra i vettori estratti dai testi, evitando l'applicazione del DRO per escludere le dimensioni latenti generate da tale tecnica. Dai risultati emerge che il testo più simile alla *Questio* è la *Monarchia* di Dante, seguito da altri trattati come le *Questiones* di Nicola Oresme e il commento alla *Sfera* di Sacrobosco, attribuito a Michele Scoto. Il secon-

do testo di Dante più simile risulta essere, prevedibilmente, l'altro trattato dantesco del *corpus*, ovvero il *De Vulgari Eloquentia*, mentre quella meno simile è l'*Epistola* IV che, insieme ad altre epistole (*Epistolae* VIII, II, X, IX), risultano perfino meno simili di un'opera in versi come l'*Egloga* II. I testi meno simili in assoluto appartengono invece a Guido Faba, di cui il nostro *corpus* era ricco di epistole. In generale, quindi, i testi appartenenti a questa tipologia testuale risultano essere i meno simili alla *Questio*, suggerendo un ruolo meno rilevante nella classificazione.

Abbiamo poi eseguito test aggiuntivi per esplorare il legame tra la *Monarchia* e la *Questio*: in questo caso, abbiamo ipoteticamente etichettato la *Questio* come dantesca ed effettuato un test di *LOO* con la configurazione in cui non si applica DRO, che etichettava la *Monarchia* come non dantesca. I risultati ottenuti mostrano un generale leggero miglioramento delle prestazioni, con un punteggio F_1 pari a 0,609 (rispetto allo 0,400 ottenuto senza la *Questio*). Il sistema, infatti, anche in questo caso classifica correttamente tutti i documenti negativi, ma migliora le prestazioni sui testi danteschi. Un risultato significativo è quello ottenuto proprio sul *Monarchia*, che nella configurazione senza DRO viene classificato come non dantesco con una confidenza dello 0,81 e che anche in quella completa risulta essere particolarmente difficile da classificare, se comparata al resto dei documenti, in quanto viene classificato come positivo con una confidenza intorno allo 0,76 (vedi Tabella 4). Aggiungendo la *Questio* tra i documenti danteschi, infatti, la classificazione della *Monarchia* diventa positiva con una impressionante confidenza di 0,999. Un tale livello di similarità rafforza dunque l'ipotesi di attribuzione dantesca della *Questio* e sottolinea il ruolo centrale della *Monarchia* nella classificazione.

La presenza della *Monarchia* nel *corpus* di addestramento, comunque, non risulta essere determinante nella classificazione: rimuovendo tale testo e sottoponendo nuovamente la *Questio* al sistema, la classificazione resta comunque positiva, con una confidenza di circa 0,98. Al contrario, mantenendo il solo *Monarchia* tra i testi danteschi, la *Questio* viene classificata come non dantesca, con una confidenza di 0,79. Questo risultato evidenzia come il *De Vulgari Eloquentia*, insieme alle epistole e alle egloghe, sebbene quest'ultime meno informative rispetto ai trattati, fornisca comunque un contri-

buto decisivo alla classificazione positiva e dimostra che tali testi siano necessari e sufficienti per l'attribuzione dantesca della *Questio*.

I risultati ottenuti dimostrano dunque come non tutti i fattori che caratterizzano il nostro sistema abbiano avuto lo stesso peso nel determinare la classificazione finale. Da un punto di vista implementativo, infatti, l'adozione di un insieme di *feature* consolidato in letteratura, come i *character n-grams*, insieme a un metodo innovativo di *oversampling* come il DRO, si sono rivelati elementi cruciali per le buone performance del classificatore. Allo stesso tempo, l'analisi della composizione del *corpus* ha evidenziato come non tutti i testi siano stati ugualmente discriminanti: alcuni testi, come la *Monarchia*, hanno rivelato uno stretto legame con il documento di test, mentre intere tipologie testuali, come quella delle epistole, si sono dimostrate poco informative per il nostro specifico contesto.

8. Conclusioni

L'evidenza stilometrica emersa dal nostro studio suggerisce fortemente l'attribuzione della *Questio de Aqua et Terra* a Dante Alighieri. Questa conclusione si basa sull'alta confidenza del nostro sistema di *AV*, la cui affidabilità nel distinguere tra testi danteschi e non danteschi è stata dimostrata tramite test di *LOO*. Sebbene il *corpus* presenti potenziali *bias*, sia di natura distribuzionale, con una predominanza di testi appartenenti alla classe negativa, sia tipologica, data la presenza di molti testi negativi appartenenti al genere delle *questiones* medievali, quindi stilisticamente e tematicamente vicini alla *Questio*, il nostro approccio ha mostrato la capacità di mitigare tali rischi. Questo risultato è stato possibile grazie alla selezione di *feature* dimostratesi indipendenti dal contenuto e dal genere, nonché al ribilanciamento dei dati tramite il *Distributional Random Oversampling* (DRO).

Tuttavia, l'analisi qui presentata va considerata preliminare: sebbene i risultati ottenuti offrano un contributo significativo al dibattito attributivo, infatti, ulteriori indagini potrebbero confermare o rivedere le nostre conclusioni. Uno sviluppo futuro potrebbe prevedere la creazione di un nuovo *corpus*, includendo testi individuati come ancora più vicini alla *Questio* per stile e contenuto grazie

ai progressi nella ricerca filologica. Tali indagini potrebbero fornire nuovi indizi sulla collocazione diacronica dell'opera o sulla possibilità di eventuali manipolazioni del testo originale.

Nonostante il dibattito sulla *Questio* non possa essere considerato chiuso, dunque, il presente studio dimostra come l'integrazione tra competenze filologiche e strumenti computazionali possa aprire nuove prospettive di ricerca nell'ambito dell'attribuzione autoriale di testi medievali.

Postilla: Verifica dell'autorialità dell'Epistola a Cangrande e dell'Epistola a Enrico VII

Il recente lavoro di Corbara et al. (2022) sottopone al vaglio della *computational authorship verification* due testi la cui natura dantesca o meno è stata, in misura certamente diversa, oggetto di indagine, e cioè l'*Epistola a Cangrande* (qui *Epistola XIII*) e *l'Epistola a Enrico VII* (qui *Epistola XIV)*, entrambe decretate non dantesche dall'*authorship verifier* di Corbara et al. Abbiamo ritenuto interessante sottoporre i medesimi testi anche all'*authorship verifier* sviluppato nel presente lavoro, forti del fatto che entrambi gli algoritmi sono addestrati ad affrontare la medesima classificazione binaria (*Dante* vs. *nonDante*) e la medesima lingua (il latino medievale). Le differenze principali fra l'algoritmo di Corbara et al. (2022) e quello sviluppato nel presente lavoro risiedono

- Nei dati di addestramento usati: il nostro sistema è stato addestrato e testato su un *soprainsieme* dei dati usati per il sistema di Corbara et al., e che comprende, oltre ai testi assemblati dagli autori citati, 46 fra trattati scientifici e *questiones*, e le due *Egloge* dantesche. In entrambi, i medesimi testi sono stati poi usati in fase di validazione sperimentale.
- Nelle *feature* utilizzate: a differenza del sistema di Corbara et al., il presente sistema non usa né gli *n-grams* di parole né i suffissi verbali, ma usa invece i POS *n-grams*. Inoltre, il presente sistema non adotta, a differenza di Corbara et al., alcuna strategia di riduzione delle *feature*.

- Nel metodo di *oversampling*: mentre Corbara et al. non usano alcun metodo di *oversampling*, il nostro algoritmo è basato sul DRO, il quale, oltre a meglio bilanciare i dati di addestramento, arricchisce le rappresentazioni vettoriali, aumentandone quindi la dimensionalità.

Abbiamo quindi sottoposto al nostro verificatore i due testi in oggetto, suddividendo, come nel lavoro di Corbara et al. (2022), l'*Epistola XIII* in due parti distinte (paragrafi 1-13 e 14-90, rispettivamente), essendo queste ultime radicalmente differenti per stile e intenzione. Oltre ad aver separatamente sottoposto al verificatore le due parti dell'*Epistola XIII*, gli abbiamo anche successivamente sottoposto il testo nella sua interezza.

In tutti i quattro i casi, il verificatore *nega l'autorialità dantesca dei testi in* oggetto (come indicato in Tabella 5), in ciò confermando le conclusioni di Corbara et al.

Tab. 5 - Confronto tra le determinazioni del sistema sviluppato da Corbara (I) e del nostro sistema (II) sull'autenticità dell'Epistola XIII e dell'Epistola XIV.

Testo	Predizione (I)	Confidenza (I)	Predizione (II)	Confidenza (II)
Epistola XIII	-	-	nonDante	0,999
Epistola XIII	nonDante	0,633	nonDante	0,964
Epistola XIII	nonDante	0,978	nonDante	0,999
Epistola XIV	nonDante	0,974	nonDante	0,999

Il fatto rilevante di questo studio è che le conclusioni di Corbara et al. (2022) ne escono non solo confermate, ma anche *rafforzate*, dato che

1. Il nostro verificatore risulta essere ancora più accurato di quello di Corbara et al., dato che nella fase di validazione realizzata con il metodo *leave-one-out* ottiene un'accuratezza sostanzialmente più elevata (F_1=0,970) di quella da loro otte-

nuta (F_1=0,500 e F_1=0,857, rispettivamente, per i due verificatori utilizzati sulle due parti dell'*Epistola XIII*).

2. Il test di verifica dell'accuratezza che abbiamo realizzato è ancora più probante di quello realizzato da Corbara et al., dato che è stato condotto (come evidenziato sopra) su un soprainsieme del *corpus* sul quale è stato condotto il loro test. Assieme al punto 1., ciò suggerisce che le conclusioni tratte dal nostro verificatore sono più *credibili* di quelle tratte dal verificatore di Corbara et al.
3. Come indicato in Tabella 5, il nostro verificatore ha un livello di confidenza ancora maggiore nelle proprie conclusioni rispetto all'analogo sistema di Corbara et al.; sia per l'*Epistola XIII* che per l'*Epistola XIV*, tale livello è maggiore di 0,999.

In conclusione, un *computational authorship verifier* che, in un separato test, si è dimostrato in grado di decidere correttamente sull'autorialità dantesca o meno di 329 testi su 330 (e, in questo test, mai ha indicato come non dantesco un testo autenticamente dantesco), indica non dantesche, con un livello di certezza superiore a 0,999, sia l'*Epistola a Cangrande* che l'*Epistola a Enrico VII.* Possiamo quindi dire che il presente lavoro offre un supporto importante a coloro che, prima di noi e utilizzando diversi strumenti, si erano pronunciati contro la paternità dantesca di questi due testi.

Ringraziamenti

Il lavoro di Martina Leocata è stato finanziato dal progetto "Italian Strengthening of ESFRI RI RESILIENCE" (ITSERR), finanziato dall'Unione Europea nel quadro dello schema di finanziamento Next Generation EU (CUP:B53C22001770006). Per la definizione del *corpus* di testi medievali latini impiegati in questo studio desideriamo ringraziare Francesca Galli, che ci ha generosamente fornito un file editabile con il testo del *De luce* di Bartolomeo da Bologna, tratto dalla sua edizione critica dell'opera; Aurora Panzica, che ha fatto lo stesso per le *Quaestiones in Meteorologica de ultima lectura* di Nicola Oresme; e Gianfranco Fioravanti, che ci ha gentilmente trasmesso la sua trascrizione di lavoro del commento di Antonio Pelacani da Parma alla prima *fen* del *Canone* di Avicenna. Un sentito ringrazia-

mento anche a Paolo Pontari e Mirko Tavosanis per il loro concreto sostegno allo svolgimento della ricerca qui presentata.

Riferimenti bibliografici

Canettieri E.P., *Chi non ha scritto il Fiore*, in *Sulle tracce del Fiore*, N. Tonelli (a cura di), Centro di Studi e Documentazione Dantesca e Medievale, Società Dantesca Italiana, 2016, pp. 121-134.

Corbara S., Moreo A., Sebastiani F., Tavoni M., *L'Epistola a Cangrande al vaglio della* computational authorship verification: *Risultati preliminari* (con una postilla sulla cosiddetta "XIV Epistola di Dante Alighieri"), in A. Casadei (ed.), Atti del Seminario "Nuove Inchieste sull'Epistola a Cangrande", Pisa, Pisa University Press, 2022, pp. 153-192.

Corbara S., Moreo A., Sebastiani F., Tavoni M., MedLatinEpi and MedLatinLit: Two datasets for the computational authorship analysis of medieval Latin texts. *ACM Journal on Computing and Culture Heritage 15*(3): Articolo 57.

Corbara S., Moreo A., Sebastiani F. (2023). Same or different? Diff- vectors for authorship analysis. *ACM Transactions on Knowledge Discovery from Data* 18(1):Article 12.

Crystal D. (2008). *Think on my words: Exploring Shakespeare's language*. Cambridge University Press, Cambridge, UK.

Freund Y., Schapire R.E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory (EuroCOLT 1995)*, Barcelona, ES, 23–37.

Ginzburg C. (1989). *Clues: Roots of an Evidential Paradigm*. In: Id., Clues, Myths and the Historical Method. Baltimore, Johns Hopkins University Press: 96-125.

Harris Z.S. (1954) Distributional Structure. *Word* 10(2–3): 146–162.

Haykin S. (2009). *Neural networks and learning machines*. Pearson, London, UK, 3rd edition.

James G., Witten D., Hastie T., Tibshirani R. (2013). *An introduction to statistical learning with applications in R.* Springer, Heidelberg, DE.

Kabala J. (2020). Computational authorship attribution in medieval Latin corpora: The case of the Monk of Lido (ca. 1101–08) and Gallus Anonymous (ca. 1113–17). *Language Resources and Evaluation* 54(1): 25–56.

Kestemont M., Moens S., Deploige J. (2015). Collaborative authorship in the twelfth century: A stylometric study of Hildegard of Bingen and Guibert of Gembloux. *Digital Scholarship in the Humanities* 30(2): 199–224.

Kestemont M., Stover J., Koppel M., Karsdorp F., Daelemans W., Authenticating the writings of Julius Caesar. *Expert Systems and Applications* 63: 86–96.

Mitchell T.M. (1997). *Machine learning*. McGraw-Hill, New York, US.

Moreo A., Esuli A., Sebastiani F. (2016). Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*, 805–808.

Savoy J. (2020). *Machine learning methods for stylometry: Authorship attribution and author profiling*. Springer, Cham, CH.

Stamatatos E. (2018). Masking topic-related information to enhance authorship attribution. *Journal of the Association for Information Science and Technology* 69(3): 461-473.

Tuccinardi E. (2017). An application of a profile-based method for authorship verification: Investigating the authenticity of Pliny the Younger's letter to Trajan concerning the Christians. *Digital Scholarship in the Humanities* 32(2): 435-447.

Van Halteren H., Baayen R.H., Tweedie F.J., Haverkort M., Neijt A. (2005). New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics* 12(1): 65-77.

Finito di stampare nel mese di ottobre 2025
da Digital Team Srl – Fano (PU)
per conto di Pisa University Press - Polo Editoriale CIDIC - Università di Pisa