UNIVERSITÁ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI SCIENZE STATISTICHE
CORSO DI LAUREA TRIENNALE IN STATISTICA PER L'ECONOMIA E L'IMPRESA

RELAZIONE FINALE

Applicazione di metodi di clustering per l'identificazione delle e-mail spam

Relatrice Prof.ssa Manuela Cattelan
Dipartimento di Scienze Statistiche

Laureanda Maroua Badrane
Matricola 2045575

Anno Accademico 2023/2024

Indice

Introduzione

Al notevole progresso tecnologico verificatosi negli ultimi decenni corrisponde l'evoluzione delle differenti forme di comunicazione. In particolare, al giorno d'oggi, assumono fondamentale importanza le cosiddette *Electronic Mail*, meglio note come e-mail. Si tratta di testi di posta elettronica, il cui utilizzo avviene in diversi contesti, grazie ai loro innumerevoli vantaggi, quali il recapito istantaneo ed il costo nullo d'utilizzo per l'utente.
Tuttavia, con il trascorrere degli anni, questi mezzi cominciarono ad essere impiegati per finalità non del tutto lecite. I messaggi di posta elettronica che ricadono in questa categoria si definiscono *spam*. Si tratta, fondamentalmente, di annunci pubblicitari, e-mail contenenti virus elettronici e proposte d'investimento, volte ad ottenere informazioni personali o dati bancari dal destinatario.
Al fine di limitare tale fenomeno, sono stati ideati molteplici metodi di bloccaggio e filtraggio dei messaggi di posta elettronica. Quest'ultimi identificano le e-mail fraudolente sulla base dell'indirizzo del mittente, dell'oggetto del messaggio oppure dai vocaboli che esso contiene.
Il seguente elaborato si pone come obiettivo quello di valutare la possibilità di utilizzare i metodi del *clustering* per effettuare l'operazione di riconoscimento delle e-mail spam, sulla base delle parole in esse contenute. A tale scopo, vengono effettuate le suddette analisi su un dataset di elevate dimensioni.
Nel primo capitolo avviene la definizione dei messaggi *spam* e dei loro differenti impieghi. Inoltre, vengono descritte alcune delle numerose tecniche tradizionalmente impiegate per il loro bloccaggio e filtraggio, con particolare attenzione al *filtro Bayesiano*, in quanto metodologia maggiormente utilizzata dalle caselle di posta elettronica.
Il secondo capitolo, invece, presenta il dataset oggetto di studio. Vengono, inoltre, effettuate le analisi esplorative sull'insieme di dati tramite il software *R*, che verrà utilizzato anche per l'applicazione degli algoritmi di clustering.
La definizione delle diverse tipologie di classificazione esistenti avviene nel capitolo 3, in cui si espongono i metodi di clustering tradizionale e sparso nelle loro due caratterizzazioni, ossia gerarchica e partizionale.
L'effettiva applicazione delle diverse metodologie introdotte nel capitolo 3 viene eseguita, d'altra parte, nel capitolo 4. Queste analisi, tuttavia, vengono effettuate su un sottoinsieme dei dati per motivi di carattere computazionale.
Segue la conclusione, nella quale si definiranno i risultati ottenuti e si stabilirà quale tecnica permette di ottenere gli esiti più conformi ai dati.

1 Le e-mail spam

1.1 Definizione

Le e-mail spam, anche note con l'acronimo UBE (Unsolicited Bulk E-mails), sono messaggi indesiderati inviati in massa per posta elettronica [9].
Gli autori di questa categoria di messaggi si definiscono con il termine inglese *spammers*. Quest'ultimi, al fine di svolgere la loro attività, si servono di liste di indirizzi di posta elettronica ottenuti in maniera non del tutto legittima sulla rete, oppure generati autonomamente tramite la combinazione di vari nomi e domini.
In termini quantitativi, il traffico delle e-mail spam a livello globale dal 2011 al 2023 si è ridotto notevolmente, passando dal 80.26% del totale dei messaggi inviati via posta elettronica nel 2011, al 45.6% nel 2023, com'è esplicitato dalla Figura 1 [11]. Tuttavia, la percentuale rimane ancora piuttosto elevata, in quanto i messaggi indesiderati rappresentano attualmente circa la metà del totale traffico e-mail. Quindi si tratta di un fenomeno ancora considerevolmente diffuso, nonostante i diversi accorgimenti sviluppati per contrastarlo.

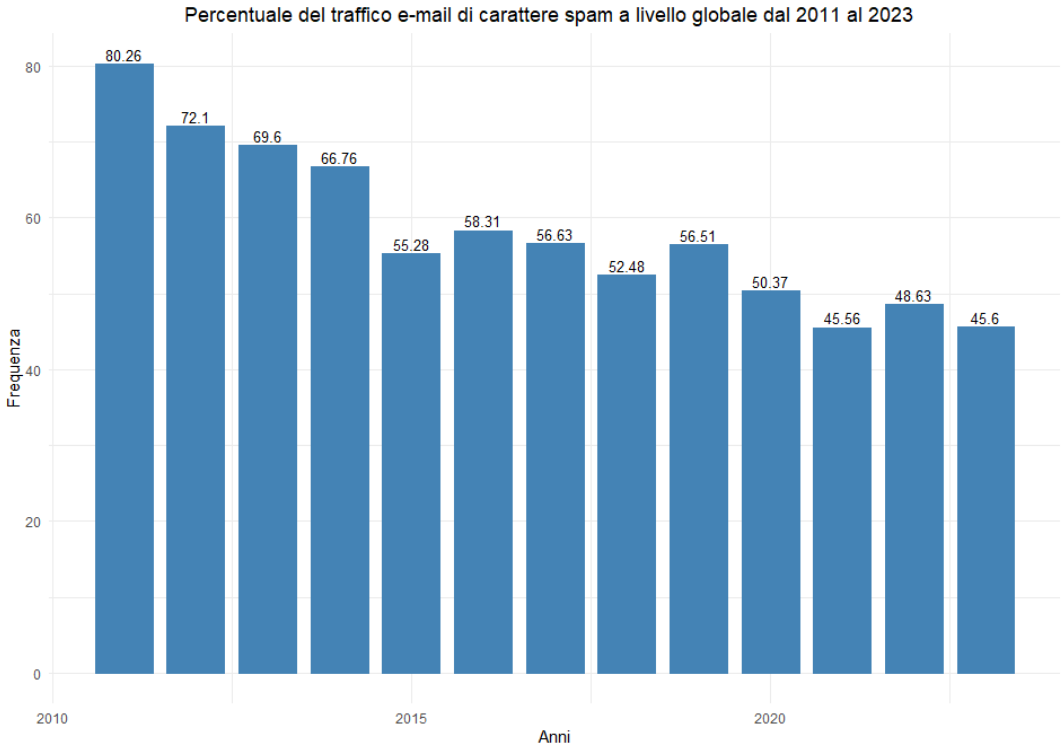

Figura 1: Traffico spam globale dal 2011 al 2023.

Lo spam via e-mail può avere diversi scopi, con risvolti più o meno dannosi per il ricevente. Tra questi vi è la pubblicità, uno degli impieghi principali di questa classe di messaggi, diffusa tramite annunci che promuovono prodotti o servizi commerciali [10].
Questo è, tuttavia, l'uso meno dannoso dei messaggi spam, i quali possono assumere sfumature che mettono a repentaglio la sicurezza in rete dei destinatari. Difatti, essi costituiscono il mezzo a cui ricorrono frequentemente organizzazioni criminali al fine di compiere frodi. Si tratta, in questo caso, di *phishing*, pratica con la quale utenti malintenzionati cercano di indurre le controparti a fare investimenti in denaro o a rivelare dati sensibili, quali chiavi d'accesso o credenziali bancarie. Il phishing rientra in una categoria di truffa che inganna l'utente con e-mail dall'aspetto apparentemente ufficiale, che imitano, nella forma e nel contenuto, messaggi legittimi di fornitori di servizi o di persone a lui note, con l'obiettivo di indurlo a trasmettere le proprie informazioni personali [10].
I messaggi spam possono, in aggiunta, essere strumento di diffusione di *malware*, cioè virus informatici utilizzati per danneggiare, interferire, o trarre dati da un dispositivo. I metodi utilizzati per perseguire questo scopo sono principalmente due: allegare direttamente all'e-mail un file infetto, oppure inserire nel messaggio di spam collegamenti a siti malevoli, tramite i quali avverrà poi il trasferimento del malware [9].

Riconoscere i messaggi di posta elettronica spam è fondamentale nel contesto digitale odierno.
Esistono diversi indicatori che permettono al singolo di identificare questa classe di e-mail [9]. Innanzitutto, è possibile procedere esaminando l'indirizzo e il dominio del mittente. Infatti, i messaggi autentici provengono solitamente da fonti attendibili, con nomi di dominio riconoscibili, mentre la posta spam giunge spesso da indirizzi la cui denominazione risulta non affidabile o priva di senso.
Ulteriore aspetto degno di attenzione è il contenuto dell'e-mail. Tra gli elementi più comuni all'interno dei messaggi indesiderati figurano errori ortografici e grammaticali, l'uso eccessivo di lettere maiuscole, richieste di informazioni personali o finanziarie e tono di voce urgente.
Per quanto riguarda l'ambito informatico, invece, esistono diversi programmi, i cosiddetti *antispam*, che permettono di arginare il problema della posta indesiderata, implementando metodologie basate su bloccaggio o filtraggio del traffico e-mail. Si tratta di tecniche che concentrano la loro attenzione sul mittente, oggetto e contenuto del messaggio. A partire da tali caratteristiche, classificano l'e-mail di riferimento come *ham*, se autentica, o *spam*, nel caso in cui rientri nella categoria della posta indesiderata.

1.2 I metodi di filtraggio e bloccaggio

Sono stati sviluppati molteplici servizi e programmi, denominati *antispam*, che i server e-mail impiegano al fine di ridurre il carico di messaggi indesiderati sulle caselle di posta. Alcuni di questi procedono con il rifiuto automatico dei messaggi provenienti da domini identificati come *spammers*. Altri analizzano il contenuto delle e-mail ed eliminano o spostano in una cartella apposita quelle che risultano essere non autentiche. Questi due approcci al problema si definiscono rispettivamente bloccaggio e filtraggio.
Entrambe le tecniche riducono l'ammontare di spam inviata alle caselle postali degli utenti. Inoltre, il bloccaggio permette anche di attenuare il costo computazionale dovuto alla posta elettronica indesiderata, in quanto rifiuta i messaggi prima che siano trasmessi al server dell'utente [6].

1.2.1 Il bloccaggio

I server di posta elettronica che fanno uso di metodologie di bloccaggio impediscono l'arrivo alle caselle postali dei destinatari di messaggi provenienti da indirizzi sconosciuti. Molteplici sono i sistemi di bloccaggio. In questo paragrafo, si procede con la presentazione dei quattro metodi più impiegati.

Una prima tecnica di bloccaggio prevede l'utilizzo delle *DNSBL* (Domain Name System-based Blackhole Lists), denominate anche *blacklist*. Si tratta di un insieme di liste che permettono all'amministratore del sistema di rifiutare messaggi provenienti da indirizzi con un trascorso nell'ambito spam [13].
Il funzionamento si basa su un principio semplice: il server accede alle liste *DNSBL* disponibili in rete, e verifica l'eventuale presenza del dominio da cui riceve il messaggio [14]. Se il controllo dimostra che l'indirizzo è nell'elenco, l'e-mail non viene accettata. Infatti, la quasi totalità dei server di posta elettronica sono configurati in maniera tale da rifiutare o contrassegnare posta elettronica proveniente da mittenti presenti in una o più liste *DNSBL*.
Opposta alla *blacklist* è la *whitelist*, che, invece di determinare una lista di utenti inaffidabili, crea un elenco di domini da cui consentire la posta [6].

Ulteriore metodo di bloccaggio frequentemente utilizzato è il *graylisting*. Nel momento in cui il messaggio arriva per la prima volta da mittenti sconosciuti, il server di posta elettronica, che fa uso del sistema di *graylisting*, respinge l'e-mail [15]. Si procede successivamente con l'invio di una comunicazione al mittente, in cui gli viene richiesto di provare nuovamente il recapito dopo un periodo d'attesa preciso. Un server di posta elettronica lecito riuscirà a soddisfare tale richiesta e procederà ad inviare nuovamente il messaggio, che, in questo caso, sarà accettato.

D'altra parte, i mittenti non legittimi, generalmente, non effettuano una seconda prova. Senza un secondo tentativo, l'e-mail spam non sarà mai consegnata [6].

In conclusione, un sistema di bloccaggio alternativo è il *DKIM*, o DomainKeys Identified Mail, cioè un metodo di autenticazione dell'e-mail, che consente al destinatario di verificare che essa provenga da un dominio specifico e che sia stata effettivamente autorizzata dal relativo proprietario, tramite l'apposizione di una firma digitale [16]. Si intende, adoperando questo metodo, prevenire l'uso di indirizzi falsificati nelle e-mail, in quanto il server destinatario procede con la verifica della firma. Nel caso di fallimento dei controlli, il messaggio non verrà trasmesso nella casella di posta elettronica.

1.2.2 Il filtraggio

I filtri antispam presentano un approccio differente al problema, in quanto non prevedono l'espulsione del messaggio, ma la sua semplice categorizzazione come posta indesiderata. Questi si basano sull'analisi del contenuto, dell'oggetto e dell'indirizzo di provenienza del messaggio.
Due sono le principali metodologie che rientrano in questa categoria: il filtraggio euristico ed il filtraggio statistico [12].

I filtri che fanno uso di metodi euristici esaminano il contenuto delle e-mail ricevute ed assegnano determinati punteggi alle parole che costituiscono il messaggio. Termini frequentemente presenti nella posta spam riceveranno un punteggio superiore rispetto a vocaboli comunemente usati. Successivamente, il filtro procede con la somma dei punti e calcola la valutazione complessiva. Se quest'ultima risulta maggiore di una soglia predefinita, l'e-mail viene etichettata come indesiderata e, conseguentemente, trasferita nell'apposita cartella, altrimenti viene trasmessa all'utente destinatario.
I filtri euristici, tuttavia, generano un numero elevato di falsi positivi, specialmente nel momento in cui un mittente legittimo utilizzi una specifica combinazione di parole. Inoltre, lo spammer è in grado di evitare i termini considerati spam dal filtro euristico, aggirandolo [6].

Il filtraggio statistico, d'altra parte, fa uso dei cosiddetti filtri bayesiani. Questi si basano su leggi matematiche e probabilistiche al fine di determinare la possibile legittimità del messaggio. Si tratta della tipologia di filtro maggiormente impiegata dai server di posta elettronica, poiché considerato essere il sistema più efficace ed affidabile [12].

1.3 Filtro Bayesiano

1.3.1 Bayes

Lo spam filter bayesiano è un programma in grado di riconoscere se un'e-mail è autorizzata o meno. Questo si basa sulla *Naive Bayes Classification* (NBC), ossia una tecnica di machine learning che permette di ripartire oggetti in due o più classi. In questo caso si occupa della classificazione dei testi dei messaggi di posta elettronica in due categorie: *spam* (posta indesiderata) e *ham* (messaggio reale).

Le variabili analizzate dal modello sono rappresentate dalle parole presenti all'interno del messaggio. Ogni termine avrà una determinata probabilità di essere contenuto nella posta spam. Maggiore risulta questa quantità, più è verosimile che il messaggio sia illegittimo. Al fine di calcolare la probabilità che un determinato vocabolo sia presente all'interno di un'e-mail indesiderata, si impiega il teorema probabilistico di Bayes [3].

Sia S l'evento che l'e-mail di riferimento sia spam, H l'evento che sia ham (autentica) e W l'evento che il messaggio contenga una determinata parola. La probabilità

$$\mathrm{P(S|W)}$$

ossia che l'e-mail di riferimento sia spam dato che contiene la parola W, è ottenuta tramite l'applicazione del teorema di Bayes:

$$P(S \mid W) = \frac{P(W \mid S)P(S)}{P(W \mid S)P(S) + P(W \mid H)P(H)}. \tag{1}$$

In questo modo si calcola la probabilità relativa ad una sola parola. Il filtro antispam estende, quindi, questo concetto, ad un numero maggiore di termini, in quanto esamina ogni vocabolo contenuto nel messaggio.

Consideriamo le variabili x_1,...,x_n come le parole distinte che costituiscono il testo di un'e-mail. Il nostro obiettivo è quello di quantificare la probabilità che, date le x_1,...,x_n parole, il messaggio sia spam, ovvero

$$\mathrm{P(S \mid } x_1,...,x_n),$$

dove indichiamo con S l'evento che il messaggio sia spam, con H l'evento che il messaggio sia leggittimo (ham), che equivale a $\bar{\mathrm{S}}$ (S negato). Applicando il teorema presentato nella formula (1) si ottiene

$$P(S \mid x_1, ..., x_n) = \frac{P(x_1, ..., x_n \mid S)P(S)}{P(x_1, ..., x_n \mid S)P(S) + P(x_1, ..., x_n \mid H)P(H)}. \tag{2}$$

Dalla definizione di probabilità condizionata deriva

$$P(x_1, ..., x_n \mid S)P(S) = P(x_1, ..., x_n, S). \tag{3}$$

Inoltre, utilizzando la regola della probabilità composta, si riscrive il numeratore della (2) a partire dalla (3) nel seguente modo

$$\begin{aligned} P(x_1, ..., x_n, S) &= P(x_1 \mid x_2, ..., x_n, S)P(x_2, ..., x_n, S) \\ &= P(x_1 \mid x_2, ..., x_n, S)P(x_2 \mid x_3, ..., x_n, S)P(x_3, ..., x_n, S) \\ &= ... \\ &= P(x_1 \mid x_2, ..., x_n, S)P(x_2 \mid x_3, ..., x_n, S)... \\ &\quad ...P(x_{n-1} \mid x_n, S)P(x_n \mid S)P(S). \end{aligned} \tag{4}$$

Per poter procedere con il calcolo di tale quantità, è necessario specificare l'assunzione sulla quale si basa la *Naive Bayes Classification*. Questa, infatti, prevede che tutte le parole contenute in un'e-mail siano condizionatamente indipendenti l'una dall'altra, essendo noto a priori se si tratta di un messaggio spam o reale. Quindi, in altre parole, sapere che in un testo di posta elettronica è presente una determinata parola non fornisce informazioni riguardo l'eventuale presenza di un'altro vocabolo. Perciò si parla di *Naive Bayes*, letteralmente il "Bayes ingenuo": assume "ingenuamente" l'indipendenza, dove potrebbe effettivamente non esserci.
Quest'ipotesi permette di semplificare il problema e di ottenere risultati pratici, ma non è del tutto realistica, in quanto molto spesso si incontrano insiemi di parole utilizzate congiuntamente.
Tornando al problema di nostro interesse (formula (4)) ed applicando l'assunzione esplicitata, risulta

$$\begin{aligned} P(x_1, ..., x_n, S) &= P(x_1 \mid x_2, x_3, ..., x_n, S)P(x_2 \mid x_3, ..., x_n, S)... \\ &\quad ...P(x_{n-1} \mid x_n, S)P(x_n \mid S)P(S) \\ &\approx P(x_1 \mid S)P(x_2 \mid S)...P(x_{n-1} \mid S)P(x_n \mid S)P(S) \\ &= P(S)\prod_{i=1}^{n} P(x_i \mid S). \end{aligned}$$

In modo analogo,

$$P(x_1, ..., x_n, H) \approx P(H) \prod_{i=1}^{n} P(x_i \mid H).$$

Unendo il tutto, è possibile riscrivere la (2) come

$$P(S \mid x_1, ..., x_n) \approx \frac{P(S) \prod_{i=1}^{n} P(x_i \mid S)}{P(S) \prod_{i=1}^{n} P(x_i \mid S) + P(H) \prod_{i=1}^{n} P(x_i \mid H)}. \tag{5}$$

Dunque, in questo modo è possibile individuare il risultato se si è a conoscenza della probabilità a priori che un'e-mail sia spam, e le eventualità di riscontrare una specifica parola in un messaggio legittimo ed in uno indesiderato.

1.3.2 Calcolo della probabilità spam di una parola

Al fine di calcolare la probabilità che una parola sia spam, ossia che appartenga ad un messaggio di posta elettronica indesiderata, è possibile procedere nel seguente modo: si calcola la frequenza della posta spam contenente la parola di riferimento e la si divide per il numero totale di messaggi indesiderati, ovvero

$$P(parola \mid S) = \frac{n_{ps}}{n_s},$$

con n_{ps} frequenza della parola all'interno dei messaggi spam, e n_s numero totale della posta elettronica spam [3].
In modo del tutto analogo, per calcolare la probabilità che il vocabolo appartenga ad un'e-mail reale, si calcola il numero di messaggi contenenti quella determinata parola, e si divide tale quantità per il totale di testi autentici, cioè:

$$P(parola \mid H) = \frac{n_{ph}}{n_h},$$

con n_{ph} frequenza della parola all'interno dei messaggi ham, e n_h numero complessivo della posta elettronica legittima.
Questa tipologia di approccio, tuttavia, presenta un ostacolo. Se si considera una parola che si è riscontrata solo in messaggi ham, la probabilità che l'e-mail sia spam risulta nulla. Consideriamo come esempio il vocabolo *real*. Se nell'analisi dei dati si incontra questa parola solo in posta reale, allora

$$P(real \mid S) = 0,$$

e, conseguentemente, la probabilità che il messaggio sia spam risulta zero, dato che la si ottiene, come precisato nel paragrafo precedente dalla formula (5), come prodotto delle probabilità spam di ogni termine. Lo spammer può, conseguentemente, sfruttare questa fragilità del classificatore per evitarlo.
La soluzione è fornita dalla tecnica del lisciamento di Laplace, o lisciamento esponenziale. Questa prevede che le probabilità non siano mai nulle, se avviene il loro livellamento verso l'alto. Infatti, invece che iniziare il conteggio di ogni parola da zero, si inizia da uno. Ciò conduce alla sovrastima delle probabilità di ogni parola, perciò si corregge anche il denominatore, aggiungendo 2.
Quindi le probabilità corrette diventano

$$P(parola \mid S) = \frac{n_{ps} + 1}{n_s + 2}$$
$$P(parola \mid H) = \frac{n_{ph} + 1}{n_h + 2}.$$

1.3.3 L'algoritmo di Bayes

L'algoritmo di Bayes richiede, innanzitutto, la definizione di un *training set*, cioè un complesso di messaggi di posta elettronica di cui si conosce a priori la legittimità [3]. Per ogni parola W nell'insieme di e-mail di interesse, si procede

1. Calcolando la probabilità che la parola sia spam

$$P(W \mid S) = \frac{numero\ di\ e-mail\ spam\ contenenti\ W + 1}{numero\ di\ e-mail\ spam + 2},$$

2. Calcolando la probabilità che i vocaboli siano ham

$$P(W \mid H) = \frac{numero\ di\ e-mail\ ham\ contenenti\ W + 1}{numero\ di\ e-mail\ ham + 2},$$

3. Calcolando la probabilità che il messaggio nella sua totalità sia spam

$$P(S) = \frac{e-mail\ spam}{e-mail\ spam + e-mail\ ham},$$

4. Calcolando la probabilità che il messaggio nella sua totalità sia ham

$$P(H) = \frac{e - mail\ ham}{e - mail\ spam + e - mail\ ham}.$$

Una volta computate le suddette quantità sul *training set*, è possibile procedere valutando l'autenticità di un insieme di testi di posta elettronica, la cui natura è sconosciuta, secondo il seguente approccio:

a. Si crea l'insieme $\{x_1,...,x_n\}$ delle parole uniche che costituiscono il corpo dei messaggi. Il gruppo di vocaboli di riferimento deve corrispondere a quello considerato nel training set.

b. Si calcola la probabilità che l'email sia spam, dato il relativo insieme di parole, a partire dalle quantità definite nel training set.

$$P(S \mid x_1, ..., x_n) \approx \frac{P(S)\prod_{i=1}^{n} P(x_i \mid S)}{P(S)\prod_{i=1}^{n} P(x_i \mid S) + P(H)\prod_{i=1}^{n} P(x_i \mid H)}.$$

c. Se

$$P(S \mid x_1, ..., x_n) > 0.5$$

l'output è spam, altrimenti è ham.

1.4 Obiettivi dello studio

L'analisi oggetto di studio si pone come obiettivo quello di individuare un metodo statistico alternativo al filtraggio bayesiano per il riconoscimento delle e-mail spam.
Il suddetto approccio impiega i metodi dell'analisi dei dati multidimensionali, in particolar modo il clustering. Questo sistema si basa sul raggruppamento di oggetti in classi omogenee al loro interno, ed eterogenee tra di loro. In questo caso i gruppi di riferimento sono due, ossia spam e autentico.
Procederemo presentando le diverse metodologie di clustering, in particolar modo il clustering tradizionale ed il clustering sparso, per poi applicarli ad un insieme di dati, i cui dettagli saranno presentati nel capitolo 2.

2 I dati

2.1 Il dataset

Il dataset oggetto di studio, ricavato dalla piattaforma di data science *Kaggle* [8], è di elevate dimensioni, in quanto composto da 5172 righe e 3002 colonne.
Le righe rappresentano le unità statistiche, in questo caso un insieme di messaggi di posta elettronica, sia spam che autentici, selezionati in maniera del tutto casuale.
Le colonne, invece, ricoprono differenti funzioni. La prima si riferisce al singolo messaggio, che viene codificato tramite l'uso di numeri e non con il nome del destinatario, in maniera tale da proteggerne la privacy. L'ultima colonna, d'altra parte, contiene la variabile risposta, dicotomica, la cui analisi si svolgerà in modo dettagliato nel paragrafo 2.2.1.
Le restanti 3000 colonne, cioè dalla numero 2 alla 3001, rappresentano le parole in lingua inglese contenute in tutte le e-mail prese in considerazione, senza ripetizioni. Nella raccolta dei dati sono stati esclusi, quindi, numeri, simboli e qualsiasi carattere non alfabetico.
Le variabili esplicative sono di natura quantitativa discreta, in quanto esprimono la frequenza con cui si riscontra il rispettivo vocabolo all'interno dell'e-mail di riferimento.

2.2 Analisi esplorativa

L'analisi esplorativa del dataset di riferimento procederà nel modo seguente.
Innanzitutto, si effettuerà lo studio delle caratteristiche della variabile risposta e si evidenzieranno i risultati derivanti da tale analisi.
Lo studio delle esplicative, invece, avverrà seguendo tre percorsi differenti, dato il volume significativo di variabili.
In un primo momento, si considereranno tutti i termini congiuntamente, in maniera tale da poter ottenere un quadro generale relativo alla totalità dei dati.
In seguito, si dividerà il dataset in due sottogruppi, prendendo in considerazione separatamente le parole contenute esclusivamente in e-mail spam, e quelle presenti solo in messaggi autentici.
In conclusione, avverrà la selezione di un complesso di parole, dalle caratteristiche peculiari, di cui si esamineranno le relazioni esistenti e i comportamenti rispetto alla variabile risposta.

2.2.1 Variabile risposta

La variabile risposta è *Prediction*. Quest'ultima rappresenta la natura del messaggio, determinandone l'autenticità.
Nel dataset oggetto di studio, *Prediction* viene originariamente contrassegnata dall'autore come quantitativa discreta. Considerando, tuttavia, il ruolo della variabile, specificato precedentemente, è stata effettuata una codifica della stessa in fattore a due livelli.
Si ottiene, così, una risposta dicotomica, che assume valore 1 nel caso in cui l'e-mail sia spam, 0 altrimenti, ovvero, denotando con Y_i il valore della risposta nell'i-esimo messaggio:

$$Y_i = \begin{cases} 1 & \text{spam,} \\ 0 & \text{ham,} \end{cases} \qquad i = 1, ..., 5172.$$

Nel complesso, dall'analisi della risposta, si conta un totale di 3672 e-mail autentiche, e 1500 e-mail spam, per un rapporto di 71% circa di messaggi reali rispetto a 29% illegittimi. La Figura 2 rappresenta a livello grafico tali risultati, ossia il volume di e-mail spam e ham nell'insieme dei dati di riferimento.

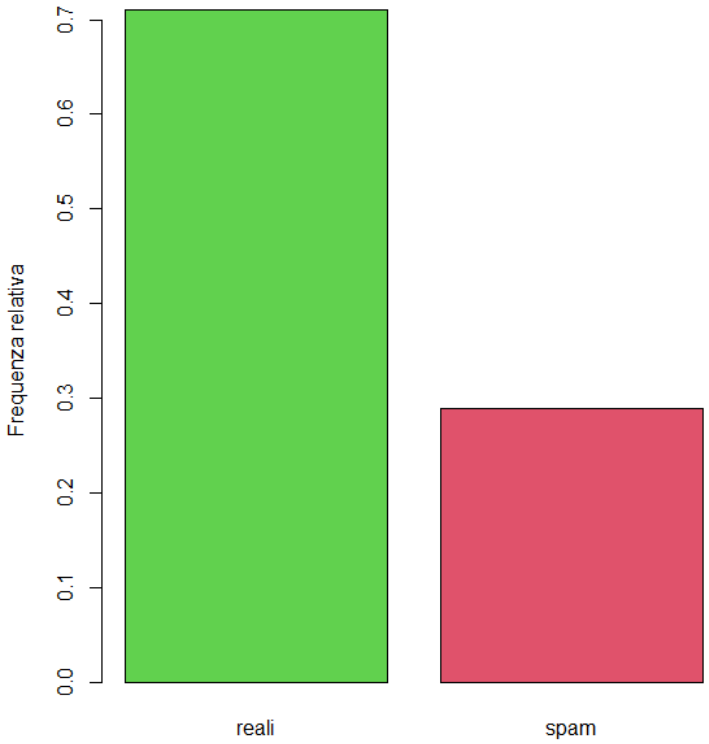

Figura 2: Volume delle e-mail spam e autentiche.

2.2.2 Variabili esplicative

Le variabili esplicative, come esplicitato precedentemente, sono 3000. Si tratta dei vocaboli utilizzati all'interno delle 5172 e-mail d'interesse, in lingua inglese ed ovviando a ripetizioni.

Poiché si tratta di parole contenute in testi, vi è la possibilità che un termine non figuri in un certo numero di messaggi, conducendo conseguentemente ad un'inflazione di zeri. Ciò accade nei dati oggetto di studio, in quanto circa il 94,37% della totalità dei valori sono nulli.

Ciascun vocabolo appare mediamente 2008.959 volte nei testi di posta elettronica considerati nella totalità. Dall'analisi generale, risulta che i termini con frequenza maggiore sono particolarmente brevi. Si tratta, fondamentalmente, di

monosillabi, congiunzioni o singoli caratteri alfabetici.
D'altra parte, le parole che appaiono più raramente sono notabilmente elaborate e lunghe. Esempi di tale aspetto sono visibili nelle tabelle 1 e 2, in cui sono riportate, rispettivamente, le tre parole più frequenti e i tre vocaboli che figurano occasionalmente.

parola	frequenza
e	438561
t	312791
a	287136

Tabella 1: Parole a frequenza maggiore.

parola	frequenza
felipe	21
explosion	21
encoding	21

Tabella 2: Parole a frequenza minore.

In particolare, la parola che si presenta il numero massimo di volte nella totalità dei messaggi, ossia 438561 volte, è *e*. Questa costituisce, difatti, il 7,28% del volume totale dei termini.
Gli aspetti descrittivi della suddetta variabile sono contenuti nella tabella 3, dalla quale emerge che il valore minimo assunto da *e* è 1, mentre il massimo è 2327. Quindi, da questo risultato, si deduce che si tratta di un vocabolo che appare in tutte le e-mail di riferimento, almeno 1 volta.
Tale termine figura maggiormente in e-mail reali, nelle quali si presenta 287499 volte, costituendo il 65,56% del totale delle apparizioni. Il restante 34,44% del volume della lettera *e* è contenuto nei messaggi illegittimi, in cui si presenta 151062 volte. Questa differenza, riconducibile principalmente al fatto che nel dataset vi sia predominanza di messaggi autentici, emerge anche dal grafico riportato in Figura 3, in cui è evidente un andamento decrescente della frequenza di tale parola, sia per le e-mail autentiche che per le spam, oltre alla predominanza del vocabolo nella posta elettronica reale.
Nella Figura 4, d'altronde, si nota il comportamento della variabile considerata rispetto alla risposta. Il boxplot, difatti, evidenzia particolare variabilità sia per le e-mail reali che per le e-mail spam, con lievi differenze tra le due categorie, ed un numero considerevole di valori anomali.

Min.	1.0
1st Qu.	18.0
Median	42.0
Mean	84.8
3rd Qu.	97.0
Max.	2327.0

Tabella 3: Statistiche descrittive di *e*.

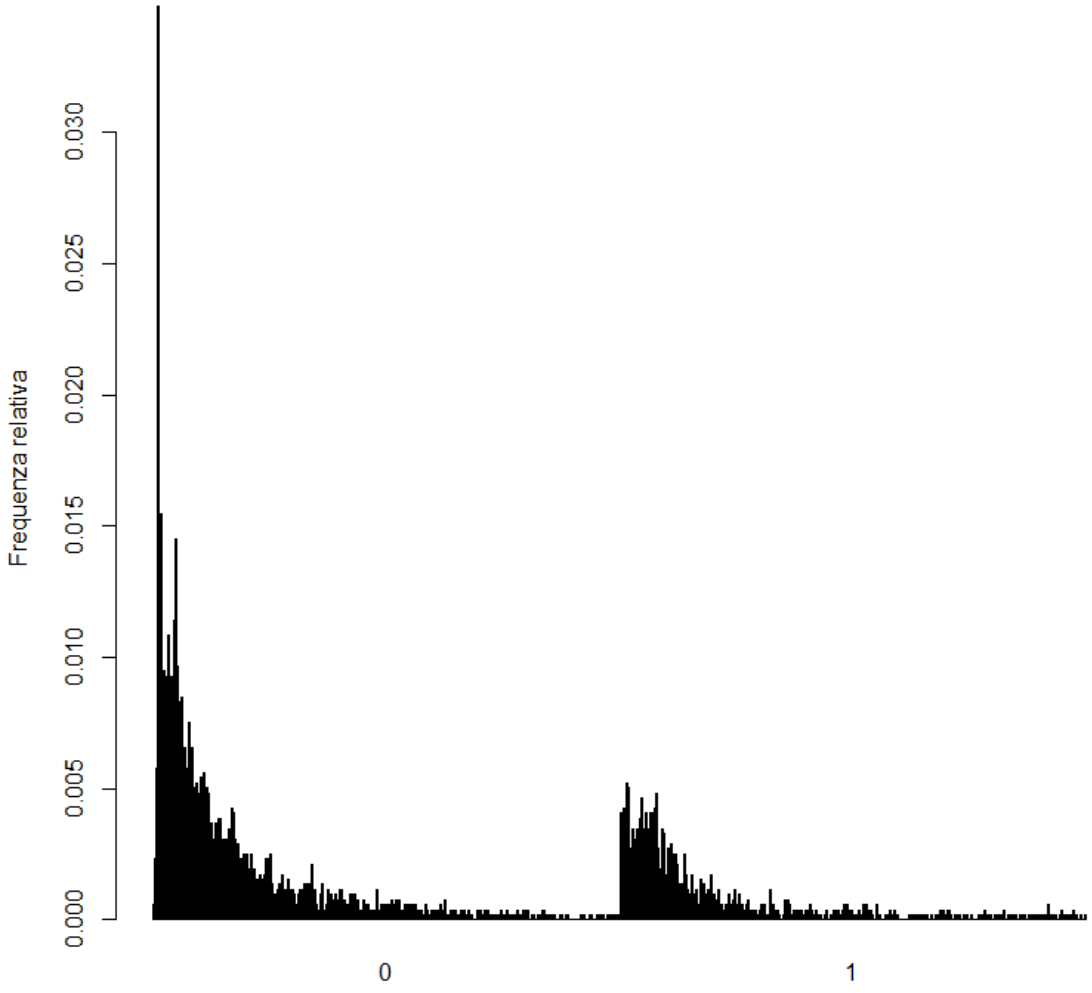

Figura 3: Frequenza di *e* in e-mail spam (1) e ham (0).

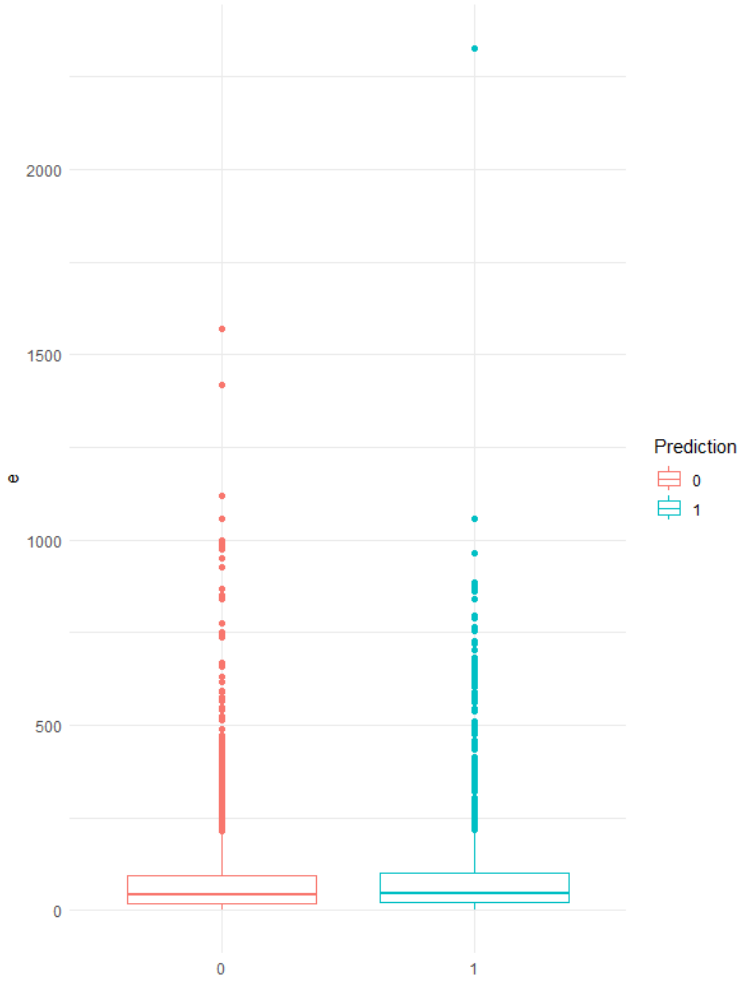

Figura 4: Boxplot di *e* e *Prediction*.

Proseguendo con lo studio delle variabili, le parole che appaiono con volume minore sono *felipe*, *explosion*, *encoding*, *returns*, *padre*, *flw*, *formosa*, *dorcheus*, *pooling*, *offsystem*, *decisions*, *greatest*, *remains* e, infine, *allowing*. Tutti i vocaboli elencati si presentano con frequenza pari a 21, contribuendo al 0.0003% circa del volume totale dei termini.
Ciascuna di queste variabili ha un comportamento particolare rispetto alla risposta, *Prediction*. Molteplici sono le parole con andamento estremo. Tra queste vi sono *felipe*, *dorcheus*, *pooling* e *offsystem*, che si manifestano unicamente in messaggi di posta elettronica reale. Non figurano mai, quindi, in e-mail di carattere spam. Anche *padre* si presenta con notevole prevalenza in e-mail autentiche, circa il 90.47% delle volte. Comportamento simile vanta anche *flw*, con 85.71% del relativo volume nei messaggi autentici. *Formosa*, a sua volta, ha frequenza net-

tamente maggiore nelle e-mail reali, pari a 95.24%. *Remains* e *allowing*, inoltre, si presentano nei messaggi legittimi con volumi rispettivamente pari a 71.43% e 61.90%.
D'altra parte, termini che appaiono principalmente in e-mail spam sono *explosion*, con il 95.24% del volume totale. *Encoding*, a sua volta, prevale nella posta elettronica fraudolenta, con il 90.48% del totale delle manifestazioni.
Trattando, invece, di *returns*, si nota come questa si distribuisca quasi equamente tra messaggi spam e reali, con il 52.38% della relativa frequenza in quelli autentici e il 47.62% in quelli illegittimi.
Infine, *decisions* e *greatest* hanno comportamento identico tra di loro, in quanto si presentano entrambe il 57.14% delle volte nei messaggi autentici. Tutti questi risultati sono visibili nei grafici che seguono.

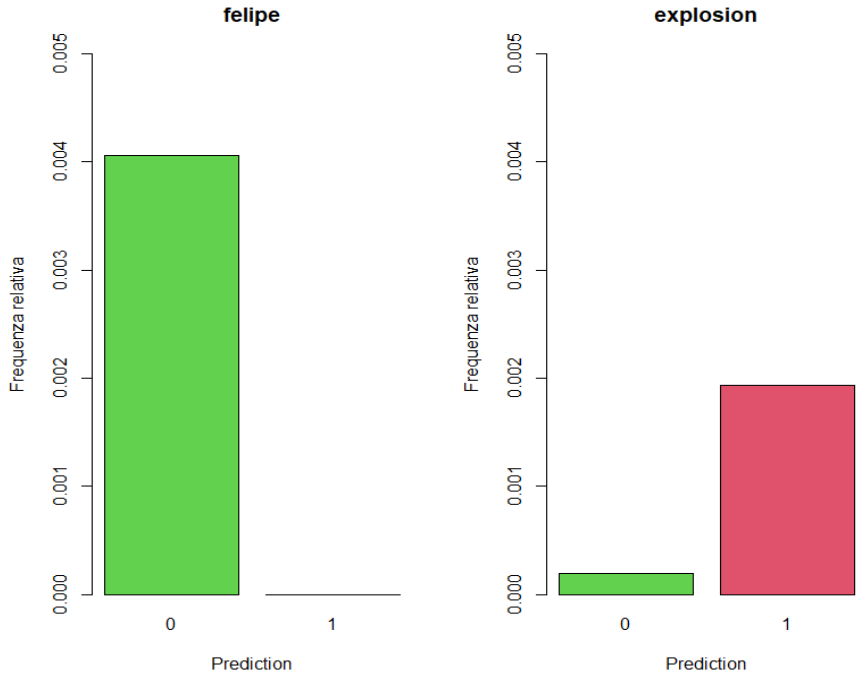

Figura 5: Frequenza di *felipe* e *explosion*.

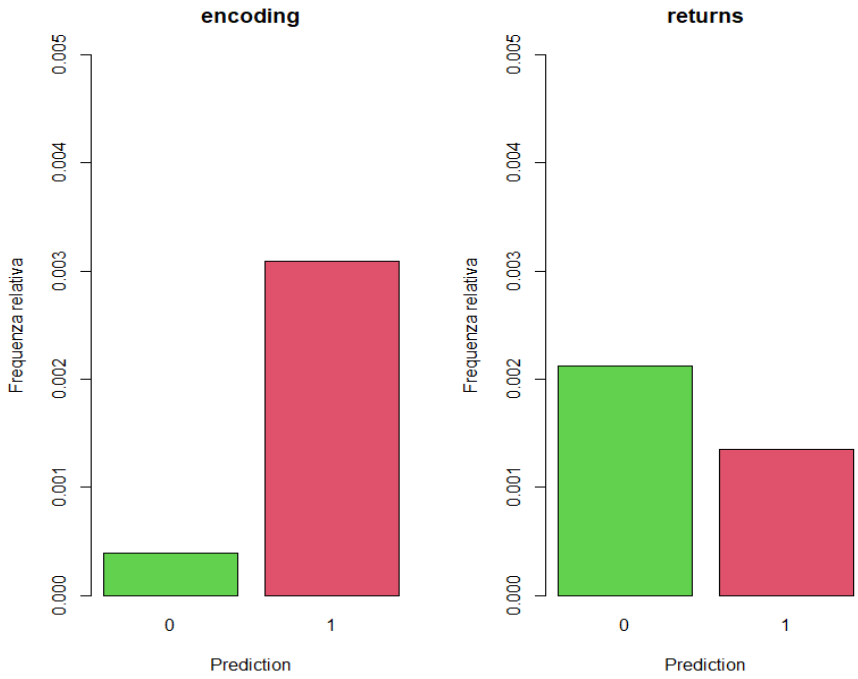

Figura 6: Frequenza di *encoding* e *returns*.

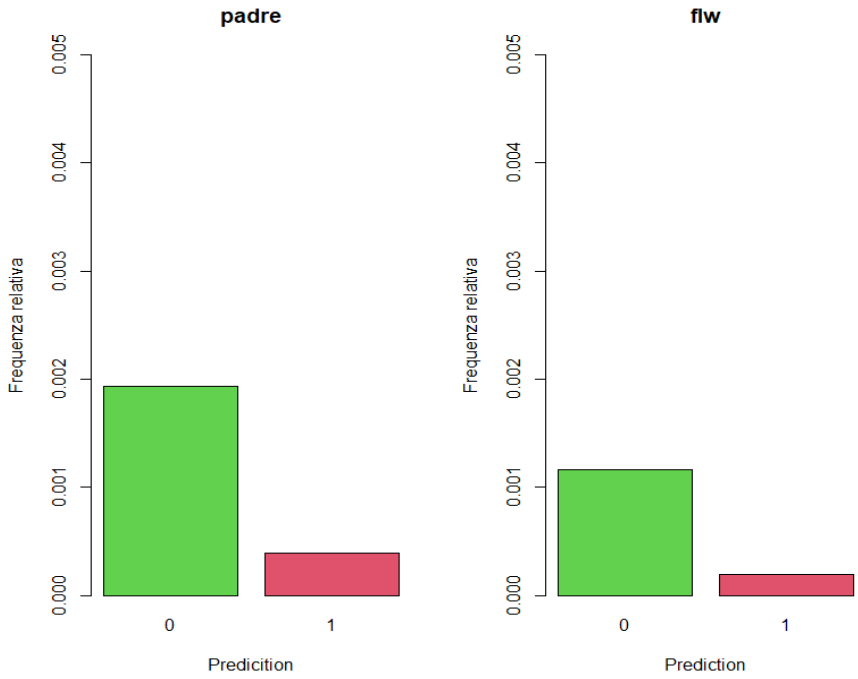

Figura 7: Frequenza di *padre* e *flw*.

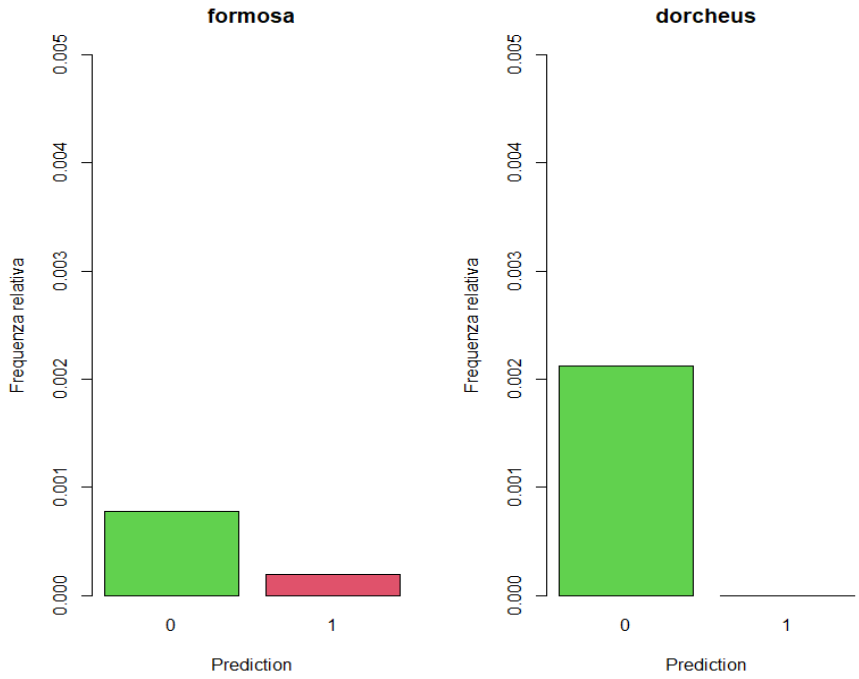

Figura 8: Frequenza di *formosa* e *dorcheus*.

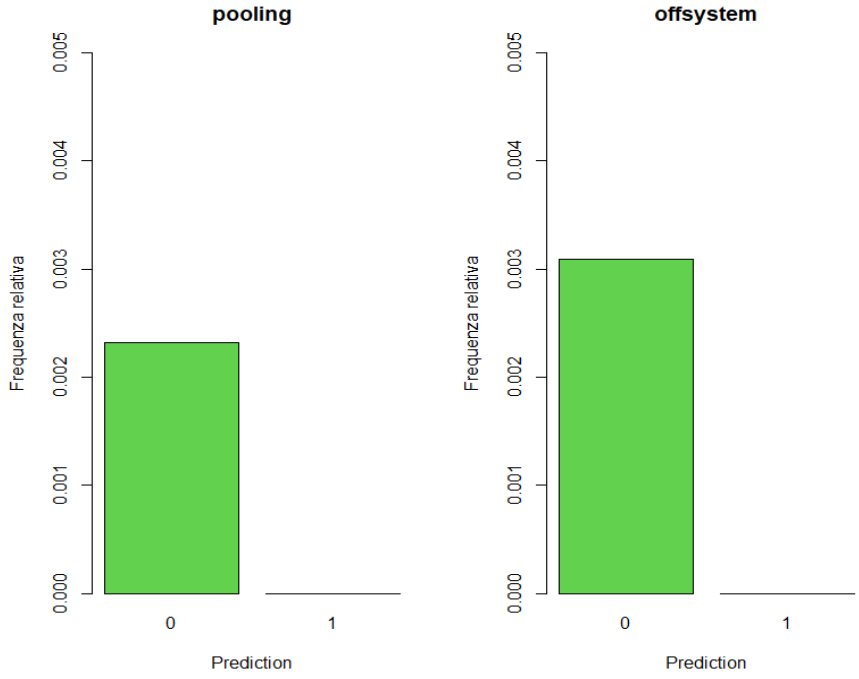

Figura 9: Frequenza di *pooling* e *offsystem*.

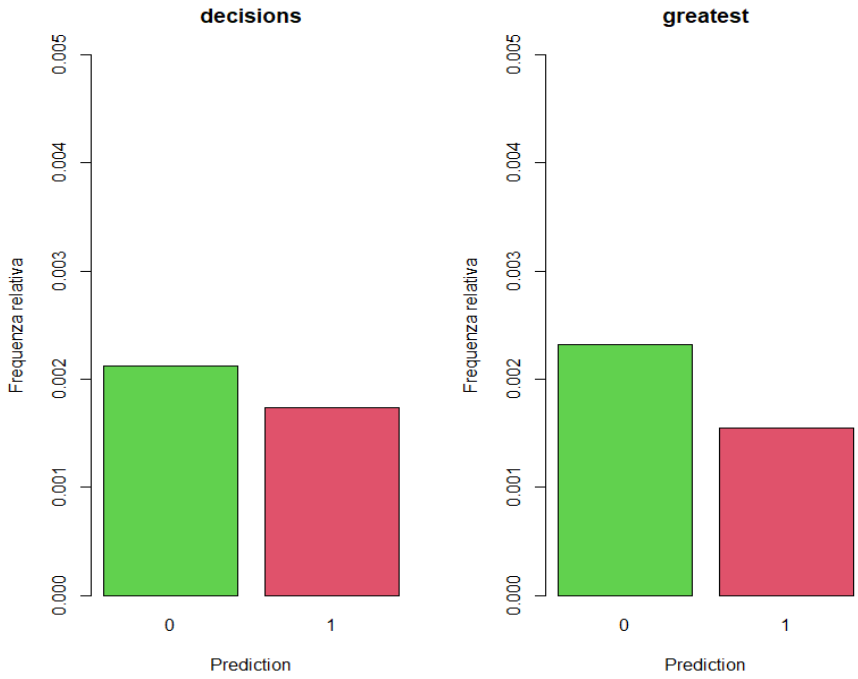

Figura 10: Frequenza di *decisions* e *greatest*.

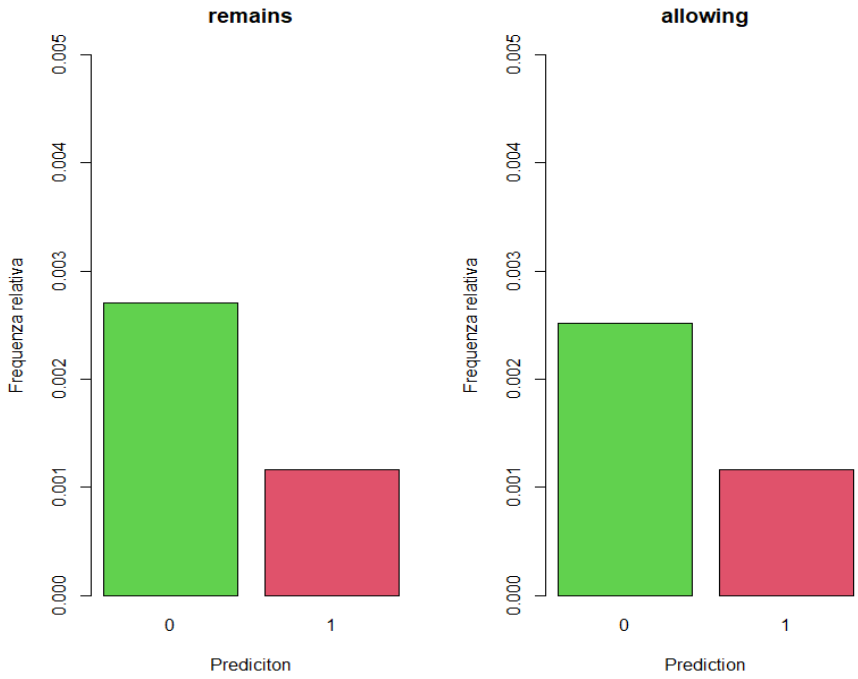

Figura 11: Frequenza di *remains* e *allowing*.

Considerando, invece, le singole e-mail, cioè le unità statistiche, come ci si aspetterebbe, tutte contengono almeno una parola, cioè nessun testo è vuoto.
L'e-mail con il numero massimo di vocaboli è la numero 626, con un totale di 29178 parole. In termini di autenticità, questa risulta spam.
D'altra parte, il messaggio con numero minore di termini è l'e-mail 303, contenente 8 parole, a sua volta non reale.
In media, il numero di vocaboli contenuto nei messaggi di posta elettronica è 1165.289. Circa il 64.49% di e-mail contengono un numero di parole superiore alla media, contro il 31.51% sotto la media.
Dalle analisi emerge che, la maggior parte dei testi con un numero di vocaboli superiore alla media sono autentici. Quest'ultimo risultato, visibile nella tabella 4, evidenzia una peculiarità dei messaggi illegittimi, ossia la brevità dei contenuti.

0 (reali)	1 (spam)
0.6849	0.3151

Tabella 4: Numero di e-mail autentiche e spam contenenti un volume di parole maggiore della media.

In generale, gran parte dei vocaboli contenuti nell'insieme dei dati risiedono nelle e-mail autentiche, come mostrato dalla tabella 5.

0 (reali)	1 (spam)
0.7178	0.2822

Tabella 5: Distribuzione delle parole tra e-mail spam e reali.

2.3 Parole solo in e-mail spam o ham

In questo paragrafo ci si focalizzerà sugli aspetti caratteristici di sottoinsiemi di termini. Si tratta di due sottogruppi ottenuti selezionando separatamente i vocaboli che figurano esclusivamente in e-mail non autentiche, e quelli che appaiono solo in messaggi reali nel dataset oggetto d'analisi.

Si inizia esaminando i vocaboli che figurano unicamente in e-mail spam. Quest'ultimi vengono definiti dall'espressione *parole spam*.
Sono 89 i termini che non compaiono mai in e-mail autentiche, con frequenze molto variabili. In particolar modo si contano 23 apparizioni (0.0011% del volume totale delle parole spam) per i vocaboli *macintosh*, *constitutes* e *newsietter*, che

costituiscono, quindi, le espressioni che si presentano il minimo numero di volte. D'altra parte, la parola che si presenta con frequenza maggiore è *nbsp*, che, in quest'ambito, figura 485 volte (0.022% del volume totale delle parole spam).

L'analisi delle interazioni tra le parole spam fa emergere un aspetto particolare. Difatti molteplici sono le correlazioni importanti, con valori superiori al 90%. Tale aspetto è evidente nel grafico riportato nella Figura 12, in cui le tonalità più scure di colore indicano un'associazione particolarmente importante. Inoltre, emerge che non vi siano correlazioni negative tra le parole. Per rendere tale grafico leggibile, data l'elevata numerosità di termini, si sono codificati quest'ultimi tramite l'uso di numeri.

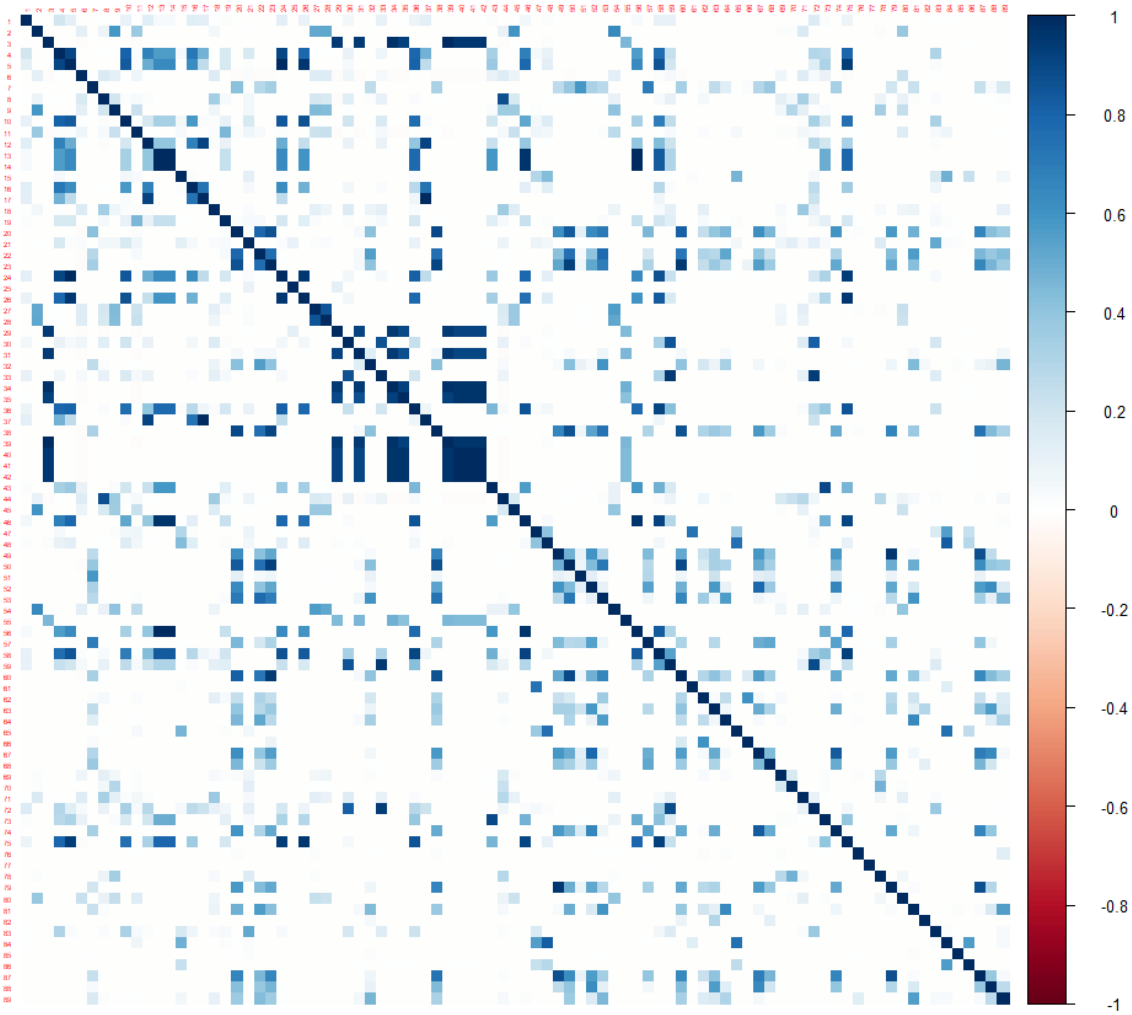

Figura 12: Grafico di correlazione delle parole spam.

Consideriamo alcuni dei vocaboli maggiormente correlati e analizziamone le caratteristiche.
Tra questi si presenta la coppia *htmlimg* e *img*, con correlazione di 0.9557. L'elevato valore della relazione che intercorre tra i suddetti vocaboli è dovuta, principalmente, al fatto che appartengano allo stesso contesto. Difatti, il termine *img* è generalmente impiegato nell'ambito informatico per riferirsi alle immagini. Quest'uso viene effettuato anche in HTML [1], in cui l'elemento utilizzato per mostrare un'immagine è *img*. Nel grafico rappresentato in Figura 13 è evidente l'andamento congiunto delle due variabili, monotono crescente. Infatti, è evidente che al crescere della frequenza di una delle due parole, cresce il volume dell'altra.

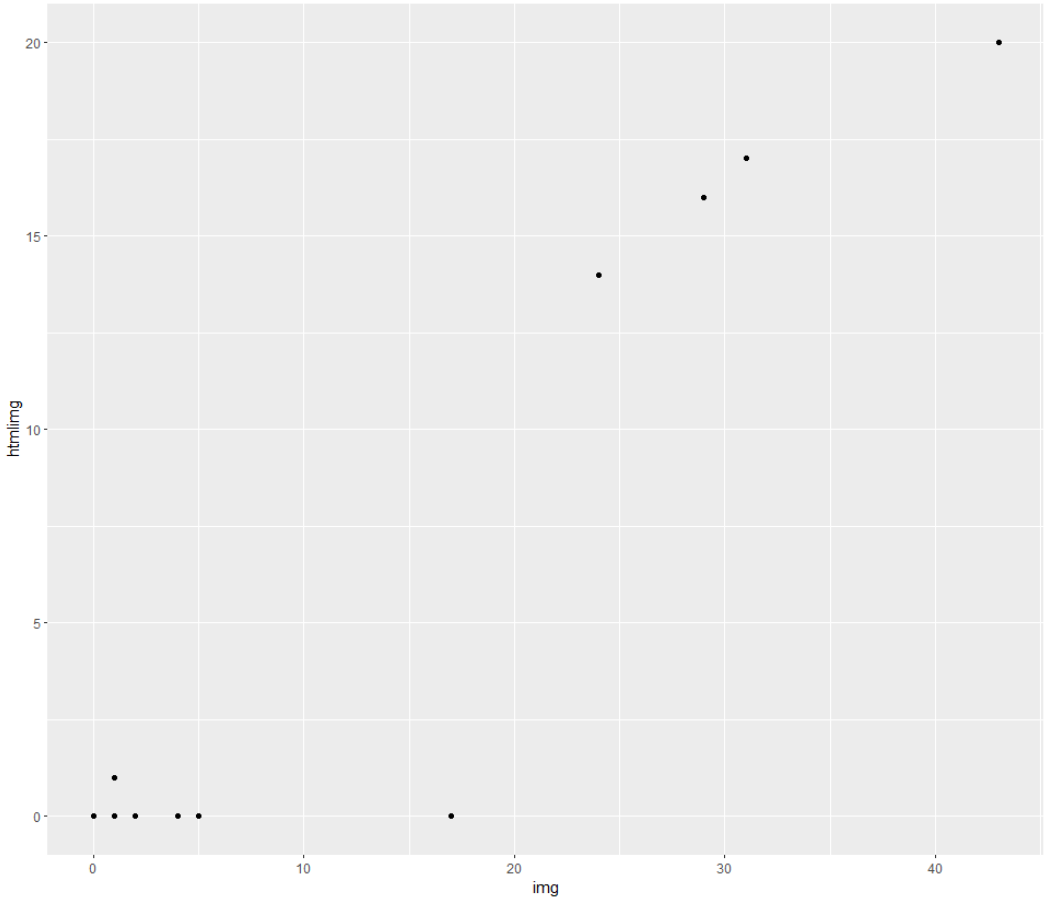

Figura 13: Dispersione di *img* e *htmlimg*.

[1] HTML, ossia HyperText Markup Language, è il linguaggio di annotazione utilizzato per creare pagine web.

Altra coppia dall'andamento interessante è *cellpadding* e *cellspacing*, con correlazione di 0.9040, la cui dispersione è illustrata nella Figura 14. Si tratta sempre di vocaboli adoperati nello stesso contesto. In particolar modo, *cellspacing* e *cellpadding* vengono utilizzati simultaneamente in HTML per controllare la spaziatura tra le celle di una tabella, tra il contenuto delle celle e i bordi delle celle.

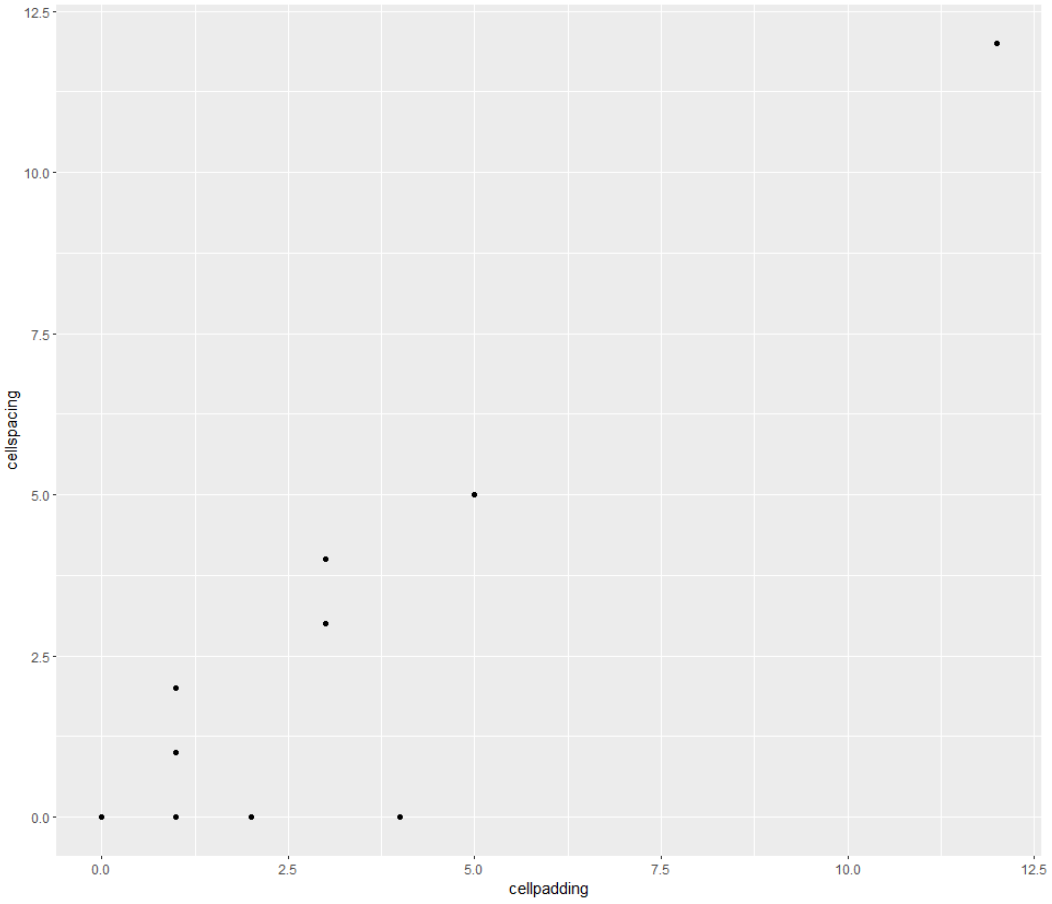

Figura 14: Dispersione di *cellpadding* e *cellspacing*.

Infine, un risultato altrettanto degno di nota è quello recepito con la coppia *female* e *male*, rispettivamente *femmina* e *maschio*, la cui appartenenza al medesimo ambito è del tutto esplicita. Questi presentano correlazione pari a 0.8988, che ne giustifica l'andamento, visibile nella Figura 15.

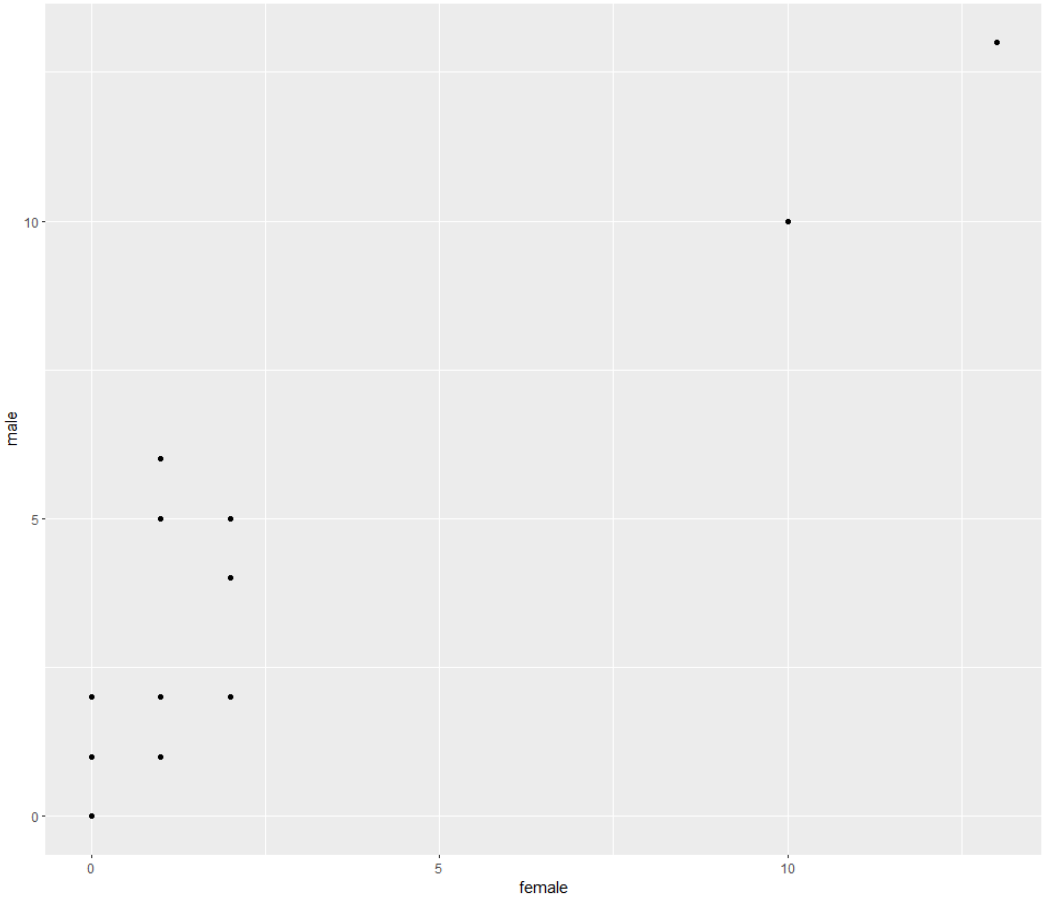

Figura 15: Dispersione di *female* e *male*.

Quindi si nota come molteplici termini, relativi allo stesso dominio, come dimostrato dagli esempi precedenti, siano estremamente correlati. Questo risultato è del tutto attendibile, in quanto molto spesso si individuano coppie di parole impiegate congiuntamente nei testi. Dunque, come osservato nel seguente paragrafo, si indebolisce l'ipotesi sulla quale si basa la Naive Bayes Classification, di cui si è trattato nel capitolo 1, che prevede l'assunzione dell'indipendenza tra la totalità delle parole.

Se, invece, ci si focalizza sui vocaboli che appaiono solo in e-mail reali, questi sono 254, nettamente superiori rispetto alle parole spam.

In questo caso i termini che figurano con frequenza minore sono *felipe*, *dorcheus*, *pooling* ed *offsystem*, con 21 apparizioni, a livello percentuale il 0.065% dell'ammontare totale delle parole ham.
L'espressione più frequente, invece, è *enron*, con frequenza 6906, contribuendo al 21.46% del volume totale delle parole autentiche.
Trattando di correlazione, il risultato visto nel caso delle parole spam è decisamente meno evidente nel caso dei soli lemmi autentici. Difatti, pochi sono i termini a correlazione elevata. Si tratta principalmente di coppie di nomi e cognomi. Esempi di quest'ultimi sono le associazioni *aimee* e *lannou*, con correlazione 0.9589, *eileen* e *ponton*, con correlazione pari a 0.9337, e, infine *christy* e *sweeney*, con correlazione pari a 0.9287. In generale, dunque, le associazioni tra parole autentiche risultano più deboli, come emerge dalla Figura 16.

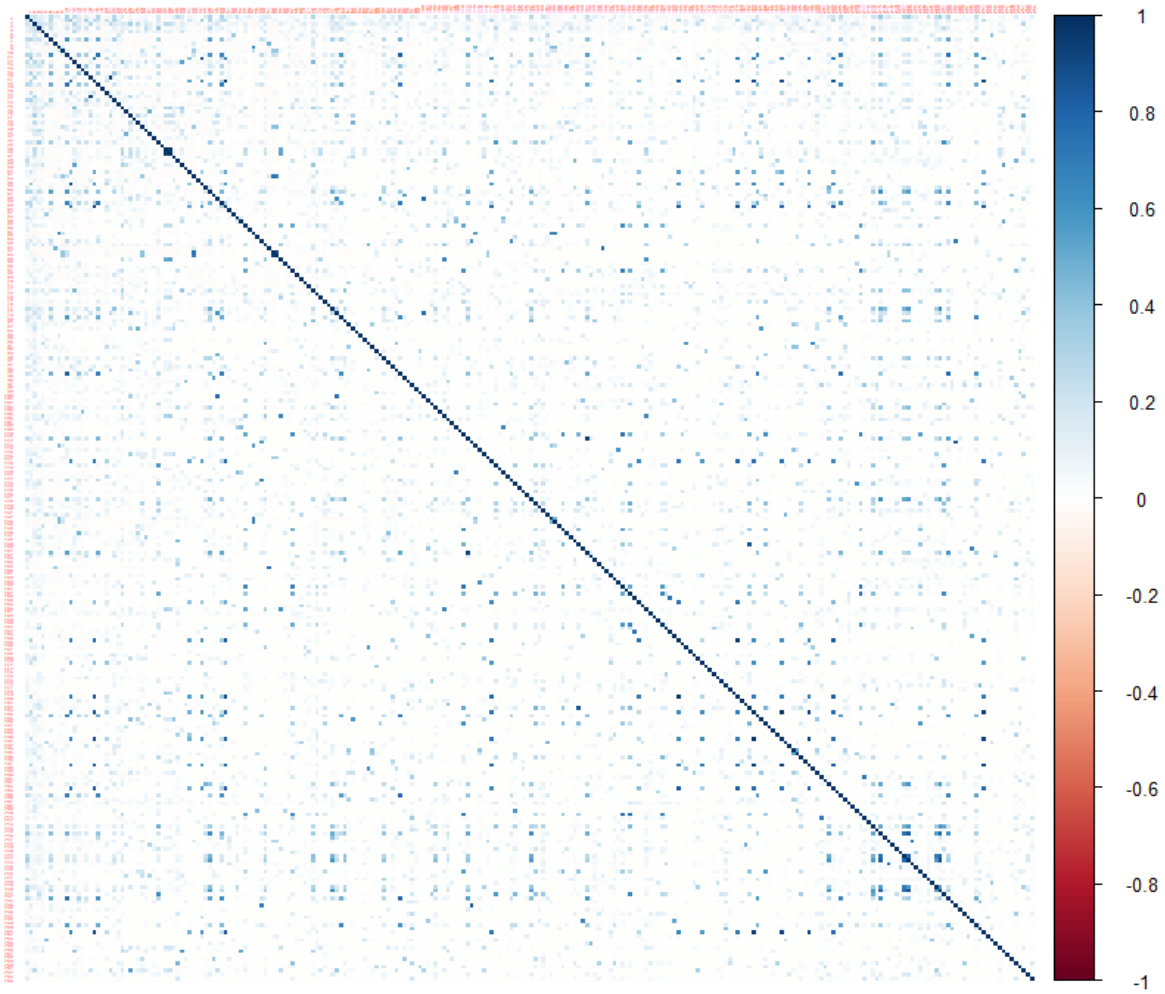

Figura 16: Grafico di correlazione delle parole ham.

2.4 Analisi delle “spam trigger words"

In questo paragrafo, si procederà analizzando vocaboli ed espressioni che scatenano il meccanismo di filtraggio e la conseguente classificazione dell'e-mail come fraudolenta, in maniera tale da osservarne il comportamento nell'insieme di dati oggetto di studio. Di tali termini si considereranno gli aspetti descrittivi, le relazioni che intercorrono tra di loro ed il comportamento rispetto alla variabile risposta.

Come esplicitato nel capitolo 1, molteplici filtri antispam identificano la posta elettronica illegittima a partire dai vocaboli impiegati nei testi di tali messaggi. I suddetti termini, generalmente, appartengono a contesti specifici [17]. Uno di questi, e probabilmente quello principale, è l'ambito economico-finanziario. Innumerevoli testi di e-mail spam contengono, difatti, annunci relativi alla compravendita di prodotti, offerte riguardanti opportunità di investimento o vincite di denaro e richieste di aiuto a livello finanziario. Esempi di questa categoria all'interno del dataset oggetto di analisi sono termini quali *credit*, *investment*, *money*, *offer* e *deal*. Le statistiche descrittive relative a queste variabili sono presentate nella tabella 6.

	credit	investment	money	offer	deal
Min.	0.0000	0.0000	0.0000	0.0000	0.0000
1st Qu.	0.0000	0.0000	0.0000	0.0000	0.0000
Median	0.0000	0.0000	0.0000	0.0000	0.0000
Mean	0.0375	0.0690	0.0706	0.1083	0.7345
3rd Qu.	0.0000	0.0000	0.0000	0.0000	0.0000
Max.	7.0000	11.0000	8.0000	9.0000	25.0000

Tabella 6: Statistiche descrittive di *credit*, *investment*, *money*, *offer*, *deal*.

Partendo dall'espressione *credit*, com'è possibile visualizzare dal grafico 17, questa figura in e-mail spam con frequenza maggiore rispetto alle autentiche. Stesso comportamento si può notare anche nel caso di *investment*, *money* e *offer*, in cui tale differenza, visibile nelle figure 18, 19 e 20, risulta ancora più netta. L'unico vocabolo che si distingue, è, quindi, *deal*, che, come visibile nel grafico illustrato in Figura 21, risulta prevalere nei messaggi reali.
Considerando congiuntamente tali variabili, data la loro appartenenza al medesimo settore, vi è la possibilità che siano correlate. Quest'ultimo, tuttavia, è un risultato da escludere, in quanto si nota dal grafico riportato in Figura 22 che la relazione che intercorre tra questi vocaboli è particolarmente lieve.

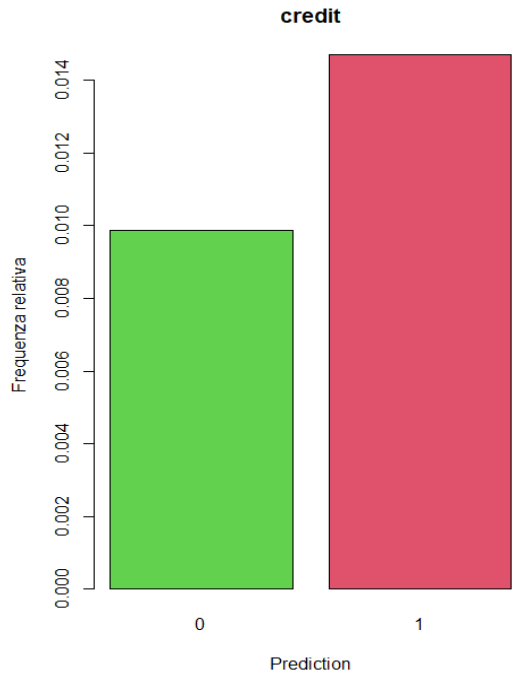

Figura 17: Frequenza *credit*.

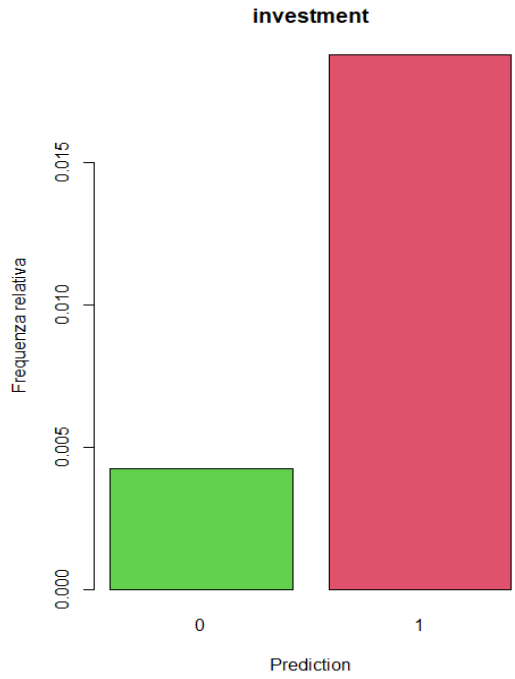

Figura 18: Frequenza *investment*.

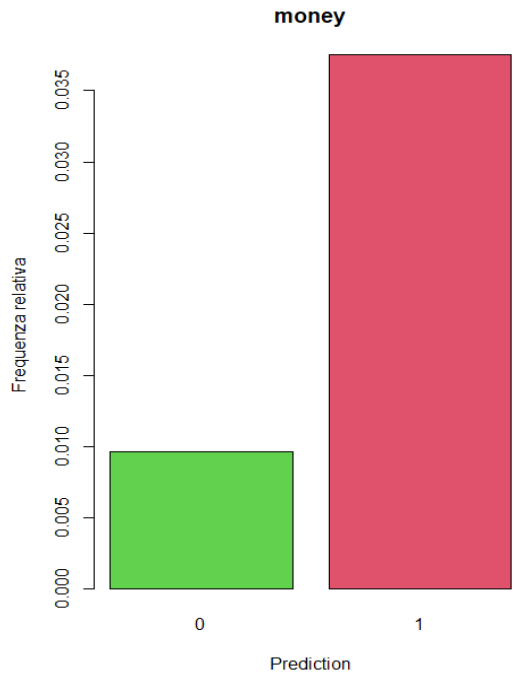

Figura 19: Frequenza *money*.

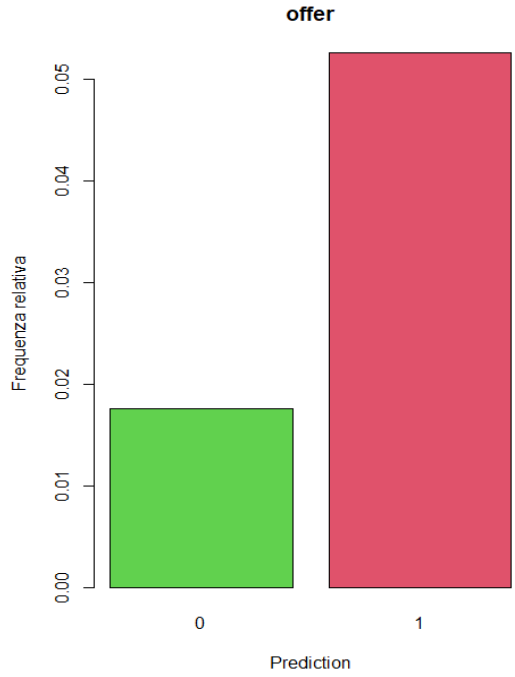

Figura 20: Frequenza *offer*.

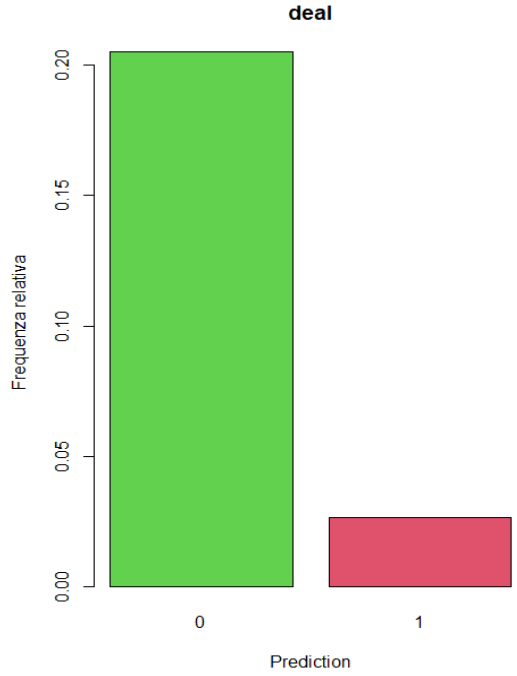

Figura 21: Frequenza *deal*.

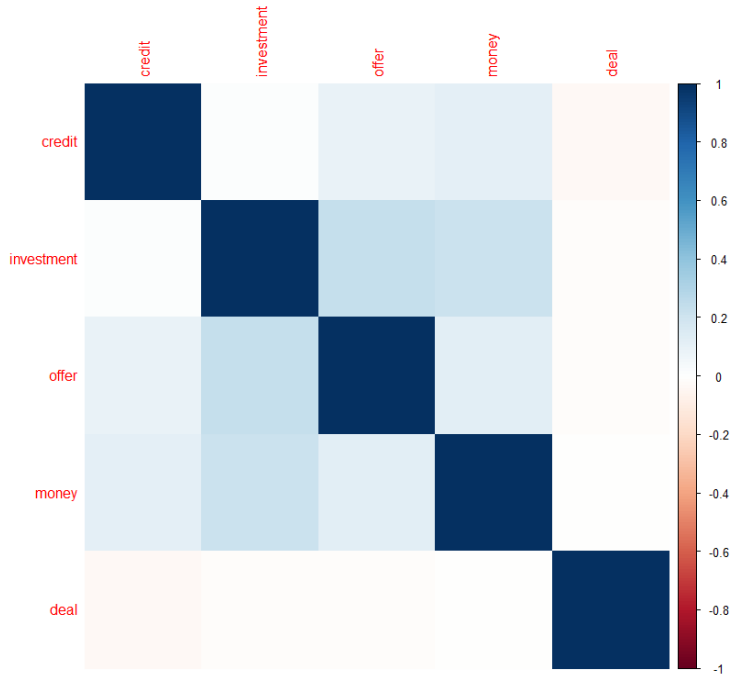

Figura 22: Grafico di correlazione della categoria economica.

Risulta estremamente frequente, inoltre, riscontrare all'interno delle e-mail spam la presenza di molteplici termini appartenenti al campo medico-farmaceutico. Quest'ultimi figurano fondamentalmente in messaggi relativi alla compravendita di farmaci o prodotti medici senza la necessità prescrizione medica e risparmiando denaro.
Nell'insieme dei dati oggetto di studio, appartengono a questa classe i vocaboli *medications*, *drugs* e *treatment*, i cui aspetti descrittivi sono contenuti nella tabella 7.

	medications	drugs	treatment
Min.	0.0000	0.0000	0.0000
1st Qu.	0.0000	0.0000	0.0000
Median	0.0000	0.0000	0.0000
Mean	0.0122	0.0226	0.0085
3rd Qu.	0.0000	0.0000	0.0000
Max.	3.0000	5.0000	2.0000

Tabella 7: Statistiche descrittive di *medications*, *drugs*, *treatment*.

Tutti e tre i vocaboli hanno un andamento piuttosto simile, visibile nelle figure 23, 24 e 25, caratterizzato dalla netta prevalenza in testi di posta elettronica fraudolenta. In particolar modo, la variabile *drugs* presenta un comportamento estremo, in quanto figura esclusivamente in e-mail spam nell'insieme dei dati analizzati.
Anche in questo caso non si evidenzia correlazione significativa tra i tre termini, nonostante l'appartenenza al medesimo ambito, com'è possibile dedurre dal grafico delle correlazioni riportato in Figura 26.

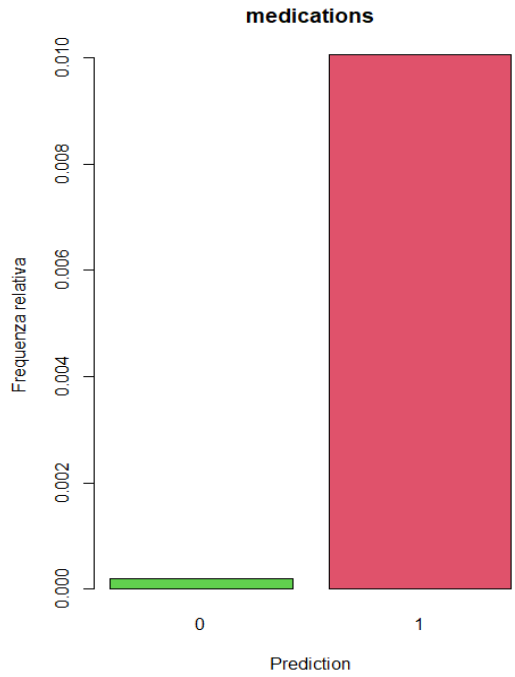

Figura 23: Frequenza *medications*.

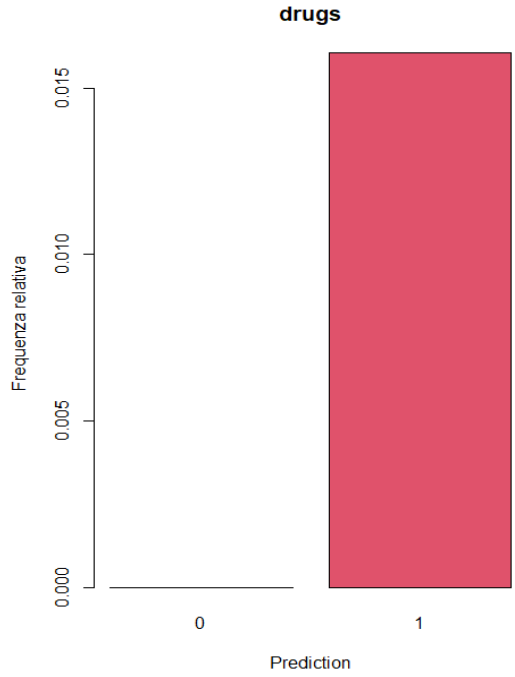

Figura 24: Frequenza *drugs*.

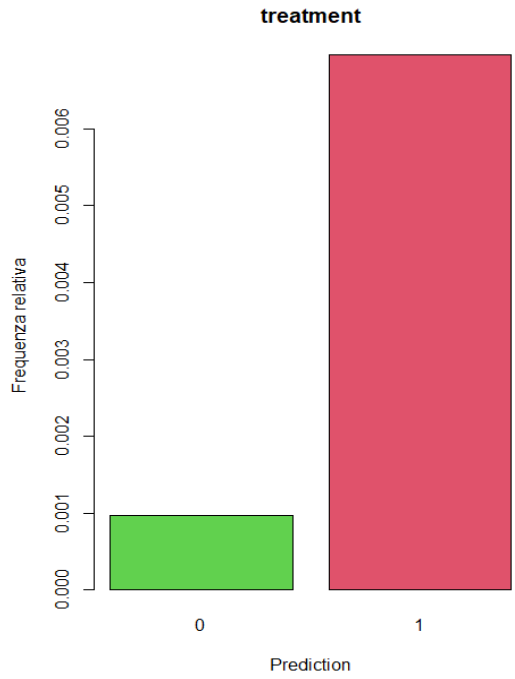

Figura 25: Frequenza *treatment*.

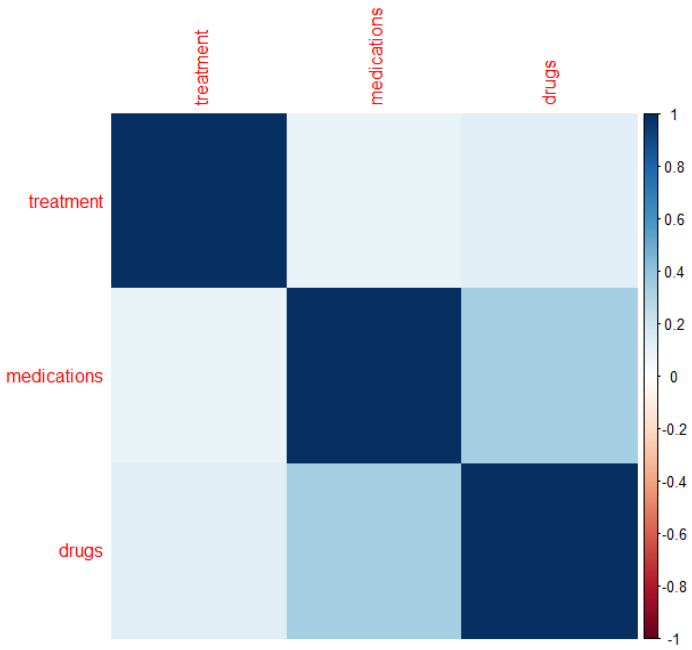

Figura 26: Grafico di correlazione della categoria medico-farmaceutica.

Molto spesso, nella posta elettronica fraudolenta, si notano, come anticipato nel capitolo 1, inesattezze a livello ortografico. Quest'ultime riguardano, principalmente, vocaboli riportati in maniera scorretta. Alcuni esempi di questa categoria di parole nella raccolta dei dati presa in considerazione sono i termini *newsietter*, *pubiisher* e *technoiogies*. La versione corretta delle parole menzionate sarebbe, in ordine, *newsletter*, *publisher* e *technologies*. Gli aspetti descrittivi delle tre variabili sono contenuti nella tabella 8.

	newsietter	pubiisher	technoiogies
Min.	0.0000	0.0000	0.0000
1st Qu.	0.0000	0.0000	0.0000
Median	0.0000	0.0000	0.0000
Mean	0.0044	0.0060	0.0054
3rd Qu.	0.0000	0.0000	0.0000
Max.	3.0000	4.0000	6.0000

Tabella 8: Statistiche descrittive di *newsietter*, *pubiisher*, *technoiogies*.

Le frequenze di tutte e tre le parole considerate, come evidente nelle figure 27, 28 e 29, sono particolarmente basse. Inoltre, la totalità delle apparizioni delle suddette variabili si concentra nelle e-mail spam. Il grafico rappresentato in Figura 30 delinea il livello di associazione esistente tra i tre termini. Si nota come la correlazione più significativa sia quella esistente tra *technoiogies* e *pubiisher*.

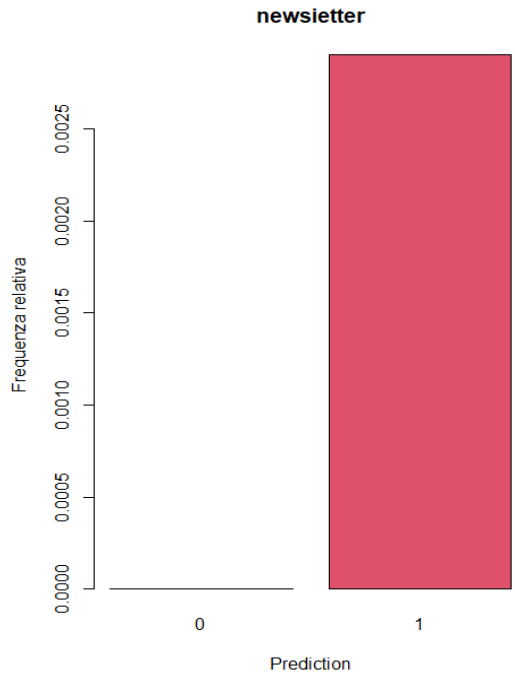

Figura 27: Frequenza *newsietter*.

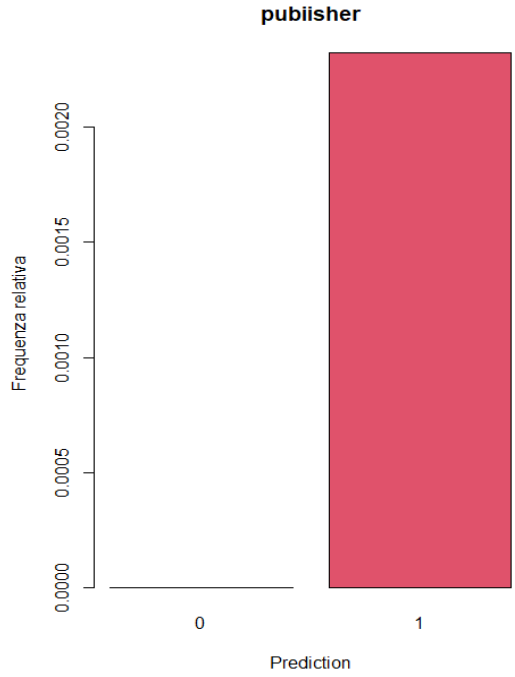

Figura 28: Frequenza *pubiisher*.

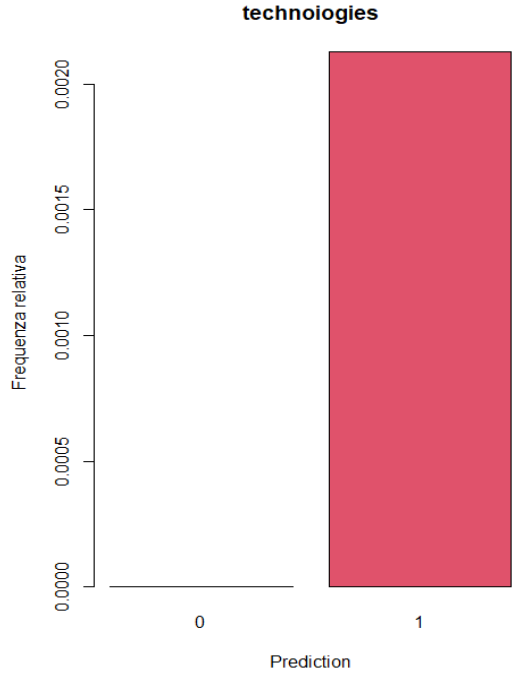

Figura 29: Frequenza *technoiogies*.

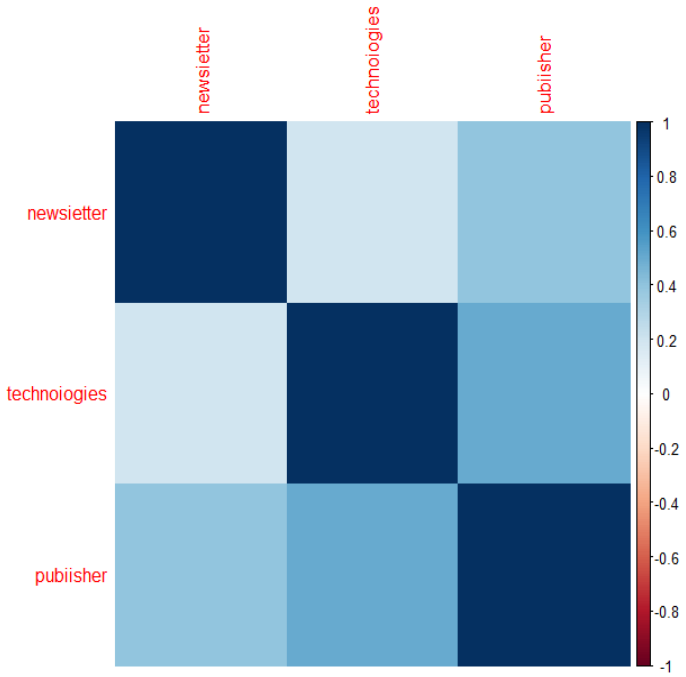

Figura 30: Grafico di correlazione della categoria relativa all'ortografia.

In conclusione, come descritto nel capitolo 1, numerosi messaggi illegittimi invitano il destinatario ad accedere a link fraudolenti o a scaricare materiale dannoso. Termini appartenenti a tale ambito, nel caso d'interesse, sono *click*, *open* e *attached*, le cui statistiche descrittive sono raccolte nella tabella 9.

	click	open	attached
Min.	0.0000	0.0000	0.0000
1st Qu.	0.0000	0.0000	0.0000
Median	0.0000	0.0000	0.0000
Mean	0.1025	0.0425	0.2121
3rd Qu.	0.0000	0.0000	0.0000
Max.	14.0000	10.0000	7.0000

Tabella 9: Statistiche descrittive di *click*, *open*, *attached*.

Appare piuttosto evidente, nelle figure 31 e 32, che *click* e *link* abbiano un comportamento affine, in quanto si presentano in entrambe le categorie di messaggi, ma con lieve predominanza nelle e-mail spam.
D'altra parte, *attached* si contraddistingue per l'elevato volume all'interno dei messaggi reali, nettamente superiore rispetto alla frequenza di tale termine all'interno della posta elettronica spam, aspetto deducibile dal grafico riportato in Figura 33.
In termini di correlazione, questa risulta considerevole tra *click* e *link*, com'è naturale attendersi, dato che si tratta di vocaboli usati spesso congiuntamente, mentre *attached* presenta un'associazione leggermente negativa con entrambe le altre variabili. Tali risultati sono riportati in Figura 34.

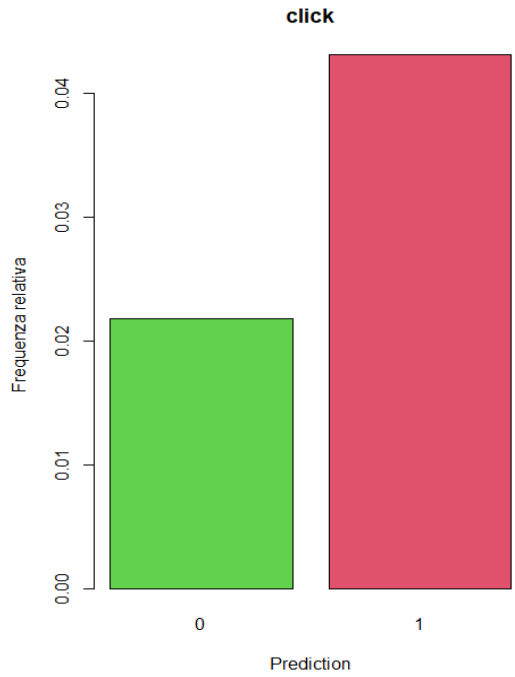

Figura 31: Frequenza *click*.

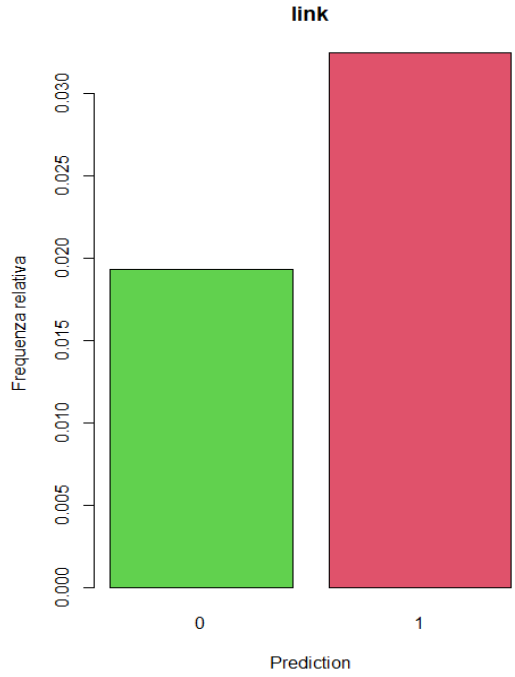

Figura 32: Frequenza *link*.

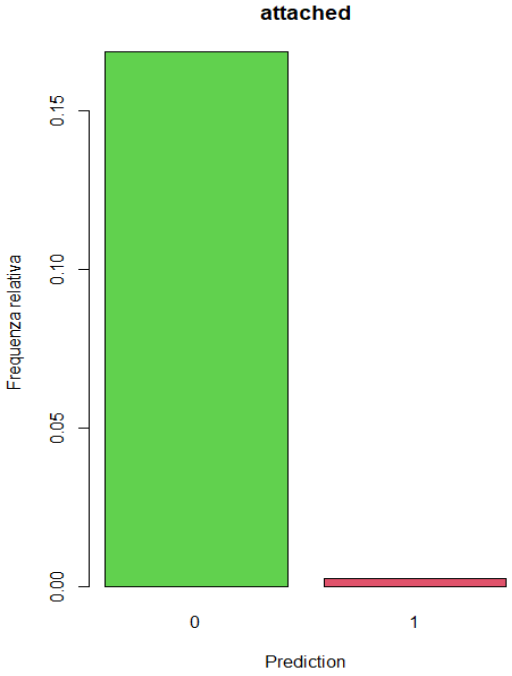

Figura 33: Frequenza *attached*.

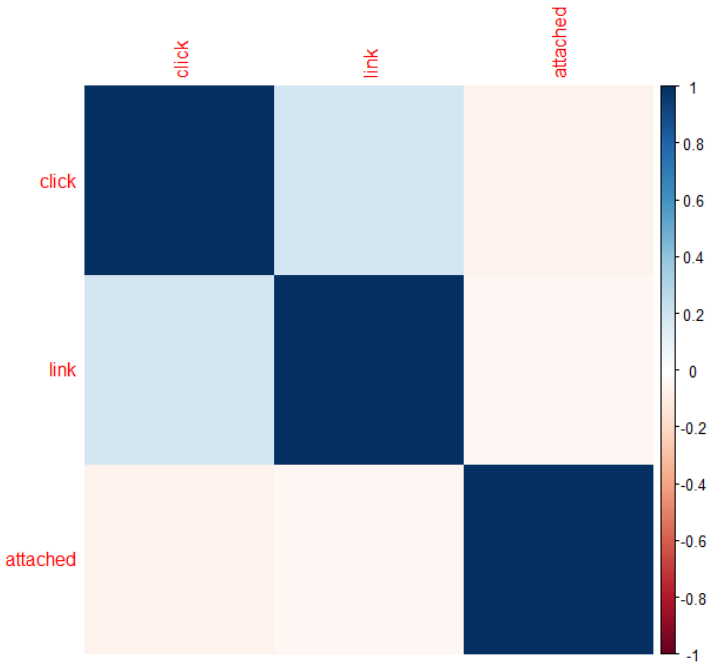

Figura 34: Grafico di correlazione della categoria relativa al web.

3 I metodi di clustering

3.1 Introduzione

Per clustering dei dati si intende il processo volto ad identificare raggruppamenti naturali, definiti cluster, sulla base di alcune misure di similarità. Si tratta di una metodologia impiegata in vari contesti, in particolar modo nel machine learning e nell'ambito del data mining [4].

Il problema del clustering può essere delineato come segue. Dato il dataset $\mathbf{X}=\{x_1, x_2,..., x_p\}$, con $x_j=\{x_{j,1}, ..., x_{j,n}\}$, $j=1,...,p$, vettore di n osservazioni appartenenti a $\mathbf{X}$. Si definisce il clustering di $\mathbf{X}$ come la suddivisione di tale insieme in K gruppi $\{C_1, C_2,...,C_K\}$, dati i seguenti requisiti:

- Ciascuna osservazione deve essere assegnata ad un cluster, cioè

$$\bigcup_{k=1}^{K} C_k = \mathbf{X}$$

- Ciascun cluster deve contenere almeno un'osservazione

$$C_k \neq \emptyset \qquad \text{con k=1,..., K.}$$

- Ciascuna osservazione è assegnata ad un unico cluster

$$C_k \cap C_q = \emptyset \qquad \text{dove } k \neq q.$$

3.2 Misure di similarità

Come esplicitato precedentemente, l'identificazione dei cluster avviene basandosi su una qualche misura di similarità. Queste, infatti, stanno alla base dei fondamentali algoritmi di clustering [2] [4].
La metodologia maggiormente adottata per valutare la presenza di similarità prevede l'utilizzo di misure di distanza.
La distanza euclidea risulta essere quella più impiegata, e, date x_j e x_t variabili costituite da n osservazioni, viene definita come

$$d(x_j, x_t) = \sqrt{\sum_{i=1}^{n}(x_{j,i} - x_{t,i})^2} = \sqrt{(x_j - x_t)^T(x_j - x_t)}.$$

Si tratta di un caso particolare della distanza di Minkowski, ossia

$$d_a(x_j, x_t) = \left[\sum_{i=1}^{n} |x_{j,i} - x_{t,i}|^a\right]^{1/a}$$

con a =2.

Nel momento in cui, invece, a è unitario, ossia a=1, allora, a partire dalla distanza di Minkowski, si ricava la distanza di Manhattan

$$d(x_j, x_t) = \sum_{i=1}^{n} |x_{j,i} - x_{t,i}|.$$

Ulteriore misura di distanza è quella di Mahalanobis, ottenuta applicando la seguente formula

$$d(x_j, x_t) = \sqrt{(x_j - x_t)^T \mathbf{\Sigma}^{-1} (x_j - x_t)}$$

con $\mathbf{\Sigma}^{-1}$ l'inversa della matrice di covarianza di x_j e x_t.

3.3 Tipologie di clustering

3.3.1 Il clustering gerarchico

I principali algoritmi di clustering si basano su due tecniche note come clustering gerarchico e partizionale [4].
L'applicazione del metodo gerarchico produce come risultato un albero di cluster, definito dendogramma, tramite l'utilizzo di processi di suddivisione o di fusione euristica. Gli algoritmi che utilizzano la suddivisione per generare il dendogramma si denotano come divisivi. Tuttavia, le tecniche maggiormente utilizzate impiegano la fusione per generare l'albero di cluster. Quest'ultimi sono noti come metodi agglomerativi.

Le procedure di clustering divisivo procedono assegnando tutti gli oggetti ad un unico cluster di riferimento. Successivamente, viene applicata la suddivisione. I gruppi risultanti vengono a loro volta divisi in ulteriori sottogruppi, ripetendo, quindi, l'operazione fino a quando ciascun cluster è costituito da una singola osservazione.

In modo opposto rispetto alla metodologia analizzata precedentemente, gli algoritmi gerarchici agglomerativi si avviano attribuendo ciascun elemento ad un

cluster. In secondo luogo, i due gruppi che risultano essere più simili vengono uniti. Tale processo viene rieseguito finchè tutti gli elementi non vengono assegnati ad un singolo cluster.
Le differenti tipologie di clustering gerarchico agglomerativo si distinguono sulla base delle tecniche impiegate per individuare i due insiemi più simili. Gli algoritmi più utilizzati sono quelli a legame singolo, a legame completo, e, infine, a legame medio. Tali approcci si basano sulla fusione di cluster in cui la distanza tra i relativi elementi è la minima possibile. Viene, in seguito, ricalcolata la suddetta misura secondo diverse modalità.
Nell'ambito del legame singolo, la distanza viene ridefinita come il valore minimo delle distanze tra le osservazioni appartenenti agli insiemi di riferimento. Dati, dunque, i gruppi G_1 e G_2, con n_{g1} ed n_{g2} elementi rispettivamente, la distanza viene calcolata come

$$d(G_1, G_2) = min\{d(x_i, x_f) : x_i \in G_1, x_f \in G_2\},$$

con x_i e x_f due generiche osservazioni.

Nel caso, invece, del legame completo avviene l'unione dei cluster che risultano essere più vicini, la cui distanza viene ridefinita come la massima tra gli elementi considerati

$$d(G_1, G_2) = max\{d(x_i, x_f) : x_i \in G_1, x_f \in G_2\}.$$

D'altra parte, trattando del legame medio, si procede aggregando i gruppi più prossimi, riformulandone la distanza come media tra le misure di riferimento

$$d(G_1, G_2) = \frac{1}{n_{g1} n_{g2}} \sum_{x_i \in G_1} \sum_{x_f \in G_2} d(x_i, x_f).$$

In alternativa, è possibile utilizzare il metodo di Ward. Questo ha come obiettivo quello di raggruppare le unità, massimizzando la proporzione di varianza spiegata. Quindi, considerando $x_{k,i,j}$, osservazione i relativa alla variabile j ed appartenente al cluster k. Date $\bar{x}_{k,\cdot,j}$, media relativa a ciascuna variabile per ogni cluster, e $\bar{x}_{\cdot,\cdot,j}$ media rispetto ad ogni variabile, è possibile definire *ESS*, somma dei quadrati degli errori, *TSS*, somma dei quadrati totale, e r^2, proporzione di varianza spiegata dal clustering, come

$$ESS = \sum_k \sum_i \sum_j |x_{k,i,j} - \bar{x}_{k,\cdot,j}|^2 \tag{6}$$

$$TSS = \sum_k \sum_i \sum_j |x_{k,i,j} - \bar{x}_{\cdot,\cdot,j}|^2 \tag{7}$$

$$r^2 = \frac{TSS - ESS}{TSS}.$$

Il metodo di Ward ha, dunque, come obiettivo quello di massimizzare r^2.

Gli algoritmi di clustering gerarchico presentano molteplici vantaggi, tra cui il principale risulta essere l'assenza della necessità di specificare a priori il numero di insiemi.
Tuttavia, si tratta di approcci complessi a livello computazionale, per cui poco adatti per l'analisi di dataset ad elevata numerosità. Inoltre, i suddetti algoritmi sono statici, tali per cui oggetti assegnati ad un determinato insieme non possono essere spostati ad un altro. Infine, potrebbero trovare difficoltà nel distinguere cluster sovrapposti a causa della mancanza di informazioni sulla forma o sulla dimensione degli stessi [4].

3.3.2 Clustering partizionale

Le tecniche di clustering partizionale prevedono la suddivisione dei dati in un numero predeterminato di insiemi. Si tratta, fondamentalmente, di algoritmi iterativi che convergono a ottimi locali, i cui passaggi risultano essere i seguenti:

- Si inizializzano casualmente i centroidi dei k cluster di riferimento.
- Per ciascun oggetto x_j nel dataset si calcola l'appartenenza $u(m_k|x_j)$ all'insieme di centroide m_k ed il relativo peso $\mathrm{w}(x_j)$.
- Si ricalcolano, in seguito, i k centroidi utilizzando

$$m_k = \frac{\sum_{j=1}^{p} u(m_k|x_j) w(x_j) x_j}{\sum_{j=1}^{p} u(m_k|x_j) w(x_j)}.$$

Gli ultimi due punti vengono ripetuti fino a quando non viene soddisfatto uno specifico criterio di arresto.
Nella procedura precedente, $\mathrm{u}(m_k|x_j)$ è la funzione che quantifica l'appartenenza dell'elemento x_j al cluster k. Quest'ultima deve soddisfare i seguenti vincoli

$$u(m_k|x_j) \geq 0$$

$$\sum_{k=1}^{K} u(m_k|x_j) = 1$$

Il peso w(x_j), invece, definisce quanta influenza ha l'oggetto x_j nel ricalcolo dei centroidi nella successiva iterazione, dove $w(x_j) > 0$.

L'algoritmo partizionale più diffuso si basa sul metodo delle k-medie, definito alternativamente *k-means*. Si tratta di una metodologia che ha come obiettivo quello di minimizzare la distanza intra-cluster, ottimizzando la seguente funzione

$$J_{K-means} = \sum_{k=1}^{K} \sum_{\forall x_j \in C_k} d^2(x_j, m_k).$$

Tale procedura prende avvio con k centroidi, i cui valori iniziali sono selezionati casualmente o derivanti da informazioni a priori. In seguito, ogni dato viene assegnato al cluster il cui centroide risulta essere più vicino all'oggetto stesso. In conclusione, i centroidi vengono ricalcolati secondo gli schemi associati. Tale processo viene ripetuto sino al raggiungimento della convergenza.

La funzione peso nella metodologia k-means risulta essere costante, in quanto tutti gli elementi assumono la medesima importanza.
L'impiego di tale tecnica presenta molteplici vantaggi, quali la facilità di implementazione, da cui deriva anche la possibilità di applicarla a dataset di dimensioni considerevoli. D'altra parte, però, risulta necessario specificare il numero di cluster a priori. Infatti, si tratta di un algoritmo che dipende dalle condizioni iniziali, che possono condurre alla convergenza verso soluzioni subottimali [4] [2].

3.4 Valutazione della classificazione

Il problema di valutazione dei cluster è volto a determinare il numero ottimale di gruppi per un determinato insieme di dati [4] [2].
Due sono i criteri impiegati per misurare la qualità della partizione dei dati, ossia:

- Compattezza: gli elementi all'interno di un cluster devono essere simili tra di loro. L'insieme, deve, dunque, risultare omogeneo.
- Separazione: i cluster devono essere distinti. Ci deve essere, quindi, eterogeneità tra i gruppi.

Alcuni algoritmi di clustering, in particolare quelli partizionali, dettano la necessità di stabilire il numero di cluster k in cui suddividere i dati a priori. L'identificazione di k può essere facilitata da informazioni che si hanno a disposizione sul dataset. In alternativa, è possibile impiegare approcci basati sui dati stessi [5].
Tra questi vi è il metodo dei *gomiti*. Esso si basa sul calcolo della varianza spiegata da differenti numerosità di k e la sua rappresentazione grafica. Si sceglie il numero di gruppi che corrisponde al *gomito* del suddetto grafico, ossia il punto in cui si ottiene una elevata inclinazione della curva, in quanto rappresenta la suddivisione per cui avviene un incremento notevole dell'informazione spiegata. Quest'ultima si ottiene dalla seguente espressione

$$varianza \;\; spiegata = \frac{betweenSS}{totSS} \tag{8}$$

$$betweenSS = \sum_{k} (\bar{x}_k - \bar{x})^2 \tag{9}$$

$$totSS = \sum_{k,i} (x_{k,i} - \bar{x})^2. \tag{10}$$

con $x_{k,i}$ osservazione i-esima appartenente al k-esimo insieme, $\bar{x}$ media generale e $\bar{x}_k$ media del k-esimo cluster.

Approccio alternativo è rappresentato dalla statistica *Gap*. Si tratta di una metodologia che prevede il confronto tra la somma dei quadrati interni ai cluster (9) per diversi valori di k nel dataset originale e la medesima misura sull'insieme di dati permutato. Si sceglie il valore k che massimizza la statistica *Gap*, ottenuta effettuando la differenza tra le due suddette quantità.

La classificazione gerarchica, invece, non richiede un'iniziale definizione del numero di gruppi. Infatti, in questo caso, si effettua la suddivisione, che viene in seguito rappresentata graficamente tramite il dendogramma. Quest'ultimo permette di visualizzare il numero di cluster in cui sono state ripartite le osservazioni.

Una volta effettuato il clustering, al fine di valutare la correttezza del raggruppamento, è possibile utilizzare la *silhouette*, indipendentemente dalla metodologia adottata. Si tratta di un indicatore che misura quanto un oggetto appartenga al suo cluster, rispetto agli altri insiemi, valutando, così, la qualità della separazione dei gruppi. Assume valori compresi nell'intervallo [-1,1], in cui

- un valore prossimo a 1 indica che l'elemento è correttamente assegnato al proprio insieme e che è distante dai punti dei cluster vicini.

- un valore prossimo a 0 indica che il punto è collocato al confine tra due cluster.
- un valore negativo suggerisce una collocazione erronea dell'elemento.

Tale indicatore si esprime come

$$S(x_i) = \frac{D_0 - D(x_i, R_g)}{max\{D_0, D(x_i, R_g)\}}$$

dove D(x_i, R_g) è la distanza dell'osservazione x_i, appartenente all'insieme R_i, dal gruppo R_g con n_g elementi, cioè

$$D(x_i, R_g) = \frac{1}{n_g} \sum_{x_g \in R_g} d(x_i, x_g).$$

D_0, d'altra parte, indica la distanza dell'osservazione x_i dal gruppo più vicino differente rispetto a quello di appartenenza. Essendo R_i il cluster in cui risiede l'elemento x_i,

$$D_0 = \min_{g \neq i} D(x_i, R_g).$$

3.5 Clustering sparso

Una metodologia alternativa per la classificazione dei dati in gruppi risulta essere il clustering sparso. Quest'ultimo viene applicato nel momento in cui si ritiene che i cluster sottostanti differiscano solo rispetto ad alcune caratteristiche, in quanto permette di suddividere le osservazioni sulla base di un definito sottoinsieme di variabili. Inoltre, è un insieme di metodi che permettono di ottenere un'identificazione degli insiemi molto più accurata rispetto alle metodologie standard di clustering, specialmente per dati che presentano un numero di variabili particolarmente elevato, o addirittura maggiore rispetto alle unità statistiche [1].

Come per i casi analizzati in precedenza, anche il clustering sparso necessita dell'assunzione di una qualche misura di similarità, che può essere selezionata tra quelle definite nel paragrafo 3.2.
In particolare, $d(x_i, x_f)$ denota la distanza tra le osservazioni i e f. Quest'ultima è additiva rispetto alle caratteristiche, in quanto deriva dalla somma delle distanze nelle singole variabili, ossia

$$d(x_i, x_f) = \sum_{j=1}^{p} d_{i,f,j}$$

in cui $d_{i,f,j}$ è una misura di similarità tra le unità i e f rispetto alla caratteristica j e p è il numero di variabili totali.

Si consideri un dataset $\mathbf{X}$ contenente n unità statistiche e p variabili. Un generico problema di clustering può essere espresso come

$$\max_{\Theta \in D} \sum_{j=1}^{p} f_j(X_j, \Theta)$$

con $f_j(X_j, \Theta)$ funzione che coinvolge solo la j-esima variabile e Θ parametro appartenente all'insieme D.

Si definisce il clustering sparso come estensione del suddetto problema di ottimizzazione nella seguente forma

$$\max_{w;\, \Theta \in D} \sum_{j=1}^{p} w_j f_j(X_j, \Theta) \qquad \text{dato}$$

$$\sum_{j=1}^{p} w_j^2 \leq 1, \qquad \sum_{j=1}^{p} w_j \leq s, \qquad w_j \geq 0, \tag{11}$$

dove w_j è il peso corrispondente alla caratteristica j, interpretabile come contributo della variabile j al risultante clustering sparso.
D'altra parte, s è un parametro di ottimizzazione tale per cui $1 \leq s \leq \sqrt{p}$. Quest'ultimo monitora il numero di variabili che verranno coinvolte nel processo, garantendo, così, la sparsità dei risultati.

Nel momento in cui vi è corrispondenza tra i pesi, ossia $w_1 = \ldots = w_p$, allora si ricade nell'ambito del clustering tradizionale. Infatti, data la finalità di sparsità nel processo di classificazione, si prendono in considerazione solo alcune delle variabili del dataset. Per perseguire tale obiettivo, quindi, è necessario che alcuni pesi w_j risultino nulli. Dunque, la nullità del peso w_j suggerisce che la caratteristica j non viene coinvolta nel raggruppamento.
A determinare quali variabili non risultano significative per la classificazione dei dati è la *Lasso penalty* [7]. Si tratta di una tecnica di regressione che viene generalmente impiegata al fine di selezionare determinate variabili e ridurre la complessità di un modello, permettendo, così, la sparsità. Questa risulta particolarmente utile in contesti in cui si ritiene che solo un sottoinsieme delle variabili coinvolte in un modello sia effettivamente rilevante. Nell'ambito del clustering sparso, tale

penalità viene applicata ai pesi w=$\{w_1,...,w_p\}$ e dipende dal valore assunto dal parametro s. Infatti, applicare la *Lasso penalty* sui pesi risulta nella nullità di alcuni di essi e la significatività di altri. Il numero di variabili coinvolte nella classificazione viene stabilito, quindi, dal parametro s, il cui valore dipende dalla tipologia di clustering applicata.

3.5.1 Il clustering gerarchico

Similmente al clustering gerarchico tradizionale, analizzato nel paragrafo 3.3.1, anche le tecniche di classificazione sparsa si basano su una matrice di similarità **U**, che corrisponde a

$$\mathbf{U} = \{\sum_j d_{i,f,j}\}_{i,f}.$$

Al fine di ottenere sparsità nei risultati, invece che utilizzare la tradizionale matrice di similarità **U**, ne si impiega una versione ponderata. L'applicazione del clustering gerarchico alla matrice pesata si traduce nella sparsità, in quanto la nullità di alcuni pesi conduce automaticamente all'esclusione di alcune caratteristiche dalla classificazione [1].

Un tradizionale problema di clustering gerarchico può essere espresso come

$$max_{\mathbf{U}}\{\ \textstyle\sum_j \sum_{i,f} d_{i,f,j} U_{i,f}\}$$

con vincolo

$$\sum_{i,f} U_{i,f}^2 \leq 1.$$

Per perseguire la finalità di sparsità, si modifica l'espressione precedente come segue

$$max_{\mathbf{w},\mathbf{U}}\{\ \textstyle\sum_j w_j \sum_{i,f} d_{i,f,j} U_{i,f}\}$$

con vincoli

$$\sum_{i,f} U_{i,f}^2 \leq 1, \qquad \sum_{j=1}^{p} w_j^2 \leq 1, \qquad \sum_{j=1}^{p} w_j \leq s, \qquad w_j \geq 0 \quad \forall j. \tag{12}$$

Si selezionano, quindi, solo determinate variabili, applicando, come visto in precedenza, la penalità di tipo Lasso sui pesi w.
Si nota come la formula (12) è un caso particolare della (11), con

$$\Theta = \mathbf{U},$$
$$f_j(X_j, \Theta) = \sum_{i,f} d_{i,f,j} U_{i,f}.$$

Dunque, in generale, si applicano semplicemente le metodologie di clustering gerarchico alla matrice pesata $\mathbf{U}$, ottenendo risultati che dipendono da un sottoinsieme di caratteristiche.

3.5.2 Clustering partizionale: metodo K-means

La classificazione che si basa sul metodo delle k-medie ha come obiettivo, assumendo come misura di similarità la distanza euclidea, quello di minimizzare la somma dei quadrati interna ai cluster, ossia la distanza euclidea tra le osservazioni interne ai gruppi

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,f \in C_k} \sum_{j=1}^{p} d_{i,f,j},$$

con n_k numero di osservazioni appartenenti all'insieme k e C_k k-esimo cluster.
Minimizzare la somma dei quadrati interna ai cluster equivale alla massimizzazione della somma dei quadrati tra gli insiemi, cioè la distanza esistente tra di essi, che corrisponde a

$$max_{C_1,\ldots,C_K,\mathrm{w}} \{ \textstyle\sum_{j=1}^{p} w_j (\frac{1}{n} \sum_{i=1}^{n} \sum_{f=1}^{n} d_{i,f,j}) - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,f \in C_k} d_{i,f,j} \}$$

sotto i vincoli

$$\sum_{j=1}^{p} w_j^2 \leq 1, \qquad \sum_{j=1}^{p} w_j \leq s, \qquad w_j \geq 0 \quad \forall j. \tag{13}$$

Vi sarà sparsità nei pesi per un dato valore del parametro s, il cui criterio di scelta verrà trattato nel successivo paragrafo. Come esplicitato precedentemente, il parametro di ottimizzazione deve sempre rispettare la condizione $1 \leq \mathrm{s} \leq \sqrt{p}$, con p numero di variabili totali.
Anche in questa categoria, nel momento in cui vi è equivalenza tra i pesi, cioè

w_1=...=w_p, la questione si riduce ad un caso di clustering tradizionale con il metodo delle k-medie.
Si nota, inoltre, che il problema di ottimizzazione nella formula (13) equivale al caso generale presentato nell'espressione (11), con

$$\Theta = (C_1, ..., C_K),$$
$$f_j(X_j, \Theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{f=1}^{n} d_{i,f,j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,f \in C_k} d_{i,f,j}.$$

L'utilizzo della *Lasso penality* condurrà alla definizione dei pesi nella formula (13), il che ha come conseguenza l'esclusione di un set di caratteristiche dal processo di clustering [1].

3.5.3 Selezione del parametro di ottimizzazione

Il parametro di ottimizzazione s stabilisce il numero di variabili da considerare nel processo di clustering, in quanto la penalizzazione lasso si basa sul valore da esso assunto.

L'entità del parametro di ottimizzazione dipende dalla metodologia di classificazione impiegata.
Nel caso del clustering tramite metodo delle k-medie, si selezionano dei possibili candidati per s, rispettando la condizione $1 \leq s \leq \sqrt{p}$. Per ciascuno di questi, si applica la seguente procedura [1].

- Si ottengono i dataset X_1,...,X_B permutando indipendentemente le osservazioni di ciascuna variabile dell'insieme di dati originali.
- Per ciascun candidato per il parametro s si calcola

  $$O(s) = \sum_j w_j \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{f=1}^{n} d_{i,f,j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,f \in C_k} d_{i,f,j} \right)$$

  ossia la funzione obiettivo ottenuta applicando il clustering sparso con metodo delle k-medie con il parametro s sui dati originali $\mathbf{X}$.

  Si calcola, inoltre, $O_b(s)$, cioè la funzione obiettivo ottenuta tramite l'impiego del metodo delle k-medie dato il parametro s sui dati permutati X_b, con b=1,...,B.

- Infine, si calcola la statistica *Gap*, data da

$$Gap(s) = log(O(s)) - \frac{1}{B}\sum_{b=1}^{B} log(O_b(s)).$$

Si sceglie, quindi, s corrispondente al valore massimo di Gap(s). La statistica *Gap*, infatti, valuta l'efficienza del clustering applicato ai dati originali. Perciò, il valore ottimale del parametro s è quello che massimizza la suddetta quantità.

D'altra parte, per ciò che concerne i metodi di clustering gerarchico, la procedura da seguire è la medesima. Tuttavia, la funzione obiettivo in questo caso risulta essere

$$O(s) = \sum_{j} w_j \sum_{i,f} d_{i,f,j} U_{i,f}.$$

4 Applicazione dei metodi di clustering

4.1 Riduzione del dataset

Nel seguente capitolo si procederà applicando le metodologie di clustering presentate in precedenza al dataset di riferimento, utilizzando i dati originali, di cui non si è eseguita, quindi, la standardizzazione.
In un primo momento si analizzeranno i metodi tradizionali, ponendo attenzione alle tecniche gerarchiche e partizionali. In modo del tutto simmetrico, si impiegheranno i suddetti approcci nell'ambito del clustering sparso, con lo scopo di paragonare le due metodologie.
Al fine di perseguire tale obiettivo, è risultato necessario ridurre significativamente la dimensionalità del dataset. Difatti, invece delle originali 5172 unità statistiche e 3001 variabili, ne si considera un sottoinsieme, costituito da 653 osservazioni e 188 variabili. Si sono, quindi, selezionate solo le e-mail con un numero di parole compreso nell'intervallo [50, 125] , in quanto definito range ottimale di vocaboli contenuti nei testi di posta elettronica [18].
D'altra parte, sono state eliminate le variabili con un quantitativo considerevole di valori nulli. Per eseguire quest'ultima operazione, si è selezionata come soglia di riferimento 4000, in quanto corrisponde all'approssimazione del numero medio di zeri per vocabolo nel dataset. Dunque, ogni variabile contenente una quantità di valori nulli superiore a 4000 è stata rimossa.
La risposta *Prediction*, dicotomica, rimane invariata.
In conclusione, il nuovo insieme di dati è costituito da 653 e-mail, di cui 81.32% autentiche e 18.68% spam, e 188 variabili, di cui 187 parole e, infine, la risposta *Prediction*.

0 (reali)	1 (spam)
531	122

Tabella 10: Numero di e-mail reali e spam nel dataset modificato.

4.2 Metodi tradizionali

4.2.1 Clustering gerarchico

Gli algoritmi di clustering gerarchico, nella loro totalità, si basano su una matrice di distanza. Quest'ultima può essere ottenuta applicando una delle misure di similarità presentate nel paragrafo 3.2 del capitolo precedente. Nelle seguenti analisi si utilizzerà la distanza euclidea.

Inizialmente ci si concentrerà sul clustering gerarchico agglomerativo, in tutte le sue caratterizzazioni, per poi considerare anche i risultati prodotti dagli algoritmi di carattere divisivo.
Le diverse tipologie di classificazione gerarchica generano alberi di cluster, o dendogrammi, da cui emerge la suddivisione effettuata.

Nell'ambito del clustering gerarchico agglomerativo, dall'applicazione del metodo completo, risulta il seguente dendogramma.

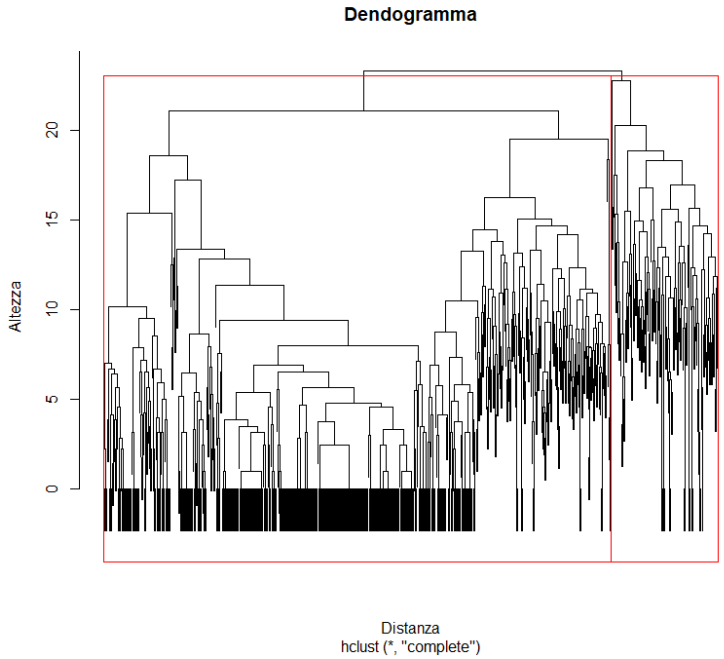

Figura 35: Dendogramma ottenuto tramite il metodo completo.

Dato l'obiettivo di classificare le osservazioni, ossia le e-mail, in due cluster, cioè posta spam e reale, nella Figura 35 si evidenziano, inoltre, i due insiemi che emergono dall'impiego del legame completo.

D'altra parte, l'uso del legame singolo produce il seguente risultato.

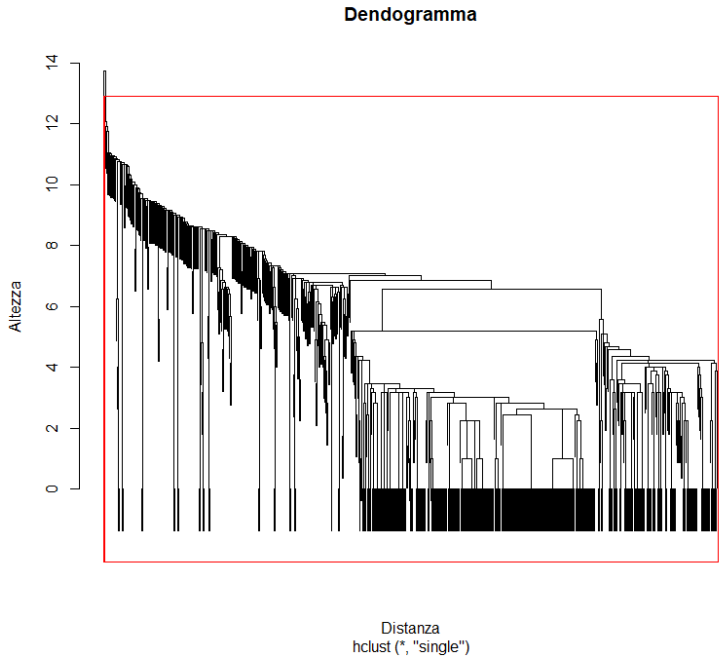

Figura 36: Dendogramma ottenuto tramite il metodo singolo.

Tale albero suggerisce la presenza di un unico gruppo. Infatti, la selezione di due insiemi al suo interno rende chiara l'esistenza di un singolo grande cluster.

Risultato analogo si ottiene, inoltre, dall'uso del legame medio. Anche in questo caso, infatti, appare esservi un solo grande insieme. Tale deduzione è confermata dalla Figura 37, in cui è esplicita la presenza di un primo gruppo, contenente un numero trascurabile di osservazioni, ed un secondo cluster, che comprende la quasi totalità delle unità.

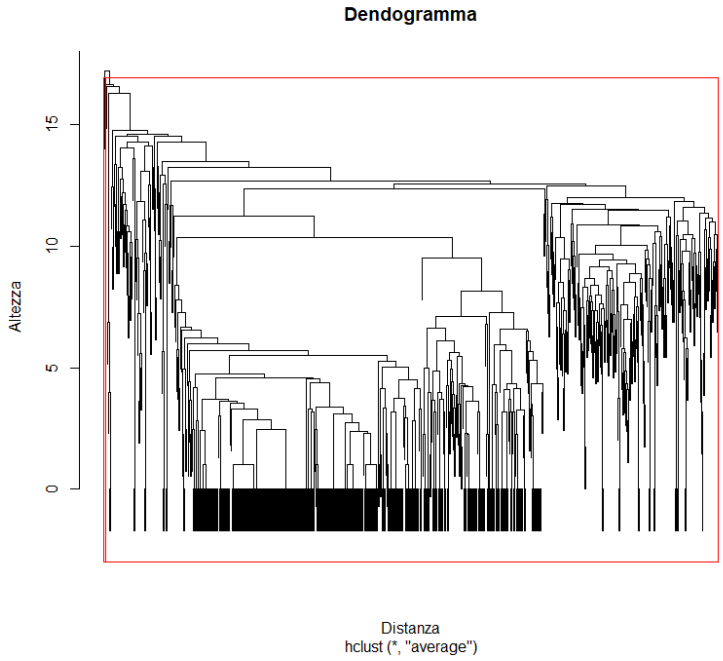

Figura 37: Dendogramma ottenuto tramite il legame medio.

Infine, nelle seguenti immagini, sono riportati i risultati ottenuti tramite il metodo di Ward. Il dendogramma da esso derivante suggerisce la possibilità di raggruppare le unità sia in due che in tre cluster. Al fine di visualizzare tale aspetto, nelle figure 38 e 39 sono stati evidenziati i diversi possibili gruppi.

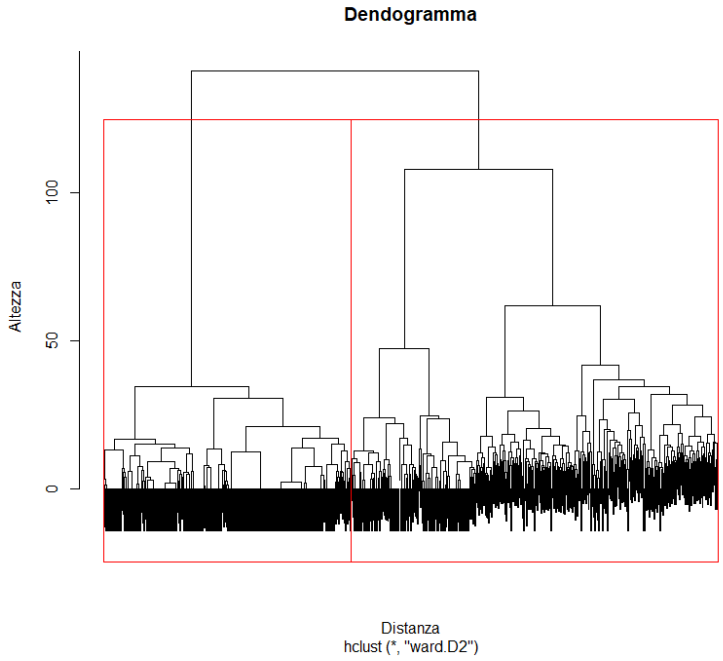

Figura 38: Dendogramma ottenuto tramite il metodo di Ward con due cluster.

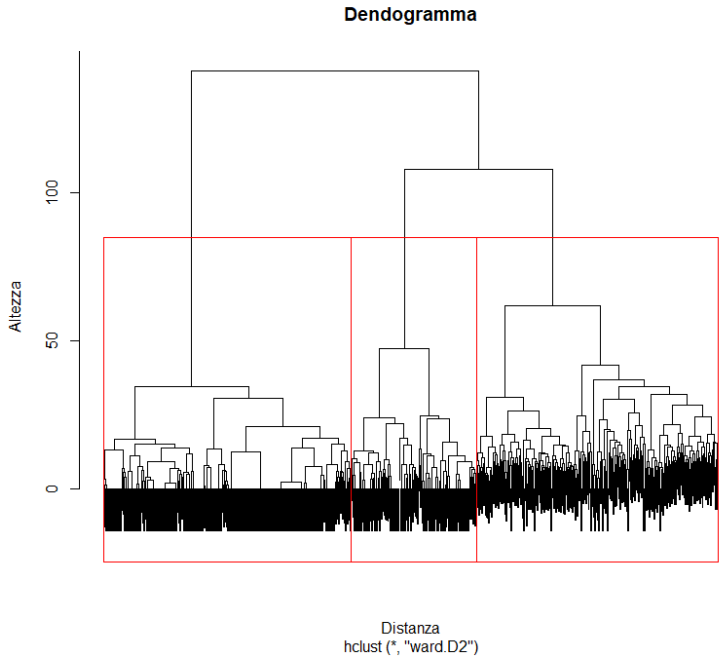

Figura 39: Dendogramma ottenuto tramite il metodo di Ward con tre cluster.

Nella tabella 11 si visualizzano il numero di osservazioni per gruppo individuate dalle differenti metodologie di clustering gerarchico agglomerativo applicate. In modo del tutto conforme ai dendogrammi ottenuti dalle analisi, si nota come i legami singolo e medio raggruppino la quasi totalità delle unità in un unico cluster. D'altra parte, la classificazione ottenuta tramite il metodo di Ward e con il legame completo permette di distinguere chiaramente due insiemi. In particolare, la suddivisione risultante dal legame completo si avvicina estremamente, a livello di numerosità, alla distinzione tra e-mail spam e ham del dataset, visibile nella tabella 10.
L'assunzione dell'esistenza di tre cluster, nel caso del legame di Ward, produce la suddivisione riportata nella tabella 12.

metodo	cluster 1	cluster 2
completo	539	114
singolo	652	1
medio	651	2
Ward	390	263

Tabella 11: Numero di osservazioni per cluster.

metodo	cluster 1	cluster 2	cluster 3
Ward	257	263	133

Tabella 12: Numero di osservazioni per cluster.

Tuttavia, al fine di valutare le differenti metodologie impiegate, è necessario paragonare le classificazioni ottenute con la variabile risposta. Ciò viene effettuato tramite le tabelle 13, 14 e 15, relative, rispettivamente, al legame completo e al metodo di Ward nel caso di due e di tre cluster. Da queste risulta chiaro che la categorizzazione, in tutte le casistiche, non avviene in modo coerente alla variabile *Prediction*. Infatti, nel caso del legame completo, si nota come la distinzione dei messaggi nei due gruppi non rifletta la loro effettiva natura, ossia spam e reali, in quanto 446 e-mail autentiche vengono riposte nel cluster 1, e 85 nel secondo insieme. In merito alle e-mail fraudolente, 93 di esse vengono poste nel gruppo 1 e 29 nel cluster 2.
Per ciò che concerne il metodo di Ward con due cluster, le e-mail autentiche vengono suddivise tra il primo ed il secondo gruppo, con, rispettivamente, 268 e 263 osservazioni. La totalità dei testi illeciti, invece, appartiene ad un unico cluster, il primo.
Nel caso di tre gruppi, invece, gran numero dei messaggi reali si concentra nel

cluster 2, mentre le e-mail spam vengono quasi tutte riunite nel gruppo 1. Quindi l'assunzione di tre cluster permette di avere una suddivisione tra testi spam e ham, anche se non del tutto netta, date le 271 osservazioni classificate erroneamente.
Il legame singolo e medio, d'altra parte, producono lo stesso risultato, poiché, come espresso precedentemente, raggruppano la quasi totalità delle osservazioni in un unico cluster.

Prediction	cluster 1	cluster 2
0	446	85
1	93	29

Tabella 13: Osservazioni distinte per cluster e per variabile risposta nel legame completo.

Prediction	cluster 1	cluster 2
0	268	263
1	122	0

Tabella 14: Osservazioni distinte per cluster e per variabile risposta nel metodo di Ward con due gruppi.

Prediction	cluster 1	cluster 2	cluster 3
0	138	263	130
1	119	0	3

Tabella 15: Osservazioni distinte per cluster e per variabile risposta nel metodo di Ward con tre gruppi.

È possibile, inoltre, proseguire con la valutazione delle suddivisioni sulla base dei valori assunti dalla *silhouette* nelle varie casistiche.
Nel caso della classificazione tramite legame completo, si hanno 38 osservazioni con *silhouette* negativa. D'altra parte, il suddetto valore si riduce per quanto riguarda il legame singolo e medio, a, rispettivamente, 3 e 2 unità. Infine, il metodo di Ward produce il numero maggiore di osservazioni con *silhouette* negativa, ossia 187, nel caso della distinzione in due gruppi, e 94 per tre.
Le unità che presentano una *silhouette* minore di zero non sono raggruppate nel cluster per loro ottimale, in quanto risultano più vicine a insiemi differenti dal proprio. Tale classificazione erronea si presenta in tutti i casi, in modo più o meno

marcato, indipendentemente dalla metodologia utilizzata per il clustering gerarchico agglomerativo.

Trattando, invece, degli algoritmi divisivi, il dendogramma prodotto è il seguente.

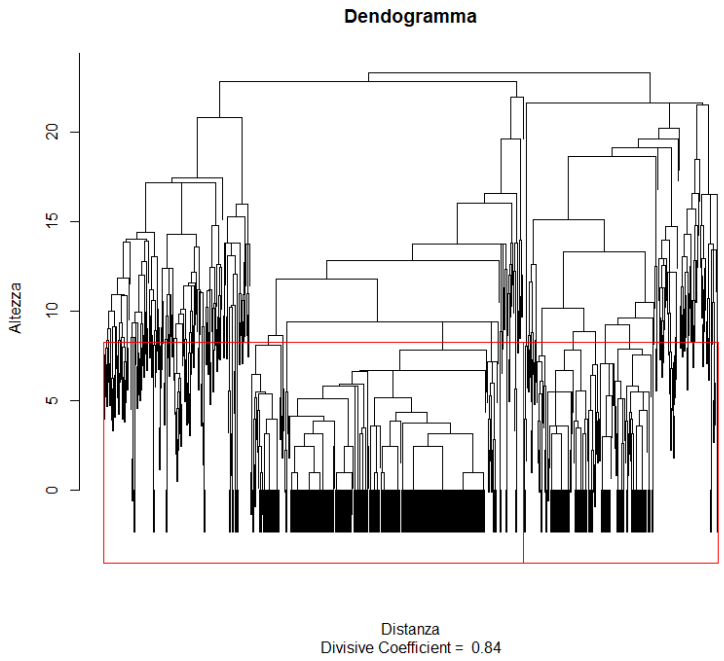

Figura 40: Dendogramma derivante dal clustering gerarchico divisivo.

La suddivisione in 2 cluster avviene assegnando 446 osservazioni al primo gruppo e 207 al secondo, come visibile nella Figura 40.

Anche in questa casistica, la classificazione rispetto alla variabile risposta non è del tutto netta, come si evince dalla tabella 16. Infatti 357 e-mail ham appartengono al cluster 1, mentre 174 al secondo. D'altra parte, 89 osservazioni spam vengono poste nell'insieme 1, 33 nel secondo.

Prediction	cluster 1	cluster 2
0	357	174
1	89	33

Tabella 16: Osservazioni distinte per cluster e per variabile risposta nel metodo divisivo.

Infine, in termini di silhouette, 6 sono le osservazioni con valore negativo.

Concludendo, tutte le metodologie viste conducono a raggruppamenti non del tutto soddisfacenti rispetto ai risultati suggeriti dalla variabile risposta. L'unico caso in cui si è riscontrata una suddivisione delle unità in linea con quanto determinato dalla variabile *Prediction* è stato con il clustering in tre gruppi tramite il metodo di Ward. Quest'ultimo, tuttavia, presenta una *silhouette* particolarmente elevata, che suggerisce un raggruppamento non del tutto corretto. Inoltre, l'ipotesi dell'esistenza di tre gruppi non è coerente con la tipologia di studio che si intende effettuare, poiché il fine perseguito è quello di categorizzare le osservazioni in due cluster sulla base della loro natura, spam o reale.
Gli esiti ottenuti potrebbero essere connessi all'elevato numero di zeri che l'insieme di dati contiene, oppure a variabili non del tutto significative che vengono coinvolte nel processo di classificazione. Quest'ultima possibilità verrà valutata nell'applicazione dei metodi sparsi.

4.2.2 Clustering partizionale

Nel seguente paragrafo, la classificazione delle osservazioni viene effettuata tramite l'utilizzo del clustering partizionale. In particolare, si impiegherà il metodo delle k-medie.
Come definito nel capitolo 3, gli algoritmi di clustering partizionale necessitano di molteplici informazioni per il loro funzionamento, quali il numero di cluster. Tale dato, in questo caso, è deducibile dalla finalità dello studio. Infatti, visto l'obiettivo di suddividere le e-mail in spam e ham, verranno considerati 2 gruppi nelle analisi.
Tuttavia, per determinare il numero ottimale di gruppi, è possibile valutare la percentuale di varianza spiegata da cluster di diverse numerosità (9). Applicando questo approccio a raggruppamenti di diversa composizione, in cui il numero di gruppi varia da 1 a 6, si ottiene il grafico in Figura 41.

L'insieme con il numero ideale di unità corrisponde al "gomito" della curva, ossia il punto in cui il grafico si inclina in modo significativo, in quanto coincide con il gruppo che genera un incremento rilevante della varianza spiegata. Effettuando questa tipologia di valutazione, quindi, appaiono necessari tre cluster.
Conseguentemente, nelle analisi seguenti, si valuteranno i risultati ottenuti considerando sia due insiemi, dato l'obiettivo dello studio, che tre gruppi, come evidenziato dalla percentuale di varianza spiegata.

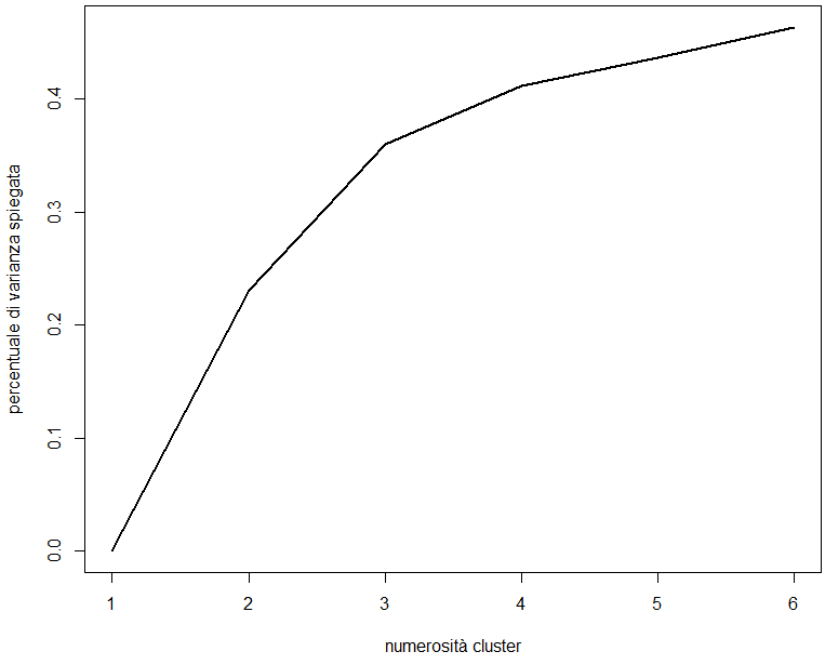

Figura 41: Varianza spiegata per diverse numerosità di cluster.

L'assunzione di due cluster conduce ad una suddivisione quasi equa delle unità tra i due gruppi, aspetto visibile nella tabella 17. Nel caso dei tre insiemi, invece, si nota come le osservazioni si concentrano nei primi due gruppi, com'è possibile dedurre dalla tabella 18.
In aggiunta, paragonando le classificazioni ottenute con la variabile risposta *Prediction*, è possibile constatare se vi è coerenza tra la suddivisione effettuata tramite il metodo delle k-medie e i dati effettivi.
Nel caso dei due cluster, è evidente come la maggior parte delle e-mail autentiche vengano assegnate al gruppo 2, mentre la quasi totalità dei testi spam appartiene al primo cluster, segnando, quindi, 213 osservazioni collocate in modo errato. Comportamento analogo si ha anche considerando tre cluster, in quanto un elevato numero di messaggi autentici risiede nel primo insieme, mentre i testi spam si collocano quasi interamente nel cluster 2.
Queste considerazioni sono rafforzate dalle figure 42 e 43, nelle quali ciascuna tipologia di e-mail, identificata dalla forma del punto, viene assegnata al relativo insieme.

Prediction	cluster 1	cluster 2
0	210	321
1	119	3
totale	329	324

Tabella 17: Osservazioni distinte per cluster e per variabile risposta considerando due gruppi.

Prediction	cluster 1	cluster 2	cluster 3
0	264	135	132
1	0	119	3
totale	264	254	135

Tabella 18: Osservazioni distinte per cluster e per variabile risposta considerando tre gruppi.

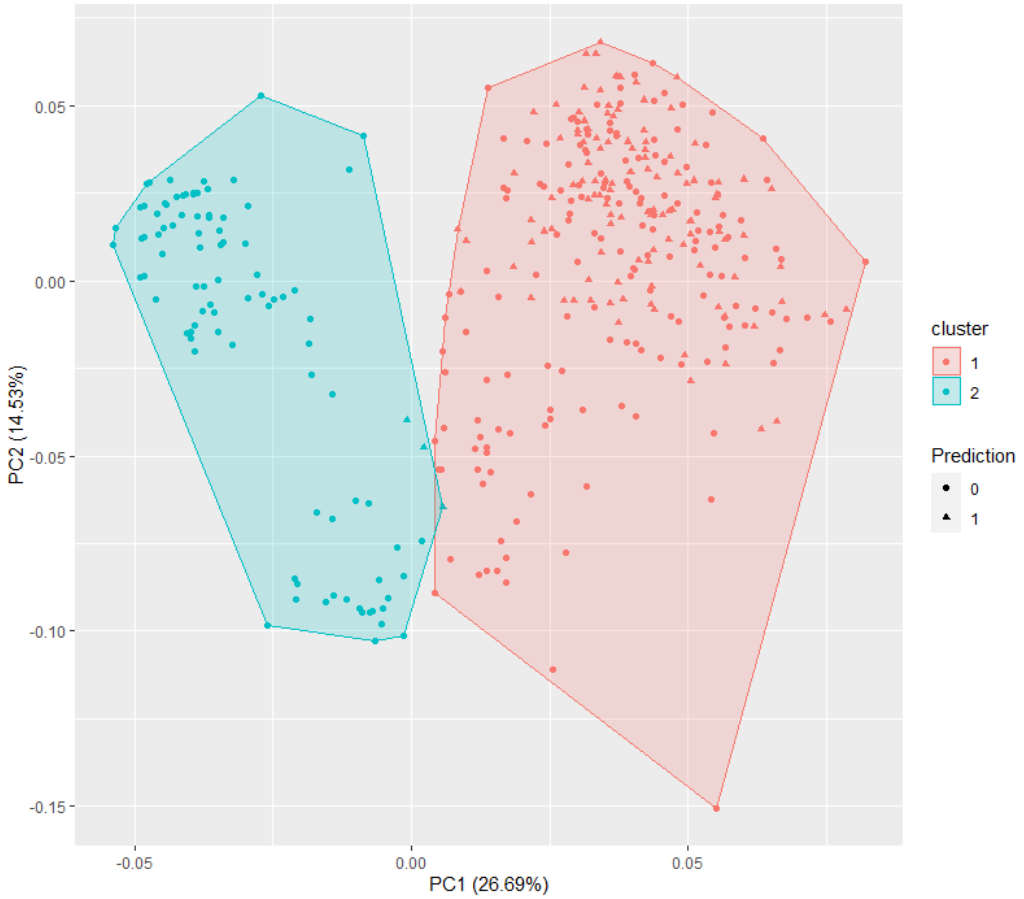

Figura 42: Grafico delle classificazioni in due cluster.

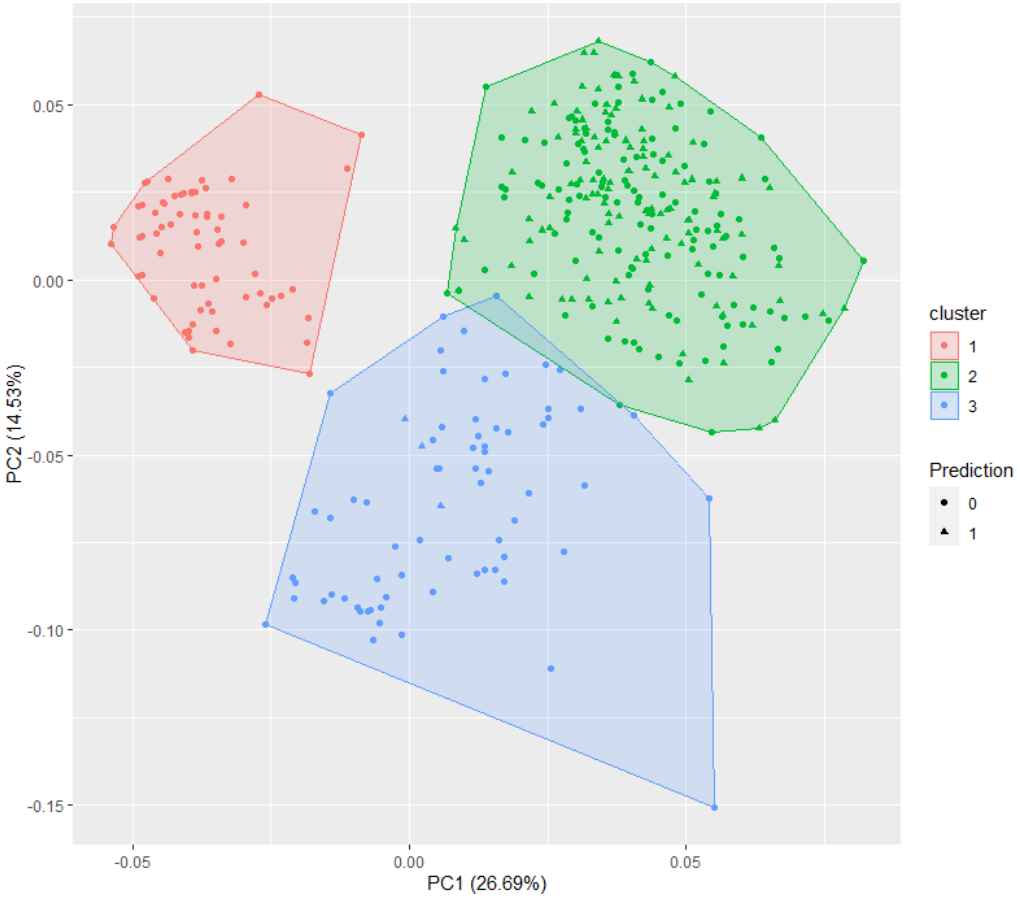

Figura 43: Grafico delle classificazioni in tre cluster.

In termini di *silhouette*, d'altra parte, non si presentano notevoli differenze tra le due categorie considerate, ossia due e tre gruppi. Infatti, 88 sono le osservazioni con *silhouette* negativa nel caso dei due cluster, 85 con tre insiemi.

Concludendo, l'assunzione di un numero maggiore di cluster, in questo caso tre, non sembra apportare considerevoli vantaggi, nonostante il livello maggiore di informazione spiegata. Inoltre, la categorizzazione in due insiemi risulta essere più in linea con la tipologia di studio considerato.
Tuttavia, si nota come l'utilizzo dei metodi partizionali abbia permesso di ottenere esiti decisamente più soddisfacenti rispetto alle tecniche gerarchiche. Difatti, il metodo k-means è riuscito ad identificare e a porre in un singolo gruppo quasi tutte le e-mail spam, in modo distinto dalle autentiche. Le difficoltà, tuttavia, si sono presentate nella classificazione dei messaggi reali, la quale è avvenuta in maniera decisamente meno rigorosa.

4.3 Metodi sparsi

4.3.1 Clustering gerarchico

Si prosegue con l'analisi dei dati presentati precedentemente, applicando una metodologia alternativa di classificazione, ossia il clustering sparso. Infatti, considerando l'elevato volume di zeri presente nel dataset, nonostante la sua riduzione, i metodi sparsi permettono di categorizzare le osservazioni basandosi esclusivamente su un sottoinsieme di variabili rilevanti.

In questo paragrafo ci si focalizzerà sul clustering gerarchico, in modo simmetrico a quanto effettuato con le tecniche tradizionali, poiché si visualizzeranno i dendogrammi prodotti dai diversi legami impiegati ed i risultati relativi ad ogni tipologia di suddivisione. Nell'ambito della classificazione sparsa, tuttavia, si utilizzano solo le tecniche gerarchiche agglomerative.

Per poter avviare l'algoritmo di clustering gerarchico sparso è necessario definire il parametro di ottimizzazione s, come anticipato nel capitolo 3. Esso viene determinato tramite l'impiego della statistica *Gap*, che, nel dataset di riferimento, apporta ai risultati visibili nella tabella 19 e nel grafico in Figura 44.

parametro di ottimizzazione s	pesi non nulli	statistica *Gap*	deviazione standard
1.1000	2	0.0032	1e-04
2.0414	12	0.0306	4e-04
2.9827	23	0.0515	4e-04
3.9241	44	0.0611	4e-04
4.8655	130	0.0645	4e-04
5.8069	186	0.0649	4e-04
6.7482	186	0.0649	4e-04
7.6896	186	0.0649	4e-04
8.6310	186	0.0649	4e-04
9.5724	186	0.0649	4e-04

Tabella 19: Selezione del parametro di ottimizzazione nel clustering gerarchico sparso.

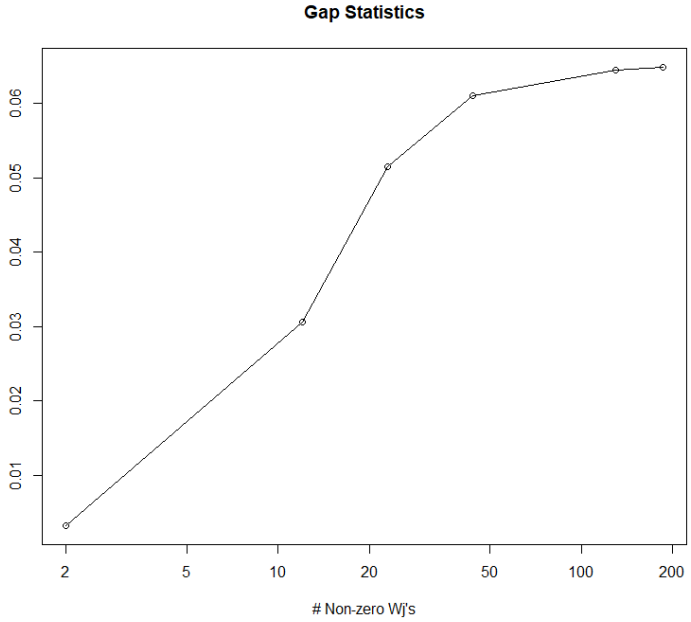

Figura 44: Statistica *Gap* nel clustering gerarchico sparso rapportata al numero di pesi w_j non nulli.

Il parametro ottimo massimizza la statistica *Gap*. In questo caso, il valore s che conduce ad ottenere il massimo della suddetta statistica, ossia 0.0649, è 5.8069. Ne deriva che 186 parole sono significative, in quanto presentano pesi non nulli, la cui distribuzione è evidente nella Figura 45. Nel loro complesso, i pesi soddisfano

i vincoli specificati nell'espressione (12), in quanto la somma dei loro quadrati è unitaria e la loro semplice addizione è pari a 5.2048, valore che risulta strettamente minore di s.
In termini di pesi w_j associati alle variabili coinvolte nel processo di classificazione sparsa, essi variano nell'intervallo [0.00, 0.53]. Si attribuiscono pesi inferiori a 0.01 a molteplici parole, quali *this*, *that*, *will*, *by*, *any*, *has*, *know*, *need*, *let*, *one*, *pm*, *now*, *thanks*, *his*, *end*, *subject*, *hanks*, *man*, *mm*, *cc*, *ok*, *sit*, *war*, *mai*, *ten*, *wil* e *ward*.
Ad assumere, invece, importanza nel clustering sono i vocaboli *e* e *l*, poiché presentano valori di w_j maggiori di 0.5.
Si nota, fondamentalmente, che nella suddivisione delle osservazioni in gruppi entrano con pesi notevoli i termini che si presentano nelle e-mail con frequenza maggiore. Infatti, vocaboli che figurano occasionalmente, hanno coefficienti w_j pressoché nulli.
Una volta definito il parametro s e stabilita la distribuzione dei pesi, è possibile proseguire osservando i risultati derivanti dalla classificazione sparsa dei dati, nei casi di legame completo, singolo e medio. Difatti, nella classificazione sparsa, non è disponibile il metodo di Ward.

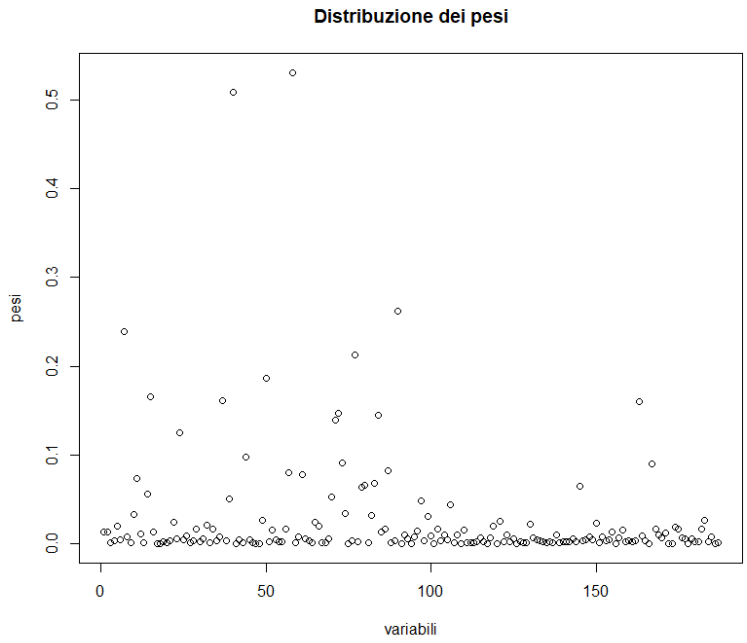

Figura 45: Distribuzione dei pesi nel clustering gerarchico sparso.

Dall'esecuzione del clustering gerarchico con legame completo deriva il dendogramma riportato in Figura 46. In esso è evidente la distinzione delle unità in due gruppi, con uno nettamente più numeroso dell'altro.

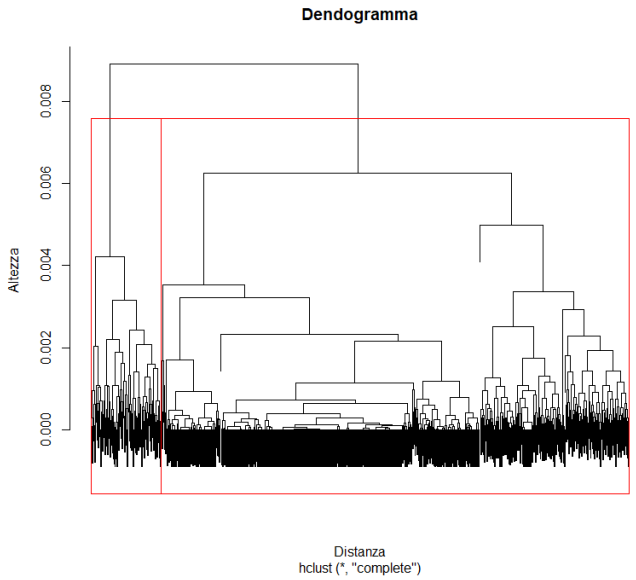

Figura 46: Dendogramma derivante dal clustering gerarchico sparso con legame completo.

Considerando, invece, il legame singolo, esso produce l'albero di cluster riportato in Figura 47, che suggerisce la presenza di un singolo gruppo. Tale ipotesi è rafforzata dalla selezione di due insiemi, che porta ad ottenere un unico grande cluster contenente la quasi totalità delle osservazioni.

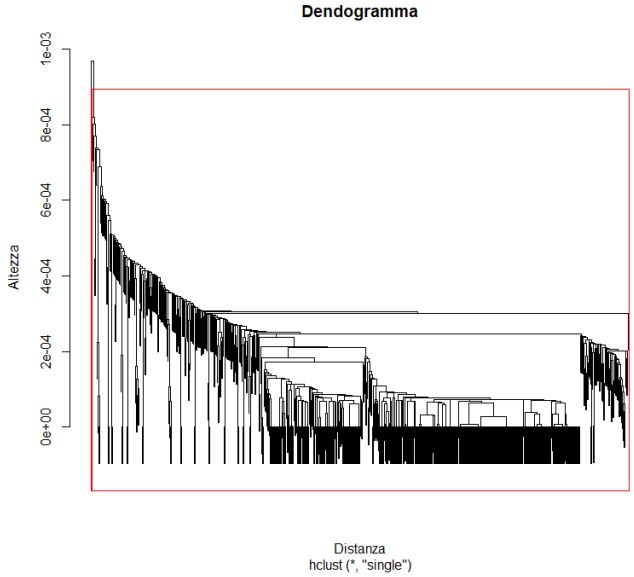

Figura 47: Dendogramma derivante dal clustering gerarchico sparso con legame singolo.

Utilizzando il legame medio, in modo analogo al metodo singolo, si riscontra l'evidenza di un unico gruppo, visibile in Figura 48.

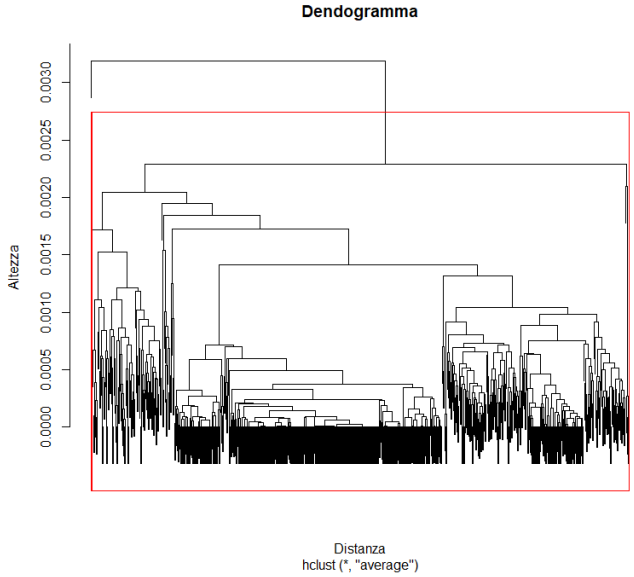

Figura 48: Dendogramma derivante dal clustering gerarchico sparso con legame medio.

Nella tabella 20 sono riportate le numerosità dei cluster per ogni metodologia utilizzata. È evidente come i risultati siano i medesimi nei casi di legame singolo e medio. Infatti, quest'ultimi pongono la quasi totalità delle osservazioni in un unico gruppo. Dunque, il legame completo appare essere l'unico in grado si identificare due cluster, la cui numerosità è simile a quella dei dati, visibile nella tabella 10.

metodo	cluster 1	cluster 2
completo	568	85
singolo	652	1
medio	652	1

Tabella 20: Numerosità dei cluster.

Tuttavia, dal paragone dei risultati ottenuti con i dati effettivi, si nota che la classificazione non riproduce correttamente la suddivisione definita dalla variabile risposta. Infatti, si nota che, in tutti i casi, le unità vengono assegnate ai due cluster in modo non del tutto coerente con la risposta *Prediction*. Non vi è, quindi, una divisione netta tra e-mail spam e ham nei due gruppi. Questa conclusione discende dalla tabella 21, per quanto riguarda il legame completo, nella quale è esplicito come la maggior parte dei messaggi, sia spam che autentici, vengano collocati nel cluster 1.
I legami singolo e medio, com'è stato esplicitato precedentemente, riuniscono quasi tutte le osservazioni in un unico cluster, non riproducendo, quindi, la ripartizione stabilita da *Prediction*.
Ne deriva che, anche con i metodi sparsi, il clustering gerarchico non conduce a risultati in linea con i dati. Ciò potrebbe essere dovuto alla cattiva performance della classificazione gerarchica su dati di elevate dimensioni.

Prediction	cluster 1	cluster 2
0	471	60
1	97	25

Tabella 21: Numero di osservazioni per gruppo considerando il legame completo e la variabile risposta.

4.3.2 Clustering partizionale

In questo paragrafo si applicherà il metodo partizionale delle k-medie sparso. Al fine di procedere in modo analogo a quanto fatto nel caso delle tecniche tradizionali, si effettueranno le analisi considerando sia due che tre cluster.

L'utilizzo dei metodi sparsi detta la necessità di stabilire il valore del parametro di ottimizzazione s, come effettuato precedentemente per la classificazione gerarchica. Esso varia dipendentemente dal numero di cluster considerati. Nel caso dei due gruppi, tale indicatore, ottenuto dalla statistica *Gap*, è 2.6072, in quanto corrisponde al valore massimo raggiunto dalla suddetta misura, ossia 0.1112. Tutti questi risultati sono deducibili dalla tabella 22 e dal grafico riportato in Figura 49. Il numero di variabili rilevanti è, quindi, 186. I pesi associati ad esse, visibili in Figura 50, soddisfano tutte le condizioni espresse nella (13), ossia la loro positività, la somma dei quadrati unitaria, ed il totale, pari 2.6888, minore del parametro di ottimizzazione.
Nella suddetta figura, inoltre, è evidente come quasi tutte le parole presentano pesi w_j minori di 0.2. Risulta, anche in questo caso, che i coefficienti di ponderazione vengono determinati sulla base della frequenza con cui figura il vocabolo all'interno dei testi considerati. Infatti, l'unica variabile con w_j maggiore di 0.5 è *l*, che è uno dei vocaboli che appaiono più spesso all'interno delle e-mail analizzate.

parametro di ottimizzazione s	pesi non nulli	statistica *Gap*	deviazione standard
1.2000	3	0.0432	0
1.5542	8	0.0834	0
2.0130	15	0.1041	0
2.6072	56	0.1112	0
3.3768	186	0.1112	0
4.3736	186	0.1112	0
5.6646	186	0.1112	0
7.3367	186	0.1112	0
9.5024	186	0.1112	0
12.3073	186	0.1112	0

Tabella 22: Selezione del parametro di ottimizzazione nel clustering partizionale sparso con due gruppi.

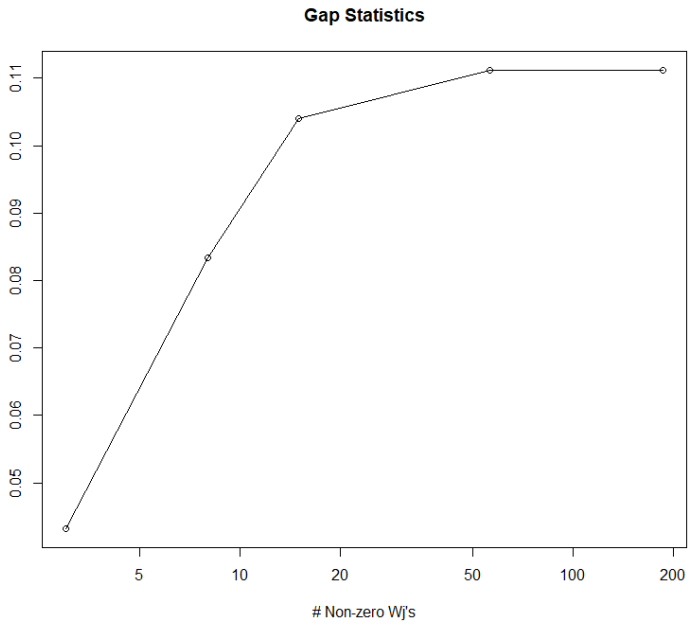

Figura 49: Statistica *Gap* nel clustering partizionale sparso con due gruppi rapportata al numero di pesi w_j non nulli.

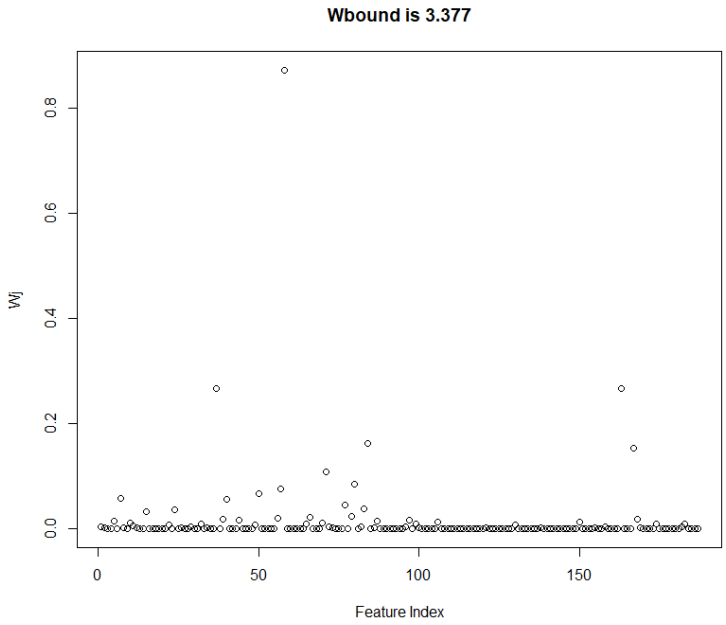

Figura 50: Distribuzione dei pesi w_j con il metodo k-means sparso considerando due cluster.

In questo momento, poiché è stato determinato il valore di s nel caso di due cluster, è possibile procedere con l'applicazione del metodo delle k-medie sparso ai dati. I due gruppi risultanti da tale suddivisione presentano le composizioni riportate nella tabella 23.
Dato l'obiettivo di suddividere le osservazioni, cioè le e-mail, in due insiemi sulla base della loro natura, spam o ham, si prosegue paragonando la classificazione eseguita con i dati effettivi. Si nota come gran parte dei messaggi autentici vengono riposti nel secondo cluster, mentre la quasi totalità delle e-mail spam vengono collocate nel primo insieme. Si totalizzano, quindi, 206 osservazioni classificate in modo errato, cioè 11 e-mail spam collocate nel cluster 2 e 195 messaggi reali nel gruppo 1. Dunque, anche con i metodi partizionali sparsi, i messaggi fraudolenti vengono quasi del tutto identificati. Si presentano difficoltà, tuttavia, nel riconoscimento di quelli reali, che si concentrano nel secondo gruppo, ma sono comunque presenti in numero notevole nel primo.

Prediction	cluster 1	cluster 2
0	195	336
1	111	11
totale	306	347

Tabella 23: Numerosità dei due cluster rapportata alla variabile risposta.

Si valutano, ora, i risultati ottenuti assumendo la presenza di tre cluster. Il parametro di ottimizzazione è 3.3768, corrispondente al valore massimo della statistica *Gap*, ossia 0.1424, com'è possibile dedurre dalla tabella 24 e dal grafico in Figura 51. I pesi associati a ciascuna parola, riportati Figura 52, presentano lo stesso andamento riscontrato nella classificazione in due cluster. Anch'essi rispettano i vincoli definiti nella (13). In particolare, la loro somma, pari a 3.3769 è quasi equivalente al parametro s.

parametro di ottimizzazione	pesi non nulli	statistica *Gap*	Deviazione standard
1.2000	3	0.0910	0.0071
1.5542	7	0.1139	0.0108
2.0130	15	0.1359	0.0108
2.6072	61	0.1423	0.0108
3.3768	186	0.1424	0,0108
4.3736	186	0.1424	0.0108
5.6646	186	0.1424	0.0108
7.3367	186	0.1424	0.0108
9.5024	186	0.1424	0.0108
12.3073	186	0.1424	0.0108

Tabella 24: Selezione del parametro di ottimizzazione nel clustering partizionale sparso con tre gruppi.

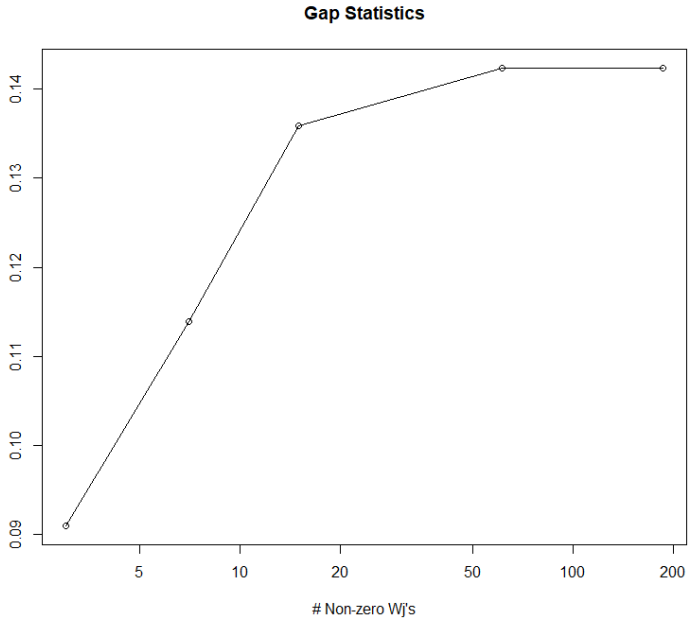

Figura 51: Statistica *Gap* nel clustering partizionale sparso con tre gruppi rapportata al numero di pesi w_j non nulli.

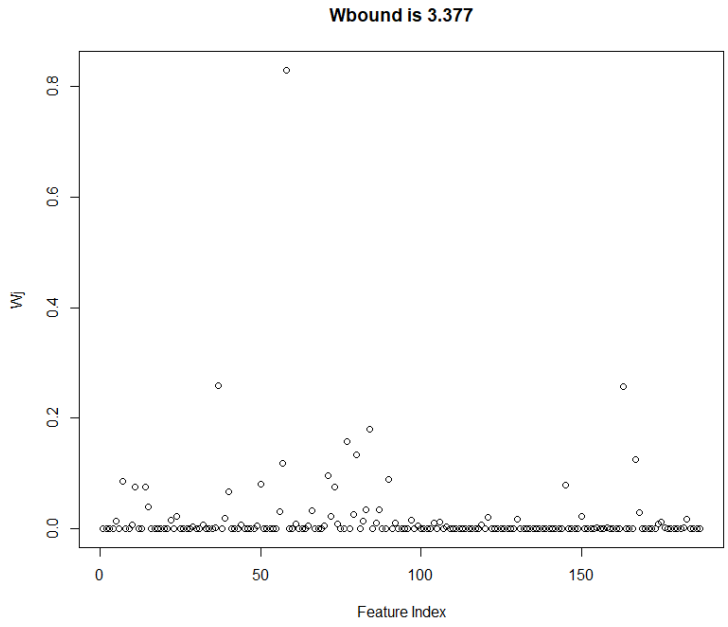

Figura 52: Distribuzione dei pesi w_j nel clustering partizionale sparso considerando tre cluster.

Le numerosità dei tre cluster che risultano da tale categorizzazione sono visibili nella tabella 25. Da quest'ultima emerge che la maggior parte delle osservazioni si concentra nei primi due gruppi. Inoltre, dal confronto con la variabile risposta, deriva che i messaggi autentici prevalgono nel secondo cluster, mentre la quasi totalità delle e-mail spam appartiene al primo insieme. Si ottengono 286 unità mal classificate.

Prediction	cluster 1	cluster 2	cluster 3
0	152	261	118
1	106	1	15
totale	258	262	133

Tabella 25: Numerosità dei tre cluster rapportata alla variabile risposta.

In conclusione, l'introduzione di un terzo cluster non apporta ad alcun vantaggio. Quindi, dato il fine dello studio, sono sufficienti due insiemi.
Come per il metodo tradizionale delle k-medie, anche le metodologie sparse permettono di identificare due gruppi. Nel primo vengono collocate quasi tutte le e-mail spam. Nel secondo, invece, si concentrano i messaggi autentici.
Tramite l'utilizzo dell'algoritmo tradizionale delle k-medie si sono totalizzate 213 osservazioni classificate erroneamente. Tale numero si riduce a 206 con i metodi sparsi. Dunque la suddivisione che permette di avere il minore margine d'errore si basa sul metodo *k-means* sparso.

Conclusione

L'elaborato si è posto come obiettivo quello di individuare una potenziale metodologia alternativa basata sull'analisi dei dati multidimensionali per l'identificazione dei messaggi di posta elettronica di carattere fraudolento.
Per poter constatare la concretezza del suddetto sistema, si è introdotto un dataset di elevate dimensioni, presentato nel capitolo 2, sul quale effettuare le analisi.
La sua notevole dimensionalità, che ha comportato problematiche a livello di costo computazionale, e l'elevata inflazione di zeri hanno dettato la necessità di selezionare un sottoinsieme dei dati su cui applicare gli algoritmi di clustering.
In un primo momento si sono utilizzate le tecniche tradizionali di classificazione, in particolare il clustering gerarchico, di cui si sono studiate tutte le diverse caratterizzazioni, e le tecniche di clustering partizionale, con particolare attenzione al metodo delle k-medie. Quest'ultimo è stato utilizzato sui dati tenendo in considerazione sia due che tre possibili cluster.
Si sono, inoltre, impiegate nelle analisi anche le metodologie di clustering sparso, in quanto esse danno la possibilità di classificare le osservazioni sulla base di un sottoinsieme di variabili rilevanti, aspetto vantaggioso nel momento in cui non si è certi che siano tutte significative nel processo di categorizzazione.
In entrambe le casistiche, quindi sia con i metodi tradizionali che sparsi, si nota che il clustering gerarchico nella sua totalità non permette di distinguere le osservazioni in due gruppi distinti. Dunque, con l'utilizzo delle tecniche gerarchiche sui dati, non risulta possibile dividere le e-mail spam dalle ham, indipendentemente dal metodo utilizzato. Tale esito potrebbe essere connesso alla difficoltà che riscontrano tali algoritmi nel classificare dati di elevate dimensioni.
Con le tecniche partizionali, invece, i risultati sono decisamente più soddisfacenti. Infatti, sia con le metodologie tradizionali che con le tecniche sparse, quasi tutte le e-mail spam vengono poste all'interno di un unico insieme, separatamente da quelle autentiche, la cui maggior parte viene collocata in un secondo gruppo. Il margine d'errore risulta più contenuto nel caso della classificazione partizionale sparsa.
Concludendo, è possibile affermare che l'identificazione delle e-mail spam tramite i vocaboli in esse contenuti con l'utilizzo dei metodi di clustering può avvenire solamente per mezzo delle metodologie partizionali, in quanto le tecniche gerarchiche non permettono una netta distinzione tra testi reali e fraudolenti.

Riferimenti bibliografici

[1] Daniela Witten e Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

[2] Johnson Richard Arnold e Wichern Dean. *Applied Multivariate Statistical Analysis*. Pearson Education Limited, 2014.

[3] Jonathan Lee. *Notes on Naive Bayes Classifiers for Spam Filtering*. School of Computer Science and Engineering University of Washington.

[4] Mahamed Omran; Andries Engelbrecht; Ayed Salman. An overview of clustering methods. *Intelligent Data Analysis*, 2007.

[5] Manish Sharma. Clustering evaluation strategies. *Towards Data Science*, 2019.

[6] Sushma Wakchaure; Shailaja Pawar; Ganesh Ghuge; Bipin Shinde. Overview of anti-spam filtering techniques. *International Research Journal of Engineering and Technology (IRJET)*, 2017.

[7] Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2015.

Sitografia

[8] **Dataset oggetto di studio**
https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv?resource=download
Autore: Balaka Biswas
Ultimo accesso: 18/03/24

[9] **Cos'è lo spam email?**
https://www.proofpoint.com/it/threat-reference/spam
Ultimo accesso: 20/04/24

[10] **Caratteristiche delle e-mail spam**
https://www.kaspersky.it/resource-center/preemptive-safety/protect-yourself-from-spam-mail-using-these-simple-tips
Ultimo accesso: 20/04/24

[11] **Statista, volume delle e-mail spam a livello globale dal 2011 al 2023**
https://www.statista.com/statistics/420400/spam-email-traffic-share-annual/
Autore: Ani Petrosyan
Ultimo accesso: 23/04/24

[12] **Tecniche di filtraggio**
https://www.process.com/products/pmas/whitepapers/explanation_filter_techniques.pdf
Ultimo accesso: 3/05/24

[13] **DNSBL, definizione**
https://www.dnsbl.info/
Ultimo accesso: 3/05/24

[14] **Caratteristiche della DNSBL**
https://www.ionos.it/digitalguide/server/know-how/dnsbl/
Ultimo accesso: 3/05/24

[15] **Graylisting**
https://www.ionos.it/digitalguide/e-mail/sicurezza-e-mail/cose-il-greylisting/
Ultimo accesso: 7/05/24

[16] **DKIM**
https://www.mimecast.com/content/dkim/
Ultimo accesso: 7/05/24

[17] **Parole spam**
https://mailtrap.io/blog/email-spam-words/
Ultimo accesso: 16/05/24

[18] **EmailAnalytics, lunghezza ideale delle e-mail**
https://emailanalytics.com/ideal-email-length/
Autore: Jayson DeMers
Ultimo accesso: 19/05/24

[19] **Metodo di Ward**
https://online.stat.psu.edu/stat505/lesson/14/14.7
Ultimo accesso: 25/07/24

[20] **Libreria sparcl**
https://cran.r-project.org/web/packages/sparcl/sparcl.pdf
Autore: Daniela Witten e Robert Tibshirani
Ultimo accesso: 30/07/24

Appendice: codice R

```r
## Analisi esplorativa

#Librerie
library(tidyverse)
library(corrplot)
library(skimr)


#Caricamento dati
data<-read.csv("emails.csv", header=T, row.names=1,
    sep=",")


#Aspetti descrittivi
dim(data)
str(data)
colnames(data)


#Analisi della variabile risposta
data$Prediction<-as.factor(data$Prediction)
contrasts(data$Prediction)
tab.1<-table(data$Prediction)
tab.2<-prop.table(table(data$Prediction))
barplot(tab.2, col=c(3,2), names=c("reali", "spam"),
    ylab="Frequenza relativa")


#Analisi variabili esplicative nella totalità
data.esp<-data[, 1:3000]
summary(data.esp)

length(which(data.esp==0))/length(which(data.esp>=0))
    *100

which(colSums(data.esp)==0)
which.max(colSums(data.esp))
which.min(colSums(data.esp))
sort(colSums(data.esp), decreasing=T)
mean(colSums(data.esp))
length(which(colSums(data.esp)>mean(colSums(data.esp)
    )) )


#Analisi e-mail
mean(rowSums(data.esp))
```

```
prop.table(table(data[which(rowSums(data.esp)>
mean(rowSums(data.esp))),
3001]))
which.max(rowSums(data.esp))
data[626, 3001]
rowSums(data.esp[626,])
which.min(rowSums(data.esp))
data[303, 3001]
rowSums(data.esp[303,])


#Dataset di sole e-mail spam
data.spam<-data.esp[data$Prediction==1,]
dim(data.spam)
sort(colSums(data.spam) )
which(colSums(data.spam)==0)
which(colSums(data.spam)!=0)
which.max(colSums(data.spam))
which.min(colSums(data.spam))


#Dataset di sole e-mail ham
data.ham<-data.esp[data$Prediction==0, ]
dim(data.ham)
colSums(data.ham)
which(colSums(data.ham)==0)
which.max(colSums(data.ham))
which.min(colSums(data.ham))


#Parole spam
spam.words<-which(colSums(data.ham)==0)
length(spam.words)
spam.data<-data.esp[, spam.words]
dim(spam.data)
sort(colSums(spam.data))
summary(spam.data)
D<-cor(spam.data)
colnames(D)<-seq(1,89,1)
rownames(D)<-seq(1,89,1)
corrplot(D, method="color", tl.cex=0.4)


#Parole ham
ham.words<-which(colSums(data.spam)==0)
length(ham.words)
ham.data<-data.esp[, ham.words]
dim(ham.data)
sort(colSums(ham.data))
```

```
summary(ham.data)
D1<-cor(ham.data)
colnames(D1)<-seq(1,254,1)
rownames(D1)<-seq(1,254,1)
corrplot(D1, method="color", tl.cex=0.4)


#Analisi parole con correlazione più significativa

#spam
cor(data$img, data$htmlimg)
ggplot(data=spam.data)+
geom_point(mapping=aes(x=img,y=htmlimg))

cor(data$cellpadding, data$cellspacing)
ggplot(data=spam.data)+
geom_point(mapping=aes(x=cellpadding,y=cellspacing))

cor(data$female, data$male)
ggplot(data=spam.data)+
geom_point(mapping=aes(x=female,y=male))

#ham
cor(data$aimee, data$lannou)
cor(data$eileen, data$ponton)
cor(data$christy, data$sweeney)


#Analisi di singole parole

#e
summary(data.esp$e)
(sum(data.esp$e)/sum(data.esp))*100
sum(data.spam$e)/sum(data$e)*100
sum(data.ham$e)/sum(data$e)*100
ggplot(data, aes(x=Prediction, y=e, color=Prediction,
    legend=F)) +
geom_boxplot()+
labs(title = "",
x = "" , y="e") +
theme_minimal()

#parole meno frequenti:
#felipe, explosion, encoding, returns, padre, flw,
   formosa, dorcheus, pooling, #offsystem, decisions,
    greatest, remains, allowing

tab1<-prop.table(table(data$felipe, data$Prediction))
tab1<-tab1[-1,]
```

```
tab2<-prop.table(table(data$explosion, data$Prediction))
tab2<-colSums(tab2[-1,])
tab3<-prop.table(table(data$encoding, data$Prediction))
tab3<-colSums(tab3[-1,])
tab4<-prop.table(table(data$returns, data$Prediction))
tab4<-colSums(tab4[-1,])
tab5<-prop.table(table(data$padre, data$Prediction))
tab5<-colSums(tab5[-1,])
tab6<-prop.table(table(data$flw, data$Prediction))
tab6<-colSums(tab6[-1,])
tab7<-prop.table(table(data$formosa, data$Prediction))
tab7<-colSums(tab7[-1,])
tab8<-prop.table(table(data$dorcheus, data$Prediction))
tab8<-colSums(tab8[-1,])
tab9<-prop.table(table(data$pooling, data$Prediction))
tab9<-colSums(tab9[-1,])
tab10<-prop.table(table(data$offsystem, data$Prediction))
tab10<-colSums(tab10[-1,])
tab11<-prop.table(table(data$decisions, data$Prediction))
tab11<-colSums(tab11[-1,])
tab12<-prop.table(table(data$greatest, data$Prediction))
tab12<-colSums(tab12[-1,])
tab13<-prop.table(table(data$remains, data$Prediction))
tab13<-colSums(tab13[-1,])
tab14<-prop.table(table(data$allowing, data$Prediction))
tab14<-colSums(tab14[-1,])

par(mfrow=c(1,2))
barplot(tab1, ylim=c(0,0.005), main="felipe",  xlab="Prediction", beside=F, col=c(3,2))
barplot(tab2, ylim=c(0,0.005), main="explosion", xlab="Prediction", beside=F, col=c(3,2))
barplot(tab3, ylim=c(0, 0.005), main="encoding", col=c(3,2), xlab="Prediction", beside=F)
barplot(tab4, ylim=c(0,0.005), main="returns", col=c(3,2), xlab="Prediction", beside=F)
barplot(tab5, ylim=c(0, 0.005), main="padre", col=c(3,2), xlab="Predicition", beside=F)
```

```
171         barplot(tab6, ylim = c(0,0.005), main="flw", col=c
                (3,2), xlab="Prediction", beside=F)
172         barplot(tab7, ylim=c(0, 0.005), main="formosa", col=c
                (3,2), xlab="Prediction", beside=F)
173         barplot(tab8, ylim=c(0, 0.005), main="dorcheus", col=
                c(3,2), xlab="Prediction", beside=F)
174         barplot(tab9, ylim=c(0, 0.005), main="pooling", col=c
                (3,2), xlab="Prediction", beside=F)
175         barplot(tab10, ylim=c(0, 0.005), main="offsystem",
                col=c(3,2), xlab="Prediction", beside=F)
176         barplot(tab11, ylim=c(0, 0.005), main="decisions",
                col=c(3,2), xlab="Prediction", beside=F)
177         barplot(tab12, ylim=c(0, 0.005), main="greatest", col
                =c(3,2), xlab="Prediction", beside=F)
178         barplot(tab13, ylim=c(0, 0.005), main="remains", col=
                c(3,2), xlab="Prediciton", beside=F)
179         barplot(tab14, ylim=c(0, 0.005), main="allowing", col
                =c(3,2), xlab="Prediction", beside=F)
180
181
182         #Analisi gruppi
183
184         #gruppo 1:economico
185         summary(data$credit)
186         prop.table(table(data$credit, data$Prediction))
187         barplot(prop.table(table(data$credit, data$Prediction
                )))
188
189         summary(data$investment)
190         prop.table(table(data$investment, data$Prediction))
191         barplot(prop.table(table(data$investment, data$
                Prediction)))
192
193         summary(data$money)
194         prop.table(table(data$money, data$Prediction))
195         barplot(prop.table(table(data$money, data$Prediction)
                ))
196
197         summary(data$offer)
198         prop.table(table(data$offer, data$Prediction))
199         barplot(prop.table(table(data$offer, data$Prediction)
                ))
200
201         summary(data$deal)
202         prop.table(table(data$deal, data$Prediction))
203         barplot(prop.table(table(data$deal, data$Prediction))
                )
204
```

```
205         g1<-cbind(data$credit, data$investment, data$offer,
                data$money, data$deal)
206         colnames(g1)<-cbind("credit", "investment", "offer",
                "money", "deal")
207         D2<-cor(g1)
208         corrplot(D2, method="color")
209
210         #gruppo 2:medico-farmaceutico
211         summary(data$medications)
212         prop.table(table(data$medications, data$Prediction))
213         barplot(prop.table(table(data$medications, data$
                Prediction)))
214
215         summary(data$drugs)
216         prop.table(table(data$drugs, data$Prediction))
217         barplot(prop.table(table(data$drugs, data$Prediction)
                ))
218
219         summary(data$treatment)
220         prop.table(table(data$treatment, data$Prediction))
                  barplot(prop.table(table(data$treatment,
                data$Prediction)))
221
222         f<-cbind(data$treatment, data$medications, data$drugs
                )
223         colnames(f)<-c("treatment" ,"medications", "drugs")
224         D3<-cor(f)
225         corrplot(D3, method="color")
226
227         #gruppo 3:informatico
228         summary(data$click)
229         prop.table(table(data$click, data$Prediction))
230         barplot(prop.table(table(data$click, data$Prediction)
                ))
231
232         summary(data$link)
233         prop.table(table(data$link, data$Prediction))
234         barplot(prop.table(table(data$link, data$Prediction))
                )
235
236         summary(data$attached)
237         prop.table(table(data$attached, data$Prediction))
                  barplot(prop.table(table(data$attached, data
                $Prediction)))
238
239         l<-cbind(data$click, data$link, data$attached)
240         colnames(l)<-cbind("click", "link", "attached")
241         D4<-cor(l)
242         corrplot(D4, method="color")
```

```
243
244         #gruppo 4:errori ortografici
245         summary(data$newsietter)
246         prop.table(table(data$newsietter, data$Prediction))
                barplot(prop.table(table(data$newsietter,
            data$Prediction)))
247
248         summary(data$technoiogies)
249         prop.table(table(data$technoiogies, data$Prediction))
              barplot(prop.table(table(data$technoiogies,
            data$Prediction)))
250
251         summary(data$pubiisher)
252         prop.table(table(data$pubiisher, data$Prediction))
                barplot(prop.table(table(data$pubiisher, data
            $Prediction)))
253
254         j<-cbind(data$newsietter, data$technoiogies, data$
            pubiisher)
255         colnames(j)<-cbind("newsietter", "technoiogies", "
            pubiisher")
256         D5<-cor(j)
257         corrplot(D5, method="color")
258
259
260         ##Riduzione del dataset
261         count_zeros <- function(x) {
262                 sum(x == 0)
263         }
264         zero_counts <- apply(data.esp, 2, count_zeros)
265         mean(zero_counts)
266         data2.esp<-data.esp[-which(rowSums(data.esp)>125|
            rowSums(data.esp)<50),-which(zero_counts>4000)]
267         dim(data2.esp)
268         head(data2.esp)
269
270         Prediction<-data$Prediction[-which(rowSums(data.esp)
            >125| rowSums(data.esp)<50)]
271         newdata<-cbind(data2.esp, Prediction)
272         head(newdata)
273         prop.table(table(newdata$Prediction))
274
275
276         ##Clustering tradizionale
277         library(cluster)
278         D<-dist(data2.esp)
279
280         #metodo delle k-medie
281         km1<-kmeans(data2.esp, centers=1, nstart=5)
```

```
km2<-kmeans(data2.esp, centers=2, nstart=5)
km3<-kmeans(data2.esp, centers=3, nstart=5)
km4<-kmeans(data2.esp, centers=4, nstart=5)
km5<-kmeans(data2.esp, centers=5, nstart=5)
km6<-kmeans(data2.esp, centers=6, nstart=5)

explained.var<-c(km1$betweenss/km1$totss,
km2$betweenss/km2$totss,
km3$betweenss/km3$totss,
km4$betweenss/km4$totss,
km5$betweenss/km5$totss,
km6$betweenss/km6$totss

)
explained.var
plot(c(1:6), explained.var, type="l", lwd=2, xlab="
    numerosità␣cluster", ylab="percentuale␣di␣varianza
    ␣spiegata")

table(newdata$Prediction, km1$cluster)
table(newdata$Prediction, km2$cluster)
table(newdata$Prediction, km3$cluster)
table(newdata$Prediction, km4$cluster)
table(newdata$Prediction, km5$cluster)
table(newdata$Prediction, km6$cluster)

library(ggfortify)
autoplot(km2, data=newdata, shape="Prediction", frame
    =T)
autoplot(km3, data=newdata, shape="Prediction", frame
    =T)

sil2<-silhouette(km2$cluster, D)
length(which(sil2[,3]<0))
sil3<-silhouette(km3$cluster, D)
length(which(sil3[,3]<0))

#metodo gerarchico
hcc<-hclust(D, method="complete")
plot(hcc, labels=F)
rect.hclust(hcc, k=2, border="red")

hcs<-hclust(D, method="single")
plot(hcs, labels=F)
rect.hclust(hcs, k=2, border="red")

hca<-hclust(D, method="average")
plot(hca, labels=F)
rect.hclust(hca, k=2, border="red")
```

```
327
328         hcw<-hclust(D, method="ward.D2")
329         plot(hcw, labels=F)
330         rect.hclust(hcw, k=2, border="red")
331         rect.hclust(hcw, k=3, border="red")
332
333         table(newdata$Prediction, cutree(hcc, k=2))
334         table(newdata$Prediction, cutree(hcs, k=2))
335         table(newdata$Prediction, cutree(hca, k=2))
336         table(newdata$Prediction, cutree(hcw, k=2))
337         table(newdata$Prediction, cutree(hcw, k=3))
338
339         sil1<-silhouette(cutree(hcc, k=2), D)
340         length(which(sil1[,3]<0))
341         sil2<-silhouette(cutree(hcs, k=2), D)
342         length(which(sil2[,3]<0))
343         sil3<-silhouette(cutree(hca, k=2), D)
344         length(which(sil3[,3]<0))
345         sil4<-silhouette(cutree(hcw, k=2), D)
346         length(which(sil4[,3]<0))
347         sil5<-silhouette(cutree(hcw, k=3), D)
348         length(which(sil5[,3]<0))
349
350         da<-diana(data2.esp)
351         plot(dat, which.plots=2, labels=F)
352         rect.hclust(da, k=2, border="red")
353         table(cutree(da, k=2))
354         table(newdata$Prediction,cutree(da, k=2))
355         sil<-silhouette(cutree(da, k=2), D)
356         length(which(sil<0))
357
358
359         ##Clustering sparso
360         library(sparcl)
361
362         #metodo delle k-medie
363         ks<-KMeansSparseCluster.permute(data2.esp, K=2)
364         print(ks)
365         plot(ks)
366
367         kms<-KMeansSparseCluster(data2.esp,K=2, wbounds=ks$
              bestw)
368         print(kms)
369         plot(kms, xlab="variabili", ylab="pesi" ,main="
              Distribuzione␣dei␣pesi")
370
371         ks<-KMeansSparseCluster.permute(data2.esp, K=3)
372         print(ks)
373         plot(ks)
```

```
374
375         kms<-KMeansSparseCluster(data2.esp,K=3, wbounds=ks$
                bestw)
376         print(kms)
377         plot(kms, main="p")
378
379         #metodo gerarchico
380         h<-HierarchicalSparseCluster.permute(as.matrix(data2.
                esp))
381         print(h)
382         plot(h)
383
384         sh<-HierarchicalSparseCluster(as.matrix(data2.esp),
                wbound=h$bestw, method="complete")
385         plot(sh)
386         par(mfrow=c(1,1))
387         plot(sh$hc, labels=F)
388         plot(sh$ws, ylab="pesi", xlab="variabili", main="
                Distribuzione␣dei␣pesi")
389         rect.hclust(sh$hc, k=2, border="red")
390         print(sh)
391         sum(sh$ws^2)
392         table(cutree(sh$hc, k=2))
393         table(newdata$Prediction, cutree(sh$hc, k=2))
394
395         sh2<-HierarchicalSparseCluster(as.matrix(data2.esp),
                wbound=h$bestw, method="single")
396         print(sh2)
397         plot(sh2)
398         plot(sh2$hc, labels=F)
399         plot(sh2$ws, ylab="pesi", xlab="variabili", main="
                Distribuzione␣dei␣pesi")
400         rect.hclust(sh2$hc, k=2, border="red")
401         table(cutree(sh2$hc, k=2))
402         table(newdata$Prediction, cutree(sh2$hc, k=2))
403
404         sh3<-HierarchicalSparseCluster(as.matrix(data2.esp),
                wbound=h$bestw, method="average")
405         print(sh3)
406         plot(sh3)
407         plot(sh3$hc, labels=F)
408         rect.hclust(sh3$hc, k=2, border="red")
409         table(cutree(sh3$hc, k=2))
410         table(newdata$Prediction, cutree(sh3$hc, k=2))
```