UNIVERSITÀ DEGLI STUDI DI SALERNO

Tesi di Laurea Triennale

Identificazione della correlazione tra dati biomedici attraverso tecniche di Machine Learning

Relatore:
Prof. ssa Tortora Genoveffa

Correlatori:
Prof. Risi Michele
Dott. ssa Maria Frasca

Laureando:
Russo Vincenzo
Matr: 0512104130

Anno Accademico 2018/2019

Abstract

L'oggetto di questa ricerca è la correlazione tra i dati medici, forniti dalla PPMI aggiornata a luglio 2019 nelle visite dei pazienti.
il principale obiettivo è un sistema di supporto ai medici per classificare in maniera veloce i pazienti sani dai malati, utilizzando i report realizzati durante le visite.
È stato sviluppato un processo che permette la classificazione dei pazienti, utilizzando tecniche di machine learning non supervisionato fra cui LSA, che basa sull'analisi dei documenti per trovare per trovarne il significato sottinteso, ci aspetteremmo di associare ad una parola un concetto, ma nella vita reale usiamo i sinonimi. Ciò che realmente ci interessa è confrontare i concetti, non le parole. LSA tenta di risolvere questo problema mappando parole e documenti in uno spazio dei concetti.
Un ulteriore obiettivo è stato estendere il processo ed impiegare una tecnica basata sulle reti neurali chiamata text2vec, essa crea una rappresentazione numerica del documento e cerca dei documenti simili. Entrambe le le tecnologie ci restituiscono una matrice delle similarità tra i vari pazienti, che vengono utilizzate dagli algoritmi di clustering per classificarli. Gli algoritmi di clustering raggruppano gli elementi sulla base della loro similarità reciproca, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme stesso.
In questo studio vengono utilizzate due tecniche il clustering k-means in questa un paziente può appartenere esattamente ad un solo cluster. Al contrario nel Fuzzy clustering un paziente può appartenere ai due cluster con una determinata probabilità. Abbiamo assunto per denominare i cluster, una supervisione basata sulla maggioranza dei pazienti malati.
Si è scoperto che la tecnica LSA fornisce i risultati migliori durante la visita SC (screening) accoppiata ad entrambi gli algoritmi di clustering. La situazione si sconvolge nel gruppo di visite SC-V03 in quanto LSA con K-means diminuisce ampiamente e al contempo la tecnica text2vec con K-means aumenta sensibilmente, text2vec con Fuzzy discretamente. I processi con i risultati maggiormente accurati nel gruppo di visite SC-LOG sono LSA con Fuzzy e text2Vec con K-means. text2vec con Fuzzy è calato notevolmente, mentre LSA con K-means ha recuperato leggermente ma con molto divario rispetto alle prime due.
In conclusione entrambi i processi hanno avuto buoni risultati, ma ciò dipende dal tipo di tecnica utilizzata. Nel nostro caso text2vec con K-means e LSA con Fuzzy, ci hanno offerto i migliori risultati, considerando tutte le visite. In futuro l'analisi dei dati e la correlazione tra essi potrà essere ulteriormente estesa. In particolare, i possibili sviluppi possono essere: L'allenamento della rete neurale text2vec utilizzando vocaboli medici.

Introduzione

Nel corso dell'ultimo decennio l'applicazione dell'informatica combinata alle attività di ricerca in ambito medico ha subito una spinta senza precedenti, consentendo l'estrapolazione di un'enorme quantità di informazioni, raccolte e raggruppate successivamente, in banche di dati, utilizzate per interrogare i profili di espressione genetica, identificare l'interazione biologica e per ottenere la visualizzazione dei dati biomedici. Da ciò nasce l'esigenza di disporre di dati ben organizzati, curati e standardizzati, nell'ottica di formulare delle diagnosi precise e trarre nuove indicazioni biologiche.

L'iniziativa PPMI è uno studio di riferimento lanciato nel 2010 che ha come scopo quello di trovare dei nuovi biomarcatori, ovvero degli indicatori di malattia che rappresentano dei collegamenti mancanti, importanti nella ricerca di trattamenti migliori per la malattia di Parkinson. Lo studio ha portato alla costruzione, grazie alla cooperazione di numerosi ricercatori, di un ampio dataset costituito dai dati e i campioni raccolti e acquisiti dai partecipanti (volontari e quasi tutti affetti dalla malattia).

Si vuole identificare una correlazione tra i dati biomedici, presenti all'interno del dataset PPMI [**ppmi**], attraverso delle tecniche di Information Retrieval, al fine di verificare se le analisi formulate durante le visite dai medici siano coerenti fra loro riuscendo a categorizzare i vari pazienti in modo corretto. Tutto ciò avverrà tramite l'identificazione di un processo di elaborazione per correlare le informazioni delle visite di ogni paziente attraverso, inizialmente, i) l'utilizzo della tecnica di latent semantic analysis [**lsa**] (LSA) per costruire uno spazio dei concetti sulle informazioni dei pazienti, ii)l'utilizzo della tecnica text2vec [**text2vec**], fondata sulle reti neurali, per costruire uno spazio dei concetti alternativo iii) l'utilizzo di algoritmi di clustering [**cluster**] per poter raggruppare i pazienti in base alle descrizioni delle visite, ed infine iv) una fase di data analysis per poter assegnare una diagnosi ai gruppi di pazienti.

La tesi è strutturata come di seguito:

- Nel primo capitolo viene descritto in modo dettagliato il dataset PPMI.
- Nel secondo capitolo vengono descritte tutte le tecnologie utilizzate per poter implementare i processi di correlazione tra i dati.
- Nel terzo capitolo vengono definiti i processi di correlazione fra i dati tra i pazienti e le visite mediche effettuate.
- Nel quarto capitolo viene descritto nel dettaglio gli script che implementano il processi e fornisce come risultato le diagnosi.
- Nel quinto capitolo vengono discussi e motivati i risultati raggiunti, quindi delineate le conclusioni e prospettati gli sviluppi futuri.

Capitolo 1

Parkinson's Progression Marker Initiative (PPMI)

L'iniziativa "Parkinson's Progression Markers Initiative" (PPMI) [**ppmi**] è uno studio clinico basato solo ed esclusivamente sull'osservazione, per valutare in modo completo le coorti di interesse significative utilizzando l'imaging avanzato, il campionamento biologico e le valutazioni cliniche e comportamentali per identificare i biomarcatori della progressione della malattia di Parkinson che potrebbero offrire ai ricercatori uno strumento essenziale per la ricerca di terapie in grado di rallentare o arrestare la progressione del morbo.

È uno studio "open source", i dati e i campioni raccolti e acquisiti dai partecipanti (volontari e quasi tutti affetti dalla malattia) permetteranno lo sviluppo di un database e di un biorepository completa, che è attualmente disponibile online e aggiornata ogni 8 mesi. Può essere scaricata mediante l'accesso al portale del sito del PPMI per permettere alla comunità scientifica di condurre ricerche complete ed esaustive.

La PPMI si svolge presso i siti clinici negli Stati Uniti, in Europa, Israele ed Australia ed è resa possibile dagli sforzi di una collaborazione di ricercatori e finanziatori ma, in particolar modo, è sponsorizzata da "The Michael J.Fox Foundation for Parkinson's Research", una fondazione creata nel 2000 dall'omonimo attore Michael J.Fox, affetto da tale patologia da 28 anni.

1.1 Parkinson disease

Il morbo di Parkinson si manifesta quando la produzione di dopamina nel cervello cala consistentemente. I livelli ridotti di dopamina sono dovuti alla degenerazione di neuroni, in un'area chiamata "sostanza nera" (la perdita cellulare è di oltre il 60% all'esordio dei sintomi), inoltre dal midollo al cervello cominciano a comparire anche accumuli di una proteina chiamata alfa-sinucleina, forse è proprio questa proteina che diffonde la malattia in tutto il cervello.

Le cause non sono ancora note ma, sembra che vi siano molteplici elementi che concorrono al suo sviluppo. Questi fattori sono principalmente:

- Alcune mutazioni genetiche
- Fattori tossici ed esposizione lavorativa

Per diagnosticare la malattia di Parkinson nella maggior parte degli studi sono utilizzati i criteri diagnostici previsti da Schoenberg, i quali per la diagnosi di parkinsonismo considerano essenziali i quattro sintomi motori cardinali della PD:

– il tremore a riposo,

– la rigidità degli arti e del tronco,

– la bradicinesia (la lentezza del movimento),

– l'instabilità posturale (problemi di equilibrio).

La diagnosi della malattia di Parkinson resta tuttora una diagnosi clinica poiché non esiste un test obiettivo o dei marcatori biochimici e neuroradiologici specifici. Nell'ultimo decennio, però, uno degli obiettivi della ricerca è stato migliorare la specificità dei criteri diagnostici classici: infatti la "United Kingdom Parkinson's disease Society Brain Bank" ha proposto criteri clinici che tuttora sono ampiamente utilizzati nella pratica clinica e nei protocolli di ricerca. Tali criteri diagnostici stabiliscono che il segno necessario per porre diagnosi della malattia del Parkinson sia la bradicinesia o acinesia, associata ad almeno uno degli altri segni cosiddetti maggiori, accennati precedentemente, ovvero la rigidità muscolare, tremore a riposo e l'instabilità posturale.
Tali criteri diagnostici sono stati recentemente rivisti da Gelb, Oliver e Gilman nel libro "Diagnosti Criteria for Parkinon's Disease" dove sottolineano come la diagnosi clinica sia basata sulla combinazione di alcuni segni motori "cardinali" e sull'esclusione di sintomi ritenuti "atipici".

In conclusione, i sintomi del morbo di Parkinson si manifestano in modo diverso nei diversi pazienti, i quali possono sperimentare alcuni sintomi e non altri e anche il ritmo con cui la malattia progredisce varia da individuo a individuo. Per questo, il tasso di diagnosi errata può essere relativamente alto.

1.2 Descrizione del Dataset

Il dataset è in formato CSV, consta di 131 file contenenti informazioni di ogni genere (3488 variabili): dai valori rilevati da un semplice prelievo di sangue, alle risposte di questionari neuro-cognitivi fino a passare a risultati di esami diagnostici, tutti raccolti sotto forma di tabelle e collocabili in specifiche aree di riferimento classificabili in base al tipo di studio condotto sul paziente.

Sono state individuate 6 macro-aree e sono classificate come segue:

1. **Biospecimen:** raccolta di dati relative ad esami clinici quali: prelievo del sangue, DNA, puntura lombare.
2. **Imaging:** utilizzo di tecniche di imaging, come la Risonanza Magnetica, DatScan grazie alle quali è possibile osservare aree non visibile dell'organismo.
3. **Medical History:** storia clinica dei pazienti dai primi sintomi della patologia alle ultime condizioni di salute. La raccolta include eventuali effetti collaterali dei medicinali assunti, risultati degli esami neurologici, fisici e cosi via.
4. **Motor MDS –UPDRS:** raccolta di dati relativi a disturbi motori attraverso l'utilizzo della scala MDS-UPDRS per valutare lo stadio della malattia del Parkinson. La scala è cosi organizzata:
   - Parte 1: esamina le esperienze non motorie della vita quotidiana ed e divisa in due componenti:
     - Parte 1a: contiene sei domande che vengono valutate dall'investigatore si concentra sui comportamenti complessi;
     - Parte 1b: contiene sette domande che fanno parte del questionario.
   - Parte 2: valuta le esperienze motorie della vita quotidiana. Ci sono ulteriori domande che sono anche parte del questionario della parte 1b.
   - Parte 3: valuta i segni motori del PD ed e amministrato dall'investigatore.
   - Parte 4: valuta complicanze motorie, discinesie (alterazione del movimento) e fluttuazioni motorie utilizzando informazioni storiche ed oggettive.

     L'investigatore completerà questa valutazione una volta che il soggetto ha iniziato la cura.
5. **Non Motor Assessments:** raccolta di dati relativi a disturbi cognitivi ed emotivo-comportamentali.
6. **Study Enrollment:** raccolta di dati conclusivi su particolari studi condotti sui pazienti.

Nell'analisi del dataset, di particolare importanza sono risultati due file, necessari per la comprensione dei dati presenti nel dataset: Data_Dictionary.csv e Page_Descriptions.csv.

- Il **Data_Dictionary.csv** descrive il significato delle variabili presenti all'interno di ogni file.
- **Page_Descriptions.csv** è un dizionario delle abbreviazioni dei moduli in cui ricercare una variabile di interesse.

Dalla fase di ricognizione di questi due file è stato possibile capire come sono stati categorizzati i pazienti, quali visite sono state effettuate in relazione alla categoria e quali sono stati i tipi di test condotti sui singoli pazienti.

La classificazione dei pazienti nel PPMI è strutturata come segue:

- **HC:** controlli sani – non portatori della malattia.
- **PD:** malati di Parkinson.
- **GENUN:** genetic Unaffected.
- **GENPD:** genetic PD.
- **SWEDD:** soggetti senza evidenti deficit dopaminergici – deficit di dopamina.
- **PRODOMAL:** soggetti che soffrono di insonnia e presentano mutazioni del gene LRRK2.

La classificazione delle visite, invece, è come segue:

- **LOG:** visita della registrazione degli eventi avversi
- **SC:** Screening Visit - precedente alla visita di baseline dura circa 8 ore.
- **BL:** Baseline Visit.
- **V01-V15:** sequenza di test programmati per tutti i pazienti.
- **St** (Symptomatic Therapy): trattamento volto ad attenuare i sintomi del morbo.
- **Unsch.visit:** visite non programmate che possono essere effettuate in qualsiasi momento.

Di seguito viene riportato lo schema ER di alto livello del dataset PPMI, in cui sono presenti cinque entità:

- **PATIENT:** rappresenta l'insieme dei pazienti che partecipano allo studio.
- **EVENT:** rappresenta l'insieme delle tabelle che fanno riferimento alle visite e analisi a cui sono sottoposti i pazienti.
- **BIOSPECIMEN ANALYSIS RESULT:** rappresenta l'insieme delle tabelle in cui sono presenti le analisi dei risultati per i controlli a cui i pazienti sono stati sottoposti.
- **FAMILY HISTORY:** rappresenta l'insieme delle tabelle in cui vengono descritte le storie familiari dei pazienti.
- **MEDICATION:** rappresenta l'insieme delle tabelle in cui vengono catalogati tutti i medicinali presi dai pazienti.

Le entità sopra descritte sono collegate fra loro attraverso delle relazioni:

- **R:** rappresenta la relazione che esiste tra le entità, EVENT, PATIENT E BIOSPECIMEN ANALISYS RESULT, attraverso il PATNO, identificatore numerico univoco per ogni paziente.
- **HAS:** relazione che rappresenta il collegamento tra i pazienti e la loro storia familiare
- **ASSUME:** relazione che associa ad ogni paziente i medicinali che assumono.

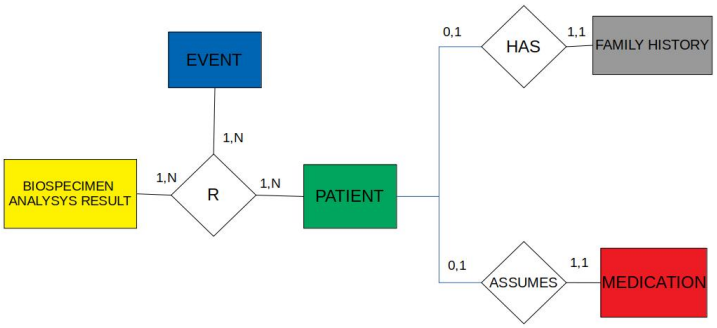

Figura 1.1: Schema ER.

Capitolo 2

Tecnologie utilizzate

In questo capitolo verranno descritte tutte le tecnologie utilizzate per la trasformazione del dataset e per l'analisi sintattica effettuate su quest'ultimo.

2.1 Tecnologie per la gestione del dataset

Per la trasformazione del dataset da valori numerici a valori testuali è stato scelto Microsoft Excel [**excel**], il dataset trasformato è stato memorizzato sul filesystem.

2.1.1 Microsoft Excel

Microsoft Excel ha le funzionalità di base di tutti i fogli di calcolo, utilizzando una griglia di celle disposte in righe numerate e colonne con nome lettera per organizzare manipolazioni dei dati come operazioni aritmetiche. Ha una serie di funzioni fornite per rispondere alle esigenze statistiche, ingegneristiche e finanziarie.

Ha un aspetto di programmazione, Visual Basic, Applications Edition, che consente all'utente di utilizzare un'ampia varietà di funzioni pronte all'uso come concatenazione e sostituzione di valori.

Di seguito verranno elencate alcune principali caratteristiche:

- *Foglio di calcolo:* Il programma funziona su dati immessi nelle celle di una tabella. Ogni cella può contenere dati numerici o di testo oppure i risultati di formule che calcolano e visualizzano automaticamente un valore in base al contenuto di altre celle.

- *Visual Basic for Applications:* La programmazione con VBA consente la manipolazione del foglio di calcolo che è scomoda o impossibile con le tecniche di foglio di calcolo standard. Ad esempio è possibile selezionare una colonna e concatenare una stringa ad ogni elemento della colonna.

2.2 Tecnologie utilizzate per il text mining

2.2.1 L'ambiente integrato R

R [**r**] è un potente sistema, che contemporaneamente è un linguaggio di programmazione ed un software libero (in quanto viene distribuito con la licenza GNU GPL), costituito da una varietà di strumenti per l'analisi statistica dei dati e per la loro visualizzazione.
Il suo linguaggio orientato agli oggetti deriva direttamente dal pacchetto S, distribuito con una licenza non open source negli anni '80 ,sviluppato da Rick A. Becker, John Chambers e Allan Wilks presso i Bell Laboratories, dando origine ad un noto software commerciale S-PLUS.
La versione iniziale di R è stata realizzata nel 1996 da Ross Ihaka e Robert Gentleman del Dipartimento di Statistica dell'Università di Auckland in Nuova Zelanda. Successivamente, un nutrito gruppo di ricercatori operanti in ambito statistico e informatico hanno iniziato a fornire il loro contributo, dando così vita al "R Development Core Team", che dal 1997 si occupa dei codici sorgenti di R. Nel 2003 'e stata costituita l'organizzazione non-profit "R Foundation for Statistical Computing" avente come obiettivo quello di promuovere lo sviluppo e la diffusione del software, di gestire e tutelare il copyright di R e della relativa documentazione. Il linguaggio R è stato sviluppato in modo tale da mantenere la massima compatibilità con il software commerciale S-Plus poiché se le risorse economiche lo permettono viene utilizzato quest'ultimo e non R.

Le caratteristiche principali di R possono essere così sintetizzate:

- è un software open source;
- è un linguaggio di programmazione object-oriented (come C++ e Java) e quindi l'utente finale può avere accesso al codice interno di R ed eventualmente proporne modifiche;
- è un linguaggio di programmazione interpretato e quindi il sistema è in grado di elaborare le frasi inserite immediatamente, senza dover passare attraverso un processo di compilazione;
- è un software multipiattaforma, ossia può essere installato su Unix, Windows o Mac;
- è semplice da utilizzare nella gestione e nella manipolazione dei dati;
- è dotato di notevoli e particolarmente flessibili potenzialità grafiche 2D e 3D consentendo la rappresentazione grafica di dati;
- dispone di un insieme di strumenti per il calcolo su vettori, matrici, data frame e per altre operazioni complesse;
- consente l'accesso a un vasto insieme di strumenti integrati per analisi statistiche;
- fornisce la possibilità di programmare, creando funzioni e programmi ad hoc definiti dall'utente;

- è dotato di una funzione di help in linea per ciascun comando facilmente richiamabile dal programma;
- possiede numerosi data frame di esempio e ottimi manuali di riferimento (in lingua inglese) consultabili o scaricabili direttamente da Internet.

R eredita da S le caratteristiche di essere un linguaggio interpretato e di tipo object-oriented. Queste due caratteristiche rendono il linguaggio molto flessibile e facilmente estendibile attraverso la creazione di nuove funzioni definite dall'utente. R, inoltre, consente di programmare calcoli matematici e statistici e di effettuare analisi e computazioni anche molto complesse.
Tutto, in questo ambiente, viene rappresentato mediante oggetti. Ogni oggetto (vettore, dataset, tabella, grafico, ecc.) è trattato dalle funzioni di R con uno specifico metodo ed è possibile implementare nuovi metodi per ampliare le possibilità delle stesse funzioni.
Questo linguaggio ha avuto un notevole successo grazie all'ampia disponibilità di moduli distribuiti con la licenza GPL che estendono di molto le capacità del programma e sono organizzati in un apposito sito chiamato "CRAN" dal quale è possibile scaricare e installare i pacchetti in base alle esigenze dell'utente.
Inizialmente nell'ambiente vi è presente solo il cosiddetto "modulo base" che offre gli strumenti fondamentali per effettuare le usuali operazioni di lettura e scrittura dei dati da e su file, le operazioni su matrici e vettori e le elaborazioni statistiche connesse alla statistica descrittiva e inferenziale, alla regressione, all'analisi esplorativa dei dati, alla produzione di grafici e alla simulazione di variabili aleatorie.
Anche se il linguaggio è fornito con un'interfaccia a riga di comando, sono disponibili interfacce grafiche, di seguito verrà descritta quella utilizzata in questo lavoro di tesi.

2.2.2 R STUDIO

RStudio è un ambiente di sviluppo integrato (IDE) gratuito e open source per R , è in parte scritto nel linguaggio di programmazione C ++ , utilizza il framework Qt per la sua interfaccia utente grafica ma la percentuale più grande del codice è scritta in Java.
Il lavoro su RStudio è iniziato verso dicembre 2010 con la prima versione beta e nell'Ottobre 2017 è stata rilasciata l'ultima versione.
RStudio rende R più facile da usare includendo un editor di codice, strumenti di debug e visualizzazione, aiuta a realizzare applicazioni web interattive per la visualizzazione dei dati ed inoltre, include numerosi pacchetti per espandere le funzionalità di R.

2.2.3 Document-term matrix

Una document-term matrix è una matrice che descrive la frequenza dei termini che appaiono in una raccolta di documenti. Le righe corrispondono ai documenti e le colonne ai termini. Esistono vari modi per determinare i valori all'interno della matrice, uno di questi è la tf-idf.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	…
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
…							

Figura 2.1: Un esempio di document-term matrix trasposta.

2.2.4 tf-idf

La funzione di peso tf-idf (term frequency–inverse document frequency) è una funzione utilizzata in information retrieval per misurare l'importanza di un termine rispetto ad un documento o ad una collezione di documenti. Tale funzione aumenta proporzionalmente al numero di volte che il termine è contenuto nel documento, ma cresce in maniera inversamente proporzionale con la frequenza del termine nella collezione. L'idea alla base di questo comportamento è di dare più importanza ai termini che compaiono nel documento, ma che in generale sono poco frequenti.

La funzione può essere scomposta in due fattori: Il primo fattore della funzione è il numero dei termini presenti nel documento. In genere questo numero viene diviso per la lunghezza del documento stesso per evitare che siano privilegiati i documenti più lunghi.

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{|d_j|}$$

dove $n_{i,j}$ è il numero di occorrenze del termine i nel documento j, mentre il denominatore $|d_j|$ è semplicemente la dimensione, espressa in numero di termini, del documento j. L'altro fattore della funzione indica l'importanza generale del termine i nella collezione:

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : i \in d\}|}$$

dove $|D|$ è il numero di documenti nella collezione, mentre il denominatore è il numero di documenti che contengono il termine i. Abbiamo quindi che:

$$\text{(tf-idf)}_{\mathrm{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$$

Consideriamo un documento contenente 100 parole e nel quale il termine pluto compare 5 volte. Il fattore TF per il termine pluto è $5/100 = 0{,}05$. Assumiamo di avere ora 1 000 documenti nella collezione e pluto compare in 10 di questi. Quindi $IDF = \log 1000/10 = 2$. Da questo possiamo calcolare il valore Tf-idf relativo alla parola pluto nel documento iniziale: TF-IDF $= 0.05 \times 2 = 0.1$.

2.2.5 Stemmer

Lo stemmer è un'algoritmo che riduce le parole alla forma base, infatti il suffisso delle parole viene rimosso e resta soltanto la parte iniziale della parola (stem).

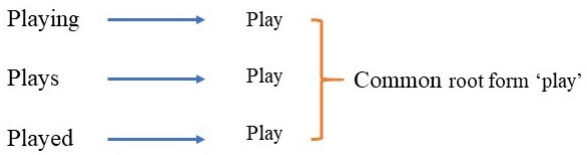

Figura 2.2: Esempio di stemming.

Un'interfaccia R per la libreria C libstemmer [**snowballc**] che implementa l'algoritmo di Porter stemming (o 'Porter stemmer'), un processo per rimuovere le terminazioni morfologiche e inflessionali più comuni delle parole e per facilitare il confronto dato un dizionario specificato.
Le lingue attualmente supportate sono Danese, olandese, inglese, finlandese, francese, tedesco, ungherese, italiano, Norvegese, portoghese, rumeno, russo, spagnolo, svedese e turco.

2.2.6 Libreria "tm"

Il text mining [**tm**] consiste nell'applicazione di tecniche di Data Mining (l'insieme delle tecniche e delle metodologie che hanno per oggetto l'estrazione di informazioni utili da grandi quantità di dati) a testi non strutturati e più in generale a qualsiasi corpus di documenti.
La struttura principale per la gestione dei documenti in tm è un cosiddetto Corpus, che rappresenta una raccolta di testo-documenti.
Un corpus è un concetto astratto e possono esistere diverse implementazioni in parallelo, l'implementazione di default è la cosiddetta VCorpus (abbreviazione di Volatile Corpus) che realizza degli oggetti R, i corpora, che sono tenuti completamente in memoria.
Un corpus è detto volatile perché una volta distrutto, l'intero corpo viene eliminato. Un'altra implementazione può essere eseguita tramite PCorpus in cui i documenti sono fisicamente salvati al di fuori dell'ambiente R e gli oggetti creati in R sono semplicemente degli indicatori a strutture esterne. In questo caso se gli oggetti R vengono rilasciati, gli oggetti del corpus non vengono distrutti.
Al costruttore di queste due implementazioni viene passato un oggetto Source che astrae

la posizione di input; tm fornisce un insieme di fonti predefinite (DirSource, VectorSource o DataframeSourc) che gestiscono una directory, interpretando ciascun componente come documento o strutture come data frame (come i file CSV).
Il corpus può essere importato oppure, se non esiste, si può creare una directory e scrivere al suo interno tramite la funzione writeCorpus(), inoltre, sono disponibili metodi di stampa personalizzati che nascondono la quantità di informazioni non elaborate (si consideri un corpus composto da diverse migliaia di documenti). Queste funzioni sono essenzialmente due, print() che fornisce una panoramica concisa mentre altri dettagli vengono visualizzati con inspect().
Una volta che abbiamo un corpus, intendiamo modificare i documenti al suo interno.
In tm, tutte queste funzionalità sono incluse nel concetto di trasformazione. Le trasformazioni avvengono tramite la funzione tm_map() che applica una funzione a tutti gli elementi del corpus.

Le principali trasformazioni che si possono applicare su un corpus sono:

- rimozione degli spazi bianchi
- rimozione delle stopword ossia parole comuni del linguaggio che non aggiungono significato al discorso
- conversione in lower case
- effettuare lo stemming del documento (creare una singola rappresentazione della parola indipendentemente dal suo tempo verbale)

Spesso è di particolare interesse filtrare i documenti che soddisfano determinate proprietà, a questo scopo la viene utilizzata la funzione tm_filter(). Successivamente un approccio comune nel text mining è quello di creare una matrice termini-documento da un corpus e per questo compito in questo pacchetto sono disponibili le classi TermDocumentMatrix e DocumentTermMatrix (a seconda se si desidera i termini come righe e i documenti come colonne o viceversa). Inoltre, si ha a disposizione, anche, la funzione findFreqTerms() grazie alla quale possiamo trovare delle informazioni in base al loro contenuto e alla loro frequenza.
Le matrici termine-documento tendono ad essere molto grandi. Pertanto, si utilizza un metodo per rimuovere i termini sparsi, cioè termini che si verificano solo in pochissimi documenti.
Normalmente, questo riduce la dimensione della matrice senza perdere significative relazioni. Inoltre, per ovviare a questo problema, è possibile utilizzare un dizionario.
Un dizionario è un (multi-) set di stringhe. Viene spesso utilizzato per indicare termini rilevanti nel text mining e rappresentato come un vettore di caratteri che può essere passato al costruttore DocumentTermMatrix() come controllo. Quindi, la matrice creata viene tabulata sul dizionario, cioè solo termini dal dizionario appaiono nella matrice, ciò consente di limitare la dimensione della matrice a priori e di focalizzarsi su termini specifici.

2.2.7 Decomposizione ai valori singolari

In algebra lineare, la decomposizione ai valori singolari, detta anche SVD (dall'acronimo inglese Singular Value Decomposition), è una particolare fattorizzazione di una matrice basata sull'uso di autovalori e autovettori. Data una matrice M reale o complessa di dimensione $m \times nm \times n$, si tratta di una scrittura del tipo:

$M = U\Sigma V^* M = U\Sigma V^*$

dove U è una matrice unitaria di dimensioni $m \times m$, Σ è una matrice diagonale rettangolare di dimensioni $m \times n$ e V^* è la trasposta coniugata di una matrice unitaria V di dimensioni $n \times n$.

Gli elementi di Σ sono detti valori singolari di M; ognuna delle m colonne di U è detta vettore singolare sinistro mentre ognuna delle n colonne di V è detta vettore singolare destro. Si verifica che:

- I vettori singolari di sinistra di M sono gli autovettori di MM^*.
- I vettori singolari di destra di M sono gli autovettori di M^*M.
- I valori singolari non nulli di M (che si trovano sulla diagonale principale di Σ) sono le radici quadrate degli autovalori non nulli di MM^* e M^*M.

2.2.8 Similarità del coseno

La similarità del coseno, o cosine similarity, è una tecnica euristica per la misurazione della similitudine tra due vettori effettuata calcolando il coseno tra di loro, usata generalmente per il confronto di testi nel data mining e nell'analisi del testo. Dati due vettori di attributi numerici, A e B, il livello di similarità tra di loro è espresso utilizzando la formula:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}.$$

Nel caso tipico del confronto fra testi il contenuto dei due vettori è la frequenza dei termini, ossia il numero di volte in cui una certa parola ricorre all'interno del testo. Il k-simo elemento di ogni vettore conterrà dunque il numero di volte in cui la parola numerata con k ricorre nel testo, oppure 0 se non ricorre mai.

In base alla definizione del coseno, dati due vettori si otterrà sempre un valore di similitudine compreso tra -1 e +1, dove -1 indica una corrispondenza esatta ma opposta (ossia un vettore contiene l'opposto dei valori presenti nell'altro) e +1 indica due vettori uguali.

Nel caso dell'analisi dei testi, poiché le frequenze dei termini sono sempre valori positivi, si otterranno valori che vanno da 0 a +1, dove +1 indica che le parole contenute nei due testi sono le stesse (ma non necessariamente nello stesso ordine) e 0 che non c'è nessuna parola che appare in entrambi.

Per rendere più efficace il confronto, in genere, si eliminano le parole più corte e molto frequenti che servono a costruire le frasi, come e, che, ma, quindi e altre, che possono essere identificate velocemente con un'euristica appropriata. È possibile anche usare la similarità per riconoscere la lingua in cui è scritto un testo, senza ovviamente ignorare le parole corte e frequenti.

In genere, questa euristica viene usata per confrontare degli elementi che sono indicati da dei parametri il cui numero e significato non è noto a priori.

2.2.9 Latent semantic analysis

L'analisi semantica latente (LSA) [**lsa**] afferma che un qualsiasi testo ha una struttura di ordine superiore (= semantica latente) oscurata, però, dall'uso delle parole (ad esempio attraverso l'uso di sinonimi o polisemia).
Usando le varie funzioni messe a disposizione dal pacchetto questo problema di variabilità può essere superato.
L'idea generale è che esiste un insieme di dipendenze latenti tra le parole e i loro contesti (frasi, paragrafi e testi), entrambi rappresentati nello stesso spazio semantico. I termini possono essere correlati l'uno con l'altro anche se non coincidono nello stesso documento purché entrambi i termini coincidano con altri termini condivisi; le co-occorrenze di termini attraverso i documenti possono indicare che anche i documenti, e non solo i termini all'interno di essi, possono essere presi in considerazione in gruppi basati su tali co-occorrenze.
L'algoritmo LSA applica un approccio chiamato "bag-of-words": tipicamente analizza le parole indipendentemente dal loro ruolo nel discorso (come nome, verbo, aggettivo, avverbio) o dalla loro posizione nella frase.
LSA combina il classico modello spaziale vettoriale - ben noto in text mining - con una "Singular Value Decomposition" (SVD), un'analisi fattoriale in due modi. Così facendo, le rappresentazioni nel testo delle parole possono essere mappate in uno spazio vettoriale modificato che si presume rifletta la struttura semantica.
Il processo del LSA può essere riassunto in quattro passi:

- *Preparazione dei documenti:* si costruisce un corpus di documenti selezionando solo quelli che coincidono con il contesto in cui risiedono I fenomeni di interesse, poiché, anche l'inclusione di una vasta raccolta di documenti di alta qualità potrebbe fallire se il contesto di tali documenti non è allineato con il fenomeno di interesse. Inoltre, è necessario avere un corpus molto grande per creare un campione rappresentativo e per aumentare le probabilità che una parola compaia in esso.

- *Creazione dello spazio semantico:* una volta creato il corpus i documenti vengono pre-elaborati, in questo lavoro di tesi questa trasformazione viene effettuata tramite il pacchetto "tm", descritto nel paragrafo precedente. Tutti i documenti all'interno del corpus devono avere le stesse trasformazioni applicate nello stesso ordine utilizzando le stesse regole e anche tutti gli pseudo-documenti proiettati sul corpo devono usare le stesse trasformazioni; non farlo potrebbe produrre risultati senza senso. Una volta che le parole/termini sono stati trasformati, il passo successivo nella creazione dello spazio semantico sta creando la DTM. Essa è una matrice che contiene i termini nel corpus come righe e i documenti come colonne. Le celle contengono il numero di volte in cui un termine appare all'interno di un documento ed è noto come conteggio grezzo. Successivamente, una seconda serie di trasformazioni viene applicata in base alla distribuzione dei termini all'interno del corpus. Questo processo è noto come ponderazione e si presenta in due forme: ponderazione locale e ponderazione globale. Nella ponderazione locale, viene data maggiore importanza ai termini che appaiono più volte all'interno di un singolo documento, nella ponderazione globale, viene data minore importanza ai termini che appaiono in un numero maggiore di documenti all'interno del corpus. Questi pesi locali vengono utilizzati per tenere conto del crescente aumento di importanza delle apparizioni aggiuntive di un termine in un documento. I pesi globali, invece, spiegano quanto spesso un termine appare in altri documenti sulla base del concetto che i termini che appaiono in molti documenti sono meno importanti. Il passo finale nella creazione dello spazio semantico è eseguire SVD. Quando si esegue, è necessario selezionare un numero appropriato di dimensioni, noto anche come "rango". Una volta eseguito produrrà tre matrici che costituiranno lo spazio semantico.

- *Proiezione di pseudo-documenti:* è il processo di creazione di vettori per testi che non erano già presenti come documenti nel corpus. Se l'obiettivo dell'analisi è esaminare le relazioni tra i documenti all'interno del corpus, non è necessario proiettare gli pseudo-documenti nello spazio semantico perché esiste già l'accesso ai vettori per ciascun documento. La proiezione di pseudo-documenti è necessaria solo, quando si confrontano, ad esempio, i coseni di due nuove frasi mentre vengono proiettate nello spazio semantico.

- *Confronto dei vettori:* sono in genere confrontati l'un l'altro con la somiglianza del coseno. Queste somiglianze vengono utilizzate per trovare quali vettori sono più simili tra loro e quali documenti hanno una somiglianza superiore a una soglia specificata.

2.2.10 Text2vec

Text2vec [**text2vec**] è un pacchetto estremamente utile per la costruzione di algoritmi di apprendimento automatico basati su dati di testo. Questo pacchetto consente di costruire una matrice di termini del documento (DTM). Pertanto, elaborare il testo creando una mappa da parole in uno spazio vettoriale. Questo pacchetto è efficiente perché è accuratamente scritto in C ++, il che significa anche che text2vec è moderato sull'utilizzo di memoria.

text2vec è basato sul concetto del *Word Embedding*, una metodologia dell'elaborazione del linguaggio naturale per mappare parole o frasi presenti in un vocabolario, in un corrispondente vettore di numeri reali, utilizzato per scoprire correlazioni semantiche tra esse.
Per identificare Documenti simili utilizziamo identicamente la similarità del coseno.

- *Preparazione dei documenti:* Anche in questo caso si costruisce un corpus di documenti selezionando solo quelli che coincidono con il contesto in cui risiedono I fenomeni di interesse, poiché, anche l'inclusione di una vasta raccolta di documenti di alta qualità potrebbe fallire se il contesto di tali documenti non è allineato con il fenomeno di interesse. Inoltre, è necessario avere un corpus molto grande per creare un campione rappresentativo e per aumentare le probabilità che una parola compaia in esso.

- *Creazione della DTM:* una volta creato il corpus i documenti. In questo caso non viene applicato lo stemming e l'SVD. Per rappresentare i documenti in uno spazio vettoriale, dobbiamo mappare i termini con degli identificatori. In maniera tale da rappresentare un insieme di documenti come una matrice sparsa, dove ogni riga corrisponde ad un documento e ogni colonna ad un termine. Nel nostro caso abbiamo creato una Document Term Matrix basata su un vocabolario. Facendo altro che catalogare i termini univoci e assegnare un ID univoco.
- *Creazione della Matrice delle similarità:* proseguendo viene calcolata la matrice delle similarità usando la DTM applicando la similarità del coseno. Ugualmente avviene il confronto fra documenti con la somiglianza del coseno. Queste vengono utilizzate per trovare quali vettori sono più simili tra loro e quali documenti hanno una somiglianza superiore a una soglia specificata.

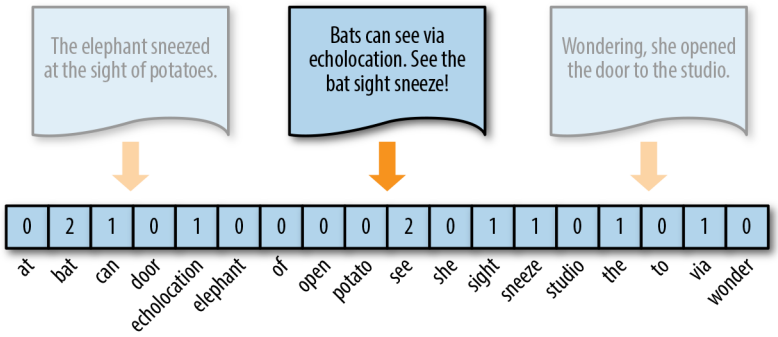

Figura 2.3: Esempio vettorizzazione di un testo.

2.2.11 Clustering

Il clustering [**cluster**] è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e al raggruppamento di elementi omogenei in un insieme di dati. Le tecniche di clustering si basano su misure relative alla somiglianza tra gli elementi. In molti approcci questa similarità, o meglio, dissimilarità, è concepita in termini di distanza in uno spazio multidimensionale.

Gli algoritmi di clustering raggruppano gli elementi sulla base della loro distanza reciproca, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme stesso. Abbiamo assunto per denominare i cluster, una supervisione basata sulla maggioranza dei pazienti malati.

Questo pacchetto mette a disposizione svariate funzioni che permettono di usare in modo semplice ed intuitivo gli algoritmi di clustering; esistono varie classificazioni di quest'ultimi e nel seguente lavoro di tesi sono state prese in considerazione solo due tecniche:

- ***K-means***

L'algoritmo K-means, rientra nelle tecniche di "hard clustering" ed è utilizzato tramite la funzione kmeans().
È un algoritmo di clustering partizionale che permette di suddividere un insieme di oggetti in K gruppi sulla base dei loro attributi, ovvero, questa tecnica partiziona il set di dati in cluster omogenei unici le cui osservazioni sono simili tra loro ma diverse rispetto ad altri cluster. I cluster risultanti rimangono mutualmente esclusivi, cioè cluster non sovrapposti.
In questa tecnica, "K" si riferisce al numero di cluster tra i quali desideriamo suddividere i dati (per cui è supervisionata). Ogni gruppo ha un centroide. Il nome "K-means" deriva dal fatto che i centroidi del cluster sono calcolati come la distanza media delle osservazioni assegnate a ciascun cluster. Questo valore K è dato dall'utente, proprio per questo è una tecnica di clustering molto difficile. L'algoritmo può essere riassunto in cinque passi:

1. Diciamo che il valore di k = 2. Inizialmente, la tecnica di clustering assegnerà due centroidi casualmente nell'insieme di osservazioni.
2. Quindi, inizierà a suddividere le osservazioni in base alla loro distanza dai centroidi. Le osservazioni relativamente più vicine a qualsiasi centroide verranno suddivise di conseguenza.
3. Quindi, in base al numero di iterazioni (dopo quante volte vogliamo che l'algoritmo converga) che abbiamo dato, i centroidi del cluster verranno riavvolti in ogni iterazione. Con questo, l'algoritmo proverà a ottimizzare continuamente per il più basso all'interno della variazione del cluster.
4. In base ai centroidi appena assegnati, si assegna ciascuna osservazione che si avvicina di più ai nuovi centroidi.
5. Questo processo continua fino a quando i centri del cluster non cambiano o viene raggiunto il criterio di arresto.

Graficamente possiamo rappresentare il processo in questo modo:

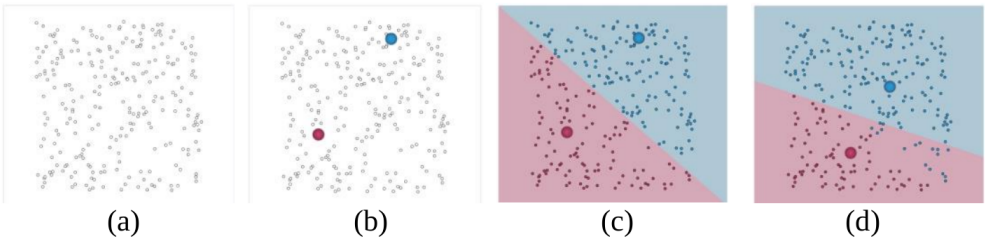

Figura 2.4: Esempio di K-means. Informazioni uniformemente distribuite (a); assegnazione random dei centroidi (k=2) (b); clusterizzazione delle informazioni (c); riassegnazione dei centroidi (d).

Determinare il miglior valore di K gioca un ruolo fondamentale. Per selezionare il miglior valore di K, non esiste "una regola applicate a tutte le situazioni". Dipende dalla forma e dalla distribuzione delle osservazioni in un set di dati. Intuitivamente, trovando il miglior valore di K, cerchiamo di trovare un equilibrio tra il numero di cluster e la variazione media all'interno di un cluster.

Di seguito è riportato il metodi utile per trovare un valore k ottimale:

1. **Convalida incrociata:** è un metodo comunemente usato per determinare il valore di K. Divide i dati in X parti. Quindi, allena il modello sulle parti X-1 e convalida (prova) il modello sulla parte rimanente. Il modello viene convalidato controllando il valore della somma della distanza al quadrato rispetto al centroide. Questo valore finale viene calcolato facendo la media su X cluster. In pratica, per diversi valori di K, eseguiamo la convalida incrociata e quindi scegliamo il valore che restituisce l'errore più basso.

Fuzzy clustering

Il Fuzzy clustering (indicato anche come soft clustering o soft K-means) è utilizzato con la funzione fanny() ed è una forma di clustering in cui ciascun punto di dati può appartenere a più di un cluster. Ogni elemento, pertanto, ha un insieme di coefficienti di appartenenza corrispondenti al grado del legame con un dato cluster; questo valore può variare da 0 a 1.

Ha il vantaggio che non forza ogni oggetto deve essere posizionato in un cluster specifico ma, ha lo svantaggio che c'è molta più informazione da interpretare. L'algoritmo Fuzzy c-means (FCM) è uno degli algoritmi di clustering Fuzzy più diffusi, il centroide di un cluster viene calcolato come media ponderata di tutti i punti, in base al grado di appartenenza al cluster. Questa tecnica si basa su una scelta dell'utente che specifica il numero di cluster da adottare per il set di dati da raggruppare. Anche in questo caso, come per il K-means standard, avviene in modo iterativo e si interrompe quando l'errore è inferiore a un determinato valore di tolleranza o il suo miglioramento rispetto alla precedente iterazione è inferiore a una certa soglia.

Graficamente possiamo trovare dei cluster che si sovrappongono in molti punti come mostrato di seguito:

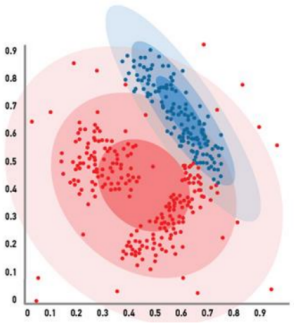

Figura 2.5: Esempio plot Fuzzy clustering.

2.2.12 Pacchetti per il disegno dei grafici

- *ggplot2:* [**ggplot2**] un sistema per la creazione grafica "dichiarativa", basato su "La grammatica della grafica". Fornendo i dati di come mappare le variabili all'estetica, quali primitive grafiche usare, "ggplot2" si prende cura dei dettagli.

- *factoextra:* [**factoextra**] fornisce alcune funzioni facili da usare per estrarre e visualizzare l'output di analisi di dati multivariati, includendo "PCA" (Componente principale di Analisi), "CA" (analisi delle corrispondenze), "MCA" (Analisi di corrispondenza multipla), "FAMD" (Analisi fattoriale di dati misti), "MFA" (Analisi fattoriale multipla) e "HMFA" (Analisi gerarchica fattoriale multipla) e funziona da diversi pacchetti R. Inoltre, fornisce anche funzioni per semplificare alcune fasi di analisi del clustering ed il pacchetto "ggplot2" che si basa su un'elegante rappresentazione dei dati.

Capitolo 3

Definizione del processo di correlazione dei dati

In questo capitolo verrà descritto per intero il processo che permette di trovare la correlazione tra le informazioni sulle visite e lo stato della malattia del paziente che può essere: malato, sano, sano con sintomi tipici della malattia.

3.1 Descrizione del processo LSA

- *Step 1*

  Dopo un'iniziale analisi del dataset, è stata effettuata una scrematura delle tabelle, selezionando solo quelle di interesse per l'analisi sintattica che viene effettuata successivamente. Questa selezione viene fatta solo sulle tabelle in cui compaiono sintomi fortemente correlati con la malattia, escludendo tutte quelle contenenti informazioni diagnostiche.

  Step 2

  In seguito alla fase di scelta e di ricognizione, viene effettuata una modifica delle tabelle, trasformando esclusivamente le colonne relative ai sintomi rilevanti, nelle quali erano presente, in forma numerica, le risposte dei pazienti ai vari questionari o dei medici alle svariate visite mediche.
  Successivamente, vengono del tutto eliminate le colonne che non contenevano nessun tipo di informazione rilevante riguardanti i sintomi come i campi numerici e i campi di testo contenenti sigle che possono variare di tabella in tabella. Infine, dopo la sostituzione è stata effettuata un'ulteriore scrematura eliminando tutte le colonne che davano delle informazioni certe sullo stato diagnostico della malattia.

- *Step 3*

  A questo punto comincia il vero processo di correlazione delle informazioni: per ogni collezione di documenti presa in considerazione si estraggono le informazioni per ogni singolo paziente. Tutte le informazioni estratte sono conservate all'interno di una nuova collezione. Successivamente, si effettua un'iniziale pulitura del testo, attraverso la tecnica del text mining per ricavare delle informazioni di alta qualità dal testo. In seguito, viene applicato l'algoritmo LSA [**lsa**] alla cui base c'è l'idea della co-occorenza dei termini (parole utilizzate nei referti delle visite mediche) su molti documenti (che identificano ogni singolo paziente), evidenziando i termini che sono in qualche modo correlati attraverso una sinonimia o attraverso un concetto latente condiviso. Infine, viene calcolata una matrice delle dissimilarità tra i documenti a partire dalla matrice di co-occorrenze termine/documenti.

  Step 4

  Dopo aver calcolato la dissimilarità fra i vari documenti, si può applicare la tecnica del clustering. [**cluster**] In questo lavoro di tesi si è scelto di utilizzare due tipi di tecniche per confrontare la categorizzazione sul tipo di pazienti: il K-means e il Fuzzy clustering. Per entrambi gli algoritmi bisogna stabilire un numero "K" grazie al quale si stabilisce il numero di cluster in cui vengono divise le informazioni. Abbiamo assunto per denominare i cluster, una supervisione basata sulla maggioranza dei pazienti malati.

- *Step 5*

  Per ultimo, vengono analizzati i dati ottenuti dal clustering, stabilendo le percentuali di appartenenza e di errore attraverso una comparazione con le informazioni certe sullo stato diagnostico della malattia (preventivamente escluse).

3.2 Descrizione del processo text2vec

Il processo si distingue soltanto per alcune differenze nello *Step 3*:

- Non viene applicato lo stemming sul Corpus.
- Non viene applicata l'LSA.

3.3 Descrizione della comparazione tra le tecniche

Per comparare le tecniche è stata effettuata una suddivisione per visite. A partire dalla visita di SC (screening) fino alla visita LOG. I due processi sono stati iterati 19 volte, ogni volta aggiungendo le informazioni su una nuova visita ai corpus di ciascun paziente.

Capitolo 4

Gli script per l'elaborazione dei dati

In questo capitolo verranno descritti tutti i principali passaggi, associati a opportune righe di codice che hanno permesso l'elaborazione del dataset e l'applicazione di tutti i pacchetti utilizzati nell'ambiente integrato R.

4.1 Progettazione dataset

Nella fase di progettazione, come citato nel capitolo 3, è stata effettuata una scrematura delle tabelle selezionandone soltanto 34 su 131, e la loro modifica per poter utilizzare l'algoritmo LSA sulle stesse, in modo tale da avere dei blocchi di testi sul quale quest'ultimo può processare.

Le tabelle selezionate sono:

- Adverse_Event_Log
- Clinical_Diagnosis_and_Management
- Cognitive_Categorization
- Concomitant_Medications
- Current_Medical_Conditions_Log
- DaTscan_Imaging
- Diagnostic_Features
- Epworth_Sleepiness_Scale
- Family_History___PD
- Features_of_REM_Behavior_Disorder

- General_Medical_History
- General_Neurological_Exam
- General_Physical_Exam
- Genetic_Testing_Results
- Geriatric_Depression_Scale___Short
- Inclusion_Exclusion
- iPSC_Blood_Sample
- Magnetic_Resonance_Imaging
- MDS_UPDRS_Part_I
- MDS_UPDRS_Part_I___Patient_Questionnaire
- MDS_UPDRS_Part_II___Patient_Questionnaire
- MDS_UPDRS_Part_III
- MDS_UPDRS_Part_IV
- Modified_Schwab_+_England_ADL
- Neurological_Exam_-_Cranial_Nerves
- PD_Features
- Prodromal_Diagnostic_Questionnaire
- QUIP_Current_Short
- REM_Sleep_Disorder_Questionnaire
- SCOPA-AUT
- Screening_____Demographics
- Socio-Economics
- University_of_Pennsylvania_Smell_ID_Test
- Use_of_PD_Medication

Successivamente, da queste 34 ne sono state eliminate 3 poiché ci si è resi conto che contenevano delle colonne con informazioni non molto rilevanti e significative. Le tabelle eliminate sono:

- Features_of_REM_Behavior_Disorder
- Genetic_Testing_Results
- Inclusion_Exclusion

Infine, è stata effettuata un'ulteriore scrematura, eliminando ben 11 tabelle che contenevano ulteriori informazioni diagnostiche.

Le tabelle finali che hanno costituito l'input per questo lavoro di tesi sono:

- Adverse_Event_Log
- Clinical_Diagnosis_and_Management
- Concomitant_Medications
- Current_Medical_Conditions_Log
- DaTscan_Imaging
- Diagnostic_Features
- Epworth_Sleepiness_Scale
- Family_History_PD
- General_Medical_History
- General_Neurological_Exam
- General_Physical_Exam
- Geriatric_Depression_Scale_Short
- iPSC_Blood_Sample
- MDS_UPDRS_Part_I
- MDS_UPDRS_Part_I_Patient_Questionnaire
- MDS_UPDRS_Part_II_Patient_Questionnaire
- MDS_UPDRS_Part_III
- MDS_UPDRS_Part_IV
- Modified_Schwab_England_ADL
- REM_Sleep_Disorder_Questionnaire

4.2 Importazione del dataset su Excel

Il dataset si presenta come un insieme di file CSV. Per la creazione del dataset viene utilizzata la seguente Macro VBA, con la quale vengono caricati i vari file all'interno della nostra cartella di lavoro:

```
Sub CombineCsvFiles()
    Dim xFilesToOpen As Variant
    Dim I As Integer
    Dim xWb As Workbook
    Dim xTempWb As Workbook
    Dim xDelimiter As String
    Dim xScreen As Boolean
    On Error GoTo ErrHandler
    xScreen = Application.ScreenUpdating
    Application.ScreenUpdating = False
    xDelimiter = ";"
    xFilesToOpen = Application.GetOpenFilename("Text Files (*.csv), *.csv", , "", , True)
    If TypeName(xFilesToOpen) = "Boolean" Then
        MsgBox "No files were selected", , ""
        GoTo ExitHandler
    End If
    I = 1
    Set xTempWb = Workbooks.Open(xFilesToOpen(I))
    xTempWb.Sheets(1).Copy
    Set xWb = Application.ActiveWorkbook
    xTempWb.Close False
    Do While I < UBound(xFilesToOpen)
        I = I + 1
        Set xTempWb = Workbooks.Open(xFilesToOpen(I))
        xTempWb.Sheets(1).Move , xWb.Sheets(xWb.Sheets.Count)
    Loop
ExitHandler:
    Application.ScreenUpdating = xScreen
    Set xWb = Nothing
    Set xTempWb = Nothing
    Exit Sub
ErrHandler:
    MsgBox Err.Description, , ""
    Resume ExitHandler
End Sub
```

Figura 4.1: Istruzione per l'importazione del file csv.

Successivamente, avviene l'estrazione da Excel delle tabelle selezionate nella fase di ricognizione, eliminando tutte le colonne non rilevanti eccetto:

- **PATNO:** campo numerico non univoco, che identifica il paziente.
- **EVENT_ID:** campo che identifica il tipo di visita a cui viene sottoposto il paziente.
- Tutte le colonne contenenti **informazioni diagnostiche**.

Ed Infine, vengono convertiti i valori numerici dei questionari in frasi di senso compiuto. Viene costruito, quindi, un dataset ri-elaborato.

4.3 Creazione corpus in R

A questo punto il dataset ri-elaborato viene importato e trasformato in un Corpus in R. Le informazioni verranno raggruppate in base al valore del PATNO preso dalla tabella "Patient Status", ogni documento sarà nominato con uno specifico PATNO, al cui interno troveremo tutto il testo, situato nelle varie tabelle, relativo a quest'ultimo. Per il dataset PPMI, verrà creato un Corpus contenente 2169 documenti, uno per ogni paziente identificato. Di seguito, verranno riportati due brevi pezzi di codice utilizzati per creare i vari file di testo relativi ad ogni singolo paziente ed il Corpus:

```
if(bcreatefiles) {
  cat("Create ",len," output text files in: '",corpusdir,"'\n", sep="", file = logfile)
  for(i_p in patno) {
    patnoFile <- file(file.path(corpusdir, paste(i_p,".txt",sep="")), "w")
    close(patnoFile)
  }

  count <- 0
  for(i_p in patno) {
    count <- count + 1
    cat(count,"/",len,"\n",sep="")
    for(j_t in tables) {
      #cat("Analyze patient ",i_p," in file '",j_t,"'\n", sep="", file = logfile)
      datas <- read.csv(j_t, sep = ";")
      idatas <- which(datas$PATNO == i_p)
      subidatas <- datas[idatas,]
      patnoFile <- file(file.path(corpusdir, paste(i_p,".txt",sep="")), "a")
      cat(as.matrix(subidatas)," ",sep=" ", file = patnoFile)
      close(patnoFile)
    }
  }
}

corpusdirST <- file.path(currentdir,"CorpusST")
```

Figura 4.2: Creazione file txt per ogni PATNO.

4.4 Pre-processing

Prima di poter applicare il pacchetto lsa, è opportuno effettuare una ripulitura sul testo.

Questo pacchetto, oltre a permettere la creazione del Corpus spiegato nel paragrafo precedente, mette a disposizione la funzione *tm_map()* tramite la quale possiamo applicare tutte le azioni per effettuare un'ottima ripulitura del testo.

Di seguito verranno riportare le linee di codice usate per eliminare tutti i caratteri speciali, gli spazi bianchi, le stopwords (parole comunemente usate che non danno nessun significato al testo), la punteggiatura, i numeri ed infine per poter effettuare lo stemming del testo:

```
tokenize <- function(x) gsub("([_-])", " ", x)
docs_corpus <- tm_map(docs_corpus, tokenize)
docs_corpus <- tm_map(docs_corpus, tolower)
docs_corpus <- tm_map(docs_corpus, removeWords, c("parkinsonian","parkinsonism","parkinson","pd","parkinsan",
                                                  "parkingsons","parkisons","parkinons","parkinsonä½","gparkinson",
                                                  "parkisonism","parkinsinism","parkisnons","parksinsons","parkinsins",
                                                  "parksinons","carbidopalevoparkinsona","parkinosns","parkinsns",
                                                  "parkinspons","parkinssons","parkisnonism","parkisonsons","parknsons",
                                                  "steadyparkinsonirsadapine","parkinsonä½","gparkinson",
                                                  "carbidopalevoparkinsona","steadyparkinsonirsadapine","prkinson"))

docs_corpus <- tm_map(docs_corpus, removePunctuation)
docs_corpus <- tm_map(docs_corpus, removeNumbers)
docs_corpus <- tm_map(docs_corpus, stripWhitespace)
docs_corpus <- tm_map(docs_corpus, PlainTextDocument)

removeMinWordLength <- function(x) gsub("\\b[[:alpha:]]{1,3}\\b", "", x, perl=T)
docs_corpus <- tm_map(docs_corpus, removeMinWordLength)
docs_corpus <- tm_map(docs_corpus, removeWords, stopwords("english"))
docs_corpus <- tm_map(docs_corpus, removeWords, c("log"))
docs_corpus <- tm_map(docs_corpus, stripWhitespace)
docs_corpus <- tm_map(docs_corpus, stemDocument)
docs_corpus <- tm_map(docs_corpus, PlainTextDocument)
```

Figura 4.3: Ripulitura del testo con il pacchetto "tm".

È bene notare che per text2vec viene effettuata la stessa procedura eccetto lo stemming (*stemDocument*).

4.5 Applicazione LSA

Dopo aver effettuato ripulitura e stemming del testo, possiamo applicare l'algoritmo LSA. Inizialmente si crea una matrice termine-documento, ovvero un oggetto della classe *"text-matrix"* a cui viene passato il percorso dove è stato salvato, precedentemente, il corpus con le modifiche applicate tramite il pacchetto "tm".

Successivamente viene effettuata una normalizzazione sulla matrice, ottenuta moltiplicando una funzione locale *lw_bintf()* e una funzione globale *gw_idf()*.

A questo punto possiamo applicare l'algoritmo, con la funzione *lsa()* creando, così, lo spazio semantico.

```
myMatrix <- textmatrix("C://Users/ospite/Desktop/Nuova cartella/CorpusST")
save(myMatrix, file="textMatrix.RData")

myMatrix1 <- lw_bintf(myMatrix) * gw_idf(myMatrix)
save(myMatrix1, file="textMatrixN.RData")

myLSAspace <- lsa(myMatrix1, dims=dimcalc_share())
save(myLSAspace, file="myLSAspace.RData")
```

Figura 4.4: Algoritmo LSA

Infine viene calcolata la matrice delle similarità sottraendo 1 alla funzione cosine(). Quindi si avrà una similarità pari a 0 tra documenti simili e viceversa ad 1 per documenti totalmente dissimili. Nel caso in cui nella matrice delle similarità sia presente un valore negativo (non ammissibile come similarità), tale valore viene posto a 0.

```
distMatrix <- 1 - cosine(as.textmatrix(myLSAspace))
distMatrix[which(distMatrix < 0)] <- 0
save(distMatrix, file="distMatrix.RData")
```

Figura 4.5: Calcolo della matrice delle similarità.

4.6 Applicazione text2vec

Partendo dalla ripulitura del testo, come in precedenza, creiamo una matrice delle similarità, utilizzando questa volta il Corpus senza le modifiche avvenute col processo di stemming.

I testi possono occupare molta memoria centrale. [**text2vec_vettorizzazione**] Bisogna leggere l'intera collezione di documenti nella RAM e processarli come un singolo vettore. text2vec risolve il problema fornendo un modo migliore per costruire una matrice termine documento.
Per rappresentare i documenti in uno spazio vettoriale, dobbiamo mappare i termini con degli identificatori. In maniera tale da rappresentare un insieme di documenti come una matrice sparsa, dove ogni riga corrisponde ad un documento e ogni colonna ad un termine. Nel nostro caso abbiamo creato una Document Term Matrix basata su un vocabolario. Facendo altro che catalogare i termini univoci e assegnare un ID univoco utilizzando la funzione *create_vocabulary()*.
Abbiamo creato un iteratore *itoken()* che itera tutti i token, (o termini) e costruito il vocabolario con *create_vocabulary()*

```
it1 = itoken(pazienti$review, progressbar = FALSE)
it = itoken(pazienti$review, progressbar = FALSE)
v = create_vocabulary(it) %>% prune_vocabulary(doc_proportion_max = 0.1, term_count_min = 5)
vectorizer = vocab_vectorizer(v)
```

Figura 4.6: Creazione del vettore text2vec.

A seguire creiamo la Document Term Matrix.

```
dtm1 = create_dtm(it1, vectorizer)
dim(dtm1)
```

Figura 4.7: Calcolo della Document Term Matrix.

Infine calcoliamo la matrice delle similarità tra i pazienti. Utilizzando come similarità il coseno e normalizziamo.

```
dist_matrix = 1-sim2(x = dtm1, method = "cosine", norm = "l2")
rownames(dist_matrix)<-file_list
colnames(dist_matrix)<-file_list

distMatrix<- as.data.frame(apply(d1_d2_tfidf_cos_sim, 2, function(x) (x - min(x))/(max(x)-min(x))))
save(distMatrix,file = "../distMatrixDoc2Vec.RData")
```

Figura 4.8: Calcolo della Matrice delle similarità tra i pazienti.

4.7 Applicazione clustering

Dopo aver applicato l'algoritmo lsa e in un secondo momento text2vec, sulla matrice delle similarità possiamo applicare il clustering. Come precisato utilizziamo due tipi di tecniche per confrontare la categorizzazione sul tipo di pazienti: il K-means e il Fuzzy clustering.

Per entrambi gli algoritmi bisogna stabilire un numero "K" grazie al quale si stabilisce il numero di cluster in cui vengono divise le informazioni. Per ottenere una netta distinzione tra i pazienti malati e non malati la scelta del "K" è ricaduta sul numero 2.

4.7.1 K-means

Il pacchetto cluster, spiegato nel capitolo due, mette a disposizione la funzione K-means che di seguito sarà riportata in funzione del contesto di questo lavoro di tesi:

```
k<- 2
cluster.kmeans<-kmeans(as.matrix(distMatrix),k)
```

Figura 4.9: Funzione K-means.

In seguito, i pazienti sono stati categorizzati con delle etichette, in modo da avere una netta distinzione tra quelli malati e non malati, con una supervisione basata sulla maggioranza. Inoltre è stato effettuata una verifica sulle visite, per controllare se un paziente con un numero maggiore di visite fosse più informativo rispetto ad un altro che ne possiede di meno, dal risultato possiamo affermare che tutto ciò è ininfluente.

Con la funzione *fviz_cluster()* otteniamo graficamente il clustering ottenuto dal K-means:

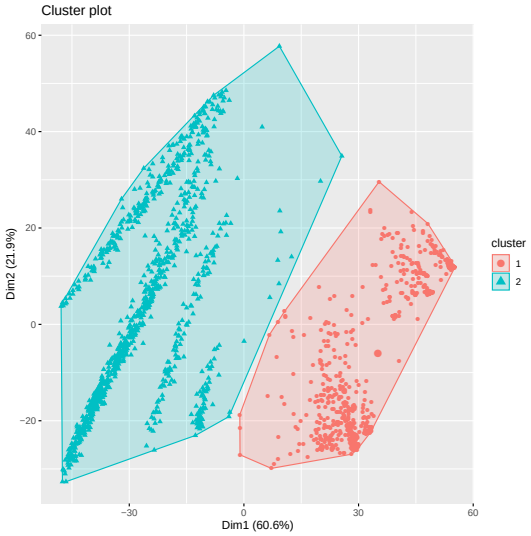

Figura 4.10: Funzione plot K-means per LSA nella visita di Screening (SC).

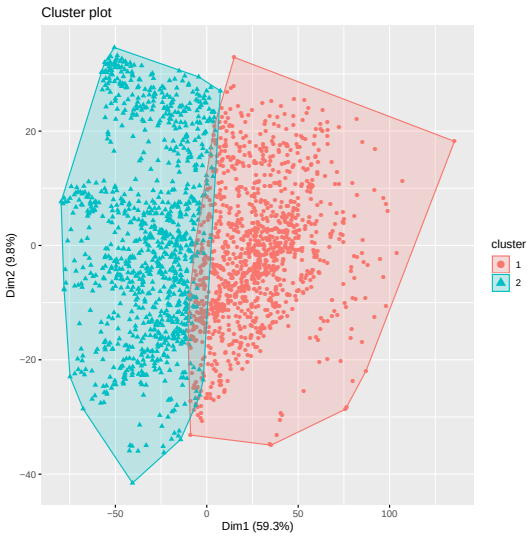

Figura 4.11: Plot K-means text2vec visite SC-LOG.

Dal seguente grafico possiamo evincere come ci sia una netta distinzione fra i due cluster e che solo alcuni, pochi, punti si sovrappongono.

4.7.2 Fuzzy clustering

La seconda tecnica utilizzata è la tecnica del Fuzzy clustering, applicata grazie alla funzione *fanny()* e di seguito sarà riportato il codice utilizzato in questa tesi:

```
K <- 2
fcluster <- fanny(distMatrix, k = K , maxit = 1000,  diss = TRUE, memb.exp = 1.01, keep.diss = TRUE,tol = 1e-20)
```

Figura 4.12: Funzione K-means.

In seguito, come per il K-means, i pazienti sono stati categorizzati con delle etichette, in modo da avere una netta distinzione tra quelli malati e non malati ed inoltre sono stati eliminati tutti gli elementi che avessero una percentuale di appartenenza a un cluster inferiore a una determinata soglia. In questo caso la soglia è rappresentata dalla variabile "th" che assume valore k/2.

Con la funzione *clusplot()* otteniamo graficamente il clustering ottenuto dalla funzione *fanny()*:

```
# plot
clusplot(distMatrix, fcluster$clustering, color=T, shade=T, labels=4, lines=0)
```

Figura 4.13: Funzione clusplot.

Dal seguente grafico possiamo evincere come, in questo caso, non ci sia una netta distinzione fra i due cluster, nei quali si sovrappongono molti punti.

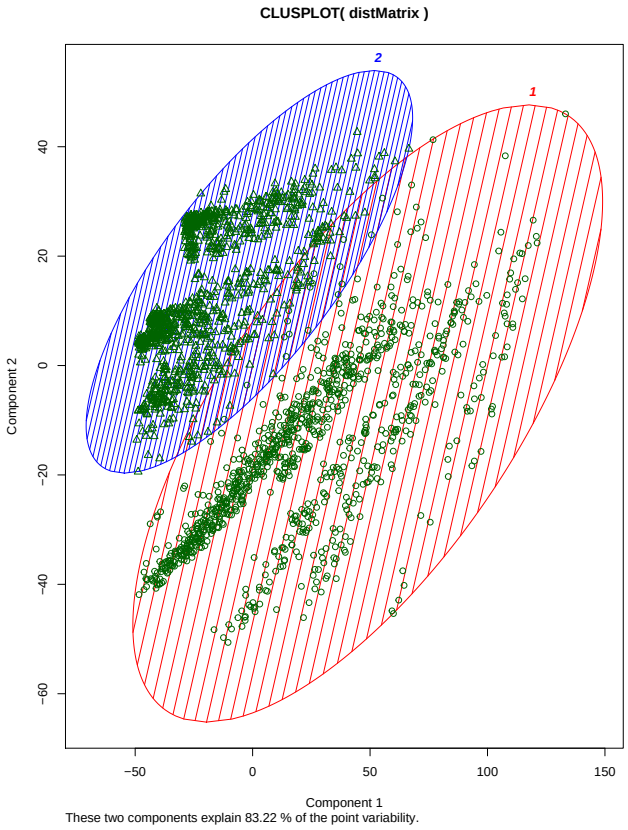

Figura 4.14: Plot del Fuzzy clustering per LSA nelle visite SC-LOG.

Infine, viene applicata una tecnica di ripulitura basata sulle seguenti linee di codice:

Tutti le osservazioni che hanno probabilità minore di 1/K vengono rimosse, per questo motivo i cluster si ripuliscono di valori spuri e convergono su una buona classificazione di "malato" "non malato".

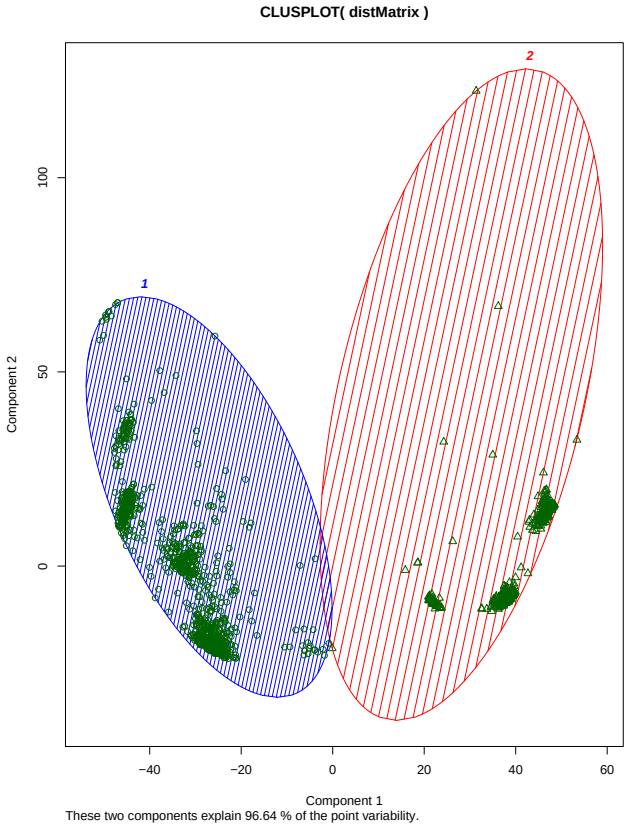

Figura 4.15: Plot del Fuzzy clustering text2vec nella visita di Screening (SC).

```
th <- 1/K;
cls <- vector("list",K)
varcls <- vector("list",k)

for(i in 1:K)
{
  cat("cluster all:",i,length(fcluster$membership[,i]),"\n", sep=" ")
  cls[[i]] <-  as.vector(which(fcluster$membership[,i] > th))
}
```

Figura 4.16: Funzione clusplot.

4.7.3 Analisi dei dati

Per comprendere i risultati impieghiamo queste metriche ampiamente diffuse, ***Precision*** e ***recall*** sono due comuni classificazioni statistiche, utilizzate in diversi ambiti del sapere, come per es. l'information retrieval.
La precision può essere vista come una misura di esattezza o fedeltà, mentre il recall è una misura di completezza.

$$\text{Precision} = \frac{veropositivo}{veropositivo + falsopositivo}$$

$$\text{Recall} = \frac{veropositivo}{veropositivo + falsonegativo}$$

Nell'analisi statistica della classificazione binaria, l'***F1 score*** (nota anche come F-score o F-measure) è una misura dell'accuratezza di un test. La misura tiene in considerazione precision e recall del test.

L'F-score è solitamente usata nel campo del recupero dell'informazione per misurare l'accuratezza delle ricerche o della classificazione dei documenti.
Inoltre viene utilizzata perchè precision e recall non sono equiparabili fra loro

$$F_1 = \frac{2}{\frac{1}{\mathrm{r}} + \frac{1}{\mathrm{p}}} = 2 \cdot \frac{\mathrm{p} \cdot \mathrm{r}}{\mathrm{p} + \mathrm{r}}.$$

Figura 4.17: Formula F-measure.

Di seguito vengono riportati i valori dei due cluster utilizzando l'algoritmo K-means:

Visits	K-means												Patients		
	cluster PD			cluster GP			Patients			Precision	Recall	F-measure			
	PD	GP	Total	PD	GP	Total	Total	PD	GP				Total	PD	GP
SC	761	240	1001	282	439	721	1722	1043	679	0,76	0,73	0,74	1722	1043	679
SC-BL	747	248	995	495	679	1174	2169	1242	927	0,75	0,60	0,67	2169	1242	927
SC-V01	742	248	990	500	679	1179	2169	1242	927	0,75	0,60	0,66	2169	1242	927
SC-V02	747	263	1010	495	664	1159	2169	1242	927	0,74	0,60	0,66	2169	1242	927
SC-V03	744	263	1007	498	664	1162	2169	1242	927	0,74	0,60	0,66	2169	1242	927
SC-V04	626	263	889	616	664	1280	2169	1242	927	0,70	0,50	0,59	2169	1242	927
SC-V05	629	275	904	613	652	1265	2169	1242	927	0,70	0,51	0,59	2169	1242	927
SC-V06	590	268	858	652	659	1311	2169	1242	927	0,69	0,48	0,56	2169	1242	927
SC-V07	590	268	858	652	659	1311	2169	1242	927	0,69	0,48	0,56	2169	1242	927
SC-V08	586	254	840	656	673	1329	2169	1242	927	0,70	0,47	0,56	2169	1242	927
SC-V09	591	255	846	651	672	1323	2169	1242	927	0,70	0,48	0,57	2169	1242	927
SC-V10	588	240	828	654	687	1341	2169	1242	927	0,71	0,47	0,57	2169	1242	927
SC-V11	596	243	839	646	684	1330	2169	1242	927	0,71	0,48	0,57	2169	1242	927
SC-V12	600	239	839	642	688	1330	2169	1242	927	0,72	0,48	0,58	2169	1242	927
SC-V13	599	238	837	643	689	1332	2169	1242	927	0,72	0,48	0,58	2169	1242	927
SC-V14	606	225	831	636	702	1338	2169	1242	927	0,73	0,49	0,58	2169	1242	927
SC-V15	608	225	833	634	702	1336	2169	1242	927	0,73	0,49	0,59	2169	1242	927
SC-ST	610	225	835	632	702	1334	2169	1242	927	0,73	0,49	0,59	2169	1242	927
SC-LOG	729	419	1148	513	508	1021	2169	1242	927	0,64	0,59	0,61	2169	1242	927

Figura 4.18: Processo K-means LSA.

Visits	K-means												Patients		
	cluster PD			cluster GP			Patients			Precision	Recall	F-measure			
	PD	GP	Total	PD	GP	Total	Total	PD	GP				Total	PD	GP
SC	698	251	949	345	428	773	1722	1043	679	0,74	0,67	0,70	1722	1043	679
SC-BL	690	257	947	552	670	1222	2169	1242	927	0,73	0,56	0,63	2169	1242	927
SC-V01	677	255	932	565	672	1237	2169	1242	927	0,73	0,55	0,62	2169	1242	927
SC-V02	652	255	907	590	672	1262	2169	1242	927	0,72	0,52	0,61	2169	1242	927
SC-V03	642	254	896	600	673	1273	2169	1242	927	0,72	0,52	0,60	2169	1242	927
SC-V04	1053	825	1878	189	102	291	2169	1242	927	0,56	0,85	0,68	2169	1242	927
SC-V05	1066	825	1891	176	102	278	2169	1242	927	0,56	0,86	0,68	2169	1242	927
SC-V06	1077	873	1950	165	54	219	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V07	1077	873	1950	165	54	219	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V08	1077	876	1953	165	51	216	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V09	1077	876	1953	165	51	216	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V10	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V11	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V12	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V13	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V14	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-V15	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-ST	1077	877	1954	165	50	215	2169	1242	927	0,55	0,87	0,67	2169	1242	927
SC-LOG	1123	879	2002	119	48	167	2169	1242	927	0,56	0,90	0,69	2169	1242	927

Figura 4.19: Processo K-means text2vec.

per quanto riguarda il clustering effettuato con la tecnica del K-means, si è potuto evincere come ci sia una netta distinzione tra i due cluster, portando, però, a una percentuale di errore molto alta. Questo è dovuto alla struttura intrinseca dell'algoritmo che non permette ad un elemento di poter essere posizionato in entrambi i cluster o di poter essere spostato successivamente da un cluster a un altro, costringendo a un'errata classificazione delle informazioni trattate.

Di seguito vengono riportati i valori dei due cluster utilizzando l'algoritmo Fuzzy c-means:

Fuzzy c-means													Patients		
cluster PD			cluster GP			Patients			Precision	Recall	F-measure				
PD	GP	Total	PD	GP	Total	Total	PD	GP				Total	PD	GP
757	239	996	286	440	726	1722	1043	679	0,76	0,73	0,74	1722	1043	679
722	242	964	520	685	1205	2169	1242	927	0,75	0,58	0,65	2169	1242	927
718	241	959	524	686	1210	2169	1242	927	0,75	0,58	0,65	2169	1242	927
718	241	959	524	686	1210	2169	1242	927	0,75	0,58	0,65	2169	1242	927
720	241	961	522	686	1208	2169	1242	927	0,75	0,58	0,65	2169	1242	927
752	248	1000	490	679	1169	2169	1242	927	0,75	0,61	0,67	2169	1242	927
752	248	1000	490	679	1169	2169	1242	927	0,75	0,61	0,67	2169	1242	927
763	250	1013	479	677	1156	2169	1242	927	0,75	0,61	0,68	2169	1242	927
765	250	1015	477	677	1154	2169	1242	927	0,75	0,62	0,68	2169	1242	927
766	250	1016	476	677	1153	2169	1242	927	0,75	0,62	0,68	2169	1242	927
766	250	1016	476	677	1153	2169	1242	927	0,75	0,62	0,68	2169	1242	927
766	250	1016	476	677	1153	2169	1242	927	0,75	0,62	0,68	2169	1242	927
766	250	1016	476	677	1153	2169	1242	927	0,75	0,62	0,68	2169	1242	927
766	251	1017	476	676	1152	2169	1242	927	0,75	0,62	0,68	2169	1242	927
768	252	1020	474	675	1149	2169	1242	927	0,75	0,62	0,68	2169	1242	927
771	252	1023	471	675	1146	2169	1242	927	0,75	0,62	0,68	2169	1242	927
771	252	1023	471	675	1146	2169	1242	927	0,75	0,62	0,68	2169	1242	927
771	252	1023	471	675	1146	2169	1242	927	0,75	0,62	0,68	2169	1242	927
948	508	1456	294	419	713	2169	1242	927	0,65	0,76	0,70	2169	1242	927

Figura 4.20: Processo Fuzzy c-means LSA.

Fuzzy c-means													Patients		
cluster PD			cluster GP			Patients			Precision	Recall	F-measure				
PD	GP	Total	PD	GP	Total	Total	PD	GP				Total	PD	GP
697	251	948	346	428	774	1722	1043	679	0,74	0,67	0,70	1722	1043	679
701	257	958	541	670	1211	2169	1242	927	0,73	0,56	0,64	2169	1242	927
695	259	954	547	668	1215	2169	1242	927	0,73	0,56	0,63	2169	1242	927
681	256	937	561	671	1232	2169	1242	927	0,73	0,55	0,63	2169	1242	927
668	256	924	574	671	1245	2169	1242	927	0,72	0,54	0,62	2169	1242	927
712	251	963	530	676	1206	2169	1242	927	0,74	0,57	0,65	2169	1242	927
712	248	960	530	679	1209	2169	1242	927	0,74	0,57	0,65	2169	1242	927
706	242	948	536	685	1221	2169	1242	927	0,74	0,57	0,64	2169	1242	927
707	242	949	535	685	1220	2169	1242	927	0,74	0,57	0,65	2169	1242	927
704	228	932	538	699	1237	2169	1242	927	0,76	0,57	0,65	2169	1242	927
709	228	937	533	699	1232	2169	1242	927	0,76	0,57	0,65	2169	1242	927
704	224	928	538	703	1241	2169	1242	927	0,76	0,57	0,65	2169	1242	927
707	221	928	535	706	1241	2169	1242	927	0,76	0,57	0,65	2169	1242	927
695	211	906	547	716	1263	2169	1242	927	0,77	0,56	0,65	2169	1242	927
674	208	882	568	719	1287	2169	1242	927	0,76	0,54	0,63	2169	1242	927
1059	868	1927	183	59	242	2169	1242	927	0,55	0,85	0,67	2169	1242	927
1059	868	1927	183	59	242	2169	1242	927	0,55	0,85	0,67	2169	1242	927
1059	868	1927	183	59	242	2169	1242	927	0,55	0,85	0,67	2169	1242	927
674	209	883	568	718	1286	2169	1242	927	0,76	0,54	0,63	2169	1242	927

Figura 4.21: Processo Fuzzy c-means text2vec.

con la tecnica del Fuzzy clustering si è notata come la distinzione non sia più così netta, ma molti elementi che appartengo al cluster "non malato" appartengono, contemporaneamente, al cluster "malato", pertanto, come si nota dalla figura 4.16, in buona parte i due cluster si sovrappongono. Ciò è reso possibile grazie al principio di fondo sul quale si basa questa tecnica: un elemento può appartenere ad entrambi i cluster senza restrizioni e con percentuali di appartenenza differenti.

Il grafico in figura 4.22 ci fornisce una chiara comparazione:

- Durante la visita SC (screening) la tecnica LSA esibisce i risultati migliori accoppiata ad entrambi gli algoritmi di clustering.
- La situazione si sconvolge nel gruppo di visite SC-V03 in quanto LSA con K-means diminuisce ampiamente e al contempo text2vec con K-means aumenta sensibilmente, text2vec con Fuzzy discretamente.
- I processi con i risultati maggiormente accurati nel gruppo di visite SC-LOG sono LSA con Fuzzy e text2Vec con K-means. text2vec con Fuzzy è calato notevolmente, mentre LSA con K-means ha recuperato leggermente ma con molto divario rispetto alle prime due.

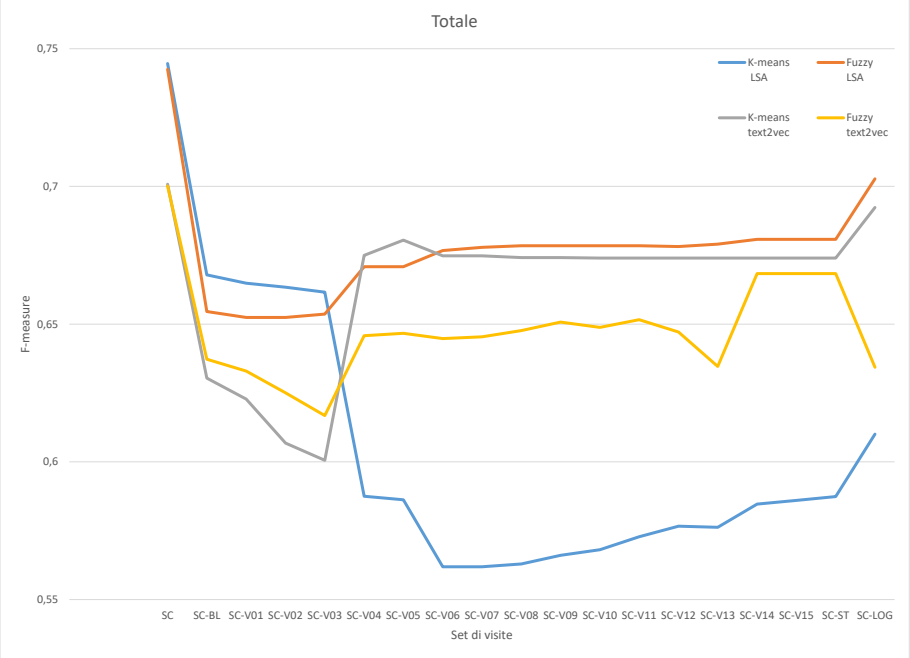

Figura 4.22: Confronto LSA text2vec nei set di visite.

Capitolo 5

Conclusioni

In questo lavoro di tesi è stato evidenziato come la tecnologia può influire e migliorare la ricerca, aiutando ad evidenziare ed a correggere errori laddove ce ne fossero e producendo numerosi risultati significativi.

Durante la visita SC (screening) la tecnica LSA esibisce i risultati migliori accoppiata ad entrambi gli algoritmi di clustering.
La situazione si sconvolge nel gruppo di visite SC-V03 in quanto LSA con K-means diminuisce ampiamente e al contempo text2vec con K-means aumenta sensibilmente, text2vec con Fuzzy discretamente.
I processi con i risultati maggiormente accurati nel gruppo di visite SC-LOG sono LSA con Fuzzy e text2Vec con K-means. text2vec con Fuzzy è calato notevolmente, mentre LSA con K-means ha recuperato leggermente ma con molto divario rispetto alle prime due.

In conclusione entrambi i processi hanno avuto buoni risultati, ma ciò dipende dal tipo di tecnica utilizzata. Nel nostro caso text2vec con K-means e LSA con Fuzzy, ci hanno offerto i migliori risultati.

In futuro l'analisi dei dati e la correlazione tra essi potrà essere ulteriormente estesa. In particolare, i possibili sviluppi possono essere:

- le descrizioni dei sintomi potrebbero essere standardizzate in modo migliore;
- l'analisi ed il processo di elaborazione potrebbe essere esteso su più tabelle in modo da avere delle analisi più precise e dettagliate in maniera da accettare o rifiutare i risultati ottenuti nel lavoro di tesi;
- Addestrare in maniera più diligente la rete neurale text2vec utilizzando vocaboli medici.