REPORT DATA MINING

Glasgow Norms Dataset

GIOVANNI MANGANO

A.A. 2021/2022

Sommario

Introduzione

Il presente report si occuperà di analizzare il dataset *Glasgow Norms*[1], frutto di un esperimento di psicolinguistica che si è occupato di analizzare le parole polisemiche, ovvero parole che pur essendo omografe sono caratterizzate da significati differenti, descrivendole per mezzo di nove diversi attributi. In questo progetto si cercherà di analizzare e comprendere se la polisemia possa essere riconosciuta in maniera automatica per mezzo di algoritmi di clustering, classificazione e pattern mining e quali sono le caratteristiche che rendono tali parole riconoscibili.

La prima parte del report sarà quindi destinata ad una prima fase esplorativa del dataset, in cui si cercherà di comprendere la distribuzione delle diverse *feature* e l'eventuale presenza o meno di valori mancanti e/o *outlier,* in modo da procedere più agevolmente nelle fasi successive. Si procederà quindi alla preparazione dei dati per la successiva applicazione degli algoritmi di *machine learning* tramite l'eliminazione e la normalizzazione delle feature.

A questo seguirà una fase di clustering, in cui si vorranno individuare automaticamente *gruppi* di parole simili, per verificare se le parole polisemiche sono effettivamente differenti – e per questo più facilmente riconoscibili – dalle parole che non possiedono questa caratteristica. Successivamente ci si sposterà sulla classificazione, e quindi sulla creazione di modelli che riconoscano automaticamente la polisemia; infine, si sfrutterà il *pattern mining* per l'estrazione di regole che descrivano tale caratteristica.

Si trarranno infine le conclusioni sul lavoro svolto.

Data understanding

Il dataset *Glasgow Norms* si compone di 4682 record e 12 features, di cui 9 risultato dell'esperimento psicolinguistico già citato nell'Introduzione.

Feature	Tipo attributo	Descrizione
Word	Categorico	Lemma (in inglese)
Length	Discreto	Numero di caratteri del lemma
Arousal	Continuo	Agitazione vs calma (in scala da 1 a 9)
Valence	Continuo	Valore positivo o negativo associato alla parola (in scala da 1 a 9)
Dominance	Continuo	Controllo delle emozioni (in scala da 1 a 9)
Concreteness	Continuo	Concreto vs astratto (in scala da 1 a 7)
Imageability	Continuo	Facilità o difficoltà nell'immaginare un lemma (in scala da 1 a 7)
Familiarity	Continuo	Misura della familiarità di un concetto espresso da un lemma (in scala da 1 a 7)
Aoa	Continuo	Stima dell'età di acquisizione di una parola.
Semsize	Continuo	Misura delle dimensioni o estensioni del concetto evocato dal lemma (in scala da 1 a 7)
Gender	Continuo	Associazione del lemma a un concetto maschile o femminile
Polysemy	Binario (0,1)	Definisce se il lemma è polisemico (1) o meno (0)
Web_corpus_freq	Continuo	Frequenza del lemma nel *corpus Google Newspapers Corpus*

Tabella 1: Descrizione delle feature del dataset

[1] https://link.springer.com/article/10.3758/s13428-018-1099-3

Di queste features è particolare *aoa (Age of Acquisition)*, in quanto i valori contenuti in essa corrispondono alla serie di coppie di anni compresi tra i 0 e i 12 (es. 0-2 anni: 1 nel dataset), mentre il 7 della scala corrisponde a un'età superiore ai 13 anni inclusi.

Dopo aver visualizzato il dataset e aver individuato la *variabile target*, ovvero la polisemia, si è proceduto a verificare se fossero presenti dei valori mancanti – anche se poi ci si ritornerà nella fase di data preparation – e si è notato che il dataset è abbastanza completo, in quanto risultano mancanti solamente 14 valori nella varabile *web_corpus_freq*.

Si è quindi proceduto a visualizzare alcune statistiche di base sui diversi attributi: il valore medio, la deviazione standard, il minimo, il primo, secondo, terzo quartile e il valore massimo. Le parole inglesi nel dataset hanno una lunghezza media di 6 caratteri, mentre il resto delle *feature* psicolinguistiche tende ad avere valori compresi tra 4 e 5.

	length	arousal	valence	dominance	concreteness	imageability	familiarity	aoa	semsize	gender	polysemy	web_corpus_freq
count	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4682.000000	4.668000e+03
mean	6.348355	4.678129	5.086797	5.044939	4.566273	4.723018	5.271335	4.143427	4.136403	4.099933	0.080948	2.988976e+07
std	2.006230	1.097163	1.594344	0.930669	1.433689	1.363110	0.921218	1.252770	1.023293	0.912293	0.272785	8.490144e+07
min	2.000000	2.057000	1.030000	1.941000	1.636000	1.737000	1.647000	1.219000	1.375000	1.000000	0.000000	1.277000e+04
25%	5.000000	3.849000	4.115000	4.529000	3.242000	3.519250	4.706000	3.114000	3.438000	3.606000	0.000000	1.671100e+06
50%	6.000000	4.571000	5.290000	5.123000	4.471000	4.677000	5.438000	4.177000	4.186500	4.121000	0.000000	5.702982e+06
75%	8.000000	5.419000	6.088000	5.600000	5.971000	6.032000	5.969000	5.152000	4.882000	4.656000	0.000000	2.232705e+07
max	16.000000	8.177000	8.647000	8.371000	6.938000	6.941000	6.939000	6.971000	6.912000	6.971000	1.000000	2.022460e+09

Figura 1: Statistiche di base del dataset

Dopo questa prima analisi, si è passati ad analizzare le variabili e la loro distribuzione per mezzo di grafici. La prima ad essere visualizzata graficamente è stata la *feature* rappresentante la polisemia, in quanto oggetto principale delle analisi. Si è così potuto osservare che il dataset è fortemente sbilanciato poiché su un totale di 4682 record, appena 379 sono polisemiche, il che probabilmente costituirà un problema anche nelle fasi successive.

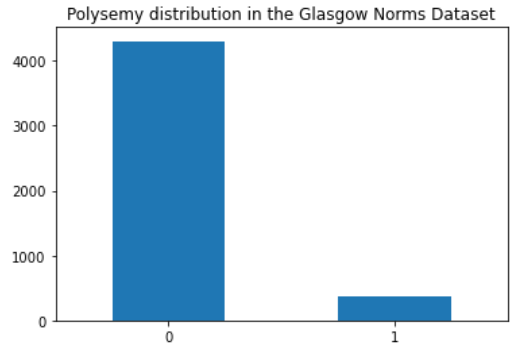

Figura 2: Distribuzione della polisemia nel dataset

Si è proceduto quindi a visualizzare le distribuzioni di alcune variabili numeriche per mezzo di istogrammi, così da visualizzare se le variabili possiedono o meno una distribuzione normale, informazione utile da sapere in vista della visualizzazione della matrice di correlazione e della applicazione degli algoritmi. Si è notato che molte delle variabili (un'eccezione è *gender*, come si vede nella figura in basso), possiedono una distribuzione asimmetrica, con un picco della curva lontano dal centro.

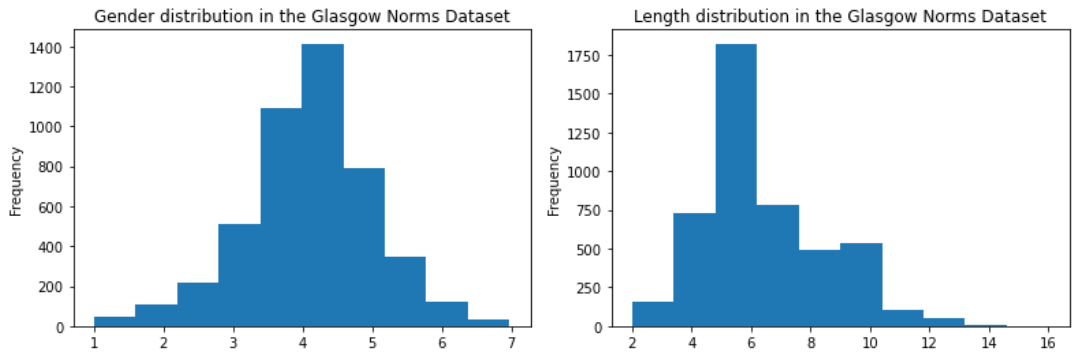

Figura 3: Distribuzione feature Gender e Length

Per velocizzare lo studio della distribuzione delle variabili è stato anche utilizzato un *pairplot* di Seaborn, che ha permesso di studiare sia le correlazioni a coppie di variabili che la distribuzione di ogni singola variabile.

Sono state confrontate successivamente le distribuzioni di variabili che possiedono un valore massimo e minimo differente, così da capire se vi fosse una somiglianza, che si riscontra solo minimamente nel caso delle feature con distribuzione compresa tra 1 e 9.

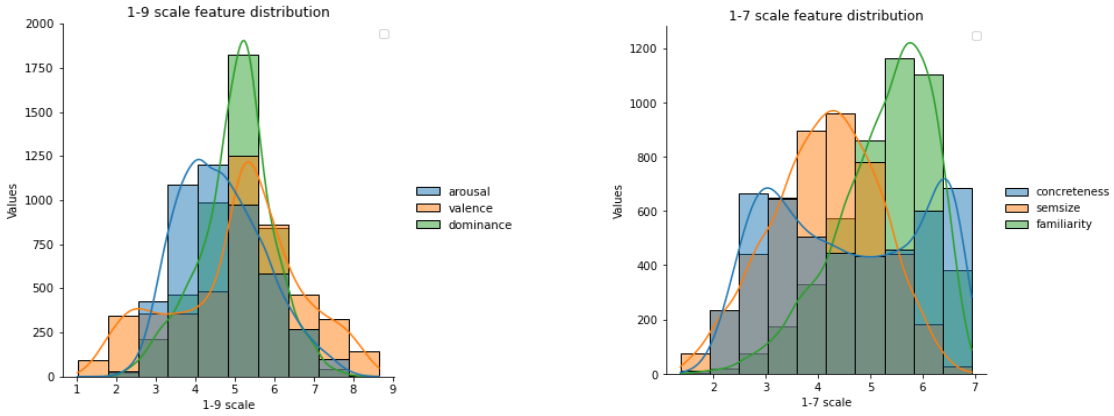

Figura 4: Distribuzione delle feature su scala 1-9 e su scala 1-7

Le successive analisi hanno voluto indagare la distribuzione della polisemia in relazione alle altre feature: si è, per esempio, visualizzato come si distribuisse la lunghezza delle parole in base al loro essere polisemiche o no: si è notato così che gran parte delle parole polisemiche è corta, dato che gran parte di esse possiede 4 o 5 caratteri e solamente una ne possiede 10, il valore massimo.

Infine, si è studiata la correlazione tra le variabili utilizzando il coefficiente di Correlazione di Spearman, vista la distribuzione non normale di alcuni attributi.

È stata quindi ricavata la seguente matrice di correlazione.

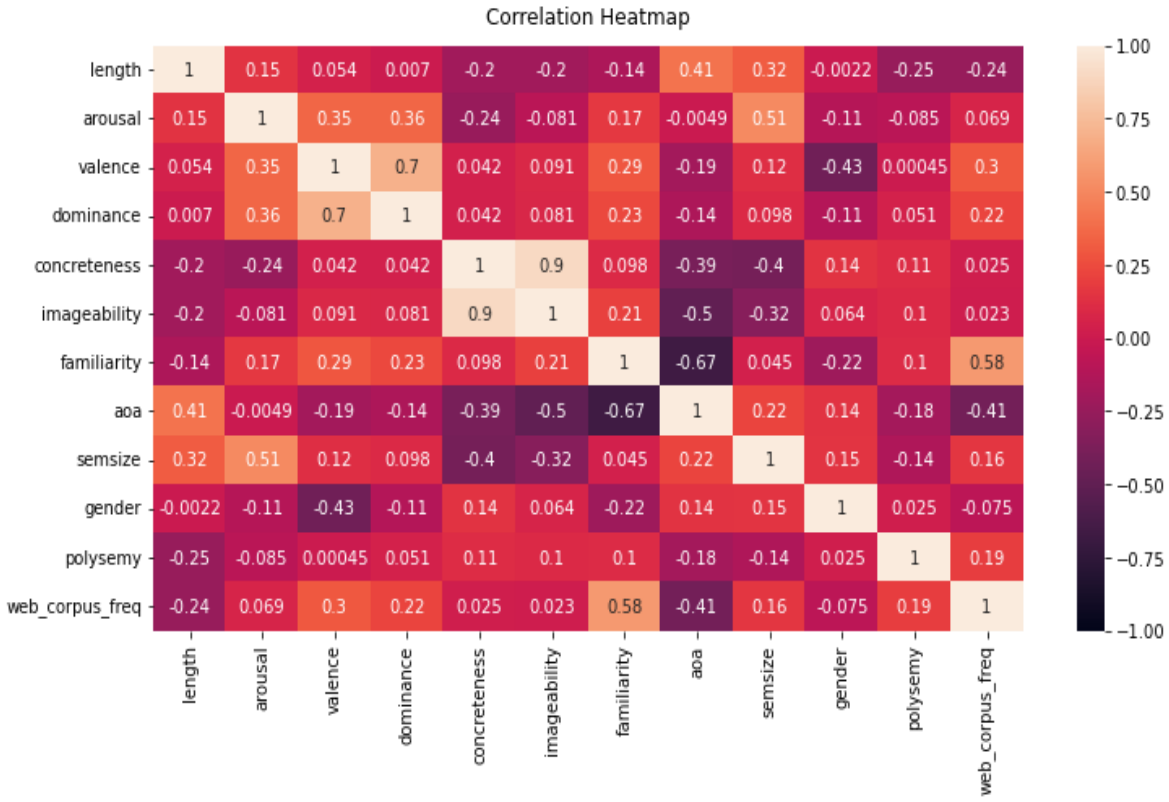

Figura 5: Matrice di correlazione del dataset prima della data preparation

In generale, le feature maggiormente correlate sono *concreteness* e *imageability*, che sono correlate positivamente (0.9), così come *valence* e *dominance* (0.7). Dall'altro lato, passando alle feature negativamente correlate, si ritrovano *familiarity* ed *aoa*, quindi al crescere dell'una, diminuisce l'altra (-0.67). Per il resto si può dire che vi sano dei livelli di correlazione bassi, che non superano il ± 0.25.

Data preparation

Il dataset, per la sua natura linguistica, è stato allo stesso tempo semplice ma complesso da trattare.

Dalla fase di *Data Understanding* era emersa la presenza di 14 valori mancanti nell'attributo *web_corpus_freq*. In questo caso si è prima verificato se le parole di cui era mancante l'informazione fossero polisemiche: dopo aver constatato la loro natura non polisemica si è deciso di cancellare le 14 righe del dataset, ottenendo quindi un totale di 4668 record. Se i record fossero stati polisemici sarebbe stato meglio intervenire recuperando i valori da quelli di parole simili in quanto si sarebbero perse importanti informazioni che avrebbero aumentato ulteriormente lo sbilanciamento del dataset.

Passando invece alla parte di *feature engineering*, si è deciso di non creare delle nuove variabili, ma di eliminare *imageability*, in quanto fortemente correlata con *concreteness*, in modo da evitare problematiche in ottica di clustering e classificazione. Le altre variabili sono state tenute, in quanto il dataset risultava estremamente ridotto in termini di numero di features.

Si sono poi studiati gli outlier analizzando la distribuzione delle feature numeriche all'interno del dataset per mezzo di boxplot. Si sono trovati valori considerati outlier in *length, arousal, valence, dominance, familiarity, gender* e *web_corpus_freq*.

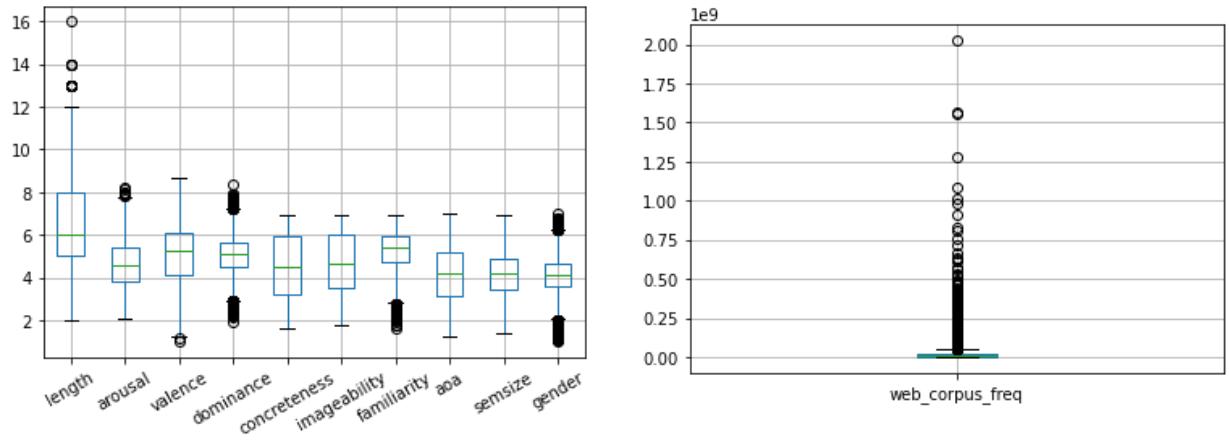

Figura 6: Boxplot delle features del dataset

Per comprendere meglio se questi outlier fossero dovuti ad errori nel dataset o meno, si è controllato se vi fossero valori di *arousal*, *valence* e *dominance* maggiori di 9, valore massimo fissato per la variabile, e si è fatto lo stesso, ma impostando la *threshold* a 7 per controllare i valori di *gender*. Si è controllata anche la soglia opposta, quindi se vi fossero valori minori di 1, situazione che non si verifica in nessuna delle variabili. Studiando *web_corpus_freq*, si è potuto constatare che il valore considerato più *outlier* è in realtà una parola reale, *all*, che ricorre con una frequenza estremamente elevata, ma comunque plausibile considerando che la distribuzione della frequenza delle parole in un corpus di dati linguistici segue una *power law* (Legge di Zipf). Anche in questo caso non sono stati quindi eliminati valori.

Clustering

Per effettuare il clustering dei dati sono stati sfruttati diversi approcci. Il primo utilizzato è stato il K-means, seguito poi da DBSCAN e OPTICS e, per finire, si sono tentate diverse configurazioni di clustering gerarchico. Per effettuare il clustering si è deciso di ricorrere ad un subset di features tra la rosa di quelle presenti nel dataset: si è deciso di tenere come feature *arousal, dominance, concreteness, familiarity, aoa, semsize, gender*. Si sono quindi tenute solo le feature proprie dell'esperimento che ha originato il dataset, ad eccezione di *valence*, che era positivamente correlata con *dominance* ed è stata scartata.

K-Means

Per effettuare il clustering con K-means, dato che esso si basa sulla distanza dei datapoint da un centroide, si è deciso di effettuare lo scaling dei dati utilizzando il *MinMaxScaler,* dato che le feature non hanno distribuzione normale.

Per trovare il valore corretto del parametro K su cui si basa un buon risultato del clustering, si è utilizzato un SSE plot, in cui sono stati provati valori di K compresi tra 1 e 21 e si è osservato in occorrenza di quale K vi fosse il maggiore abbassamento dell'SSE.

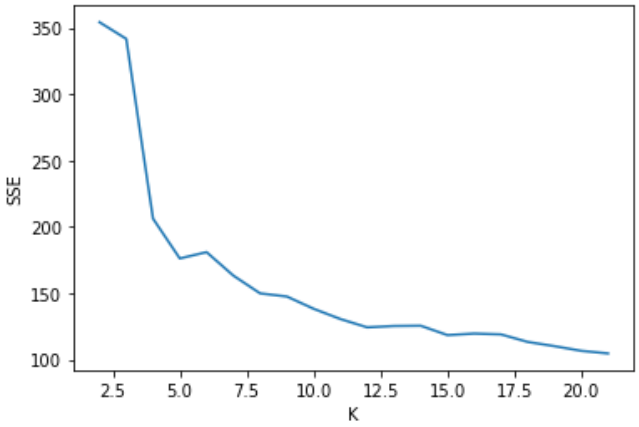

Figura 7: Calcolo dell'SSE per la scelta del K

In questo caso si è preso un $K = 3$ e si è effettuato il clustering vero e proprio, ottenendo un valore di Silhouette pari a 0.233. Sono stati ottenuti tre cluster abbastanza bilanciati in termini di dimensioni, come verrà descritto successivamente. Si sono visualizzati poi i centroidi utilizzando il *parallel coordinates* nella figura in basso e dei grafici a barre per osservare la distribuzione della polisemia all'interno di ciascun cluster.

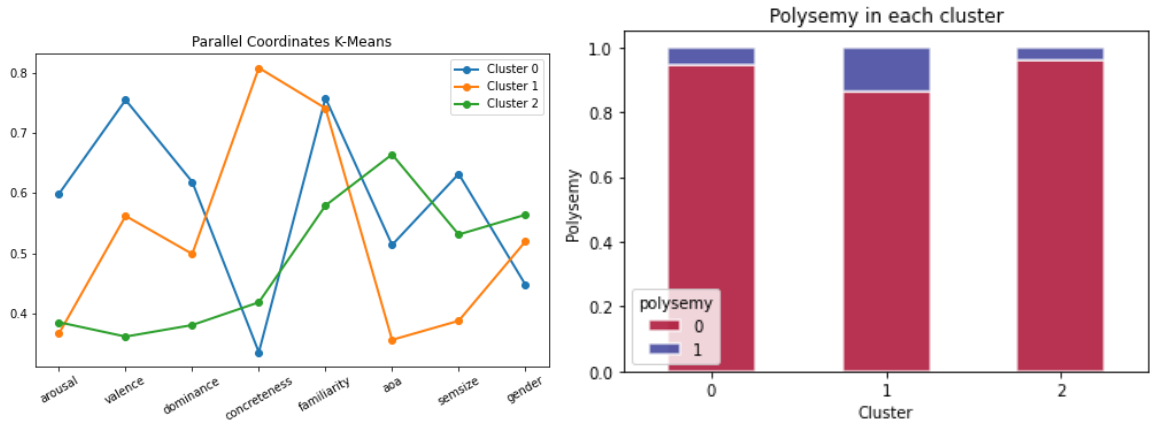

Figura 8: Grafico a coordinate parallele e distribuzione della polisemia nei cluster

Cluster ottenuti

Cluster 0: composto da 1088 record, sono parole apprese intorno ai 10 anni, considerate più concrete ma meno familiari delle altre. Sono inoltre parole tendenzialmente associate ad una connotazione maschile. Al suo interno c'è una piccola componente di 57 parole polisemiche.

Cluster 2: 1833 parole, è il cluster con la maggiore componente di parole polisemiche (252 totali). Contiene parole apprese quando si è molto piccoli, caratterizzate dal loro essere meno familiari (si ricorda che *familiarity* è su una scala dal più familiare al meno familiare) e più astratte. È anche il cluster in cui c'è il maggior numero di parole di lunghezza media/breve; quindi, coincide con

l'informazione appresa in fase di *data understanding* legata alla breve lunghezza delle parole polisemiche.

Cluster 3: 1747 parole, è il cluster in cui si trovano le parole apprese in età più avanzata, dopo i tredici anni e meno familiari ai partecipanti dell'esperimento psicolinguistico; sono parole considerate positive, più astratte e legate a categorie semantiche di oggetti e concetti "grandi" e associati al genere femminile. Anche in questo caso, così come nel *Cluster 0*, vi è una piccola componente di parole polisemiche (70).

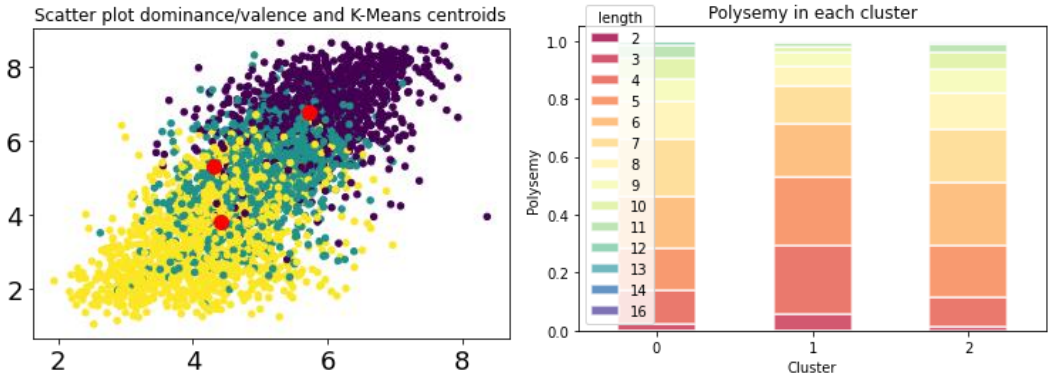

Figura 9: Scatter plot con le feature dominance e valence (a sinistra) e distribuzione della lunghezza delle parole nei cluster

Dallo scatter plot sopra, oltre ai centroidi dei tre cluster, si può osservare come i cluster non siano divisi nettamente a causa del valore basso di *Silhouette.*

Clustering density-based

DBSCAN

Il DBSCAN è stato il primo approccio di clustering basato sulla densità testato. Come per K-Means si è effettuata la ricerca del miglior valore per il parametro fondante del DBSCAN, *l'eps*, che è stato ricercanto attraverso il grafico sottostante, in cui si è calcolata la distanza, per ciascun punto del dataset, dal suo vicino più vicino.

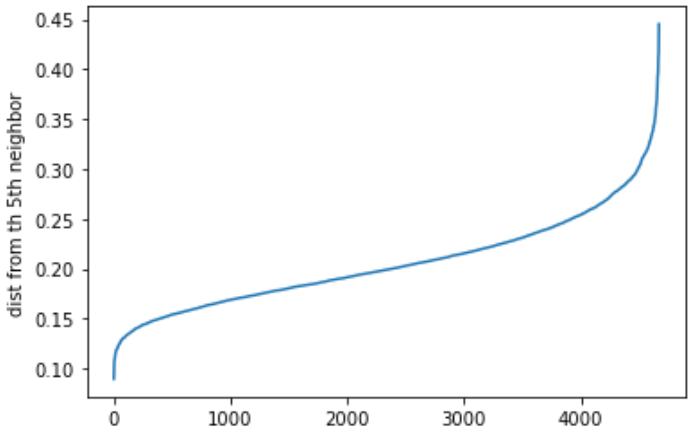

Figura 10: Scelta dell'EPS

Il valore ottimale di *eps* è 0.30, per cui è stato effettuato il clustering usando tale valore, impostando il parametro *min_samples* a 3 e utilizzando la distanza euclidea come misura di distanza. In tale modo è stata ottenuta una silhouette di 0.14, ma dei cluster molto approssimativi: è infatti stato riconosciuto solo un grosso cluster di 4645 record, un altro che conteneva però 0 record e 23 punti considerati dall'algoritmo del rumore. Si è provato anche con la distanza di Manhattan, ma senza successo: in questo caso, il grosso cluster si è frammentato in numerosi piccoli cluster di al più 114 record l'uno e con una Silhouette pari a -0.47.

Si è anche provato ad aumentare il numero di *min_samples*, ma ottenendo comunque risultati deludenti sia con la distanza di Manhattan che con l'Euclidea.

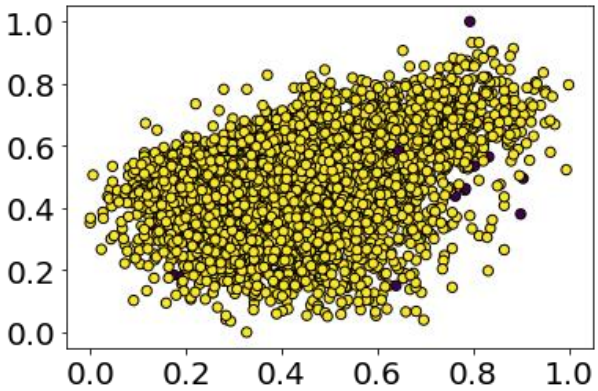

Figura 11: Clustering DBscan con un cluster (in giallo) e rumore (in viola) ottenuto con eps = 0.3

OPTICS

Un altro approccio è stato quello con l'OPTICS, dove però si è verificato nuovamente il problema della frammentazione in piccoli e numerosi cluster, 238 in totale più un cluster di *noise points*. Il problema è stato che gran parte del dataset viene considerato come rumore, tra cui le stesse parole polisemiche. Il tentativo di migliorare quanto ottenuto con il DBscan può dirsi quindi non essere andato a buon fine.

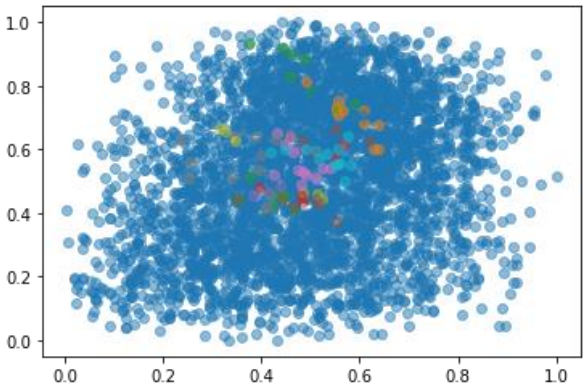

Figura 12: Microcluster identificati dall'OPTICS e rumore (azzurro)

Clustering gerarchico

Per il clustering gerarchico sono stati usati quattro diversi approcci: *Complete*, *Ward*, *Average* e *Single*, ottenendo i dendrogrammi nella figura 13.

Sono stati utilizzati gli stessi attributi usati nelle fasi precedenti del lavoro.

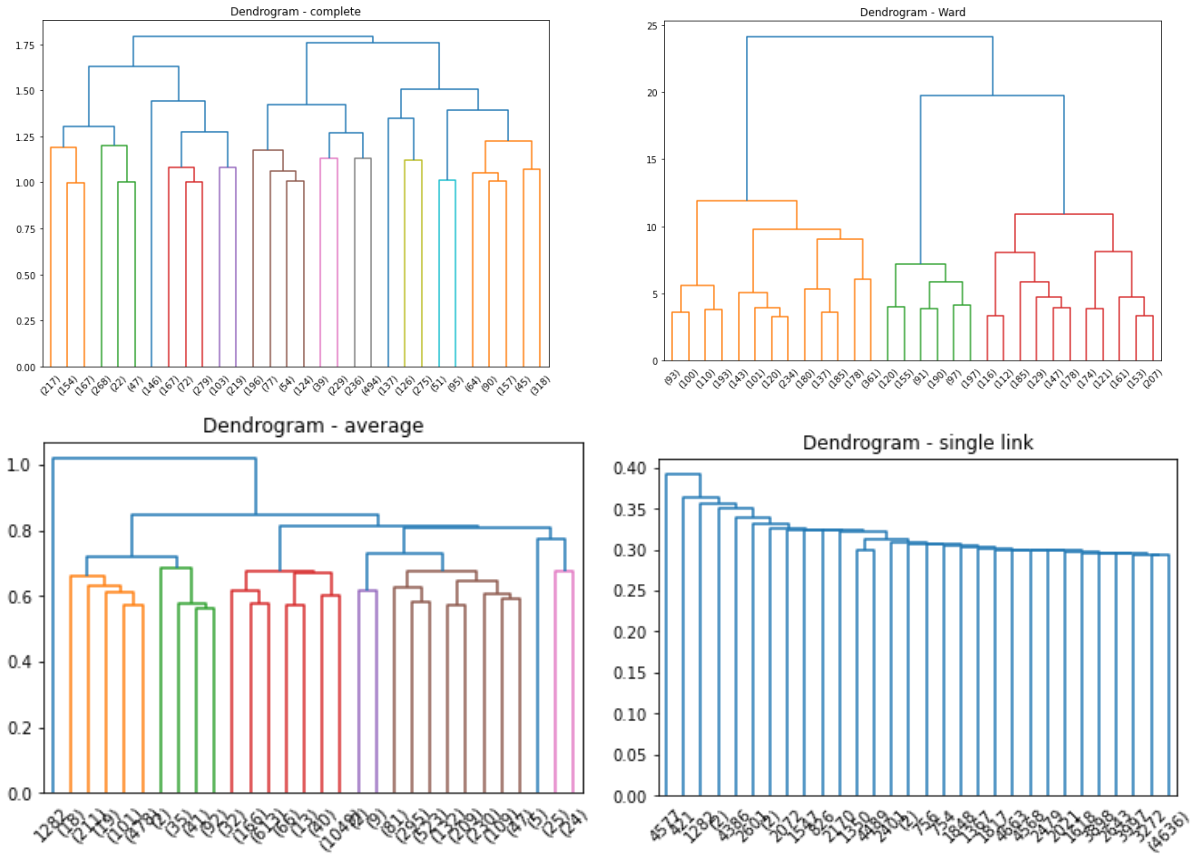

Figura 13: Dendrogrammi delle diverse configurazioni di clustering gerarchico

Oltre all'osservazione del dendrogramma risultante da ciascun approccio, si è osservata la variazione della *Silhouette*, una volta individuato il numero ottimale di cluster attraverso il dendrogramma. Sono state effettuate diverse prove, di cui si possono osservare i risultati nella seguente tabella.

Metodo	Silhouette	Dimensioni cluster
Complete + Euclidean	0.0003	0: 683, 1: 168, 2: 3817
Complete + Manhattan	-0.009	0: 3782, 1: 637, 2: 249
Ward + Euclidean	0.19	0: 2135, 1: 1683, 2: 850
Average + Euclidean	0.23	0: 4667, 1: 1

Average + Manhattan	0.24	0: 4667, 1: 1
Single Link + Euclidean	0.18	0: 4667, 1: 1

Tabella 2: Risultati configurazioni Gerarchico

Si può quindi affermare che i migliori risultati in termini di *Silhouette* si siano ottenuti con Ward e distanza euclidea, che ha permesso oltre che di ottenere uno dei risultati più elevati di *Silhouette* anche tre cluster piuttosto bilanciati, a differenza di quanto accade con la configurazione *Average* e *Euclidean*, in cui sì, si ottiene un valore superiore di Silhouette, ma un fortissimo sbilanciamento delle dimensioni dei due cluster. In questo caso, si è provato ad aumentare anche il numero di cluster, ma ottenendo di contro solo la creazione di nuovi piccoli cluster di 1 o 2 elementi, a discapito della S*ilhouette*.

Conclusioni sul clustering

In tabella sono riassunti i risultati migliori ottenuti con i diversi cluster. I risultati migliori sono stati ottenuti con il K-Means, che ha dato il maggior valore di Silhouette, seguito dalla configurazione Ward e distanza euclidea del clustering gerarchico. Quelli peggiori sono stati ottenuti invece con entrambi gli approcci *density-based*, sia in termini di *silhouette* che di tipologia di cluster identificati, in quanto è stato trovato un unico cluster e un "cluster" di *noise points,* oppure tanti microcluster di pochi punti accompagnati da un cluster più grande esclusivamente di "rumore".

Algoritmo	Cluster	Silhouette
K-Means	3	0.23
DBscan	1 (+ noise)	0.14
Ward+euclidea	3	0.19

Tabella 3: Risultati finali clustering

Classificazione

Nel caso della classificazione, ci si è dovuti scontrare con la problematica principale del dataset, il suo sbilanciamento nella distribuzione della variabile oggetto della classificazione, la *polisemia*. Il primo approccio è stato quello di selezionare il subset delle variabili da usare con i diversi algoritmi. Sono state scelte le seguenti: *length, arousal, valence, dominance, concreteness, familiarity, aoa, semsize* e *gender*; *polysemy* è diventata invece la *target variable*, l'etichetta da assegnare ai dati.

Il dataset è stato suddiviso in *training* e *test* (con proporzione 70/30) per poter svolgere l'addestramento del modello e la valutazione vera e propria dello stesso; in questa fase di *split* dei

dati è stato inoltre impostato il parametro *stratify*, così da mantenere la proporzione tra parole polisemiche e non in entrambi i subset.

Il primo modello realizzato è stato il *Decision Tree*. Prima dell'addestramento vero e proprio si è effettuata la fase di *parameter tuning* per mezzo di una *RandomSearch*, in cui è stato provato l'algoritmo con diverse combinazioni dei valori dei parametri *max_depth, min_samples_split, min_samples_leaf* e *criterion*. La *RandomSearch* è stata eseguita sull'intero *training set* e il modello migliore è stato trovato valutando l'accuratezza con *cross-validation*.

I parametri migliori secondo questa prima *RandomSearch* sono stati: *min_samples_split*: 10, *min_samples_leaf*: 100, *max_depth*: 12, criterion: *gini*. Il modello è stato quindi addestrato e testato sull'apposito subset di test. I risultati ottenuti sono stati piuttosto deludenti, in quanto nonostante l'accuratezza estremamente elevata (91% sul *test set*), a causa dello sbilanciamento del dataset, il modello ha riconosciuto precisamente solo i record non polisemici, classificando come non polisemiche anche le parole in realtà polisemiche. Questo è stato osservato attraverso la matrice di confusione, dove il modello ha dimostrato di non riconoscere nemmeno un elemento della classe minoritaria, considerandolo di classe 0 (non polisemico).

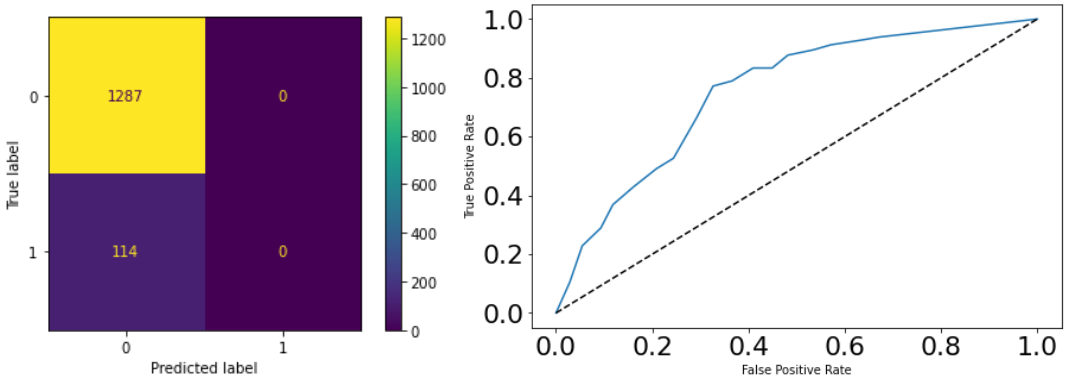

Tabella 4: Confusion Matrix e ROC curve del Decision Tree

Questo risultato, proprio per il fatto che non vengono riconosciuti i record effettivamente di interesse, non può essere considerato soddisfacente, per cui si sono utilizzati diversi approcci per provare a migliorare il riconoscimento della classe minoritaria.

Il primo approccio tentato è stato l'intervenire sull'algoritmo in sé, andando a impostare un peso superiore alla classe 1, così da spingere l'algoritmo a classificare correttamente quella classe. Impostando il peso a 10 per la classe 1 e a 1 per la classe 0, si è ottenuto un aumento dell'accuratezza nei confronti della classe di interesse, come si può vedere in Tabella 6. Si è ottenuto anche un aumento della *recall* e dell'*f1-score*, che invece prima erano pari a 0.

Feature	Peso
Length	0.21
Aoa	0.15
Dominance	0.15
Semsize	0.09
Concreteness	0.08

Tabella 5: Pesi delle cinque features più importanti del Decision Tree

Dopo questo primo approccio, ci si è spostati su altre metodologie di *Unbalanced Learning* per capire se si potesse ottenere un migliore riconoscimento della polisemia e si è intervenuti sui dati. Sono stati provati anche due approcci differenti di *oversampling*: prima si è usato il *Random Oversampling*, creando quindi delle copie di record polisemici per bilanciare le due classi e,

successivamente, visti i risultati ottenuti, ci si è rivolti allo SMOTE, quindi alla creazione di record sintetici per mezzo dell'interpolazione. In entrambi i casi, come si può osservare nella Tabella 6, i risultati ottenuti sono stati migliori in termini di riconoscimento della classe minoritaria, in particolare con il *Random Oversampling*, in cui si è ottenuto un forte miglioramento della *precision* e della *recall* (sebbene il migliore risultato in termini di *recall* si ottenga con lo SMOTE).

	Precision classe 0	Precision classe 1	Recall classe 0	Recall classe 1	F1-Score classe 0	F1-Score classe 1
Decision Tree	0.92	0	1	0	0.96	0
Decision Tree e pesi (0: 1; 1: 10)	0.93	0.17	0.88	0.27	0.91	0.21
Decision Tree e Random Oversampling	0.94	0.22	0.88	0.39	0.91	0.28
Decision Tree e SMOTE	0.95	0.20	0.83	0.47	0.88	0.28
Random Forest	0.94	0.00	0.93	0.00	0.94	0.00
Random Forest e pesi (0: 1; 1: 10)	0.93	0.32	0.97	0.18	0.95	0.23
KNN	0.92	0.00	1.00	0.00	0.96	0.00

Tabella 6: Performance dei modelli sul test set

Il risultato migliore ottenuto è stato quello del *Decision Tree* con il *Random Oversampling*, impostando come parametri *min_sample_split* a 2, *max_depth* a 17, l'*entropia* come misura di impurità dei nodi e *min_samples_leaf* a 1. Di esso, si è deciso di visualizzare, oltre alla matrice di confusione, anche la *ROC curve* e i pesi assegnati dal classificatore a ciascuna delle feature utilizzate.

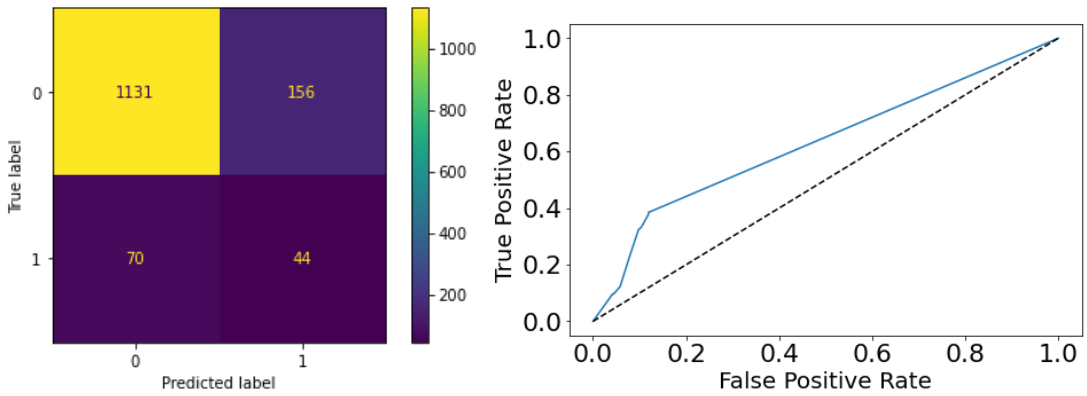

Figura 14: Matrice di confusione e ROC curve Decision Tree con Random Oversampling

In questo caso, si può osservare dalla matrice di confusione, che vengono riconosciuti 44 record polisemici. Anche se questo è un risultato tutto meno che positivo, in quanto è ancora presente un elevato numero di record classificati erroneamente come *non polisemici*, può essere considerato un

risultato positivo rispetto ai risultati ottenuti con il *decision tree* nella sua versione base.

Si è infine osservata la variazione nel peso assegnato a ciascuna feature rispetto al decision tree, in cui si è notata una maggiore importanza della feature *dominance*, rispetto ad *aoa.*

Infine, in Figura 15 viene presentato l'albero ottenuto con il *Random Oversampling.*

Feature	Peso
Length	0.20
Dominance	0.13
AoA	0.11
Gender	0.11
Concreteness	0.10

Tabella 7: Pesi delle prime cinque features più importanti (DT e Random Oversampling)

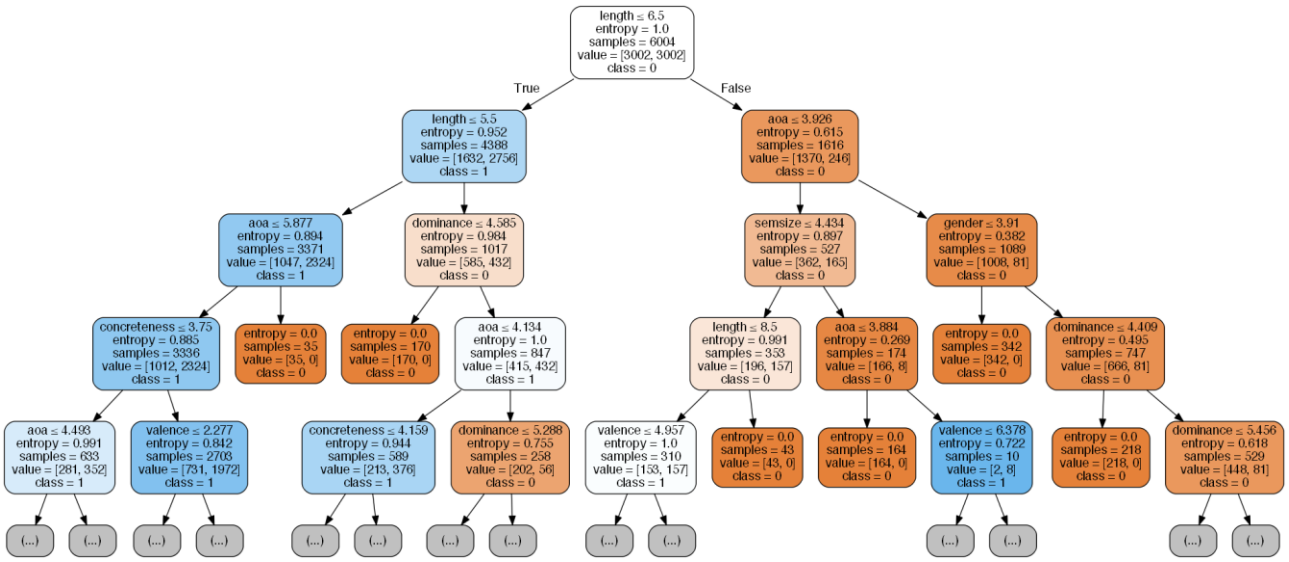

Figura 15: Albero Decision Tree con Random Oversampling

Altri tentativi sono stati poi effettuati utilizzando la versione *ensemble* del *decision tree,* il Random Forest.

Anche in questo caso si è deciso di utilizzare il *RandomSearch* per trovare i parametri migliori per addestrare il classificatore. Si è quindi addestrato il modello con *min_ samples_ split* pari a 2, *min_ samples_ leaf* pari a 1, *max_ depth* pari a 19 e l'indice di Gini come misura di impurità. In questo caso, il modello ottenuto ha dato il risultato migliore in termini di *precision* per il riconoscimento della classe minoritaria, anche se questo non coincide con quanto visualizzato poi nella matrice di confusione, dove si riscontrano solo 20 record polisemici classificati correttamente. Migliore, in questo caso, la ROC-AUC curve, pari a 0.79.

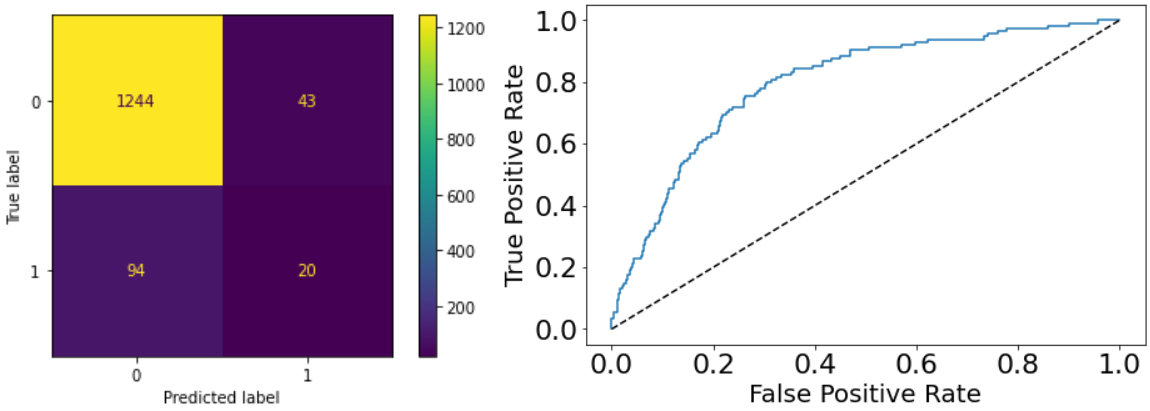

Figura 16: Matrice di confusione del Random Forest e ROC-AUC curve

È stato infine provato il K-Nearest Neighbour, effettuando anche in questo caso il *tuning* per la ricerca dei migliori valori dei parametri. I risultati ottimali sono stati ottenuti impostando 24 vicini, p pari a 2 e *leaf_size* pari a 1. Anche in questo caso, però, si è verificato il problema di riconoscimento della classe minoritaria, in quanto non è stato riconosciuto nemmeno un record polisemico, come si può osservare dai risultati riportati in Tabella 6.

Conclusioni sulla classificazione

I modelli addestrati hanno dimostrato di soffrire nella maggior parte dei casi, di problematiche legate allo scarso (se non nullo) riconoscimento della classe minoritaria, quella di maggiore interesse. Nonostante i diversi tentativi di migliorare i risultati ottenuti (*oversampling* e pesi nel classificatore), non sono stati ottenuti risultati ottimali, ma solo risultati mediocri. I modelli migliori sono risultati il *Decision Tree* dopo essere stato addestrato con i dati bilanciati con il *Random Oversampling* e il *Random Forest*, che però ha dimostrato di performare in maniera inferiore rispetto al *Decision Tree* nel riconoscimento della polisemia.

Association Rules e Pattern Mining

In questa ultima parte di analisi, si è svolta l'estrazione di *association rules* per mezzo della libreria Python *Pyfim*.

Prima di poter applicare i metodi di estrazione delle regole, è stato necessario effettuare il preprocessing dei dati, in particolare si è dovuta effettuare la discretizzazione delle variabili numeriche in una serie di intervalli (quattro per ogni feature). I valori numerici originali sono stati così sostituiti dai nuovi valori discretizzati rappresentati attraverso un range nella forma *(n1,n2]_nome_feature*.

Una volta effettuata questa piccola operazione di preprocessing, sono stati estratti i *basket* per mezzo della libreria *Pyfim* ed è stato analizzato il numero di *frequent, closed* e *maximal itemset* al variare del support.

Si è esclusa fin da subito la possibilità di estrarre degli *itemset* di quattro elementi, in quanto si è ottenuto un solo itemset anche con il parametro *min_sup* impostato al 10%, quindi estremamente basso. Abbassando la lunghezza minima degli *itemset* a 3, si ottengono invece 21 *itemset* a patto che si utilizzi un support minimo del 10% di cui sopra. Non sono stati ottenuti invece *frequent itemset* di tre elementi utilizzando un support minimo pari al 20%, mentre, abbassando ulteriormente il numero di elementi per *itemset,* è stato possibile ottenere 38 *frequent itemset* con *min_sup* del 20%.

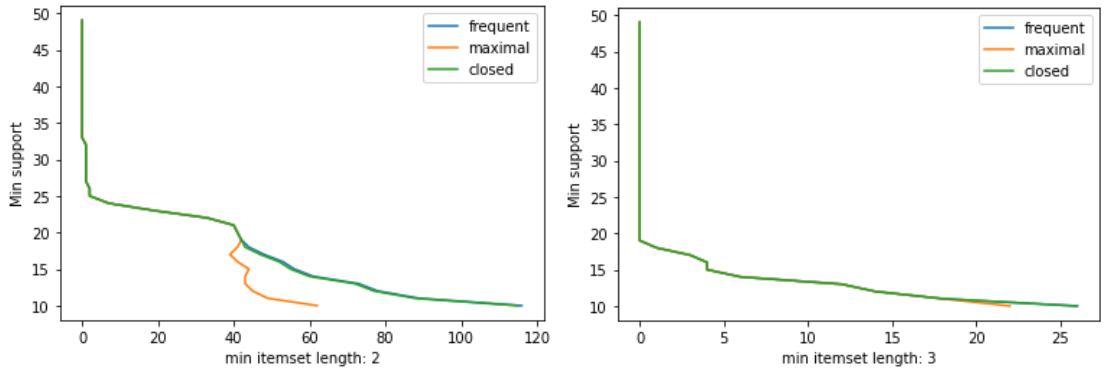

Figura 17: Numero di itemset al variare di min_support

Si sono quindi visualizzati i 38 *frequent itemset* di almeno due elementi con support minimo del 20%, come è possibile osservare in tabella 8.

Itemset	Support
(1.999, 5.0]_length, Not polysemic')	32.11
(6.0, 8.0]_length, Not polysemic'	26.45
(1.635, 3.24]_concreteness, Not polysemic	24.29

Tabella 8: Itemset di due elementi con min_sup>20%

Dagli *itemset* estratti in questa fase, si può già osservare che due delle *itemset* tra quelli con *support* più elevato sono legati alla lunghezza della parola, che se compresa tra una lunghezza di 2 e 8 caratteri sarebbe non polisemica: questo va in contrasto con quanto appurato inizialmente in fase di *data understanding*, dove si era notata proprio la maggiore diffusione di parole brevi con il tratto della polisemia. Sono stati successivamente visualizzati gli itemset di 3 elementi con *min_sup>10%* (Tabella 9).

Itemset	Support
('(2.056, 3.846]_arousal', '(4.094, 5.286]_valence', 'Not polysemic')	10.83
('(5.152, 6.833]_aoa', '(1.6460000000000001, 4.706]_familiarity', 'Not polysemic')	14.80
('(6.088, 8.647]_valence', '(5.6, 8.371]_dominance', 'Not polysemic'), 14.674378748928877)	14.67

Tabella 9: Itemset con 3 elementi e min_sup>10%

A questo punto, sono stati estratti i closed itemset di tre elementi e *min_sup* pari a 10%.

Closed Itemset	Support
(5.969, 6.939]_familiarity, (1.37, 3.118]_aoa, Not polysemic	13.06
(2.056, 3.846]_arousal, (4.094, 5.286]_valence, Not polysemic	10.83
(5.152, 6.833]_aoa, (1.635, 3.24]_concreteness, Not polysemic	10.17

Tabella 10: Frequent Itemsets

Si è proseguito andando a estrarre i *maximal itemset*, alcuni dei quali presentati in Tabella 10.

Maximal Itemset	Support
(4.882, 6.912]_semsize', '(6.088, 8.647]_valence', 'Not polysemic')	10.26
(5.152, 6.833]_aoa', '(1.6460000000000001, 4.706]_familiarity', 'Not polysemic'),	14.80
(('(12769.999, 1671099.5]_web_corpus_freq', '(1.6460000000000001, 4.706]_familiarity', 'Not polysemic'),	13.21

Tabella 11: Maximal Itemsets

Anche in questo caso, come nella tabella precedente, si può notare che vengono associati solo diversi intervalli degli attributi numerici con la caratteristica delle parole di *non essere polisemiche*. Questo viene inoltre confermato dal grafico in figura 18, dove si nota la presenza di regole polisemiche solo in presenza di *support* eccessivamente basso.

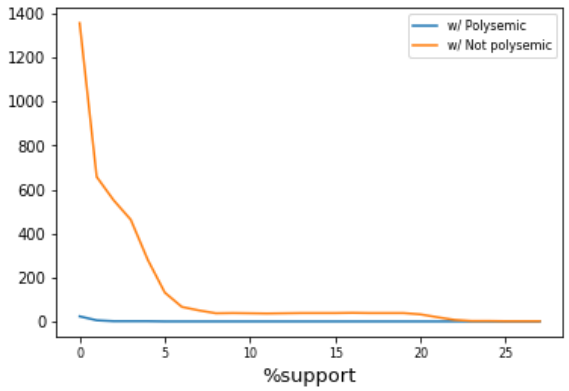

Figura 18: Numero di regole sulla polisemia e sulla non polisemia al variare del support

Una volta estratti i *maximal itemset,* si è passati alla generazione delle regole e allo studio della loro variazione al mutare della *confidence*: con degli itemset di 3 elementi, è possibile generare regole con un alto livello di confidence (maggiore di 10) solo con *min_sup* pari a 10, casistica in cui

vengono generate circa 30 regole; in caso aumenti il support minimo, come si vede in figura 19, verranno generate pochissime regole.

Diversa la situazione con gli *itemset* di due elementi, dove con livelli di *confidence* minima all'80% è possibile avere circa 50 regole anche mantenendo un *support* minimo al 20%

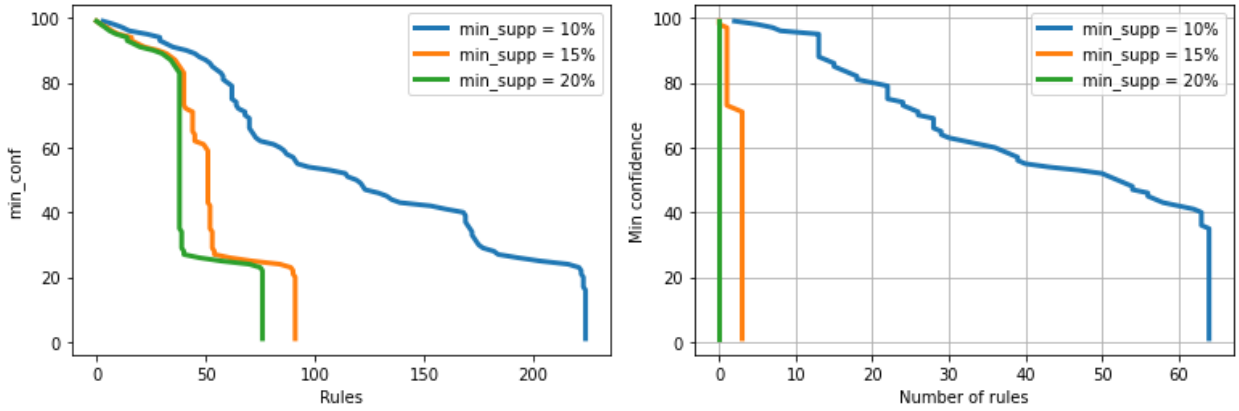

Figura 19: Variare del numero di regole al variare del min_sup e della confidence con itemset di 2 e 3 elementi

Con la configurazione di 3 elementi per *itemset* e una *confidence* minima del 60%, sono state ottenute 30 regole con *lift* superiore ad 1 e 6 con *lift* intorno a 1. Il numero più elevato di regole lo si trova in corrispondenza di una *confidence* elevata, pari al 96%.

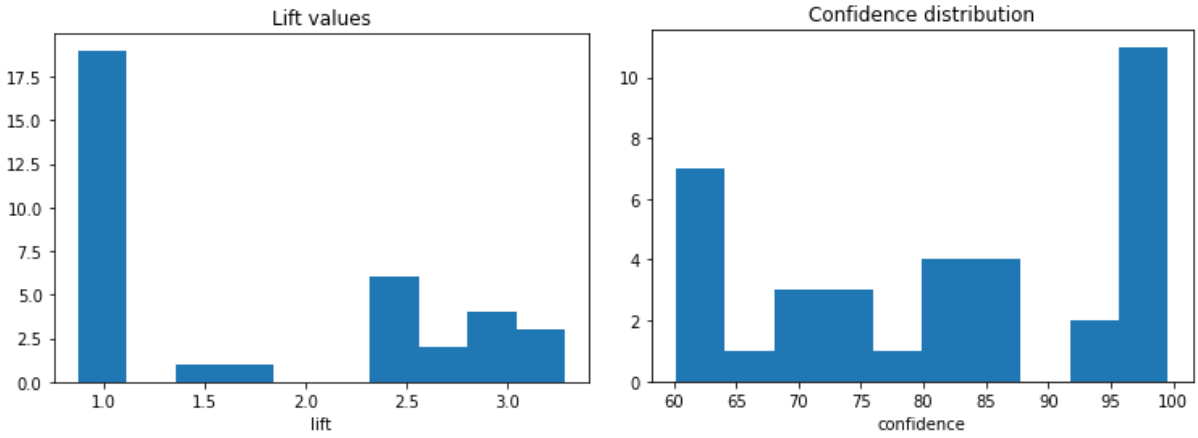

Figura 20: Valori di Lift e Confidence

Tra le regole ottenute, solo alcune hanno come conseguente la polisemia, in particolare la *non polisemia*. Dall'analisi del lift, si nota che solo alcune regole hanno un lift superiore a 2, per cui le variabili nelle regole ottenute sono positivamente correlate, mentre la maggior parte di esse possiedono lift pari o inferiore a 1, quindi sono regole poco informative, in quanto l'associazione tra antecedente e conseguente della regola è casuale.

In tabella 21 sono riportate alcune delle regole ottenute con lift più elevato.

Regola	Support	Confidence	Lift
('(6.088, 8.647]_valence', '(5.6, 8.371]_dominance', 'Not polysemic') => (5.419, 8.177]_arousal	10.26	69.92	2.79
('(5.969, 6.939]_familiarity', '(1.37, 3.118]_aoa') => (1.999, 5.0]_length	10.06	65.36	1.7
('(1.37, 3.118]_aoa', 'Not polysemic') => (1.999, 5.0]_length	13.02	60.92	1.58
('(1.37, 3.118]_aoa', '(1.999, 5.0]_length') => (5.969, 6.939]_familiarity	10.06	62.08	2.49
('(5.6, 8.371]_dominance', '(5.419, 8.177]_arousal') => (6.088, 8.647]_valence	10.71	79.49	3.19

Figura 21: Alcune delle regole estratte con i rispettivi valori di Support, Confidence e Lift

Si può osservare che nelle regole con lift più alto non appare la polisemia come conseguente ma solo come antecedente; importante, inoltre, il fatto che tutte le regole estratte presentino la *non polisemia*, visti gli *itemset* ottenuti inizialmente. Le regole che possiedono come conseguente la *non polisemia* sono invece caratterizzate da lift estremamente basso e per questo sono poco informative.

Regola	Support	Confidence	Lift
('(4.882, 6.912]_semsize', '(6.088, 8.647]_valence') => Not polysemic	10.26	96.76	1.05
('(4.882, 6.912]_semsize', '(5.419, 8.177]_arousal') => Not polysemic	13.13	97.61	1.06

Figura 22: Alcune delle regole estratte per la non polisemia

Conclusioni

In questo elaborato si è cercato di studiare il fenomeno della polisemia per mezzo di tecniche e algoritmi di *data mining.*

Il primo passo è stato quello di studiare il dataset e prepararlo per le successive analisi: in questa fase è emerso il problema del dataset, il suo sbilanciamento nei confronti delle parole non polisemiche.

Si è poi passati all'applicazione di algoritmi di clustering, nella quale, grazie soprattutto all'algoritmo K-Means si è osservata la presenza di tre diversi raggruppamenti di dati simili. In questa fase si è distinto particolarmente un cluster contenente un numero superiore rispetto agli altri di parole *polisemiche.*

Con la classificazione sono stati ottenuti risultati discreti, in particolare con il *Random Oversampling* in combinazione con il *Decision Tree* e con il Random Forest. Si è osservato che la *feature* determinante per l'identificazione della polisemia è la lunghezza delle parole, seguita da *dominance* ed *age of acquisition.*

L'ultima parte è stata destinata all'estrazione di regole di associazione, in cui sono state trovate esclusivamente regole legate alla *non polisemia*.

Ulteriori indagini potrebbero essere legate all'utilizzo di ulteriori tecniche per bilanciare il dataset o all'utilizzo di altri algoritmi.