DIPARTIMENTO
DI INFORMATICA

PROGETTO DI INGEGNERIA DELLA CONOSCENZA: SENTIMENT ANALYSIS SU RECENSIONI DI AMAZON KINDLE

Indice

1 Gruppo

Il progetto è stato realizzato da:

– Ferrulli Francesco - MAT: 716836 - f.ferrulli14@studenti.uniba.it

2 Introduzione progetto e Link

Il progetto consiste in un'analisi di recensioni di libri prese dallo store di Amazon Kindle. Il dataset su cui ho lavorato è disponibile su Kaggle al seguente link:

- `https://www.kaggle.com/datasets/meetnagadia/amazon-kindle-book-review-for-sent`
  `select=all_kindle_review+.csv`

Il progetto, insieme alla documentazione e presentazione, è scaricabile tramite il seguente link (Repository GitHub):

- `https://github.com/Ferru2000/SentimentAnalysis`

3 Ambiente di sviluppo

Il progetto è stato sviluppato in Python tramite l'IDE open-source Jupyter Notebook. Le librerie utilizzate per il progetto sono:

- **pandas:** per operazioni di lettura/scrittura sul dataset
- **matplotlib, seaborn:** per la visualizzazione di grafici
- **sklearn:** per l'implementazione di metodi di apprendimento
- **WordCloud, PIL** per la realizzazione del wordcloud

È stato scelto come IDE Jupyter Notebook per via della sua flessibilità di poter scrivere codice in appositie "celle" e di poterle eseguire separatamente secondo le proprie necessità. Jupyter notebook salva automaticamente i risultati ottenuti nei file .ipynb, detti notebook, in modo tale da poterli rivedere senza eseguire, ogni volta, il run del codice.

4 Dataset

Il dataset utilizzato contiene 12.000 righe di recensioni di libri. Il dataset contiene le seguenti 11 colonne:

- **asin:** ID della recensione
- **helpful:** quanto utile è stata la recensione
- **rating:** la valutazione complessiva in range [1,5]
- **reviewText:** recensione del libro
- **reviewTime:** momento della recensione
- **reviewerID:** ID del recensore

- **reviewerName:** nome del recensore

- **summary:** sommario della recensione

- **unixReviewTime:** unix timestamp

5 Preprocessing

Inizialmente ho applicato alcune operazioni di **dataset cleaning**. Come prima operazione ho rimosso alcune colonne che non erano utili allo scopo del progetto o che fornivano informazioni non rilevanti. Le colonne che ho eliminato sono:

- asin

- helpful

- reviewTime

- reviewerID

- reviewerName

- unixReviewTime

Dopodiché, ho effettuato un controllo su possibili valori mancanti e, in caso affermativo, la rimozione delle righe con valori mancanti. Successivamente ho creato una ulteriore colonna chiamata 'sentiment' per avere una distribuzione dei dati più uniforme. Tale colonna assume valore:

- 0: se il rating è compreso nell'intervallo [1,3]

- 1: se il rating è compreso nell'intervallo [4,5]

In seguito ho applicato alcune operazioni di **text cleaning** alle colonne 'reviewText' e 'summary'. Come prima operazione sono stati rimossi i segni di punteggiatura e le stopwords, in quanto irrilevanti per la classificazione. Poi è stato applicato il text lowering e la lemmatizzazione del testo, in modo tale da poter utilizzare il testo come feature per la classificazione.

Il file in questione è "Preprocessing.ipynb"

6 Integrazione della conoscenza

Effettuato il preprocessing del dataset, l'operazione successiva è stata quella di integrare la conoscenza. Questa operazione consiste nella creazione delle seguenti 4 colonne:

- **numberPositiveReview**: numero di parole positive nella colonna reviewText per ogni recensione

- **numberNegativeReview**: numero di parole negative nella colonna reviewText per ogni recensione

- **numberPositiveSummary**: numero di parole positive nella colonna summary per ogni recensione

- **numberNegativeSummary**: numero di parole negative nella colonna summary per ogni recensione

Per il calcolo del numero di parole positive e negative sono stati utilizzati due database esterni.
Il database che contiene la lista di parole positive è reperibile al seguente link:

– `https://ptrckprry.com/course/ssd/data/positive-words.txt`

Il database che contiene la lista di parole negative è reperibile al seguente link:

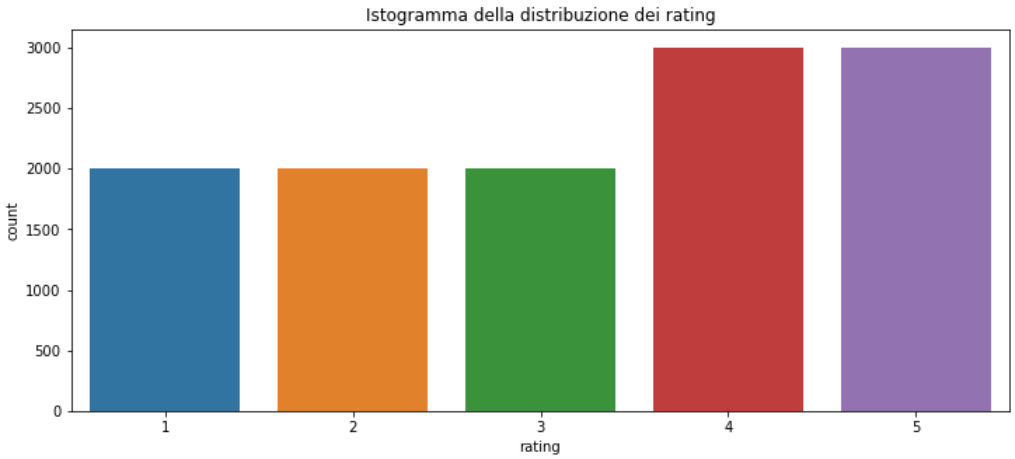
– `https://ptrckprry.com/course/ssd/data/negative-words.txt`

Per ogni parola in reviewText e summary viene effettuata la ricerca binaria per trovare una possibile corrispondenza con le liste di parole positive e negative. Ogni risultato ottenuto viene poi normalizzato sul numero di parole totali del reviewText/summary.

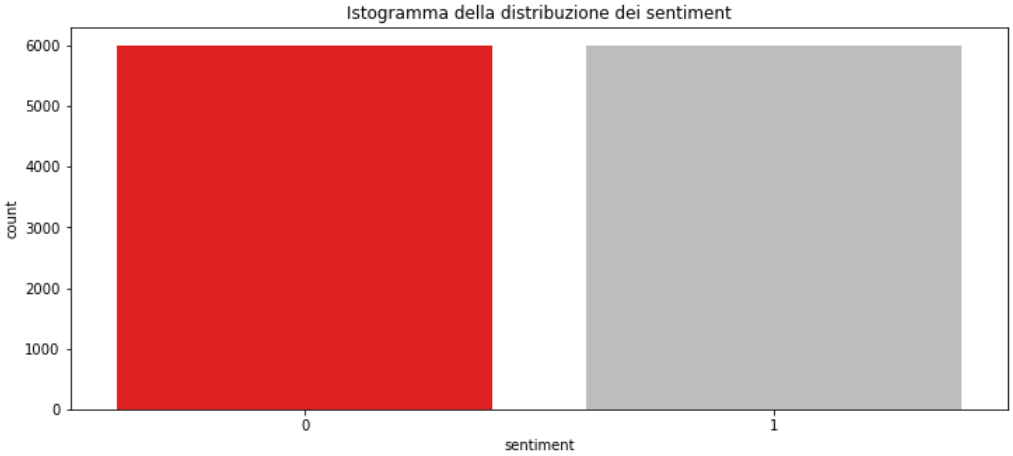
Il file in questione è "Integrazione_Conoscenza.ipynb"

7 Visualizzazione dei dati

Distribuzione dei valori della colonna rating:



Distribuzione dei valori della colonna sentiment dopo aver effettuato il raggruppamento:



8 WordCloud

Per visualizzare i termini più significativi presenti nelle recensioni è stato creato il WordCloud tramite la libreria WordCloud.

9 Apprendimento supervisionato

L'apprendimento supervisionato necessita di:

- Insieme di feature di input
- Insieme di feature di output
- Insieme di esempi suddiviso in:
  - training set, con i valori disponibili per tutte le feature
  - test set, con valori disponibili solo per le feature di input

L'obiettivo è quello di predire i valori delle feature di output degli esempi di test, addestrando il classificatore sugli esempi di training. Per l'apprendimento supervisionato sono stati implementati i seguenti classificatori:

1. **Random Forest**
2. **Multinomial Naive Bayes**
3. **K-Nearest Neighbours**

4. **Neural Network: Multi-layer Perceptron**

5. **Support Vector Machine**

Per ogni classificatore è stato applicato il **K-Fold Cross Validation** per avere una valutazione fair dei modelli. Il K-Fold Cross Validation suddivide i dati in k insiemi o fold. L'addestramento viene ripetuto k volte, ed ogni volta viene usato un fold diverso come validation set mentre gli altri k-1 fold vengono usati per il training.
Il valore K applicato ai classificatori prende valori nell'intervallo [10,15]. Per ragioni dovute a limiti computazionali, il valore K per i classificatori SVM e Neural Netword prende i valori in {5,10}.
L'apprendimento supervisonato applicato su questo dataset è diviso in due parti:

- Apprendimento supervisionato applicato alla colonna reviewText, che contiene la recensione completa, il cui file si trova in "Supervised/Supervised_Review"

- Apprendimento supervisionato applicato alla colonna summary, che contiene un riassunto della recensione, il cui file si trova in "Supervised/Supervised_Summary"

NOTA: per avere una valutazione equa, ogni classificatore è stato addestrato sullo stesso insieme di training e sullo stesso insieme di test, definendo un random seed uguale per tutti i classificatori. Per ogni classificatore, inoltre, sono stati calcolati accuratezza sull'insieme di training, accuratezza sull'insieme di test, precision e recall delle classi e matrice di confusione.

10 Apprendimento supervisionato - reviewText

Consideriamo ora l'apprendimento applicato alla recensione completa. Il file in questione si trova in "Supervised/Supervised_Review.ipynb".

10.1 Random Forest

Il Random Forest è un modello di apprendimento composito utilizzato per task di classificazione. Esso genera e addestra multipli alberi di decisione in fase di training per predire la classe di appartenenza di un esempio, utilizzando la moda per la predizione. Ho deciso di utilizzare il Random Forest piuttosto che un singolo albero di decisione perchè si ha un maggiore controllo sull'overfitting.

Come prima cosa ho calcolato la matrice TF-IDF delle recensioni (colonna reviewText), partendo dalle recensioni lemmatizzate, ottenendo così le feature di input. In aggiunta a queste feature di input, ho incluso le due colonne (normalizzate) che indicano la percentuale di parole positive e negative di ogni recensione (colonne numberPositiveReview e numberNegativeReview). Per quanto riguarda le feature target, si hanno due classi distinte identificate tramite numeri interi:

- Negativo = 0, indica una recensione negativa

- Positivo = 1, indica una recensione positiva

10.1.1 Random Forest: Feature Selection

Prima di eseguire il task di classificazione ho effettuato la **Feature Selection**. Avendo come feature di input le recensioni sottoforma di matrice TF-IDF, ho applicato la feature selection in modo da ridurre il numero di variabili e ridurre sia il costo computazionale sia la sua complessità. Da circa 41.000 feature di input, grazie alla feature selection, il numero di variabili è stato ridotto a circa 3.330 feature.

Queste feature, inoltre, verranno utilizzate anche per i successivi classificatori, invece di utilizzare le 41.000 feature.

10.1.2 Random Forest: Parameter Tuning

Il Random Forest è un modello che può risultare molto complesso e richiedere molto tempo. Per questo, ho effettuato il **Tuning dei parametri**, in modo tale da raggiungere la massima accuratezza con un numero appropriato di alberi e foglie.
Tralasciando i parametri più "tecnici" su cui è stato ottimizzato il Random Forest, la ricerca dei parametri migliori è stata effettuata sui seguenti valori:

- n_estimator: indica il numero di alberi, effettuato sui valori [40,60,80,100]
- min_sample_split: indica il numero minimo di esempi necessari allo split, effettuato sui valori [15, 30]
- min_sample_leaf: indica il numero minimo di esempi in ogni foglia, effettuato sui valori
  [5, 15, 30]

I parametri migliori restituiti dal GridSearch (ricerca esaustiva sui parmaetri) consistono in: numero di alberi = 100, numero di foglie = 5, numero di split = 30.

10.1.3 Random Forest: Classificazione

Dopo aver effettuato la feature selection e il tuning dei parametri, ho eseguito il task di classificazione con il Random Forest. Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.892 \qquad \text{accuracy\_test} = 0.798$$

I valori di precision e recall delle classi sono:

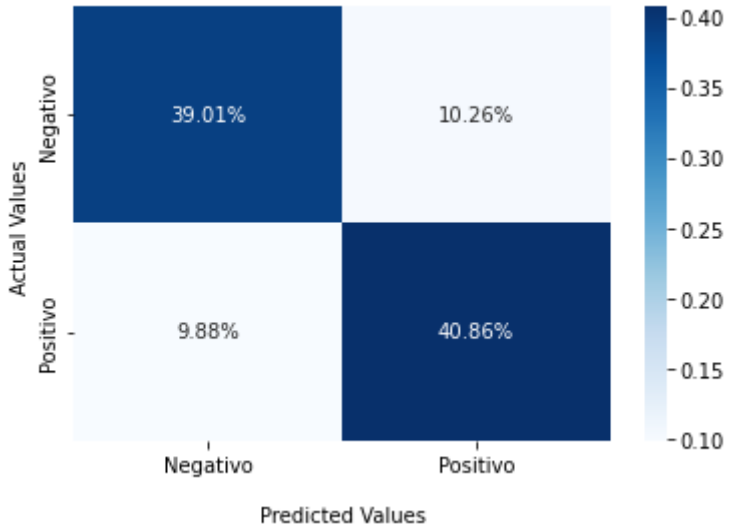
	Random Forest	
Classe	Precision	Recall
Negativo	0,8	0,79
Positivo	0,8	0,81

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,8027	0,0093
11	0,8026	0,006
12	0,8048	0,0075
13	0,8055	0,0164
14	0,8048	0,011
15	0,8025	0,0106

Il miglior risultato è stato ottenuto con K = 14 insiemi.

Matrice di confusione:



10.2 Multinomial Naive Bayes

Il classificatore Bayesiano è un classificatore probabilistico basato sull'applicazione del Teorema di Bayes, usando assunzioni di indipendenza tra le features (Naive). Il classificatore Multinomial Naive Bayes utilizza la distribuzione multinomiale, che generalizza la distribuzione binomiale. Nel modello multinomiale i campioni, vettori di feature, rappresentano le frequenze con cui determinati eventi sono stati generati da una distribuzione multinomiale di parametri $(p_i, \ldots, p_k)$.
Questo modello è tipicamente usato per la classificazione di docuementi, con eventi che rappresentano l'occorrenza di una parola in un singolo documento.
Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.861 \qquad \text{accuracy\_test} = 0.809$$

I valori di precision e recall delle classi sono:

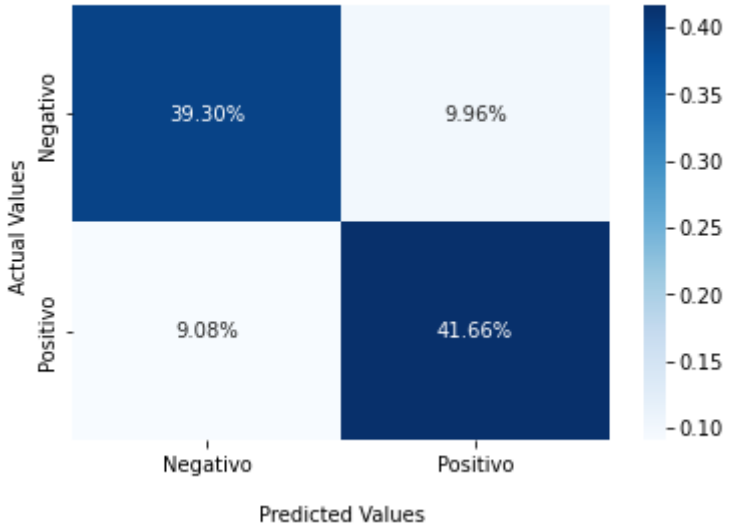
	Multinomial Naive Bayes	
Classe	Precision	Recall
Negativo	0,81	0,8
Positivo	0,81	0,82

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,821	0,0093
11	0,8216	0,0092
12	0,8223	0,0097
13	0,8218	0,0118
14	0,8201	0,0117
15	0,8214	0,0123

Il miglior risultato è stato ottenuto con K = 12 insiemi.

Matrice di confusione:

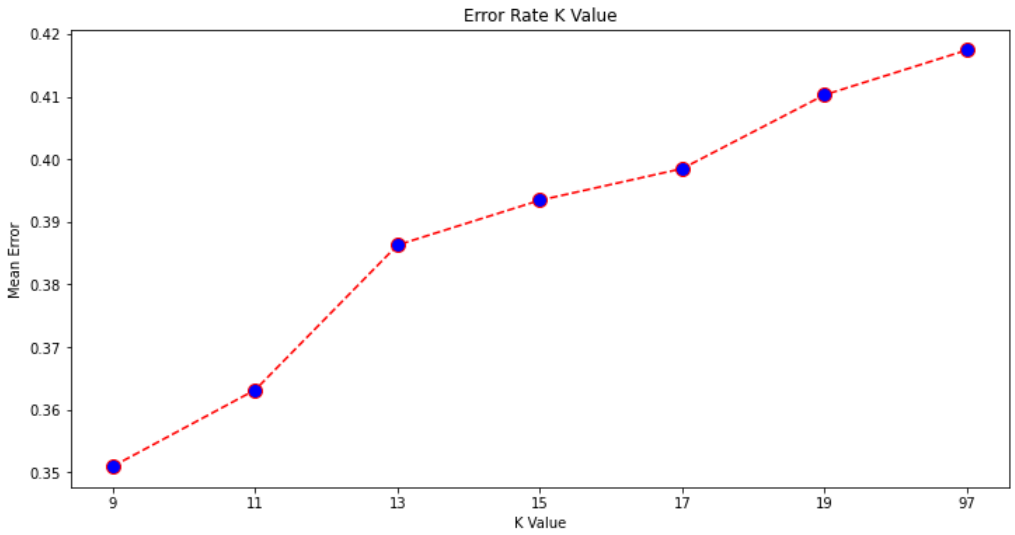


10.3 KNN

Il classificatore KNN è una modalità di apprendimento case-based, in cui gli esempi vengono immagazzinati e la predizione di un nuovo esempio si basa sul ritrovamento di casi o esempi simili. Questo tipo di apprendimento richiede poco lavoro offline, in quanto gli esempi devono essere solo memorizzati.
Il KNN richiede di definire un intero K, che indica il numero di vicini di un esempio da considerare per la predizione. Ho eseguito la classificazione con valori di K uguali

a: [9, 11, 13, 15, 17, 19, 97]. I numeri considerati sono tutti dispari, in modo tale da evitare situazioni di parità nella classificazione, inoltre è stato provata la classificazione con K = 97. Questo numero proviene dalla radice quadrata della dimensione del dataset.
Questo metodo, tuttavia, non ha prodotto buoni risultati per via della dimensionalità delle feature di input. Il KNN, infatti, ha un alto rischio di overfitting, producendo modelli inaccurati. Questo perchè, con molte dimensioni, è più difficile raggrupare i dati e molti esempi risultano equidistanti da ogni altro. Anche sfruttando la feature selection, questo metodo di apprendimento, su questo dataset, non va bene.

Il valore migliore di k per cui si è ottenuta la miglior accuratezza è k = 9.



Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 1.0 \qquad \text{accuracy\_test} = 0.649$$

I valori di precision e recall delle classi sono:

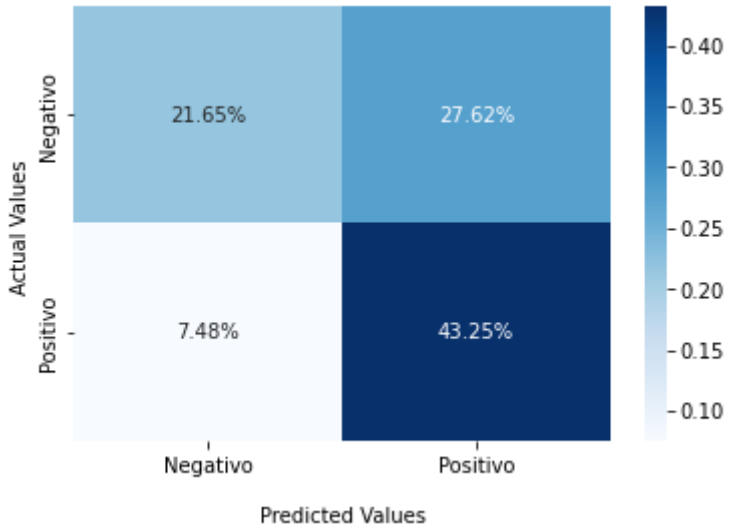
	KNN	
Classe	Precision	Recall
Negativo	0,74	0,44
Positivo	0,61	0,85

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,634	0,0229
11	0,6388	0,0284
12	0,6337	0,0284
13	0,6377	0,0306
14	0,6346	0,033
15	0,6393	0,0304

Il miglior risultato è stato ottenuto con K = 15 insiemi.

Matrice di confusione:



10.4 Rete Neurale Artificiale

Le reti neurali artificiali sono modelli di apprendimento ispirati ai neuroni cerebrali. Una rete neurale è composta da un layer che indica le variabili in input, un layer che rappresenta le predizioni degli output, e una serie di layer nascosti di feature non osservate ma utili alla predizione.
Ciascun nodo della rete si connette ai nodi del layer successivo, al cui arco viene associato un peso che determina l'importanza del valore e quanto esso influisca sull'output. Quando si passa da un layer ad un altro, viene eseguita la funzione di attivazione che genera l'input per il layer successivo.
La rete neurale effettua inoltre i passi di retropropagazione dell'errore o back-propagation, in cui si modificano i pesi delle connessioni, con discesa del gradiente, in modo tale da minimizzare una funzione di errore. Per la rete neurale è stata effettuata una ricerca dei parametri in modo tale da avere la massima accuratezza. Tra i parametri più importanti della rete neurale, si è riscontrato che la miglior funzione di attivazione è stata la Rectified Linear Unit, con 3 layer contenenti rispettivamente 200, 150, 100 unità.

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

accuracy_train = 1.0 accuracy_test = 0.799

I valori di precision e recall delle classi sono:

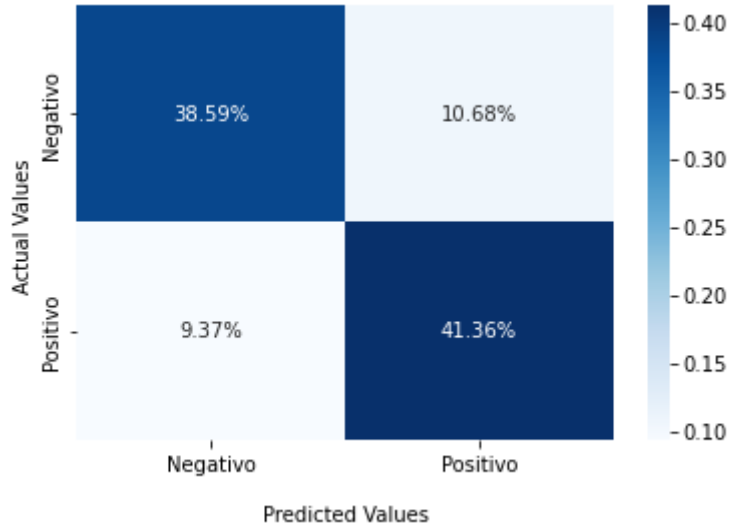
	Neural Network	
Classe	Precision	Recall
Negativo	0,8	0,78
Positivo	0,79	0,82

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
5	0,7957	0,0066
10	0,7991	0,0108

Il miglior risultato è stato ottenuto con K = 10 insiemi.

Matrice di confusione:



10.5 Support Vector Machine

Il Support Vector Machine è un classificatore basato su funzioni kernel, il quale costruisce un iperpiano detto superficie di decisione. Su questa superficie vengono rappresentati gli esempi come punti nello spazio, mappati in modo tale che gli esempi appartenenti a diverse categorie siano separati nello spazio.
L'SVM trova l'iperpiano di massimo margine, e gli esempi che sono più vicini alla superficie sono quelli che definiscono il supporto della superficie.
Questo classificatore è resistente all'overfitting scegliendo l'iperpiano di massimo margine, che massimizza la distanza minima dall'iperpiano al punto di training più vicino.
Tra i parametri su cui è stato ottimizzato l'SVM, si è ottenuto che la miglior funzione

kernel è quella lineare. È stata provata la funzione polinomiale e sigmoidale, ma si sono ottenuti scarsi risultati.

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.873 \qquad \text{accuracy\_test} = 0.820$$

I valori di precision e recall delle classi sono:

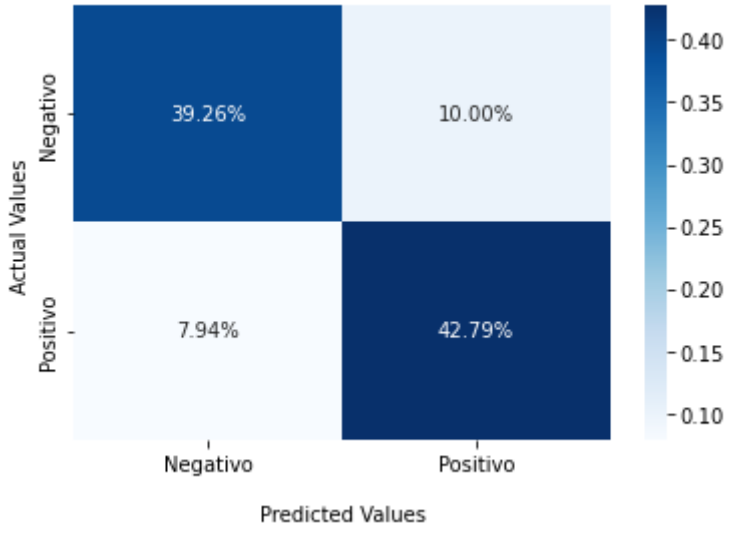
	Support Vector Machine	
Classe	Precision	Recall
Negativo	0,83	0,8
Positivo	0,81	0,84

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
5	0,8225	0,0032
10	0,8212	0,0084

Il miglior risultato è stato ottenuto con K = 5 insiemi.

Matrice di confusione:



10.6 Ensamble Learning

Dopo aver applicato i diversi metodi di apprendimento, è stato implementato l'Ensamble Learning, una tecnica di apprendimento che si basa sulle predizioni dei classificatori precedenti per poi effettuare la predizione finale come media. Combinando i vari classificatori si sono ottenuti risultati lievemente migliori, in termini di accuratezza e precision-recall.

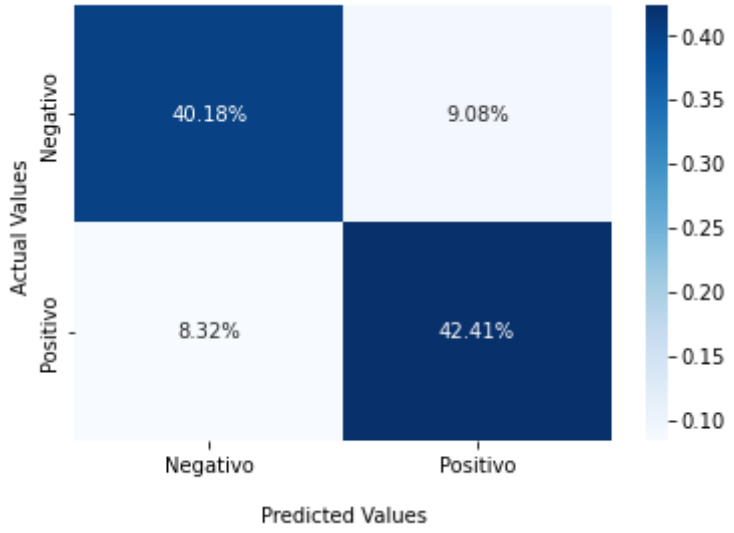
Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.895 \qquad \text{accuracy\_test} = 0.825$$

I valori di precision e recall delle classi sono:

	Ensamble Learning	
Classe	Precision	Recall
Negativo	0,83	0,82
Positivo	0,82	0,84

Matrice di confusione:



11 Apprendimento supervisionato - Summary

L'intero procedimento è stato ripetuto considerando la colonna summary, che contiene un riassunto, invece della colonna reviewText. Al posto delle features numberPositiveReview e numberNegativeReview sono state utilizzate le features numberPositiveSummary e numberNegativeSummary.
Il file in questione si trova in "Supervised/Supervised_Summary.ipynb".
Utilizzando come colonna un sommario, il numero di features è ridotto rispetto all'utilizzare la recensione completa. I classificatori dimostrano risultati lievemente peggiori, in quanto le features non sono molto distintive rispetto a quanto lo possono essere quelle della recensione completa. L'unica eccezione è stata il KNN, che ha prodotto risultati migliori data la presenza di meno features e diminuzione dell'overfitting.

11.1 Random Forest

11.1.1 Random Forest: Feature Selection

Anche in questo caso è stata effettuata la Feature Selection. Da circa 5.000 feature di input, grazie alla feature selection, il numero di variabili è stato ridotto a circa 959 feature.

11.1.2 Random Forest: Parameter Tuning

Il Random Forest è un modello che può risultare molto complesso e richiedere molto tempo. Per questo, ho effettuato il **Tuning dei parametri**, in modo tale da raggiungere la massima accuratezza con un numero appropriato di alberi e foglie.
Per quanto riguarda il parameter tuning, la ricerca dei parametri migliori è stata effettuata sui seguenti valori:

- n_estimator: indica il numero di alberi, effettuato sui valori [40,60,80,100]
- min_sample_split: indica il numero minimo di esempi necessari allo split, effettuato sui valori [15, 30]
- min_sample_leaf: indica il numero minimo di esempi in ogni foglia, effettuato sui valori
  [5, 15, 30]

I parametri migliori restituiti dal GridSearch (ricerca esaustiva sui parmaetri) consistono in: numero di alberi = 80, numero di foglie = 5, numero di split = 15.

11.1.3 Random Forest: Classificazione

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.725 \qquad \text{accuracy\_test} = 0.689$$

I valori di precision e recall delle classi sono:

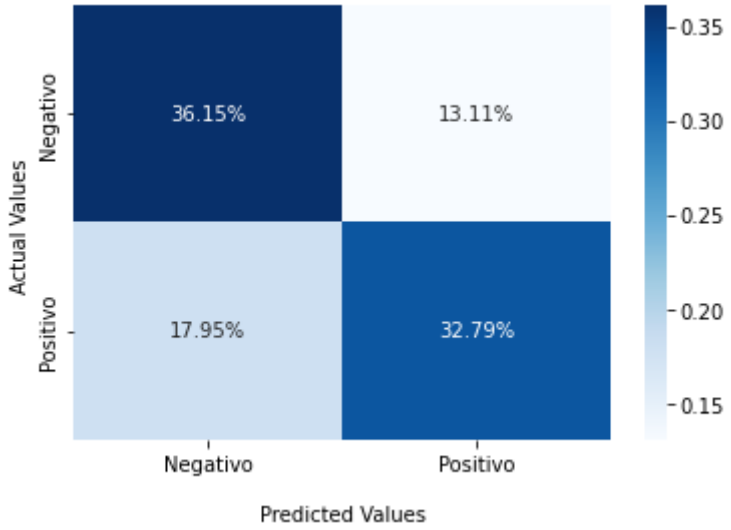
	Random Forest	
Classe	Precision	Recall
Negativo	0,67	0,73
Positivo	0,71	0,65

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,7069	0,0094
11	0,7082	0,0127
12	0,7044	0,012
13	0,7076	0,0144
14	0,708	0,0144
15	0,7081	0,0154

Il miglior risultato è stato ottenuto con K = 11 insiemi.

Matrice di confusione:



11.2 Multinomial Naive Bayes

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.766 \qquad \text{accuracy\_test} = 0.720$$

I valori di precision e recall delle classi sono:

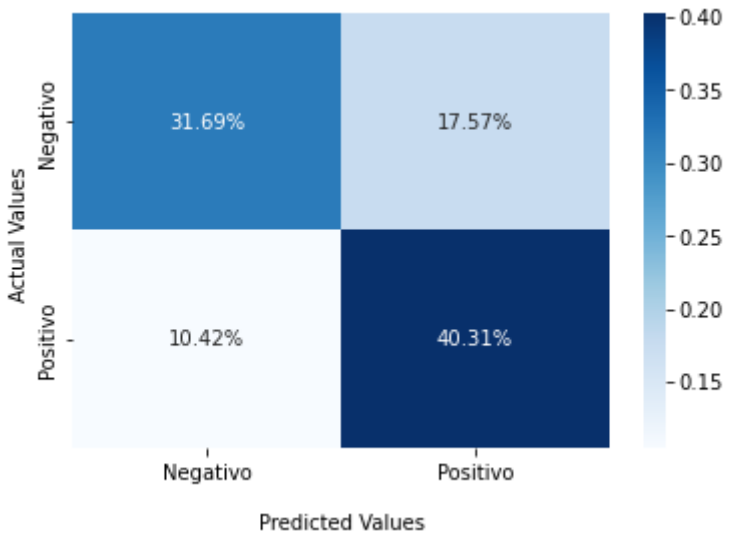
	Multinomial Naive Bayes	
Classe	Precision	Recall
Negativo	0,75	0,64
Positivo	0,7	0,79

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,7277	0,0111
11	0,7276	0,0147
12	0,729	0,0095
13	0,7285	0,0128
14	0,7276	0,0168
15	0,729	0,0145

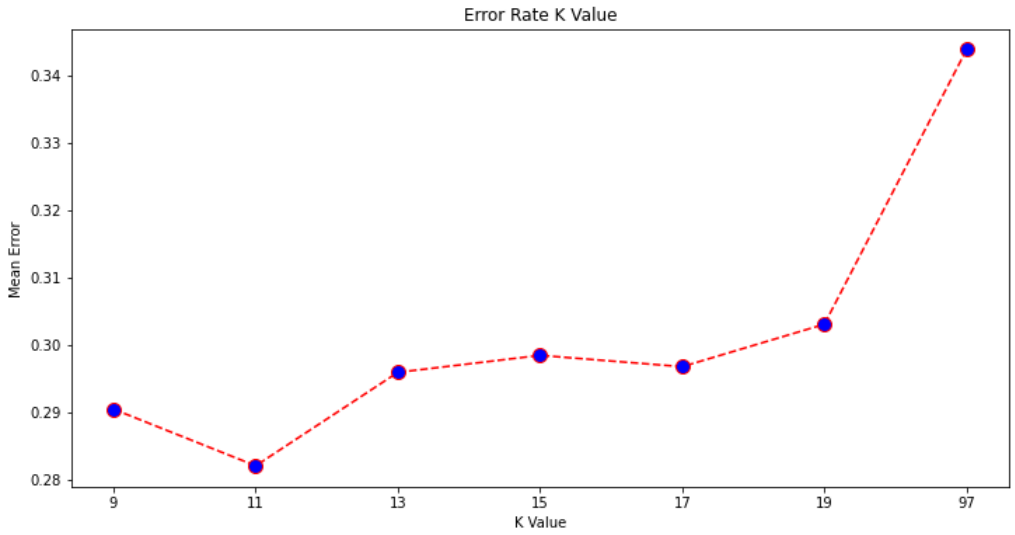
Il miglior risultato è stato ottenuto con K = 12 insiemi.

Matrice di confusione:



11.3 KNN

Il valore migliore di k per cui si è ottenuta la miglior accuratezza è k = 11.



Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.936 \qquad \text{accuracy\_test} = 0.724$$

I valori di precision e recall delle classi sono:

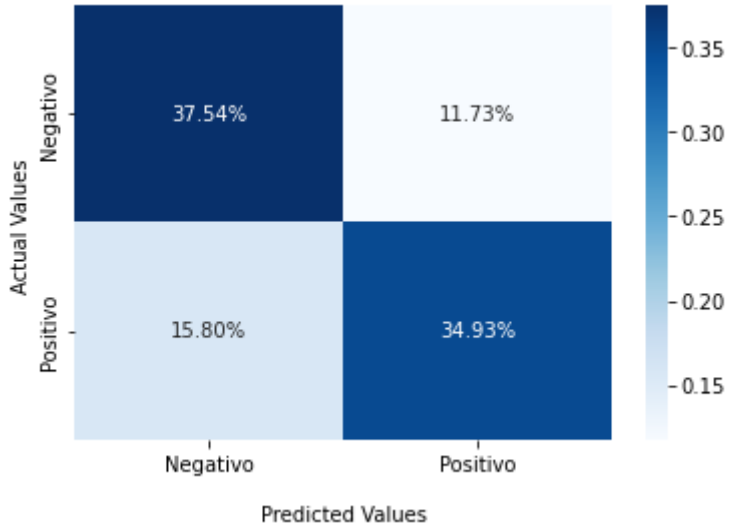
	KNN	
Classe	Precision	Recall
Negativo	0,7	0,76
Positivo	0,75	0,69

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
10	0,7316	0,0098
11	0,7322	0,0144
12	0,7329	0,0155
13	0,732	0,0121
14	0,7367	0,0161
15	0,7343	0,017

Il miglior risultato è stato ottenuto con K = 14 insiemi.

Matrice di confusione:



11.4 Rete Neurale Artificiale

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.932 \qquad \text{accuracy\_test} = 0.735$$

I valori di precision e recall delle classi sono:

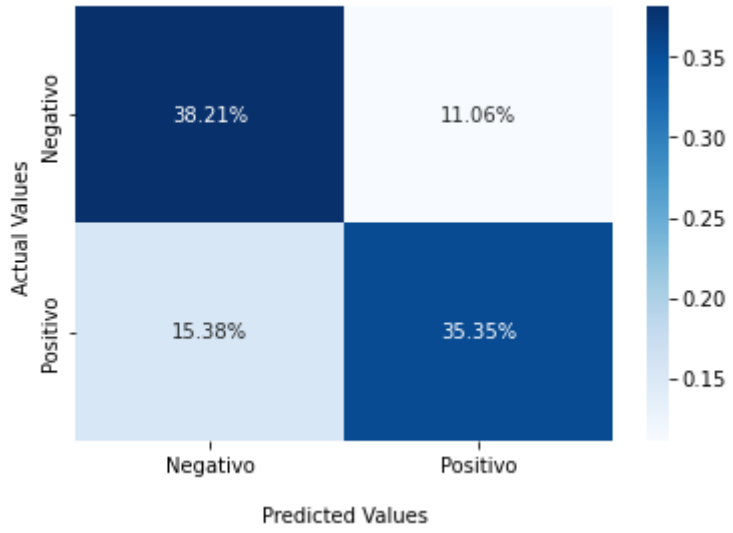
	Neural Network	
Classe	Precision	Recall
Negativo	0,71	0,78
Positivo	0,76	0,7

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
5	0,7537	0,0098
10	0,7554	0,0161

Il miglior risultato è stato ottenuto con K = 5 insiemi.

Matrice di confusione:



11.5 Support Vector Machine

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.798 \qquad \text{accuracy\_test} = 0.732$$

I valori di precision e recall delle classi sono:

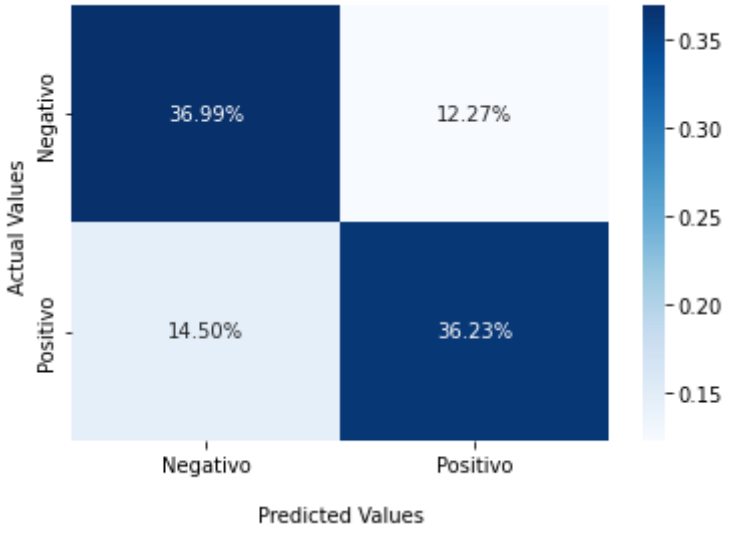
	Support Vector Machine	
Classe	Precision	Recall
Negativo	0,72	0,75
Positivo	0,75	0,71

Successivamente ho applicato il K-Fold Cross Validation e ho ottenuto i seguenti valori:

K	Avarage accuracy	Standard Deviation
5	0,7489	0,0055
10	0,7491	0,012

Il miglior risultato è stato ottenuto con K = 5 insiemi.

Matrice di confusione:



11.6 Ensamble Learning

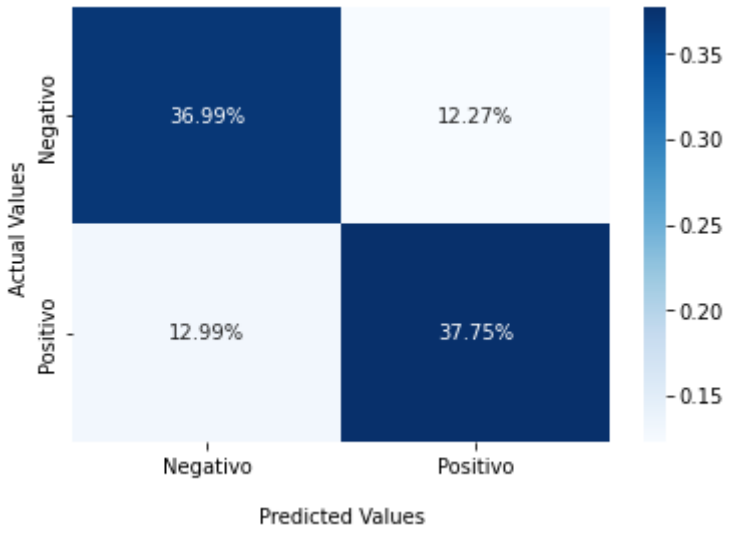
Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.897 \qquad \text{accuracy\_test} = 0.747$$

I valori di precision e recall delle classi sono:

	Ensamble Learning	
Classe	Precision	Recall
Negativo	0,74	0,75
Positivo	0,75	0,74

Matrice di confusione:



12 Apprendimento non supervisionato

L'apprendimento supervisionato è una tecnica di apprendimento in cui i dati forniti non sono etichettati, e l'obiettivo è quello di classificare i dati cercando dei modelli nascosti e relazioni esistenti tra i dati.

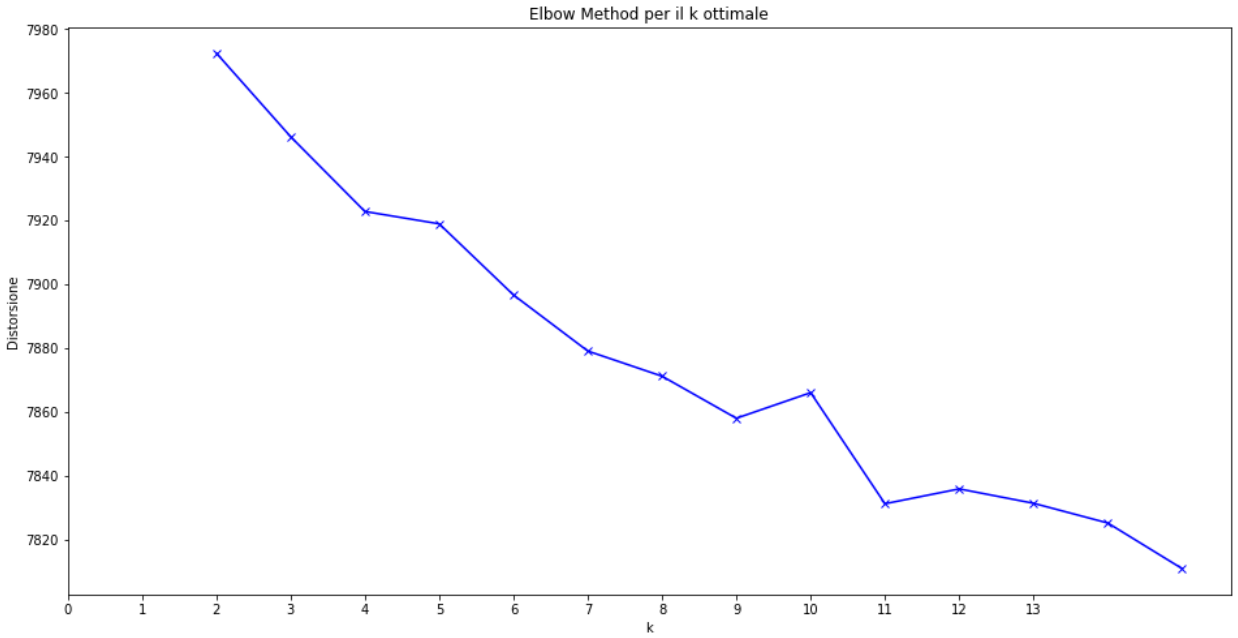
Il file che contiene l'apprendimento non supervisionato si trova in "Unsupervised/Unsupervised.ipynb".

12.1 K-Means

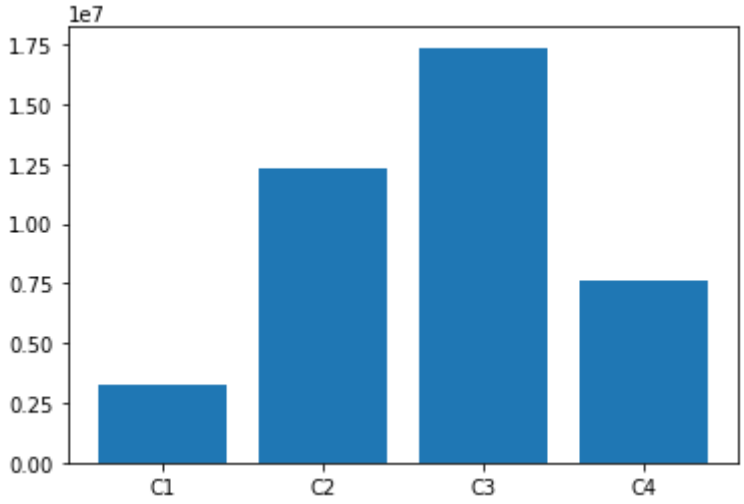
L'algoritmo K-Means è un algoritmo di apprendimento non supervisionato per l'hard clustering, in cui ad ogni esempio viene assegnata una classe specifica. Il primo passo di questo algoritmo consiste nel scegliere il valore di K, che indica il numero di classi o cluster in cui si vogliono suddividere i dati. Si scelgono poi, in modo casuale, K centroidi, considerabile come punto medio di un cluster. Si calcola la distanza di ogni punto (esempio) da ogni centroide, e si associa ad ogni esempio il cluster la cui distanza dal centroide è minima. Si ricalcolano poi i centroidi di ogni cluster. Questo procedimento viene iterato fino alla convergenza, cioè quando non si ottiene alcun cambiamento.

L'obiettivo è stato quello di vedere, attraverso la creazione dei cluster basati sulle descrizioni, se fosse possibile estrarre nuove feature per la classificazione mediante SVM e Rete Neurale.

Per la scelta del valore K è stato utilizzato l' **Elbow Method**, in cui si itera il K-Means per diversi valori di K calcolando la somma delle distanze al quadrato tra ogni centroide e i vari punti. Il risultato migliore ottenuto iterando il K-Means per valori in range [2,15] è stato K = 4.



La distribuzione dei 4 cluster identificati è la seguente:



Le features più importanti di ogni centroide sono:

- Centroide Cluster 0: ['love', 'end', 'get', 'really', 'like', 'good', 'great', 'character', 'little', 'read', 'story', 'write', 'well', 'much', 'would', 'one', 'author', 'book', 'short', 'enjoy']
- Centroide Cluster 1: ['like', 'story', 'read', 'book', 'really', 'would', 'character', 'one', 'get', 'go', 'end', 'time', 'want', 'sex', 'much', 'know', 'think', 'love', 'good', 'make']
- Centroide Cluster 2: ['time', 'kindle', 'story', 'find', 'love', 'read', 'book', 'get', 'well', 'author', 'character', 'like', 'good', 'write', 'one', 'would', 'great', 'much', 'make', 'enjoy']
- Centroide Cluster 3: ['read', 'great', 'book', 'love', 'good', 'series', 'character', 'enjoy', 'story', 'like', 'really', 'would', 'author', 'one', 'recommend', 'write', 'well', 'get', 'look', 'keep']

12.2 Classificazione SVM con classi derivanti dall'output del K-Means

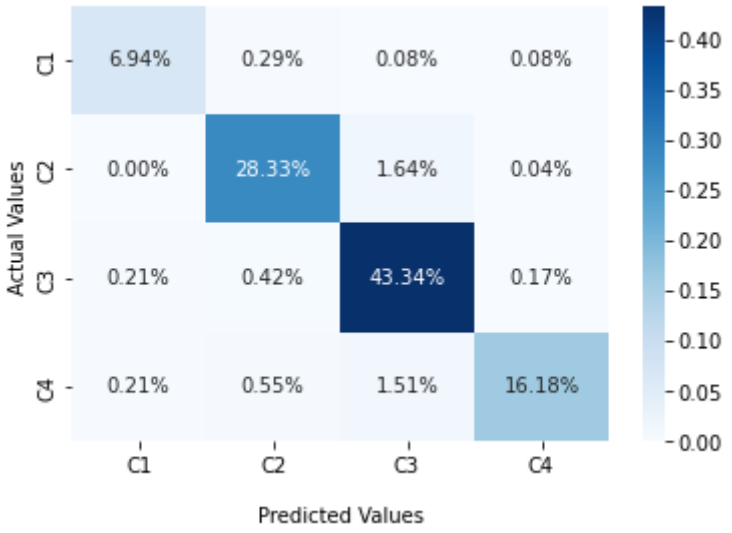
Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 0.984 \qquad \text{accuracy\_test} = 0.947$$

Il K-Fold Cross Validation ottenuto ha i seguenti valori:

	Classificazione SVM	
K	Avarage accuracy	Standard Deviation
5	0,9405	0,0027
10	0,9414	0,0067

Matrice di confusione:



12.3 Classificazione Neural Network con classi derivanti dall'output del K-Means

Per quanto riguarda l'accuratezza sui training e test set, si sono ottenuti i seguenti valori:

$$\text{accuracy\_train} = 1.0 \qquad \text{accuracy\_test} = 0.890$$

Il K-Fold Cross Validation ottenuto ha i seguenti valori:

	Classificazione Neural Network	
K	Avarage accuracy	Standard Deviation
5	0,8794	0,0104
10	0,8856	0,0128

Matrice di confusione: