# Università degli studi di Milano - Bicocca

# Text Mining and Search

# Final Project

20 Newsgroups Classification

Autori:

Corrado Montoro - 841489 - c.montoro@campus.unimib.it

Luca Lazzati - 850334 - l.lazzati2@campus.unimib.it



## Sommario

Il dataset utilizzato è "20 news group" ,una raccolta di circa 20.000 documenti di notizie, suddivise in 20 diverse categorie. Il Dataset è stato spesso utilizzato per applicazioni di Machine Learning, come Text Classification e Text Clustering. E' stato interessante dunque capire, a partire semplicemente dal testo, quale sia la categoria di appartenenza di una notizia e successivamente individuare i topic modelling che permettono di identificare i contenuti del discorso(Topic) presenti all’interno di una collezione di documenti.

Come prima fase del progetto si è fatto il parsing dei file, perchè i vari documenti erano posti in diverse cartelle(identificati come “Target” nel nostro progetto), quindi si è provveduto all’estrazione dei file con la classe associata.

Successivamente si è controllato il testo facendo prima una fase di Esplorazione e poi di Preprocessing. Dopo di che il dataset è stato splittato in train set e test set e successivamente sono state create le matrici (tf ,tf-idf e binary ) da utilizzare negli algoritmi di Machine Learning. Prima della modellazione dei classificatori però c’è stata una fase di Feature Selection, questo ha permesso da eliminare i features irrilevanti e quindi migliorare le performance dei modelli. Le metriche di valutazione sono state applicate sia al train set sia al test set in modo tale da valutare la bontà del modello finale.

Infine si è visualizzato, per ogni matrice con i relativi pesi e token, i vari Topic Modelling.

# 1 Introduzione

La classificazione di documenti testuali è uno dei tipici compiti dell'apprendimento automatico supervisionato (Supervised Machine Learning). Il dataset "20 Newsgroups" è spesso utilizzato per lo svolgimento di questa tipologia di task. E’ interessante dunque capire, a partire semplicemente dal testo, quale sia la categoria di appartenenza di una notizia e anche identificare i contenuti del discorso. Invece la classificazione non supervisionata (Unsupervised Machine Learning) servirà per identificare i topic più rilevanti in una collezione di documenti.

# 1.1 Obiettivo

Gli obiettivi del progetto sono essenzialmente due:

* Il primo punto è stato quello di classificare correttamente le notizie sulla base del loro contenuto, assegnando a ciascuna la corretta categoria di appartenenza e valutare che la classificazione delle news sia stata svolta correttamente e quindi il risultato finale sarà dunque il calcolo del miglior classificatore possibile. E' stato scelto, per il progetto, un task di Text Classification.
* Il secondo punto è stato quello di utilizzare un classificatore non supervisionato come il Topic Modelling(LDA model) per identificare, i contenuti del discorso presenti all’interno di una collezione di documenti. Il calcolo è stato fatto su tutte le matrici con il loro peso e token differenti(quindi si vedranno 12 Topic Modelling). In questo caso è stato scelto, per il progetto, un task di Unsupervised Text Classification.

# 

# 2 Dataset

Il dataset “20 newsgroups", è una raccolta (in formato .Rar da estrarre) di documenti, suddivisi in 20 diverse categorie che sono:

alt.atheism (0)

comp.graphics (1)

comp.os.ms-windows.misc (2)

comp.sys.ibm.pc.hardware (3)

comp.sys.mac.hardware (4)

comp.windows.x (5)

misc.forsale (6)

rec.autos (7)

rec.motorcycles (8)

rec.sport.baseball (9)

rec.sport.hockey (10)

sci.crypt (11)

sci.electronics (12)

sci.med (13)

sci.space (14)

soc.religion.christian (15)

talk.politics.guns (16)

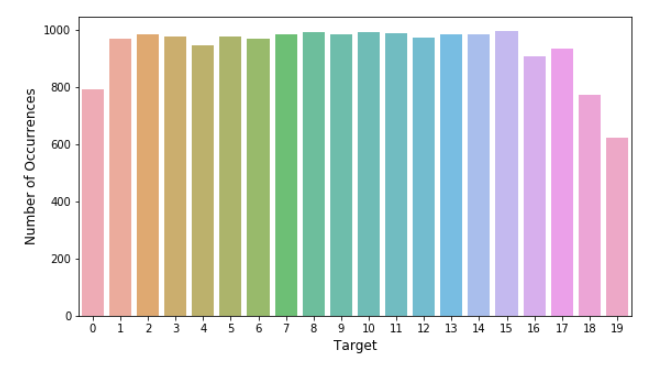
talk.politics.mideast (17)

talk.politics.misc (18)

talk.religion.misc (19)

Ogni record presente è in realtà un file di testo in inglese, che presenta la seguente struttura: metadati - intestazione - testo del documento.

Il dataset e stato splittato in training set (80%) e test set (20%) per l'applicazione dei modelli di Text Classification e successivamente per il Text Classification Unsupervised.



# 3 Text Preprocessing

Nella fase di preprocessing del testo sono state svolte le seguenti operazioni:

* testo in minuscolo
* eliminazione indirizzi mail
* eliminazione dei numeri
* eliminazione caratteri di punteggiatura
* rimozione parole composte da una e due lettere
* sistemazione degli spazi creati con le precedenti operazioni
* tokenizzazione
* rimozione stop words
* lemmatizzazione
* stemmatizzazione

# 4 Text Representation

La Text Representation è stata effettuata misurando i pesi associati a ciascun

termine attraverso tre euristiche:

* Term Frequency (TF): la Term Frequency del termine t nel documento d è definita come il numero di volte che t si verifica in d.
* Binary Term Frequency: Se True, tutti i conteggi diversi da zero sono impostati su 1. E’ utile per modelli probabilistici discreti che modellano eventi binari anziché conteggi interi.
* Term Frequency - Inverse Document Frequency (TF-IDF): il peso tf-idf

di un termine e il prodotto del suo peso tf e del suo peso idf.

Tutte e tre le matrici sono stata create per unigram e per bigram.

Il risultato di questa fase è la creazione di ventiquattro matrici (dodici per il train set e dodici per il test set), infatti le euristiche elencate sono state applicate separatamente ai dati lemmatizzati e stemmatizzati.

# 5 Feature Selection

Per una migliore classificazione del testo viene utilizzato il test del chi-quadro, dove si ottiene una feature selection e si tengono solo le prime 5000 features (per ogni matrice) ordinate in ordine decrescente in base al valore del test.

L'operazione di feature selection viene giustificata in quanto:

* Si riduce il rischio di overfitting.
* Eliminazione degli features irrilevanti e quindi potenzialmente migliora le performance dei modelli.
* Permette di addestrare modelli utilizzando minor sforzo computazionale.

Successivamente le matrici verranno salvate in un dizionario, dove la chiave corrisponde al nome della matrice e il valore ad una tupla contenente la matrice di train e quella di test corrispondenti alla chiave.

# 6 Text Classification

Per la Text Classification sono stati testati tre modelli su ogni rappresentazione precedentemente descritta, al fine di stabilire quale combinazione rappresentazione-modello fornisca la migliore classificazione.

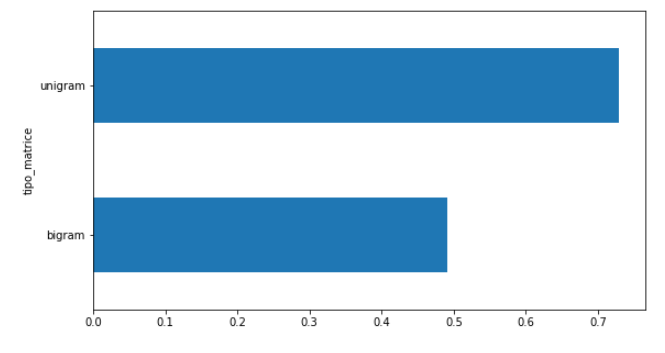
Nella valutazione dei risultati ottenuti è stata presa in considerazione l'accuracy, ossia la percentuale di classificazioni corrette.

L'accuratezza è stata valutata sia sul train set sia sul test set.

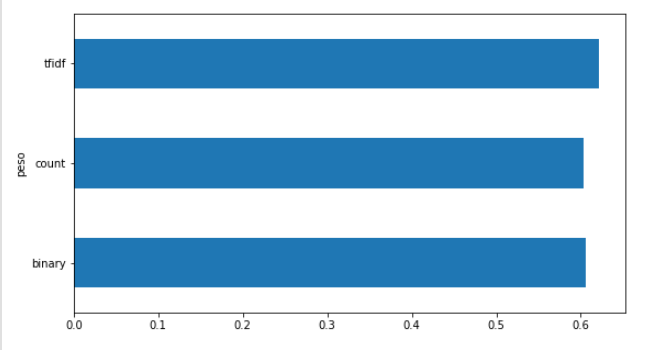
# 7 Risultati Text Classification

Prima di verificare la miglior classificazione sul testo,ossia la bontà del modello di classificazione sia sul train che sul test set, si sono visualizzati le performance dei vari tipi di matrice, peso e classificazione.

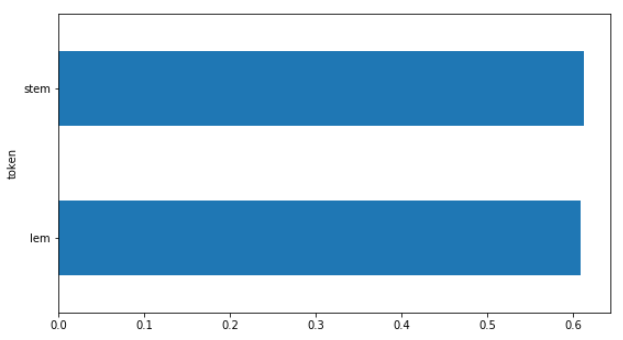
* Il tipo di rappresentazione mediamente più performante è Unigram:



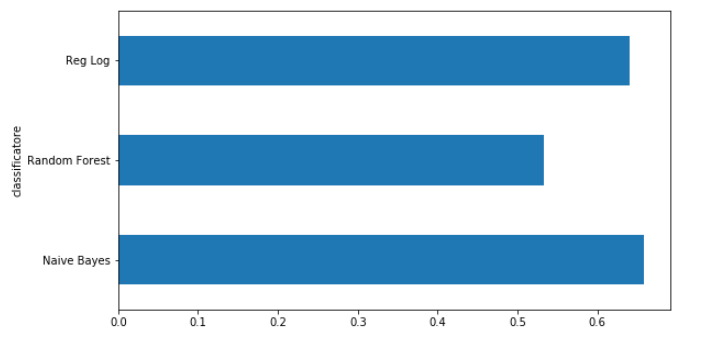
* Il peso mediamente più performante è il tf-idf:



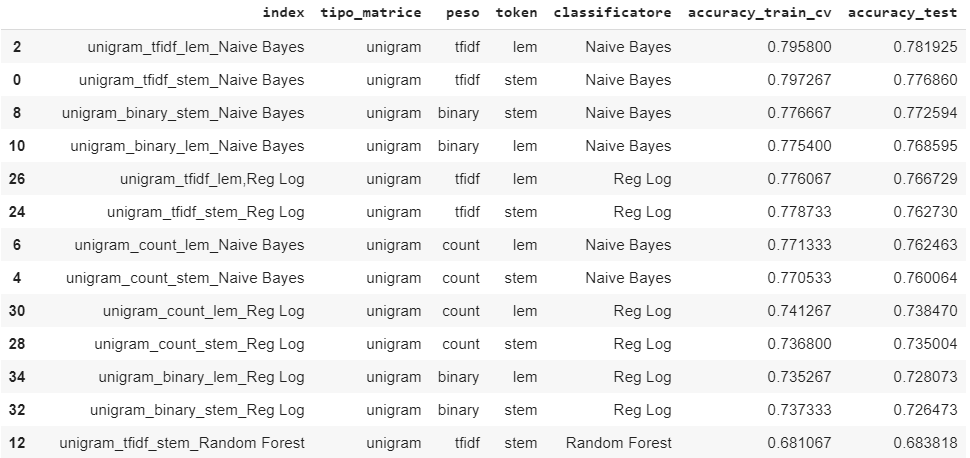
* Mediamente l'operazione di stemming è il più performante(anche se di poco):



* Il classificatore mediamente più performante è:



* Infine, tramite tabella, si può notare quale classificatore(in base al tipo matrice, peso, token e accuratezza) sia il migliore e si può visualizzare che sia Naive Bayes sia la Regressione Logistica siano molto più performanti del Random Forest.



# 8 Topic Modelling

Per classificare i target(argomenti) sulla base della testo, si è utilizzato il Topic Modelling che è un processo non supervisionato che permette l'identificazione dei contenuti del discorso (Topic) presenti all'interno di una collezione di documenti.

Permette quindi di determinare quali documenti trattano di uno stesso argomento (da dedurre) raggruppandoli in cluster.

L’algoritmo di Topic Modelling utilizzato è l’LDA, che determina la probabilità di appartenenza di un documento ad un Topic.

E' basato su due considerazioni:

* ogni documento è una distribuzione di probabilità sui topic
* ogni topic è una distribuzione di probabilità sulle parole

I documenti appartenenti ad uno stesso topic avranno gruppi di parole comuni.

L’algoritmo:

* Assegna casualmente ogni parola in ogni documento ad un Topic

Per ogni documento, per ogni topic:

* calcola la probabilità p(topic t | documento d)
* calcola la probabilità p(word w | topic t)
* moltiplica p(topic t | documento d) \* p(word w | topic t) assegnando la parola ad un topic sulla base di questa probabilità

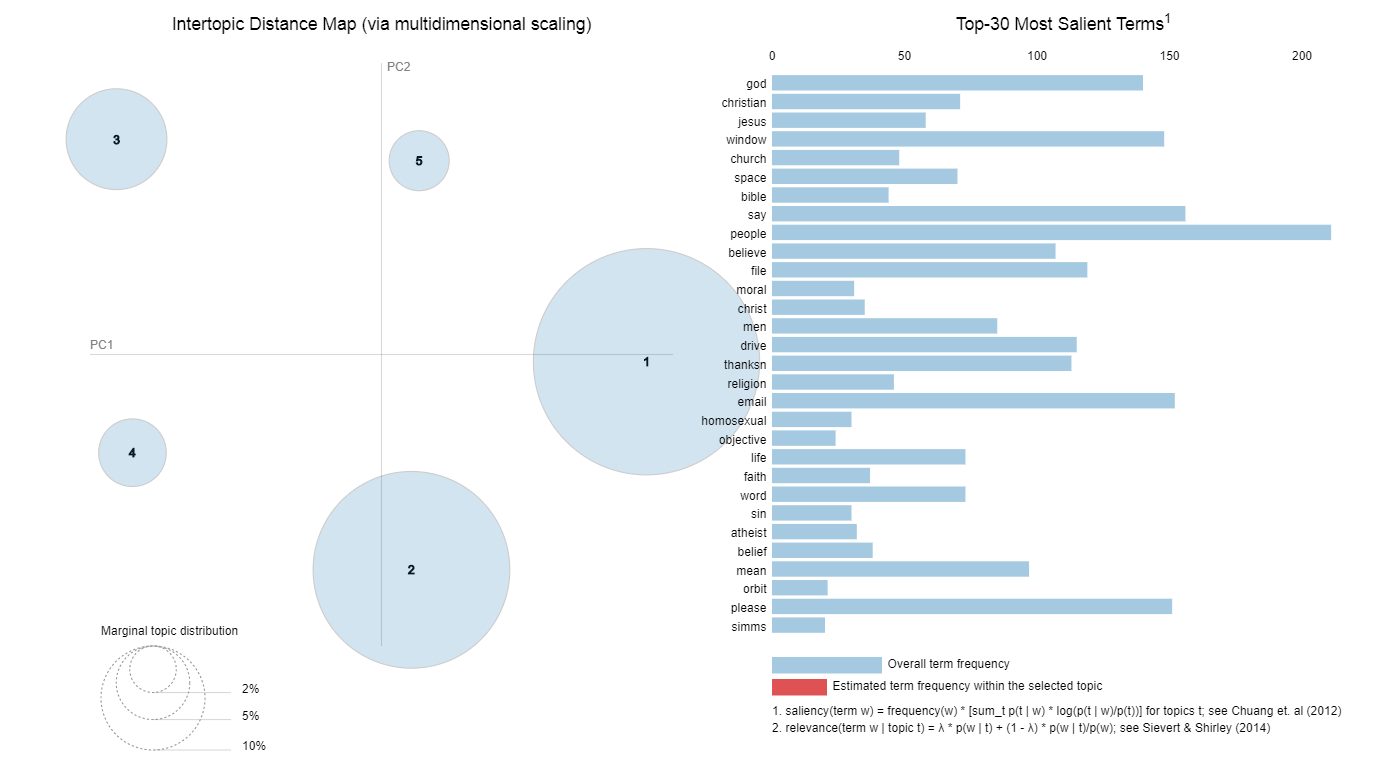
# 9 Analisi Topic Modelling

La prima fase è stato riprendere tutte le 12 matrici create in precedenza con pesi e token differenti, stimare il numero ottimale di Topic(da 5 a 30) e poi visualizzare il risultato. Per definire il numero ottimale di Topic si sono utilizzati diversi attributi della libreria “GridSearchCV” quali:

* “Best\_estimator\_”: Stima del miglior modello(stima il punteggio più alto sui dati)
* “Best\_params\_”: Impostazione dei parametri che ha dato i migliori risultati sui dati di controllo(ovvero sulla inizializzazione del GridSearchCV)
* “Best\_score\_” : Punteggio medio del best\_estimator
* “Perplexity”: Calcola perplessità approssimativa per i dati X. La perplessità è definita come exp (-1. \* Probabilità logaritmica per parola)

Verrà utilizzato questo procedimento per tutte le matrici prese in considerazione.

Esempio di Visualizzazione di Topic Modelling(LDA Model Unigram - tf-idf - lem):



# 10 Considerazioni Topic Modelling

In base a tutte le visualizzazioni dei Topic, si è concluso che i documenti che trattano uno stesso argomento e sono maggiormente “identificabili” rispetto ad altri Topic, sono gli LDA Model Unigram- Tf-Idf - Lem/Stem.

# 11 Conclusioni

Il modello di Text Classification migliore, in termini di accuratezza, risulta essere il Naive Bayes nella rappresentazioni Unigram- Tf-Idf-Lem e ci si aspetta dunque che, analizzando una qualsiasi news, essa venga classificata correttamente all'interno della categoria di appartenenza con un'elevata probabilità.

Infine il modello di classificazione di Text Clustering migliore, in termini di Topic, risulta essere LDA Model Unigram- Tf-Idf - Lem/Stem.