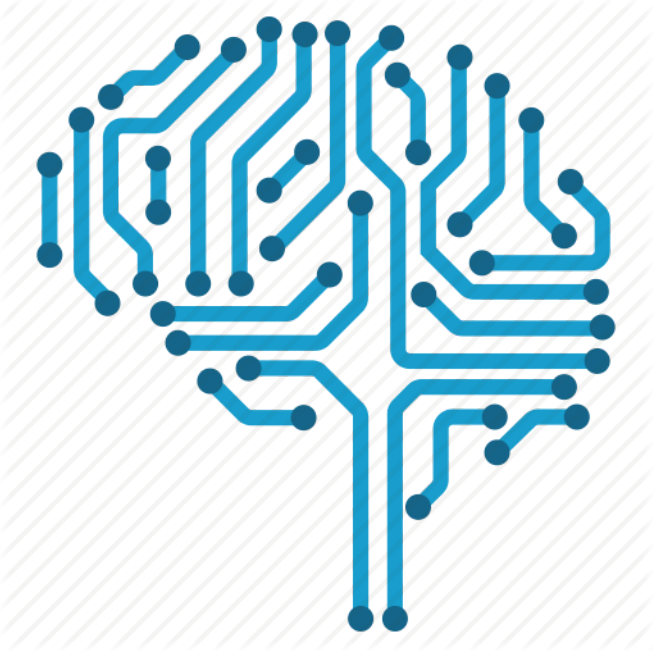
Big Data e Business Intelligence



Machine Learning

Giulio Angiani - UniPr

Machine Learning
Text Mining

Text Mining

- Processo di estrazione informazioni dai testi
- Dove si usa
  - Social Network
  - Siti E-commerce
  - Ovunque siano presenti testi non strutturati
- Scopi principali
  - Text categorization
  - Text clustering
  - Estrazione di concetti e di entità
  - Individuazione di relazioni fra entità
  - Individuazione di tassonomie*
  - Sentiment analysis

*In Scienze naturali: studio della teoria e delle regole di classificazione.
In questo contesto indica la ricerca di elementi caratteristici dei vari contenuti

Text Mining - Estrazione delle informazioni

Information Extraction

- parte del NLP (Natural Language Processing)
- recupero informazioni **strutturate** da documenti **non strutturati**

Problema:

- I documenti sono espressi in linguaggio naturale
- Sintassi non strutturate

Esempi d'uso:

- Di cosa parla quella recensione?
- Questa email contiene dati di un appuntamento importante o è spam?
- Devo leggere questo documento perché interessa la mia azienda o posso cestinarlo?

Text Mining - In azienda

Customer feedback analysis

- Categorizzazione automatica dei feedback
  - spesso anche in sottocategorie
  - analisi della positività o negatività di una recensione
- Individuazione dell'argomento del feedback
  - si parla di un certo prodotto?
  - è una recensione al customer service?
- Miglioramento rapporto col cliente
  - implementazione sezione FAQ
  - aggiunta di funzionalità ai prodotti

Text Mining - come fare?

- Abbiamo un testo in linguaggio naturale da analizzare
- Vogliamo capire se parla di un argomento o di un altro
- Problema di **classificazione**
  - di cosa si parla?
  - c'è un sentimento espresso ?
    - è positivo o negativo ?
  - rientra nelle categorie di interesse ?

Text Mining - come fare?

- Problema di **classificazione**
- Costruzione del **dataset**
- **Labeling** delle istanze
- Scelta di un **algoritmo** adeguato
- **Addestramento** del classificatore
- **Valutazione** delle prestazioni

Text Mining - primo esempio

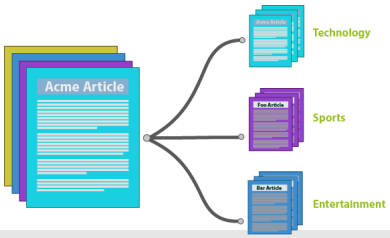
Abbiamo le seguenti frasi:


```
Oggi c'è un bel sole, è caldo, si sta bene al mare
Ho voglia di acqua e di mare
Che freddo con questa neve! si sta bene sotto le coperte
Le previsioni danno ghiaccio è neve
Mi piace la vendemmia... Il marrone delle foglie...
Le foglie stanno cadendo tutte... è proprio autunno
Domenica, se fa caldo, faremo un bel picnic in mezzo ai fiori
In questa stagione è bello camminare anche sotto le stelle
```

- categorie: stagioni

Text Mining - primo esempio

Labeling




```
Oggi c'è un bel sole, è caldo, si sta bene al mare : ESTATE
Ho voglia di acqua e di mare : ESTATE
Che freddo con questa neve! si sta bene sotto le coperte : INVERNO
Le previsioni danno ghiaccio è neve : INVERNO
Mi piace la vendemmia... Il marrone delle foglie... : AUTUNNO
Le foglie stanno cadendo tutte... è proprio autunno : AUTUNNO
In questa stagione è bello camminare anche sotto le stelle : PRIMAVERA
Domenica, se fa caldo, faremo un bel picnic in mezzo ai fiori : PRIMAVERA
```

Text Mining - con python

Librerie python avanzate per la data-analysis

Pandas

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language

https://pandas.pydata.org/

Numpy

NumPy is the fundamental package for scientific computing with Python

https://numpy.org/

Text Mining - con python

DATASET: FILE DI TESTO 'DATASET.CSV'

```
text,class
"oggi c'è un bel sole, è caldo, si sta bene al mare", ESTATE
"Ho voglia di acqua e di mare", ESTATE
"Che freddo con questa neve! si sta bene sotto le coperte", INVERNO
"Le previsioni danno ghiaccio è neve", INVERNO
"Mi piace la vendemmia... Il marrone delle foglie...", AUTUNNO
"Le foglie stanno cadendo tutte... è proprio autunno", AUTUNNO
"Domenica, se fa caldo, faremo un bel picnic in mezzo ai fiori", PRIMAVERA
"In questa stagione è bello camminare anche sotto le stelle ", PRIMAVERA
```

Text Mining - con python

PYTHON

```python
import pandas as pd
import numpy as np
from io import StringIO

data = pd.read_csv("dataset.csv", ",")
print(data)
```

OUTPUT

```
                                                text       class
0  oggi c'è un bel sole, è caldo, si sta bene al ...      ESTATE
1                      Ho voglia di acqua e di mare      ESTATE
2  Che freddo con questa neve! si sta bene sotto ...     INVERNO
3                Le previsioni danno ghiaccio è neve     INVERNO
4  Mi piace la vendemmia... Il marrone delle fogl...     AUTUNNO
5  Le foglie stanno cadendo tutte... è proprio au...     AUTUNNO
6  Domenica, se fa caldo, faremo un bel picnic in...   PRIMAVERA
7  In questa stagione è bello camminare anche sot...   PRIMAVERA
```

Text Mining - Vettorizzazione

Problema:

- Gli algoritmi lavorano con features numeriche (intere o float)
- Le istanze del dataset sono stringhe

Soluzione:

- Vettorizzazione e Tokenizzazione
  - Tokenizzazione
    - ogni stringa è rappresentata come insieme di **token** (parole)
  - Vettorizzazione
    - Sia **S** l'insieme **senza ripetizioni** di tutti i token del dataset
    - Ogni elemento $s_i \in S$ rappresenti una dimensione di un vettore di booleani
    - Ogni istanza del dataset è rappresentata con un **vettore di booleani** (1 se il token è presente, 0 altrimenti)

Text Mining - Vettorizzazione

Esempio:

- sia dataset **D** = {"oggi è mercoledi", "oggi non piove", "se piove prendo un ombrello"}
- sia **S** = ['mercoledi', 'non', 'oggi', 'ombrello', 'piove', 'prendo', 'se', 'un', 'è']

Rappresentazione

	mercoledi	non	oggi	ombrello	piove	prendo	se	un	è
oggi è mercoledi	1	0	1	0	0	0	0	0	1
oggi non piove	0	1	1	0	1	0	0	0	0
se piove prendo un ombrello	0	0	0	1	1	1	1	1	0

Vettorizzazione - con python

PYTHON

```python
# libreria sklearn per tokenization e vectorization
from sklearn.feature_extraction.text import CountVectorizer
# separiamo features dalle classi
X = data["text"]    # in questo momento abbiamo ancora una sola feature
print(X)
```

OUTPUT

```
0    Oggi c'è un bel sole, è caldo, si sta bene al ...
1                        Ho voglia di acqua e di mare
2    Che freddo con questa neve! si sta bene sotto ...
3                  Le previsioni danno ghiaccio è neve
4    Mi piace la vendemmia... Il marrone delle fogl...
5    Le foglie stanno cadendo tutte... è proprio au...
6    Domenica, se fa caldo, faremo un bel picnic in...
7    In questa stagione è bello camminare anche sot...
Name: text, dtype: object
```

Vettorizzazione - con python

PYTHON

```python
y = data["class"]
print(y)
```

OUTPUT

```
0         ESTATE
1         ESTATE
2       INVERNO
3       INVERNO
4       AUTUNNO
5       AUTUNNO
6     PRIMAVERA
7     PRIMAVERA
Name: class, dtype: object
```

Vettorizzazione - con python

- otteniamo la lista delle classi ordinate
- N.B. le classi **possono** essere anche di tipo non numerico in problemi di classificazione perché vengono tradotte automaticamente in numeri interi ordinati dal sistema

PYTHON

```python
ordered_class_list = list(set(y))
ordered_class_list.sort()
print(ordered_class_list)
```

OUTPUT

```
[' AUTUNNO', ' ESTATE', ' INVERNO', ' PRIMAVERA']
```

Vettorizzazione - con python

PYTHON

```python
# istruzioni per trasformare X in rappresentazione array
vectorizer_train = CountVectorizer(min_df=0)
# min_df=0 significa "considerare tutti i token, anche se compaiono una sola volta"
tokens = vectorizer_train.get_feature_names()
print(tokens)
```

OUTPUT

```
['acqua', 'ai', 'al', 'anche', 'autunno', 'bel', 'bello', 'bene', 'cadendo',
 'caldo', 'camminare', 'che', 'con', 'coperte', 'danno', 'delle', 'di', 'domenica',
 'fa', 'faremo', 'fiori', 'foglie', 'freddo', 'ghiaccio', 'ho', 'il', 'in', 'la', 'le',
 'mare', 'marrone', 'mezzo', 'mi', 'neve', 'oggi', 'piace', 'picnic', 'previsioni',
 'proprio', 'questa', 'se', 'si', 'sole', 'sotto', 'sta', 'stagione', 'stanno',
 'stelle', 'tutte', 'un', 'vendemmia', 'voglia']
```

- ottengo la lista di tutti* i tokens presenti nel dataset D

ref: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

*i token di dimensione 1 vengono esclusi per default perché poco significativi

Vettorizzazione - con python

- rappresentazione della prima istanza
  "Oggi c'è un bel sole, è caldo, si sta bene al mare"

ARRAY REPRESENTATION

```
[0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
 1, 0, 0, 0, 0, 1, 0, 0]
```

- adesso è possibile addestrare un classificatore perché siamo passati ad uno spazio numerico

PYTHON

```
vectorizer_train = CountVectorizer(min_df=0)
vectorizer_train.fit(X)
x_train_array = vectorizer_train.transform(X).toarray()
```

Vettorizzazione - con python

- Text analysis : in letteratura algoritmo Bayesiano
- probabilità che una certa parola P sia presente in un certo documento D data alla classe C

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

PYTHON

```python
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(x_train_array, y)
# classifichiamo una nuova frase
msg = "adoro giocare con l'acqua e la sabbia sotto il sole caldo"
# trasformiamo in array sulla base della **medesima vettorizzazione**
msg_array = vectorizer_train.transform([msg]).toarray()
print(msg_array)
```

OUTPUT

```
[1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
 0, 0, 0, 0, 0, 0, 0, 0]
```

Classificazione

- abbiamo trasformato una istanza di tipo testo nello spazio vettoriale booleano generato dall'operazione di tokenizzazione

PYTHON

```
# il msg è "adoro giocare con l'acqua e la sabbia sotto il sole caldo"
class_probabilities = clf.predict_proba(msg_array)[0]
print(list(zip(ordered_class_list, class_probabilities)))
```

OUTPUT

```
[(' AUTUNNO', 0.23512362166747044),
 (' ESTATE', 0.42392349623391468),
 (' INVERNO', 0.21196174811695767),
 (' PRIMAVERA', 0.12899113398165785)]
```

PYTHON

```
print(clf.predict(msg_array))
```

OUTPUT

```
[' ESTATE']
```

Giulio Angiani
Universita' degli Studi di Parma