SENTIMENT ANALYSIS SU RECENSIONI AMAZON

BIG DATA AND BUSINESS INTELLIGENCE

CDL INGEGNERIA INFORMATICA, ELETTRONICA E DELLE TELECOMUNICAZIONI

UNIVERSITÀ DEGLI STUDI DI PARMA – A.A. 2020-2021

PAPPANI FEDERICO – 298223

Codice Sorgente e Report disponibile su codes.pappani.me

OBIETTIVI DEL PROGETTO

Analisi di recensioni lasciate dagli utenti riguardo prodotti acquistati su Amazon, tramite diversi algoritmi di machine learning.

Confronto dei vari algoritmi per trovare quello che produce i risultati migliori.

Creazione di un modello per predizione di sentiment analysis su stringhe di testo arbitrarie.

DATASET E STRUMENTI UTILIZZATI

Per l'analisi e la creazione dei modelli è stato utilizzato Python in combinazione con la libreria scikit-learn e il software Jupyter Notebook.

Il dataset utilizzato è un dump di più di un milione e mezzo di recensioni utente per prodotti di elettronica di consumo, reso disponibile liberamente da Julian McAuley dell'Università della California a San Diego.

Le recensioni all'interno del dump sono salvate in formato JSON, contenente svariati parametri; ma per l'analisi sono stati utilizzati solo i campi di "testo della recensione" e "punteggio lasciato dall'utente".

```
{
"reviewerID": "A11AA01YRZT8DP",
"asin": "B004LGNB0A",
"reviewerName": "NP",
"helpful": [0, 0],
"reviewText": "Does the job well.",
"overall": 5.0,
"summary": "Five Stars",
"unixReviewTime": 1404345600,
"reviewTime": "07 3, 2014"
}
```


Se voto < 3: negativo
Se voto > 3: positivo
Voto = 3 è ignorato

PREPARAZIONE DEL DATASET

Per rendere possibile l'elaborazione dei dati su un PC comune, sono state estratte 75 mila recensioni casuali dal dump originale.

Il dataset è stato poi diviso in due parti, una per il training e una per il testing, rispettivamente il 67% e il 33% del totale.

Per escludere bias legati ad asimmetrie, il dataset è stato pulito, rendendo uguale il numero di sample con voto positivo e il numero di sample con voto negativo.

ADDESTRAMENTO DEL MODELLO

Il dataset è stato vettorializzato in bags of words tramite la funzione CountVectorizer, su cui poi il modello è stato addestrato utilizzando 4 diversi algoritmi di classificazione: albero decisionale, Naive Bayes gaussiano, SVM lineare, e regressione logistica.

SVM lineare: 86.2%, (86.1% positivi, 86.2% negativi)

Regressione logistica: 86.0%, (86.0% positivi, 86.1% negativi)

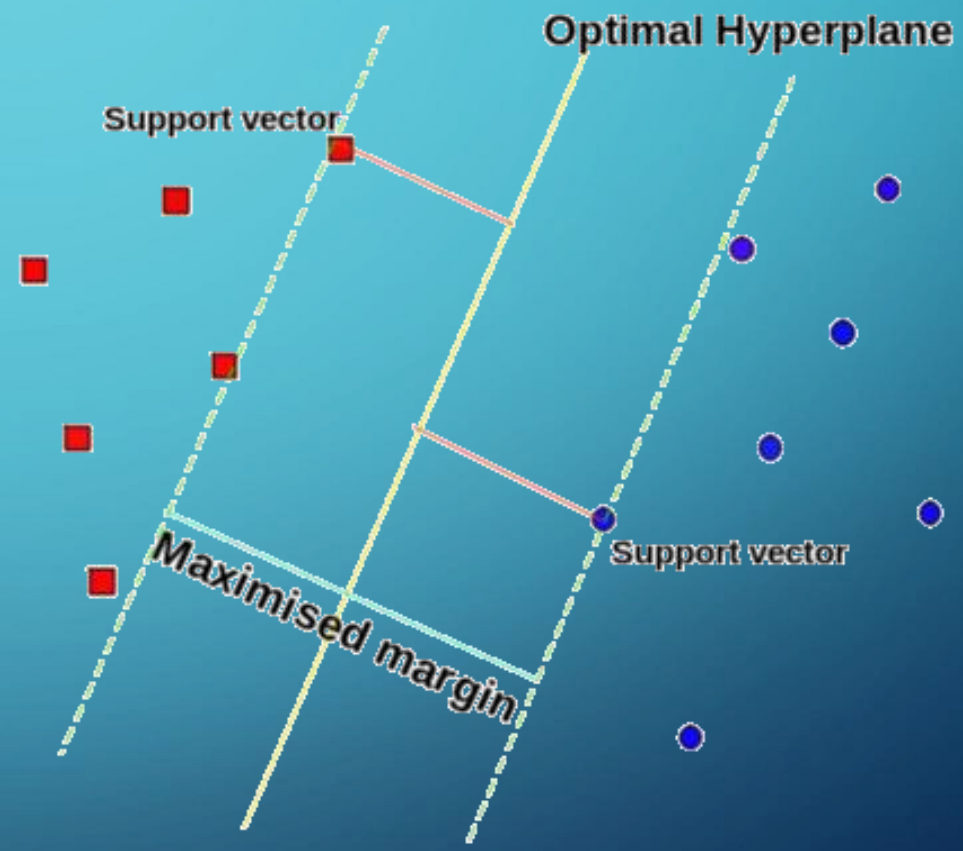
Naive Bayes gaussiano: 69.6%, (69.9% positivi, 69.3% negativi)

Albero decisionale: 68.8%, (69.2% positivi, 68.4% negativi)

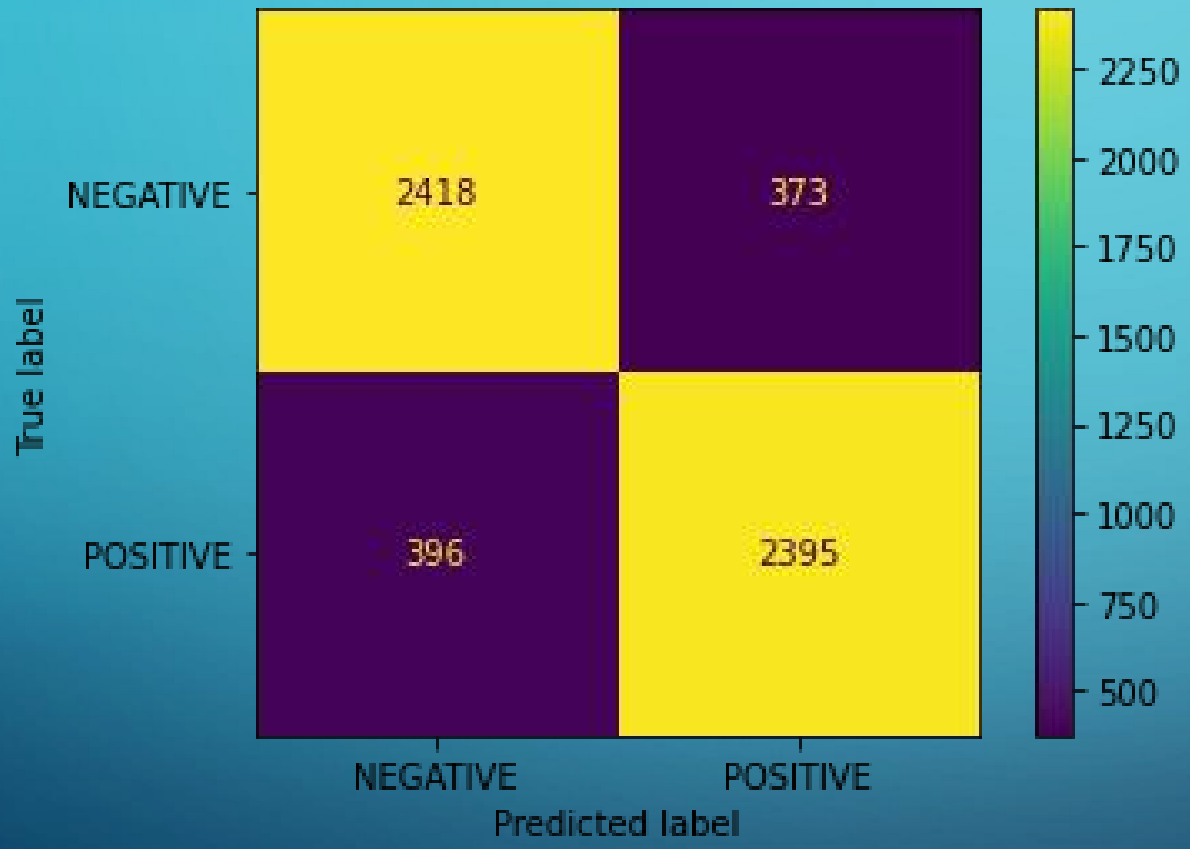
SVM LINEARE

L'algoritmo ad aver meglio performato è stato l'algoritmo a macchine a vettori di supporto lineare.

Questo algoritmo funziona bene su dataset linearmente separabili; utilizzando gli iperpiani per dividere il dataset in insiemi distinti.



MATRICE DI CONFUSIONE



La matrice di confusione rappresenta l'accuratezza statistica della classificazione.

In questo caso riporta i risultati corretti e, i falsi positivi e falsi negativi ottenuti dal modello predittivo.

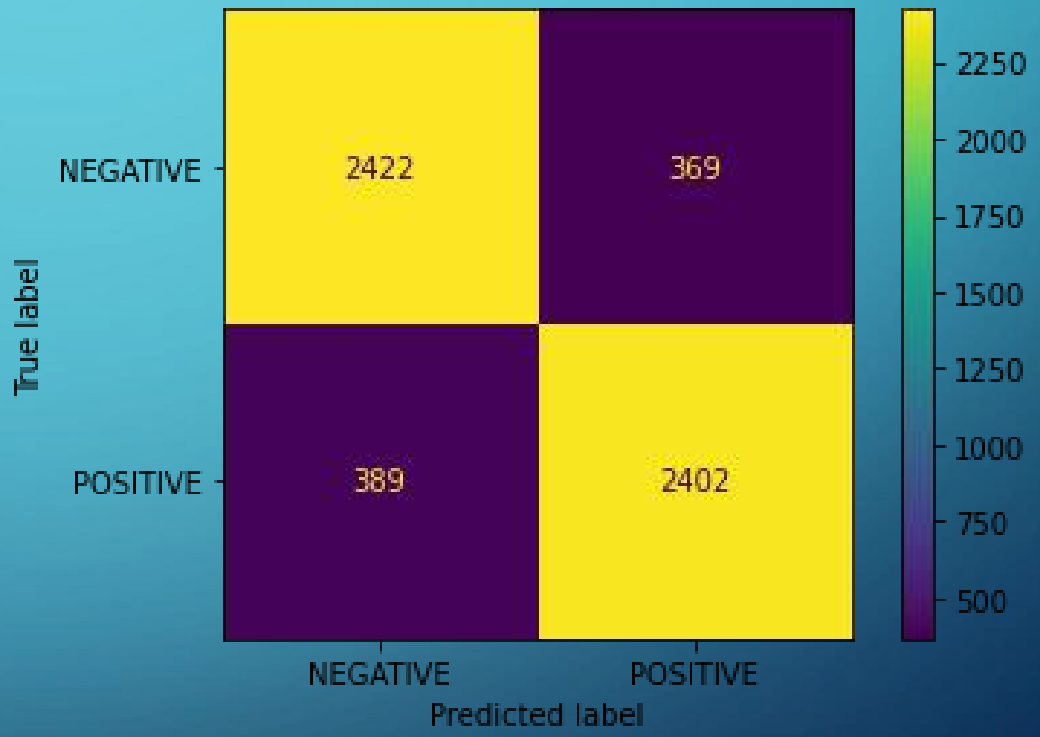
PREDIZIONE DI STRINGHE ARBITRARIE

Il modello è stato testato su stringhe di testo create arbitrariamente. Il modello ha sbagliato la predizione 2 volte su 22.

String	Expected Result	Model Result
"nice product!"	Positive	Positive
"incredibly good device"	Positive	Positive
"not good at all"	Negative	Negative
"it broke after 3 days"	Negative	Negative
"it doesn't work properly"	Negative	Negative
"it's amazing"	Positive	Positive
"I had to return it"	Negative	Negative
"it feels old"	Negative	Negative
"it fell apart quickly"	Negative	Negative
"I don't know why Amazon still sells this"	Negative	Negative
"it went straight to the trash"	Negative	Negative
"it's garbage"	Negative	Negative
"it does what it's meant to do"	Positive	Positive
"it works fine"	Positive	Positive
"recommended"	Positive	Positive
"you should buy it now"	Positive	Negative
"it looks good but it does not work as intended"	Negative	Negative
"today is a sunny day"	Positive	Negative
"good morning my dear"	Positive	Positive
"traffic around here has been quite noisy in the past few days"	Negative	Negative
"I crashed my car into a tree"	Negative	Negative
"today is a rainy day"	Negative	Negative

TUNING DEL MODELLO

Tramite la funzione grid search di scikit-learn sono stati testati vari parametri dell'addestramento, l'accuratezza è migliorata di 0.2%; ma ripetendo il test delle stringhe arbitrarie, il risultato è rimasto invariato.



SALVATAGGIO DEL MODELLO

Tramite la funzione Pickle, nativa di Python, è stato poi creato un dump del modello ottenuto, per poter essere eventualmente utilizzato in futuro.

CONCLUSIONI

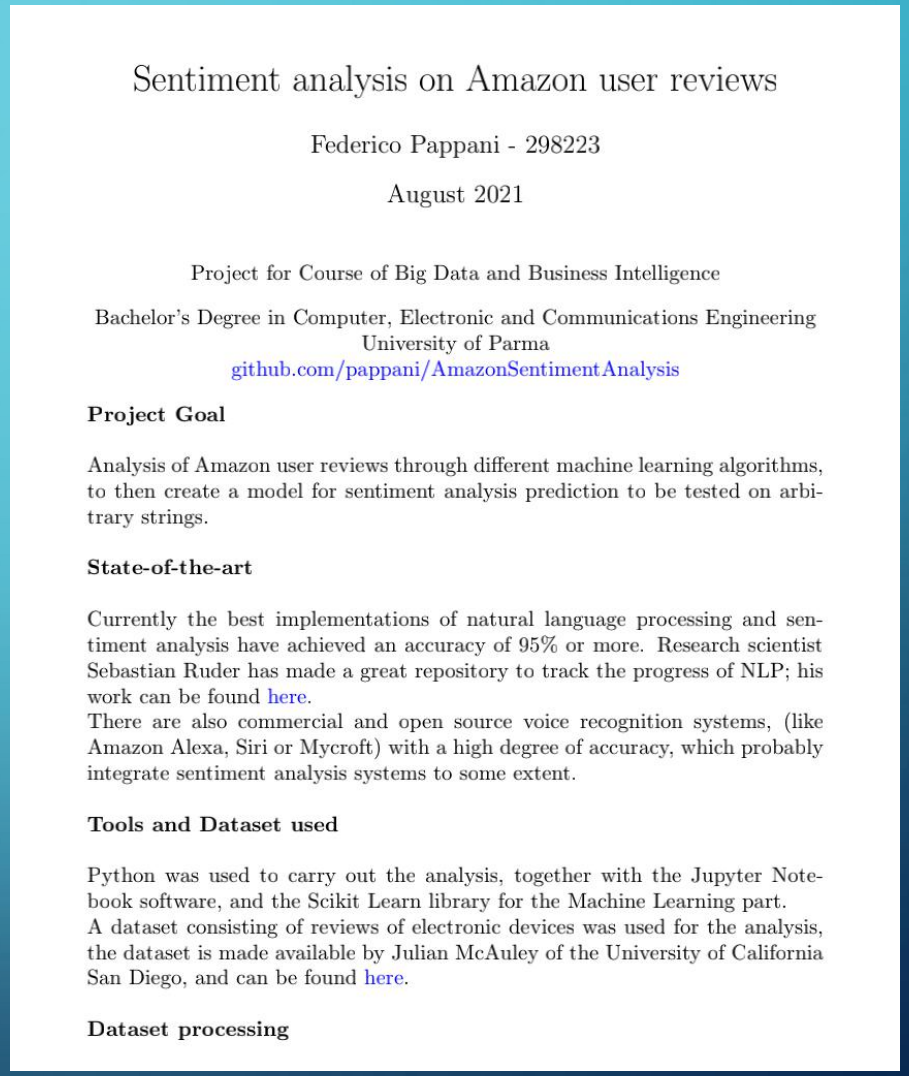
Il modello ha performato abbastanza bene, restituendo un'accuratezza del 86%.

Utilizzando una parte più grande del dataset iniziale, se non addirittura il dataset completo, il risultato sarebbe stato probabilmente migliore.

Purtroppo le risorse computazionali a disposizione hanno limitato il dataset a un numero ridotto di sample, ma che ha comunque permesso risultati soddisfacenti.

Codice sorgente e report annesso alla presentazione sono disponibili su GitHub: codes.pappani.me

pappani/AmazonSentimentAnalysis (github.com)



Sentiment analysis on Amazon user reviews

Federico Pappani - 298223

August 2021

Project for Course of Big Data and Business Intelligence

Bachelor's Degree in Computer, Electronic and Communications Engineering
University of Parma
github.com/pappani/AmazonSentimentAnalysis

Project Goal

Analysis of Amazon user reviews through different machine learning algorithms, to then create a model for sentiment analysis prediction to be tested on arbitrary strings.

State-of-the-art

Currently the best implementations of natural language processing and sentiment analysis have achieved an accuracy of 95% or more. Research scientist Sebastian Ruder has made a great repository to track the progress of NLP; his work can be found here.
There are also commercial and open source voice recognition systems, (like Amazon Alexa, Siri or Mycroft) with a high degree of accuracy, which probably integrate sentiment analysis systems to some extent.

Tools and Dataset used

Python was used to carry out the analysis, together with the Jupyter Notebook software, and the Scikit Learn library for the Machine Learning part.
A dataset consisting of reviews of electronic devices was used for the analysis, the dataset is made available by Julian McAuley of the University of California San Diego, and can be found here.

Dataset processing