UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Matematica e Informatica

Giuseppe Condorelli

Stackoverflow Topic Classification

Progetto di Ingegneria dei Sistemi Distribuiti

Prof. Emiliano Alessio Tramontana

Anno Accademico 2023 - 2024

Abstract

Il progetto propone una fase di raccolta di dati presenti su StackOverflow tramite le API offerte da StackExchange seguita da una pulizia, analisi ed infine allenamento di un classificatore con l'obiettivo di riuscire, dato del codice o argomento, a predirne correttamente il topic di appartenenza.

Contents

Chapter 1

Metodi

L'obiettivo di questo progetto è costruire un modello di Machine Learning in grado di classificare automaticamente il topic di appartenenza di un dato input, sia esso codice o testo.

Una delle sfide più cruciali è stata ottenere un dataset appropriato per affrontare questo problema. Per raggiungere questo scopo, ho sfruttato la vasta raccolta di dati offerta da StackOverflow [1], una delle piattaforme più popolari tra i programmatori per chiedere aiuto e condividere conoscenze tecniche.

1.1 StackExchange API

Per raccogliere i dati necessari, ho utilizzato le API offerte da StackExchange [2]. Queste API, grazie alla loro semplicità e flessibilità, hanno permesso di estrarre efficacemente i dati richiesti dalla piattaforma e organizzarli in un dataset strutturato. L'utilizzo delle API di StackExchange si è rivelato fondamentale per il successo di questa fase del progetto.

Nello specifico, al fine di specializzare il classificatore, ho ridotto la ricerca dei post usando come tag principale "Java", il che mi ha permesso di ottenere un totale di 1470 post. Di questi ho preso solo i dati che ho ritenuto essenziali per la corretta classificazione, ovvero:

- **id & link**: utili per poter identificare il post originale presente su StackOverFlow

- **title**: essenziale poichè contenente la domanda principale posta dal creatore del post
- **body**: contiene la domanda in maniera più estesa, presentando spesso anche del codice
- **answers**: risposte degli utenti al post, molto utili per identificare il topic
- **tags**: inseriti dall'utente, li userò come Ground Truth per la classificazione del topic

1.2 Pulizia dati

La fase di acquisizione dei dati è stata subito seguita da una di pulizia.
Il focus principale di questa fase è stato quello di rimuovere il possibile bias presente tra i dati. Per fare ciò ho filtrato le keyword presente nel campo "tags" dai campi "title", "body" ed "answers".
Ho inoltre rimosso la keyword "Java" presente nei campi "tags".
Infine, in modo tale da avere un singolo campo da usare come vettore features, ho accorpato insieme le colonne "title", "body" ed "answers" in un'unica colonna denominata "Full_text".

1.3 scikit-learn

Una volta ottenuti i dati, ho impiegato diversi strumenti offerti dalla libreria sklearn [3] per prepararli e utilizzarli nel modello di Machine Learning. In particolare:

- **TfidfVectorizer**: Questo strumento trasforma una collezione di documenti di testo in una matrice di numeri TF-IDF (Term Frequency-Inverse Document Frequency), che rappresenta l'importanza di una parola in un documento rispetto a una collezione di documenti.

- **MultiLabelBinarizer**: Questo strumento converte una lista di etichette multiple in una matrice binaria, necessaria per i modelli di Machine Learning che supportano la classificazione multilabel.
- **KNeighborsClassifier**: Questo è un algoritmo di apprendimento supervisionato che può essere utilizzato sia per problemi di classificazione che di regressione. Nel contesto di questo progetto, è stato utilizzato per costruire un modello di classificazione capace di determinare il topic di un dato input.
- **MultiOutputClassifier**: Questo strumento permette di trattare problemi di classificazione multilabel e multiclasse, avvolgendo un altro classificatore per supportare la previsione di più target.

1.4 Split dati

I dati sono stati divisi in training e testing set con un rapporto 80-20 permettendo di, finita la fase di training, poter testare il modello su dati da lui mai visti.

	precision	recall	f1-score	support
spring	1.00	1.00	1.00	18
maven	1.00	1.00	1.00	15
spring-mvc	1.00	1.00	1.00	44
jpa	1.00	1.00	1.00	8
mysql	1.00	1.00	1.00	10
jdbc	1.00	1.00	1.00	10
javascript	1.00	1.00	1.00	42
rest	1.00	1.00	1.00	20
generics	1.00	1.00	1.00	6
algorithm	1.00	1.00	1.00	10
sockets	1.00	1.00	1.00	23
user-interface	1.00	1.00	1.00	9
kotlin	1.00	1.00	1.00	8
methods	1.00	1.00	1.00	55
micro avg	1.00	1.00	1.00	278
macro avg	1.00	1.00	1.00	278
weighted avg	1.00	1.00	1.00	278
samples avg	1.00	1.00	1.00	278

Figure 2.1: Confusion Matrix

	id	Full_text	tags	predictions	False Positive	True Positive	False Negative	True Negative
0	72857806	JAXB don't marshal attribute <p>I have som...	['xml']	['xml']	0	1	0	48
1	75516396	Email address inside <>tag is taken as ...	['html', 'spring-boot']	['html', 'spring-boot']	0	2	0	47
2	75516396	Email address inside <>tag is taken as ...	['html', 'spring-boot']	['html', 'spring-boot']	0	2	0	47
3	8400301	<cout/> unknown tag <p>Why I get error ...	['eclipse', 'jsp']	['eclipse', 'jsp']	0	2	0	47
4	76858009	How to show this tags on intellij idea communi...	['intellij-idea']	['intellij-idea']	0	1	0	48
..	..	..	..	..	..	..	..	..
201	75516396	Email address inside <>tag is taken as ...	['html', 'spring-boot']	['html', 'spring-boot']	0	2	0	47
202	73717178	5 order tests with @Tag from some es <p>I hav...	['junit']	['junit']	0	1	0	48
203	653085	Purpose of tag library? <p>What is the purpose...	['jsp']	['jsp']	0	1	0	48
204	653085	Purpose of tag library? <p>What is the purpose...	['jsp']	['jsp']	0	1	0	48
205	7568433	How to get text between two Elements in DOM ob...	['html']	['html']	0	1	0	48

Figure 2.2: Risultati

Chapter 2

Risultati

Grazie all'uso combinato degli strumenti proposti nel capitolo precedente, è stato possibile addestrare un modello di Machine Learning che, dato un nuovo testo o codice, è in grado di classificare il suo topic di appartenenza.

Al fine di verificarne le capacità, il modello è stato testato sul testing set mostrando degli ottimi risultati, con una precisione del 100%, come è anche possibile constatare

dalla Confusion Matrix (Fig.2.1) e dalla tabella (Fig.2.2).

Bibliography

[1] StackOverflow. *StackOverflow*. https://stackoverflow.com/. Accessed: 2024-07-27.

[2] StackExchange. *StackExchange API*. https://api.stackexchange.com/. Accessed: 2024-07-27.

[3] Scikit-learn. *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org/. Accessed: 2024-07-27.