

**Cognitive Computing System applicato allo studio del Glioblastoma**

*Docenti:*

Prof. Paolo Maresca

*Autori:* Francesca Lettieri - M63000956 Angelo Russo - M63001016

Guido Maria Secondulfo - M63000927

## Indice

1. [Introduzione](#_bookmark0) 3
   1. [Traccia](#_bookmark1) 3
2. [Architettura](#_bookmark2) 3
   1. [Tecnologie utilizzate](#_bookmark3) 3
      1. [E-utilities](#_bookmark4) 3
      2. [IBM Cloud Db2](#_bookmark5) 3
      3. [GitHub](#_bookmark6) 3
      4. [IBM Cloud Foundry](#_bookmark7) 3
      5. [IBM Watson Machine Learning](#_bookmark8) 4
      6. [IBM Watson Knowledge Studio](#_bookmark9) 4
      7. [fastai](#_bookmark10) 4
      8. [IBM Cloud API](#_bookmark11) 4
      9. [Jupyter](#_bookmark12) 4
3. [Implementazione](#_bookmark13) 5
   1. [Flusso di esecuzione del sistema globale](#_bookmark14) 5
4. [Apprendimento sugli articoli medici tramite NLP](#_bookmark15) 6
   1. [Flusso di esecuzione del sistema di retrieval degli articoli](#_bookmark16) 6
      1. [E-utilities](#_bookmark17) 6
      2. [Query di ricerca](#_bookmark18) 7
      3. [GitHub e IBM Cloud Foundry](#_bookmark20) 9
      4. [IBM Cloud Db2](#_bookmark21) 9
      5. [Funzioni implementate](#_bookmark23) 14
   2. [Flusso di esecuzione del sistema di apprendimento sugli articoli](#_bookmark24) . 16
   3. [IBM Watson Knowledge Studio](#_bookmark25) 16
      1. [Import degli Articoli](#_bookmark26) 16
      2. [Creazione del Modello Entit`a-Relazione](#_bookmark27) 16
   4. [Valutazione del modello](#_bookmark29) 23
5. [Apprendimento sulle Risonanze Magnetiche tramite Visual Re- cognition](#_bookmark30) 27
   1. [Flusso di esecuzione del sistema di apprendimento sulle MRI](#_bookmark31) 27
   2. [Visual Recognition di Watson Studio](#_bookmark32) 27
      1. [Implementazione](#_bookmark33) 28
      2. [Risultati ottenuti - Primo Ciclo](#_bookmark35) 31
      3. [Risultati ottenuti - Secondo Ciclo](#_bookmark36) 33
      4. [Risultati ottenuti - Terzo Ciclo](#_bookmark37) 35
      5. [Risultati ottenuti - Quarto Ciclo](#_bookmark38) 37
6. [Considerazioni e Sviluppi Futuri](#_bookmark39) 39
7. [RadiAnt](#_bookmark40) 41
   1. [Pre-processing effettuato tramite RadiAnt](#_bookmark41) 42

##### [Evaluation Metrics](#_bookmark42) 44

* 1. [Accuracy](#_bookmark43) 44
  2. [Precision e Recall](#_bookmark44) 44
  3. [F1-Score](#_bookmark45) 45

##### [fastai](#_bookmark46) 46

* 1. [CNN con la libreria *fastai*](#_bookmark47)46
     1. [Implementazione](#_bookmark48) 46
     2. [Risultati ottenuti](#_bookmark50) 48

##### [Rimozione testi pediatrici](#_bookmark51) 51

## Introduzione

#### Traccia

Il progetto da noi sviluppato si articola in due percorsi distinti:

* + 1. Retrieval da PubMed di articoli inerenti al Glioblastoma Multiforme (GBM), creazione di un database atto a contenere le informazioni rilevanti ed, in- fine, creazione di un modello di NLP (Natural Language Processing) al fine di realizzare un sistema di apprendimento in grado di realizzare una standard meta-analysis accurata;
    2. Acquisizione delle risonanze magnetiche (MRI) di pazienti affetti da GBM e costruzione di un modello di apprendimento su di esse.

## Architettura

#### Tecnologie utilizzate

##### E-utilities

Le E-utilities (Entrez Programming Utilities) costituiscono un insieme di 8 pro- grammi server-side che forniscono un’interfaccia stabile per il sistema di databa- se e query Entrez. Le E-utilities utilizzano una sintassi a URL fisso che traduce un insieme di parametri di input nei valori necessari ai componenti software del NCBI per ricercare e recuperare i dati richiesti.

##### IBM Cloud Db2

IBM Db2 on Cloud `e un database cloud transazionale progettato per offrire pre- stazioni elevate. Esso assicura la scalabilit`a orizzontale e verticale per soddisfare le richieste di business.

##### GitHub

Git `e un software di controllo versione distribuito utilizzabile da interfaccia a riga di comando, sviluppato per gestire anche progetti di elevate dimensioni con elevate velocit`a ed efficienza.

##### IBM Cloud Foundry

Cloud Foundry `e una PaaS (platform-as-a-service) open source su IBM Cloud che consente di implementare e scalare le app senza gestire i server. E` stato da noi utilizzato al fine di poter eseguire uno script in linguaggio Python all’interno dell’ambiente IBM Cloud.

##### IBM Watson Machine Learning

IBM Watson Machine Learning aiuta i data scientist e gli sviluppatori ad accele- rare l’implementazione di AI e machine-learning. Con i suoi modelli estensibili, aperti, Watson Machine Learning aiuta le aziende a semplificare e sfruttare l’AI in scala su qualsiasi cloud.

##### IBM Watson Knowledge Studio

IBM Watson consente l’apprendimento del linguaggio del dominio con modelli personalizzati che identificano le entit`a e le relazioni specifiche del settore in testo non strutturato. I modelli possono essere creati in un ambiente collaborativo progettato sia per gli sviluppatori che per gli esperti di dominio, senza il bisogno di scrivere codice, e possono essere utilizzati in IBM Watson Discovery, IBM Watson Natural Language Understanding e IBM Watson Explorer.

##### fastai

fastai `e una libreria di deep learning che fornisce agli operatori componenti di alto livello in grado di fornire rapidamente e facilmente risultati di ultima gene- razione in domini di deep learning standard, e fornisce ai ricercatori con com- ponenti di basso livello che possono essere mescolati e abbinati per costruire nuovi approcci. fastai mira a fare entrambe le cose senza compromessi sostan- ziali in facilit`a d’uso, flessibilit`a, o prestazioni. Questo `e possibile grazie ad un’architettura a strati, che esprime schemi comuni di base di molte tecniche di deep learning e di elaborazione dati in termini di astrazioni disaccoppiate. Queste astrazioni possono essere espresse in modo conciso e chiaro sfruttando il dinamismo del linguaggio Python sottostante e la flessibilit`a della libreria Pytorch.

##### IBM Cloud API

Le IBM Cloud API sono delle API che consentono di garantire l’accesso ai servizi IBM Cloud ai propri servizi ed alle proprie applicazioni che lo necessitano.

##### Jupyter

Il Progetto Jupyter `e un’organizzazione creata per sviluppare software open- source, e supportare ambienti di esecuzione in decine di lingue.

JupyterLab offre un’interfaccia per la gestione di documenti di diverso formato, per attivit`a di text editing e attivit`a da terminale.

## Implementazione

#### Flusso di esecuzione del sistema globale

Il diagramma di flusso rappresentante l’elaborazione effettuata `e il seguente:

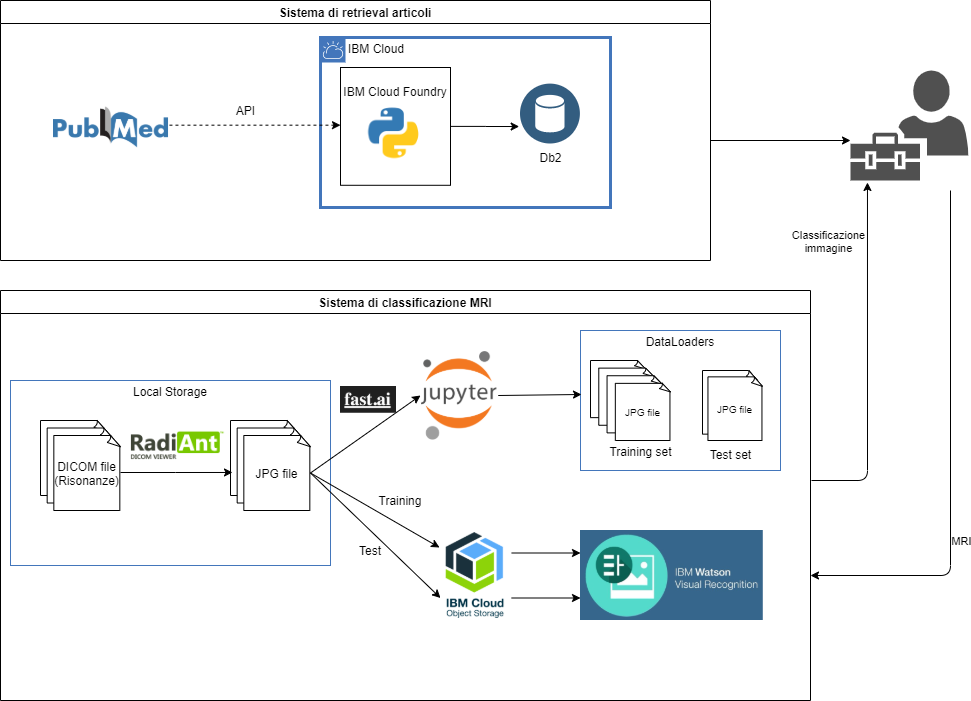


Figura 1: Diagramma di flusso dell’esecuzione

Al fine di agevolare la trattazione, affronteremo i problemi separatamente.

## Apprendimento sugli articoli medici tramite NLP

#### Flusso di esecuzione del sistema di retrieval degli ar- ticoli

Figura 2: Diagramma di flusso dell’esecuzione

Innanzitutto `e stato effettuato il download degli articoli desiderati da PubMed. Dopo una prima elaborazione di tali documenti, essi sono stati memorizzati in un database relazionale all’interno dell’ambiente IBM Cloud.

##### E-utilities

In seguito ad un confronto con gli esperti di dominio, abbiamo ritenuto oppor- tuno effettuare un retrieval di articoli provenienti unicamente da PubMed, in quanto source affidabile e attendibile.

A tal fine `e stato sviluppato uno script in linguaggio Python (*pubmed.py* ) che, sfruttando le E-utilities, `e in grado di effettuare il retrieval degli articoli deside- rati da PubMed stesso.

In particolare, la funzione *getPubMedIDArticles* recupera gli ID degli articoli desiderati da PubMed, in base ai termini specificati. L’implementazione di tale funzione `e la seguente:

**def** getPubMedIDArticles ( query ) :

t e s t = ” esearch . f c g i ?db=pubmed”

term = ”&term=”+query *#query d e l l a r i c e r c a*

maxArts =”&retmax=100000 ” *#Numero massimo d i a r t i c o l i r e c u p e rat i d a l l a r i c e r c a*

l i n k = ” https : // e u t i l s . ncbi . nlm . nih . gov/ e ntre z / e u t i l s

/%22+ t e s t+term+maxArts

s = u r l l i b . request . urlopen ( l i n k ) contents = s . read ()

f i l e = open ( ” e x p o r t i d A r t i c l e s . xml” , ’wb ’ ) f i l e . write ( contents )

f i l e . c l o s e ()

##### Query di ricerca

La query effettuata `e stata elaborata dagli esperti di dominio. Essa espone quali siano i termini da ricercare all’interno di tutti i campi di un articolo, inclusi i MeSH Terms [1](#_bookmark19) associati all’articolo stesso.

I criteri di inclusione per la selezione degli articoli sono i seguenti:

* Studi randomizzati di confronto fra trattamenti;

Fase II o III (compresi studi Fase I/II purch`e contengano risultati adeguati della Fase II;

•

OS sopravvivenza globale (overall survival) e PFS sopravvivenza libera da progressione (progression free survival) e ORR (Overall Response Rate) Tasso di risposta complessiva;

•

Solo Glioblastoma multiforme (GBM) o High Grade Glioma in ripresa dopo pregressa chirurgia e terapia adiuvante o in progressione.

•

I criteri di esclusione per la selezione degli articoli sono, invece, i seguenti:

* Tumori pediatrici;
* Studi non randomizzati;
* Studi a singolo braccio;
* Studi di fase I;
* GBM in trattamento adiuvante;
* Review/metanalisi;
* Studi preclinici;
* Studi di farmacoeconomia;
* Studi di confronto senza analisi statistiche.

La formulazione risultante della query `e, dunque, la seguente:

1Medical Subject Headings (acronimo: MeSH) `e un enorme vocabolario controllato (o si- stema di metadati) ideato con l’obiettivo di indicizzare la letteratura scientifica in ambito bio- medico. Il tesauro `e stato creato dalla National Library of Medicine (NLM) degli Stati Uniti, che `e responsabile anche della sua gestione. Il MeSH viene adoperato per l’indicizzazione degli articoli delle oltre 5000 riviste mediche presenti nel database bibliografico Medline/PubMed e nel catalogo dei libri della NLM. Il vocabolario pu`o essere consultato e scaricato gratuitamente da tutti gli utenti di internet. La stampa dell’edizione cartacea `e stata abbandonata nel 2007.

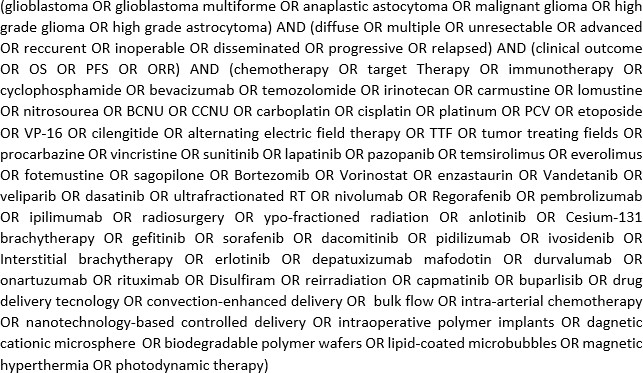


Figura 3: Query originale

Tale query `e stata, quindi, da noi raffinata al fine di poter effettuare il retrieval degli articoli che soddisfacessero i criteri richiesti.

L’operazione di raffinamento `e stata dettata dalla risposta delle API di PubMed a determinati termini (es. ”ypo-fractioned”, ”dagnetic”) che facevano parte della query iniziale; tali termini, omessi in fase di raffinamento, non risultavano presenti all’interno del tesauro (i.e. il MeSH).



Figura 4: Query Raffinata

In entrambi i casi, il numero di articoli recuperati `e pari a 1’318, per questo l’eliminazione dei termini `e risultata ininfluente per quanto riguarda il risultato finale ottenuto. Un ulteriore filtraggio degli articoli recuperati `e stato fatto in base alla data di pubblicazione, secondo quanto richiesto e consigliato dagli esperti di dominio: escludendo gli articoli pubblicati prima del 1990, il numero finale di articoli recuperati `e pari a *1’313*.

##### GitHub e IBM Cloud Foundry

Per poter importare gli articoli in ambiente IBM Cloud, innanzitutto `e stato creato un repository su GitHub all’interno del quale `e stato memorizzato il file *pubmed.py*.

In particolare, sono state effettuate le operazioni di **add**, **commit** e **push** del file all’interno del repository attraverso il prompt dei comandi.

##### IBM Cloud Db2

Innanzitutto `e stata istanziata una versione lite del Database Db2. Utilizzando le credenziali di accesso fornite dal sistema e le API IBM DB, `e stato possibile accedere al Database attraverso una funzione all’intero del file *pubmed.py*.

Per la modellazione del database, volendo tener traccia di un’unica entit`a, gli articoli, `e stato realizzato il seguente diagramma ER[2](#_bookmark22):



Figura 5: Diagramma ER Abbiamo ipotizzato di memorizzare, per ogni articolo:

* + - * DOI (Digital Object Identifier) - quando disponibile;
      * ID dell’articolo nel database PubMed;
      * Titolo;

• Lista di autori - quando disponibile;

2In informatica, nell’ambito della progettazione dei database, il modello entity-relationship (detto anche modello E-R; in italiano: modello entit`a-associazione o modello entit`a-relazione)

`e un modello teorico per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione, formalizzato dal prof. Peter Chen nel 1976.

Il modello entit`a-associazione viene spesso utilizzato nella prima fase della progettazione di una base di dati, nella quale `e necessario tradurre le informazioni risultanti dall’analisi di un determinato dominio in uno schema concettuale, chiamato schema E-R (schema entit`a- associazione) o diagramma E-R (diagramma entit`a-associazione).

* + - * Data di pubblicazione - quando disponibile;
      * Rivista di pubblicazione - quando disponibile;
      * Abstract - quando disponibile.

Osserviamo che, durante l’attivit`a di recupero dalle API Pubmed, il DOI era assente in alcuni articoli, il che lo rendeva inadeguato al ruolo di chiave primaria dell’entit`a. Per questo motivo, abbiamo utilizzato un ulteriore identificativo numerico con auto-incremento come chiave primaria.

Quindi, `e stato possibile creare una tabella ARTICOLIGBM attraverso la se- guente query:

**CREATE TABLE** arti c o l i g bm (

id **INTEGER NOT NULL PRIMARY KEY** GENERATED ALWAYS **AS IDENTITY** (START WITH 1 INCREMENT **BY** 1 ) ,

pubmedId **INTEGER UNIQUE NOT NULL**, doi **VARCHAR**( 80 ) ,

t i t o l o **VARCHAR**( 1000 ) ,

auto r i **VARCHAR**( 5000 ) ,

data Pubblicazione **DATE**, abs trac t **VARCHAR**( 10000 ) ,

nome Rivista **VARCHAR** ( 1000 ) ”

)

La connessione al database `e realizzata attraverso la funzione *connectionDB2*, mentre tale connessione viene chiusa attraverso la funzione *endConnectionDB2*. L’implementazione di entrambe le funzioni `e riportata di seguito:

**def** connectionDB2 () :

c o nn i nf o = ””DATABASE=BLUDB;HOSTNAME=dashdb txn sbox yp lon02 13.

− − −

− −

s e r v i c e s . eu gb . bluemix . net ;PORT=50000;PROTOCOL=TCPIP ; UID=xxd26106 ;PWD=s5bzk6p ˆ3 s 94 hght ; ””

−

conn = ibm db . connect ( conn info , ”” , ”” )

**return** conn

**def** endConnectionDB2 ( conn ) : ibm db . c l o s e ( conn )

Si noti che le informazioni relative al Database, quali nome dello stesso, l’Host- name, il porto, il protocollo, l’username e la password, sono strettamente legate all’account IBM Cloud che ha istanziato il database. In caso di migrazione del database, o di creazione di uno nuovo, `e necessario modificare l’intera variabi- le *conn info* con le rispettive Credenziali di Servizio generate sul portale IBM Cloud.

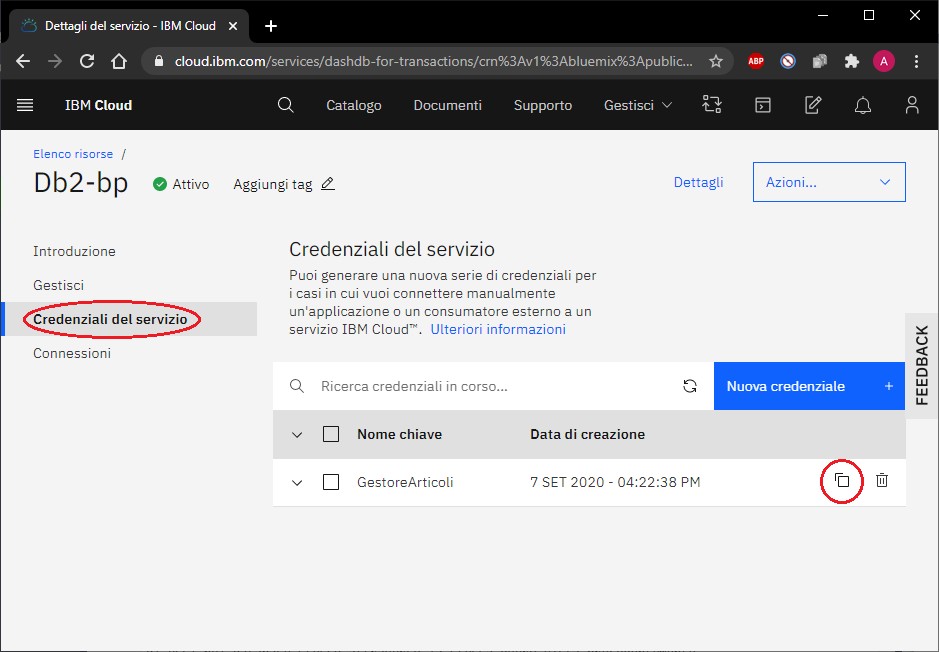


Figura 6: Schermata DB2 per ottenere le credenziali del servizio

Infine, il popolamento del database pu`o essere effettuato attraverso due diverse funzioni. La prima, *insertArticleDB2*, inserisce un singolo articolo all’interno del database attraverso la seguente query sql:

**INSERT INTO** a r t i c o l i ( pubmedId , doi , t i t o l o , autori , data Pubblicazione , abstract , nome Rivista )

**VALUES** ( valore pubmedId , v al o re do i , v a l o r e t i t o l o , v a l o re auto r i ,

valore data Pubblicazione , v al o re abs trac t , valore nome Rivista ) ;

La seconda, *insertArrayDB2*, `e quella che `e stata utilizzata per realizzare il primo popolamento del database. Essa riceve in ingresso un array di articoli, andando ad inserire ognuno di essi nel database stesso attraverso la stessa query vista precedentemente.

In entrambi i casi, si `e resa, quindi, necessaria la definizione di una classe Articolo, la cui implementazione `e riportata di seguito:

**class** A rti c o l o () :

**def** i n i t ( s e l f , t i t o l o , autori , pubDate , pubMedId , doi , abstract , j o u r n a l T i t l e ) :

s e l f . t i t o l o = t i t o l o s e l f . auto r i = auto r i

s e l f . abs trac t = abs trac t

s e l f . doi = doi

s e l f . pubMedId = pubMedId s e l f . pubDate = pubDate

s e l f . j o u r n a l T i t l e = j o u r n a l T i t l e

**def** s t r ( s e l f ) :

**return** ”PUBMED ID : %s nDOI : %s n Titolo : %s nAutori : %s nData di Pubblicazione : %s

\ \

\ \ \

n Abstract : %s n Journal Ti t l e : %s n n” % ( s e l f . pubMedId , s e l f . doi , s e l f . t i t o l o , ’ , ’ . j o i n (

\ \ \

s e l f . auto r i ) , s e l f . pubDate , s e l f . abstract , s e l f . j o u r n a l T i t l e )

L’implementazione delle funzioni sopra menzionate `e la seguente:

**def** insert Article DB 2 ( a r t i c o l o ) : abstract temp = ” ”

t i to l o te mp = ” ” j t i t l e t e m p = ” ”

a u t o r i i n s e r t = ’ , ’ . j o i n ( a r t i c o l o . auto r i )

**i f** ( a u t o r i i n s e r t != None ) :

a u t o r i i n s e r t = a u t o r i i n s e r t . re pl ac e ( ” ’ ” , ” ’ ’ ” )

**i f** ( a r t i c o l o . t i t o l o != None ) :

t i to l o te mp = a r t i c o l o . t i t o l o . re pl ac e ( ” ’ ” , ” ’ ’ ” )

**i f** ( a r t i c o l o . abs trac t != None ) :

abstract temp = a r t i c o l o . abs trac t . re pl ac e ( ” ’ ” , ” ’ ’ ” )

**i f** ( a r t i c o l o . j o u r n a l T i t l e != None ) :

j t i t l e t e m p = a r t i c o l o . j o u r n a l T i t l e . re pl ac e ( ” ’ ” , ” ’ ’ ” )

i n s e r t = ibm db . exec immediate ( conn , ”INSERT INTO a r t i c o l i ( pubmedId , doi , t i t o l o , autori ,

data Pubblicazione , abstract , nome Rivista ) VALUES ( ’ ”+a r t i c o l o . doi+” ’ , ’ ”+ti to l o te mp+” ’ , ’ ”+

a u t o r i i n s e r t+” ’ , ’ ”+ a r t i c o l o . pubDate+” ’ , ’ ”+ abstract temp+” ’ , ’ ”+j t i t l e t e m p+” ’ ’ ) ” )

**print** ( ” [ ”+**str** ( i )+” ] I n s e r i t o n” ) *#DEBUG d e l l ’ a r t i c o l o i n s e r i t o*

− \

**def** insertArrayDB2 ( a r t i c o l i ) :

**for** a r t i c o l o **in** a r t i c o l i : abstract temp = ” ”

t i to l o te mp = ” ”

j t i t l e t e m p = ” ”

a u t o r i i n s e r t = ’ , ’ . j o i n ( a r t i c o l o . auto r i )

**i f** ( a u t o r i i n s e r t != None ) :

a u t o r i i n s e r t = a u t o r i i n s e r t . re pl ac e ( ” ’ ” , ” ’ ’ ” )

**i f** ( a r t i c o l o . t i t o l o != None ) :

t i to l o te mp = a r t i c o l o . t i t o l o . re pl ac e ( ” ’ ” , ” ’ ’ ” )

**i f** ( a r t i c o l o . abs trac t != None ) :

abstract temp = a r t i c o l o . abs trac t . re pl ac e ( ” ’ ”

, ” ’ ’ ” )

**i f** ( a r t i c o l o . j o u r n a l T i t l e != None ) :

j t i t l e t e m p = a r t i c o l o . j o u r n a l T i t l e . re pl ac e ( ” ’ ” , ” ’ ’ ” )

i n s e r t = ibm db . exec immediate ( conn , ”INSERT INTO a r t i c o l i ( pubmedId , doi , t i t o l o , autori ,

data Pubblicazione , abstract , nome Rivista ) VALUES ( ’ ”+a r t i c o l o . doi+” ’ , ’ ”+ti to l o te mp+” ’ ,

’ ”+ a u t o r i i n s e r t+” ’ , ’ ”+ a r t i c o l o . pubDate+” ’ , ’ ”+abstract temp+” ’ , ’ ”+j t i t l e t e m p+” ’ ’ ) ” )

**print** ( ” [ ”+**str** ( i )+” ] I n s e r i t o n” ) *#DEBUG d e l l ’ a r t i c o l o i n s e r i t o*

− \

Inizialmente, si era pensato di mantenere in memoria le informazioni degli ar- ticoli recuperati, per poi caricarle tutte insieme sul database: questo metodo per`o `e risultato poco efficiente a causa della grande mole di articoli da caricare. Tale risultato `e stato ottenuto stressando il sistema tramite una query generica di prova, che ricercasse gli articoli che contenessero in qualsiasi campo (sia esso Titolo, MeSH o Abstract) il termine ”glioblastoma”. Questa query di prova ha prodotto come risultato ben 41’364 articoli e pubblicazioni, rallentando di molto il processo di recupero, il mantenimento temporaneo in memoria e infine

l’inserimento nel Database degli stessi. E` risultato molto piu` conveniente, piut-

tosto, gestire grosse moli di articoli recuperandoli e inserendoli immediatamente nel database, in modo da non perdere l’eventuale cluster di articoli in caso di disconnessioni causate dal server di PubMed (problema che, piu` volte, `e stato riscontrato durante il testing dello script). Conviene utilizzare, quindi, la fun- zione *insertArrayDB2* quando vengono recuperati quantit`a minime di articoli (nell’ordine massimo delle centinaia), come nel caso della query utilizzata ed esposta nel sottoparagrafo 3.1.2, dove la query ha restituito 1’318 articoli come esito della ricerca.

##### Funzioni implementate

Dopo aver ottenuto la lista contenente gli ID degli articoli, la funzione *pubMe- dIDConverter* recupera le informazioni ad essi associate. L’implementazione di tale funzione `e la seguente:

**def** PubMedIdConverter ( a r t i c o l i ) : d a t a f i l e = ’ export . xml ’

t re e = ET. parse ( d a t a f i l e ) root = t re e . g e tro o t ()

i d c o n v e r te r = ” https : // e u t i l s . ncbi . nlm . nih . gov/ e ntre z / e u t i l s / e f e tc h . f c g i ?db=pubmed&id=”

i =0

**for** c h i l d **in** root . f i n d a l l ( ” . / Id L i s t / Id” ) : i=i +1

**id** = c h i l d . text

i d c o nv e rte r 1 = i d c o n v e r te r+**id**+”&retmode=xml”

xml = u r l l i b . request . urlopen ( i d c o nv e rte r 1 ) . read ()

xml tree = ET. f ro mstri ng ( xml ) *# Root*

**for** c hi l d 2 **in** xml tree . f i n d a l l ( ” PubmedArticle ” ) : te m p t i t l e = ””

temp abstract = ”” temp author = ””

temp doi = ” Mancante” temp pubdate = ””

a u t o r i v e c t o r = [ ]

**for** c h i l d 2 0 **in** c hi l d 2 . f i n d a l l ( ” Medline Citation / A r t i c l e ” ) :

**for** c h i l d 2 J o u r n a l T i t l e **in** c h i l d 2 0 . f i n d a l l ( ” Journal / Ti t l e ” ) :

t e m p j t i t l e = c h i l d 2 J o u r n a l T i t l e . text

**for** c h i l d 2 1 **in** c h i l d 2 0 . f i n d a l l ( ” A r t i c l e T i t l e ” ) :

te m p t i t l e = c h i l d 2 1 . text

**for** c h i l d 2 2 **in** c h i l d 2 0 . f i n d a l l ( ” Abstract / AbstractText ” ) :

temp abstract = c h i l d 2 2 . text

**for** c h i l d 2 3 **in** c h i l d 2 0 . f i n d a l l ( ” Author List / Author” ) :

*#La presente s o l u z i o n e e ’ concepita*

*nel caso s i v o g l i a t enere t r a c c i a*

*s o l o d i i n i z i a l i de l nome e cognome ( per in t e ro )*

path cognome = c h i l d 2 3 . f i nd ( ” LastName” )

p a t h i n i z i a l i = c h i l d 2 3 . f i nd ( ” I n i t i a l s ” )

**i f** ( path cognome != None ) :

**i f** ( p a t h i n i z i a l i != None ) :

temp author = c h i l d 2 3 . f i nd ( ” LastName” ) . text +” ”+

p a t h i n i z i a l i . text+” . ”

##### else :

temp author = c h i l d 2 3 . f i nd ( ”

LastName” ) . text

a u t o r i v e c t o r . append ( temp author )

**for** c h i l d 2 4 **in** c hi l d 2 . f i n d a l l ( ”PubmedData” ) :

**for** c h i l d 2 5 **in** c h i l d 2 4 . f i n d a l l ( ” History

/PubMedPubDate /[ @PubStatus=’ e ntre z ’ ] ” )

:

temp pubdate = c h i l d 2 5 [ 0 ] . text+” ”+ c h i l d 2 5 [ 1 ] . text+” ”+c h i l d 2 5 [ 2 ] . text

−

−

c h i l d Fi n a l e = c h i l d 2 4 . f i nd ( ”

A r t i c l e I d L i s t / A r t i c l e I d /[ @IdType=’ doi ’ ] ” )

**i f** ( c h i l d Fi n a l e != None ) :

temp doi = c h i l d Fi n a l e . text

**print** ( ” [ ”+**str** ( i )+” ] ” , temp doi ) *#*

−

*Debug d e l l ’ a r t i c o l o recuperato*

a r t i c o l o = A rti c o l o ( te mp t i t l e ,

auto r i v e c to r , temp pubdate , temp pubmedid

, temp doi , temp abstract , t e m p j t i t l e ) a r t i c o l i . append ( a r t i c o l o )

**return** a r t i c o l i

#### Flusso di esecuzione del sistema di apprendimento sugli articoli

Figura 7: Diagramma di flusso dell’esecuzione

Osserviamo innanzitutto che all’interno di IBM Watson Knowledge Studio ven- gono memorizzati direttamente gli articoli completi. In particolare, momen- taneamente gli articoli sono stati scaricati manualmente sfruttando i Proxy di diverse universit`a campane, quali l’Universit`a degli Studi di Napoli Federico II, l’Universit`a degli Studi della Campania Luigi Vanvitelli e l’Universit`a degli Studi di Salerno.

#### IBM Watson Knowledge Studio

##### Import degli Articoli

Come anticipato, inizialmente `e stato utilizzato un set costituito da soli 8 artico- li scaricati manualmente dalle rispettive riviste, attraverso i DOI corrispondenti memorizzati all’interno del database DB2.

In particolare, nella seguente immagine `e possibile osservare gli articoli memo- rizzati all’interno di IBM Watson Knowledge Studio:



Figura 8: Primo set di articoli caricati su IBM Watson Knowledge Studio

##### Creazione del Modello Entit`a-Relazione

Per realizzare il modello di NLP, innanzitutto, `e stato necessario definire le Entit`a e le Relazioni che dovranno essere annotate all’interno degli articoli.

A seguito di un primo confronto con gli esperti di dominio, essi hanno fornito un primo insieme di entit`a e relazioni sufficienti per effettuare un’operazione di pre-annotation[3](#_bookmark28). In particolare, in seguito a tale primo colloquio, sono state definite le seguenti relazioni e le seguenti entit`a (evidenziate in grassetto), con rispettivi sinonimi, abbreviazioni e termini appartenenti a quella data entit`a:

3Un pre-annotator `e un dizionario di Knowledge Studio, un modello Rule-Based oppure un modello di machine learning che pu`o essere eseguito per annotare automaticamente un testo.

# Entità

**TUMOR**

Tumor Neoplasm Malignancy Malignant glioma

Recurrent malignant glioma Progressive malignant glioma Oligodendroglioma Astrocytoma

Anaplastic glioma Anaplastic astrocytoma HGG

GBM

Progressive Glioblastoma multiforme Recurrent Glioblastoma multiforme High grade glioma

WHO grade IV

**Molecular classification** 1p/19q codeletion IDH1 or IDH2 mutation TERT mutation

EGFR amplicons MGMT

MGMT status PTEN mutation ATRT

Peroxisome proliferator-activated receptor gamma

**TRIAL**

Clinical trial registry Registry

Phase I Phase II Phase III Study design Open-label

Exclusion criteria Inclusion criteria

**CHEMOTHERAPY**

Chemotherapy Nimustine Temozolomide Carboplatin Procarbazine Lomustine Vincristine Carmustine Verubulin Etoposide Cyclophosphamide Ifosfamide Teniposide

Interferon

Convention-enhanced delivery CED

Gliadel wafers Carmustine wafers Chemotherapy delivery Intravenous delivery Intra-arterial delivery

**TARGET THERAPY**

Target therapy Anti-VEGF Bevacizumab Anti VEGFR Sorafenib

Anti-EGFR erlotinib

HIV protease inhibitors Ritonavir

Lopinavir

**RADIOTHERAPY**

Radiotherapy Radiation therapy Irradiation

Re-irradiation Gamma Knife

Fractioned radiation therapy Unfractioned radiation therapy Stereotactic radiosurgery

**SURGERY**

Surgery Biopsy

Stereotactic neurosurgery Neuronavigation Inoperable

Operable

Gross total tumor resection Complete tumor resection Partial resection

Residual tumor

**MAGNETIC RESONANCE IMAGING**

Magnetic Resonance Imaging MRI

Enhancement

Contrast-enhancement

Delayed contrast extravasation MRI Treatment Response Assessment Maps TRAMs

Perfusion imaging

Dynamic Susceptibility Contrast-enhanced

DSC

Dynamic Contrast-Enhanced DCE

Susceptibility Weighted Imaging

**RESPONSE TO THERAPY**

Survival Overall survival OS

Progression free survival PFS

Karnofsky status Life quality

Objective response rate ORR

**ADVERSE EFFECT**

Collateral effect Adverse effect Adverse reaction Secondary reaction Dosage

Toxicity Efficacy Tolerance Safety

**RESPONSE ASSESSMENT**

Progression True progression Regression

Complete response Partial response Stable disease Pseudoprogression Pseudoresponse Radionecrosis

Cerebral radiation necrosis

Response assessment in neuro-oncology RANO criteria

iRANO criteria

# Relazioni

Tumor – molecular classification Tumor – chemotherapy

Tumor - TARGET THERAPY

Tumor- radiotherapy Tumor- surgery

Tumor- Magnetic Resonance Imaging Tumor - trial

Trial – chemotherapy Trial- radiotherapy Trial- target therapy Trial-surgery

chemotherapy - response to therapy chemotherapy – response assessment chemotherapy – adverse effect

target therapy – responce to therapy target therapy - response assessment target therapy - adverse effect radiotherapy- response assessment radiotherapy – adverse effect radiotherapy – responce to therapy surgery-response assessment surgery– adverse effect

surgery – responce to therapy

# Sinonimi e abbreviazioni

Neoplasm - malignancy - tumor

Glioblastoma - GBM - High grade glioma - HGG Therapy - treatment

HIV protease inhibitors - HIVPI

Radiotherapy - irradiation - radiation therapy - RT Gross total tumor resection - Complete tumor resection Tolerance - tolerability

Overall Survival - OS Progression free survival - PFS Karnofsky status – KS Objective response rate - ORR Temozolomide - TMZ Nimustine - INN

Interferon – IFN Gliadel – carmustine Reccurent-progressive

CED-convection enhanced delivery

Response assessment in neuro-oncology criteria – RANO Magnetic Resonance Imaging - MRI

Dynamic Susceptibility Contrast-enhanced – DSC Dynamic Contrast-Enhanced – DCE

Treatment Response Assessment Maps - TRAMs

Definito il modello, `e stato definito un Dizionario tale che associasse i termini specificati nell’elenco ricevuto dagli esperti di dominio alle rispettive entit`a.

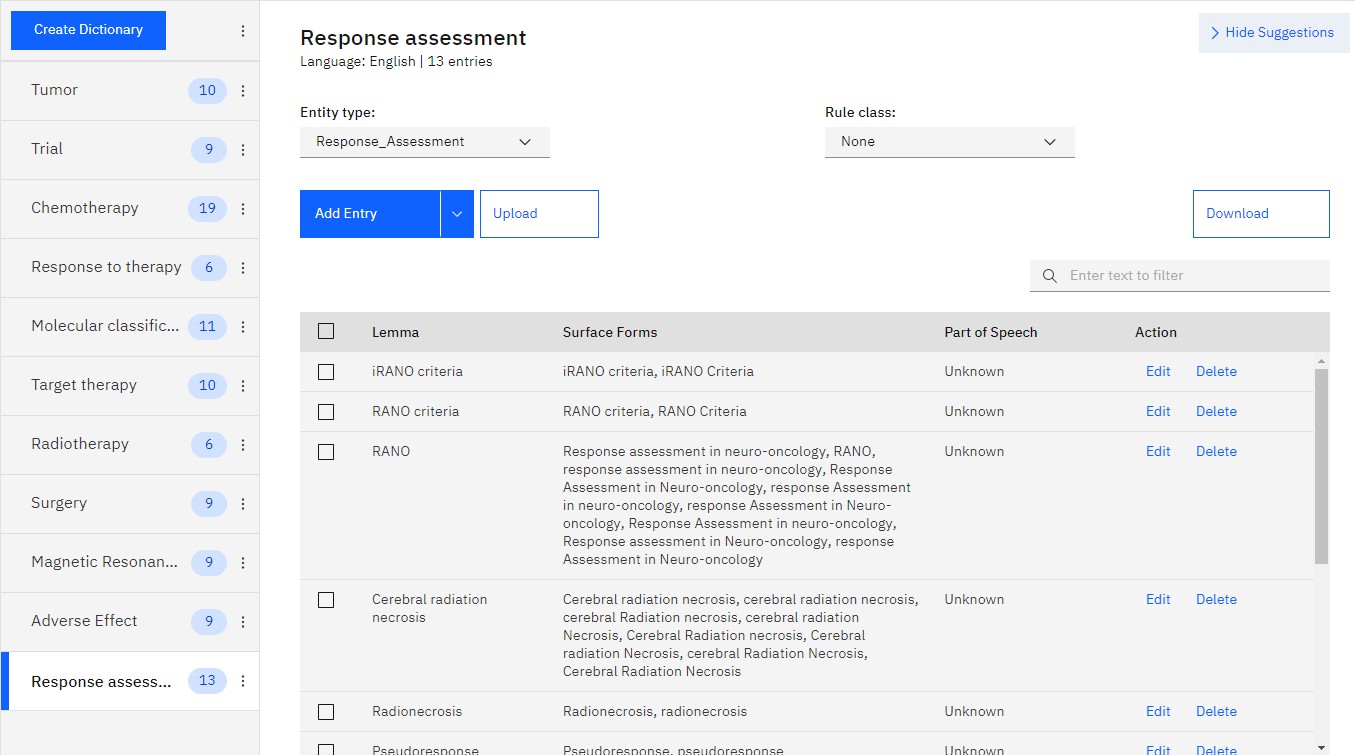


Figura 9: Dizionario definito

Una volta prodotto il Dizionario, `e stata processata la pre-annotazione dei testi precedentemente caricati.

Di seguito `e possibile osservare alcuni esempi di risultati ottenuti sui 10 testi:

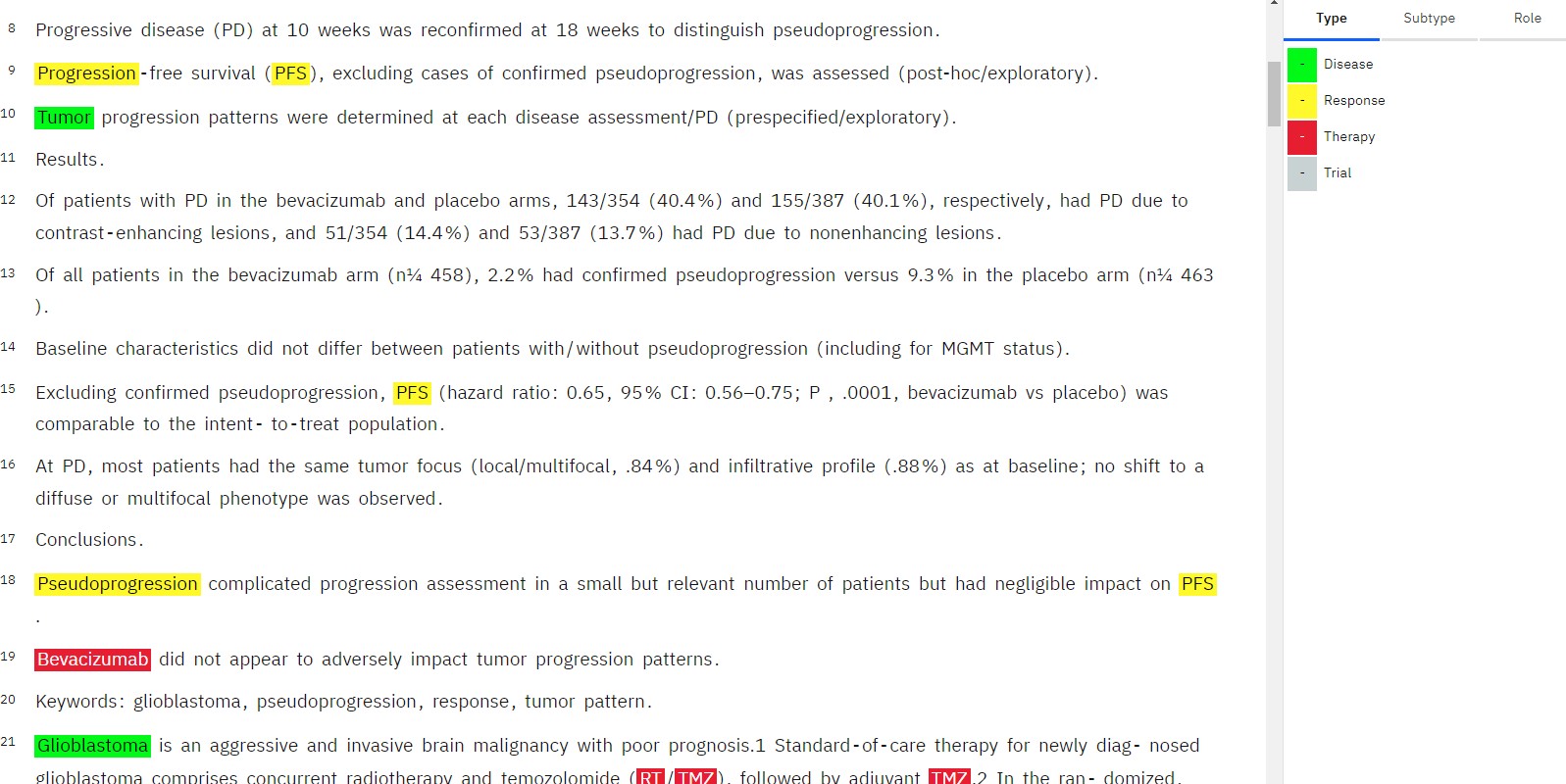


Figura 10: Esempio di annotazione

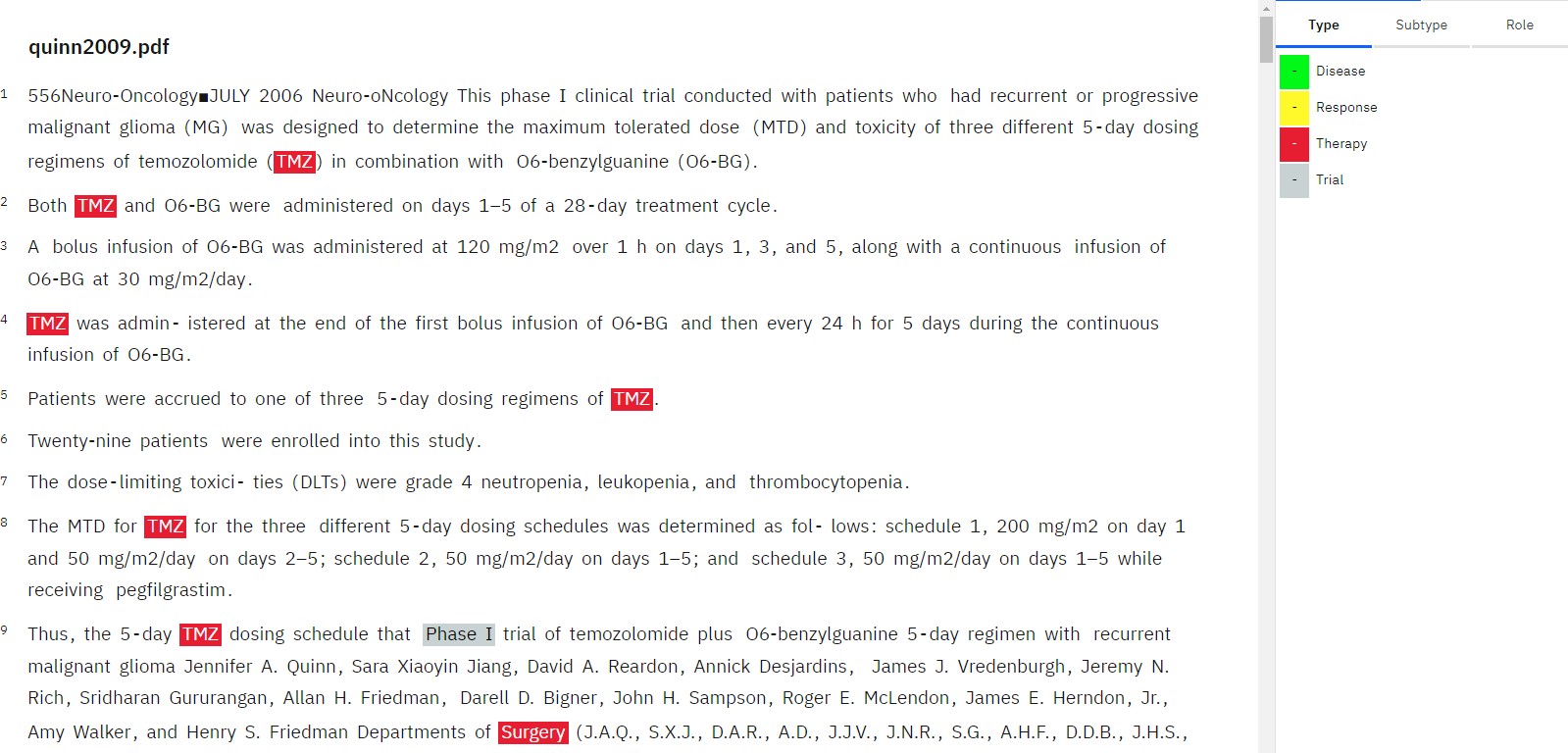


Figura 11: Esempio di annotazione

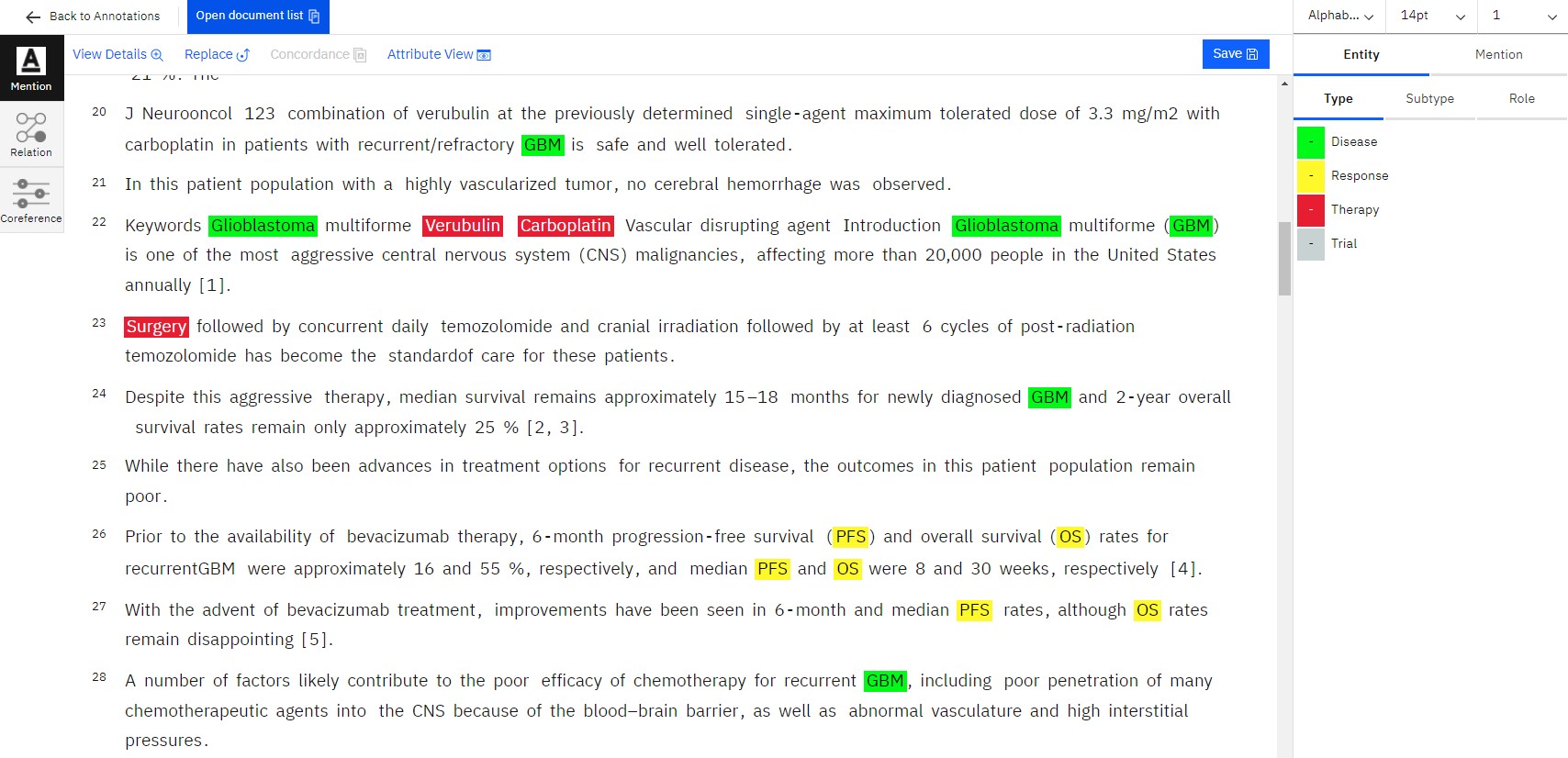


Figura 12: Esempio di annotazione

#### Valutazione del modello

Per valutare il modello prodotto, `e stato inserito un ulteriore pull di 40 articoli presenti nel database ed inseriti all’interno di Knowledge Studio. Una volta caricati, `e stato effettuata la pre-annotazione su di essi e, una volta terminata, si

`e proceduto con l’addestramento del modello e la sua valutazione; per effettuare tale procedura, si `e proceduto con uno split dell’intero set di documenti nel seguente modo:

* Training Set: 70% dell’intero set di documenti
* Test Set: 23% dell’intero set di documenti
* Blind Set: 7% dell’intero set di documenti

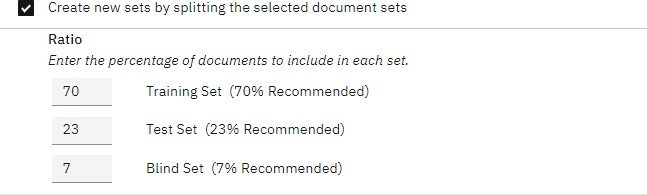


Figura 13: Suddivisione del set di documenti

Si osservi che il numero di document sets differisce dal numero totale di docu- menti. Ci`o `e dovuto alle limitazioni imposte da Knowledge Studio sui documenti caricati: infatti, il tool non effettua il caricamento di documenti di dimensione maggiore di 5MB oppure eccedenti i 40’000 caratteri. Per rientrare nei limiti imposti, abbiamo operato la suddivisione di un articolo in piu` parti, e abbiamo successivamente caricato le parti come un unico document set.

Nello specifico, il Training Set viene utilizzato per addestrare il modello sul- le annotazioni effettuate, il Test Set viene utilizzato dal modello (dopo essersi addestrato tramite il Training) per effettuare una nuova annotazione sui docu- menti che contiene per poi valutare gli errori effettuati essendo questi ultimi gi`a stati pre-annotati in precedenza; il Blind Set, infine, viene utilizzato per testare periodicamente il sistema a valle di diverse iterazioni di testing e miglioramento. Rispetto al test set, il blind set deve contenere elementi che non sono stati visti dagli utenti coinvolti nell’annotazione. Quest’aspetto permette di distinguere il blind set dal test set: quest’ultimo, infatti, contiene documenti che vengono periodicamente esaminati dagli utenti coinvolti nel processo di annotation, e questo tipo di conoscenza pu`o contribuire ad alterare impropriamente il valo- re di precision. Avere un insieme di documenti non visualizzato da tali utenti permette di soffermarsi solo sugli score di alto livello, senza pilotare il miglio- ramento del modello sulla base della conoscenza che gli utenti hanno del test set.

Prodotta la suddivisione, si `e proceduto con l’addestramento del sistema e la sua valutazione tramite le metriche di accuracy, precision, recall e F1-score: l’intera operazione ha richiesto un tempo di 18’14”.

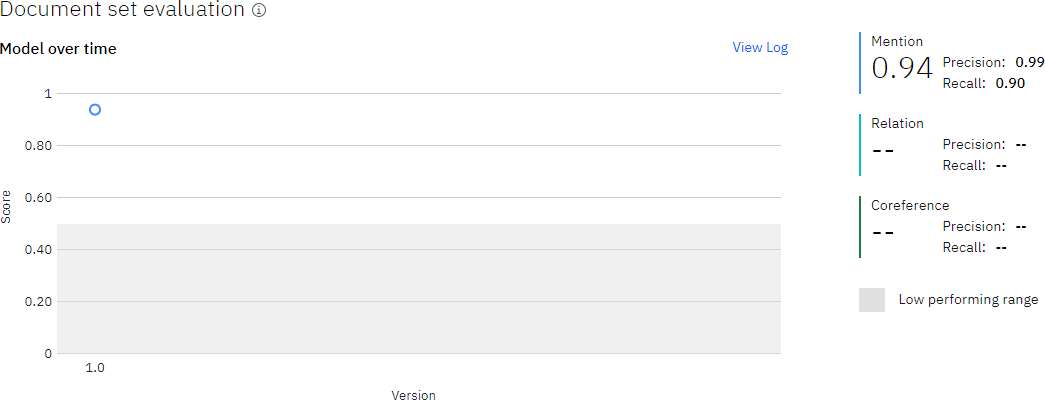


Figura 14: Accuracy, Precision e Recall del modello

Osserviamo che il dizionario `e stato popolato solo da termini appartenenti a determinate entit`a, la valutazione del modello e delle relative metriche `e stata effettuata solo in base all’annotazione delle entit`a, in quanto la pre-annotazione tramite dizionario non consente di individuare automaticamente all’interno dei documenti le relazioni da noi specificate. Il sistema restituisce nel dettaglio i valori delle metriche per ogni entit`a definita nel modello.

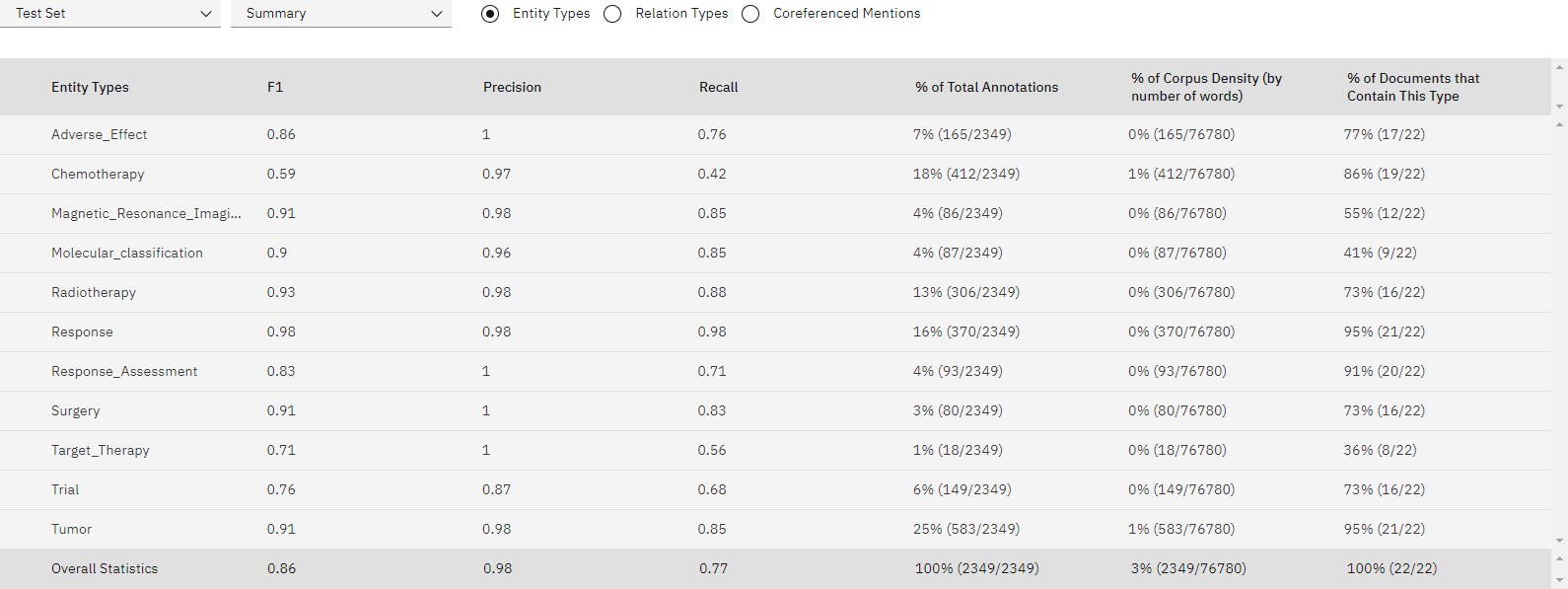


Figura 15: Valori di Accuracy, Precision, Recall e F1-Score delle entit`a

Infine, `e possibile valutare l’accuracy di ogni singola entit`a tramite un istogram- ma che riassume i singoli punteggi ottenuti.

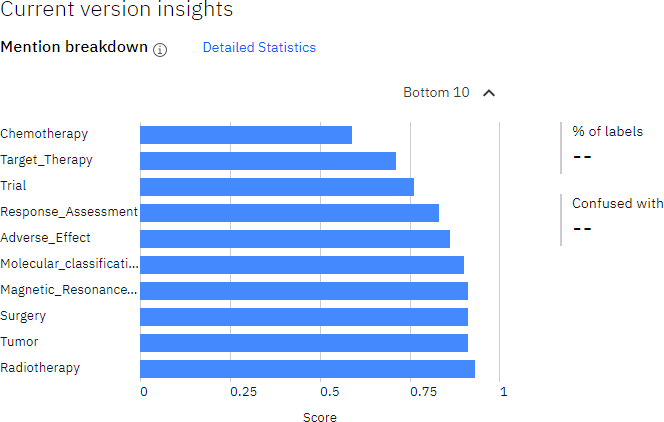


Figura 16: Score di accuracy ottenuto dalle entit`a delle entit`a

## Apprendimento sulle Risonanze Magnetiche tramite Visual Recognition

#### Flusso di esecuzione del sistema di apprendimento sulle MRI

Figura 17: Diagramma di flusso dell’esecuzione

In un primo momento, i file DICOM sono stati sottoposti ad una fase di pre- processing: essi sono stati aperti tramite il tool *RadiAnt DICOM Viewer* e convertiti in formato JPG.

Dopo la fase di pre-processing, le risonanze sono state utilizzate per la realizza- zione di due reti differenti:

* + 1. il tool di Visual Recognition di Watson Studio.
    2. una CNN implementata nella libreria *fastai* ;

Quest’ultima `e stata usata in fase preliminare e come metro di paragone con Watson Visual Recognition. La sua implementazione e utilizzo sono specificati nella Sezione C dell’Appendice del presente documento.

#### Visual Recognition di Watson Studio

L’addestramento del classificatore attraverso Watson Studio `e stato addestrato attraverso diversi cicli di raffinamento che analizzeremo nel dettaglio di seguito:

* + 1. Primo Ciclo: il classificatore `e stato addestrato fornendogli in ingresso un Training set costituito da un totale di 691 immagini, divise in due diverse classi. La prima `e la classe GBM che contiene 489 immagini di glioblastomi multiformi, mentre la seconda `e la classe MET che contiene 202 immagini di metastasi. Osserviamo che, in pratica il Training Set adoperato per realizzare il primo ciclo di addestramento `e, in pratica, lo stesso Training Set utilizzato per addestrare il classificatore con **fastai**;
    2. Secondo Ciclo: il classificatore `e stato addestrato fornendogli in ingresso un Training set costituito da un totale di 1’838 immagini, divise in due

diverse classi ed un set di immagini definite come Negative. In particolare, le classi GBM e MET sono rimaste invariate rispetto al primo ciclo di addestramento, mentre le 1’147 immagini Negative sono immagini che non appartengono a nessuna di esse;

* + 1. Terzo Ciclo: il classificatore `e stato addestrato fornendogli in ingresso un Training set costituito da un totale di 2’103 immagini, divise in due diverse classi ed un set di immagini definite come Negative. In particolare, le immagini definite come Negative sono rimaste invariate, mentre sono state aggiunte 221 immagini classificate come GBM e 44 immagini classificate, invece, come MET;
    2. Quarto Ciclo: il classificatore `e stato addestrato fornendogli in ingresso un Training set costituito da un totale di 956 immagini, divise in due diverse classi. In particolare, le immagini appartenenti alle classi GBM e MET sono le stesse utilizzate nel terzo ciclo di addestramento. Sono state, invece, rimosse le immagini definite come Negative.

Al termine di ogni ciclo di addestramento sono state valutate le prestazioni del classificatore in termini di metriche di **Accuracy**, **Precision**, **Recall** ed **F1- score**, attraverso l’utilizzo di un Test Set costituito da un totale di 327 elementi suddivisi in due classi, quali GBM e MET. In particolare, 223 elementi sono classificati come GBM ed i restanti 104 elementi sono, invece, classificati come MET.

##### Implementazione

Innanzitutto, una prima parte di dati etichettati forniti dagli esperti di dominio

`e stata utilizzata come Training Set con lo scopo di far apprendere il modello.

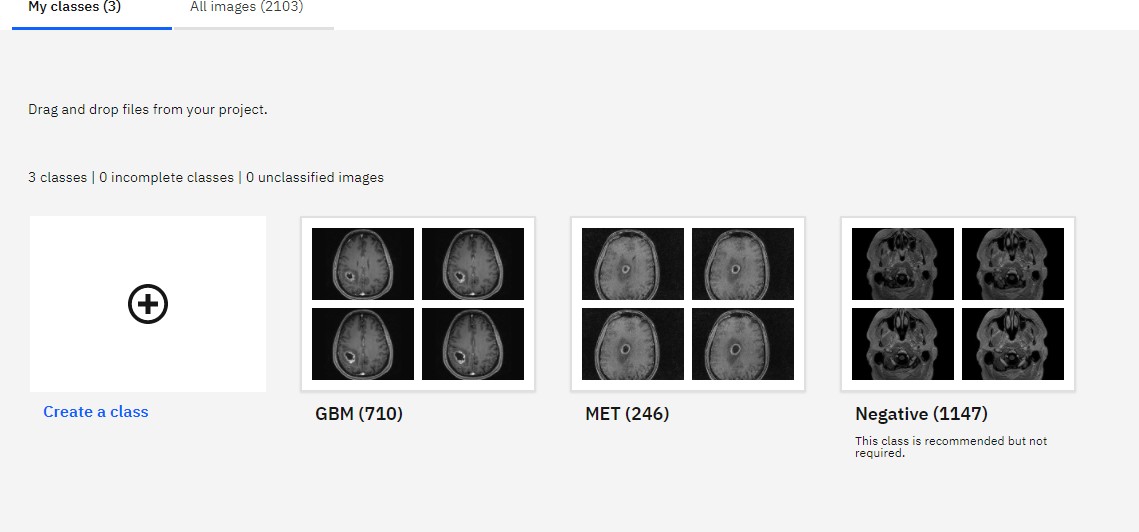


Figura 18: Definizione delle classi su Visual Recognition

Dopodich´e, un secondo insieme di dati etichettati dagli esperti di dominio `e sta- to utilizzato come Test Set al fine di poter valutare la metriche di **Accuracy**, **Precision**, **Recall** ed **F1-score**.

In particolare, `e stato realizzato un notebook in linguaggio Python che, sfrut- tando le IBM Cloud API, ci ha consentito di utilizzare il classificatore prece- dentemente realizzato per effettuare una classificazione del Test Set.

In primo luogo `e stata generata un’istanza di **VisualRecognitionV3**, che con- sente di realizzare il collegamento con l’istanza di Visual Recognition addestrata all’interno di Watson Studio. Ci`o `e stato realizzato attraverso la funzione *crea- teVRinstance(parameters)*, che riceve in ingresso un insieme di parametri spe- cificati all’interno di un file JSON. In particolare, i campi specificati all’interno del file sono i seguenti:

* + - * url;
      * apikey;
      * vr id: id del classificatore da noi istruito;

test csv file: file csv contenente i nomi delle immagini appartenenti al Test Set e le rispettive classificazioni;

•

* + - * results csv file: file csv in cui verranno memorizzati i risultati;

confmatrix csv file: file csv in cui verr`a memorizzata la Confusion Matrix (Matrice di Confusione[4](#_bookmark34)).

•

Le informazioni contenute all’interno del file JSON vengono memorizzate all’interno di un oggetto di tipo Parameters:

**def** read Parameters ( parmsPath ) :

*#Lettura dei parametri u t i l i da un f i l e j son*

vr Parms File = parmsPath parms = ’ ’

with **open** ( vrParms File ) as parmFile : parms = j son . load ( parmFile )

parameters = Parameters ( parms [ ’ url ’ ] , parms [ ’ apikey ’

] , parms [ ’ v r i d ’ ] , parms [ ’ t e s t c s v f i l e ’ ] , parms [ ’ r e s u l t s c s v f i l e ’ ] , parms [ ’ c o n f m a t r i x c s v f i l e ’ ] , parms [ ’ ve rs i o n ’ ] )

j son . dumps( parms )

**return** parameters

4Nell’ambito dell’Intelligenza artificiale, la matrice di confusione, detta anche tabella di errata classificazione, restituisce una rappresentazione dell’accuratezza di classificazione sta- tistica. Ogni colonna della matrice rappresenta i valori predetti, mentre ogni riga rappresenta i valori reali. L’elemento sulla riga i e sulla colonna j `e il numero di casi in cui il classificatore ha classificato la classe ”vera” i come classe j. Attraverso questa matrice `e osservabile se vi `e ”confusione” nella classificazione di diverse classi.

I parametri appena acquisiti, come anticipato, vengono utilizzati per la creazione di un’istanza di Visual Recognition, ovvero un’istanza della classe **VisualReco- gnitionV3**, definita all’interno della libreria **ibm watson**. L’implementazione della funzione *createVRinstance(parameters)* `e riportata di seguito:

**def** create VRinstance ( parameters ) :

*#Creazione d i una i s t a n z a d i Visual Recognition*

authe nti c ato r = IAMAuthenticator ( parameters . apiKey ) vi s ual Re c o g ni t i o n = Visual Recognition V 3 (

v e rs i o n = parameters . version , authe nti c ato r = authe nti c ato r

)

vi s ual Re c o g ni t i o n . s e t s e r v i c e u r l ( parameters . url )

**return** vi s ual Re c o g ni t i o n

Dopo la generazione dell’istanza della classe **VisualRecognitionV3**, `e possi- bile effettuare la classificazione delle immagini contenute all’interno del Test Set. A tal fine, innanzitutto `e stata definita la funzione *getVRresponse* che va a classificare una singola immagine restituendo una lista delle classi associate all’immagine stessa, ordinata in base al valore di Confidence (score). La funzio- ne riceve in ingresso 3 parametri, quali l’istanza di visual recognition, l’id del classificatore ed il path dell’immagine da classificare. L’implementazione della funzione `e riportata di seguito:

**def** getVRresponse ( vr i ns tanc e , c l a s s i f i e r I D , image path ) : with **open** ( image path , ’ rb ’ ) as i m a g e s f i l e :

i m ag e re s ul t s = v r i ns tan c e . c l a s s i f y ( i m a g e s f i l e = i m a g e s f i l e ,

thre s ho l d=’ 0 . 6 ’ ,

c l a s s i f i e r i d s=c l a s s i f i e r I D ) . g e t r e s u l t ()

*#Ogni s i n g o l a immagine v iene c l a s s i f i c a t a a t t ra v e r s o i l c l a s s i f i c a t o r e da noi i s t r u i t o*

c l a s s L i s t = [ ]

**for** c l a s s i f i e r **in** i m ag e re s ul t s [ ’ images ’ ] [ 0 ] [ ’ c l a s s i f i e r s ’ ] :

**i f** c l a s s i f i e r [ ’ c l a s s i f i e r i d ’ ] == parameters . vr Id

:

c l a s s L i s t = c l a s s i f i e r [ ’ c l a s s e s ’ ]

##### break

*#Le c l a s s i vengono ordinate in base a l l o score*

s o r t e d c l a s s L i s t = **sorted** ( c l a s s L i s t , key=**lambda** k : k . get ( ’ s c o re ’ , 0 ) , re v e rs e=True )

**return** s o r t e d c l a s s L i s t

Sfruttando la funzione appena definita, `e stato possibile definire un’ulteriore funzione *batchVR*, che consente di classificare un batch di immagini restituendo,

per ciascuna di esse, i risultati ottenuti. In particolare, i path delle immagini appartenenti al batch da classificare vengono specificati all’interno di un file **csv**, insieme alla classe di appartenenza dell’immagine stessa. La funzione riceve in ingresso 3 parametri, quali l’istanza di visual recognition, l’id del classificatore ed il path del file csv. L’implementazione della funzione `e riportata di seguito:

**def** batchVR( vr i ns tanc e , c l a s s i f i e r I D , c s v f i l e ) : t e s t C l a s s e s =[]

v r Pre di c t Cl as s e s =[]

v r p r e d i c t c o n f i d e n c e =[] images =[]

i =0

with **open** ( c s v f i l e , ’ r ’ ) as c s v f i l e :

csv Reader=csv . DictReader ( c s v f i l e )

**for** row **in** csv Reader :

t e s t C l a s s e s . append ( row [ ’ c l a s s ’ ] )

vr re s po ns e = getVRresponse ( vr i ns tanc e , c l a s s i f i e r I D , row [ ’ image ’ ] )

v r Pre di c t Cl as s e s . append ( vr re s po ns e [ 0 ] [ ’ c l a s s ’ ] )

v r p r e d i c t c o n f i d e n c e . append ( vr re s po ns e [ 0 ] [ ’ s c o re ’ ] )

images . append ( row [ ’ image ’ ] ) i = i +1

**i f** ( i %250 == 0 ) :

**print** ( ”” )

**print** ( ” Processed ” , i , ” re c o rds ” )

**i f** ( i %10 == 0 ) :

sys . stdout . write ( ’ . ’ )

**print** ( ”” )

**print** ( ” Finished pro c e s s i ng ” , i , ” re c o rds ” )

**return** te s t Cl as s e s , vr Predict Cl asses , vr Predict Confidence , images

##### Risultati ottenuti - Primo Ciclo

Il tempo impiegato per effettuare il Training del classificatore dato il Training Set descritto precedentemente (che consta di 691 immagini) `e pari a 36’17”.

Le metriche di **Accuracy**, **Precision**, **Recall** ed **F1-score** sono state stimate attraverso l’utilizzo delle funzioni *accuracy score* e *classification report*, entram- be definite all’interno della libreria **sklearn**.

Di seguito sono riportati i risultati ottenuti:

accuracy = ac c urac y s c o re ( te s t Cl as s e s , v r Pre di c t Cl as s e s )

**print** ( ’ C l a s s i f i c a t i o n Accuracy : ’ , accuracy )



Figura 19: Accuracy della classificazione ottenuta

c l a s s i f i c a t i o n r e p o r t ( te s t Cl as s e s , vr Predi ct Cl asses , l a b e l s=l a b e l s )

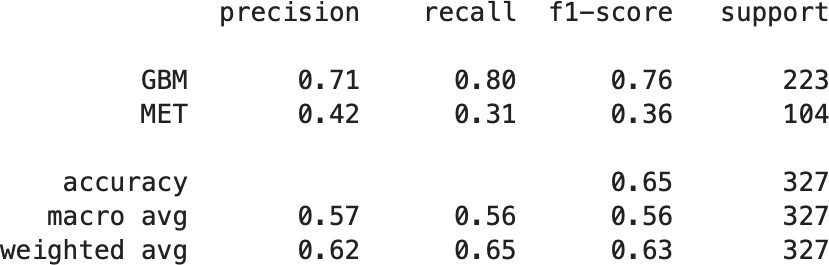


Figura 20: Valori di Precision, Recall e F1-Score

Inoltre, `e stata generata e memorizzata all’interno di un file **csv** la matrice di confusione attraverso la funzione *confusion matrix*, anch’essa definita all’interno della libreria **sklearn**:

l a b e l s=**l i s t** ( **set** ( t e s t C l a s s e s ) )

vr Confusion Matrix = conf usion matrix ( te s t Cl as s e s , vr Predict Cl asses , l a b e l s )

confmatrix 2 csv ( vrConfusion Matrix , l abe l s , parameters . conf Matrix Csv File )

La matrice di confusione ottenuta `e, quindi, la seguente:

df cm = pd . DataFrame ( vrConf Matrix , **range** ( 2 ) , **range** ( 2 ) ) sn . **set** ( f o n t s c a l e =1.4) . *# Label s i z e*

sn . heatmap ( df cm , annot=True , annot kws= ” s i z e ” : 16 )

{ }

*# Font s i z e*

pl t . show ()

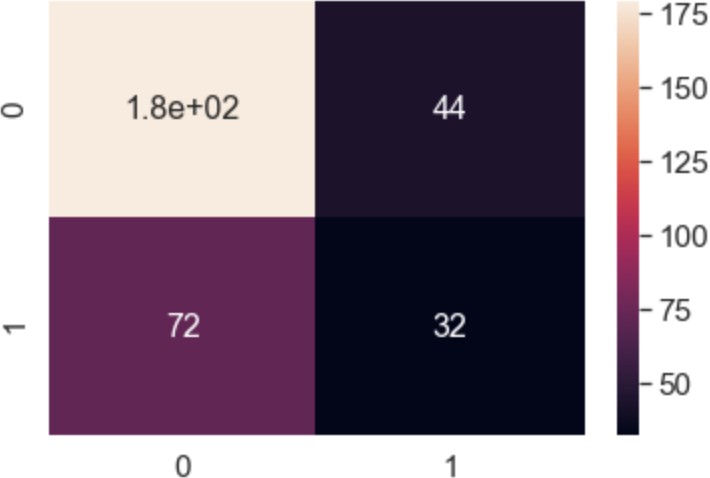


Figura 21: Matrice di Confusione

Infine, tale matrice `e stata memorizzata all’interno di un file **csv** attraverso la funzione *confmatrix2csv*, la cui implementazione `e riportata di seguito:

**def** confmatrix 2 csv ( conf matrix , l abe l s , c s v f i l e ) : with **open** ( c s v f i l e , ’w ’ ) as c s v f i l e :

csv Writer = csv . w r i te r ( c s v f i l e ) row=**l i s t** ( l a b e l s )

row . i n s e r t ( 0 , ”” )

csv Writer . writerow ( row )

**for** i **in range** ( conf matrix . shape [ 0 ] ) : row=**l i s t** ( conf matrix [ i ] )

row . i n s e r t ( 0 , l a b e l s [ i ] ) csv Writer . writerow ( row )

##### Risultati ottenuti - Secondo Ciclo

Il tempo impiegato per effettuare il Training del classificatore dato il Training Set descritto precedentemente (che consta di 1’838 immagini) `e pari a 1h56’59”. Le metriche di **Accuracy**, **Precision**, **Recall** ed **F1-score** sono state stimate attraverso l’utilizzo delle funzioni *accuracy score* e *classification report*, entram- be definite all’interno della libreria **sklearn**.

Di seguito sono riportati i risultati ottenuti:

accuracy = ac c urac y s c o re ( te s t Cl as s e s , v r Pre di c t Cl as s e s )

**print** ( ’ C l a s s i f i c a t i o n Accuracy : ’ , accuracy )



Figura 22: Accuracy della classificazione ottenuta

c l a s s i f i c a t i o n r e p o r t ( te s t Cl as s e s , vr Predi ct Cl asses , l a b e l s=l a b e l s )

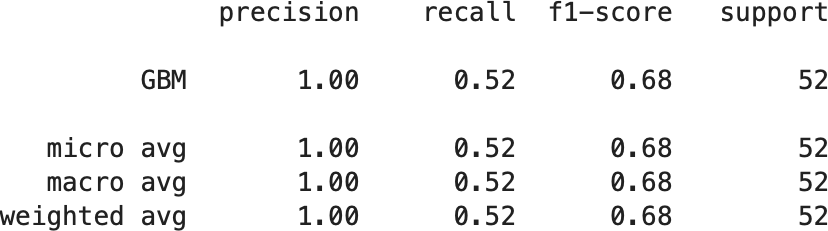


Figura 23: Valori di Precision, Recall e F1-Score

Osserviamo, quindi, che c’`e stato un calo dell’accuracy del classificatore di circa il 20% rispetto al primo ciclo di addestramento. Tale diminuzione `e una con- seguenza del fatto che il classificatore non sia piu` stato in grado di riconoscere le metastasi (immagini appartenenti alla classe MET). Infatti, tutte le imma- gini classificate come MET appartenenti al Test Set sono state erroneamente riconosciute come Negative.

Inoltre, `e stata generata e memorizzata all’interno di un file **csv** la matrice di confusione attraverso la funzione *confusion matrix*, anch’essa definita all’interno della libreria **sklearn**:

l a b e l s=**l i s t** ( **set** ( t e s t C l a s s e s ) )

vr Confusion Matrix = conf usion matrix ( te s t Cl as s e s , vr Predict Cl asses , l a b e l s )

confmatrix 2 csv ( vrConfusion Matrix , l abe l s , parameters . conf Matrix Csv File )

La matrice di confusione ottenuta `e, quindi, la seguente:

df cm = pd . DataFrame ( vrConf Matrix , **range** ( 2 ) , **range** ( 2 ) ) sn . **set** ( f o n t s c a l e =1.4) . *# Label s i z e*

sn . heatmap ( df cm , annot=True , annot kws= ” s i z e ” : 16 )

{ }

*# Font s i z e*

pl t . show ()

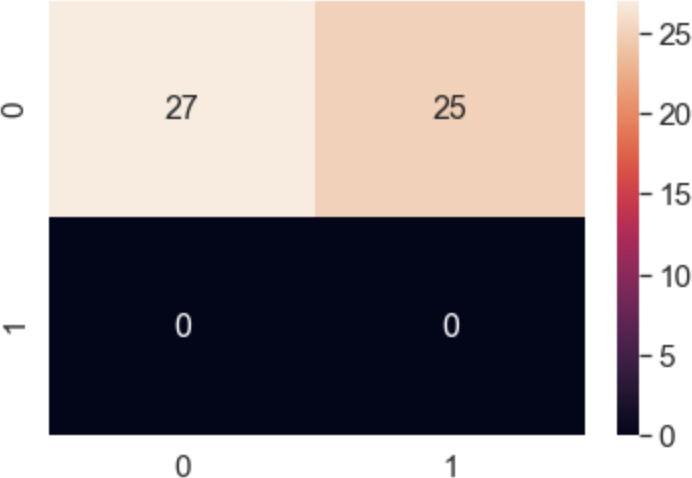


Figura 24: Matrice di Confusione

Dalla matrice `e possibile osservare come, appunto, il classificatore non abbia classificato affatto le immagini appartenenti alla classe MET che, invece, sono state riconosciute tutte come Negative.

Infine, tale matrice `e stata memorizzata all’interno di un file **csv** attraverso la funzione *confmatrix2csv*.

##### Risultati ottenuti - Terzo Ciclo

Il tempo impiegato per effettuare il Training del classificatore dato il Training Set descritto precedentemente (che consta di 2’103 immagini) `e pari a 2h0’31”. Le metriche di **Accuracy**, **Precision**, **Recall** ed **F1-score** sono state stimate attraverso l’utilizzo delle funzioni *accuracy score* e *classification report*, entram- be definite all’interno della libreria **sklearn**.

Di seguito sono riportati i risultati ottenuti:

accuracy = ac c urac y s c o re ( te s t Cl as s e s , v r Pre di c t Cl as s e s )

**print** ( ’ C l a s s i f i c a t i o n Accuracy : ’ , accuracy )



Figura 25: Accuracy della classificazione ottenuta

c l a s s i f i c a t i o n r e p o r t ( te s t Cl as s e s , vr Predi ct Cl asses , l a b e l s=l a b e l s )

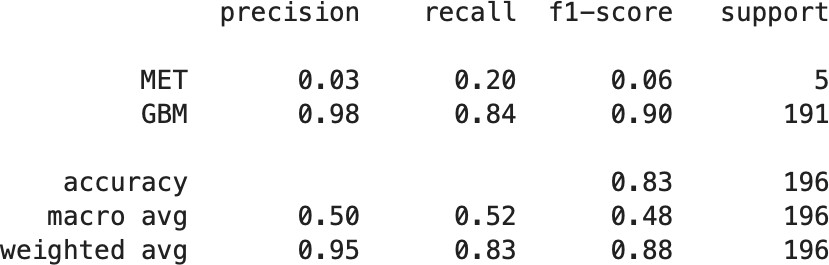


Figura 26: Valori di Precision, Recall e F1-Score

Inoltre, `e stata generata e memorizzata all’interno di un file **csv** la matrice di confusione attraverso la funzione *confusion matrix*, anch’essa definita all’interno della libreria **sklearn**:

l a b e l s=**l i s t** ( **set** ( t e s t C l a s s e s ) )

vr Confusion Matrix = conf usion matrix ( te s t Cl as s e s , vr Predict Cl asses , l a b e l s )

confmatrix 2 csv ( vrConfusion Matrix , l abe l s , parameters . conf Matrix Csv File )

La matrice di confusione ottenuta `e, quindi, la seguente:

df cm = pd . DataFrame ( vrConf Matrix , **range** ( 2 ) , **range** ( 2 ) ) sn . **set** ( f o n t s c a l e =1.4) . *# Label s i z e*

sn . heatmap ( df cm , annot=True , annot kws= ” s i z e ” : 16 )

{ }

*# Font s i z e*

pl t . show ()

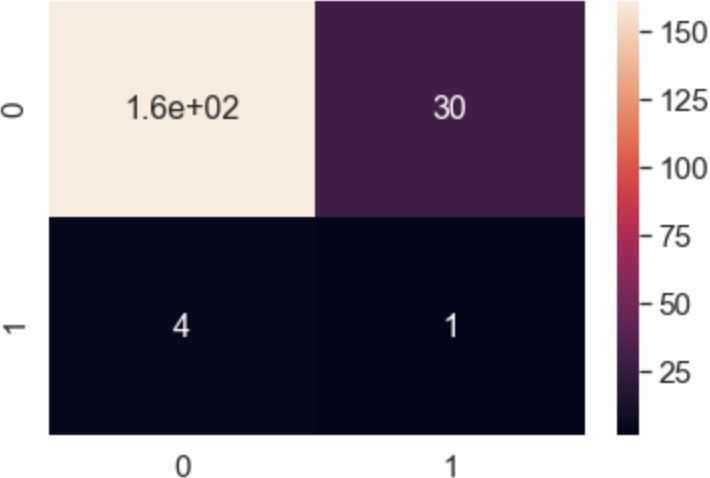


Figura 27: Matrice di Confusione

Infine, tale matrice `e stata memorizzata all’interno di un file **csv** attraverso la funzione *confmatrix2csv*.

##### Risultati ottenuti - Quarto Ciclo

Il tempo impiegato per effettuare il Training del classificatore dato il Training Set descritto precedentemente (che consta di 956 immagini) `e pari a 1h04’28”. Le metriche di **Accuracy**, **Precision**, **Recall** ed **F1-score** sono state stimate attraverso l’utilizzo delle funzioni *accuracy score* e *classification report*, entram- be definite all’interno della libreria **sklearn**.

Di seguito sono riportati i risultati ottenuti:

accuracy = ac c urac y s c o re ( te s t Cl as s e s , v r Pre di c t Cl as s e s )

**print** ( ’ C l a s s i f i c a t i o n Accuracy : ’ , accuracy )



Figura 28: Accuracy della classificazione ottenuta

c l a s s i f i c a t i o n r e p o r t ( te s t Cl as s e s , vr Predi ct Cl asses , l a b e l s=l a b e l s )

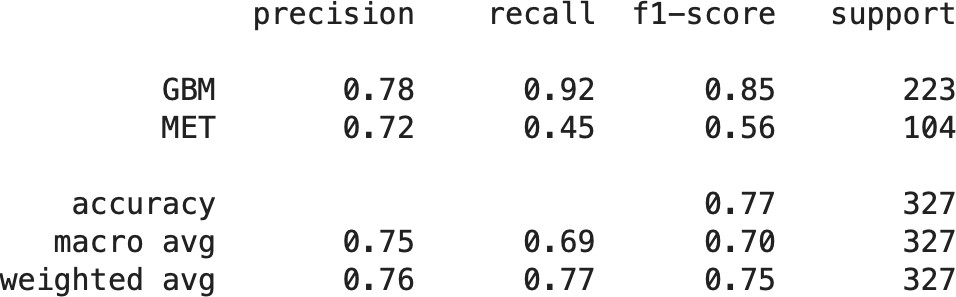


Figura 29: Valori di Precision, Recall e F1-Score

Inoltre, `e stata generata e memorizzata all’interno di un file **csv** la matrice di confusione attraverso la funzione *confusion matrix*, anch’essa definita all’interno della libreria **sklearn**:

l a b e l s=**l i s t** ( **set** ( t e s t C l a s s e s ) )

vr Confusion Matrix = conf usion matrix ( te s t Cl as s e s , vr Predict Cl asses , l a b e l s )

confmatrix 2 csv ( vrConfusion Matrix , l abe l s , parameters . conf Matrix Csv File )

La matrice di confusione ottenuta `e, quindi, la seguente:

df cm = pd . DataFrame ( vrConf Matrix , **range** ( 2 ) , **range** ( 2 ) ) sn . **set** ( f o n t s c a l e =1.4) . *# Label s i z e*

sn . heatmap ( df cm , annot=True , annot kws= ” s i z e ” : 16 )

{ }

*# Font s i z e*

pl t . show ()

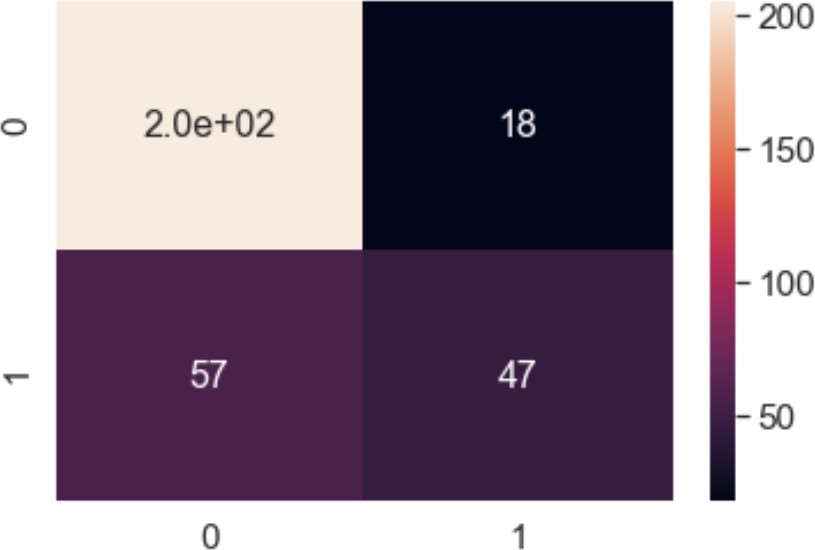


Figura 30: Matrice di Confusione

Infine, tale matrice `e stata memorizzata all’interno di un file **csv** attraverso la funzione *confmatrix2csv* :

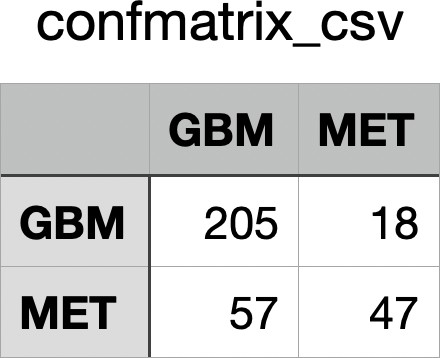


Figura 31: File CSV contenente la Matrice di Confusione

## Considerazioni e Sviluppi Futuri

Il modello di Visual Recognition fornito da IBM risulta essere molto efficiente e rapido nell’addestramento su grosse moli di dati di training; dopo un confron- to con gli esperti di dominio, `e emerso che il lasso di tempo per addestrare il classificatore sul set di 2’103 (ovvero la cardinalit`a del dataset maggiore su cui abbiamo lavorato) pari a circa 2h risulta essere inferiore al tempo che impieghe- rebbe un essere umano ad analizzare lo stesso campione di dati.

Infatti, come `e possibile osservare dal seguente grafico, i tempi impiegati per ef- fettuare l’addestramento del classificatore sono pressoch´e lineari con l’aumentare dei dati presenti all’interno del Training Set:

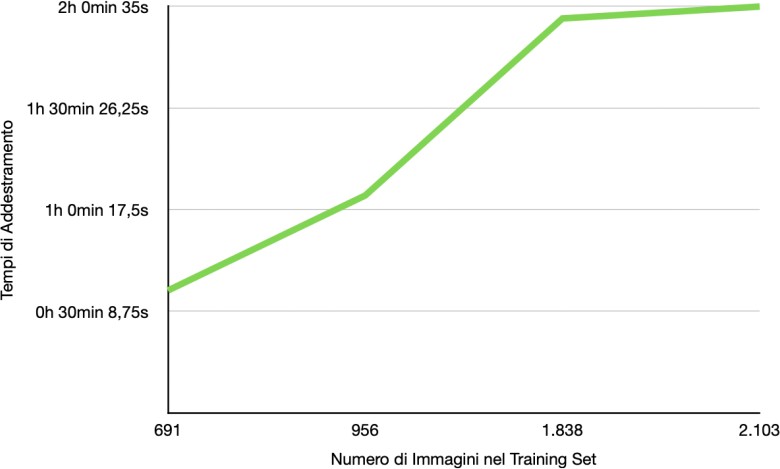


Figura 32: Grafico sull’andamento dei tempi di addestramento del classificatore

Le criticit`a emerse durante l’addestramento e la valutazione del modello ri- guardano principalmente l’utilizzo di immagini di risonanze ”neutre” (inserite nel classificatore come *Negative*) che inducevano il modello a classificare erro- neamente diverse immagini inerenti alle Metastasi: queste venivano identificate come Negativi, quindi non appartenenti alla classe Glioblastoma (*GBM* ) n´e alla classe Metastasi (*MET* ), e dunque non venivano utilizzate per il calcolo delle metriche valutative del modello. Inoltre, la poca trasparenza nella documenta- zione fornita da IBM sugli algoritmi di Deep Learning usati, non ci ha potuto permettere di usare una strategia precisa su come procedere per il superamento della criticit`a sopracitata se non tramite un procedimento di ”trial and error”, ovvero generando un modello che presentasse solamente immagini appartenenti alle classi GBM e MET rimuovendo i Negative. Seppur inizialmente questo tipo modello non ci aveva fornito risultati soddisfacenti, soprattutto per la classifi- cazione delle Metastasi, aumentando la cardinalit`a del training set al numero finale di 956 immagini complessive (710 GBM, 246 MET) il modello ha mostra- to risultati soddisfacenti sulla classificazione di entrambe le classi.

Per poter migliorare ancora il modello, bisognerebbe reperire ulteriori risonanze

di Metastasi con cui indurre nuovi addestramenti incrementali del sistema, oltre che valutarne l’efficienza fornendogli set di risonanze prese da piu` piani (tutte le immagini fornite erano risonanze T1 con mezzo di contrasto sullo stesso piano) e con spessori diversi.

Il modello di annotazione dei documenti creato su Watson Knowledge Studio ha permesso un’individuazione facile dei termini necessari per l’analisi e lo stu- dio dei testi inerenti al Glioblastoma; tuttavia, la difficolt`a nel reperire tutti gli articoli individuati dalla query elaborata insieme agli esperti di dominio e la limitazione di Knowledge Studio nel poter operare su file contenenti un massimo di 40’000 caratteri ha richiesto un lungo e dispendioso lavoro di pre-processing che ha inficiato la possibilit`a di poter operare insieme agli esperti di dominio per un tagging manuale delle entit`a e delle relazioni presenti all’interno dei testi utilizzati al fine di migliorare ulteriormente le performance del sistema (nono- stante risultino essere elevate anche con la sola pre-annotazione) e individuare termini non presenti nel dizionario fornito. Una piu` profonda fase di tagging permetterebbe di poter elaborare una futura strategia di analisi dei testi in mo- do da poter superare i metodi precedentemente utilizzati in altri studi, e forniti a noi come punto di partenza per la creazione del prototipo esposto in questa documentazione. Inoltre, sarebbe ideale elaborare una strategia in grado di ov- viare ad eventuali criticit`a dell’analisi del testo, quali relazioni tra periodi diversi nello stesso articolo, simboli e segni di interpunzione che potrebbero indurre in errore il sistema durante l’analisi del testo. Ci`o pu`o avvenire tramite una stretta collaborazione e confronto con gli esperti di dominio.

**Appendice**

## RadiAnt

RadiAnt DICOM Viewer `e un visualizzatore di file DICOM sviluppato dalla compagnia Medixant. Il suo primo rilascio, in versione Beta, risale al 2010 con la versione 0.1, ad oggi `e arrivato alla versione 2020.1 rilasciato il 6 Aprile 2020. Attraverso un’interfaccia grafica user-friendly, RadiAnt fornisce strumenti per lo studio delle immagini DICOM quali:

* + Zoom e spostamento;
  + Regolazione della luminosit`a e del contrasto;
  + Rotazione e capovolgimento delle immagini;
  + Conversione in formati BMP e JPG.

Il software supporta l’apertura di file ottenuti da differenti modalit`a di immagini quali:

* + Radiografia Digitale (CR, DX)
  + Mammografia (MG)
  + Tomografia Computerizzata (CT)
  + Risonanza Magnetica (MR)
  + Tomografia ad Emissione di Positroni PET-CT (PT)
  + Ecografia (US)
  + Angiografia Digitale (XA)
  + Gamma Camera, Medicina Nucleare (NM)
  + Immagini secondarie e immagini digitalizzate (SC)
  + Structured Reports (SR)

RadiAnt fornisce numerose funzionalit`a, tra cui alcune da noi sfruttate, quali:

##### Confronto di serie o studi differenti

•

E` possibile aprire contemporaneamente piu` serie di un determinato stu- dio oppure di studi differenti nelle stesse o in diverse finestre al fine di effettuare una comparazione fra essi.

##### Ricostruzioni Multiplanari

•

Strumento in gradi di ricostruire immagini su piani ortogonali differenti (coronali, sagittali, obliqui, assiali).

##### Esportazione di ftles DICOM come immagini e ftlmati

•

E` possibile esportare i file DICOM aperti sia come immagini nei formati JPG e BMP sia come video in formato WMV (Windows Media Video) e MP4.

##### Rendering del Volume 3D

•

Strumento che consente di visualizzare grandi volumi di dati generati dagli scanner CT/MR in uno spazio tridimensionale.

#### A.1 Pre-processing effettuato tramite RadiAnt

L’utilizzo di RadiAnt `e risultato fondamentale nella fase di pre-processing dei file DICOM prima che venissero dati in pasto al classificatore dell’istanza di IBM Visual Recognition; il software permette la lettura di intere cartelle di file DICOM, unendole in singole sequenze.

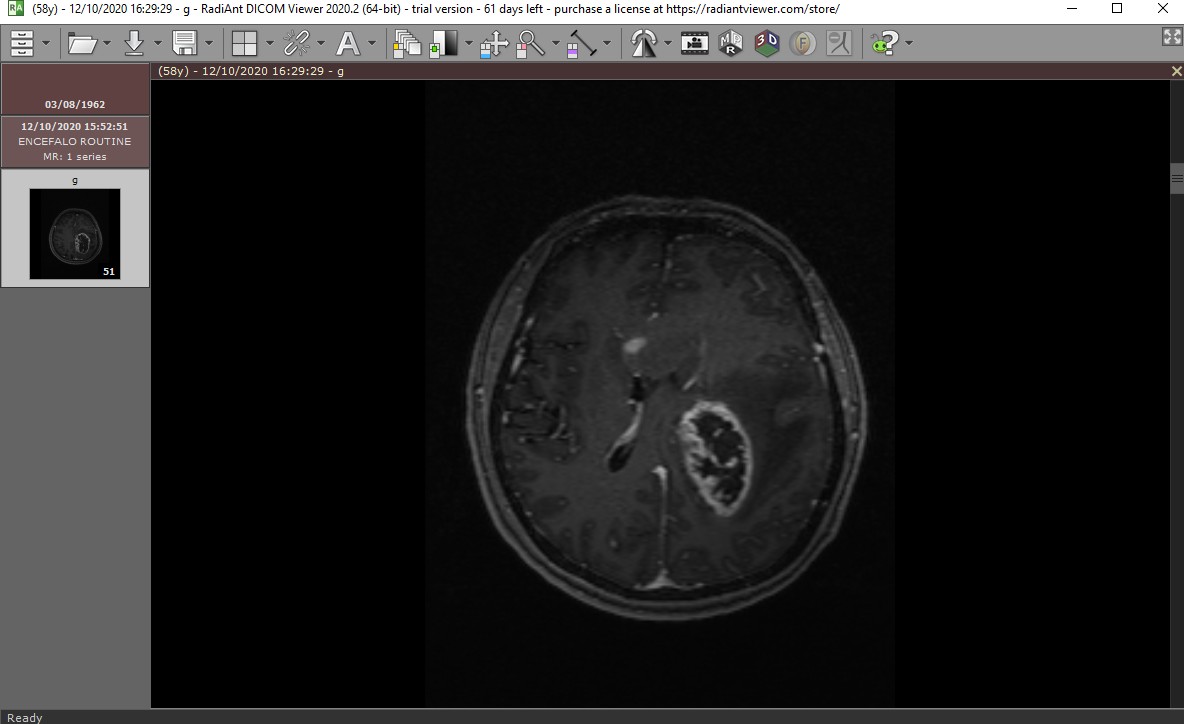


Figura 33: Apertura di una singola cartella contenente una sequenza composta da 52 file DICOM

E` anche possibile caricare nel software piu` cartelle contemporaneamente in modo da poter visualizzare contemporaneamente piu` sequenze.

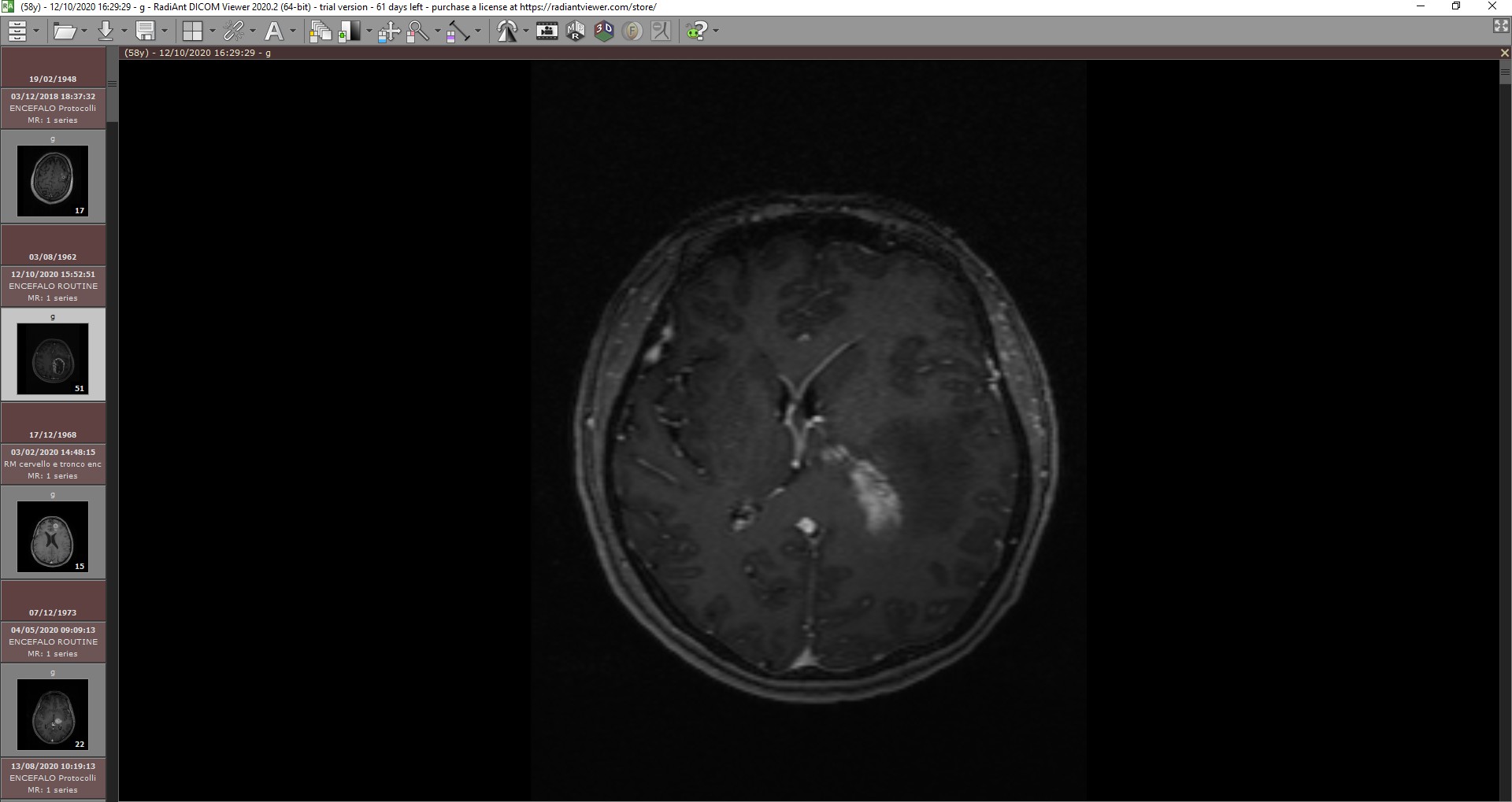


Figura 34: Apertura di piu` sequenze

Una volta aperte le MRI di nostro interesse, `e stata utilizzata la funzionalit`a di conversione in JPG.

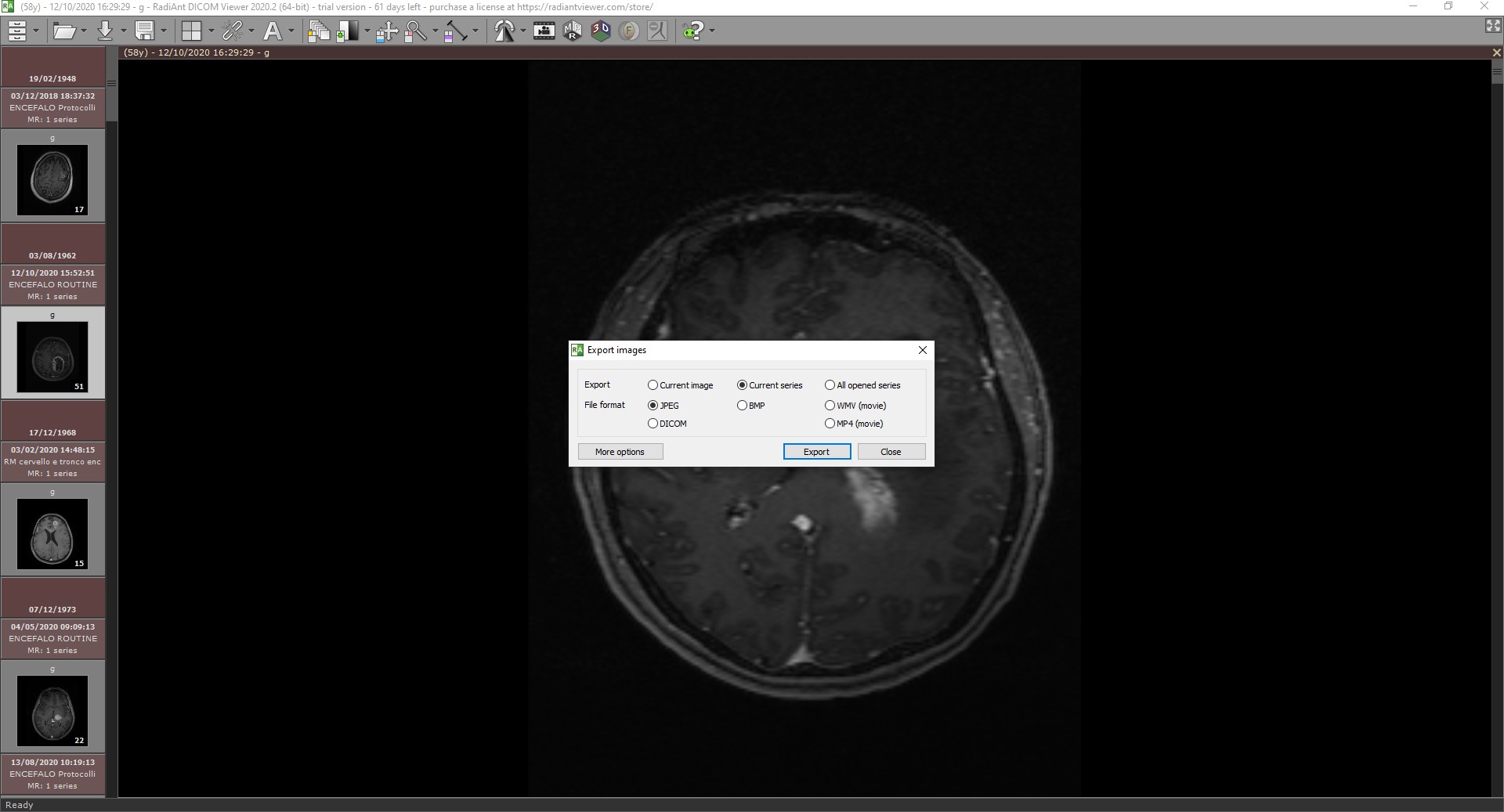


Figura 35: Conversione ed Export in JPG

Le immagini delle risonanze in JPG sono state compresse in un file ZIP in modo da poter caricare l’intero set di sequenze nel Visual Recognition.

## Evaluation Metrics

#### Accuracy

L’Accuracy, nell’ambito del Machine Learning, pu`o essere definita come il rap- porto tra il numero di elementi predetti correttamente ed il supporto (ovvero il numero totale di elementi).

#### Precision e Recall

In un processo di classificazione, Precision e Recall vengono calcolate in termini di Veri Positivi, Falsi Positivi, Veri Negativi e Falsi Negativi, effettuando un confronto tra la classe assegnata all’oggetto dal classificatore stesso e quella a cui realmente esso appartiene.

Formalmente, Precision e Recall sono definite come:

*V eri Positivi V eri P ositivi* + *Falsi P ositivi*

*Precision* =

*V eri Positivi V eri P ositivi* + *Falsi Negativi*

*Recall* =

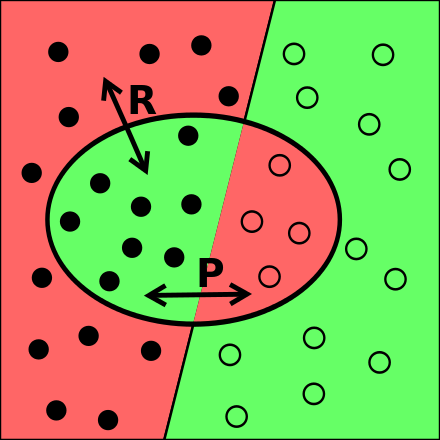
Ad esempio, si pu`o considerare la seguente figura:

Figura 36: Precision e Recall

In essa gli elementi rilevanti sono tutti quelli che si trovano alla sinistra della retta, mentre gli elementi ritrovati sono quelli appartenenti all’ovale. Le regioni colorate di rosso rappresentano gli errori. In particolare, alla regione rossa di

destra (all’interno dell’ovale) appartengono gli elementi recuperati che, per`o, non sono rilevanti, ovvero i Falsi Positivi. Invece, alla regione rossa di sinistra (all’esterno dell’ovale) appartengono gli elementi rilevanti che non sono stati recuperati, ovvero i Falsi Negativi. Precision e Recall rappresentano il quoziente dell’area della regione verde con, rispettivamente, l’ovale (freccia orizzontale) e la regione di sinistra (freccia diagonale).

#### F1-Score

Nell’analisi statistica della classificazione binaria, l’F1 score (nota anche come F-score o F-measure, letteralmente ”misura F”) `e una misura dell’accuratezza di un test che tiene in considerazione precision e recall del test. L’F1 viene calcolato tramite la media armonica di precision e recall:

*F*1 − *Score* =

2

1 1

+

*R P*

= 2 ∗ *R* ∗ *P*

dove P e R indicano rispettivamente Precision e Recall La formula generale `e:

*P* + *R*

*Fβ Score* = (1 + *β*2)  *P* ∗ *R*

− ∗

(*β*2 ∗ *P* )+*R*

per valori di *β* reali positivi. Altre istanze di questa formula solitamente utiliz- zate sono la *F*2 (che pone maggiore enfasi sui falsi negativi) e la *F*0*.*5 (la quale attenua l’influenza dei falsi negativi).

L’F-score `e solitamente usata nel campo del recupero dell’informazione per mi- surare l’accuratezza delle ricerche o della classificazione dei documenti. Inizial- mente l’F1 score era l’unica misura ad essere considerata, ma con la prolifera- zione in larga scala di motori di ricerca gli obiettivi di prestazione iniziarono a variare, divenendo necessario porre maggiore enfasi su precisione o recupero.

L’F-score `e usata anche nel campo dell’apprendimento automatico ed `e vasta- mente impiegata nella letteratura sull’elaborazione del linguaggio naturale.

In particolare, nel nostro caso, l’F1-score `e stata utilizzata per entrambi i modelli di apprendimento (sia quello sugli articoli che quello sulle MRI) al fine di sti- mare l’accuracy sul Test Set utilizzato e determinare il verificarsi del fenomeno dell’overfitting in caso di valori molto elevati.

## fastai

* 1. **CNN con la libreria *fastai***

##### Implementazione

I file DICOM fornitici dagli esperti di dominio sono stati da essi etichettati, e sono state distinte due classi di immagini. In particolare, ci `e stato fornito un insieme di 691 immagini, divise in due diverse classi. La prima `e la classe GBM che contiene 489 immagini di glioblastomi multiformi, mentre la seconda `e la classe MET che contiene 202 immagini di metastasi.

Innanzitutto, i file DICOM etichettati dagli esperti di dominio sono stati sud- divisi nei due insiemi di Training e Validation. In particolare, per realizzare la suddivisione `e stata sfruttata la tecnica dell’Holdout[5](#_bookmark49), per cui il Training Set contiene l’80% dei file, mentre il Validation Set contiene il 20% rimanente.

Ai file DICOM deve essere associato un file **.csv** in cui si specifica l’etichet- ta associata ad ognuno dei file. Specificando il path in cui `e presente il file di etichettamento, il dataset viene caricato utilizzando la classe built-in *DataBlock* :

gbm = DataBlock ( blocks =(ImageBlock , Category Block ) ,

get x=**lambda** x : gbm source / f ” x [ 0 ] ” , get y=**lambda** x : x [ 1 ] ,

{ }

batch tfms=aug transforms ( s i z e =300) )

*gbm*: variabile che conterr`a l’intero DataBlock, ovvero l’intero dataset delle risonanze;

•

* *ImageBlock* : funzione che specifica cosa dovranno contenere i blocchi;

*gbm source*: variabile precedentemente dichiarata contenente il Path del file di labeling delle risonanze.

•

Caricato il dataset, si generano il Training Set ed il Validation Set che verranno utilizzati per l’addestramento del sistema e la valutazione della sua accuracy. Ci`o viene effettuato tramite la funzione *dataloaders* presente all’interno della classe DataBlock:

dl s = gbm. datal oade rs ( df . values , num workers=0 , bs = **int**

( **len** ( df . values ) ∗ 0 . 8 0 ) , d ro p l as t=True )

* *df.values*: carica i valori delle etichette;

*num workers*: indica il numero di sottoprocessi per il caricamento dei dati. Il valore di default 0 indica che viene effettuato dal main process;

•

5La tecnica dell’Holdout prevede di suddividere un dataset in modo tale che 2/3 di esso siano usati come Training Set e il restante 1/3 viene usato come Test Set.

*bs*: indica il dimensionamento dei due Set tramite una suddivisione in

•

##### batch;

*drop last* : elimina eventuali batch generati in eccesso non contenenti ele- menti del dataset.

•

E` possibile visualizzare il caricamento effettuato sui due set: dl s . t rai n . show batch ( max n=100)

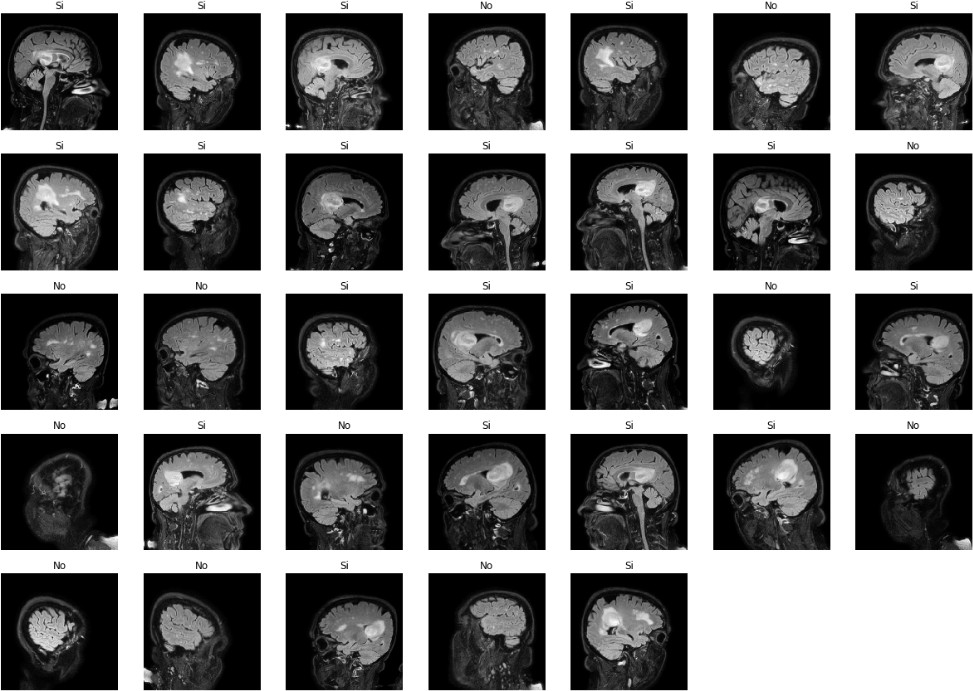


Figura 37: Visualizzazione del Training Set dl s . v a l i d . show batch ( max n=100)

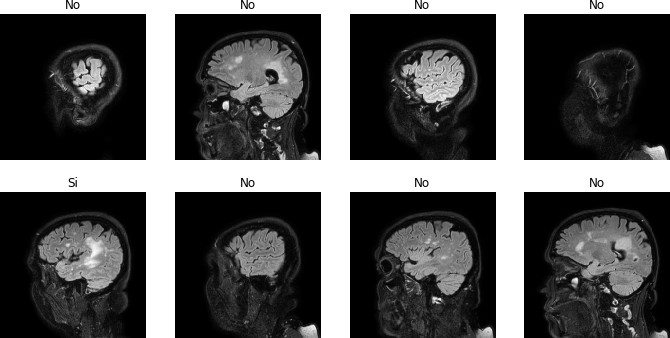


Figura 38: Visualizzazione del Validation Set

##### Risultati ottenuti

Generati Training e Validation Sets, `e stato effettuato l’addestramento del clas- sificatore, che ha impiegato un tempo pari a 6’03”.

Dopodich´e, il classificatore ottenuto `e stato valutato tramite le metriche di **Ac- curacy**, **Precision** e **Recall**, ripetendo i cicli di addestramento della CNN fino a quando non raggiungesse un valore di accuracy pari all’83%. Tale valore di soglia `e stato scelto in modo tale da evitare il verificarsi del fenomeno dell’over- fitting per valori maggiori di accuracy.

Un esempio di addestramento della CNN `e osservabile nella figura sottostante:



Figura 39: Esempio di addestramento della CNN Il risultato della classificazione ottenuto `e, quindi, il seguente:

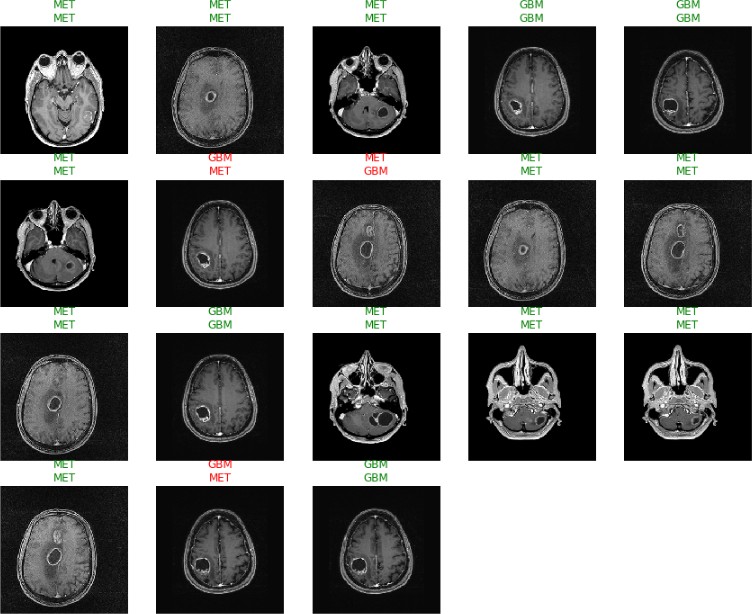


Figura 40: Classificazione effettuata dalla CNN

## Rimozione testi pediatrici

A seguito di un confronto con gli esperti di dominio, `e emerso che dalla query prodotta sono stati recuperati da PubMed articoli di tipo pediatrico che non interessavano per lo studio da effettuare. Dopo sono stati forniti gli id degli articoli da rimuovere, si `e prodotto un comando SQL di DELETE per rimuovere i suddetti dal database.

**DELETE FROM** arti c o l i g bm

**WHERE** pubmedid **IN** ( value 1 , value 2 , . . . )

Dove i valori inseriti all’interno della query sono gli id numerici di pubmed for- niti dagli esperti.

Al termine dell’operazione, sono stati rimossi 319 articoli dai 1’311 precedente- mente presenti.

## Suddivisione del lavoro

**Guido Maria Secondulfo**: Supervisione generale del progetto, carica- mento articoli su Db2, addestramento classificatore di Visual Recognition;

•

**Angelo Russo**: Pre-processing sui documenti recuperati, definizione del modello Entit`a-Relazione su IBM Watson Knowledge Studio;

•

**Francesca Lettieri**: Pre-processing sul Test Set delle risonanze, genera- zione dello script per la valutazione del modello di Visual Recognition e valutazione del classificatore;

•

**Attivit`a svolte comunemente**: Definizione query di ricerca, definizione dello script per il recupero delle informazioni degli articoli da PubMed, valutazione del modello di Watson Knowledge Studio, pre-processing sui file DICOM, creazione della documentazione.

•

### Esperti di dominio

* Dott.ssa Camilla Russo
* Dott.ssa Fernanda Picozzi

## Riferimenti bibliografici

1. E-utilities: <https://www.ncbi.nlm.nih.gov/books/NBK25501/>;
2. IBM Cloud Db2: <https://www.ibm.com/it-it/cloud/db2-on-cloud>;
3. IBM Cloud Foundry:

<https://www.ibm.com/it-it/cloud/cloud-foundry>;

1. GitHub: [https://github.com](https://github.com/);
2. MeSH: <https://it.wikipedia.org/wiki/Medical_Subject_Headings>;
3. IBM Watson Machine Learning:

<https://www.ibm.com/it-it/cloud/machine-learning>;

1. IBM Watson Knowledge Studio:

<https://www.ibm.com/it-it/cloud/watson-knowledge-studio>;

1. fastai: [https://www.fast.ai](https://www.fast.ai/);
2. IBM Cloud API: [https://cloud.ibm.com/apidocs/visual-recognition/ visual-recognition-v3](https://cloud.ibm.com/apidocs/visual-recognition/visual-recognition-v3);
3. Jupyter: [https://jupyter.org](https://jupyter.org/).
4. RadiAnt DICOM Viewer: <https://www.radiantviewer.com/it>