Titolo tesi

Sottotitolo

Giacomo Saccaggi

Facoltà di
Scienze statistiche ed economiche

Dipartimento di Statistica
Universitá Bicocca
Sede di Milano
2019/2020

Dedica

Indice

Capitolo 1

Introduzione

Il riciclo è un processo di conversione che trasforma i rifiuti in nuovi materiali, oggetti o sostanze del tutto differenti dai rifiuti d'origine. Questo processo porta principalmente 4 vantaggi:

- Conserva le risorse: i materiali ricavati vengono convertiti in nuovi prodotti, riducendo la necessità di consumare nuove risorse naturali. Se i materiali usati non vengono riciclati, i nuovi prodotti vengono realizzati estraendo materie prime fresche dalla Terra, attraverso l'estrazione e la silvicoltura. Il riciclaggio aiuta a conservare importanti materie prime e protegge gli habitat naturali per il futuro; inoltre può essere una grande possibilità per uno stato come l'Italia scarsa di risorse naturali.

- Consente di risparmiare energia: l'uso di materiali riciclati nel processo di produzione consuma molta meno energia di quella necessaria per la produzione di nuovi prodotti; inoltre, si ottiene un ulteriore risparmio energetico perché è necessaria più energia per estrarre, raffinare, trasportare e elaborare materie prime pronte per l'industria rispetto alla fornitura di materiali pronti per l'industria.

- Aiuta a proteggere l'ambiente: il riciclaggio riduce la necessità di estrazione (estrazione, silvicoltura e disboscamento), raffinazione e lavorazione di materie prime che creano un notevole inquinamento dell'aria e dell'acqua. Poiché il riciclaggio consente di risparmiare energia, riduce

anche le emissioni di gas serra, il che aiuta a contrastare i cambiamenti climatici.

- Riduce l'accumulo di rifiuti: i materiali riciclabili vengono rielaborati in nuovi prodotti e, di conseguenza, la quantità di rifiuti inviati alle discariche si riduce.

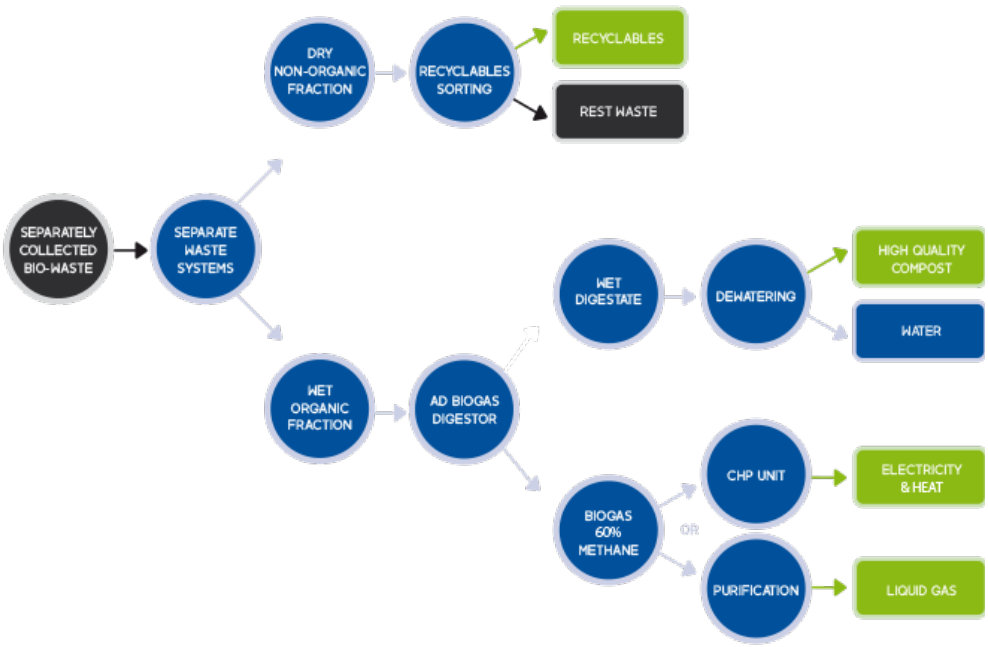

Figura 1.1: Diversi possibili riutilizzi rifiuti riciclati. (Fonte: separate waste systems)

La raccolta differenziata è il processo mediante il quale i rifiuti vengono separati in diversi elementi. Questa procedura è essenziale affinchè le aziende che si occupano di riciclo possano operare nel settore.

Un'economia circolare è un sistema economico volto a eliminare gli sprechi e l'uso continuo delle risorse. I sistemi circolari impiegano il riutilizzo, la condivisione, la riparazione, il rinnovo, la rigenerazione e il riciclaggio per creare un sistema a circuito chiuso, riducendo al minimo l'uso di input di

risorse e la creazione di rifiuti, inquinamento ed emissioni di carbonio. L'economia circolare mira a mantenere i prodotti, le attrezzature e le infrastrutture in uso più a lungo rispetto invece alla classica economia lineare che si basa sull'utilizzo di risorse sempre nuove e l'eliminazione di quelle vecchie.

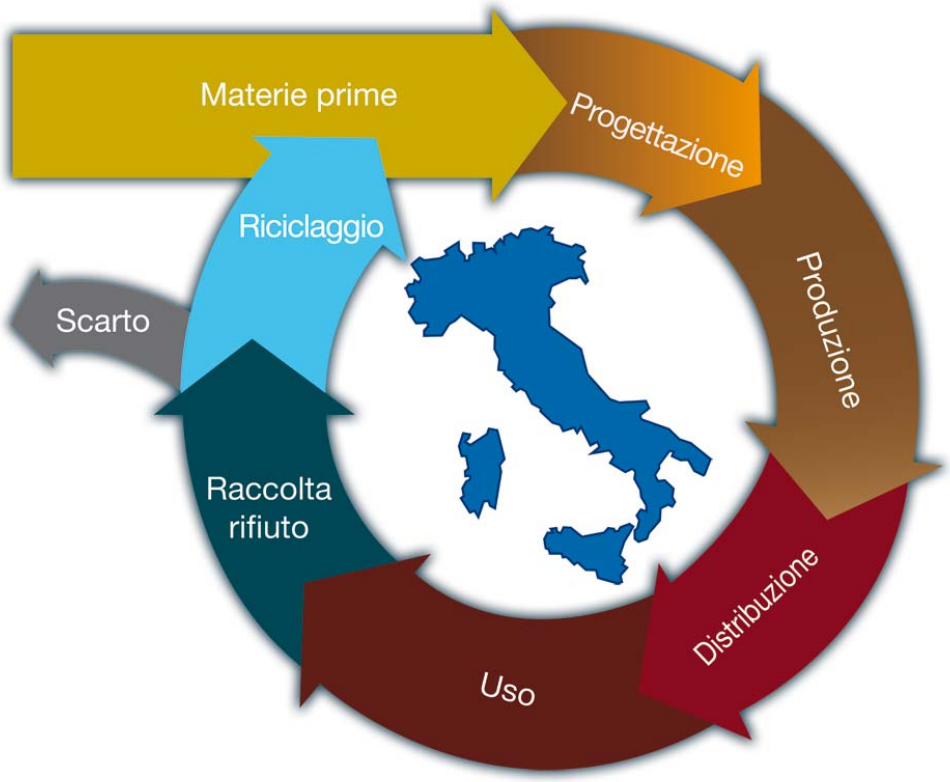

Figura 1.2: Economia circolare. (Fonte: Ecosport economia circolare)

L'economia italiana è oggi la più performante d'Europa per circolarità di materia, produttività delle risorse, capacità di riciclo. A dimostrarlo sono i numeri del rapporto "L'Economia Circolare in Italia – la filiera del riciclo

asse portante di un'economia senza rifiuti" [1]. Il documento, pubblicato nel gennaio del 2019 e curato dall'esperto ambientale Duccio Bianchi di Ambiente Italia, rappresenta il primo vero bilancio sulla "circolarità" nazionale, settore che vale oggi 88 miliardi di fatturato e 22 miliardi di valore aggiunto, ovvero l'1,5 % del valore aggiunto nazionale.

In questo libro emerge come l'Italia sia attualmente capofila in Europa nei tre indice che l'autore definisce fondamentali per valutare un economia circolare: **tasso di produttività nell'uso delle risorse** (quanti euro di PIL si producono per ogni kg di risorse consumate), **tasso di circolarità della materia nell'economia** (quante materie seconde impieghiamo sul totale dei consumi di materia) e infine il **tasso di riciclo dei rifiuti** (quanti rifiuti, urbani e non urbani, inclusi l'import ed export, avviamo a riciclo internamente). Le performance nazionali risultano non solo superiori alla media UE ma anche alle prestazioni dei principali stati come Germania, Spagna, Regno Unito e Francia.

Tra le tante eccellenze italiane nell'ambito del riciclo una delle più importanti riguarda il riciclo del legno che, come viene evidenziato nell'articolo del giornale Il Sole 24 Ore da Giovanna Mancini [2], in poco più di 20 anni il sistema del recupero e del riciclo del legno ha creato una nuova economia che ha prodotto risultati importanti sia in termini ambientali, sia per la capacità di creare sviluppo e occupazione. L'impatto economico sulla produzione nazionale delle attività della filiera del recupero del legno post consumo è stimabile, secondo il rapporto del politecnico, in circa 1,4 miliardi di euro, mentre il contributo sull'occupazione è di quasi 6mia posti di lavoro complessivamente sostenuti in Italia.

L'obbiettivo di questo lavoro non vuole essere quello di proporre un metodo di risoluzione definitivo di tutti i problemi relativi alla raccolta differenziata bensì quello di proporre alcune idee di come la statistica possa essere d'aiuto nel riuscire a migliorare lo sfruttamento delle risorse. A livello statistico questo si declina in un problema di classificazione; durante il lavoro si andranno a sfruttare sia dati strutturati che dati non strutturati sfruttando algoritmi di Deep Learning per riuscire a classificare in modo accurato i differenti tipi di rifiuti.

Capitolo 2

Reperimento dei dati

2.1 Struttura del lavoro

Il riciclo come evidenziato nell'introduzione può essere una risorsa molto importante che può non solo garantire un economia più eco sostenibile ma può essere una grande risorsa per l'occupazione e per la possibilità di avere materie prime in uno stato come l'Italia carente di molte materie prime. Per affrontare questo problema si è deciso di strutturare il lavoro in tre fasi: reperimento dei dati, elaborazione e previsione e infine scelta del modello e applicazioni.

Nella prima fase, si è deciso di reperire i dati per poter classificare i rifiuti utilizzando tre strategie: Web Scraping, programmazione di un Bot di Telegram e costruzione di un cestino che potesse registrare i dati dei rifiuti.Ognuna di queste strategie è stata scelta per ragioni differenti.

Il Web Scraping è stata scelta come strategia iniziale perché internet costituisce una fonte quasi infinita di dati (nel nostro caso di immagini) e non usarlo sarebbe stata una scelta che avrebbe portato all'esclusione di risorse

Il Bot di Telegram è stato scelto come strumento per avere immagini più precise riguardo ai materiali più comuni per il riciclo quali: carta, plastica e vetro.

Infine si è deciso di costruire un cestino così da poter raccogliere dei dati selezionati reputati importanti nel classificare gli oggetti, come tecnologia di

implementazione in questa prima fase si è deciso di utilizzare Arduino per la sua semplicità e versatilità. Poi è stato necessario costruire un software che fosse in grado di registrare e salvare i dati inviati dall'Arduino così da automatizzare e sveltire il più possibile il processo di costruzione del dataset.

La fase successiva costituisce il fulcro di questo progetto in quanto si è dovuto analizzare tutti i dati raccolti precedentemente e capire come, sfruttando algoritmi di Deep Learning e Machine Learning, riuscire a classificare nel modo migliore gli oggetti. Nell'ultima fase si sono combinati tramite meccanismi di Ensemble Learning i vari classificatori così da trovare un previsore migliore ed è stato creato un software che potesse comunicare con l'Arduino ricevendo e inviando segnali per far corrispondere alla previsione non solo la classificazione teorica dell'oggetto ma anche quella pratica.

2.2 Come differenziare

Il regolamento della raccolta differenziata può differire da zona a zona. Come regole di riferimento si è deciso di utilizzare il regolamento vigente nella città metropolitana di Milano. Il capoluogo lombardo, infatti, è diventato il Comune che ha raggiunto la quota record del 54% di raccolta differenziata, situandosi in cima alla classifica italiana e al secondo posto in Europa dopo Vienna. Un risultato importante e un fenomeno che ha catturato anche l'attenzione di New York, il cui Assessorato all'ambiente di New York ha deciso di studiare il modus operandi di Milano per adattarlo alla metropoli americana. Entro il 2020, poi, l'amministrazione cittadina si propone di arrivare al traguardo simbolico del 65%. Per capire bene quando nei prossimi capitoli parleremo genericamente di plastica, carta o altro rifiuto in questa sezione descriviamo in breve quali materiali o oggetti rientrano nelle rispettive categorie.

Innanzitutto va ricordato come esistano cinque differenti tipi di rifiuti, da depositare in appositi cassonetti: si tratta del sacco trasparente neutro per i rifiuti generici, il sacco giallo per plastica e metallo, il cassonetto verde per il vetro, il cassonetto bianco per carta e cartone e il cassonetto marrone per rifiuti organici. Vediamo le varie corrispondenze.

- Indifferenziato (sacco trasparente): vi finiscono tutti i rifiuti generici come i piatti rotti, la ceramica in generale, la carta sporca e oleata, cd, DVD, videocassette e musicassette, piccoli accendini, filtri dell'aspirapolvere, pannolini e assorbenti e mozziconi di sigaretta.

- Carta (cassonetto bianco): vi finiscono tutti gli oggetti di carta o cartone quali giornali, riviste, libri e quaderni privati delle parti di plastica, adesive e in metallo, ma anche i contenitori in Tetra Pak (quelli usati per il latte, i succhi di frutta e così via), le scatole in cartone e i cartoni della pizza senza avanzi.

- Plastica (sacco giallo): vi finiscono le plastiche e i metalli, inclusivi di bottiglie, flaconi e sacchetti di plastica, le vaschette per gli alimenti anche in polistirolo, tutte le scatolette e i barattoli per alimenti una volta sciacquate, lattine per bevande, tubetti di plastica (del dentifricio, per esempio), fogli di alluminio, pellicole per imballaggio, nonché oggetti in metallo come pentole, posate, caffettiere, tappi, capsule, chiavi e lucchetti.

- Vetro (cassonetto verde): vi finiscono tutti i materiali in vetro come bottiglie e bicchieri, vasi, caraffe e barattoli. Da non inserirvi invece lampadine, specchi e gli oggetti in cristallo.

- Umido (cassonetto marrone): vi finiscono tutti i rifiuti di natura organica in tutte le loro parti, inclusivi di frutta, frutta secca, verdura, carne e pesce, pane, riso, pasta, scarti di cucina, avanzi, fondi di caffè, filtri di tè e tisane, fiori, semi e foglie, e alimenti avariati.

2.3 Web Scraping

2.3.1 Funzionamento

Il Web scraping (chiamato anche web harvesting, o web data extraction) è un insieme di tecniche utilizzate per l'estrazione di dati dai siti Web. È una forma di copia, in cui i dati specifici vengono raccolti e copiati dal web accedendo al World Wide Web direttamente utilizzando il protocollo Hypertext

Transfer Protocol o tramite un browser Web, in genere in un database locale centrale o foglio di calcolo, per il successivo recupero o analisi, il termine in genere si riferisce a processi automatizzati implementati utilizzando un server, un bot o un web crawler.

Tra le tante possibili soluzioni si è deciso di utilizzare la tecnica di Web Scraping chiamata HTML parsing ossia una deserializzazione delle pagine HTML. Questo processo riceve il codice HTML non elaborato, la legge e genera dal codice una struttura ad albero DOM (Document Object Model).

Questa tecnica è molto versatile in quanto gran parte dei siti Web hanno grandi raccolte di pagine generate dinamicamente da un'origine strutturata sottostante come un database. I dati della stessa categoria sono in genere codificati in pagine simili da uno script o un modello comune. Nel data mining, un programma che rileva tali modelli in una particolare fonte di informazioni, ne estrae il contenuto e lo traduce in un modulo relazionale, viene chiamato wrapper. Gli algoritmi di generazione dei wrapper presuppongono che le pagine di input di un sistema di induzione di wrapper siano conformi a un modello comune e che possano essere facilmente identificate in termini di schema comune URL.

2.3.2 Implementazione

Per fare Web Scraping si è deciso di utilizzare il linguaggio R, questo perchè oltre ai pacchetti appositi che aiutano nell'operazione di creazione dello wrapper come in ogni altro linguaggio in R sono presenti delle ottime librerie per fare text mining la più famosa "stringr" che aiutano in determinate operazioni gestire operazioni di HTML parsing più complicate.

La libreria utilizzata in questa fase è "rvest" (creata da Hadley Wickham [aut, cre], RStudio [cph]) la quale gestisce e migliora alcune funzioni contenute nel pacchetto "xml2" (creata da Hadley Wickham [aut], Jim Hester [aut, cre], Jeroen Ooms [aut], RStudio [cph], R Foundation [ctb]), queste due librerie si basano su un pacchetto creato da Daniel Veillard (versione definitiva pubblicata alla fine del 2012) di nome "libxml2". Le funzioni che abbiamo utilizzato di queste librerie sono tre: *read_html*() del pacchetto "xml2"; *xml_attrs* e *html_nodes*() del pacchetto "rvest".

La prima funzione read_html() serve per deserializzare le pagine HTML e trasformarle in una lista strutturata come un albero DOM.

```
install.packages("xml2")
library("xml2")
page<-read_html("###---link---###")
```

Le due funzioni del pacchetto "rvest", invece, aiutano ad individuare i diversi elementi del documento. Nel nostro caso vengono usate congiuntamente per trovare il link di origine delle immagini per poi scaricarle.

```
install.packages("rvest")
library("rvest")
t<-1 # img t-esima presente nella pagina
link_img<-xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]]
```

Infine una volta trovati i link delle immagini presenti nelle varie pagine le salviamo sul computer, tramite la funzione $download.file()$

```
download.file(link_img, destfile="PATH", method='curl')
```

Per la ricerca dei siti dove scaricare le immagini si è utilizzata la logica alla base delle ricerche sui più comuni motori di ricerca per immagini: Google, Bing e Pixabay.

La logica che sta alla base delle ricerche online consta nel, data una parola chiava, trovare tutte le immagini collegate. Quindi si è deciso di creare delle liste di parole per ogni categorie specificate nella sezione precedente e di trovare le immagini collegate al quella particolare Keyword.

In termini generali abbiamo una volta creato un array con una serie di parole chiave queste sono state cercate all'interno di un ciclo for creando, attraverso le funzioni $gsub()$ e $paste0()$ un URL che potesse comunicare con il motore di ricerca.

Le parole chiavi scelte per l'analisi:

```
Plastica<-c("Plastica", "bottiglie di plastica", "sacchetti di
    plastica", "tappi di plastica", "giochi di plastica", "utensili
    di plastica", "pacchi di plastica", "posate di plastica",
```

```
    "bicchieri di plastica", "vaschette di plastica", "tubetti
    dentifricio", "pellicola cucina", "sedie di plastica", "secchi
    di plastica", "vaschette plastica", "contenitore uova di
    plastica")
Carta<-c("Carta", "carta giornali", "fogli di carta", "contenitore
    uova di carta", "tovaglioli di carta", "pacchi di cartone",
    "cartone pizza", "posate di carta", "bicchieri di carta",
    "libri di carta", "quaderni di carta", "sacchetti di carta",
    "tovaglie di carta", "scatole di cartone", "riviste di
    carta","carta appallottolata")
Tetra Pak<-c("Tetra Pak latte", "Tetra Pak succhi", "contenitori
    in tetra pak", "tetra pak succo brico ")
Polistirolo<-c("contenitori in polistirolo", "polistirolo",
    "vaschette polistirolo")
Vetro<-c("Vetro", "Bottiglie di vetro", "tazzine di vetro", "vetro
    rotto", "vetro bottiglie vino", "vetro bottiglie birra", "vetro
    bottiglie alcolici", "vetro bottiglie bevante", "caraffe di
    vetro", "calici di vetro", "vasi di vetro", "bicchieri di
    vetro", "bicchieri di vetro rotti", "lastre di vetro",
    "contenitori di vetro")
Indifferenziato<-c("ceramica", "carta sporca e oleata", "cd",
    "dvd", "videocassette", "musicasette", "accendini",
    "pannolini", "assorbenti", "mozziconi di sigarette",
    "lampadine", "cristallo", "specchi", "vasi ceramica", "filtri
    dell'aspirapolvere")
Legno<-c("legno", "legname", "giochi di legno", "utensili di
    legno", "assi di legno", "posate di legno", "ciotole di legno",
    "mestoli di legno", "sedie di legno", "tavoli di legno",
    "piatti di legno", "scaffali di legno", "mobili di legno",
    "tagliere di legno", "tappi di sugero")
Metalli<-c("ferro", "metallo", "pentole", "mestoli di ferro",
    "lastre metallo", "chiavi di ferro", "utensili di ferro",
    "ferro battuto", "padelle", "posate di metallo", "forchette di
    metallo", "coltelli di metallo", "cucchiai di metallo",
    "caffettiere", "Lucchetti di ferro")
Umido<-c("scarti cibo", "umido cibo", "organico cibo", "scarti
    pesce", "scarti carne", "fondi di caffe", "frutta secca",
    "avanzi cibo", "filtri te e tisane", "pane", "fiori e foglie",
    "alimenti avariati", "scarti di cucina", "avanzi pasta",
    "avanzi di riso")
```

```
Latta_e_alluminio<-c("lattine per bevande", "fogli di alluminio",
    "lattine di cocacola", "lattine di pepsi", "lattine redbull",
    "lattine di sprite", "lattine di 7up", "lattine di fanta",
    "lattine di birra", "lattine di pelati", "lattine di cereali",
    "lattine di legumi", "alluminio per cucina", "teglie di
    alluminio", "lattine")
```

La funzione per scaricare le immagini:

```
#1
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub(" ","+",parola_chiave[i]),"+jpg&source=lnms&tbm=isch"))
   for (t in 1:length(html_nodes(page,xpath = '//img'))) {
      txt<-c(txt,xml_attrs(html_nodes(page, xpath =
          "//img")[[t]])[["src"]])
}})
# 2
try({page<-read_html(paste0("https://www.bing.com/images/search?q="
    ,gsub(" ","+",parola_chiave[i]),"+jpg&FORM=HDRSC2"))
   for (t in 1:length(html_nodes(page,xpath = '//img'))) {
      txt<-c(txt,xml_attrs(html_nodes(page, xpath =
          "//img")[[t]])[["src"]])
}})
# 3
try({page<-read_html(paste0("https://pixabay.com/it/images/search/"
    ,gsub(" ","%20",parola_chiave[i])))
   for (t in 1:length(html_nodes(page,xpath = '//img'))) {
      txt<-c(txt,xml_attrs(html_nodes(page, xpath =
          "//img")[[t]])[["src"]])
}})
```

Così facendo si è stati in grado di trovare i links di tutte le immagini presenti nel sito, nella funzione sopra descritta si sono spesso utilizzati dei $try()$ per evitare gli errori 404, molto comuni quando si automatizza il parsing HTML. Funzione al completo descritta nell'Appendice A.1.1 .

2.4 Bot Telegram

2.4.1 Funzionamento

Telegram è una applicazione di messaggistica gratuita creata e ideata a Berlino da Pavel Durov e Nikolai Durov. La prima particolarità di Telegram è quella di basarai su un protocollo di comunicazione completamento open source, sviluppato per ridurre al minimo la quantità di byte inviati per ogni messaggio. Questa soluzione permette di avere una maggiore velocità anche in condizioni di scarsa recezione. La seconda qualità di Telegram riguarda l'elevato livello di sicurezza grazie all'utilizzo di algoritmi di criptazione interni alle chat che fanno aumentare notevolmente il livello di privacy.

Secondo la definizione classica data al termine Bot (abbreviazione di robot), è una chat con la quale è possibile inviare input ad un server tramite comandi precedentemente programmati e ricevere output di direttamente nella chat e/o sul server. I chatbot stanno diventando sempre più importanti in un mondo nel quale si cerca di automatizzare il più possibile ogni tipo di processo si pensi ad esempio a "TOBI" chatbot di Vodafone con il quale è possibile chattare sull'app o sul sito della società di telecomunicazioni per aiutare il cliente a risolvere eventuali problemi

2.4.2 Implementazione

2.5 Il cestino

2.5.1 Meccanica

- dimensioni
    - meccanismo

2.5.2 Robotica

- componenti utilizzati nello specifico con foto
    - immagini robotica

2.5.3 Softwaristica

2.6 Descrizione tipologie di dati

descrivere tutti i dati che il computer registra o riceve

Capitolo 3

Addestramento train e modelli

3.1 Descrizione tipologie di dati

descrivere tutti i dati che il computer registra o riceve

Dico come introduzione quanti dati ho raccolto e metto un po' di statistiche descrittive di base

(Dati non strutturati: CNN, Dati strutturati: ML [testare un po]) spiegazione teorica e applicazione

3.2 Dati Strutturati

3.2.1 Spiegazione teorica dei modelli utilizzati

3.2.2 Applicazione dei modelli

3.3 Dati non strutturati

3.3.1 Spiegazione teorica dei modelli utilizzati

3.3.2 Applicazione dei modelli

Capitolo 4

Verifica test, confronto dei modelli

Verifica test, confronto dei modelli (specificità vs accuratezza) e Ensemble Learning

4.1 Confronto modelli

4.2 Ensemble Learning

4.2.1 Spiegazione teorica dei modelli utilizzati

4.2.2 Applicazione dei modelli

Utilizzerei modelli Staked:

Staked regression algorithm

4.3 Esempi pratici di applicazioni

4.3.1 Bot di Telegram

4.3.2 Cestino

Capitolo 5

Conclusioni

- pregi del lavoro
    - punti critici
    - possibili miglioramenti (qui soffermarsi un po')
    – Amazon Web Services
    – Lettura bar code Scrivere conclusioni sull'efficacia del modello

Appendice A

Codici

A.1 Reperimento dei dati

A.1.1 Codice Web Scraping (R)

```
library(xml2)
library(rvest)
library(stringr, warn.conflicts = F)

webscraing_img <-function(img){
txt<-"img"
for(i in 1:length(img)){
# 1
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub(" ","+",img[i]),"+jpg&source=lnms&tbm=isch"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 2
try({page<-read_html(paste0("https://www.bing.com/images/search?q="
    ,gsub(" ","+",img[i]),"+jpg&FORM=HDRSC2"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
```

```
# 3
try({page<-read_html(paste0("https://pixabay.com/it/images/search/"
    ,gsub(" ","%20",img[i])))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1a
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub(" ","+",img[i]),"+jpg&source=lnms&tbm=isch&tbs=isz:l"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1b
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub(" ","+",img[i]),"+jpg&source=lnms&tbm=isch&tbs=isz:m"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1c
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub(" ","+",img[i]),"+jpg&source=lnms&tbm=isch&tbs=isz:s"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.1
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:white"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.2
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:teal"))
```

```
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.3
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:black"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.4
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:green"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.5
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:red"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.6
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:blue"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.7
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:gray"))
```

```
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.8
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:yellow"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.9
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:pink"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
# 1.10
try({page<-read_html(paste0("https://www.google.com/search?source=&q="
    ,gsub("
    ","+",img[i]),"+jpg&tbas=0&tbm=isch&tbs=ic:specific,isc:brown"))
for (t in 1:length(html_nodes(page,xpath = '//img'))) {
txt<-c(txt,xml_attrs(html_nodes(page, xpath =
    "//img")[[t]])[["src"]])
}})
}

d<-substr(txt,1,3)=="htt"

immaginidascaricare<-txt[d]
return(immaginidascaricare)
}
```

```
img<-c("nome1","nome2") #creare un vettore di nomi per scaricare
    le immagini inerenti ad un argomento


immaginidascaricare<-webscraing_img(img)

for(i in 1:length(immaginidascaricare)){
download.file(immaginidascaricare[i],destfile=paste0("#percorso
    del file/nomefile",i,".jpg"),method='curl')
}
```

A.1.2 Codice Bot Telegram per salvare i dati (Python)

```
import telepot
from datetime import datetime
import time



def rispondi(msg):
   content_type, chat_type, chat_id = telepot.glance(msg)
   if content_type == 'photo':
      bot.download_file(msg['photo'][1]['file_id'], 'dati/image'+
          str(datetime.now())[0:19] +'.png')
      bot.sendMessage(chat_id, 'Thanks for send me the photo! See
          you soon!')



bot =telepot.Bot('###---TOKEN---###')
bot.message_loop(rispondi)



print 'I am listening ...'

while 1:
   time.sleep(10)
```

A.1.3 Codice Arduino (C++)

```
num(0).
num(s(X)) :- num(X).
```

A.1.4 Codice software per salvare i dati (VB.NET)

```
num(0).
num(s(X)) :- num(X).
```

A.2 Addestramento dei modelli

A.2.1 Codice modelli (Python)

```
num(0).
num(s(X)) :- num(X).
```

A.3 Applicazioni

A.3.1 Codice software per differenziare (VB.NET)

```
num(0).
num(s(X)) :- num(X).
```

A.3.2 Codice Bot Telegram per differenziare (Python)

```
num(0).
num(s(X)) :- num(X).
```

Bibliografia

[1] Duccio Bianchi (2019), L'Economia Circolare in Italia: la filiera del riciclo asse portante di un'economia senza rifiuti, Edizioni Ambiente.

[2] Mancini Giovanna (2019), Economia circolare: il riciclo del legno vale 1,4 miliardi e 6mila posti di lavoro, Il Sole 24 Ore.