Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea in Informatica

Studio dell'engagement su social networks

Relatore: Elisabetta Fersini

Relazione della prova finale di:
Vittorio Maggio
Matricola 817034

Anno Accademico 2018-2019

Al relatore, senza il quale questa tesi non esisterebbe;

A tutta la mia famiglia, che nonostante la distanza mi è sempre stata vicina;

A mio fratello, che mi ha sempre supportato;

A tutti i miei amici e colleghi, che hanno reso questo percorso ricco di emozioni.

Indice

1 Introduzione **7**

1.1 L'evoluzione dei social networks 7

1.2 Stato dell'arte 8

1.3 Il progetto di stage 8

2 Instagram Data Scraping **11**

2.1 API Instagram vs Instagram Data Scraping 11

2.2 Web Scraping 11

2.3 Features Instagram 12

3 Sentiment Analysis **17**

3.1 Natural Language Processing 17

3.2 Tecniche di Sentiment Analysis 18

3.2.1 Pre-processing 18

3.2.2 Approccio basato su algoritmi di classificazione 19

3.2.3 Approccio basato su lessici 19

4 Analisi dell'immagine **23**

4.1 Quantificare l'estetica di un'immagine 23

4.2 NIMA Neural Image Assessment 23

5 Rete Neurale **29**

5.1 Percettrone Multistrato 29

5.2 Dataset e addestramento 31

5.3 Prestazioni 32

6 Conclusioni **35**
6.1 Commento dei risultati 35
6.2 Sviluppi futuri 35

Bibliografia **37**

Capitolo 1

Introduzione

1.1 L'evoluzione dei social networks

Negli ultimi anni l'uso dei social networks ha subito un'evoluzione drastica, nati per mettere in comunicazione più individui da ogni parte del mondo e per gestire rapporti sociali, oggi i social si sono trasformati in strumenti di propaganda, pubblicitari e molto altro. Per aziende e personaggi pubblici esserci è diventato fondamentale; tutto questo è stato reso possibile grazie alla loro rapida diffusione.
Un report del *Pew Internet research center*[1] mostra che il 64% degli italiani usano i social media per informarsi, mentre l' 81% dei consumatori si informa su un servizio o su di un brand, online. Dati che evidenziano l'importanza del ruolo che gioca il web e i social networks.
In questo scenario, è quindi indispensabile essere in grado di monitorare e analizzare le conversazioni e le interazioni degli utenti sui social media, consapevoli del fatto che, per la prima volta nella storia dell'umanità, abbiamo accesso diretto ai pensieri e alle emozioni di milioni di utenti. Le analisi che possono essere fatte sono quindi innumerevoli: dall'individuazione delle personas all'analisi delle conversazioni riguardanti uno specifico argomento.
In questa tesi si affronterà il problema di trovare un metodo per stimare l'engagment di un nuovo post, cioè prevedere la sua diffusione, quindi le reazioni degli utenti.

1.2 Stato dell'arte

Esistono già ricerche scientifiche che si sono poste l'obiettivo di stimare l'engagment su varie piattaforme social. In [2], il social network considerato è stato Twitter, sono stati considerai i profili con più di 500 followers e sono stati scaricati tutti i tweet che comprendevano un'immagine; con i dati relativi ai post e all'analisi dell'immagine sono state applicate diverse tecniche di apprendimento per analizzare e identificare i parametri che permettono ai tweets contenenti immagini di diventare virali. In [3], sono stati usati algoritmi di machine learning per tentare di assegnare un punteggio che misurasse il grado di popolarità di un account sui diversi social. Gli autori di [4] si sono concentrati invece, su una tecnica specifica per stimare la popolarità di un post sui social media, quella delle random forest. Un approccio diverso è invece, stato adottato in [5], dove è stata analizzata e stimata la variazione di popolarità di un post su Instagram nel tempo, tenendo traccia, per un mese, della popolarità dei post considerati.
Altri lavori che si sono concentrati sul social Instagram sono:
[6], in cui sono stati sviluppati strumenti di business intelligence per la visualizzazione e l'analisi degli accounts; in [7], sono stati sfruttati metodi sia di regressione, che di classificazione per prevedere il numero di likes di post contenenti immagini e video di 3 specifici profili; in [8], è stato addestrato un modello di rete neurale profonda per stimare la diffusione di un nuovo post sull'account di una rivista indiana.
In questa ricerca, prenderemo in considerazione i post di un singolo brand: *Leroy Merlin* (azienda specializzata in bricolage, edilizia, giardinaggio, decorazione e arredo). Questo per meglio analizzare e prevedere le reazioni di un'utenza precisa, che ha in comune il fatto di seguire la pagina dell'azienda precedentemente citata. In particolare considereremo il social network *Instagram* e il parametro che cercheremo di stimare è il numero di *likes* di un post. La tecnica applicata è comunque replicabile su altri social con le opportune modifiche.

1.3 Il progetto di stage

Di seguito, sono descritti i vari steps seguiti, per i quali è dedicato un capitolo esplicativo:

1. Instagram data scraping, per l'estrazione dei dati riguardanti il profilo e i post (immagine, descrizione, likes, data e ora di pubblicazione) dell'account di nostro

interesse;

2. Sentiment analysis, per stabilire la polarità (positiva o negativa) della descrizione dei post;

3. Assegnazione di un punteggio riguardante la qualità e l'estetica dell'immagine;

4. Creazione di un modello predittivo, che permetta di stimare i likes di un post.

In conclusione, saranno analizzate e commentate le prestazioni del modello predittivo. In oltre, saranno fatte considerazioni su possibili sviluppi futuri.
Il linguaggio di programmazione scelto è stato python, con le relative librerie necessarie, che verranno citate nei vari capitoli.

Capitolo 2

Instagram Data Scraping

2.1 API Instagram vs Instagram Data Scraping

Instagram mette a disposizione degli sviluppatori delle API (Application Programming Interface) per potersi interfacciare con i contenuti dell'applicazione. Nel 2018 però, l'accesso ai dati è stato fortemente rimodulato, limitandolo. Limitazioni che hanno come obiettivo quello di aumentare la protezione dei dati degli utenti, in seguito allo scandalo Cambridge Analytica, che ha visto come protagonista Facebook (proprietario di Instagram). Tra le limitazioni inserite sono state diminuite le chiamate orarie possibili, che sono passate da 5.000 a 200 l'ora.
Avendo però, la necessità di avere a che fare con migliaia di post, per la creazione del dataset, la limitazione di 200 chiamate orarie sarebbe risultata eccessivamente restrittiva. Per questo motivo, l'opzione di interfacciarsi ai dati attraverso le API offerte è stata scartata, in favore della creazione di uno scraper. Un altro vantaggio di sviluppare uno scraper personalizzato è quello di poter aver accesso a qualsiasi dato di nostro interesse, senza scendere a compromessi con le interfacce offerte dall'applicazione.

2.2 Web Scraping

Il Web Scraping è una tecnica di estrazione di dati da un sito web. Le tecniche per effettuare lo scraping sono diverse, da quelle automatizzate a quelle manuali. Noi sfrutteremo la tecnica del *Parser HTML*.
Questa tecnica consiste nell'inviare una richiesta all'indirizzo di nostro interesse, at-

traverso un browser; il server risponderà con tutti i file che compongono e permettono la renderizzazione del sito web (HTML, CSS, JavaScript e immagini), contenenti quindi, tutte le informazioni. Conoscendo la struttura del template della pagina web è possibile accedere alle informazioni di nostro interesse. È anche possibile automatizzare tale processo, nel caso in cui, le pagine web dell'applicativo usassero un template standard (come nel caso di Instagram).

Le librerie che sfrutteremo per implementare tale tecnica sono:

Selenium Webdriver per avviare e automatizzare l'esecuzione del browser;

BeautifulSoap per estrarre i dati dal DOM;

Pandas per la creazione e gestione dei dataframes contenenti i dati estratti.

2.3 Features Instagram

Gli account Instagram considerati sono: *Leroy Merlin Italia*, *Leroy Merlin Brasile*, *Leroy Merlin Francia*.
Per ogni post le features che saranno scaricate sono:

- Descrizione del post;
- Numero di like;
- Copertura degli hastags (numero di post condivisi su Instagram con gli hastags usati nella descrizione del post);
- Data e orario di pubblicazione;
- Immagine condivisa.

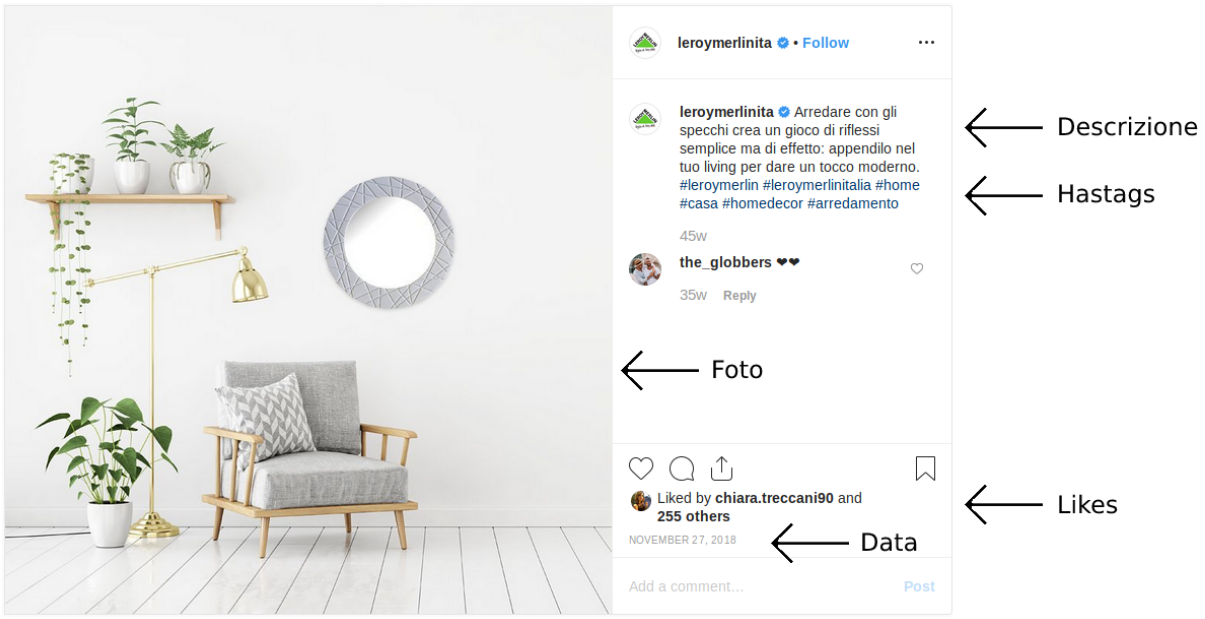

Figura 2.1: Struttura di un post su Instagram

Di seguito sono indicati alcuni tag usati per individuare le features sopra elencate.

Descrizione del post:

```
<div class="C4VMK">
  <h2 class="_6lAjh">…</h2> inline-flex
  <span title="Edited">
    Passare da un ambiente all'altro? È molto più facile e rilassante con una porta capace di ridurre de
    dell'anta.
  </span>
  <div class=" Igw0E IwRSH eGOV_ _4EzTm … aGBdT ">…</div> flex
</div>
```

Figura 2.2: Tag descrizione

Numero di like:

```
<div class="Nm9Fw"> flex
  Liked by
  <a class="FPmhX notranslate cqXBL" title="latazzinablu" href="/latazzinablu/">latazzinablu</a> event
  and
  <button class="sqdOP yWX7d _8A5w5 " type="button"> event
    <span>791</span>
    others
  </button>
```

Figura 2.3: Tag like

Numero di post condivisi su Instagram con gli hastag usati nella descrizione del post:

```
<span class="-nal3 ">
  <span class="g47SY ">2,283</span>
  posts
</span>
```

Figura 2.4: Tag hastag

Data e orario di pubblicazione:

```
<div class="k_Q0X NnvRN">
  <a class="c-Yi7" href="/p/B0qX6obngWm/"> event
    <time class="_1o9PC Nzb55" datetime="2019-08-02T12:45:49.000Z" title="Aug 2, 2019">August 2</time>
  </a>
</div>
```

Figura 2.5: Tag data e ora

Immagine condivisa:

```
▼<div class="KL4Bh" style="padding-bottom: 100%;">
   <img class="FFVAD" alt="Image may contain: indoor" decoding="auto" style="object-fit: cover;"
   sizes="1904px" srcset="https://scontent-mxp1-1.cdninstagram.com/vp/0af8469382ed9460…
   pg?_nc_ht=scontent-mxp1-1.cdninstagram.com&_nc_cat=106 1080w" src="https://scontent-
   mxp1-1.cdninstagram.com/vp/cfe2fac92b627846…79_n.jpg?_nc_ht=scontent-mxp1-1.cdninstagram.com&
   _nc_cat=106"> event
 </div>
```

Figura 2.6: Tag immagine

Capitolo 3

Sentiment Analysis

3.1 Natural Language Processing

L'elaborazione del linguaggio naturale (NLP, Natural Language Processing) è un campo dell'informatica che si occupa di analizzare la struttura sintattica del testo, con il fine di trasformare il dato testuale da un dato non strutturato (cioè che non può essere immagazzinato secondo uno schema preciso), in un dato strutturato, in modo da poterne individuare le relazioni semantiche e estrarre informazioni.
Generalmente, il Natural Language Processing prevede diverse fasi:

Parts of Speech (POS) Tagging: ogni parola viene etichettata con la sua specifica categoria lessicale (nome, verbo, aggettivo, ...). Ogni tag può essere suddiviso in ulteriori categorie. Ad esempio, un nome può appartenere alla categoria: nome singolare, nome plurale o nome proprio.

Shallow parsing or chunking: vengono etichettate intere frasi in base alla loro categoria sintattica, le principali sono:

- Noun Phrase;
- Verb Phrase;
- Adjective phrase;
- Adverb phrase;
- Conjunction;
- Prepositional phrase.

Constituency parsing: vengono usate le regole della grammatica, per analizzare la struttura della frase. Gran parte delle lingue sono basate sul modello Subject-Verb-Object (SVO).

Dependency parsing: vengono analizzate le relazioni che legano le parole (e quindi, le etichette) all'interno di una frase. Il principio base è che in ogni frase tutte le parole, eccetto una, hanno una relazione con altre parole della stessa frase. La parola che non ha alcuna dipendenza è chiamata *radice*, che solitamente è il verbo della frase.

Named entity recognition: vengono individuati i termini che rappresentano specifiche entità, esse danno molte informazioni sul contesto della frase. Ad esempio: la parola *US* rappresenta una nazione, *Coca-cola* un'azienda.

Emotion and sentiment analysis: vengono analizzate le parole del testo per quantificare la polarità della frase che, in base al valore che assume, può essere categorizzata come positiva, neutrale o negativa.

Quello che a noi interessa, per la preparazione del dataset, è assegnare un valore alla polarità della descrizione del post, quindi, effettuare un'analisi del *Sentiment*. Nel paragrafo successivo, saranno descritte le diverse tecniche per effettuare l'analisi.

3.2 Tecniche di Sentiment Analysis

3.2.1 Pre-processing

Tutte le tecniche di NLP prevedono una preliminare elaborazione del testo, che ha l'obiettivo di normalizzarlo, per rendere l'analisi più precisa e più semplice possibile. Questa fase prevede di:

- Eliminare eventuali tag HTML (se, come nel nostro caso, il testo è stato acquisito mediante un processo di web scraping);
- Sostituire i caratteri accentati con caratteri semplici;
- Rimuovere caratteri speciali;
- Espandere le abbreviazioni;

- Lemmatizzazione, che consiste nella riduzione della parola alla sua forma canonica, o lemma. Ad esempio, trasformazione del verbo nel suo infinito;
- Eliminazione delle stop words, cioè tutte quelle parole che, per la loro alta frequenza in una lingua, sono di solito, ritenute poco significative. Ad esempio articoli, congiunzioni.

Una volta normalizzato il testo, è possibile procedere con l'analisi.

3.2.2 Approccio basato su algoritmi di classificazione

Questo approccio consiste nell'usare modelli di machine learning (come regressione lineare, support vector machine o algoritmi di deep learning) per prevedere e assegnare la polarità di un testo. Il dataset è composto dalla rappresentazione numerica di frasi pre-etichettate con la rispettiva polarità.
Per adottare questa tecnica, è quindi necessario conoscere in precedenza il contesto, e avere a disposizione un'ampio dataset. Non avendo a disposizione un tale dataset, questa tecnica non è adatta al nostro scopo.

3.2.3 Approccio basato su lessici

Questo approccio consiste nell'usare un Lessico, cioè un dizionario composto da una lista di parole alle quali è associato un numero rappresentante la loro polarità. Esso è il più adatto al nostro scopo, in quanto non necessita di alcun dataset e permette di assegnare un punteggio a una vasta tipologia di frasi, senza conoscerne il contesto. In rete sono disponibili vari lessici, una piccola selezione dei più popolari, per la lingua inglese, può essere composta da:

- AFINN lexicon;
- Bing Liu's lexicon;
- MPQA subjectivity lexicon;
- SentiWordNet;
- VADER lexicon;
- TextBlob lexicon.

Ogni lessico ha un suo criterio di assegnazione del punteggio di polarità, per questo motivo è stato scelto un unico lessico inglese per assegnare la polarità delle descrizioni dei post delle pagine *Leroy Merlin Italia*, *Leroy Merlin Francia* e *Leroy Merlin Brasile*. Quindi, prima del calcolo del sentiment, ogni descrizione è stata tradotta dalla rispettiva lingua all'inglese tramite le API di Google Translate. Questo, non ha portato perdita di precisione, in quanto Google Translate ha raggiunto una qualità della traduzione molto elevata[9], in particolar modo per frasi brevi e semplici, come quelle di nostro interesse.
Il lessico scelto è stato *AFINN lexicon*.

AFINN lexicon

AFINN[10] è un lessico creato e curato da Finn Arup Nielse, ad oggi la versione inglese conta più di 3.380 parole. L'autore ha iniziato la collezione nel 2009, scaricando i tweet riguardanti le conversazioni sulla conferenza ONU sui cambiamenti climatici. A ogni parola è assegnato un punteggio che va da -5 (sentiment molto negativo) a +5 (sentiment molto positivo).
L'autore ha anche rilasciato una libreria python *afinn*, per un uso immediato del lessico e che useremo per la nostra analisi:

```
1     def sentiment(text):
2         translator = Translator()
3         text_en=[]
4         for t in text:
5             en = translator.translate(t, src="fr", dest="en")
6             text_en.append(en.text)
7         sentiment_df = pd.DataFrame()
8         sentiment_df['clean_text'] = tn.normalize_corpus(text_en)
9         af = Afinn()
10        sentiment_scores = [af.score(desc)
11                                for desc in sentiment_df['clean_text']]
12        sentiment_category = ['positive' if score > 0
13                              else 'negative' if score < 0
14                                  else 'neutral'
15                                      for score in sentiment_scores]
```

```
16          sentiment_df['sentiment_scores'] = sentiment_scores
17          sentiment_df['sentiment_category'] = sentiment_category
18          return sentiment_df
```

Il codice mostra la creazione del dataframe contenente il punteggio del sentiment per ogni descrizione dei post. Nelle righe 2-7 è stata eseguita la traduzione della descrizione del post nella lingua inglese. In seguito, è stato creato il dataframe contenente il testo normalizzato (riga 8), lo score (riga 10) e la categoria (positiva, negativa o neutrale) del punteggio (riga 12).

Capitolo 4

Analisi dell'immagine

4.1 Quantificare l'estetica di un'immagine

In un post di Instagram, l'immagine ricopre un ruolo fondamentale, però quantificare la qualità e l'estetica dell'immagine non è un compito semplice, per via della sua natura soggettiva.
Le ricerche prese in considerazione per svolgere questo task sono state due:

- Photo Aesthetics Ranking Network with Attributes and Content Adaptation [11]
- NIMA: Neural Image Assessment [12]

La scelta è ricaduta sul NIMA, in quanto una sua implementazione a cura di *idealo.de*[13] (azienda leader nella comparazione dei prezzi di prodotti venduti online) è utilizzata, con successo, per il loro portale di comparazione dei prezzi di hotels. In particolare, per scegliere in modo automatico, le migliori immagini da pubblicare fra le centinaia di immagini che ricevono per ogni hotel. Quindi, immagini aventi contenuti molto simili a quelle condivise dalle pagine dell'azienda *Leroy Merlin*.
Il modello pre-addestrato di tale implementazione verrà utilizzato per valutare le nostre immagini, assegnando un punteggio relativo all'estetica e tecnica dell'immagine.

4.2 NIMA Neural Image Assessment

NIMA è il risultato di una ricerca condotta da Google, a cura di Hossein Talebi e Peyman Milanfar; esso si basa sull'addestramento di due reti neurali convoluzionali,

basate su due dataset differenti: *TID2013*[14] per la valutazione tecnica dell'immagine e *AVA*[15] per la valutazione estetica. Nella ricerca sono stati considerati diverse architetture di classificatori, nello specifico: VGG16, Inception-v e MobileNet. Nell'implementazione considerata, è stata scelta l'architettura MobileNet.
MobileNet[16] è un modello proposto da Google, pensato per lo sviluppo di applicazioni mobile. La sua caratteristica è quindi, quella di essere veloce e leggera, questo è stato reso possibile grazie alla sostituzione degli strati dei filtri convoluzionali, con filtri convoluzionali separabili; ciò permette di ridurre significativamente il numero di parametri necessari, riducendo quindi, il peso della rete e rendendola più veloce.
Al modello è stata attuata la seguente modifica: l'ultimo strato è stato sostituito con uno strato composto da 10 neuroni (che rappresentano le possibili valutazioni dell'immagine: da 1 a 10 per il modello basato sul dataset AVA, da 0 a 9 per il modello basato sul dataset TID2013) seguiti da una funzione d'attivazione soft-max; la rete è stata inizializzata con i pesi ottenuti dall'addestramento eseguito sul dataset di *ImageNet*. La struttura finale è quindi:

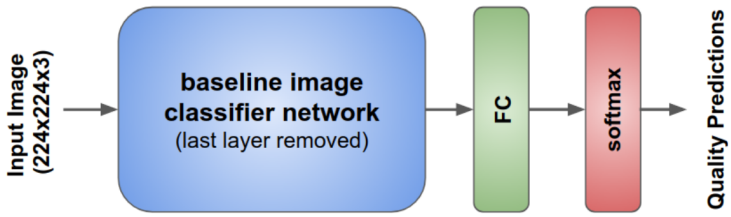

Figura 4.1: NIMA model[12]

L'addestramento è stato effettuato con la tecnica *end-to-end*, in questa fase le immagini di input sono scalate a 256 x 256 e, per ogni immagine, è stata casualmente estratta una porzione di dimensione 224 x 224; questo ha lo scopo di prevenire malfunzionamenti causati da un possibile over-fitting.
Le valutazioni (da 1 a 10) ricevute da ogni immagine possono essere espresse come una funzione di probabilità empirica: $p = [p_{s_1}...p_{s_N}]$ con $s_1 \leq s_i \leq s_N$ e $N = 10$, dove s_i rappresenta l'i-esimo possibile punteggio e N il possibile punteggio massimo. Quindi p non è altro che la distribuzione dei punteggi ricevuti dall'immagine. L'obiettivo è stimare il punteggio medio $\mu = \sum_{i=1}^{N} s_i * p_{s_i}$ di tale distribuzione.
Caratteristica del modello NIMA è l'uso dell' *Earth Mover's Distance (EMD)* come

funzione loss. L' EMD misura la distanza tra due distribuzioni di probabilità, quindi, quando usata come funzione di loss, l'obiettivo è minimizzare la distanza fra la distribuzione attesa p e la distribuzione predetta $\hat{p}$. Questa funzione permette di preservare l'ordine interno della distribuzione.
L' EMD può essere espressa da:

$$EMD(p, \hat{p}) = (\frac{1}{N} \sum_{k=1}^{N} |CDF_p(k) - CDF_{\hat{p}}(k)|^r)^{1/r}$$

dove $CDF_p(k)$ è la funzione cumulativa della distribuzione p: $CDF_p(k) = \sum_{i=1}^{k} p_{s_i}$ e $CDF_{\hat{p}}(k)$ è la funzione cumulativa della distribuzione $\hat{p}$: $CDF_{\hat{p}}(k) = \sum_{i=1}^{k} \hat{p}_{s_i}$.
La funzione soft-max posta alla fine del modello ci garantisce che $\sum_{i=1}^{N} \hat{p}_{s_i} = 1$.
r è impostato uguale a 2 per ottimizzare il processo di ottimizzazione con il metodo di discesa del gradiente.
Per l'addestramento è stato usato il 20% del dataset come test set, e il restante 80% come train set. Le prestazioni dei modelli addestrati sui dataset TID2013 e AVA sono consultabili nelle tabelle 4.1 e 4.2.

Model	*LCC (mean)*	*SRCC (mean)*	*LCC (std.dev)*	*SRCC (std.dev)*	*EMD*
Kim et al. [16]	0.80	0.80	–	–	–
Moorthy et al. [39]	0.89	0.88	–	–	–
Mittal et al. [40]	0.92	0.89	–	–	–
Saad et al. [41]	0.91	0.88	–	–	–
Kottayil et al. [42]	0.89	0.88	–	–	–
Xu et al. [35]	0.96	0.95	–	–	–
Bianco et al. [7]	0.96	0.96	–	–	–
NIMA(MobileNet)	0.782	0.698	0.209	0.181	0.105
NIMA(VGG16)	0.941	0.944	0.538	0.557	0.054
NIMA(Inception-v2)	0.827	0.750	0.470	0.468	0.064

Tabella 4.1: Prestazioni modello addestrato con il dataset TID2013[12]

Model	*Accuracy (2 classes)*	*LCC (mean)*	*SRCC (mean)*	*LCC (std.dev)*	*SRCC (std.dev)*	*EMD*
Murray et al. [1]	66.70%	–	–	–	–	–
Kao et al. [9]	71.42%	–	–	–	–	–
Lu et al. [36]	74.46%	–	–	–	–	–
Lu et al. [17]	75.42%	–	–	–	–	–
Kao et al. [37]	76.58%	–	–	–	–	–
Wang et al. [38]	76.80%	–	–	–	–	–
Mai et al. [10]	77.10%	–	–	–	–	–
Kong et al. [14]	77.33%	–	0.558	–	–	–
Ma et al. [20]	81.70%	–	–	–	–	–
NIMA(MobileNet)	80.36%	0.518	0.510	0.152	0.137	0.081
NIMA(VGG16)	80.60%	0.610	0.592	0.205	0.202	0.052
NIMA(Inception-v2)	81.51%	0.636	0.612	0.233	0.218	0.050

Tabella 4.2: Prestazioni modello addestrato con il dataset AVA[12]

Dove LCC è il coefficiente di correlazione lineare e SRCC è il coefficiente di correlazione per ranghi di Spearman.

TID2013

Considerando il tipo di immagini pubblicate dalle pagine di *Leroy Merlin* [Fig 4.2], per la valutazione delle immagini è stato scelto il modello NIMA addestrato sul dataset TID2013, il quale si basa su una valutazione tecnica dell'immagine (colori, rumore, sfocature, distorsioni ecc..). Il modello basato su dataset AVA, invece, basandosi sull'estetica generale, da rilevanza anche al soggetto della foto; quindi, avrebbe penalizzato gran parte dei post condivisi dall'azienda, contenti elementi come porte, finestre, cucine, [Fig 4.2] che verrebbero punite dal modello, ma che sono elementi di principale interesse per gli utenti che seguono la pagina. Nella Figura 4.3 viene mostrato il comportamento del modello, al variare della qualità dell'immagine.

Figura 4.2: Esempi post Leroy Merlin

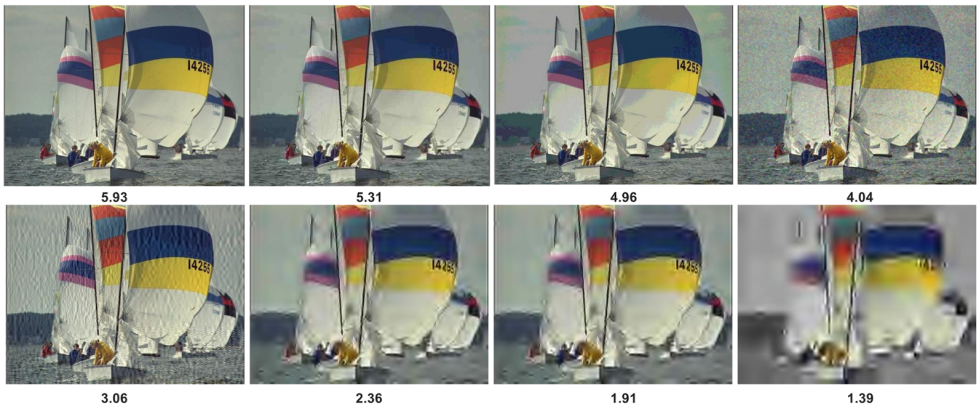

Figura 4.3: Pungeggio modello NIMA su dataset TID2013[12]

Capitolo 5

Rete Neurale

5.1 Percettrone Multistrato

Il modello scelto per stimare il numero di likes è quello del *Percettrone Multistrato*, esso consiste in una rete feedforward (le connessioni collegano i neuroni di uno strato, con i neuroni dello strato successivo; non sono consentite connessioni all'indietro o connessioni verso lo stesso livello) con almeno uno strato di input, uno statro nascosto e uno di output.
Il nostro modello è composta da 8 strati: 1 strato di input (al quale si assegnano le features del post), 6 strati nascosti e 1 strato di output (che restitutisce una stima del rapporto $\frac{Numero\ di\ likes}{Numero\ di\ Followers}$ del post dato in input); il numero di neuroni per ogni strato nascosto è: 128, 64, 32, 16, 8, 4.
La funzione di attivazione scelta per i neuroni degli stati nascosti è la funzione *Relu*:

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$$

$$f'(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

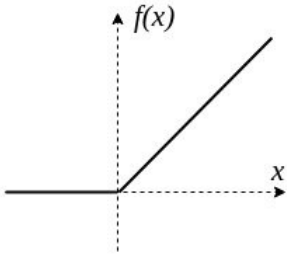

Figura 5.1: Funzione Relu

per lo strato di output è stata usata una funzione *lineare* e la funzione di ottimizzazione è *Adam*, un' estensione del metodo di discesa stocastica del gradiente. Per l'implementazione del modello è stata usata la libreria *sklearn*:

```
1   from keras.models import Sequential
2   from keras.layers.core import Dense
3
4   model = Sequential()
5   model.add(Dense(128, input_dim=25, activation="relu"))
6   model.add(Dense(64, activation="relu"))
7   model.add(Dense(32, activation="relu"))
8   model.add(Dense(16, activation="relu"))
9   model.add(Dense(8, activation="relu"))
10  model.add(Dense(4, activation="relu"))
11  model.add(Dense(1, activation="linear"))
12
13  from sklearn.model_selection import train_test_split
14  Y = df['likes/followers']
15  X = norm_df.drop(['likes/followers'], axis=1)
16  X_train, X_test, Y_train, Y_test =
17          train_test_split (X, Y, test_size = 0.25 , random_state=42)
18
19  from keras.optimizers import Adam
20  opt = Adam(lr=1e-4, decay=1e-4 / 200)
21  model1.compile(loss="mean_squared_error", optimizer=opt)
22  model1.fit(X_train, Y_train, validation_data=(X_test, Y_test),
23                                                  epochs=30, batch_size=8)
24
25  predicted = model1.predict(X_test)
26  predicted_full = model1.predict(X)
```

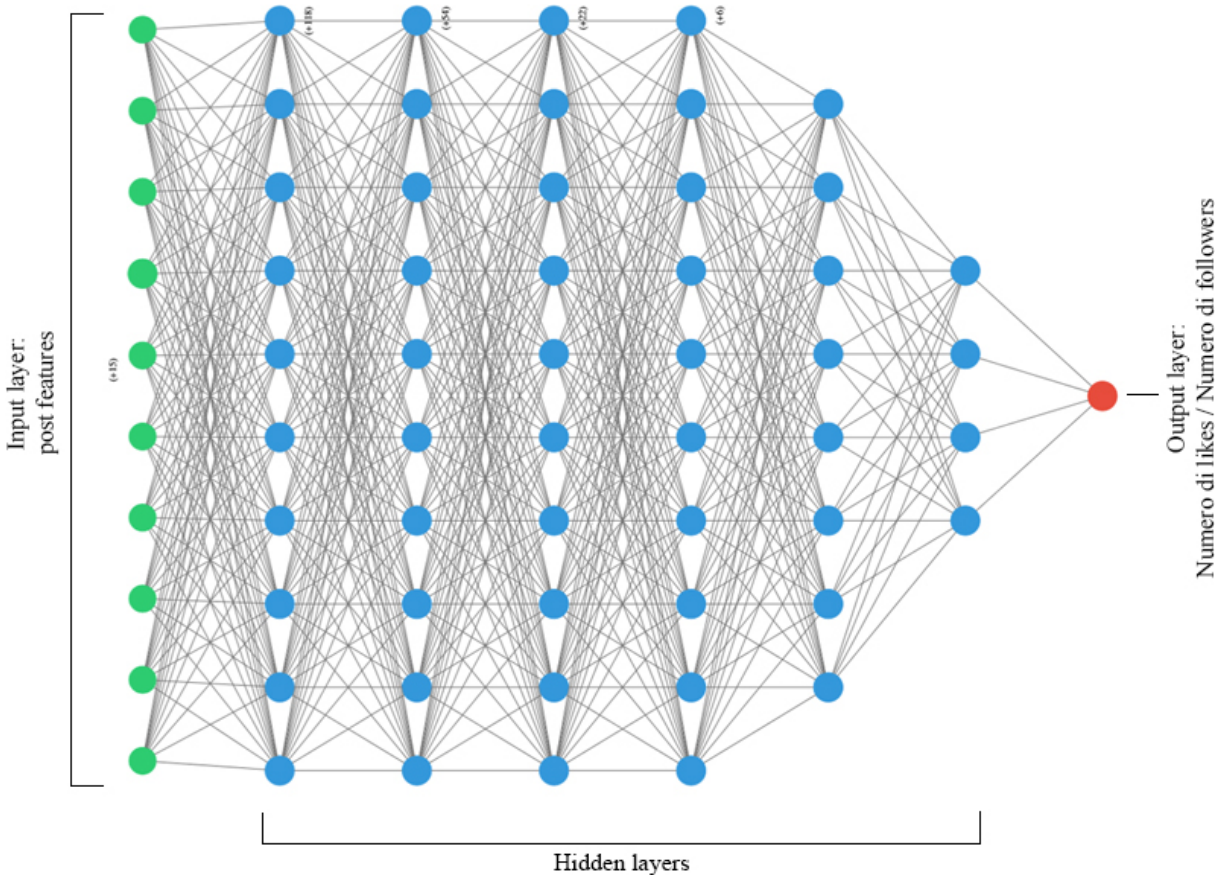

Figura 5.2: Modello finale

5.2 Dataset e addestramento

Il dataset finale comprende tutti i post pubblicati nell'anno 2018 e 2019 fino al mese di settembre, contenenti foto (non sono stati considerati i post contenti video), delle pagine *Leroy Merlin Italia*(290 post), *Leroy Merlin Francia* (625 post) e *Leroy Merlin Brasile*(563 post). Per ogni post le features considerate (quindi, gli input della rete neurale) sono:

- Numero caratteri della descrizione del post;
- Numero di hastags usati;
- Copertura totale degli hastags;

- Fascia oraria;
- Giorno della settimana;
- Tempo trascorso dalla pubblicazione del post precedente;
- Sentiment della descrizione del post;
- Punteggio dell'immagine.

Per l'addestramento il dataset (che comprende in totale 1.478 post) è stato così diviso: 25% test set e il restante 75% training set. Il parametro da stimare è il rapporto: $\frac{Numero\ di\ likes}{Numero\ di\ Followers}$, non avendo a disposizione informazioni storiche sul numero di followers di un account, è stato supposto che esso, per le rispettive pagine, non abbia subito cambiamenti significativi nel periodo sopracitato, del quale sono stati considerati i post.

5.3 Prestazioni

Le prestazioni del modello sono state misurate attraverso il coefficiente di determinazione R^2:

$$R^2 \ = \ 1 - \frac{RSS}{TSS}$$

Dove:
$RSS = \sum_{i=1}^{n}(y_i - \hat{y})^2$ è la devianza residua;
$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ è la devianza totale;
y_i sono i dati osservati, $\bar{y}$ è la loro media e $\hat{y}_i$ sono i dati stimati dal modello ottenuto dalla regressione.
Il valore massimo di R^2 è 1, più il suo valore si avvicina a 1, migliore è la funzione stimata.
La tabella 5.1 mostra i risultati di R^2 all'aumentare del dataset, partendo dai post di *Leroy Merlin Italia (LM IT)* e aggiungendo rispettivamente i post di *Leroy Merlin Francia (LM FR)* e *Leroy Merlin Brasile (LM BR)*.

		LM IT	LM IT + LM FR	LM IT + LM FR + LM BR
Test set:	R^2	-0,045	0,018	0,129
Full set:	R^2	0,071	0,263	0,318

Tabella 5.1: Coefficiente R^2 all'aumentare del dataset.

Come mostrato, il coefficiente R^2 è lontano dal valore ottimale 1. Questo perché stimare il numero esatto di likes è un compito molto difficile.
Considerando però, un margine di errore del 5% (quindi, considerando corrette tutte le risposte che si discostano al più del 5% dal valore reale), otteniamo una correttezza pari al 76% sul dataset completo.
Inoltre, analizzando la differenza tra i risultati reali e quelli predetti, si nota che la nostra rete tende a sottostimare quello che è il reale numero di likes. Per evidenziare questa tendenza è stato sottratto al numero di like predetto il 5%, questo ha permesso di interpretare i risultati predetti dal modello come un limite inferiore di likes che riceverà un post, con una correttezza dell' 85% sul dataset completo.

Capitolo 6

Conclusioni

6.1 Commento dei risultati

La ricerca ha delineato un possibile metodo per prevedere la diffusione di un post fra gli utenti delle pagine Instagram di Leroy Merlin.
Come ci si poteva aspettare, le prestazioni del modello sono migliorate all'aumentare della dimensione del dataset, fino a raggiungere una buona accuratezza nell' interpretazione dei risultati come intervallo di likes e un'ottima accuratezza se si interpretano i risultati come limite inferiore di likes che riceverà il post. Fenomeni di over-fitting si possono escludere, visto il basso valore di R^2, sia per il test set, sia nel dataset completo, e vista la bassa differenza fra le rispettive prestazioni. Quindi, nel complesso, questa ricerca potrebbe fornire uno strumento affidabile per studiare al meglio le caratteristiche di un post prima della sua pubblicazione, dando la possibilità di massimizzare la sua diffusione; o stabilire quali post conviene sponsorizzare, scegliendo quello per il quale è previsto un apprezzamento maggiore.

6.2 Sviluppi futuri

Gli sviluppi futuri possono essere molti, a partire dall'implementazione di tale metodologia su pagine Instagram di altre aziende, o su altri social networks come Facebook o Twitter. Un'ulteriore metrica che potrebbe essere usata per misurare l'engagement potrebbe essere il numero di commenti, dei quali sarebbe anche interessante analizzare il sentiment. Un'ulteriore idea potrebbe essere quella di effettuare un'analisi non

solo sui post, ma anche sul tipo di utenza che interagisce con la pagina, analizzando i vari profili, per delineare le caratteristiche comuni.

Bibliografia

[1] Pew Research Center. Facebook is the top social media site for news in western europe. https://www.journalism.org/2018/05/14/many-western-europeans-get-news-via-social-media-but-in-some-countries-substantial-minorities-do-not-pay-attention-to-the-source/pj_2018-05-14_western-europe_5-02/.

[2] Nimish Joseph, Amir Sultan, Arpan Kumar Kar, and P. Vigneswara Ilavarasan. Machine learning approach to analyze and predict the popularity of tweets with images. In Salah A. Al-Sharhan, Antonis C. Simintiras, Yogesh K. Dwivedi, Marijn Janssen, Matti Mäntymäki, Luay Tahat, Issam Moughrabi, Taher M. Ali, and Nripendra P. Rana, editors, *Challenges and Opportunities in the Digital Era*, pages 567–576, Cham, 2018. Springer International Publishing. ISBN 978-3-030-02131-3.

[3] Anuja Arora, Shivam Bansal, Chandrashekhar Kandpal, Reema Aswani, and Yogesh Dwivedi. Measuring social media influencer index- insights from facebook, twitter and instagram. *Journal of Retailing and Consumer Services*, 49:86 – 101, 2019. ISSN 0969-6989. doi: https://doi.org/10.1016/j.jretconser.2019.03.012. URL http://www.sciencedirect.com/science/article/pii/S0969698919300128.

[4] Feitao Huang, Junhong Chen, Zehang Lin, Peipei Kang, and Zhenguo Yang. Random forest exploiting post-related and user-related features for social media popularity prediction. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 2013–2017, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5665-7. doi: 10.1145/3240508.3266439. URL http://doi.acm.org/10.1145/3240508.3266439.

[5] K. Almgren, J. Lee, and M. Kim. Prediction of image popularity over time on social media networks. In *2016 Annual Connecticut Conference on Industrial Electronics, Technology Automation (CT-IETA)*, pages 1–6, Oct 2016. doi: 10.1109/CT-IETA.2016.7868253.

[6] H. Muhammad, F. Wahiduddin, N. F. A. Budi, and A. N. Hidayanto. An integrated framework to investigate influencing factors of user's engagements on instagram contents. In *2018 Third International Conference on Informatics and Computing (ICIC)*, pages 1–6, Oct 2018. doi: 10.1109/IAC.2018.8780511.

[7] A. Zohourian, H. Sajedi, and A. Yavary. Popularity prediction of images and videos on instagram. In *2018 4th International Conference on Web Research (ICWR)*, pages 111–117, April 2018. doi: 10.1109/ICWR.2018.8387246.

[8] S. De, A. Maity, V. Goel, S. Shitole, and A. Bhattacharya. Predicting the popularity of instagram posts for a lifestyle magazine using deep learning. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 174–177, April 2017. doi: 10.1109/CSCITA.2017.8066548.

[9] A neural network for machine translation, at production scale. URL `https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html`.

[10] Finn Årup Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, and Mariann Hardey, editors, *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, volume 718 of *CEUR Workshop Proceedings*, pages 93–98, May 2011. URL `http://ceur-ws.org/Vol-718/paper_16.pdf`.

[11] Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. Photo aesthetics ranking network with attributes and content adaptation, 2016.

[12] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, Aug 2018. ISSN 1941-0042. doi: 10.1109/tip.2018.2831899. URL `http://dx.doi.org/10.1109/TIP.2018.2831899`.

[13] Christopher Lennan, Hao Nguyen, and Dat Tran. Image quality assessment. https://github.com/idealo/image-quality-assessment, 2018.

[14] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57 – 77, 2015. ISSN 0923-5965. doi: https://doi.org/10.1016/j.image.2014.10.009. URL http://www.sciencedirect.com/science/article/pii/S0923596514001490.

[15] Ava: A large-scale database for aesthetic visual analysis.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.