Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Mercari Price Suggestion Challenge

Authors:
Gabriele Ferrario - 817518 - g.ferrario@campus.unimib.it
Riccardo Pozzi - 807857 - r.pozzi@campus.unimib.it

18 gennaio 2021


UNIVERSITA' DEGLI STUDI DI MILANO
BICOCCA

Sommario

The ABSTRACT is not a part of the body of the report itself. Rather, the abstract is a brief summary of the report contents that is often separately circulated so potential readers can decide whether to read the report. The abstract should very concisely summarize the whole report: why it was written, what was discovered or developed, and what is claimed to be the significance of the effort. The abstract does not include figures or tables, and only the most significant numerical values or results should be given.

1 Introduction

Il progetto trae origine dalla *Kaggle challenge* **Mercari Price Suggestion Challenge**[1] aperta a fine Novembre 2017 che come viene reso chiaro dal sottotitolo *"Can you automatically suggest product prices to online sellers?"* pone l'obiettivo di stimare più precisamente possibile il prezzo di determinati prodotti a partire da alcune loro caratteristiche.
Alla base di ciò vi è l'esigenza dell'*ecommerce Mercari*[2] di offrire ai propri venditori un suggerimento sul prezzo di vendita dei prodotti inseriti.

Si tratta quindi di un problema di *regressione* che a partire dalle varie caratteristiche dei prodotti, testuali e non, vuole calcolare il prezzo da suggerire.
Durante lo svoglimento del progetto si valuteranno vari approcci al problema, soprattuto per quanto riguarda i dati di tipo testuale, soffermandosi sulle performance di regressione sia in termini di errore rispetto ai dati di *train* che di costi computazionali.

2 Datasets

Il Dataset consiste in un elenco di 1391082 prodotti descritti tramite le seguenti caratteristiche:

2.0.1 Price

Price rappresenta il prezzo per il quale l'articolo è stato venduto (variabile target). Il prezzo medio è di circa $26, con un valore minimo pari a $0 e un valore massimo pari a $2009; inoltre, presenta una deviazione standard di

circa $38. Analizzando i percentili ci si accorge che i prezzi sono relativamente bassi in quanto il 75% dei prodotti hanno un prezzo al di sotto di $29.

2.0.2 Train id

Train_id rappresenta l'identificativo del prodotto nell'elenco.

2.0.3 Name

Name è il nome del prodotto sotto forma di dato non strutturato.

2.0.4 Shipping

Shipping è caratterizzato dal valore 1 se la tassa di spedizione è a carico del venditore, altrimenti 0 se è a carico dell'acquirente. Questo attributo è decentemente ripartito tra i venditori e gli acquirenti, in quanto il 55% dei prodotti prevede un valore di 0. Analizzando il prezzo degli articoli ci si aspetta che quelli i cui costi di spedizione sono a carico del venditore avranno un prezzo più alto. Tuttavia, ci sono una serie di fattori contrastanti. Questo può essere vero all'interno di specifiche categorie di prodotti e condizioni degli articoli, ma non quando si confrontano gli articoli sul totale. Infatti, il prezzo medio pagato dagli utenti che devono pagare le spese di spedizione (circa $30) è superiore a quello dei prodotti la cui spedizione è pagata dal venditore (circa $22).

2.0.5 Item condition

Item_condition_id rappresenta lo stato del prodotto fornito dal venditore, questo valore varia da 1 a 5. Il valore più frequente è 1, mentre 4 e 5 sono i più rari. Nei dati non è presente una descrizione dettagliata sul significato di questi valori, analizzando il dataset si può supporre che il valore 1 identifichi la condizione migliore poichè è la più frequente, mentre il valore 5 la condizione peggiore. Tuttavia calcolando i prezzi medi di vendita per ogni condizione non si riesce ad arrivare a una conclusione sicura poichè la condizione 5 è quella con il prezzo medio più alto, mentre la condizione 4 è quella con il prezzo medio più basso e le restanti categorie presentano un prezzo medio molto vicino.

2.0.6 Category Name

Category_name rappresenta la categoria di prodotto a cui appartiene l'articolo. Nel dataset sono presenti 1287 categorie univoche e tra ognuna di esse si vede una categoria principale/generale, seguita da due o più sottocategorie più specifiche (ad esempio: Women/Tops & Blouses/T-Shirts). Inoltre, ci sono 6327 articoli che non hanno una categoria assegnata. Infine, analizzando le dieci categorie più popolari, si nota che l'abbigliamento femminile è molto popolare su Mercari. Infatti, di queste prime dieci categorie 5 sono di abbigliamento femminile; Anche il trucco e l'elettronica sono categorie molto quotate.

2.0.7 Brand Name

Brand_name rappresenta il marchio dell'articolo; nel dataset sono presenti 4809 valori differenti e 632682 valori mancanti.

2.0.8 Item Description

Item_description rappresenta la descrizione del prodotto sotto forma di dato non strutturato. Nel dataset sono presenti 4 istanze senza descrizione e 82494 descrizioni con la stringa "no description yet". Inoltre, non esiste una correlazione tra la lunghezza delle descrizioni e il prezzo, in quanto c'è una correlazione di 0.048. Analizzando le word cloud ottenute tramite i bigrammi delle descrizioni dei prodotti suddivisi in quattro fasce di prezzo: prezzo >=100 (Figura 1), 50< prezzo <100 (Figura 2), 30< prezzo <=50 (Figura 3) e prezzo <= 30 (Figura 4); si riescono a notare delle differenze sulle parole più frequenti. Infatti, nella wordcloud di Figura 1 sono molto frequenti bigrammi che danno informazioni sulle buone condizioni dei prodotti, ad esempio: 100 authentic, great condition e good condition. Diminuendo di prezzo questi bigrammi diventano meno frequenti, ma aumentano i bigrammi relativi a descrizioni mancanti.

Figura 1: Word Cloud contenente i bigrammi ottenuti dalle descrizioni dei prodotti con prezzo maggiore o uguale a 100

Figura 2: Word Cloud contenente i bigrammi ottenuti dalle descrizioni dei prodotti con prezzo maggiore di 50 e minore di 100

Figura 3: Word Cloud contenente i bigrammi ottenuti dalle descrizioni dei prodotti con prezzo maggiore di 30 e minore o uguale a 50

Figura 4: Word Cloud contenente i bigrammi ottenuti dalle descrizioni dei prodotti con prezzo minore o uguale a 30

2.0.9 Pulizia Dataset

Dal dataset sono stati eliminati tutti i prodotti con prezzi minori di cinque e maggiori di 2000 poichè sul sito ufficiale di Mercari è specificato che i prezzi possono essere impostati solo nell'intervallo [5,2000] (fonte: `https://www.mercari.com/us/help_center/article/69`). Durante la fase di analisi si è scoperta la presenta di valori mancanti nei campi: item_description, brand_name e category_name, questi valori sono stati rimpiazzati con il valore NA. Inoltre, è stato effettuato un trattamento dei dati (cito paper sul text preprocessing) non strutturati convertendoli tutti in minuscolo, sono state sostituite le descrizioni mancanti con il valore NA (anche quando era presente la stringa "no description yet"). Sui dati testuali è stata effettuata una fase di lemmatizzazione (perchè usa vocabolario su cui tagliare le parole e cito

articolo). Successivamente, sulle nuove parole ottenute è stata effettuata una fase di pulizia di questi campi non strutturati eliminando le stopwords, la punteggiatura, tutti i caratteri di lunghezza pari a 1 che non sono numeri (è stato deciso di mantenere tutti i numeri poichè molte descrizioni senza di essi perdono di significato) e sono state eliminate le emoji.

I valori del campo category_name sono stati codificati in interi tramite la tecnica label encoding.

3 The Methodological Approach

3.0.1 Training e valutazione

3.0.2 Dataset split

Al fine di valutare i vari modelli più correttamente possibile il dataset è stato suddiviso in *training-set*, *validation-set* e *test-set*. Quindi il training è stato usato per l'allenamento, il validation per accertarsi che il modello non sia propenso all'overfitting ed il test per la valutazione finale.

3.0.3 Valutazione dell'errore

Per quanto riguarda le performance in termini di errore la funzione di *loss* utilizzata in fase di training è il *Root Mean Squared Logaritmic Error (RMSLE)*, in quanto è la misura scelta dalla Kaggle challenge per confrontare le performance dei vari partecipanti. Inoltre essa risulta adeguata al problema considerando il vasto intervallo dei valori dei prezzi.

Di seguito la definizione e alcune osservazioni su di RMSLE insieme alle ulteriori metriche utilizzate; ad esempio il *Mean Absolute Error (MAE)*, che fornisce una più immediata comprensione rispetto al RMSLE.
Mean Absolute Error (MAE):

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

Mean Squared Error (MSE):

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

Il quadrato fornisce un peso maggiore agli errori dei valori più elevati[3].
Root Mean Squared Error (RMSE):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{3}$$

La radice dell'MSE, nella stessa scala dei valori.
Mean Squared Logarithmic Error (MSLE):

$$\frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y_i} + 1))^2 = \frac{1}{n}\sum_{i=1}^{n}\log^2(\frac{y_i + 1}{\hat{y_i} + 1}) \tag{4}$$

Essendo calcolato a partire da un rapporto riflette l'errore relativo e di conseguenza risulta efficace laddove i valore assoluti presentano variazioni considerevoli.
Root Mean Squared Logarithmic Error (RMSLE):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y_i} + 1))^2} \tag{5}$$

La radice dell'MSLE.

3.0.4 Valutazione dei costi computazionali

Il numero dei parametri allenabili dei vari modelli sono stati annotati insieme al tempo richiesto per l'esecuzione di 10 epoche di train sulla piattaforma *Google Colab* abilitando l'accelerazione hardware GPU.

3.1 Categoriche

Dire qualcosa sulle categoriche

Inizialmente sono state valutate le performance di regressione dei prezzi a partire dalle sole features categoriche in modo tale da stabilire un punto di partenza per la successiva analisi delle features testuali *name* e *item_description*, più complesse e computazionalmente costose.

Un semplice modello composto da due layer Densi è stato sufficiente a raggiungere le seguenti performance alla fine della decima epoca con l'aiuto di un layer Dropout interposto ad essi per contrastare l'overfitting.

3.2 Rappresentazione del Testo

3.2.1 Bag of Words

Il modello bag-of-words è una rappresentazione semplificata di testi, dove ciascuno di essi è rappresentato come una "borsa" (vettore) delle sue parole. Quindi crea un vettore per ogni documento contenete il conteggio delle occorrenze di ciascuna parola del vocabolario nel documento. Questa tecnica ignora la grammatica e persino l'ordine delle parole, ma mantiene la molteplicità [4].

3.2.2 Tf-Idf

Tf-Idf (Term frequency-Inverse document frequency) [4] risolve uno dei principali problemi che si hanno con Bag of Words, poiché quest'ultima tecnica non considera relazioni tra i vari documenti. Infatti, Tf-Idf (6) considera l'importanza di una parola per un documento in una collezione di documenti (corpus). Il vettore costruito per ogni documento tramite questa tecnica considera per ciascun termine del documento la frequenza con cui il termine compare in questo documento (Term Frequency, TF (7)) rispetto alla frequenza del termine in tutti i documenti (Inverse Document Frequency, IDF (8)).

$$Tf - Idf_{t,d} = tf_{t,d} \cdot idf_t \tag{6}$$

$$tf_{t,d} = \frac{n_{t,d}}{numberoftermsinthedocumentd} \tag{7}$$

$$idf_t = \log \frac{numberofdocuments}{numberofdocumentswithtermt} \tag{8}$$

Dove in (6)(7)(8) t indica il termine e d indica il documento.

3.2.3 Word Embeddings

Le precedenti tecniche sono semplici da realizzare, robuste e funzionali per molti tasks. Tuttavia queste tecniche semplici sono al limite in molti compiti

perché trattano le parole come unità atomiche, senza alcuna correlazione poiché vengono rappresentate come indici di un vocabolario. Con lo sviluppo delle tecniche di Machine Learning sono nati modelli in grado di apprendere rappresentazioni vettoriali di alta qualità di parole provenienti da set di dati composti da miliardi di parole e con vocabolari di milioni di parole.

Queste nuove tecniche prendono il nome di **word embeddings** e consentono di rappresentare parole per mezzo di vettori densi e di lunghezza fissa [5], fornendo di conseguenza una rappresentazione più efficiente rispetto alla *Bag of words* sparsa e dimensionalmente più costosa.

Inoltre questi vettori sono in grado di rispettare la similarità tra le parole considerando più gradi di somiglianza e rappresentando la regolarità semantica e sintattica, permettendo anche operazioni algebriche [6].

3.2.4 GLOVE

GLOVE [7] è un algoritmo di apprendimento non supervisionato che fornisce una rappresentazione vettoriale di ogni parola, considerando le statistiche globali di ricorrenza parola per parola (word-word ??????) da un corpus e le rappresentazioni ottenute mostrano sottostrutture lineari dello spazio vettoriale delle parole. Inoltre, gli ideatori di questa tecnica hanno reso disponibile i word embeddings pre-addestrati. Nel dettaglio in questo lavoro sono stati utilizzati i seguenti embeddings:

- Wikipedia 2014 + Gigaword 5: allenato su 6 miliardi di parole e con un vocabolario di 400 mila parole;
- Common Crawl: allenato su 840 miliardi di parole e con un vocabolario di 2.2 milioni di parole (il più grande disponibile di GLOVE);

3.2.5 Transformers

Il Transformer è un'architettura proposta nel 2017 che si contrappone alle RNN evitando quindi la ricorrenza e utilizzando esclusivamente un meccanismo di *attention* per rappresentare i rapporti di dipendenza di input e output.

3.3 modello

Nella realizzazione di questo progetto sono stati utilizzati due modelli principali che trattano le tecniche di rappresentazione utilizzate ed entrambi i modelli condividono la seguente struttura:

- **Concatenate layer**: concatena gli input in un singolo tensore;
- **Dropout layer**: per escludere una frazione dell'input e il fattore utilizzato è 0.2;
- **Dense layer**: composto da 32 neuroni e come funzione di attivazione usa la Relu;
- **Dropout layer**: utilizza un fattore di 0.2;
- **Dense layer**: composto da 16 neuroni e come funzione di attivazione usa la Relu;
- **Dense layer**: rappresenta il layer di output ed è composto da un neurone con funzione di attivazione lineare poiché il task è una regressione;

3.3.1 Word Embeddins

Il modello principale utilizzato per i Word Embeddings (sia pretrainati che non) tratta i seguenti input differentemente: descrizione del prodotto, variabili categoriche del prodotto (nome, categorie, brand, shipping e condizione) e nome del prodotto. La descrizione e il nome vengono rappresentati sotto forma di interi, dove ogni intero rappresenta l'indice di un token in un dizionario. Le variabili categoriche che non prevedevano un valore intero sono state convertite in valori interi. La descrizione prevede un layer di embedding, il quale crea la rappresentazione vettoriale densa delle parole e nel caso di GLOVE i pesi utilizzati non sono apprendibili in quanto utilizziamo quelli pretrainati. Successivamente sono stati utilizzati due layer LSTM Bidirezionali per trattare questo input e il primo layer prevede 16 celle LSTM e un fattore di dropout di 0.2, mentre il secondo prevede 8 celle LSTM e anch'esso utilizza un fattore di dropout di 0.2. Infine sul campo descrizione viene utilizzato un GlobalMaxPooling1D per il sotto campionamento in una rappresentazione più compatta. Anche il nome viene trattato tramite un layer di embedding (anche qui con GLOVE vengono utilizzati i pesi pretrainati), successivamente è stato utilizzato un layer di LSTM Bidirezionale

con 12 celle e un fattore di dropout di 0.2; infine è stato utilizzato anche qui il GlobalMaxPooling1D. Sulle variabili categoriche non sono state effettuate altre operazioni. I tre nuovi input ottenuti vengono passati nel layer di concatenazione e quindi nella struttura spiegata precedentemente all'inizio della sezione 3.3.

Nel modello sono stati utilizzati layers di RNN per trattare i dati non strutturati, poiché sono ottimi per l'elaborazione di dati sequenziali e nei testi per assegnare un'interpretazione corretta la relazione delle parole è molto importante. Infatti, le RNN elaborano una sequenza di input un elemento alla volta, mantenendo in "memoria" informazioni sugli elementi passati della sequenza [8].

4 Results and Evaluation

The Results section is dedicated to presenting the actual results (i.e. measured and calculated quantities), not to discussing their meaning or interpretation. The results should be summarized using appropriate Tables and Figures (graphs or schematics). Every Figure and Table should have a legend that describes concisely what is contained or shown. Figure legends go below the figure, table legends above the table. Throughout the report, but especially in this section, pay attention to reporting numbers with an appropriate number of significant figures.

4.1 Risultati clean

4.2 Risultati raw

4.2.1 clean o raw

abbiamo valutato clean e raw; quali modelli beneficiano del cleanup, quali no? perchè? il deep learning impara meglio se il testo è raw?

- bag of word based models: count vectorizer - tf/idf
- embeddings: keras, glove pretrained. dire perchè non abbiamo provato qualcosa tipo word2vec?
- transformers con fine tuning

5 Discussion

The discussion section aims at interpreting the results in light of the project's objectives. The most important goal of this section is to interpret the results so that the reader is informed of the insight or answers that the results provide. This section should also present an evaluation of the particular approach taken by the group. For example: Based on the results, how could the experimental procedure be improved? What additional, future work may be warranted? What recommendations can be drawn?

6 Conclusions

Conclusions should summarize the central points made in the Discussion section, reinforcing for the reader the value and implications of the work. If the results were not definitive, specific future work that may be needed can be (briefly) described. The conclusions should never contain "surprises". Therefore, any conclusions should be based on observations and data already discussed. It is considered extremely bad form to introduce new data in the conclusions.

Riferimenti bibliografici

[1] Kaggle, "Mercari price suggestion challenge," https://www.kaggle.com/c/mercari-price-suggestion-challenge, 2017.

[2] Mercari, "Mercari," https://www.mercari.com.

[3] T. Chai and R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?– arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, pp. 1247–1250, 06 2014.

[4] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[5] F. Almeida and G. Xexéo, "Word embeddings: A survey," 2019.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[8] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *EURASIP journal on wireless communications and networking*, vol. 2017, no. 1, pp. 1–12, 2017.