MERCARI PRICE SUGGESTION

Anna Marika Biasco,
Francesca Pulerà e
Massimo Zarantonello

Anno accademico 2024/2025

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
DISCO DIPARTIMENTO DI INFORMATICA SISTEMISTICA E COMUNICAZIONE

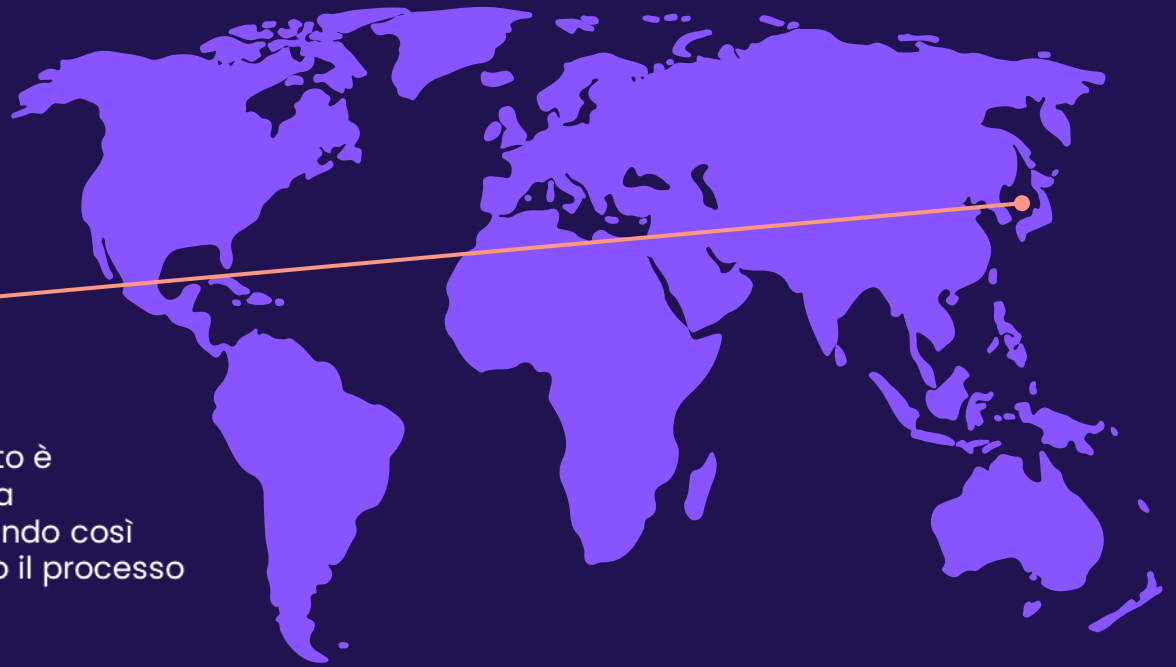
Obiettivo del Progetto

Creare un modello di apprendimento automatico in grado di prevedere automaticamente i prezzi dei prodotti online in base ai seguenti dati forniti dall'utente:

- **Nome del prodotto**
- **Categoria del prodotto**
- **Marchio**
- **Condizione dell'articolo**
- **Descrizione testuale**

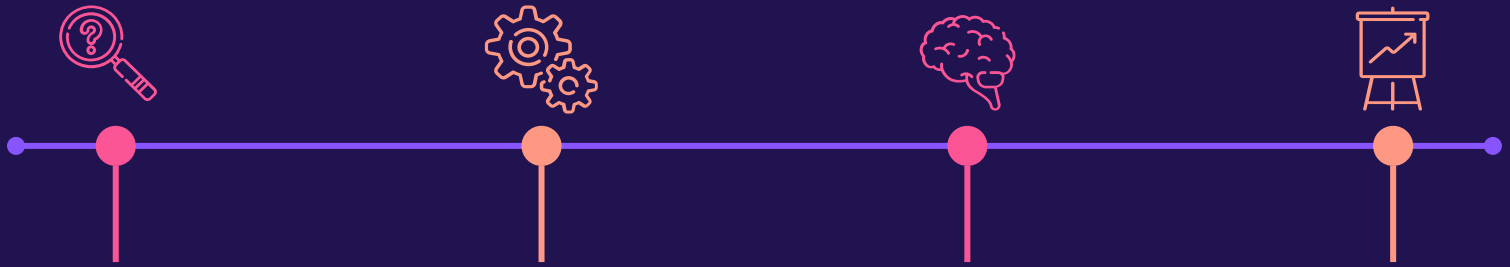
mercari

Servizio di mercato dell'usato molto popolare in Giappone



La previsione del prezzo del prodotto è utile per aiutare i rivenditori online a stabilire prezzi competitivi, migliorando così l'esperienza del cliente e facilitando il processo decisionale.

Pipeline del Progetto



Analisi Esplorativa

dei dati forniti, comprendendone la distribuzione e le peculiarità

Pre-elaborazione dei Dati

per prepararli come input da dare ai modelli di reti neurali e regressione

Creazione dei Modelli

per risolvere il task di regressione

Analisi dei Risultati

per valutare le prestazioni dei modelli tramite metriche appropriate

Analisi dei Dati

01

Valori Mancanti
Per garantire la qualità del dataset

02

Target
Per verificare che tipo di **distribuzione** dei prezzi abbiamo

03

Feature
Per comprendere distribuzione, correlazioni e **impatto sui risultati**

04

Utilità delle Feature
Per identificare quali variabili sono **più rilevanti** per l'obiettivo

OVERVIEW

- Il dataset contiene **1.482.535 valori**

Id	name	item_condition_id	category_name	brand_name	price	shipping	Item_description
0	MLB Cincin...	3	Men/Tops/T-shirts	NaN	10.0	1	No description yet
1	Razer ...	3	Electronics/Computers & Tablets/Comp...	Razer	52.0	0	This keyboard is in great condition and works ...
2	AVA-VIV...	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...
3	Leather Horse ...	1	Home/Home Décor/Home...	NaN	35.0	1	New with tags. Leather horses. Retail for [rm]...
4	24K GOL ...	1	Women/Jewelry/Necklaces	NaN	44.0	0	Complete with certificate of authenticity...
...	...	...	...	...	...	...	...
1482530	Free People ...	5	Kids/Girls 2T-5T/Dresses	Disney	20.0	1	Little mermaid handmade dress never worn size 2t...
1482531	Little...	2	Sports & Outdoors/...	NaN	14.0	0	Lace, says size small but fits medium perfectl...
...	...	...	...	...	...	...	...

01 Valori Mancanti

category_name -> 0.43% di valori mancanti -> diventano «Other»

brand_name -> 42,7% di valori mancanti -> diventano «Unknown»

item_description -> 6 righe mancanti -> diventano «No description available»

Nessun valore mancante nelle altre colonne

02 Target

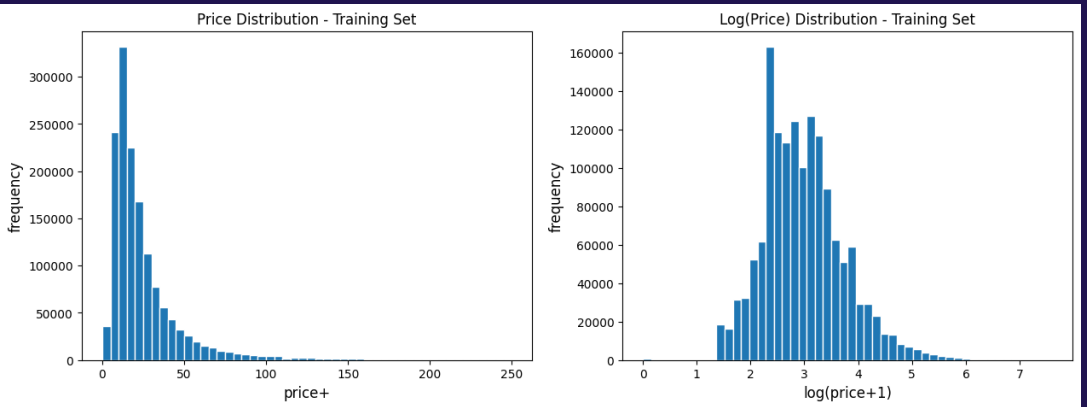
Media: $26,74 con alta deviazione standard ($38,59)

Valore massimo: $2009

75° percentile: $29

Distribuzione potenzialmente distorta a causa di alcuni prodotti molto costosi

Perché la trasformazione logaritmica?

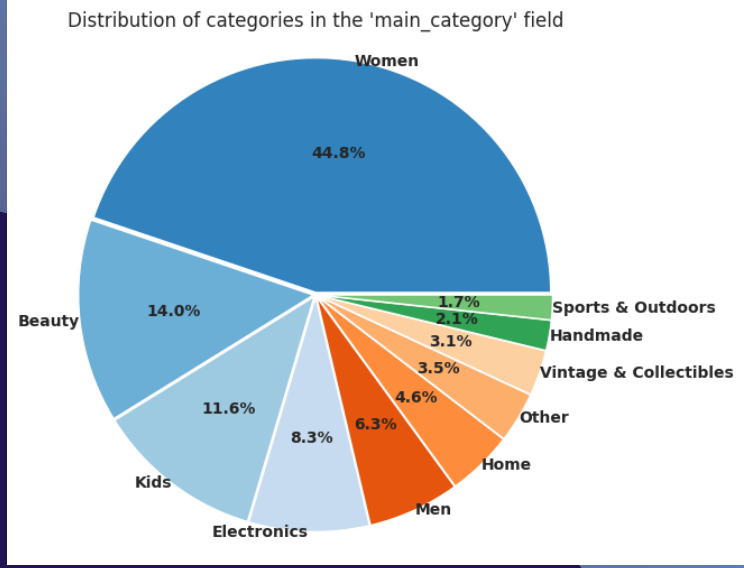


- Migliora la visibilità della distribuzione
- Minimizza l'influenza dei prezzi molto alti
- Rende l'analisi più interpretabile

03 Feature

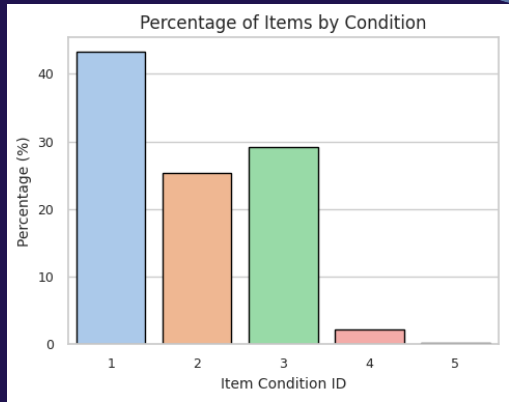
Formato: Categoria Principale / Sottocategoria 1 / Sottocategoria 2
- Esempi: Men/Tops/T-shirts, Bellezza/Trucco/Viso

Dati chiave:
- 1.287 categorie uniche
- 113 prime sottocategorie uniche
- 870 seconde sottocategorie uniche
- 6.327 elementi senza etichetta di categoria



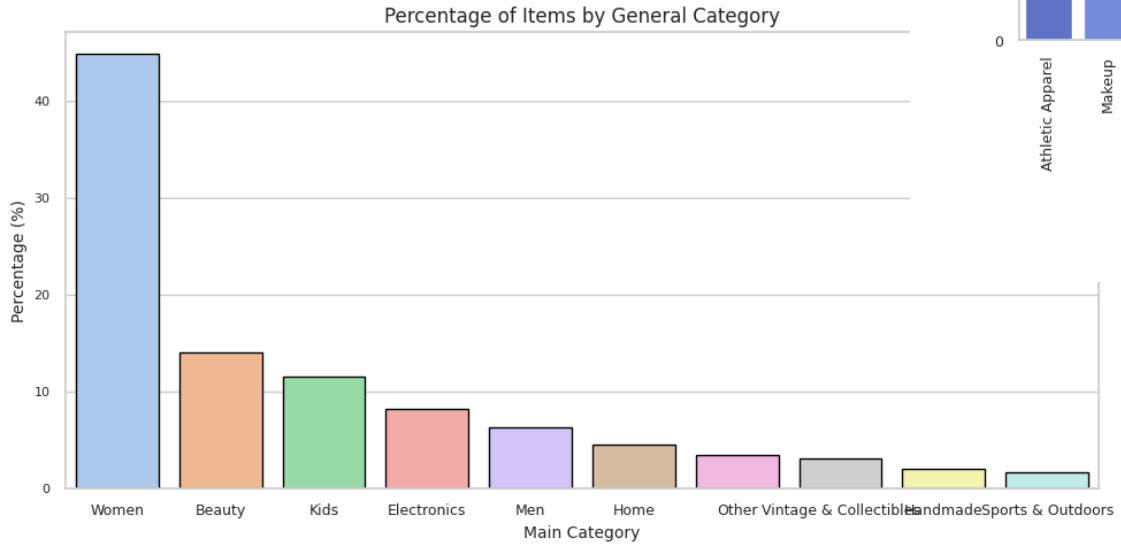
shipping

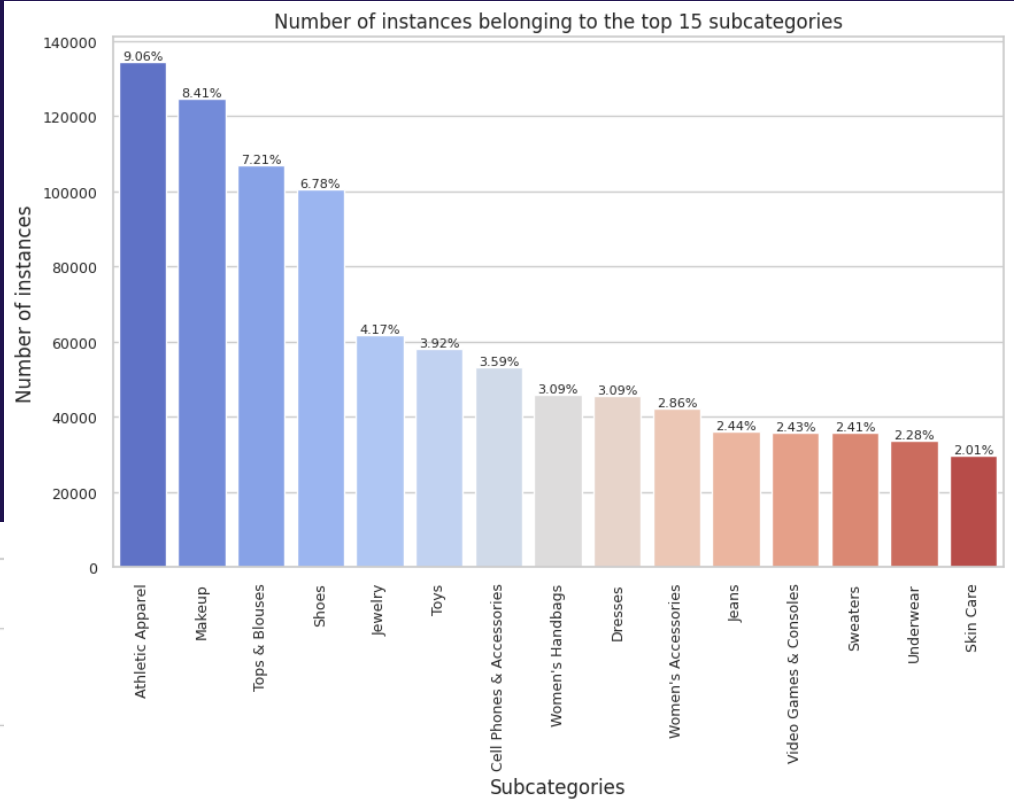
item_condition



category_name

- Le categorie più rappresentate riflettono un forte interesse per il settore della **moda femminile e prodotti di bellezza**
- Categorie come **electronics** e **sports** sono meno diffuse
- La suddivisione consente al modello di distinguere tra categorie principali e dettagli granulari, migliorando la comprensione





Il grafico mostra le 15 sottocategorie più frequenti, evidenziando la loro proporzione nel dataset. Le percentuali sopra le barre aiutano a capire il peso relativo di ciascuna, utile per valutare l'equilibrio tra le categorie

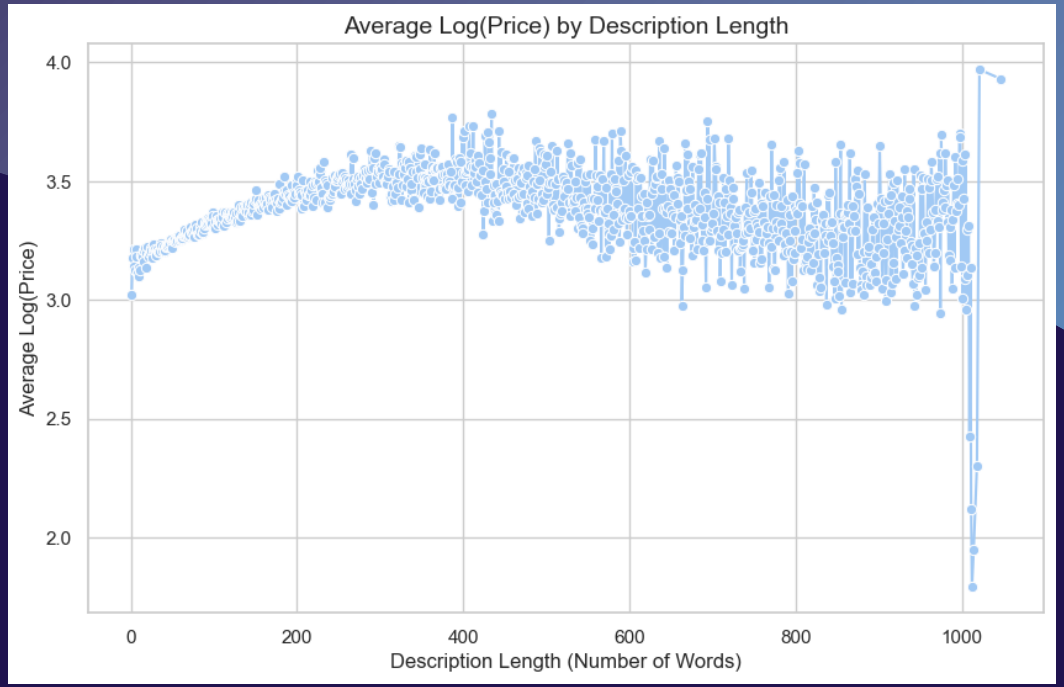
category_name

Osservazioni chiave:

- **Parole comuni**: termini come "brand new", "free shipping", e "good condition" sono ricorrenti in quasi tutte le categorie, indicando l'attenzione dei venditori su stato e vantaggi del prodotto
- **Specificità delle categorie**: alcune parole sono caratteristiche di specifici settori
- **Categorie meno descrittive**: in categorie come Men e Other, molte descrizioni sono generiche o incomplete ("description yet"), suggerendo poca attenzione nella compilazione

item_description

- Il prezzo medio (in scala logaritmica) aumenta inizialmente con la lunghezza della descrizione, fino a circa **400-500 parole**
- Dopo questo punto, l'effetto si stabilizza e mostra maggiore variabilità



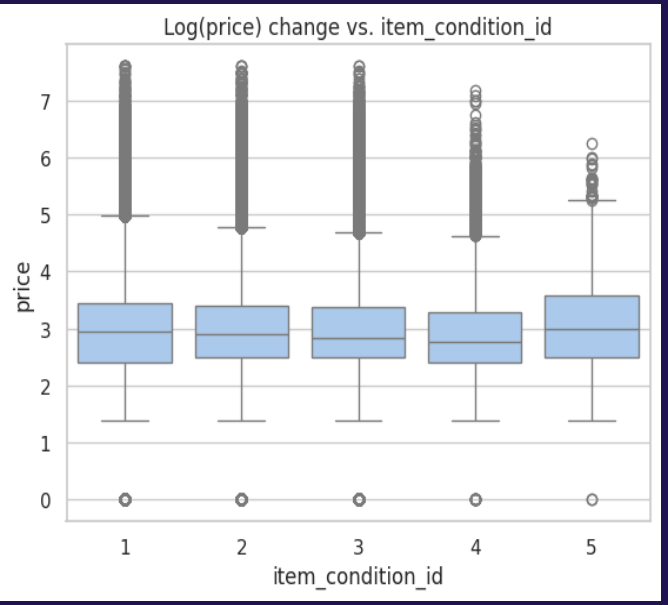
- Descrizioni molto lunghe (>900 parole) mostrano una maggiore dispersione e talvolta una diminuzione del prezzo

Possibile interpretazione: descrizioni più dettagliate possono essere associate a item più costosi, ma oltre un certo limite altri fattori diventano più rilevanti

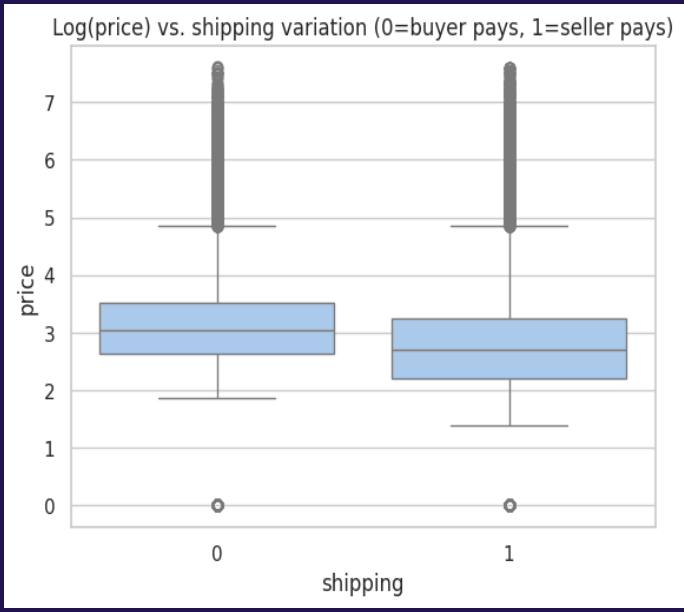
04 Utilità delle Feature

Analizzare l'utilità delle funzionalità è importante per:
- Identificare quali variabili sono più rilevanti per l'obiettivo
- Rimuovere funzionalità inutili o ridondanti per migliorare l'efficienza e ridurre l'overfitting

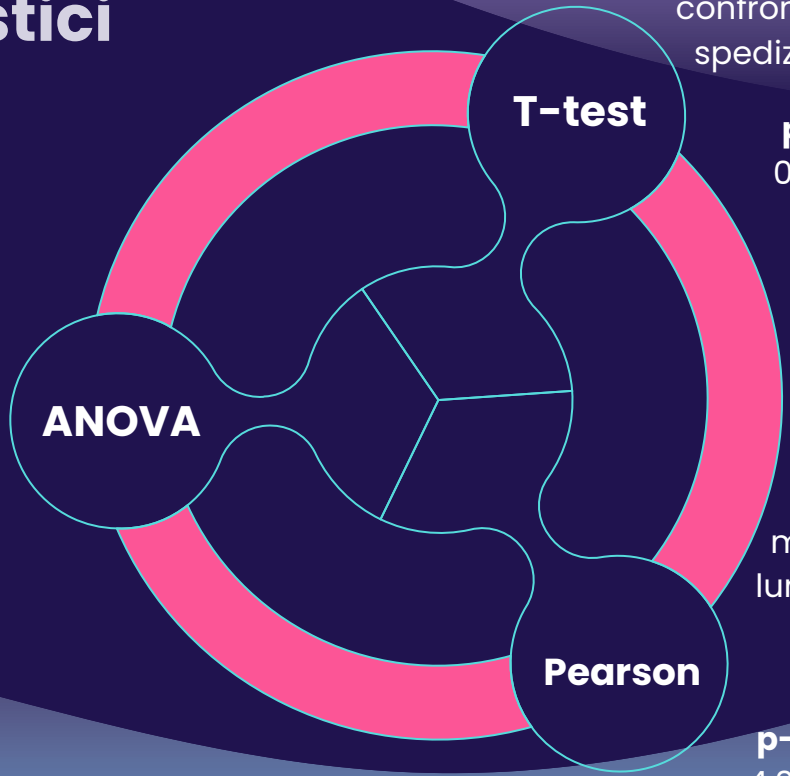
Features Numeriche



- Le medie dei prezzi tra le diverse condizioni sono simili.
- Articoli in condizioni peggiori tendono a costare meno, quelli in migliori condizioni mostrano prezzi leggermente più alti.
- Questa somiglianza potrebbe dipendere dalla natura dei prodotti
- Effetto della condizione sul prezzo presente ma non dominante
- La differenza è modesta, ma la variabile può essere utile, soprattutto per alcune categorie



Test Statistici



confronta l'effetto della spedizione sul prezzo

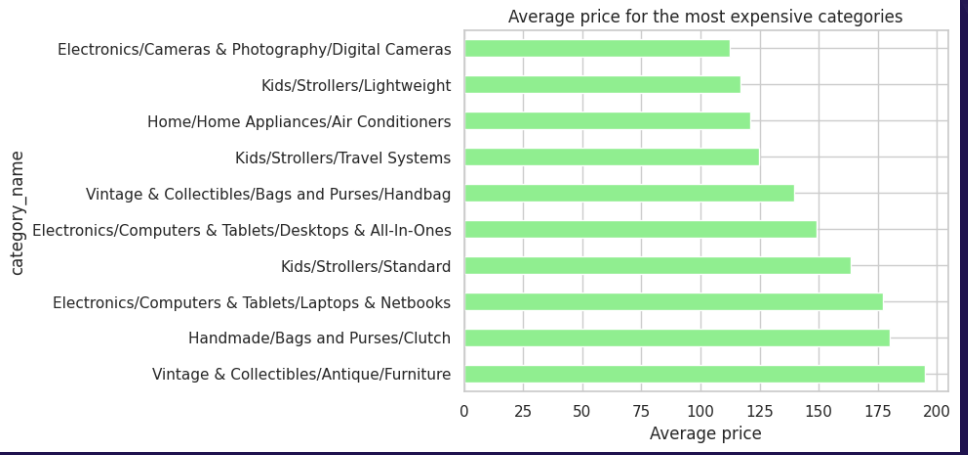
p-value:
0.0

verifica se le condizioni dell'articolo influenzano il prezzo

p-value:
2.8499431199608725e-79

misura la correlazione tra lunghezza della descrizione e prezzo

p-value:
4.092384641591602e-296



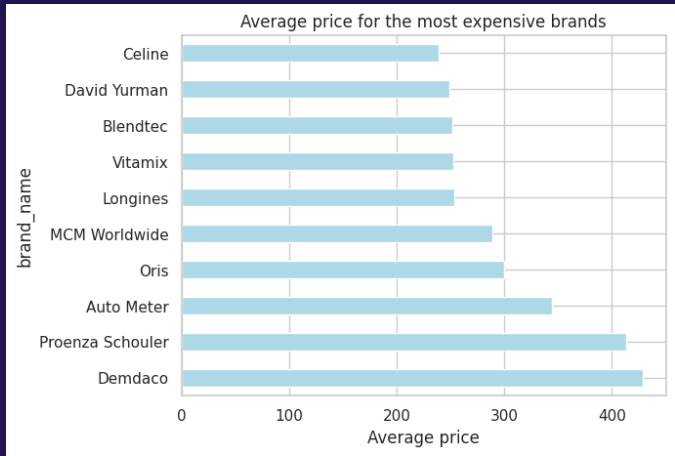
Osservazioni:

- **Categorie premium**: Vintage & Collectibles, Elettronica e Kids hanno prezzi più alti
- **Brand di lusso**: prezzo medio da $238 (Celine) a $429 (Demdaco)
- **Esclusività**: beni di lusso e tecnologici dominano la fascia alta
- **Possibili outlier**: articoli molto costosi possono influenzare la media

Conclusione: categorie e brand di lusso tendono ad avere prezzi più elevati, utile per migliorare le previsioni del modello

Features Categoriche

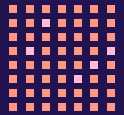
- **Distribuzione complessa**: troppe categorie uniche per analisi diretta
- **Strategie**: confronto prezzi medi/mediani per le principali categorie e marchi



Pre-elaborazione dei Dati

Preprocessing

riduce il rumore e trasforma i dati non strutturati in un **formato standardizzato**, rendendoli adatti per l'analisi e la modellazione



Vettorizzazione

abbiamo adottato **TfidfVectorizer** e **CountVectorizer** per modelli basati su matrici sparse e **Word2Vec** per modelli che richiedono rappresentazioni dense

Encoding

per le feature categoriali, abbiamo usato **One-Hot Encoding** per modelli che sfruttano dati sparsi e **Label Encoding** per modelli che richiedono input numerici compatti

Preprocessing

Tokenizzazione

Utilizziamo *sent_tokenize* e *word_tokenize* per dividere i testi in frasi e poi in token

Pulizia

Vengono rimossi punteggiatura, numeri e caratteri speciali. Parole brevi, numeri e stopwords vengono escluse

Normalizzazione

Tutte le parole sono convertite in minuscolo e lemmatizzate tramite il *WordNet*

Output

La funzione costruita restituisce una stringa di parole processate

Vettorizzazione

CountVectorizer & TfidfVectorizer	Word2Vec
CV: utilizzato per *name*, cattura la frequenza delle parole principali.	Trasforma le parole in vettori numerici densi, catturando le relazioni semantiche tra le parole
Tfidf: applicato ad *item_description*, bilancia l'importanza delle parole più significative	Addestrato sui dati delle colonne *item_description* e *name*
name: (1186028, 76227)	Ogni frase è rappresentata come un vettore medio delle parole che la compongono
item_description: (1186028, 109127)	Shape sui dati di training: (1186028, 100)

Encoding

One-Hot Encoding	Label Encoding
'brand_name', 'item_condition_id', 'shipping', 'main_cat', 'subcat_1', 'subcat_2'	*'brand_name', 'main_cat', 'subcat_1', 'subcat_2'*
Mantiene un formato interpretabile senza introdurre ordini fittizi	Adatto per modelli che richiedono dati numerici scalari, come le reti neurali
Concatenato con feature testuali vettorizzate (TF-IDF, CountVectorizer)	Concatenato con feature testuali rappresentate tramite Word2Vec.
Shape sui dati di training: (1186028, 191165)	Shape sui dati di training: (1186028, 108)

Confronto tra i due Approcci

Caratteristica	Approccio 1 (One-Hot + Bag-of-Words)	Approccio 2 (Label Encoding + Word2Vec)
Dimensione del dataset	(1186028, 191165) → Elevata dimensionalità	(1186028, 107) → Dimensioni molto ridotte
Rappresentazione testuale	Bag-of-Words: parole isolate con peso associato	Word2Vec: vettori densi semantici
Granularità delle feature	Alta granularità, dimensioni proporzionali al vocabolario	Dimensioni fisse e compatte
Colonne categoriali	One-Hot Encoder → Incremento dimensioni per ogni categoria unica	Label Encoder → Compattezza, rischio di perdere rappresentazioni complesse
Efficienza computazionale	Maggiore carico computazionale per modelli lineari	Ridotto sforzo computazionale grazie a vettori compatti
Prestazioni del modello	Funziona meglio con dati abbondanti e modelli che gestiscono bene alta dimensionalità	Potrebbe avere prestazioni migliori in scenari con dati limitati o modelli tradizionali

Creazione dei Modelli

Obiettivo: prevedere il prezzo dei prodotti online

Metodi usati

Ridge Regression

e

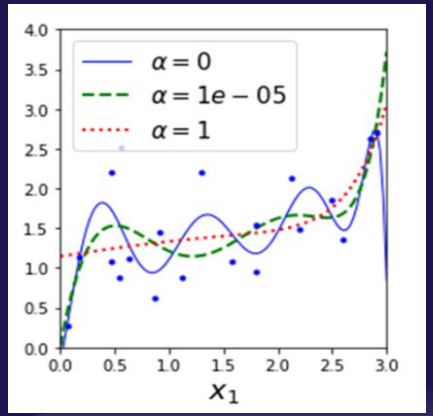
Reti Neurali

Strategia

Testare modelli di crescente complessità

Ridge Regression

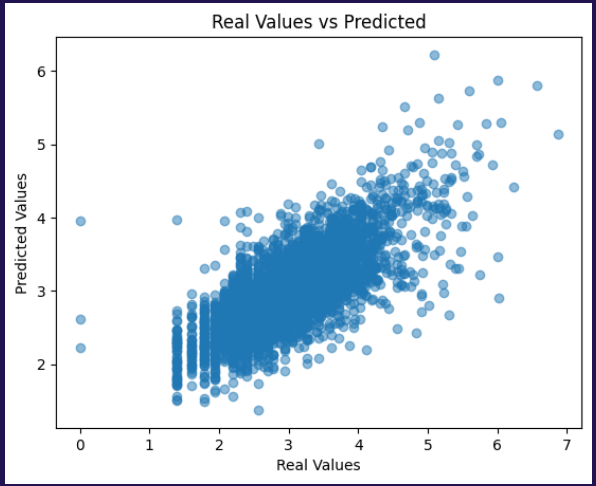
- **Tecnica**: regressione lineare con penalità L2 per prevenire l'overfitting -> spinge i coefficienti ad essere più piccoli
- **Vantaggi**: gestisce la multicollinearità e matrici sparse
- **Ottimizzazione**: GridSearch per α -> controlla la forza della penalizzazione L2
- **Limite**: difficoltà con prezzi estremi
- **Applicazione:** sia alle matrici sparse che compresse
- **Intervallo del target logaritmico:** (0.0 , 7.6059)



Ridge Regression

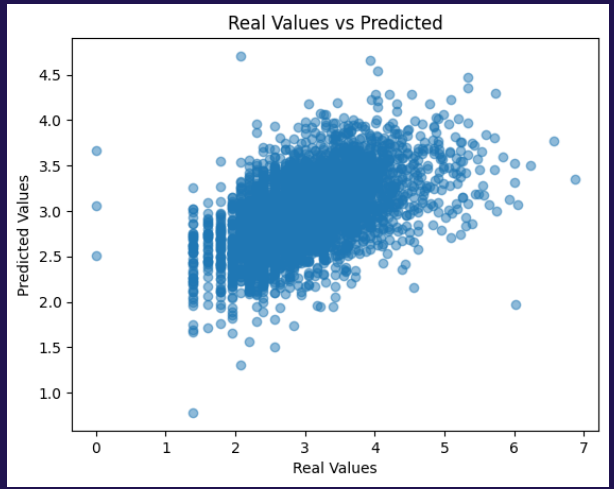
One-Hot, CountVec, TF-IDF

RMSE Val.	α Ott.	RMSE Val.	RMSE Val.
0.70	56.90	0.51	0.38



Label Encoder, Word2Vec

RMSE Val.	α Ott.	RMSE Val.	RMSE Val.
0.74	10.00	0.63	0.48



Reti Neurali

1. Densa a Tre Strati
2. Densa con Regolarizzazione e Ottimizzatore AdamW
3. Neurale con Maggiore Complessità
4. Ibrida Densa e GRU
5. LSTM con Regolarizzazione e Dropout

1. Densa a Tre Strati

Architettura

- Tre strati densi con attivazione ReLUBatch Normalization dopo ogni strato
- **Output**: strato con un singolo neurone

Ottimizzazione & Perdita

- **Ottimizzatore**: Adam
- **Funzione di perdita**: Mean Squared Error (MSE)
- **Metriche**: Mean Absolute Error (MAE) ed RMSE

Strategie di Regolarizzazione

- **Dropout** 20% per ridurre overfitting
- **Batch Normalization** per stabilizzare l'apprendimento

Addestramento

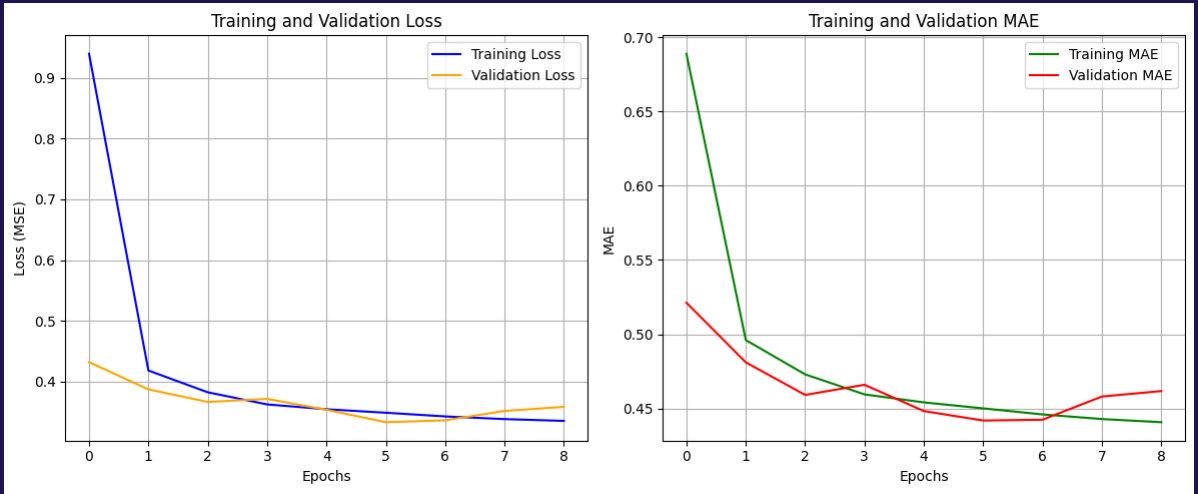
- Early Stopping: interruzione anticipata se val_loss non migliora per 3 epoche
- Batch size: 512
- Epoche: 10

Caratteristiche Principali

✅ Semplice ed efficace per dati non sequenziali
✅ Struttura ridotta rispetto a modelli più complessi
✅ Solida base per miglioramenti successivi

Risultati

Predicted Price	Real Price
31.341930	23.0
20.418499	9.0
29.739470	15.0
35.193943	17.0
18.368992	19.0
24.473978	19.0



0.60

RMSE

0.46

MAE

2. Densa con Regolarizzazione e Ottimizzatore AdamW

Architettura

- Tre strati densi con attivazione Relu
- Output: Strato con un singolo neurone

Ottimizzazione & Perdita

- **Ottimizzatore**: AdamW (learning rate iniziale più alto: 0.01)
- **Funzione di perdita**: MSE
- **Metriche**: Mean Absolute Error (MAE) ed RMSE

Strategie di Regolarizzazione

- **Regolarizzazione L2** per ridurre la complessità del modello
- **Batch Normalization** per stabilizzare l'apprendimento
- **Dropout** 20% per ridurre l'overfitting

Addestramento

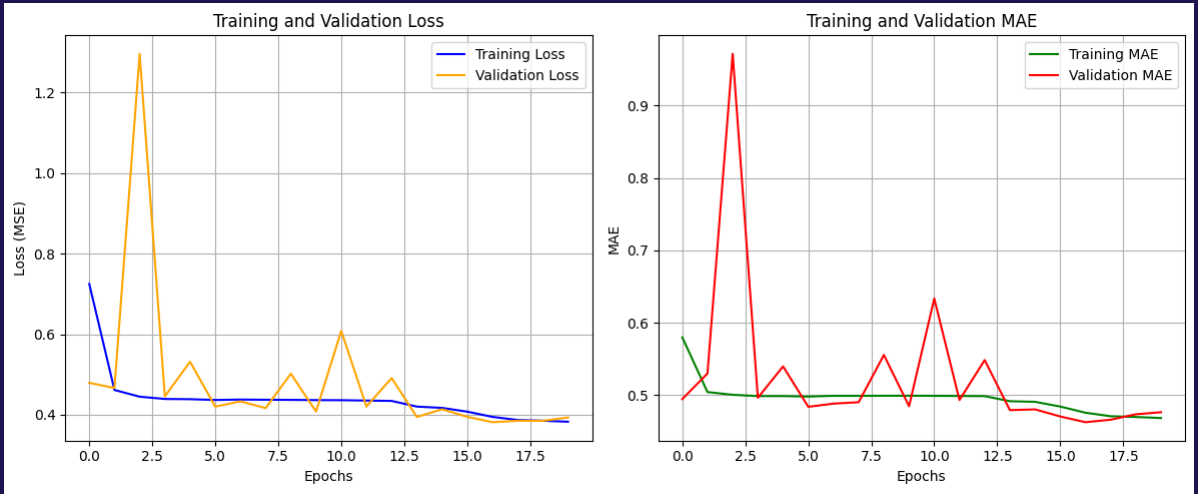
- **Early Stopping**: stop dopo 5 epoche senza miglioramenti
- **ReduceLROnPlateau**: riduce il learning rate (factor=0.1) se val_loss non migliora per 3 epoche
- **Batch size** più grande (1024) per stabilizzare l'aggiornamento dei pesi
- Numero di epoche aumentato a 20

Caratteristiche Principali

✅ Migliore gestione dell'overfitting rispetto al Modello 1
✅ Ottimizzazione più aggressiva con AdamW e regolazione dinamica del learning rate
✅ Migliore generalizzazione grazie alla regolarizzazione L2

Risultati

Predicted Price	Real Price
28.935562	23.0
15.419250	9.0
19.784893	15.0
20.667305	17.0
15.656612	19.0
16.094460	19.0



0.61

RMSE

0.47

MAE

3. Neurale con Maggiore Complessità

Architettura

- Quattro strati nascosti con attivazione ReLU
- **Batch Normalization** dopo ogni strato
- Output: strato con un singolo neurone

Ottimizzazione & Perdita

- **Ottimizzatore**: AdamW (learning rate ridotto: 0.001)
- **Funzione di perdita**: MSE
- **Metriche**: MAE e RMSE

Strategie di Regolarizzazione

- **Regolarizzazione L2 (0.0001)** per controllare la crescita dei pesi
- **Batch Normalization** per stabilizzare l'apprendimento
- **Dropout** 30% per ridurre l'overfitting

Addestramento

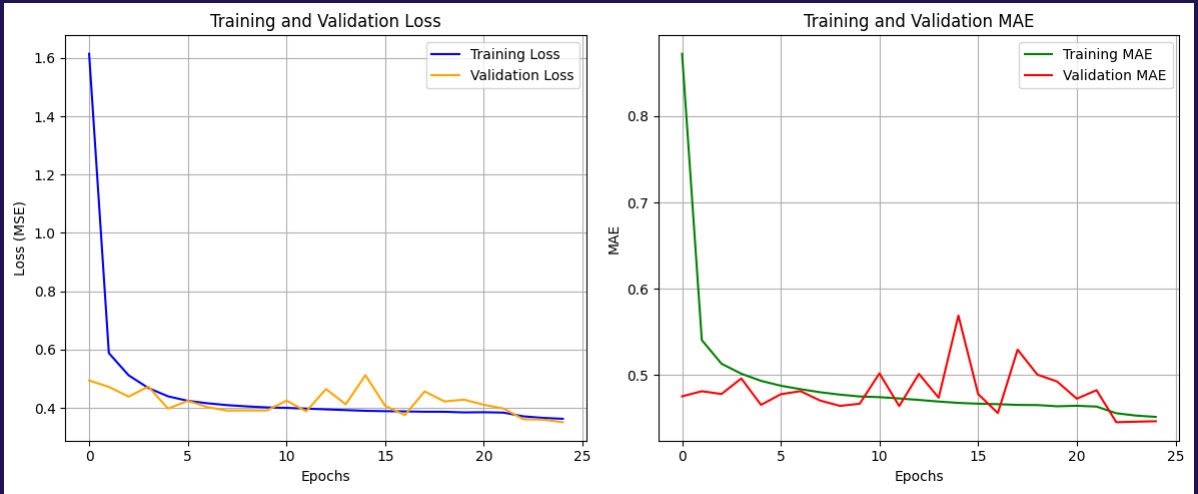
- **Early Stopping**: Interruzione anticipata dopo 10 epoche senza miglioramenti
- **ReduceLROnPlateau**: diminuzione del learning rate (factor=0.1) se val_loss non migliora per 5 epoche
- **Batch size** ottimizzato

Caratteristiche Principali

✅ Aumenta la complessità del modello con più strati nascosti
✅ Regolarizzazione più intensa rispetto ai modelli precedenti
✅ Migliore adattabilità ai dataset complessi

Risultati

Predicted Price	Real Price
31.638262	23.0
18.377466	9.0
22.717234	15.0
26.245554	17.0
16.027504	19.0
14.386086	19.0



0.58

RMSE

0.45

MAE

4. Ibrida Densa e GRU

Architettura e strategie di regolarizzazione

- Branch Denso:
  - Tre strati nascosti con attivazione ReLU
  - Batch Normalization e Dropout (30%) dopo ogni strato
  - Regolarizzazione L2 (0.0001)
- Branch GRU:
  - GRU (128) e GRU (64) per catturare pattern sequenziali
  - Reshape per adattare la forma dell'input al GRU
  - Dropout (30%)
- Concatenazione dei due rami (denso e GRU)
- Output: Strato con un singolo neurone

Ottimizzazione & Perdita

- **Ottimizzatore**: AdamW (learning rate: 0.001)
- **Funzione di perdita**: MSE
- **Metriche**: MAE e RMSE

Addestramento

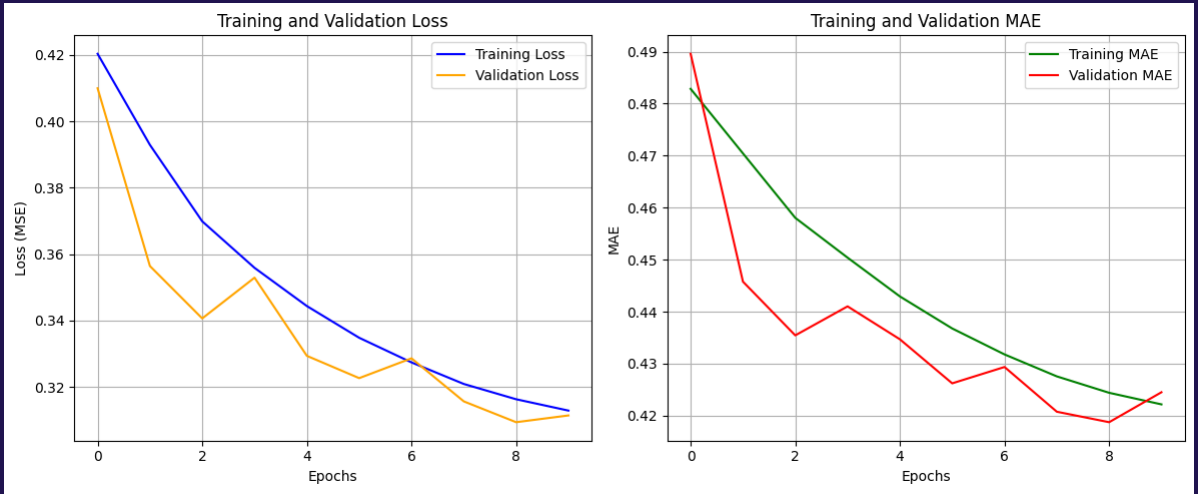
- **Early Stopping**: interruzione anticipata se val_loss non migliora per 3 epoche
- **Batch size** ottimizzato
- **ReduceLROnPlateau**: diminuzione del learning rate se val_loss non migliora per 5 epoche

Caratteristiche Principali

✅ Introduzione di GRU per gestire sequenze temporali e pattern complessi
✅ Rami combinati (denso + GRU) per apprendere sia relazioni non sequenziali che sequenziali
✅ Adattamento del modello a dati temporali o sequenziali

Risultati

Predicted Price	Real Price
16.316046	23.0
16.789236	9.0
26.572990	15.0
23.174654	17.0
20.503637	19.0
19.216766	19.0



0.55

RMSE

0.42

MAE

5. LSTM con Regolarizzazione e Dropout

Architettura e Strategie di Regolarizzazione

Strati LSTM:

- LSTM (128 unità), con attivazione ReLU e return_sequences=True per le sequenze
- LSTM (64 unità), con attivazione ReLU e return_sequences=True
- LSTM (32 unità), con attivazione ReLU e senza return_sequences (per il livello finale)
- Dropout (30% e 20%) dopo ogni strato LSTM per ridurre l'overfitting

Output: Strato denso con un singolo neurone (attivazione lineare)

Ottimizzazione & Perdita

- **Ottimizzatore**: RMSprop (learning rate: 0.0001)
- **Funzione di perdita**: MSE
- **Metriche**: MAE

Addestramento

- **ReduceLROnPlateau**: Riduzione del learning rate quando val_loss non migliora per 5 epoche
- **Batch size** ottimizzato (64) per stabilizzare l'apprendimento

Caratteristiche Principali

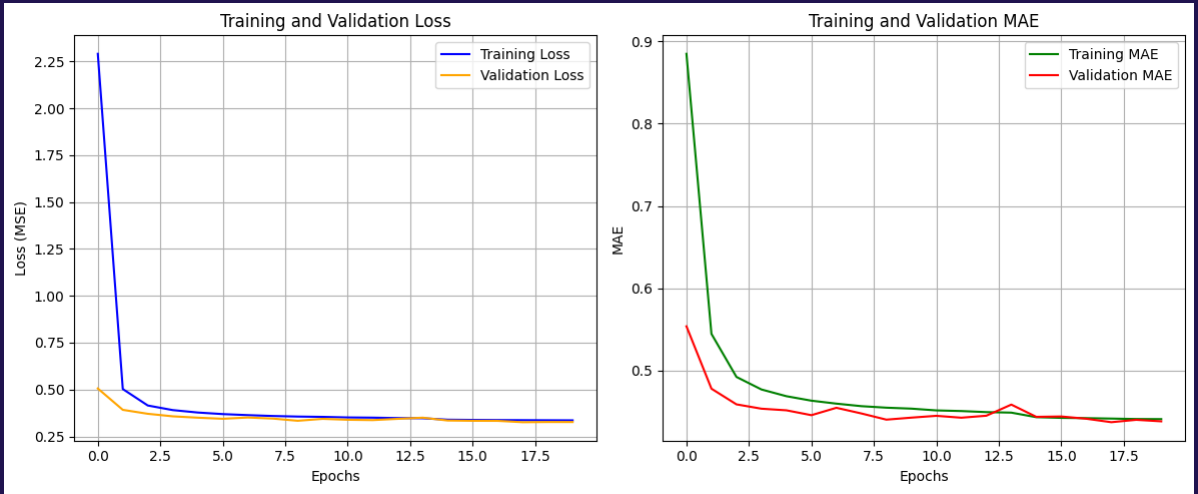
✅ Utilizzo di LSTM per memorizzare informazioni a lungo termine, ideale per dati temporali

✅ Tre strati LSTM per apprendere le dipendenze temporali nel dataset

✅ Aumento della capacità di generalizzazione grazie al dropout

Risultati

Predicted Price	Real Price
29.250782	23.0
19.010366	9.0
22.046997	15.0
24.465235	17.0
16.940519	19.0
22.100513	19.0



0.57

RMSE

0.44

MAE

Analisi dei Risultati

Metriche	Prima NN	Seconda NN	Terza NN	Quarta NN	Quinta NN
RMSE Test	0.60	0.61	0.58	**0.55**	0.57
MAE Test	0.46	0.47	0.45	**0.42**	0.44

- Miglior Modello: 4° Rete Neurale (GRU) -> La capacità di generalizzazione migliorata grazie ai layer GRU
- Secondo Miglior Modello: 5° Rete Neurale (LSTM) -> : Ottima alternativa per la memorizzazione delle informazioni temporali
- Modello Meno Performante: 2° Rete Neurale > Prestazioni inferiori rispetto agli altri modelli

- Le prestazioni sono simili tra i modelli, con margini di miglioramento
- La Quarta Rete Neurale (GRU) si distingue per il miglior comportamento di generalizzazione.
- I grafici mostrano un buon comportamento di generalizzazione senza segni di overfitting e un miglioramento continuo dell'errore medio assoluto (MAE) durante l'addestramento.

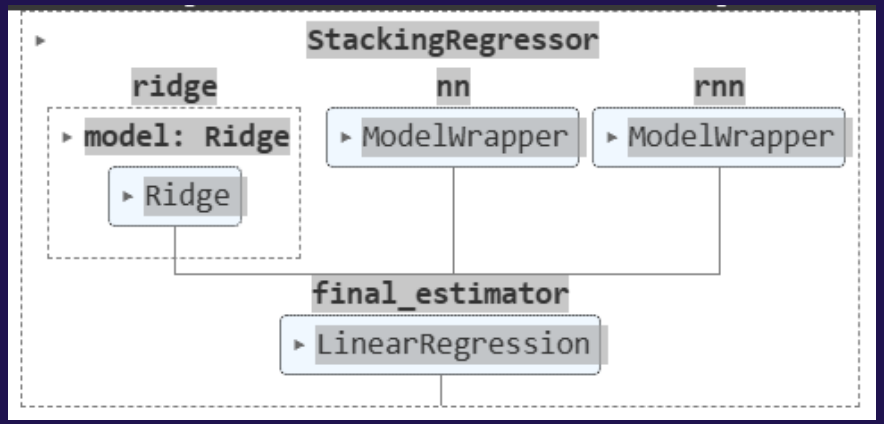
Esperimenti

Ensembling

- I tre modelli scelti (Ridge, quarta rete neurale, RNN) sono combinati tramite **Stacking** Regressor, con una regressione lineare finale come modello di combinazione

Processo:

- Ogni modello base produce previsioni
- Un modello finale (Linear Regression) è addestrato sui risultati dei modelli base per migliorare la previsione finale



RMSE = 0.5510 **MAE = 0.4245**

Questo approccio **migliora le performance** rispetto a ciascun modello individuale, sfruttando le diverse capacità dei modelli base nel catturare differenti aspetti dei dati.

Explenability

Per una maggiore comprensione dei modelli, abbiamo scelto un approccio di ***explanability***

Eli5

libreria Python progettata per l'interpretabilità dei modelli di machine learning

- Rende il modello comprensibile e giustificabile
- K-Fold Cross-Validation: ottimizzazione del modello con valutazione di MAE e RMSE su split di dati
- Pesi del Modello: valutazione dell'impatto di ogni feature sulla previsione
- Visualizzazione del contributo di ciascuna feature nelle predizioni per esempi specifici

LIME

libreria Python progettata per l'interpretabilità delle reti neurali, note per essere più difficili da interpretare

- Le reti neurali sono spesso considerate "black-box" e richiedono strumenti per estrarre e spiegare le decisioni
- LIME offre spiegazioni locali per singole predizioni, mostrando il contributo delle feature più rilevanti per una specifica previsione
- Abbiamo scelto la terza NN per semplicità computazionale
- Le feature più influenti per la predizione vengono mostrate graficamente

Eli5

Weight[?]	Feature
+2.632	<BIAS>
+1.194	category_name__electronics/computers & tablets/laptops & netbooks
+1.053	brand_name__kendra scott
+1.002	item_description__authentic
+0.945	brand_name__louis vuitton
+0.878	item_description__box
+0.692	name__mcm
+0.678	brand_name__david yurman
	... 78314 more positive ...
	... 77644 more negative ...

y (score **2.523**) top features

Contribution[?]	Feature
+2.632	<BIAS>
+0.105	shipping: Highlighted in text (sum)
+0.102	item_condition_id: Highlighted in text (sum)
+0.074	item_description: Highlighted in text (sum)
-0.096	category_name: Highlighted in text (sum)
-0.122	brand_name: Highlighted in text (sum)
-0.173	name: Highlighted in text (sum)

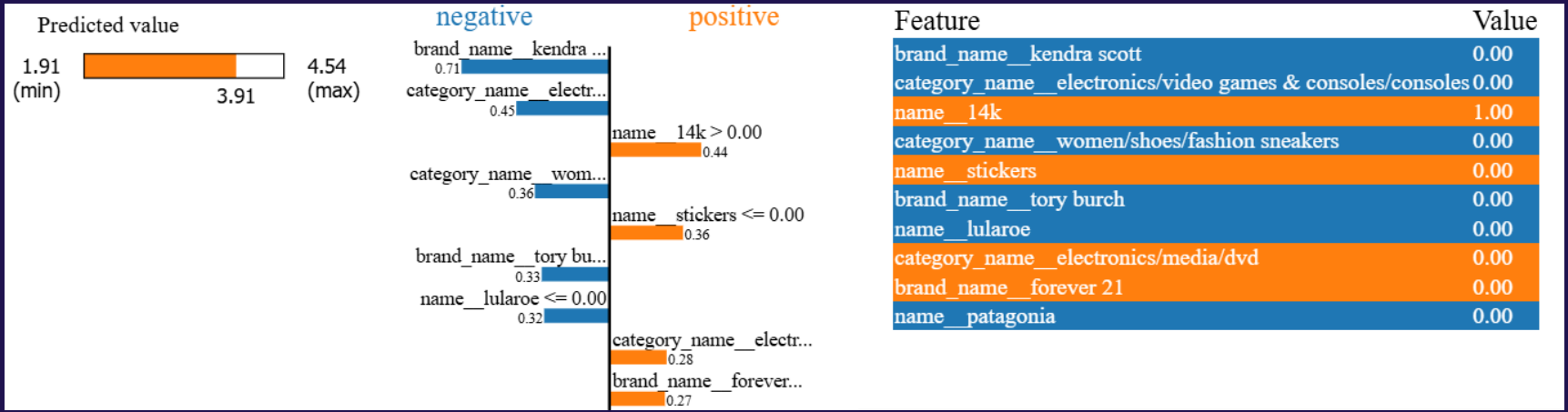
name: muscle t-shirt
category_name: women/tops & blouses/t-shirts
brand_name: missing
shipping: 0
item_condition_id: 3
item_description: what goes better with summer than tacos & tequila? chill out with friends sporting this great beachwear cover or wear as a stand alone with that oh so sexy bralette or bikini top! don't forget your cool shades! great condition! worn once, no stains, holes, rips or treats.

y (score **3.671**) top features

Contribution[?]	Feature
+2.632	<BIAS>
+0.354	item_description: Highlighted in text (sum)
+0.220	category_name: Highlighted in text (sum)
+0.160	name: Highlighted in text (sum)
+0.105	shipping: Highlighted in text (sum)
+0.102	item_condition_id: Highlighted in text (sum)
+0.098	brand_name: Highlighted in text (sum)

name: razer blackwidow chroma keyboard
category_name: electronics/computers & tablets/components & parts
brand_name: razer
shipping: 0
item_condition_id: 3
item_description: this keyboard is in great condition and works like it came out of the box. all of the ports are tested and work perfectly. the lights are customizable via the razer synapse app on your pc.

LIME



- Le feature evidenziate in blu hanno un impatto negativo sulla predizione del prezzo
- Le feature evidenziate in arancione hanno un impatto positivo
- Il valore numerico accanto alle feature indica la loro influenza relativa sulla predizione
- Il modello potrebbe attribuire un valore elevato agli oggetti che contengono "14k" (probabilmente riferito all'oro)
- Alcuni brand e categorie possono abbassare il valore del prezzo stimato
- LIME mostra una previsione compresa tra **1.91 e 4.54** perché il modello sta lavorando su un sottoinsieme della **scala trasformata** dei prezzi

Grazie

Anna Marika Biasco

Francesca Pulerà

Massimo Zarantonello