

CORSO DI LAUREA IN INFORMATICA



Link alla repository

Autori

Raffaele Coppola

Docente

Prof. Giuseppe Polese

Prof. Loredana Caruccio

Indice

1	Introduzione al problema	3
1.1	L'AI nel settore automobilistico	3
1.1.1	Le sfide principali	3
1.2	Il problema affrontato	3
1.2.1	Perché Carval?	3
1.2.2	Regressione e classificazione	4
1.3	Struttura della repository	4
2	Trattamento dei dati	5
2.1	Data understanding	5
2.1.1	Raccolta dei dati	5
2.1.2	Documentazione dei dati	5
2.1.3	Esplorazione dei dati	6
2.2	Data preparation	11
2.2.1	Pulizia dei dati	11
2.2.2	Il problema delle scale	14
2.2.3	Ingegneria delle caratteristiche	16
2.2.4	Bilanciamento delle classi	16
3	Sviluppo del modello	17
3.1	Selezione del modello	17
3.1.1	Fase di addestramento	18
3.2	Tecniche di valutazione	20
3.2.1	Metriche di validazione	20
4	Usabilità	23
4.1	Interfaccia grafica	23
4.1.1	Perché Streamlit?	23
5	Considerazioni finali	25
5.1	Possibili miglioramenti	25
5.1.1	Nei dati	25

5.1.2	Nel modello	25
-------	-----------------------	----

Elenco delle figure

1	Matrice di correlazione delle caratteristiche numeriche	7
2	Studio delle frequenze delle caratteristiche	8
4	Studio dei valori nulli	12
5	Bilanciamento delle classi	17
6	Validation loss e accuracy al variare di max_depth	18
7	Importanza delle caratteristiche	19
8	Validazione incrociata	20
9	Matrice delle confusioni (primo e ultimo fold)	21
10	Interfaccia grafica	23

Elenco delle tabelle

1	Statistiche descrittive delle caratteristiche numeriche	6
2	Studio dei domini delle caratteristiche	12
3	Valori metriche di validazione (classificazione)	21
4	Valori metriche di validazione (regressione)	22

1 Introduzione al problema

1.1 L'AI nel settore automobilistico

L'intelligenza artificiale e il Machine Learning stanno rivoluzionando numerosi settori, compreso quello della valutazione automobilistica. L'analisi automatizzata dei veicoli consente di ridurre notevolmente l'incertezza e di migliorare i processi decisionali per venditori, acquirenti e istituzioni finanziarie.

1.1.1 Le sfide principali

Come qualsiasi approccio basato sull'AI, l'implementazione di un modello predittivo per la valutazione delle auto presenta alcune criticità fondamentali:

- **qualità e disponibilità dei dati.** La qualità dei dati ha un forte impatto rispetto alle prestazioni del modello.
- **explainability.** Risulta essenziale l'interpretabilità del modello in modo da poter spiegare agli utenti finali come ogni caratteristica influenzi la valutazione finale.

1.2 Il problema affrontato

L'obiettivo principale del progetto è sviluppare un modello di machine learning che esegue un task di apprendimento supervisionato per cui viene predetto il prezzo di un'auto usata sulla base di un'esperienza maturata da un dataset che include diverse caratteristiche delle vetture automobilistiche.

1.2.1 Perché Carval?

Carval nasce con l'obiettivo di offrire una valutazione data-driven, aiutando diversi tipi di utenti a prendere decisioni più informate.

Il sistema è stato progettato per rispondere alle esigenze di diversi attori:

- **acquirenti** di auto usate. Chi desidera acquistare un veicolo può utilizzare Carval per verificare se il prezzo richiesto è in linea.
- Venditori **privati** e concessionari. Chi vende un'auto può ottenere una stima realistica del valore del proprio veicolo, facilitando la definizione di un prezzo competitivo.
- Compagnie di **assicurazione**. Le assicurazioni possono impiegare il sistema per stimare il valore residuo delle auto e pianificare meglio le operazioni di vendita o sostituzione.

L'idea alla base di Carval è quella di democratizzare l'accesso a valutazioni affidabili, riducendo l'asimmetria informativa tra venditori e acquirenti.

1.2.2 Regressione e classificazione

Per una maggiore modellazione del problema e sperimentazione delle analisi, si è deciso di affrontarlo da due diverse prospettive: sia come task di regressione, sia come task di classificazione.

Nel caso della regressione, l'obiettivo del modello è predire il prezzo esatto di un veicolo automobilistico, mentre nel caso della classificazione il prezzo è suddiviso in fasce di valore (meno di $10k$, $10k - 20k$ ecc.), ed il modello, anziché stimarne il valore esatto, deve imparare a classificare un'auto all'interno di una determinata fascia di prezzo.

1.3 Struttura della repository

La repository è organizzata in modo da garantire una chiara separazione logica tra dati, codice e documentazione. Di seguito viene descritta la struttura delle principali cartelle e file presenti:

- *datasets*: contiene il dataset di partenza reperito da Kaggle e il dataset pre-elaborato.
- *diagrams*: contiene tutti i grafici generati durante lo sviluppo della pipeline;
- *docs*: raccoglie la documentazione e la presentazione relativa al progetto;
- *notebooks*: contiene tutti i notebook sviluppati per il progetto (*.ipynb*).
- *pipeline*: contiene tutti i file sviluppati per l'esecuzione della pipeline di ml (*.py*).
- *README.md*: fornisce informazioni su come installare, configurare ed utilizzare il progetto;
- *requirements.txt*: contiene la lista delle dipendenze necessarie per eseguire gli script della repository, facilitando la configurazione dell'ambiente di sviluppo.

Questa organizzazione consente una gestione efficiente del progetto, facilitando la manutenibilità e l'eventuale estensione delle funzionalità. La repository è disponibile qui.

2 Trattamento dei dati

2.1 Data understanding

2.1.1 Raccolta dei dati

Per addestrare il modello di Machine Learning, il dataset di riferimento è stato acquisito dalla piattaforma Kaggle, al seguente link: <https://www.kaggle.com/datasets/tunguz/used-car-auction-prices>.

La scelta di utilizzare un dataset di Kaggle, anziché raccogliere dati autonomamente tramite web scraping o con altre tecniche, è motivata da fattori relativi alla **facilità di accesso**, in quanto l'uso di un dataset pubblico consente di garantire la riproducibilità degli esperimenti, evitando problemi legati alla legalità della raccolta dati e alla gestione di grandi volumi di informazioni eterogenee; inoltre, la raccolta di dati tramite scraping potrebbe essere un processo lungo e complesso, con rischi legati all'obsolescenza dei dati e alla necessità di una manutenzione costante degli script di raccolta.

Tuttavia, la reperibilità dei dati rimane una delle sfide principali nell'applicazione dell'IA. I dataset pubblici possono presentare limitazioni in termini di rappresentatività rispetto al mercato reale.

Limiti dei dati. Nonostante l'immediatezza della reperibilità del dataset, esso presenta alcune limitazioni che devono per forza di cose essere tenute in considerazione. Le principali restrizioni riguardano la copertura temporale e la rappresentatività geografica. Infatti, il dataset è composto esclusivamente da dati del mercato nord-americano fino al 2015. Ciò significa che il modello è stato addestrato su informazioni storiche e potrebbe non catturare le tendenze di mercato più recenti, non rendendo del tutto valide le stime ottenute per altri paesi in cui sono presenti dinamiche di mercato differenti.

Tuttavia, uno dei principali ostacoli nell'applicazione dell'intelligenza artificiale alla valutazione delle automobili è l'impossibilità di rappresentare completamente l'intero panorama mondiale dei veicoli disponibili in un singolo dataset. Ogni mercato automobilistico è verosimilmente influenzato da fattori economici, culturali, normativi e tecnologici che variano significativamente da paese a paese.

2.1.2 Documentazione dei dati

Il dataset di partenza è costituito da 558811 osservazioni caratterizzate da 16 feature.

Di seguito è riportata la lista delle caratteristiche con relativa descrizione.

- **year**: anno di produzione dell'auto.
- **make**: marca del veicolo.

- **model**: modello specifico dell'auto.
- **trim**: allestimento o versione del modello.
- **body**: tipo di carrozzeria del veicolo.
- **transmission**: tipo di trasmissione.
- **vin**: codice univoco per identificare il veicolo.
- **state**: stato di registrazione del veicolo.
- **seller**: venditore del veicolo.
- **condition**: condizione generale del veicolo.
- **odometer**: chilometraggio dell'auto.
- **color**: colore carrozzeria auto.
- **interior**: colore degli interni dell'auto.
- **mmr**: indicatore del valore di mercato su base storica.
- **sellingprice**: prezzo effettivo di vendita del veicolo (**target**).
- **saledate**: data in cui l'auto è stata venduta.

2.1.3 Esplorazione dei dati

In Tabella 1 sono riportate parametri statistici standard riguardo le caratteristiche numeriche.

	Count	Mean	Std	Min	Max
year	558,811	2010.04	3.97	1982	2015
condition	547,017	3.42	0.95	1.0	5.0
odometer	558,717	68,323.20	53,397.75	1.0	999,999
mmr	558,811	13,769.32	9,679.87	25.0	182,000
sellingprice	558,811	13,611.26	9,749.66	1.0	230,000

Tabella 1: Statistiche descrittive delle caratteristiche numeriche

Correlazione con il target. Nell'analisi dei dati, la matrice di correlazione è uno strumento essenziale per comprendere le relazioni tra le variabili del dataset. La correlazione misura il grado di associazione lineare tra due variabili e fornisce informazioni cruciali sulla loro dipendenza reciproca.

Il valore della correlazione varia tra -1 e 1 :

- *Correlazione positiva* ($+1$). Quando una variabile aumenta, anche l'altra tende ad aumentare.
- *Correlazione negativa* (-1). Quando una variabile aumenta, l'altra tende a diminuire.
- *Correlazione nulla* (0): Le due variabili non presentano alcuna relazione lineare.

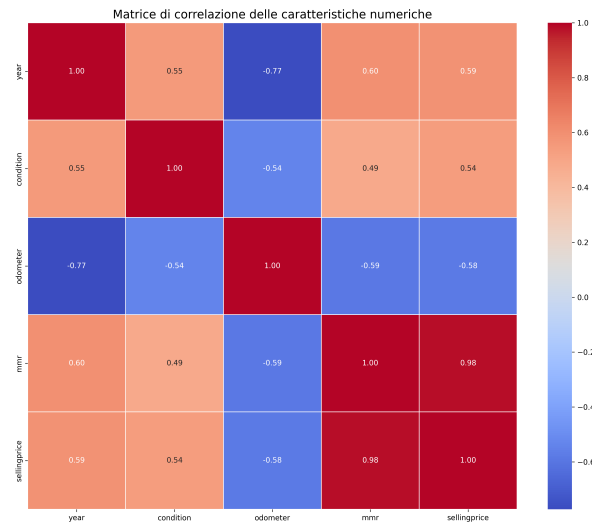


Figura 1: Matrice di correlazione delle caratteristiche numeriche

Dall'analisi della matrice di correlazione delle caratteristiche numeriche presente in Figura 1, si osserva una forte correlazione tra la variabile mmr e sellingprice. Successivamente è spiegata la problematica di questa forte correlazione.

Il problema di data leakage. Un aspetto critico nella costruzione di un modello di machine learning è il rischio di **data leakage**. Tale problema si presenta quando un modello è capace di lavorare in maniera performante ed accurata in fase di addestramento, ma non in fase di rilascio. Ciò è possibile perché magari si utilizzano caratteristiche disponibili nel training ma che non saranno disponibili a "run time" (**leaky predictor**).

Nel dataset di riferimento, l'uso di mmr come variabile predittiva introduce un rischio significativo di data leakage, in quanto contiene per sua natura informazioni derivate direttamente dal target. Se

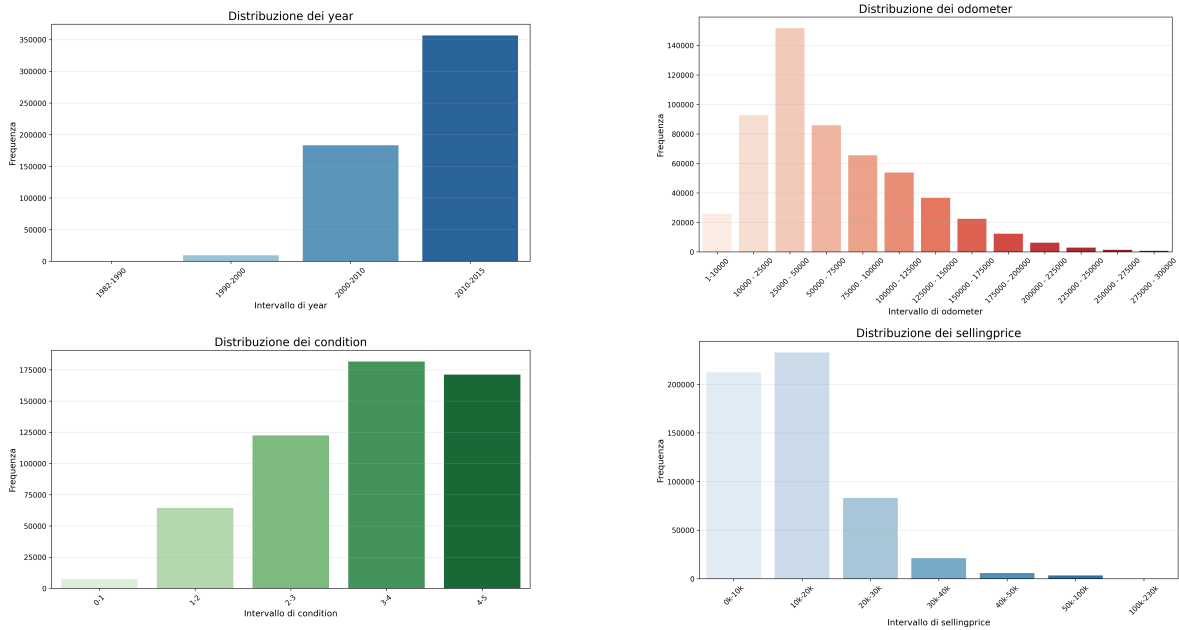


Figura 2: Studio delle frequenze delle caratteristiche

il valore mmr è stato calcolato utilizzando informazioni disponibili solo dopo la vendita del veicolo, allora il modello potrebbe imparare una relazione artefatta, sfruttando dati che in una situazione reale non sarebbero disponibili al momento della valutazione del prezzo.

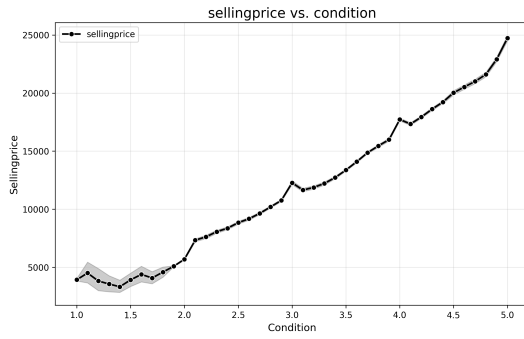
Per ovviare al problema, la caratteristica mmr è stata sin da subito esclusa, così da garantire che il modello possa apprendere solamente da caratteristiche disponibili prima della vendita del veicolo.

Distribuzione caratteristiche numeriche. In Figura 2 sono riportati i grafici relativi alle frequenze delle caratteristiche numeriche.

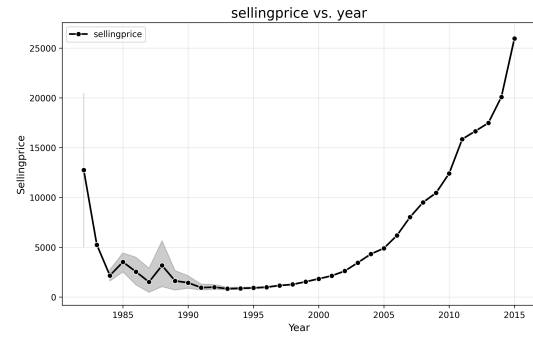
Studio variazione target. L'analisi della variazione del target, ovvero il prezzo di vendita (selling-price) in funzione di diverse variabili indipendenti, è una fase cruciale per comprendere le dinamiche di mercato e identificare eventuali pattern che influenzano il valore delle automobili. Per questo motivo, si è deciso di analizzare alcuni grafici che illustrano l'andamento del prezzo in relazione a caratteristiche chiave dei veicoli.

Da Figura 3a si può notare che all'aumentare del punteggio assegnato alle condizioni aumenta anche il valore del prezzo.

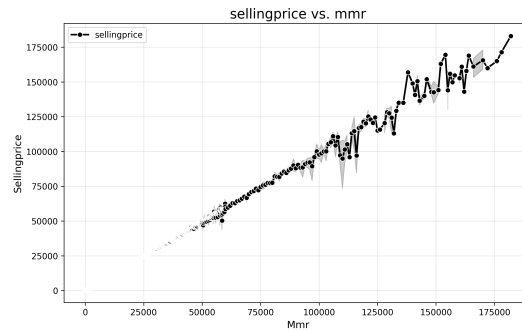
Da Figura 3b si evince che dall'anno 1995 c'è stato un andamento crescente con variazioni positive della pendenza della retta che descrive la relazione.



(a) Condizione contro prezzo



(b) Anno contro prezzo



(c) MMR contro prezzo

La Figura 3c mostra una forte correlazione tra le variabili disaminate: si nota che la retta descrittiva della relazione tra il Manheim Market Report (mmr) e il prezzo di vendita (sellingprice) ha un andamento molto simile alla bisettrice del primo e terzo quadrante del piano cartesiano. Matematicamente, ciò indica che i valori del MMR e del prezzo di vendita sono quasi coincidenti nella maggior parte dei casi, suggerendo una correlazione fortemente lineare (come già evinto dalla Figura 1). Da tali assunzioni, si evidenziano le problematiche discusse nelle sezioni precedenti circa il data leakage.

Pre-elaborazione dei dati. Nel caso specifico di Carval, si sono riscontrate diverse problematiche circa l'ambiguità dei dati di partenza. Un esempio concreto di "rumore" si trova nella variabilità delle denominazioni relative ad uno stesso marchio di auto. Ad esempio, nel dataset di partenza è presente la stessa marca registrata in diverse forme: 'mercedes', 'mercedes-b', 'mercedes-benz'. Queste variazioni, pur facendo riferimento alla stessa marca, potrebbero essere trattate come voci distinte se non correttamente gestite, causando confusione nelle analisi e introducendo inconsistenza nei dati. Per affrontare questo problema, si è adottata una tecnica di **uniformazione** delle denominazioni. In pratica, tutte le varianti di denominazione vengono consolidate in un'unica forma standard, definita attraverso un banale processo di mapping.

Un ulteriore tipo di rumore è stato riscontrato nelle feature *modello* e *allestimento*. In particolare, è emerso che alcuni campioni presentassero un valore di *allestimento* identico a quello di *modello*. Per ovviare a questa anomalia, si è deciso di sostituire, in tali osservazioni, il valore di *allestimento* con la stringa 'base', garantendo così una maggiore coerenza e uniformità nei dati.

Infine, è stato necessario gestire alcune ambiguità nei dati relativi alle colorazioni dei veicoli. Le variabili *color* e *interior* presentano in alcuni casi valori poco informativi e utili per l'analisi, come il simbolo "-". Per tali osservazioni si è deciso di impostarle come informazioni mancanti e gestirle poi nelle fasi successive della pipeline.

Raggruppamento degli allestimenti. Nel dataset di riferimento, la variabile *trim* rappresenta una versione specifica di un modello di veicolo, spesso indicante allestimenti, dotazioni aggiuntive o pacchetti opzionali. Questa informazione può essere utile nell'analisi predittiva, ma la sua elevata granularità e variabilità lessicale rendono difficile la sua integrazione diretta nei modelli di machine learning: in poche parole, il problema di fondo è che ci sono troppi valori diversi che magari indicano pressoché la stessa versione.

Per affrontare questo problema, si è scelto di raggruppare le diverse varianti in **categorie** semantiche più ampie: *base*, *special edition*, *sport*, *touring*, *luxury* e *other* (tutti i casi che non rientrano in nessuna delle categorie precedenti o sono valori mancanti/ambigui). L'uso di categorie più ampie evita problemi per alcune varianti poco frequenti e garantisce una maggiore interpretabilità.

Raggruppamento delle carrozzerie. Un'operazione simile a quella descritta precedentemente è stata realizzata per la caratteristica *body*. In particolare si è deciso di raggruppare i dati nelle seguenti categorie: *sedan*, *suv*, *hatchback*, *van*, *coupe*, *cabriolet*, *station wagon*, *pickup* e *other*.

Informazioni irrilevanti. Durante il processo di pre-elaborazione del dataset, sono state rimosse alcune variabili che, sebbene presenti nel dataset originale, sono risultate semanticamente irrilevanti per lo scopo predittivo e potrebbero addirittura introdurre un certo tasso di distorsione nei risultati.

Tali analisi, hanno causato la rimozione delle seguenti caratteristiche:

- *mmr*, per le motivazioni discusse nelle sezioni precedenti.
- *vin*, in quanto, trattandosi di un codice univoco, non consente al modello di stabilire dei pattern.
- *seller*, informazione ritenuta futile agli scopi della predizione.

- *saledate*, poiché si è deciso di non trattare esplicitamente dati di tipo temporali.
- *state*, perché i risultati finali dovranno trascendere i confini geografici di registrazione.

Ridenominazione caratteristiche. Per migliorare la comprensione nell'analisi dei dati, è stata effettuata una traduzione delle variabili da inglese ad italiano.

2.2 Data preparation

2.2.1 Pulizia dei dati

Nel contesto di machine learning, dati di bassa qualità possono portare a risultati fuorvianti e ad una generalizzazione scarsa, compromettendo le capacità del modello di apprendere le reali relazioni presenti tra le caratteristiche che si hanno a disposizione. Pertanto, un accurato processo di data cleaning risulta essere di vitale importanza.

Per pulizia dei dati si intende rendere i dati utilizzabili da un modello di ML ed in particolare capire in che modo sopravvivere alla mancanza o al rumore dei dati di partenza.

Indipendentemente dalle altre fasi di pulizia dei dati, è stato applicato un filtro minimo sul prezzo per cui sono stati considerati esclusivamente i campioni con un valore di prezzo maggiore o uguale a 500. Quest'operazione è stata eseguita per rimuovere i valori "non realistici" ed offrire una maggiore coerenza del dataset.

In aggiunta, è stata effettuata una selezione sulle marche presenti nel dataset. Sono stati esclusi i campioni appartenenti a marche con meno di 500 osservazioni. Le marche poco rappresentate non consentirebbero al modello di apprendere efficacemente degli schemi intrinseci e, peraltro, potrebbero compromettere la generalizzazione del modello.

Verifica dei valori nulli. In Figura 4 è mostrata la quantità di campioni con valori nulli per ciascuna caratteristica nella versione iniziale del dataset. La presenza di dati mancanti causa una visione non del tutto completa da parte del modello. Dunque, occorre stimare i valori dei dati mancanti sulla base di quelli disponibili (**data imputation**) oppure mitigare il problema andando a scartare basi di esperienza (rimozione di campioni o caratteristiche).

Imputazione dei dati. La sfida più impegnativa risiede nell'evitare l'introduzione di bias, un aspetto tutt'altro che banale. Per questo motivo, le tecniche classiche di imputazione – come l'utilizzo

Feature	Dtype
anno produzione	int64
marca	object
modello	object
allestimento	object
carrozzeria	object
trasmissione	object
condizione	float64
chilometraggio	float64
colorazione	object
colore interni	object
prezzo	int64

Tabella 2: Studio dei domini delle caratteristiche

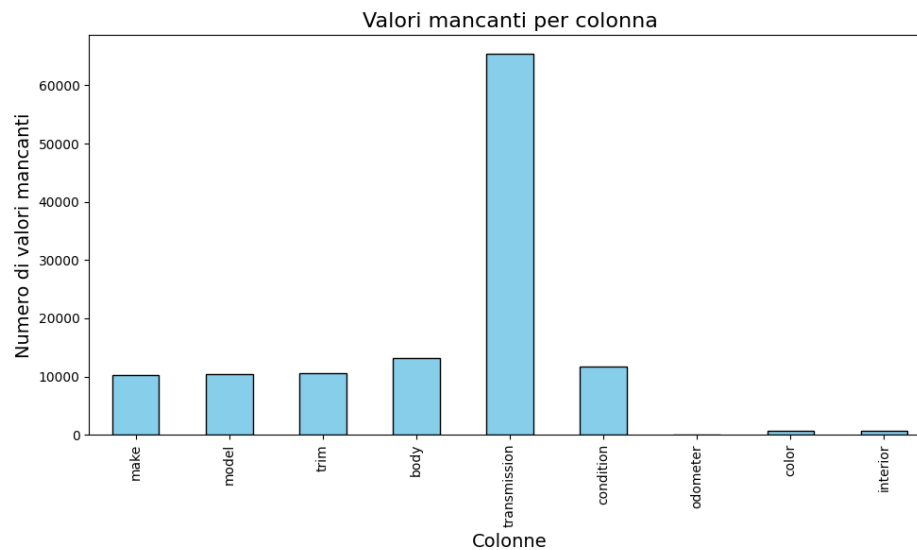


Figura 4: Studio dei valori nulli

della media o della mediana per le variabili numeriche e l'impiego della moda per quelle categoriche – vengono affiancate a valutazioni empiriche specifiche del contesto.

Per le variabili categoriali, la strategia adottata è quella di sostituire i valori mancanti con il valore più frequente (**moda**) all'interno di gruppi specifici di dati. Ad esempio, per ciascun modello di auto, viene calcolata la moda m della *carrozzeria* e della *trasmissione*, sostituendo i valori mancanti con m per quel determinato modello. Analogamente, per variabili come la *colorazione*, *colore interni*, la moda viene calcolata raggruppando per marca.

Per variabili numeriche come la *condizione* e il *chilometraggio*, vengono utilizzate le **medie** in gruppi basati sull'anno di produzione. Questo approccio consente di mantenere coerenza con il dataset, assumendo che i veicoli di età simile abbiano una condizione e un chilometraggio simili.

Verifica dei duplicati. Lo step di verifica dei dati duplicati permette di assicurare l'unicità e la coerenza dei dati, eliminando le ridondanze che possono portare a sovra-rappresentazioni di determinate osservazioni, inficiando, ineluttabilmente, la genuinità delle predizioni del modello. Fortunatamente, dopo le operazioni svolte in precedenza, il dataset non presenta campioni duplicati.

Conversioni dei tipi. Dalla Tabella 2 si nota che le feature chilometraggio e condizione sono inizialmente rappresentate come *float*. Tuttavia, è stato necessario eseguire conversioni in intero per garantire una maggiore coerenza e semplicità nell'interpretazione delle variabili.

Nella classificazione. Per affrontare il problema con un approccio di classificazione, il target originale, ovvero il prezzo, è stato discretizzato in fasce di prezzo predefinite:

1. 0 – 5000.
2. 5000 – 10000.
3. 10000 – 14000.
4. 14000 – 20000.
5. 20000+.

Questa suddivisione è stata scelta per differenziare adeguatamente le qualità, cercando al contempo di adattarsi alla distribuzione originale del dataset. Un'alternativa comune per la discretizzazione del prezzo sarebbe stata l'uso dei quantili, che garantiscono un bilanciamento tra le classi dividendo il

dataset in gruppi di dimensioni simili. Tuttavia, questa scelta avrebbe potuto portare a categorie meno intuitive per gli utenti finali, con fasce di prezzo non necessariamente coerenti.

2.2.2 Il problema delle scale

Codifica delle caratteristiche categoriche. Dalla Tabella 2 si può notare la presenza di una serie di caratteristiche categoriche. Tuttavia, le variabili categoriali non possono essere direttamente utilizzate in modelli di machine learning senza un adeguato processo di codifica.

Label encoding. La variabile 'trasmissione' ha solo due categorie, che sono mutuamente esclusive: "manuale" e "automatico". Per questo motivo, è possibile trasformare facilmente queste categorie in numeri, assegnando ad esempio 0 a "manuale" e 1 a "automatico".

Target Encoding con Smoothing. Per la gestione delle variabili categoriche ad alta cardinalità, si è scelto di utilizzare il *Target Encoding con Smoothing*, una tecnica che bilancia le informazioni locali e globali nella codifica delle categorie. Questa scelta è stata motivata dall'esigenza di mitigare gli svantaggi delle altre tecniche comunemente utilizzate:

- Il **One-Hot Encoding** può portare a un'elevata dimensionalità del dataset, rendendo il modello meno efficiente e aumentando il rischio di overfitting.
- Il **Target Encoding classico**, se applicato direttamente, rischia di sovrastimare il contributo delle categorie poco rappresentate, introducendo overfitting.

Il Target Encoding con Smoothing rappresenta un compromesso tra queste problematiche, stabilizzando la stima per le categorie meno frequenti. La formula utilizzata è la seguente:

$$encoded_value(c) = \frac{\sum_{i=1}^{N_c} y_i + m \cdot \bar{y}}{N_c + m}$$

dove:

- N_c è il numero di occorrenze della categoria c nel training set;
- $\sum_{i=1}^{N_c} y_i$ è la somma dei valori target per la categoria c ;
- \bar{y} è la media globale del target sull'intero dataset;
- m è il parametro di smoothing, che controlla l'influenza della media globale.

Scelta del valore di smoothing $m = 15$ Il parametro di smoothing determina il peso della media globale rispetto alla media locale della categoria:

- Se $N_c \ll m$, la media globale \bar{y} ha un'influenza predominante, evitando sovrastime per categorie rare.
- Se $N_c \gg m$, la stima è basata principalmente sulla media locale della categoria.

L'impostazione di $m = 15$ è stata scelta in quanto rappresenta un buon equilibrio tra stabilità e specificità, mitigando il rischio di overfitting per le categorie con poche occorrenze, senza perdere informazioni rilevanti per il target.

Applicazione in fase di inference Durante la fase di **inference**:

- Le variabili categoriche vengono trasformate utilizzando le statistiche apprese durante il training.
- Per le categorie non presenti nel training set viene assegnato il valore della media globale \bar{y} , garantendo coerenza.

Concludendo, l'adozione del Target Encoding con Smoothing offre un metodo efficace per gestire le variabili categoriche ad alta cardinalità, sfruttando l'informazione del target in modo controllato. La scelta di $m = 15$ consente di ridurre il rischio di overfitting per categorie poco rappresentate, mantenendo al contempo la capacità del modello di distinguere categorie informative.

Equiparare le scale. Le distribuzioni delle caratteristiche sono tendenzialmente diverse. Attributi con grosse differenze di scala provocano problemi per il machine learner perché potrebbe sottostimare/sovrastimare la rilevanza di una caratteristica. Lo **scaling** è un'operazione che consiste nel modificare i valori delle caratteristiche allo scopo di portarli tutti in un unico dominio. Lo scopo principale è evitare che le scale più ampie possano monopolizzare la fase di apprendimento.

In particolare si è deciso di adottare la ben nota tecnica della "*standardizzazione*" (**z-score normalization**), la quale rende la distribuzione normale con media 0 (\bar{x}) e deviazione standard 1 (σ).

$$x' = \frac{x - \bar{x}}{\sigma}$$

2.2.3 Ingegneria delle caratteristiche

Calcolo dell'età. Invece di utilizzare direttamente l'anno di produzione (ad esempio 1990), si è scelto di calcolare l'età del veicolo come la differenza tra l'anno corrente di riferimento (2015, in questo caso) e l'anno di produzione. Così facendo, la variabile "*anno produzione*" è stata sostituita da "età".

Eliminazione caratteristiche con bassa varianza. Una delle strategie di selezione delle caratteristiche è la rimozione delle feature che presentano una varianza estremamente bassa. La varianza di una caratteristica misura il grado di dispersione dei suoi valori: se una variabile assume valori quasi costanti su tutto il dataset, il suo contributo informativo per il modello è trascurabile.

Matematicamente, la varianza di una variabile X è definita come

$$Var(X) = \frac{1}{N} \cdot \sum_{i=1}^n (X_i - \mu)^2$$

dove N è il numero totale di osservazioni e μ è la media della variabile X .

Per ridurre la dimensionalità del dataset, se la varianza di una feature è inferiore ad una soglia prestabilita (0.1), essa viene rimossa dal dataset.

2.2.4 Bilanciamento delle classi

Dopo tutte le operazioni effettuate per garantire una certa soglia di qualità dei dati, per eseguire il task di classificazione, si è deciso di affrontare la problematica relativa al bilanciamento delle classi. In Figura 5 sono riportate due visualizzazioni rispetto al bilanciamento delle classi.

Nel machine learning supervisionato, il bilanciamento delle classi è un elemento fondamentale per garantire che il modello possa generalizzare efficacemente su nuovi dati. Quando una classe è significativamente più rappresentata rispetto alle altre, il modello rischia di imparare a classificare quasi esclusivamente la classe dominante, mostrando scarsa capacità di riconoscere le classi meno rappresentate.

L'analisi della distribuzione delle classi (Figura 5) ha evidenziato una forte disparità tra classe maggioritaria e classi minoritarie. Per mitigare questo squilibrio, si è adottata la tecnica di **oversampling** con SMOTE (Synthetic Minority Over-sampling Technique). SMOTE genera nuovi campioni sintetici per le classi meno rappresentate, sfruttando la varianza delle caratteristiche esistenti per interpolare dati realistici. Questo approccio permette di ridurre il rischio di overfitting, migliorando la capacità del modello di apprendere pattern significativi anche nelle classi meno frequenti.

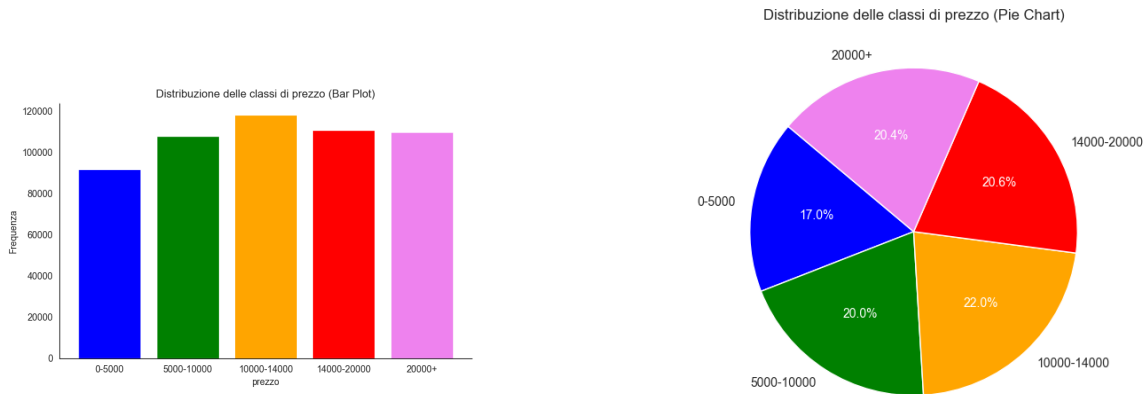


Figura 5: Bilanciamento delle classi

3 Sviluppo del modello

3.1 Selezione del modello

La scelta del modello di Machine Learning è una fase cruciale nello sviluppo di un sistema predittivo, soprattutto quando la relazione tra le variabili indipendenti e il target non segue un andamento lineare. In questo contesto, il RandomForest è stato selezionato per la sua capacità di modellare relazioni complesse e non lineari senza dover specificare esplicitamente una funzione matematica.

Perché? Uno dei motivi principali per cui il RandomForest è particolarmente adatto a questo problema è la sua flessibilità nella gestione della non linearità nei dati. Molti modelli di regressione tradizionali, come la regressione lineare o polinomiale, richiedono l'assunzione di una forma funzionale specifica tra le variabili indipendenti e il target. Tuttavia, nel caso della valutazione delle automobili, il valore finale è influenzato da un'ampia gamma di fattori (anno di produzione, chilometraggio, marca, modello, condizione del veicolo, ecc.), che interagiscono in modi complessi e difficilmente rappresentabili con una singola funzione matematica. Ad esempio, l'andamento del prezzo di un'auto non segue perfettamente una progressione/recessione rispetto al chilometraggio, in quanto ci sono altri fattori da tener conto (modello, anno di produzione, ecc.).

Il RandomForest, essendo basato su una collezione di alberi decisionali, è in grado di catturare queste relazioni senza bisogno di assumere una struttura predefinita.

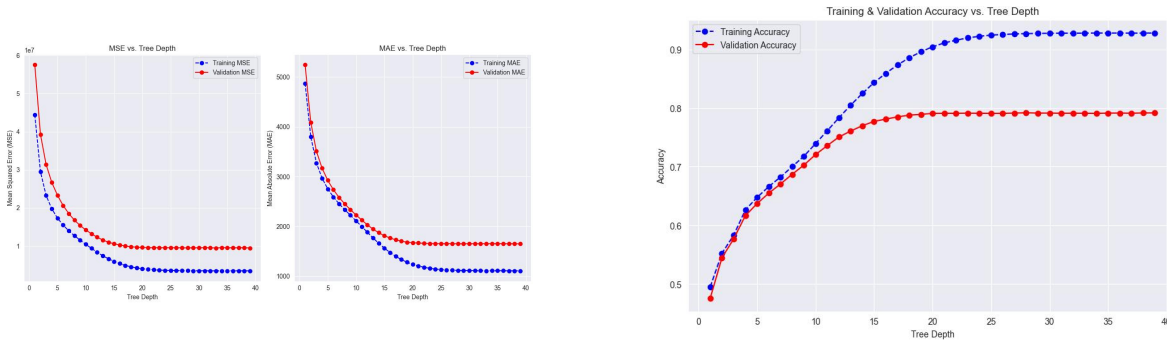


Figura 6: Validation loss e accuracy al variare di `max_depth`

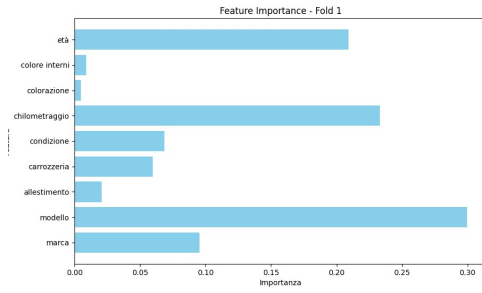
Funzionamento generale. Il RandomForest è un algoritmo di apprendimento supervisionato basato su una collezione di alberi decisionali. Si tratta di un metodo di **ensemble** learning, in cui molteplici alberi vengono addestrati su sottoinsiemi del dataset e le loro predizioni vengono poi aggregate per ottenere un risultato robusto e preciso.

Il modello funziona grossomodo secondo i seguenti passaggi:

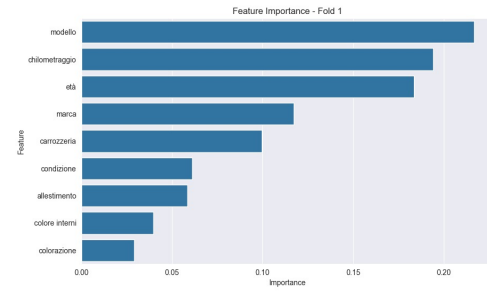
1. **campionamento bootstrap.** Vengono generati diversi sottoinsiemi del dataset tramite campionamento con sostituzione, garantendo una maggiore diversità nei dati utilizzati per addestrare ogni albero.
2. **Addestramento.** Ogni albero viene addestrato su un sottoinsieme diverso del dataset e utilizza una selezione casuale delle feature per effettuare gli split. Questo introduce un'ulteriore forma di diversificazione e previene l'overfitting.
3. **Aggregazione.** Il valore finale della previsione è ottenuto combinando i risultati di tutti gli alberi, rendendo il modello più stabile e meno sensibile alla varianza dei singoli alberi. Nel caso di regressione, il valore finale è ottenuto facendo la media delle predizioni di tutti gli alberi, mentre nel caso di classificazione, il risultato viene determinato tramite voto di maggioranza.

3.1.1 Fase di addestramento

Configurazione degli iperparametri. La configurazione degli iperparametri gioca un ruolo cruciale nell'addestramento del modello, influenzando prestazioni, generalizzazione ed efficienza computazionale. La scelta di tali parametri ha richiesto un compromesso tra accuratezza, complessità e rischio di overfitting.



(a) Regressione



(b) Classificazione

Figura 7: Importanza delle caratteristiche

Per la regressione.

- **Numero di alberi** ($n_estimators$): impostato a 500 per bilanciare accuratezza ed efficienza computazionale.
- **Feature per split** ($max_features$): impostato a $\sqrt{n_features}$ per ridurre l'overfitting e migliorare la generalizzazione.
- **Campioni minimi per split** ($min_samples_split$): valore 10 per evitare suddivisioni eccessive e garantire un buon bilanciamento.
- **Riproducibilità** ($random_state$): fissato a 20 per garantire consistenza tra esperimenti.
- **Campionamento con rimpiazzo** ($bootstrap$): impostato a `True` per migliorare la diversità degli alberi.
- **Criterio di suddivisione** ($criterion$): scelto `squared_error` per minimizzare la somma degli errori quadratici.
- **Profondità massima** (max_depth): fissata a 20 in base all'analisi della validation loss (Figura 6).
- **Campioni minimi per foglia** ($min_samples_leaf$): impostato a 4 per evitare eccessiva profondità degli alberi e ridurre l'overfitting.

Per la classificazione. Gli iperparametri principali sono stati mantenuti coerenti con la regressione, ad eccezione del criterio di suddivisione:

- **Criterio di suddivisione** (*criterion*): utilizzato **gini** per adattarsi alla natura discreta del problema.

3.2 Tecniche di valutazione

La valutazione di un modello di Machine Learning è una fase cruciale per garantirne l'affidabilità e la capacità di generalizzazione. Un errore comune è valutare il modello su un unico **train-test split**, il che può portare a risultati distorti a causa della specifica suddivisione scelta. Per mitigare questo problema, esistono diverse tecniche di validazione, tra cui la **k-fold cross validation**, che consente di ottenere una stima più robusta delle prestazioni del modello. La validazione incrociata è un metodo statistico che consiste nella ripetuta partizione e valutazione dell'insieme dei dati di partenza. Ogni elemento del dataset è assegnato ad un unico gruppo durante l'intera procedura di validazione, altrimenti il rischio è di mischiare dati di training con dati di test, andando a ricadere in un caso di **data leakage**. Mediante questa tecnica si fanno k validazioni del modello allo scopo di considerare ogni *fold* una volta sola come test set. Ad ogni iterazione vengono applicate le operazioni di **preprocessing** sui dati solamente sul training set.



Figura 8: Validazione incrociata

3.2.1 Metriche di validazione

Per la classificazione. Sono state utilizzate alcune delle metriche standard che forniscono indicazioni circa le prestazioni ottenute. Osservando la Figura 9 sono stati calcolati:

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN}$$

$$f1 - score = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

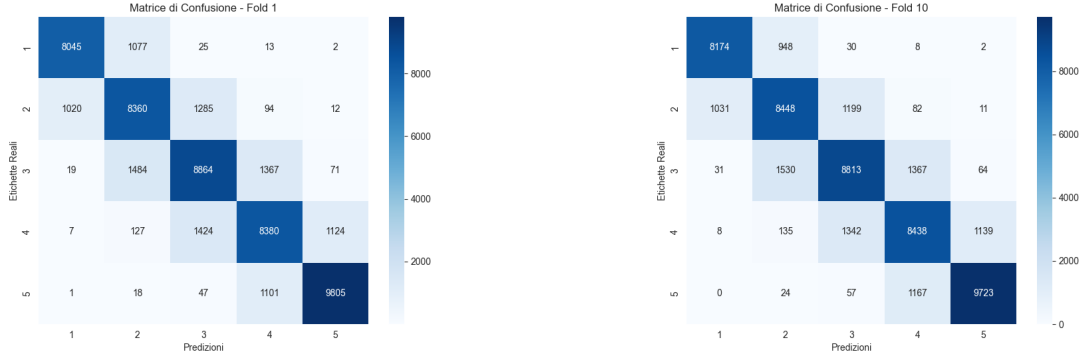


Figura 9: Matrici delle confusioni (primo e ultimo fold)

fold	accuracy	precision	recall	f1-score
1	0.8081	0.8082	0.8081	0.8081
2	0.8099	0.8097	0.8099	0.8098
3	0.8090	0.8092	0.8090	0.8090
4	0.8098	0.8099	0.8098	0.8098
5	0.8110	0.8112	0.8110	0.8110
6	0.8131	0.8131	0.8131	0.8130
7	0.8125	0.8124	0.8125	0.8124
8	0.8134	0.8134	0.8134	0.8134
9	0.8128	0.8129	0.8128	0.8127
10	0.8108	0.8107	0.8108	0.8107

Tabella 3: Valori metriche di validazione (classificazione)

In Tabella 3 sono visualizzati i risultati ottenuti fold per fold.

Per la regressione. Sono state utilizzate delle metriche standard allo scopo di comprendere l'adattabilità del modello finale rispetto al dataset.

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad MSE = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}$$

$$RMSE = \sqrt{MSE} \quad MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

In Tabella 4 sono visualizzati i risultati ottenuti fold per fold.

fold	MAE	MSE	RMSE	MAPE
1	1395.639	4920057.677	2218.120	0.158
2	1416.426	5258219.227	2293.081	0.157
3	1398.294	5913134.250	2431.694	0.158
4	1413.219	5048653.897	2246.921	0.158
5	1403.600	5193727.510	2278.975	0.156
6	1417.707	5383230.002	2320.179	0.156
7	1421.843	5431849.048	2330.633	0.158
8	1416.346	5170379.933	2273.847	0.158
9	1405.088	5042864.482	2245.632	0.160
10	1404.052	5240750.286	2289.269	0.157

Tabella 4: Valori metriche di validazione (regressione)

Dai risultati ottenuti e visualizzati, è possibile notare come le prestazioni del modello (sia per la classificazione che per la regressione) non variano considerevolmente tra una partizione e l'altra, indicando tutto sommato una buona capacità di generalizzazione sui dati.

4 Usabilità

L'usabilità di un sistema software, e in particolare di un modello di intelligenza artificiale, è un aspetto cruciale che ne determina il valore pratico. Un modello accurato, ma difficile da integrare, rischia di perdere gran parte del suo potenziale impatto.

4.1 Interfaccia grafica

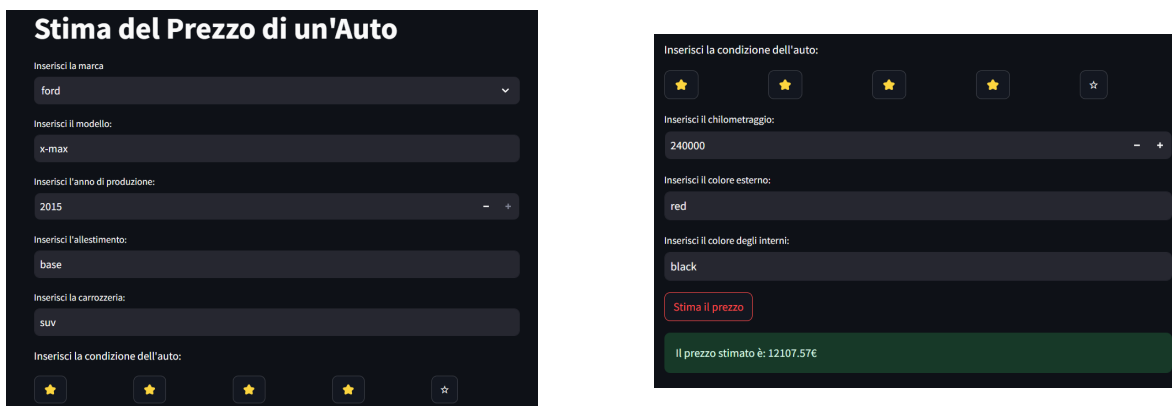


Figura 10: Interfaccia grafica

Per facilitare l'interazione con il modello addestrato ed, in generale, rendere, per quanto possibile, migliore l'interazione tra l'utente finale e la "macchina" è stata sviluppata un'interfaccia grafica utilizzando Streamlit, un framework open-source che permette di creare applicazioni web interattive in modo rapido e semplice.

L'interfaccia realizzata consente agli utenti di:

1. caricare i dati del veicolo.
2. Visualizzare il risultato in modo chiaro con un'indicazione del valore stimato.

Piccolo ma importantissimo DISCLAIMER. È fondamentale evidenziare che le valutazioni fornite non rappresentano una quotazione ufficiale o definitiva del valore di un'auto. Il modello è stato addestrato su un dataset specifico e può essere indubbiamente soggetto a errori o limitazioni. L'output deve essere interpretato come un supporto alla decisione e non come un valore assoluto o vincolante.

4.1.1 Perché Streamlit?

Streamlit è stato scelto per diversi motivi:

- **semplicità** d'uso e **velocità** di sviluppo. Permette di costruire interfacce grafiche con poche righe di codice, senza la necessità di sviluppare una web app complessa; inoltre, essendo pensato per applicazioni di data science, si integra perfettamente con modelli di machine learning.
- **Interattività** immediata. Offre componenti predefiniti per il caricamento di immagini e l'interazione con i risultati del modello.

5 Considerazioni finali

Il progetto Carval ha dimostrato come l'intelligenza artificiale possa essere uno strumento potente per la valutazione automatizzata del valore delle automobili. Attraverso l'analisi di un ampio dataset, è stato possibile sviluppare un sistema capace di offrire stime accurate e replicabili.

Tuttavia, come ogni sistema data-driven, Carval presenta alcune limitazioni e margini di miglioramento, che possono essere affrontati in sviluppi futuri per aumentarne l'efficienza e l'affidabilità.

5.1 Possibili miglioramenti

Per migliorare le prestazioni e l'affidabilità di Carval, si possono considerare diversi interventi, sia a livello di dati che di modello predittivo.

5.1.1 Nei dati

Espansione. L'integrazione di dati provenienti da più mercati permetterebbe di rendere il modello più generale e adattabile a contesti diversi. L'aggiornamento con dati più recenti migliorerebbe la capacità del modello di adattarsi alle fluttuazioni del mercato.

Granularità. L'inclusione di informazioni più dettagliate, come la cronologia delle manutenzioni, gli incidenti subiti o il numero di proprietari precedenti, potrebbe affinare ulteriormente le previsioni.

5.1.2 Nel modello

Sperimentazione. A causa delle tempistiche di realizzazione e consegna del progetto, non è stato possibile dedicare tempo sufficiente alla sperimentazione di diversi modelli di Machine Learning. Sebbene il RandomForest sia stato scelto per la sua robustezza e capacità di gestire dati non lineari, sarebbe stato interessante valutare altre alternative.

Ottimizzazione. L'uso di strategie avanzate di configurazione degli iperparametri, magari tramite algoritmi genetici, potrebbe migliorare l'accuratezza generale del modello.