



Chatbot

IV Progetto di Data Science

Implementazione di un chatbot di assistenza virtuale per analisi, confronto e suggerimento di Computer Portatili

Gabriel Piercecchi

Tosca Pierro



Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione



Tesina di:

Data Science

Esame per il corso tenuto dal Prof. Domenico Ursino,
durante l'anno accademico 2024-2025

9 CFU

Professori:
Domenico Ursino
Michele Marchetti

Redazione del documento a cura di:

- **Piercecchi Gabriel** (Matr. 1120541) - s1120541@studenti.univpm.it
- **Pierro Tosca** (Matr. 1120542) - s1120542@studenti.univpm.it

Anno Accademico 2024-2025

Anno II

Indice

1	Introduzione	4
1.1	Chatbot e Natural Language Processing (NLP)	4
1.2	Obiettivi e Ambito del Progetto	4
1.3	Repository Github	4
2	Il Framework RASA	5
2.1	Introduzione e Motivazioni	5
2.2	Componenti Fondamentali dell'Architettura	5
2.2.1	Rasa NLU	5
2.2.2	Rasa Core	5
2.3	Installazione e Configurazione	6
2.4	La Pipeline di Rasa: Dettagli Tecnici	6
2.5	Esempi di Personalizzazione e Integrazione	6
2.6	Conclusioni sull'Utilizzo di Rasa	7
3	Chatbot "LaptopGuruBot"	8
3.1	Il Dataset di Riferimento	8
3.2	Tipologie di Conversazione	9
3.2.1	Inizio Conversazione	9
3.2.2	Informazioni Generali sul Dataset "Laptop Selection Dataset"	10
3.2.3	Ricerca di Laptop Basata su Criteri Specifici	11
3.2.4	Confronto tra due modelli	14
3.2.5	Suggerimenti personalizzati in base alle esigenze dell'utente	17
3.2.6	Visualizzazione dell'immagine di un laptop	20
3.2.7	Richiesta di informazioni dettagliate su un modello	21
4	Connessione a Telegram	24
4.1	Testing	24
4.1.1	Inizio Conversazione	25
4.1.2	Informazioni Generali sul Dataset "Laptop Selection Dataset"	25
4.1.3	Ricerca di Laptop Basata su Criteri Specifici	27
4.1.4	Confronto tra due modelli	29
4.1.5	Suggerimenti personalizzati in base alle esigenze dell'utente	31
4.1.6	Visualizzazione dell'immagine di un laptop	32
4.1.7	Richiesta di informazioni dettagliate su un modello	32



1. Introduzione

Nel contesto attuale della digitalizzazione, l'interazione uomo-macchina si sta evolvendo in maniera significativa grazie all'impiego di soluzioni intelligenti. I chatbot, in particolare, rappresentano strumenti fondamentali per automatizzare e semplificare il servizio clienti, il supporto tecnico e le attività di consulenza. L'integrazione di queste applicazioni con tecniche avanzate di Natural Language Processing (NLP) consente di interpretare e rispondere alle richieste degli utenti in maniera efficiente e personalizzata.

1.1 Chatbot e Natural Language Processing (NLP)

Il cuore del sistema chatbot è l'elaborazione del linguaggio naturale, che permette al computer di comprendere e rispondere al linguaggio umano. Le tecniche NLP includono vari passaggi fondamentali:

- **Pre-elaborazione:** suddivisione del testo in token, normalizzazione e rimozione di stopwords.
- **Analisi sintattica:** estrazione delle relazioni grammaticali tra le parole per comprendere la struttura della frase.
- **Analisi semantica:** interpretazione del significato effettivo, gestendo ambiguità e polisemia.
- **Contestualizzazione:** mantenimento del contesto della conversazione per migliorare la rilevanza delle risposte.

Questi processi permettono di trasformare il testo in una rappresentazione strutturata che il sistema può utilizzare per generare risposte coerenti, anche in conversazioni complesse.

1.2 Obiettivi e Ambito del Progetto

Il progetto sviluppato si concentra sull'implementazione di un chatbot destinato a fornire assistenza nella scelta di laptop. Gli obiettivi specifici includono:

- **Ricerca:** permettere agli utenti di ricercare laptop in base a specifiche caratteristiche tecniche.
- **Confronto:** offrire la possibilità di confrontare diversi modelli in funzione di parametri come prestazioni, prezzo e autonomia.
- **Consiglio:** suggerire opzioni personalizzate che rispondano alle esigenze espresse dall'utente.

Il sistema si avvale di un dataset dedicato disponibile all'indirizzo <https://www.kaggle.com/datasets/rajugc/laptop-selection-dataset>.

1.3 Repository Github

La repository del progetto è disponibile al seguente link: <https://github.com/GabrielPiercecchi/DataScienceProject>.



2. Il Framework RASA

2.1 Introduzione e Motivazioni

La scelta di adottare **Rasa** come framework per lo sviluppo di chatbot nasce dalla necessità di disporre di una soluzione open source, flessibile e altamente personalizzabile. Rasa si distingue per la possibilità di mantenere una completa gestione dei dati in locale, garantendo elevati standard di privacy e sicurezza, aspetto cruciale in ambienti in cui la protezione dei dati sensibili è prioritaria. La capacità di integrare tecniche avanzate di Natural Language Processing (NLP) e machine learning permette al framework di interpretare il linguaggio naturale con elevata precisione, riconoscendo e classificando gli intenti dell'utente e gestendo la conversazione in maniera dinamica e contestuale.

2.2 Componenti Fondamentali dell'Architettura

L'architettura di Rasa si sviluppa in due macro-componenti principali, ognuna delle quali svolge ruoli specifici per la gestione del dialogo e l'interpretazione del linguaggio.

2.2.1 Rasa NLU

Rasa NLU è il modulo responsabile dell'analisi del testo in ingresso. Attraverso una pipeline configurabile, il sistema esegue una serie di operazioni fondamentali:

- **Tokenizzazione:** La frase in input viene suddivisa in token, ovvero unità lessicali di base, per permettere un'analisi più precisa.
- **Estrazione delle Feature:** Vengono generate rappresentazioni numeriche per ciascun token e per l'intera frase, utilizzando tecniche che spaziano da metodi tradizionali a modelli di deep learning, come quelli offerti da Hugging Face o SpaCy.
- **Classificazione degli Intents:** Il sistema assegna ad ogni input un intento, basato su un modello di classificazione addestrato con esempi predefiniti. Questo processo consente di individuare l'obiettivo della richiesta dell'utente.
- **Estrazione delle Entità:** Parallelamente alla classificazione degli intenti, Rasa NLU è in grado di identificare e estrarre entità specifiche (come date, nomi o prodotti) che arricchiscono il contesto del messaggio.

2.2.2 Rasa Core

Rasa Core si occupa invece della gestione del dialogo e della logica conversazionale. La componente di Core utilizza modelli di machine learning e regole definite in anticipo per tracciare lo stato della conversazione e predire l'azione successiva. Tra le sue funzionalità principali troviamo:

- **Gestione delle Conversazioni:** Rasa Core mantiene una cronologia dello scambio comunicativo, gestendo il contesto in modo da garantire coerenza nelle risposte anche in dialoghi complessi.

- **Previsione delle Azioni:** Utilizzando dati di training, il sistema prevede la prossima azione da eseguire, che può consistere nel rispondere direttamente all'utente oppure nell'attivare una funzione specifica, come una chiamata a un'API esterna.
- **Utilizzo delle Story e Rules:** Le storie (`stories.yml`) e le regole (`rules.yml`) permettono di definire in anticipo i flussi di dialogo, garantendo che il sistema possa gestire una varietà di scenari conversazionali.

2.3 Installazione e Configurazione

Rasa si integra nel mondo Python, e la sua installazione avviene attraverso semplici comandi da terminale. La configurazione iniziale prevede la creazione di una struttura di file standard, indispensabile per definire le funzionalità e il comportamento del chatbot. Tra i file principali troviamo:

- `domain.yml`: Contiene la definizione degli intenti, delle entità, delle azioni e delle risposte statiche. Questo file rappresenta il "dominio" dell'applicazione, definendo il contesto in cui opera il chatbot.
- `nlu.yml`: Include esempi di input degli utenti e serve ad addestrare il modello per il riconoscimento degli intenti e l'estrazione delle entità.
- `config.yml`: Specifica la pipeline di elaborazione utilizzata da Rasa NLU, includendo la configurazione dei tokenizers, featurizers, classificatori e altri componenti essenziali per il funzionamento del modello.
- `stories.yml e rules.yml`: Definiscono, rispettivamente, i flussi conversazionali complessi e le regole predefinite che guidano le risposte del chatbot in base a input specifici.

Questa organizzazione permette di personalizzare facilmente il comportamento del chatbot e di gestire in maniera modulare il processo di sviluppo e aggiornamento dell'applicativo.

2.4 La Pipeline di Rasa: Dettagli Tecnici

La pipeline in Rasa è la sequenza di operazioni che viene applicata a ogni input dell'utente. L'ordine di esecuzione delle componenti è fondamentale per ottenere una corretta interpretazione del linguaggio. Tra le componenti della pipeline rientrano:

- **Tokenizers:** Si occupano di suddividere il testo in unità significative, facilitando l'analisi sintattica e semantica.
- **Featurizers:** Generano vettori di caratteristiche dai token, utilizzando sia metodi tradizionali (come TF-IDF) che modelli pre-addestrati (come quelli basati su embedding).
- **Intent Classifiers:** Assegnano ad ogni input un intento, valutando la somiglianza tra il testo in ingresso e gli esempi di training presenti nel file `nlu.yml`.
- **Entity Extractors:** Identificano e categorizzano le entità presenti nel testo, arricchendo il contesto delle conversazioni.

Il corretto funzionamento della pipeline permette al chatbot di comprendere non solo il significato superficiale della frase, ma anche la sua contestualizzazione all'interno di una conversazione prolungata.

2.5 Esempi di Personalizzazione e Integrazione

Uno dei punti di forza di Rasa è la grande possibilità di personalizzazione offerta dagli sviluppatori. È possibile:

- **Integrare nuove azioni personalizzate:** Attraverso script Python definiti nel file `actions.py`, il sistema può eseguire operazioni complesse, come interfacciarsi con database esterni, effettuare chiamate API o eseguire calcoli specifici.
- **Modificare la pipeline:** Gli sviluppatori possono sperimentare diverse combinazioni di componenti nella pipeline per ottimizzare il riconoscimento degli intenti e l'estrazione delle entità, adattando il modello alle esigenze specifiche del progetto.
- **Definire politiche conversazionali avanzate:** Utilizzando regole e storie dettagliate, è possibile creare flussi di conversazione estremamente articolati, capaci di gestire casi d'uso complessi e rispondere in maniera dinamica alle variazioni nel dialogo.

2.6 Conclusioni sull'Utilizzo di Rasa

Rasa si rivela una soluzione estremamente versatile per la realizzazione di chatbot avanzati, in grado di sfruttare appieno le tecnologie di NLP e machine learning. La sua architettura modulare, insieme alla possibilità di personalizzare ogni aspetto del flusso conversazionale, lo rende uno strumento ideale per sviluppare applicazioni che richiedono un'interazione naturale e intelligente con l'utente. La comunità attiva e in continua crescita contribuisce inoltre al costante miglioramento del framework, offrendo supporto, aggiornamenti e nuove funzionalità che ne ampliano le possibilità di applicazione in diversi settori.



3. Chatbot "LaptopGuruBot"

La scelta di sviluppare un chatbot per la consulenza nella selezione di laptop nasce dall'esigenza concreta di supportare utenti con diversi livelli di competenza tecnologica nel processo di acquisto di un nuovo computer portatile. In un mercato sempre più ricco di opzioni e specifiche tecniche, individuare il dispositivo più adatto alle proprie necessità può risultare difficile senza una guida esperta.

Il sistema proposto si prefigge di offrire un'assistenza personalizzata in base alle abitudini e agli obiettivi dell'utente:

- **Consigli mirati:** aiutare utenti meno esperti a trovare il laptop più adatto al proprio profilo, che si tratti di gaming, studio o lavoro.
- **Confronto tra modelli:** consentire a utenti più esperti di valutare le differenze tra vari laptop in termini di prestazioni, prezzo e caratteristiche tecniche.
- **Ricerca diretta:** permettere di cercare rapidamente un modello specifico all'interno del database.
- **Visualizzazione grafica:** offrire la possibilità di visualizzare un'immagine del laptop richiesto, migliorando l'esperienza utente.

L'obiettivo è quello di colmare il divario tra la complessità delle informazioni tecniche e le reali esigenze dell'utente finale, offrendo un'interazione naturale e intuitiva. Il chatbot si configura come un assistente digitale in grado di guidare l'utente nella scelta del dispositivo più adeguato.

3.1 Il Dataset di Riferimento

Per rendere possibile l'interazione intelligente del chatbot con gli utenti, è stato utilizzato un dataset che contiene informazioni dettagliate su una vasta gamma di laptop. Il dataset proviene da Kaggle¹ e fornisce una panoramica completa del mercato dei laptop in India, con dati relativi a 1000 modelli di portatili disponibili sul sito di e-commerce Flipkart.

Le principali caratteristiche del dataset includono:

- **Tipo di processore:** informazioni sul tipo di CPU, come Intel, AMD, o altri.
- **RAM:** dimensione della memoria RAM del dispositivo.
- **Capacità di hard drive:** informazioni sul tipo di memorizzazione (SSD o HDD) e la sua capacità.
- **Dimensione e risoluzione dello schermo:** descrizione delle caratteristiche del display.
- **Sistema operativo:** sistema operativo preinstallato sul dispositivo.
- **Valutazioni e recensioni dei clienti:** valutazioni ottenute da parte degli utenti.
- **Prezzo:** prezzo del laptop, inizialmente espresso in INR e successivamente convertito in Euro.

Il dataset è stato sottoposto a operazioni di pulizia per garantirne la qualità (file CHATBOT\chatbot.ipynb):

- **Rimozione dei valori nulli:** eliminazione dei dati mancanti tramite la funzione dropna().

¹<https://www.kaggle.com/datasets/rajugc/laptop-selection-dataset>

- **Conversione della valuta INR a EUR:** i prezzi sono stati convertiti in Euro con un tasso di cambio di 0.011.
- **Rimozione dei duplicati:** eliminazione dei duplicati basati su `name`, `ram`, `processor`, e `storage`.
- **Pulizia della colonna "name":** rimozione delle informazioni non necessarie, come dettagli sul tipo di processore.

Queste modifiche hanno reso il dataset utile per il chatbot, che ora può fornire risposte precise e personalizzate, migliorando l'esperienza utente.

3.2 Tipologie di Conversazione

Per un'esperienza utente soddisfacente, è cruciale comprendere le diverse modalità di interazione del chatbot. In questa sezione, verranno descritte le varie forme di conversazione che il chatbot è in grado di gestire, al fine di garantire risposte rapide e precise.

3.2.1 Inizio Conversazione

L'inizio della conversazione è fondamentale per stabilire il tono e per presentare le funzionalità del chatbot. L'interazione parte dal messaggio dell'utente, che riceve un saluto iniziale seguito dalla spiegazione delle capacità del chatbot. Figura 3.1 mostra l'inizio della conversazione.

```

#   Bot
#   You
1  action_listen
2
3  utter_greet 1.00
    🌟 Benvenuto nel chatbot dei portatili!
    🔍 Puoi:
        1 Cercare dei laptop in base a criteri come uso e prezzo.
        2 Confrontare due laptop specificando, per ciascuno, nome e specifiche (processor, RAM, storage).
        3 ricevere un consiglio sul laptop sulla fascia qualità/prezzo migliore in base alle tue esigenze di prezzo ed uso.
        4 visualizzare l'immagine di laptop fornendo il loro nome.
    ? Per aiuto, scrivi "help".
    ciao
    intent: saluto 0.99

```

Figura 3.1: Inizio conversazione

NOTA: – I comandi utilizzati per avviare l'assistente virtuale e verificarne il corretto funzionamento sono stati i seguenti (eseguiti su due terminali distinti, uno dopo l'altro):

1. `rasa run actions`

Avvia il server delle azioni personalizzate, permettendo al chatbot di eseguire funzionalità specifiche definite nel file `actions.py`, come la visualizzazione di immagini o il confronto tra laptop.

2. `rasa interactive`

Avvia la modalità interattiva di Rasa, consentendo di testare e migliorare il comportamento del chatbot in tempo reale. In questa modalità, è possibile simulare conversazioni con l'assistente, correggere eventuali errori di interpretazione e aggiornare il modello di dialogo in modo iterativo.

3.2.2 Informazioni Generali sul Dataset "Laptop Selection Dataset"

Per chiarire come funziona il chatbot, è stato integrato un sistema di assistenza iniziale attivabile tramite il comando *help*, che fornisce esempi e spiegazioni delle principali funzionalità. Figura 3.2 mostra una sintesi² delle informazioni fornite dal comando.

```

4
help
intent: help 0.75

5 action_help 1.00
  □ Sono qui per aiutarti a trovare il laptop ideale!
  ↵ Puoi:
    * Cercare un laptop in base a criteri come uso e prezzo,
      * Confrontare due laptop specificando per ciascuno nome e specifiche.
      * Ricevere un consiglio su quale laptop ti conviene acquistare.
      * Visualizzare l'immagine di un laptop.
      * O chiedere i dettagli completi di un laptop.

  ☀ Esempi:
    * 'Cerco un laptop per lavoro sotto 800'
    * 'Confronta il [Dell Inspiron]laptop01 con il [HP Pavilion]laptop02' con [Intel Core i5](processor1) e [8GB DDR4](ram1) vs [Intel Core i7](processor2) e [16GB DDR4](ram2)'
    * 'Mi consigli un laptop per gaming con un budget di 1200'
    * 'Dammi i dettagli del [Dell XPS 15](laptop)'
    * 'Mostrami l'immagine del [HP Pavilion](laptop)'

  ☠ Dati disponibili per i laptop:
    * Esempi di utilizzo per il laptop: 'gaming', 'lavoro', 'programmazione', 'editing video', 'studenti' ed 'uso quotidiano'

  -
  Esempi di processori: 'Intel Core i5 Processor (11th Gen)', 'Intel Core i3 Processor (11th Gen)', 'Intel Core i5 Processor (10th Gen)', 'Intel Core i3 Processor (10th Gen)', 'AMD Athlon Dual Core Processor', 'Apple M1 Processor', 'Intel Celeron Dual Core Processor', 'AMD Ryzen 3 Dual Core Processor', 'Intel Core i5 Processor (12th Gen)', 'Intel Core i7 Processor (11th Gen)', 'AMD Ryzen 5 Hexa Core Processor', 'Intel Core i3 Processor (12th Gen)', 'AMD Ryzen 3 Octa Core Processor', 'Intel Core i9 Processor (12th Gen)', 'AMD Ryzen 9 Hexa Core Processor', 'Intel Core i7 Processor (12th Gen)', 'AMD Ryzen 9 Octa Core Processor', 'Intel Core i9 Processor (11th Gen)', 'AMD Ryzen 9 Octa Core Processor', 'Apple M1 Pro Processor', 'AMD Ryzen 9 Octa Core Processor', 'Apple M1 Max Processor', 'AMD Ryzen 9 Octa Core Processor', 'Intel Pentium Quad Core Processor', 'Apple M1 Ultra Processor', 'AMD Ryzen 9 Octa Core Processor', 'Intel Core i9 Processor (10th Gen)', 'Intel Celeron Quad Core Processor', 'AMD Ryzen 5 Dual Core Processor (5th Gen)', 'Intel Core i9 Processor (11th Gen)', 'AMD Ryzen 7 Quad Core Processor', 'Intel Core i5 Processor (8th Gen)', 'Intel Core i7 Processor (8th Gen)', 'AMD Ryzen 7 Hexa Core Processor', 'Intel Core i5 Processor (7th Gen)', 'Intel Core i9 Processor (7th Gen)', 'AMD Ryzen 9 Octa Core Processor (10th Gen)', 'Intel Core i3 Processor (7th Gen)', 'Intel Core i9 Processor (8th Gen)', 'AMD APU Dual Core A6 Processor', 'Intel Celeron Dual Core Processor (4th Gen)', 'Intel Core i5 Processor (5th Gen)', 'Intel Core i5 Processor (4th Gen)', 'Intel Core i7 Processor (9th Gen)', 'MediaTek MediaTek Kompanio 508 Processor'

  -
  Esempi di RAM: '16 GB DDR4 RAM', '8 GB DDR4 RAM', '4 GB DDR4 RAM', '16 GB LPDDR5 RAM', '16 GB DDR5 RAM', '4 GB LPDDR4 RAM', '8 GB LPDDR4X RAM', '32 GB LPDDR5 RAM', '4 GB LPDDR4 RAM', '16 GB LPDDR4 RAM', '8 GB DDR5 RAM', '8 GB Unified Memory RAM', '32 GB Unified Memory RAM', '16 GB Unified Memory RAM', '32 GB DDR5 RAM', '32 GB DDR4 RAM', '8 GB DDR3 RAM', '8 GB LPDDR3 RAM', '16 GB LPDDR3 RAM', '16 GB DDR3 RAM'

  -
  Esempi di storage: '512 GB SSD', '1 TB HDD|256 GB SSD', '256 GB SSD', '1 TB SSD', '2 TB SSD', '1 TB HDD|512 GB SSD', '1 TB HDD', '128 GB SSD', '256 GB HDD|256 GB SSD', '1 TB SSD|128 GB SSD', '256 GB NVMe', 'ready,sliver,linning Pro', 'Keyboard,Matrix Display (Extend),Cooler Boost 5,Hi-Res Audio,Haptic 3,14Hz Panel,Thin Bezel,RGB Gaming Keyboard,Speaker Tuning Engine,MSI Center', 'PCI-e Gen4 SSD|PS5|HDMI Matrix Display (Extend)|Cooler Boost 3|Thunderbolt 4|Finger Print Security|True Color 2.0|HDMI Res Audio|Nahimic 3|4-Sided Thin Bezel|MSI Center|Silky Smooth Touchpad|Military-Grade Durability', '2 TB HDD', '512 GB HDD|512 GB SSD', '256 GB HDD'

```

Figura 3.2: Informazioni essenziali tramite il comando *help*

Le modalità principali di interazione includono:

- **Ricerca di un laptop basata su criteri specifici.**

Esempio: "Cerco un laptop per lavoro sotto gli 800 euro"

- **Confronto tra due modelli.**

Esempio: "Confronta il Dell Inspiron con l'HP Pavilion"

- **Suggerimenti personalizzati in base alle esigenze dell'utente.**

Esempio: "Mi consigli un laptop per gaming con un budget di 1200 euro"

- **Visualizzazione dell'immagine di un laptop.**

Esempio: "Mostrami l'immagine dell'HP Pavilion"

- **Richiesta di informazioni dettagliate su un modello.**

Esempio: "Dammi i dettagli del Dell XPS 15"

Sono inoltre evidenziate le categorie d'uso comuni nel dataset: *gaming, lavoro, programmazione, editing video, studenti, uso quotidiano*.

²Data la presenza di termini simili in diverse colonne del dataset è stato necessario fornire degli esempi chiari ed esplicativi che hanno portato ad un lungo messaggio di testo

3.2.3 Ricerca di Laptop Basata su Criteri Specifici

Questa funzionalità consente di cercare un laptop in base a specifici criteri come il budget. Il chatbot guida l'utente tramite una serie di domande per raccogliere le informazioni necessarie, come nel caso di un utente che richiede un portatile da gaming con un budget superiore a 1200 euro.

I criteri da fornire in maniera obbligatoria sono presenti nella form `laptop_search_form` 3.1:

```

1 forms:
2   required_slots:
3     - price_range
4   ...

```

Listato 3.1: form per la ricerca di un laptop specifico

Una volta ottenuti i parametri, il sistema suggerisce laptop compatibili, ordinati per rilevanza. Se non vengono trovati risultati, il chatbot informa l'utente e può raccogliere ulteriori dettagli per affinare la ricerca.

Informazioni su un Laptop Specifico

Se l'utente ha in mente un laptop specifico, il chatbot può rispondere con una selezione mirata. Gli utenti possono fornire dettagli come il sistema operativo, la RAM e il tipo di storage per ottenere suggerimenti personalizzati. Il listato 3.2 mostra esempi di richieste di informazioni per un laptop.

```

1 - intent: cerca_laptop
2   examples: |
3     - Cerco un laptop per [lavoro](usage) con [64 bit Windows
4       11 Operating System](os) e [1 TB HDD](storage) a un prezzo
      sotto [800](price_range) EUR [inferiore](price_condition).
      - Cerco un laptop per [lavoro](usage) con [256 GB SSD](storage)
        a un prezzo sotto [800](price_range) EUR [
      inferiore](price_condition).

```

Listato 3.2: Esempi di richiesta di informazioni per un laptop

Se non si trovano laptop che soddisfano i criteri, il chatbot fornirà un messaggio di errore o suggerirà opzioni alternative. Se ci sono più opzioni, il sistema restituirà un elenco di modelli, chiedendo all'utente di affinare la ricerca.

Action "ActionSuggerisciLaptop"

L'azione `ActionSuggerisciLaptop` è responsabile per suggerire laptop in base ai criteri forniti dall'utente.

Il flusso di esecuzione dell'azione è il seguente:

- **Estrazione del messaggio dell'utente:** Il messaggio dell'utente, che contiene i criteri per la ricerca del laptop, viene recuperato tramite `tracker.latest_message.get("text", "")`. Il messaggio viene quindi convertito in minuscolo con `lower()` per garantire una ricerca case-insensitive.
- **Applicazione dei filtri:** La funzione `apply_common_filters` applica i filtri generali sui laptop in base ai criteri estratti dal messaggio dell'utente. Questi filtri possono includere specifiche come il tipo di utilizzo, il budget, e altre caratteristiche desiderate. Il risultato è un DataFrame `filtered_laptops` contenente i laptop che soddisfano i requisiti.

- **Ordinamento dei laptop:** I laptop filtrati vengono ordinati in base al prezzo in euro ("price(in EUR)") in ordine crescente, usando la funzione `sort_values(by="price(in EUR)", ascending=True)`. Questo assicura che i laptop più economici vengano mostrati per primi.
- **Selezione dei risultati:** Vengono selezionati i primi 3 laptop dal DataFrame ordinato, utilizzando `head(3)`. Se non vengono trovati risultati (`results.empty`), viene inviata una risposta di errore all'utente.
- **Formattazione della risposta:** Se ci sono risultati, il bot costruisce una risposta che elenca i laptop trovati. Per ogni laptop, vengono recuperati i dettagli (nome, caratteristiche, prezzo, ecc.) tramite la funzione `format_laptop_details(row)`, che formatta le informazioni in un testo leggibile. La risposta viene quindi inviata all'utente tramite `dispatcher.utter_message(text=risposta)`.
- **Gestione degli errori:** Se si verifica un errore durante l'esecuzione dell'azione (ad esempio, problemi con la ricerca nel dataset), l'errore viene loggato e una risposta di errore viene inviata all'utente.
- **Reset dei valori dei slot:** Infine, la funzione resetta i valori dei slot correnti dell'utente utilizzando `SlotSet(slot, None)` per `slot` in `tracker.current_slot_values().keys()` in modo che eventuali informazioni precedenti vengano rimosse.

Funzioni Ausiliarie

La funzione ausiliaria che supporta il processo di selezione è `apply_common_filters`, un componente fondamentale nell'azione di suggerimento dei laptop. Essa riceve come input un `DataFrame` contenente tutte le informazioni sui laptop (`df_input`) e un `Tracker` che contiene i dati degli slot relativi ai criteri di ricerca dell'utente. L'obiettivo della funzione è filtrare il `DataFrame` in base ai criteri definiti dall'utente attraverso gli slot, restituendo un nuovo `DataFrame` contenente solo i laptop che soddisfano le condizioni richieste.

Il flusso di esecuzione della funzione è il seguente:

- **Copia del DataFrame:** La funzione inizia creando una copia del `DataFrame` di input (`df_input`), per evitare di modificare i dati originali.
- **Filtraggio per il nome del laptop:** Se l'utente ha fornito un valore per il campo `laptop`, la funzione filtra i laptop contenenti il nome fornito nell'attributo `name` del `DataFrame`. La ricerca avviene in modo case-insensitive e senza l'uso di espressioni regolari.
- **Filtraggio per altri slot esplicativi:** Vengono quindi applicati filtri per altri slot come `ram`, `processor`, `storage` e `os`. Se l'utente ha specificato uno di questi valori, la funzione filtra i laptop che corrispondono ai criteri indicati.
- **Filtraggio basato sull'uso del laptop:** Se l'utente ha specificato uno slot `usage`, la funzione applica un insieme di filtri predefiniti in base all'uso del laptop. Ad esempio:
 - `gaming`: vengono selezionati laptop con processori potenti (Core i5, Core i7, Ryzen 7, Ryzen 9) e almeno 16 GB di RAM.
 - `lavoro`: si predilige il sistema operativo Windows e processori Core i5 o Ryzen 7.
 - `programmazione`: si selezionano laptop con processori equilibrati come Core i7, Core i9, Ryzen 7, Ryzen 9.
 - `editing video` o `rendering`: vengono selezionati laptop con processori di alta gamma (Core i9, Ryzen 9) e almeno 16 GB di RAM.
 - `studenti`: viene applicato un filtro di prezzo inferiore a 700 EUR e si selezionano laptop con Core i7 o Ryzen 7.
 - `uso quotidiano`: nessun filtro aggiuntivo.
- **Filtraggio per il prezzo:** Se sono stati forniti i valori di `price_condition` e `price_range`, la funzione applica un filtro sul prezzo dei laptop. Le possibili condizioni di prezzo

includono:

- **inferiore**: seleziona laptop con prezzo inferiore al valore specificato.
- **superiore**: seleziona laptop con prezzo superiore al valore specificato.
- **uguale**: seleziona laptop con prezzo uguale al valore specificato.
- **sotto, sopra, meno, più, minore, maggiore**: varianti delle condizioni di prezzo, applicando il filtro di prezzo in base al valore specificato.

Se è stato fornito solo il valore `price_range`, senza `price_condition`, la funzione selezionerà i laptop con prezzo inferiore o uguale a `price_range`.

- **Filtraggio per il rating**: Se l'utente ha specificato un valore per lo slot `rating_range`, la funzione filtra i laptop con un rating maggiore o uguale al valore indicato.
- **Filtraggio per il sistema operativo**: Se è stato fornito un valore per lo slot `os`, la funzione filtra i laptop in base al sistema operativo, selezionando quelli che contengono il valore specificato.
- **Filtraggio per il marchio**: Se è stato fornito un valore per lo slot `brand`, la funzione filtra i laptop in base al marchio, selezionando quelli che contengono il nome del marchio specificato.

Il risultato finale della funzione è un `DataFrame` filtrato, contenente solo i laptop che soddisfano i criteri specificati dall'utente. Se uno o più criteri non sono stati forniti o se la conversione di un valore (ad esempio il prezzo o il rating) fallisce, la funzione logga un errore e continua senza applicare il filtro relativo.

Esempio

La Figura 3.3 illustra un esempio di ricerca effettuata tramite l'assistente virtuale, in cui l'utente richiede un laptop per *gaming* con un budget superiore a 1200 €.

```

Chat History

#   Bot                                         You
1  action_listen
2                                         Voglio un laptop da [gaming](usage)
                                         intent: cerca_laptop 0.69
3  slot("usage": "gaming")
    laptop_search_form 1.00
    active_loop(name: "laptop_search_form")
    slot("usage": "gaming")
    slot("requested_slot": "price_range")
    slot("price_range": "1200")
    slot("price_condition": "sopra")
    slot("price_range": 1200.0)
    slot("requested_slot": null)
    active_loop(name: null)
    action_suggerisci_laptop 1.00
    Ecco alcuni laptop che potrebbero interessarti:
    - **ASUS TUF Gaming F17**:
        Processor: Intel Core i7
        Processor (12th Gen)
        RAM: 16 GB DDR5 RAM
        OS: 64 bit Windows 11 Operating System
        Storage: 1 TB SSD
        Display: 17.3 pollici
        Rating: 4.3 (basato su 59.0 valutazioni, 11.0 recensioni)
        Prezzo: 1209.89 EUR
        Immagine: https://rukminim1.flixcart.com/image/312/312/10r1jw0/computer/d/d/r/fa777rm-hx019w-gaming-laptop-asus-original-imacgcg8qkqgyccxy.jpeg?q=70
    - **HP OMEN**:
        Processor: AMD Ryzen 7 Octa Core
        Processor
        RAM: 16 GB DDR4 RAM
        OS: 64 bit Windows 11 Operating System
        Storage: 1 TB SSD
        Display: 16.1
        pollici
        Rating: 3.8 (basato su 5.0 valutazioni, 1.0 recensioni)
        Prezzo: 1210.0 EUR
        Immagine:
        https://rukminim1.flixcart.com/image/312/312/kzn17680/computer/1/i/z/-original-imaghzhqhdzy4e.jpeg?q=70
    - **MSI Crosshair 15**:
        Processor: Intel Core i7 Processor (12th Gen)
        RAM:
        16 GB DDR4 RAM
        OS: 64 bit Windows 11 Operating System
        Storage: 1 TB SSD
        Display: 15.6 pollici
        Rating:
        4.4 (basato su 40.0 valutazioni, 7.0 recensioni)
        Prezzo: 1220.89 EUR
        Immagine:
        https://rukminim1.flixcart.com/image/312/312/12ghjm08/computer/p/l/l/crosshair-15-b12uez-gaming-laptop-msi-original-imageymuhhs7dijf.jpeg?q=70

```

Figura 3.3: Esempio di ricerca di un laptop da gaming con un budget superiore a 1200 €

Come mostrato nelle immagini, l'assistente virtuale è in grado di interpretare correttamente le preferenze espresse dall'utente, ossia la necessità di un laptop per *gaming* e un budget superiore a 1200 €. In risposta, il sistema restituisce una lista di laptop che soddisfano i criteri specificati, fornendo all'utente opzioni pertinenti e ben selezionate per le sue esigenze.

3.2.4 Confronto tra due modelli

Proseguendo nella spiegazione del chatbot, è ora possibile trattare la sua seconda tipologia di conversazione: il confronto tra due modelli di laptop.

Fornendo le loro specifiche, infatti, l'assistente è in grado di determinare quale dei due sia superiore (*attraverso una logica arbitraria definita nel codice*).

Questa conversazione inizia con frasi definite nell'`intent confronto_laptop`, come mostrato nel Listato 3.3:

```

1 - intent: confronto_laptop
2   examples: |
3     - Confronta i due laptop
4     - Voglio confrontare due laptop
5     - Metti a confronto due laptop
6     - Confrontali
7     - Confronta il [Dell Inspiron](laptop1) con il [HP
8       Pavilion](laptop2)
9     - Confronta il [Dell Inspiron](laptop1) con il [HP G8](laptop2)
10    - Confronta il [ASUS Vivobook 15](laptop1) con il [HP G8](laptop2)
11    - Confronta il [APPLE 2021 Macbook Pro](laptop1) con il [HP G8](laptop2)
12    - Voglio comparare [MacBook Air](laptop1) con [ASUS
13      VivoBook](laptop2) per prestazioni e prezzo
14    - Quale laptop migliore tra [HP Spectre](laptop1) e [Asus ROG](laptop2)?
15    - Metti a confronto il [Lenovo IdeaPad](laptop1) e il [Acer Aspire](laptop2)
16    - Confronta il [Dell XPS 15](laptop1) con il [HP Envy](laptop2)

```

Listato 3.3: Esempi di inizio confronto fra laptop

A seguito del Listato 3.3, l'assistente virtuale provvederà a chiedere le specifiche dei laptop qualora mancassero (Listato 3.4):

```

1 - responses:
2 ...
3 # Domande per il form di confronto
4 utter_ask_laptop1:
5   - text: "Qual e' il nome del primo laptop?"
6 utter_ask_laptop2:
7   - text: "Qual e' il nome del secondo laptop?"
8 utter_ask_processor1:
9   - text: "Qual e' il processore del primo laptop?"
10  utter_ask_ram1:

```

```

11     - text: "Quanta RAM ha il primo laptop?"
12 utter_ask_storage1:
13     - text: "Che tipo di storage ha il primo laptop?"
14 utter_ask_processor2:
15     - text: "Qual e' il processore del secondo laptop?"
16 utter_ask_ram2:
17     - text: "Quanta RAM ha il secondo laptop?"
18 utter_ask_storage2:
19     - text: "Che tipo di storage ha il secondo laptop?"

```

Listato 3.4: Domande per le specifiche dei dispositivi

Qualora vi siano risultati ambigui o insufficienti, l'assistente chiede all'utente di specificare meglio le caratteristiche desiderate (processore, RAM o storage).

Tutto questo è gestito tramite l'utilizzo della form `laptop_compare_form` 3.5:

```

1 forms:
2 ...
3 laptop_compare_form:
4     required_slots:
5         - laptop1
6         - processor1
7         - ram1
8         - storage1
9         - laptop2
10        - processor2
11        - ram2
12        - storage2
13 ...

```

Listato 3.5: form per confronto dei dispositivi

Azione "ActionConfrontaLaptop"

L'azione `ActionConfrontaLaptop` consente all'assistente virtuale di effettuare un confronto dettagliato tra due dispositivi, sulla base dei nomi e delle specifiche tecniche fornite dall'utente.

L'azione viene invocata quando l'utente esprime l'intenzione di confrontare due laptop, indicando almeno i nomi dei modelli da confrontare.

Il funzionamento dell'azione può essere riassunto nei seguenti passaggi:

- **Recupero degli slot:** vengono estratti i valori degli slot forniti dall'utente tramite l'oggetto `tracker`. In particolare, si recuperano i nomi dei due laptop da confrontare (`laptop1`, `laptop2`) e, se disponibili, anche specifiche aggiuntive come `processor`, `ram` e `storage` per entrambi i dispositivi. Queste informazioni permettono di disambiguare i risultati nel caso in cui nel dataset siano presenti più modelli simili.
- **Filtraggio dei risultati:** utilizzando il dataset dei laptop (`df`), vengono cercati i modelli che corrispondono alle stringhe contenute negli slot. Il filtro avviene per nome del laptop tramite `str.contains()` (case-insensitive e senza espressioni regolari), e successivamente vengono applicati ulteriori filtri sulle specifiche, se fornite. Per lo storage, viene imposta anche una tolleranza sulla lunghezza della stringa per limitare le ambiguità.
- **Gestione dei casi ambigui o assenti:** se non viene trovato alcun risultato, oppure se i risultati per uno dei due laptop sono multipli, viene inviato un messaggio all'utente

che richiede una maggiore specificità nella descrizione. In tali casi, gli slot relativi al laptop ambiguo o assente vengono azzerati tramite `SlotSet`, per consentire all'utente di reinserire i dati.

- **Recupero dei dati:** se per entrambi i laptop viene identificato un solo risultato, questi vengono estratti tramite `iloc[0]`. Si assume quindi che il primo risultato del filtro sia il più rilevante, sulla base della qualità delle specifiche fornite.
- **Costruzione della risposta:** la risposta viene costruita combinando le seguenti due componenti:
 - Una sezione informativa per ciascun laptop, formattata tramite la funzione `format_laptop_details`, che restituisce un blocco testuale contenente le specifiche tecniche in modo leggibile e strutturato.
 - Una sezione analitica, generata dalla funzione `compare_laptop_performance`, che confronta i due dispositivi e fornisce un'interpretazione basata su parametri come CPU, RAM, archiviazione e rapporto qualità/prezzo.

L'intera risposta viene inviata all'utente attraverso `dispatcher.utter_message`.

- **Reset degli slot:** una volta completato il confronto, tutti gli slot attivi vengono resettati con `SlotSet(slot, None)` per ogni chiave dello stato corrente dell'utente. Questo consente di evitare conflitti o ambiguità in interazioni successive.
- **Logging:** durante l'esecuzione dell'azione, viene generato un log di debug che riporta i parametri dei dispositivi confrontati. Questo è utile per finalità di tracciamento, test e debugging del sistema.

Funzioni ausiliarie

Per supportare l'azione `ActionConfrontaLaptop`, sono state implementate tre funzioni ausiliarie che permettono di stimare e confrontare le prestazioni dei dispositivi selezionati.

- `get_processor_score(proc: str) -> int`: questa funzione riceve una stringa che descrive il processore e restituisce un punteggio numerico da 0 a 4 in base alla potenza stimata. I processori Intel e AMD vengono classificati rispettivamente in base alle famiglie i3, i5, i7, i9 e Ryzen 3, 5, 7, 9.
- `extract_ram_value(ram_str: str) -> int`: data una stringa contenente la descrizione della RAM (es. "8 GB DDR4"), estrae il primo numero intero trovato, che si assume rappresenti la quantità di RAM in gigabyte. In caso di assenza di numeri, restituisce 0.
- `compare_laptop_performance(row1: pd.Series, row2: pd.Series) -> str`: questa funzione confronta le prestazioni di due laptop, rappresentati come righe di un DataFrame. Utilizza il punteggio del processore e la quantità di RAM per calcolare uno score combinato (dove il peso del processore è maggiore, moltiplicato per 10). In base al punteggio finale, restituisce una frase che identifica quale dei due laptop è più performante, o se sono simili.

Queste funzioni consentono all'assistente virtuale di offrire un confronto qualitativo e sintetico tra i dispositivi, anche in assenza di benchmark dettagliati.

Esempio

La Figura 3.4 presenta un esempio di confronto tra due laptop: l'*ASUS TUF Gaming F15* e l'*MSI Bravo 15*.

In questo caso, dopo che l'utente ha fornito le caratteristiche principali dei due dispositivi (*Nome, Processore, RAM e Storage*), l'assistente virtuale è in grado di identificare i laptop nel dataset e di generare un confronto tra le loro specifiche. Nel dettaglio, l'assistente evidenzia che entrambi i computer hanno caratteristiche simili.

```

4
5  laptop_compare_form 1.00
active_loop{"name": "laptop_compare_form"}
slot{"requested_slot": "laptop1"}
Qual è il nome del primo laptop?
slot{"laptop1": "ASUS TUF Gaming F15"}
slot{"laptop1": "ASUS TUF Gaming F15"}
slot{"requested_slot": "processori1"}
Qual è il processore del primo laptop?
slot{"processori1": "Intel Core i5 Processor"}
slot{"processori1": "Intel Core i5 Processor"}
slot{"requested_slot": "ram1"}
Quanta RAM ha il primo laptop?
slot{"ram1": "8 GB DDR4 RAM"}
slot{"ram1": "8 GB DDR4 RAM"}
slot{"requested_slot": "storage1"}
Che tipo di storage ha il primo laptop?
slot{"storage1": "512 GB SSD"}
slot{"storage1": "512 GB SSD"}
slot{"requested_slot": "laptop2"}
Qual è il nome del secondo laptop?
slot{"laptop2": "MSI Bravo 15"}
slot{"laptop2": "MSI Bravo 15"}
slot{"requested_slot": "processori2"}
Qual è il processore del secondo laptop?
slot{"processori2": "AMD Ryzen 5 Hexa Core Processor"}
slot{"processori2": "AMD Ryzen 5 Hexa Core Processor"}
slot{"requested_slot": "ram2"}
Quanta RAM ha il secondo laptop?
slot{"ram2": "8 GB DDR4 RAM"}
slot{"ram2": "8 GB DDR4 RAM"}
slot{"requested_slot": "storage2"}
Che tipo di storage ha il secondo laptop?
slot{"storage2": "512 GB SSD"}
slot{"storage2": "512 GB SSD"}
slot{"requested_slot": null}
action{confronta_laptop:1.00}
Confronto dettagliato tra **ASUS TUF Gaming F15** e **MSI Bravo 15**:

### ASUS TUF Gaming F15
Processor: Intel Core i5 Processor (10th Gen)
RAM: 8 GB DDR4
OS: Windows 11 Operating System
Storage: 512 GB SSD
Display: 15.6 pollici
Rating: 4.4 (basato su 4733.0 valutazioni, 463.0 recensioni)
Prezzo: 549.89 EUR
Immagine:
https://rukminim1.flixcart.com/image/312/312/13rmzrk0/computer/s/z/r/-original-imageggdpgzmt.jpeg?q=70

### MSI Bravo 15
Processor: AMD Ryzen 5 Hexa Core Processor
RAM: 8 GB DDR4
RAM
OS: 64 bit Windows 11 Operating System
Storage: 512 GB SSD
Display: 15.6 pollici
Rating: 4.4 (basato su 2239.0 valutazioni, 299.0 recensioni)
Prezzo: 527.89 EUR
Immagine:
https://rukminim1.flixcart.com/image/312/312/xif0eq/computer/7/3/1/-original-imgha3pvvc2xuz2.jpeg?q=70
I due laptop sembrano offrire prestazioni simili.

```

Figura 3.4: Esempio di confronto fra *ASUS TUF Gaming F15* e *MSI Bravo 15*

Come già spiegato in precedenza, nel caso in cui l'assistente dovesse trovare più di due dispositivi che soddisfano i criteri del confronto, restituirà un messaggio personalizzato chiedendo ulteriori dettagli da parte dell'utente, per restringere i risultati e rendere il confronto più pertinente.

3.2.5 Suggerimenti personalizzati in base alle esigenze dell'utente

La conversazione successiva riguarda il suggerimento di un laptop in base alle esigenze dell'utente. A differenza della prima conversazione trattata, focalizzata sulla ricerca generica di un laptop, questa conversazione richiede non solo l'indicazione del budget, ma anche l'ambito di utilizzo desiderato. Le parole chiave per questi ambiti sono: *gaming*, *lavoro*, *programmazione*, *editing video*, *studenti* e *uso quotidiano*.

La form `laptop_recommend_form` 3.6 evidenzia proprio questi campi obbligatori:

```

1 forms:
2 ...
3 laptop_recommend_form:
4   required_slots:
5     - usage
6     - price_range

```

Listato 3.6: form per il consiglio su un laptop

L'assistente virtuale, inoltre, chiederà se l'utente desidera specificare anche il brand del computer. In caso di risposta negativa, il chatbot non applicherà alcun filtro sul marchio; al contrario, se l'utente fornisce un brand specifico, il filtro verrà applicato.

Per avviare questa conversazione è sufficiente scrivere frasi come quelle mostrate nel Listato 3.7:

```

1 - intent: consiglia_laptop
2   examples: |
3     - Mi consigli un laptop?
4     - Consigliami un portatile
5     - Consigliami un laptop
6     - Suggeriscimi un notebook
7     - Suggeriscimi un laptop
8     - Suggeriscimi un portatile per [gaming](usage)
9     - Mi consigli un laptop per [gaming](usage)?

```

Listato 3.7: Esempio di inizio conversazione per farsi consigliare un laptop dal chatbot

L'assistente, una volta riconosciuto l'intento, provvederà a richiedere le informazioni mancanti tramite le risposte definite nel Listato 3.8:

```

1 responses:
2 ...
3 # Risposte per il form di ricerca
4 utter_ask_usage:
5   - text: "Per che utilizzo cerchi il laptop?"
6 utter_ask_price_range:
7   - text: "Qual e' il tuo budget approssimativo?"
8 utter_ask_brand_option:
9   - text: "Vuoi specificare anche il brand del laptop? (si/no)"
10 utter_ask_brand:
11   - text: "Quale brand?"

```

Listato 3.8: Domande per gli argomenti necessari al chatbot per consigliare il laptop migliore

Action "ActionConsigliaLaptop"

L'azione ActionConsigliaLaptop è responsabile per consigliare un singolo laptop con elevata valutazione (rating), basandosi sui criteri forniti dall'utente.

Il flusso di esecuzione dell'azione è il seguente:

- **Estrazione del messaggio dell'utente:** Il messaggio viene recuperato tramite `tracker.latest_message.get("text", "")` e convertito in minuscolo usando `lower()` per garantire una gestione case-insensitive.
- **Applicazione dei filtri comuni:** Viene chiamata la funzione `apply_common_filters()`, che applica i filtri principali in base agli slot dell'utente (uso previsto, fascia di prezzo, brand, ecc.). Il risultato è un DataFrame `filtered_laptops` contenente i laptop compatibili con le richieste.
- **Filtraggio per rating elevato:** Vengono considerati solo i laptop con un rating compreso tra 4.5 e 5.0. A tal fine, la colonna `rating` viene convertita in formato numerico con `pd.to_numeric()` e successivamente filtrata.
- **Ordinamento per prezzo:** I laptop vengono ordinati per prezzo crescente in euro ("price(in EUR)"), utilizzando `sort_values(by="price(in EUR)", ascending=True)`.
- **Selezione del risultato:** Viene selezionato solo il primo laptop dal DataFrame ordinato tramite `head(1)`, in modo da consigliare il prodotto più economico tra quelli con rating

elevato. Se non vi sono risultati (`results.empty()`), viene inviata all'utente una risposta che notifica l'assenza di laptop corrispondenti.

- **Formattazione della risposta:** Se viene trovato un risultato, i dettagli del laptop selezionato vengono formattati tramite la funzione `format_laptop_details(row)`. Il messaggio viene quindi inviato all'utente tramite `dispatcher.utter_message()`.
- **Gestione degli errori:** Eventuali eccezioni vengono intercettate e loggiate. In caso di errore, l'assistente restituisce un messaggio all'utente che segnala l'impossibilità di completare l'operazione.
- **Reset dei valori degli slot:** Alla fine dell'azione, tutti gli slot vengono azzerati per evitare che influenzino le conversazioni successive. Questo viene fatto restituendo una lista vuota o utilizzando `SlotSet` su tutti gli slot attivi.

Funzioni Ausiliarie

Le funzioni ausiliarie utilizzate all'interno dell'azione `ActionConsigliaLaptop` sono le stesse impiegate da altre azioni precedentemente descritte, come ad esempio `ActionSuggerisciLaptop`. Esse sono state già esplicate in dettaglio nelle relative sezioni.

In particolare, l'azione fa uso delle seguenti funzioni:

- **apply_common_filters:** applica ai dataset i filtri comuni basati sui slot dell'utente (utilizzo, budget, brand, ecc.), restituendo un sottoinsieme di laptop che soddisfano i criteri specificati.
- **format_laptop_details:** formatta le informazioni di un laptop (nome, caratteristiche tecniche, prezzo, valutazione, ecc.) in una stringa ben strutturata da inviare come risposta all'utente.

Esempio

La Figura 3.5 mostra un esempio di laptop consigliato dall'assistente virtuale all'utente secondo le sue specifiche.

```

#   Bot                               You
1  action_listen
2
                                         Consigliami un laptop
                                         intent: consiglia_laptop 0.91
3  laptop_recommend_form 1.00
  active_loop{"name": "laptop_recommend_form"}
  slot{"requested_slot": "usage"}
  Per che utilizzo cerchi il laptop?
  slot{"usage": "gaming"}
  slot{"usage": "gaming"}
  slot{"requested_slot": "price_range"}
  Qual è il tuo budget approssimativo?
  slot{"price_range": "800"}
  slot{"price_condition": "sotto"}
  slot{"price_range": "800 e"}
  slot{"requested_slot": null}
  active_loop{"name": null}
  utter_ask_brand_option 0.70
  Vuoi specificare anche il brand del laptop? (si/no)
  action_listen 1.00
                                         si
                                         intent: affirm 1.00
4
5  utter_ask_brand 0.95
  Quale brand?
  action_listen 1.00
                                         Il brand è [ASUS](brand)
                                         intent: brand_chosen 1.00
6
7  slot{"brand": "ASUS"}
  action_consiglia_laptop 0.97
  Ti consiglio di dare un'occhiata a **ASUS Vivobook Pro 15**:
  Processor: AMD Ryzen 7 Octa Core Processor
  RAM:
  16 GB DDR4 RAM
  OS: 64 bit Windows 11 Operating System
  Storage: 512 GB SSD
  Display: 15.6 pollici
  Rating: 4.5
  (basato su 870.0 valutazioni, 105.0 recensioni)
  Prezzo: 659.89 EUR
  Immagine:
  https://rukminim1.flixcart.com/image/312/312/xif0q/computer/p/z/c/-original-imagr25cnfgpza.jpeg?q=70

```

Figura 3.5: Esempio di laptop consigliato da gaming, sotto gli 800€ e con brand ASUS

In questo esempio, l'utente ha richiesto un laptop adatto per il *gaming*, con un budget inferiore a *800 euro* e con il brand *ASUS*. L'assistente ha analizzato tali specifiche e ha

effettuato un filtraggio all'interno del dataset, selezionando il dispositivo con il punteggio di valutazione più alto (rating tra 4.5 e 5.0) e il prezzo più conveniente.

Come si può notare dalla figura, il chatbot restituisce il nome del laptop consigliato, accompagnato da una descrizione dettagliata contenente le caratteristiche tecniche, il prezzo e il punteggio di valutazione. Questo consente all'utente di avere una panoramica completa del prodotto suggerito, facilitando il processo decisionale.

3.2.6 Visualizzazione dell'immagine di un laptop

In questa sezione viene illustrata la possibilità di ottenere il link dell'immagine del laptop desiderato.

L'assistente virtuale, tramite frasi analoghe a quelle riportate nel Listato 3.9, consente di visualizzare l'immagine di tutti i computer che presentano un nome simile a quello fornito dall'utente memorizzati all'interno del dataset.

```

1 - intent: mostra_immagine
2   examples: |
3     - Mostrami l'immagine del [MacBook Air M1](laptop)
4     - Voglio vedere la foto del [Dell XPS 15](laptop)
5     - Fammi vedere l'immagine del [HP Pavilion](laptop)
6     - Mostrami la foto del [Lenovo ThinkPad](laptop)
7     - Puoi mostrarmi l'immagine del [ASUS ZenBook](laptop)?
8     - Vorrei vedere la foto del portatile [Acer Predator](laptop)
9     - Mostrami la foto del [Microsoft Surface Laptop](laptop)
10    - Mi serve vedere l'immagine del [Apple MacBook Pro](laptop)
11    - Fammi vedere l'immagine del computer [MSI Stealth](laptop)
12    - Mostrami la foto del [HP Envy](laptop)

```

Listato 3.9: Frasi per ottenere l'immagine del laptop desiderato

Nel caso in cui non venga trovato alcun dispositivo corrispondente, il sistema restituirà un messaggio dedicato che invita l'utente a correggere il nome inserito e riprovare: "*Non ho trovato il laptop specificato. Controlla il nome e riprova.*"

Action "ActionMostraImmagineLaptop"

L'azione ActionMostraImmagineLaptop è responsabile della visualizzazione del link immagine associato a un determinato laptop, in base al nome fornito dall'utente.

Il flusso di esecuzione dell'azione è il seguente:

- **Recupero dello slot laptop:** L'azione inizia recuperando il valore dello slot `laptop` tramite `tracker.get_slot("laptop")`. Se lo slot è vuoto, viene richiesto all'utente di specificare il nome del dispositivo desiderato.
- **Ricerca nel dataset:** Viene effettuata una ricerca all'interno del dataset `df`, utilizzando una corrispondenza parziale e case-insensitive sulla colonna `name`. I risultati vengono salvati nella variabile `matches`.
- **Gestione dei risultati:**
 - **Nessuna corrispondenza:** Se non viene trovato alcun dispositivo compatibile, viene inviato un messaggio di errore all'utente.

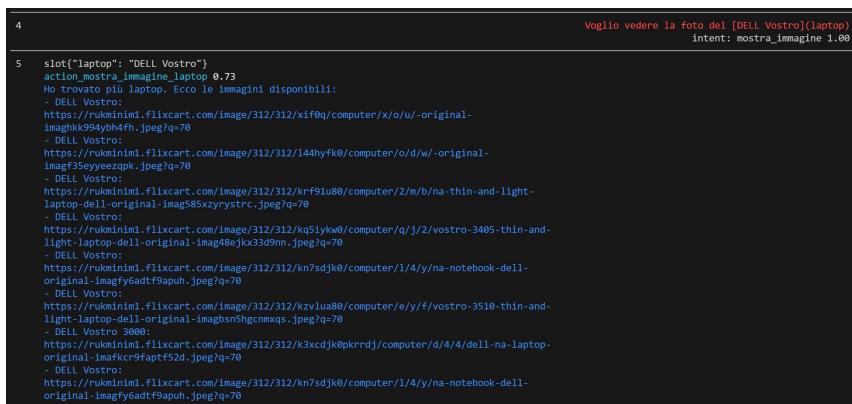
- **Singola corrispondenza:** Se viene trovato un solo laptop, il bot controlla se è presente un link immagine valido nella colonna `img_link`. Se presente, viene visualizzata l'immagine, altrimenti viene notificata la mancanza del link.
- **Più corrispondenze:** Se vengono trovati più dispositivi compatibili, l'assistente costruisce una lista di immagini (se disponibili), indicando per ogni entry il nome del laptop e il relativo link all'immagine. Se nessuna immagine è disponibile, viene notificato all'utente.
- **Invio del messaggio all'utente:** L'immagine (o le immagini) vengono inviate tramite `dispatcher.utter_message()`, utilizzando l'argomento `image=` in caso di immagine singola, o un messaggio testuale con i link in caso di immagini multiple.
- **Conclusione:** L'azione termina restituendo una lista vuota, poiché non è previsto alcun aggiornamento degli slot.

Funzioni Ausiliarie

L'azione `ActionMostraImmagineLaptop` non fa uso di funzioni ausiliarie esterne, in quanto tutta la logica necessaria per l'elaborazione è contenuta all'interno del metodo `run()` dell'azione stessa.

Esempio

Di seguito, in Figura 3.6, viene mostrato un esempio di conversazione in cui l'utente richiede l'immagine di un laptop specifico.



```

4                                     Voglio vedere la foto del [DELL Vostro](laptop)
                                         intent: mostra_immagine 1.00

5   slot("laptop": "DELL Vostro")
      action_mostra_immagine_laptop 0.73
      Ho trovato più laptop. Ecco le immagini disponibili:
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/xif0q/computer/x/o/u/-original-
        imgakk99aybh47h.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/144hyfk0/computer/o/d/w/-original-
        imgf3seyeezqk.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/krf91u80/computer/z/m/b/na-thin-and-light-
        laptop-dell-original-img585xzyystrc.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/kd5lykw0/computer/q/j/2/vostro-3405-thin-and-
        light-laptop-dell-original-img48ejx33d9rm.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/kn7sdjk0/computer/l/4/y/na-notebook-dell-
        original-imgfy6adt9apuh.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/kavluua80/computer/e/y/f/vostro-3510-thin-and-
        light-laptop-dell-original-imgbsn5hgcimxws.jpeg?q=70
      - DELL Vostro 3000:
        https://rukminim1.flixcart.com/image/312/312/k3xcdjk0pkrrdj/computer/d/4/4/dell-na-laptop-
        original-imafkr9faft520.jpeg?q=70
      - DELL Vostro:
        https://rukminim1.flixcart.com/image/312/312/kn7sdjk0/computer/l/4/y/na-notebook-dell-
        original-imgfy6adt9apuh.jpeg?q=70
  
```

Figura 3.6: Risposta del bot con i link delle immagini dei modelli contenenti "DELL Vostro" nel nome

Come illustrato, l'assistente identifica correttamente tutti i modelli compatibili con il nome fornito e restituisce i rispettivi link alle immagini, migliorando l'esperienza utente grazie alla componente visiva.

3.2.7 Richiesta di informazioni dettagliate su un modello

L'ultima tipologia di interazione possibile con l'assistente virtuale riguarda la richiesta di informazioni dettagliate su un laptop, dato il suo nome.

L'utente può anche fornire alcune specifiche del dispositivo desiderato per ottenere un filtraggio più preciso. Infatti, nel caso in cui esistano più modelli con nomi simili, il chatbot restituirà tutti i risultati compatibili, visualizzando per ciascuno le relative specifiche tecniche. In tal caso, l'assistente inviterà l'utente a fornire ulteriori dettagli per affinare la ricerca.

Il Listato 3.10 mostra alcune frasi esempio che permettono di avviare una conversazione con l'assistente virtuale al fine di ottenere le specifiche dettagliate di un laptop:

```

1 - intent: dettagli_laptop
2   examples: |
3     - Dammi i dettagli del [APPLE 2021 Macbook Pro](laptop)
4       con [512 GB SSD](storage) e [16 GB DDR4 RAM](ram).
5     - Vorrei vedere una descrizione completa del [Dell XPS 15]
6       (laptop).
7     - Mostrami tutte le informazioni tecniche del [ASUS TUF
8       Gaming](laptop).
9     - Raccontami le specifiche del [MSI Bravo 15](laptop).
10    - Quali sono le caratteristiche hardware del [Lenovo
11      ThinkPad X1 Carbon](laptop)?
12    - Voglio conoscere i dettagli tecnici del [HP Spectre x360
13      ](laptop).
14    - Puoi darmi tutte le informazioni sul [Razer Blade 15](laptop)?
15    - Dimmi tutto quello che sai sul [Acer Predator Helios 300
16      ](laptop).
17    - Quali componenti ha il [ASUS ROG Strix G15](laptop)?

```

Listato 3.10: Frasi per ottenere le specifiche del laptop desiderato

Action "ActionCaratteristicheLaptop"

L'azione `action_caratteristiche_laptop` gestisce le richieste dell'utente relative alla visualizzazione dei dettagli tecnici di un determinato laptop. Una volta ottenuto il nome del laptop tramite lo slot `laptop`, l'azione interroga il dataset alla ricerca di uno o più modelli che corrispondano al nome indicato.

Nel caso in cui siano disponibili ulteriori filtri (come processore, RAM o storage), questi vengono utilizzati per affinare la ricerca, sfruttando una corrispondenza parziale sui valori specificati.

Il comportamento dell'azione si articola in tre casi:

- Se non viene trovato alcun modello, viene restituito un messaggio informativo per indicare che non sono stati trovati risultati.
- Se vengono trovati più laptop compatibili, viene restituita una lista sintetica con i principali dettagli, invitando l'utente a specificare ulteriormente i criteri di ricerca.
- Se viene trovato un solo modello, vengono mostrati tutti i dettagli utilizzando la funzione `format_laptop_details()` per generare una descrizione formattata del dispositivo.

Al termine dell'azione, tutti gli slot correnti vengono azzerati, in modo da evitare che filtri precedenti interferiscano con conversazioni successive.

Funzioni Ausiliarie

L'azione `action_caratteristiche_laptop` fa uso della funzione ausiliaria `format_laptop_details()`, già descritta in precedenza. Questa funzione si occupa di formattare ordinatamente le caratteristiche tecniche del laptop selezionato, restituendo una stringa leggibile e ben strutturata, adatta alla visualizzazione da parte dell'utente.

Tutta la restante logica necessaria per il filtraggio e l'elaborazione dei dati è direttamente contenuta nel corpo del metodo `run()` dell'azione.

Esempio

In Figura 3.7 è riportato un esempio di conversazione in cui l'utente richiede i dettagli tecnici del modello **APPLE 2021 Macbook Pro con 512 GB SSD**.

Il bot recupera e mostra le specifiche tecniche del modello richiesto, tra cui processore, RAM, sistema operativo, display, valutazione, prezzo e un link all'immagine del prodotto.

```
16                                         Dammi i dettagli del [APPLE 2021 Macbook Pro](laptop) con [512 GB SSD](storage)
17 slot{"laptop": "APPLE 2021 Macbook Pro"}
    slot{"storage1": "512 GB SSD"}
    slot{"storage2": "512 GB SSD"}
    intent: dettagli_laptop 0.98
    action_caratteristiche_laptop 0.60
    Dettagli completi di **APPLE 2021 Macbook Pro**:
    Processor:
        Apple M1 Pro Processor
        RAM: 16 GB Unified Memory RAM
        OS: Mac
        OS Operating System
        Storage: 512 GB SSD
        Display: 16.2
        pollici
        Rating: 4.8 (basato su 119.0 valutazioni, 14.0
        recensioni)
        Prezzo: 1452.89 EUR
        Immagine: https://rukminim1.flixcart.com/image/312/312/kuyf8nk0/computer/g/z/q/mk1e3hn-a-laptop-apple-original-imag7yzmv57cvg3f.jpeg?q=70
```

Figura 3.7: Risposta del bot con i dettagli completi del laptop "APPLE 2021 Macbook Pro" con filtro "512 GB SSD"



4. Connessione a Telegram

Nella fase conclusiva del progetto, il chatbot è stato integrato con una piattaforma di messaggistica istantanea per migliorarne l'accessibilità e permetterne l'utilizzo tramite un'interfaccia familiare e intuitiva, soprattutto su dispositivi mobili.

È stata scelta la piattaforma **Telegram**, grazie alla facilità con cui consente l'aggiunta di bot attraverso la propria procedura guidata e alla compatibilità con **ngrok**, uno strumento utile per esporre localmente il servizio del chatbot e renderlo raggiungibile anche dall'esterno.

Nel dettaglio, nel file *credentials.yml* sono state configurate le credenziali necessarie per il collegamento tra Rasa e Telegram. Queste includono il token di accesso del bot, l'identificativo di verifica e urlar del webhook generato tramite ngrok, come mostrato di seguito:

```
1 telegram:  
2   access_token: "<your access token>"  
3   verify: "<your verify>"  
4   webhook_url: "<NGROK>/webhooks/telegram/webhook"  
5   parse_mode: "MarkdownV2"
```

Listato 4.1: Configurazione del canale Telegram in *credentials.yml*

Questa configurazione consente a Telegram di inviare i messaggi degli utenti al server Rasa in esecuzione locale, tramite l'URL temporaneo generato da ngrok. In questo modo, l'utente può interagire con il chatbot direttamente dall'app di Telegram, sfruttando un'esperienza d'uso più fluida e accessibile.

Per rendere operativo il sistema, sono stati utilizzati tre terminali distinti, avviati nell'ordine seguente:

1. **Terminale 1:** avvio del tunnel ngrok per esporre la porta 5005 verso l'esterno:

```
ngrok http 5005
```

2. **Terminale 2:** esecuzione del server delle action personalizzate:

```
rasa run actions
```

3. **Terminale 3:** avvio del server Rasa per la gestione del dialogo:

```
rasa run
```

Una volta completata questa configurazione, il bot risulta perfettamente funzionante su Telegram, pronto per ricevere e gestire le richieste degli utenti.

4.1 Testing

Di seguito sono proposti una sequenza di immagini raffiguranti diverse conversazioni con l'assistente virtuale attraverso Telegram (Versione applicazione per Windows 11).

4.1.1 Inizio Conversazione

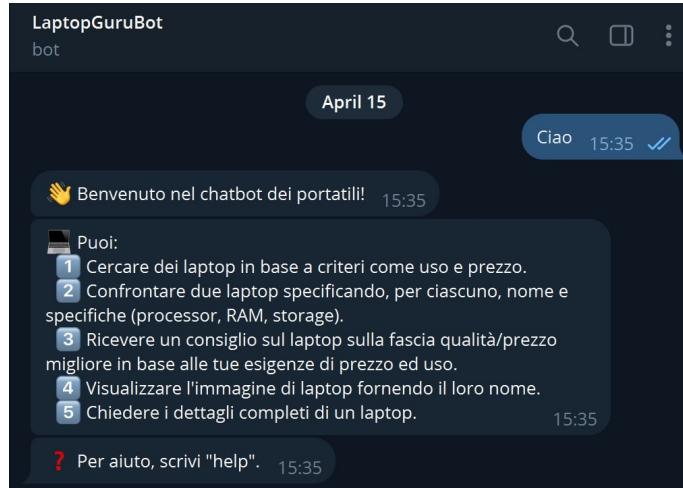


Figura 4.1: Inizio conversazione con un semplice "ciao"

4.1.2 Informazioni Generali sul Dataset "Laptop Selection Dataset"

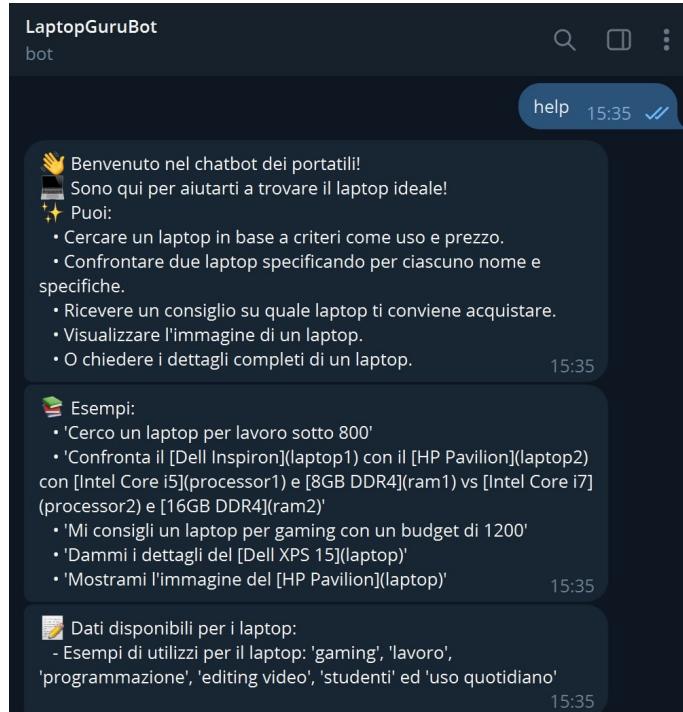


Figura 4.2: Richiesta di aiuto tramite il messaggio "help" - PARTE 1

- Esempi di processori: 'Intel Core i5 Processor (11th Gen)', 'Intel Core i3 Processor (11th Gen)', 'Intel Core i5 Processor (10th Gen)', 'Intel Core i3 Processor (10th Gen)', 'AMD Athlon Dual Core Processor', 'Apple M1 Processor', 'Intel Celeron Dual Core Processor', 'AMD Ryzen 3 Dual Core Processor', 'Intel Core i5 Processor (12th Gen)', 'Intel Core i7 Processor (11th Gen)', 'AMD Ryzen 5 Hexa Core Processor', 'Intel Core i3 Processor (12th Gen)', 'AMD Ryzen 3 Quad Core Processor', 'AMD Ryzen 7 Octa Core Processor', 'Qualcomm Snapdragon 7c Gen 2 Processor', 'Intel Core i7 Processor (12th Gen)', 'Intel Pentium Silver Processor', 'AMD Ryzen 5 Quad Core Processor', 'Intel Core i9 Processor (12th Gen)', 'AMD Dual Core Processor', 'Apple M2 Processor', 'AMD Ryzen 9 Octa Core Processor', 'Apple M1 Max Processor', 'Apple M1 Pro Processor', 'Intel Pentium Quad Core Processor', 'Intel Core i7 Processor (10th Gen)', 'AMD Ryzen 9 Octa Core Processor (5th Gen)', 'Intel Core i9 Processor (10th Gen)', 'Intel Celeron Quad Core Processor', 'AMD Ryzen 5 Dual Core Processor (5th Gen)', 'Intel Core i9 Processor (11th Gen)', 'AMD Ryzen 7 Quad Core Processor', 'Intel Core i5 Processor (7th Gen)', 'Intel Core i5 Processor (9th Gen)', 'AMD Ryzen 7 Hexa Core Processor', 'Intel Core i5 Processor (8th Gen)', 'Intel Core i7 Processor (8th Gen)', 'Intel Core i7 Processor (7th Gen)', 'AMD Ryzen 9 Octa Core Processor (10th Gen)', 'Intel Core i3 Processor (7th Gen)', 'Intel Core i9 Processor (8th Gen)', 'AMD APU Dual Core A6 Processor', 'Intel Celeron Dual Core Processor (4th Gen)', 'Intel Core i5 Processor (5th Gen)', 'Intel Core i5 Processor (4th Gen)', 'Intel Core i7 Processor (9th Gen)', 'MediaTek MediaTek Kompanio 500 Processor'

15:35

- Esempi di RAM: '16 GB DDR4 RAM', '8 GB DDR4 RAM', '4 GB DDR4 RAM', '16 GB LPDDR5 RAM', '16 GB DDR5 RAM', '4 GB LPDDR4X RAM', '8 GB LPDDR4X RAM', '32 GB LPDDR5 RAM', '4 GB LPDDR4 RAM', '16 GB LPDDR4X RAM', '8 GB DDR5 RAM', '8 GB Unified Memory RAM', '32 GB Unified Memory RAM', '16 GB Unified Memory RAM', '32 GB DDR5 RAM', '32 GB DDR4 RAM', '8 GB DDR3 RAM', '8 GB LPDDR3 RAM', '16 GB LPDDR3 RAM', '16 GB DDR3 RAM'

15:35

- Esempi di storage: '512 GB SSD', '1 TB HDD|256 GB SSD', '256 GB SSD', '1 TB SSD', '2 TB SSD', '1 TB HDD|512 GB SSD', '1 TB HDD', '128 GB SSD', '256 GB HDD|256 GB SSD', '1 TB HDD|128 GB SSD', 'PCI-e SSD (NVMe) ready', 'Silver Lining Print Keyboard', 'Matrix Display (Extend)', 'Cooler Boost 5', 'Hi-Res Audio', 'Nahimic 3, 144Hz Panel', 'Thin Bezel', 'RGB Gaming Keyboard', 'Speaker Tuning Engine', 'MSI Center', 'PCI-e Gen 4 SSD? SHIFT? Matrix Display (Extend)? Cooler Boost 3?', 'Thunderbolt 4? Finger Print Security? True Color 2.0? Hi-Res Audio? Nahimic 3? 4-Sided Thin bezel? MSI Center? Silky Smooth Touchpad?', 'Military-Grade Durability', '2 TB HDD', '512 GB HDD|512 GB SSD', '256 GB HDD'

15:35

Figura 4.3: Richiesta di aiuto tramite il messaggio "help" - PARTE 2

4.1.3 Ricerca di Laptop Basata su Criteri Specifici

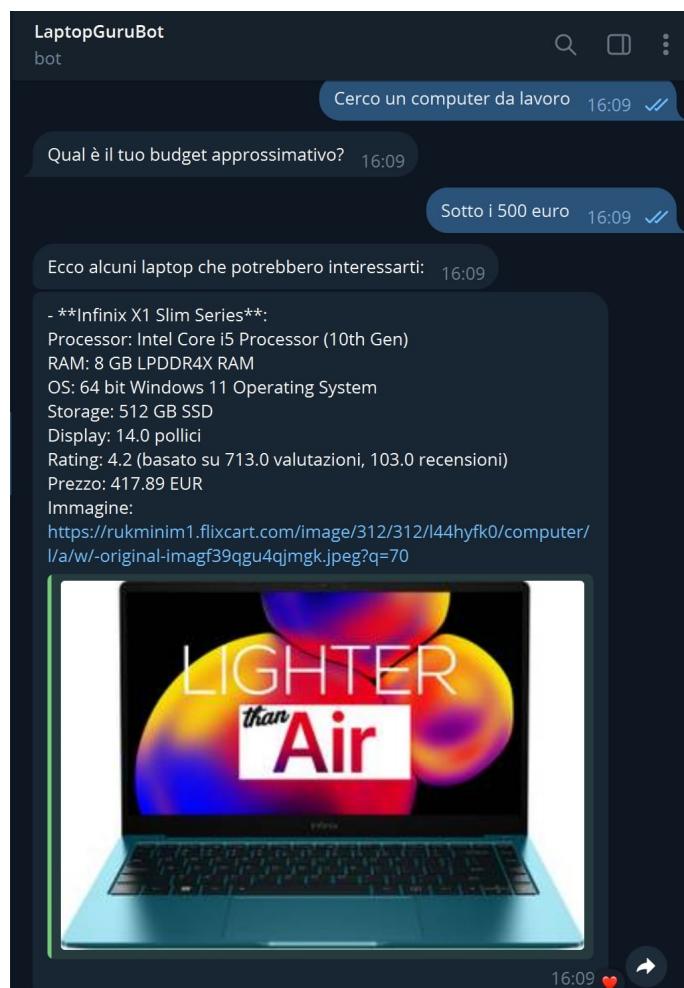


Figura 4.4: Ricerca di una lista di computer da lavoro sotto i 500 euro - PARTE 1

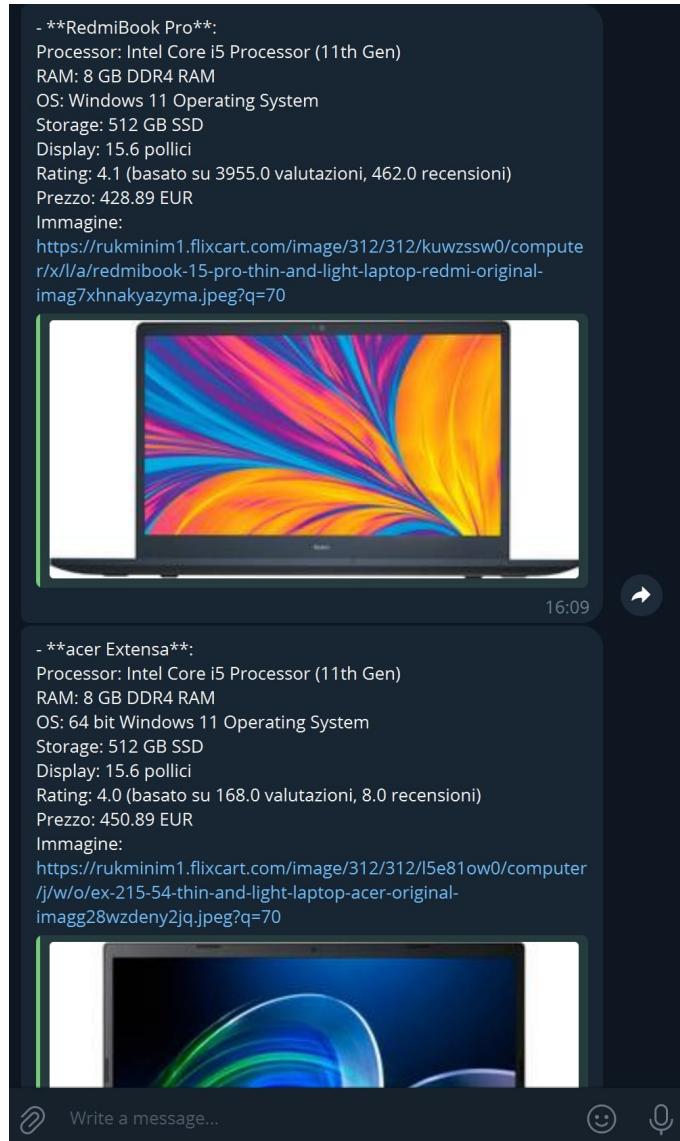


Figura 4.5: Ricerca di una lista di computer da lavoro sotto i 500 euro - PARTE 2

4.1.4 Confronto tra due modelli

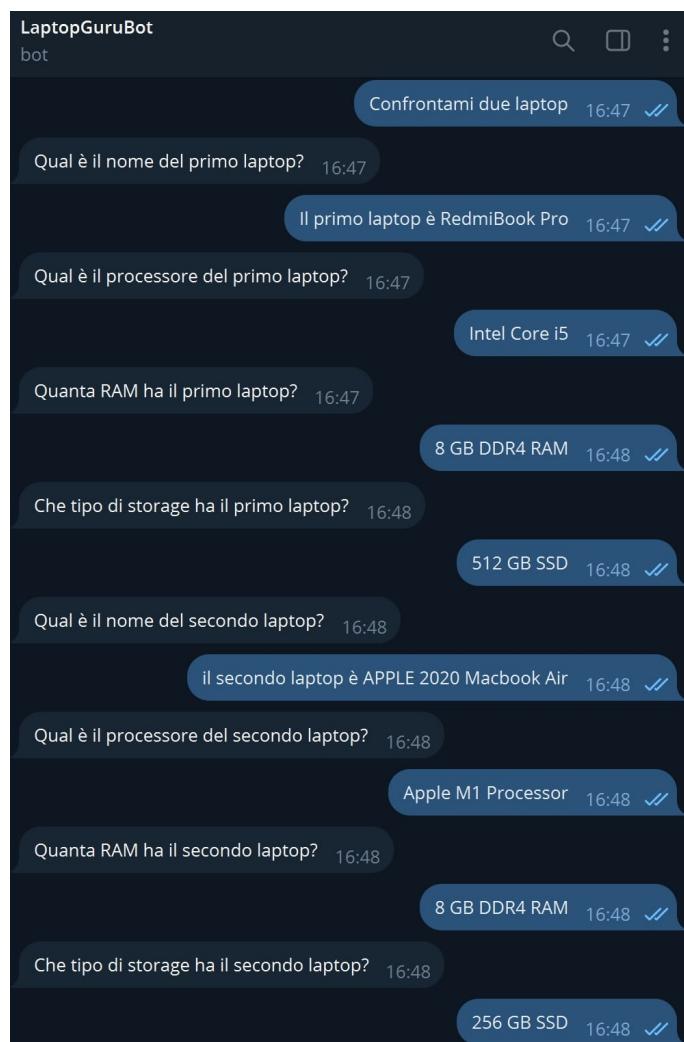


Figura 4.6: Confronto fra *RedmiBook Pro* e *APPLE 2020 Macbook Air* - PARTE 1



Figura 4.7: Confronto fra *RedmiBook Pro* e *APPLE 2020 Macbook Air* - PARTE 2



Figura 4.8: Confronto fra *RedmiBook Pro* e *APPLE 2020 Macbook Air* - PARTE 3

4.1.5 Suggerimenti personalizzati in base alle esigenze dell'utente



Figura 4.9: Computer consigliato in base alle specifiche dell'utente: per il *lavoro* e *sotto i 700 euro* - PARTE 1



Figura 4.10: Computer consigliato in base alle specifiche dell'utente: per il *lavoro* e *sotto i 700 euro* - PARTE 2

4.1.6 Visualizzazione dell'immagine di un laptop



Figura 4.11: Immagine del laptop *MSI Bravo 15*

4.1.7 Richiesta di informazioni dettagliate su un modello



Figura 4.12: Specifiche complete del laptop *MSI Bravo 15* come richiesto dall'utente