



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Informatica

A.a. 2023/2024

Sessione di laurea Dicembre 2024

**Sistema basato su IA per lo
Scraping, il Clustering e la
generazione di esperienze turistiche
personalizzate**

Relatori:

Luigi De Russis

Mirko Landolfo

Candidato:

Riccardo Renda

Ringraziamenti

Vorrei innanzitutto ringraziare la mia famiglia, per la loro vicinanza e il supporto costante durante tutti questi anni. Mia madre, mio padre e i miei fratelli Davide ed Edoardo, che con il loro amore e la loro presenza mi hanno fatto sentire a casa vicino a loro, anche quando non lo ero. Grazie per aver creduto in me nei momenti in cui io stesso faticavo a farlo.

Ringrazio i miei nonni, che per me sono stati una seconda famiglia, al pari della prima, sempre pronta ad accogliermi con affetto incondizionato. La loro generosità e il loro sostegno hanno rappresentato per me un faro, una guida nei momenti di incertezza.

Ai miei amici, quelli di una vita e quelli conosciuti lungo il mio cammino, voglio dire grazie per aver reso il viaggio più luminoso. Siete stati con me nei momenti di allegria e spensieratezza, ma anche in quelli più complessi, offrendomi vicinanza e comprensione. Insieme abbiamo creato ricordi che porterò con me per sempre. Ogni momento passato insieme ha contribuito a farmi crescere e mi ha accompagnato fino a questo traguardo.

Infine, voglio ringraziare tutte le persone incontrate lungo il mio percorso. Alcune sono passate velocemente, lasciando comunque un'impronta significativa, altre sono rimaste accanto a me fino a oggi, rendendo questo viaggio speciale in modi che non potrò mai dimenticare. Senza ognuna di esse tutto ciò non sarebbe stato possibile.

Indice

| | |
|---|-----|
| Elenco delle figure | VII |
| 1 Introduzione | 1 |
| 1.1 Contesto | 1 |
| 1.2 Obiettivo | 2 |
| 1.3 Struttura della tesi | 3 |
| 2 Background | 5 |
| 2.1 Scraping | 5 |
| 2.1.1 Arricchimento della conoscenza basata sul Web | 7 |
| 2.1.2 Tecnologie e librerie | 8 |
| 2.2 Preprocessing | 13 |
| 2.2.1 Teoria Preprocessing | 13 |
| 2.3 Clustering dei Dati e Tagging | 15 |
| 2.3.1 Teoria del Clustering | 16 |
| 2.3.2 Tecnologie per il Clustering | 17 |
| 2.3.3 Tagging | 20 |
| 2.3.4 Punteggio di unicità | 20 |
| 2.4 Copywriting generativo | 21 |
| 2.4.1 Tecnologie | 21 |
| 3 Progettazione del sistema | 24 |
| 3.1 Suddivisione pipeline | 24 |
| 3.2 Metodologia e flusso di esecuzione | 25 |
| 3.3 Portale web | 26 |
| 3.3.1 Human-AI Interaction (HAAI) | 26 |
| 3.3.2 Studio dell'interfaccia | 27 |
| 4 Scelte Tecnologiche | 32 |
| 4.1 Scraping | 34 |
| 4.1.1 Criteri iniziali | 34 |

| | | |
|----------|------------------------------------|-----------|
| 4.1.2 | Valutazione SQO-OSS | 35 |
| 4.1.3 | Scelta effettuata | 37 |
| 4.2 | Preprocessing | 38 |
| 4.2.1 | Valutazione delle scelte | 38 |
| 4.3 | Clustering e Tagging | 39 |
| 4.3.1 | Valutazione delle scelte | 39 |
| 4.4 | Copywriting generativo | 39 |
| 4.4.1 | Valutazione delle scelte | 40 |
| 4.4.2 | Valutazione SQO-OSS | 41 |
| 4.5 | Portale web | 43 |
| 4.5.1 | Valutazione scelte | 43 |
| 4.6 | Altre tecnologie | 44 |
| 4.6.1 | Python | 45 |
| 4.6.2 | JavaScript | 45 |
| 5 | Implementazione | 46 |
| 5.1 | Scraping | 47 |
| 5.1.1 | BeautifulSoup e Selenium | 47 |
| 5.1.2 | Apify | 51 |
| 5.2 | Preprocessing | 53 |
| 5.3 | Clustering e tagging | 55 |
| 5.3.1 | Clustering geografico | 56 |
| 5.3.2 | Tagging | 57 |
| 5.4 | Copywriting generativo | 58 |
| 5.5 | Portale web | 60 |
| 5.5.1 | Gestione Routes | 61 |
| 5.5.2 | Componenti Frontend | 62 |
| 5.5.3 | Backend | 67 |
| 6 | Risultati | 70 |
| 6.1 | Scraping | 70 |
| 6.2 | Preprocessing | 74 |
| 6.3 | Clustering e Tagging | 75 |
| 6.4 | Copywriting automatico | 77 |
| 6.5 | Valutazioni Finali | 80 |
| 6.5.1 | Pianificazione | 80 |
| 6.5.2 | Attività | 81 |
| 6.5.3 | Svolgimento | 82 |
| 6.5.4 | Risultati | 82 |
| 6.5.5 | Eventuali modifiche | 83 |

| | | |
|----------|--|-----------|
| 7 | Conclusioni | 86 |
| 7.1 | Sviluppi futuri | 87 |
| A | Script per la valutazione del prototipo | 88 |
| A.1 | Presentazione | 88 |
| A.2 | Istruzioni | 88 |
| A.3 | Attività | 89 |
| A.3.1 | Attività 1: Selezione degli elementi chiave per la ricerca . . . | 89 |
| A.3.2 | Attività 2: Filtri aggiuntivi e visualizzazione singole attività | 89 |
| A.3.3 | Attività 3: Generazione esperienza unica | 89 |
| A.3.4 | Attività 4: Rigenerazione e salvataggio attività | 90 |
| A.3.5 | Attività 5: Attività personali | 90 |
| A.4 | Debriefing | 90 |
| B | Questionario SUS | 92 |
| | Bibliografia | 94 |

Elenco delle figure

| | | |
|-----|--|----|
| 3.1 | Schermata di selezione tramite quattro selettori | 29 |
| 3.2 | Schermata di selezione tramite mappa | 29 |
| 3.3 | Schermata dei risultati ottenuti dal database, con possibilità di combinare o generare le attività | 30 |
| 3.4 | Schermate del risultato della generazione | 30 |
| 4.1 | Diagramma relativo al modello SQO-OSS utilizzato per la valutazione delle scelte tecnologiche per lo scraping | 37 |
| 4.2 | Diagramma relativo al modello SQO-OSS utilizzato per la valutazione delle scelte tecnologiche per il copywriting | 43 |
| 5.1 | Diagramma illustrante il funzionamento del sistema | 47 |
| 5.2 | Scraping tramite Apify, i parametri sono impostati in modo manuale | 52 |
| 5.3 | Scraping tramite Apify, i parametri sono impostati con un file JSON | 52 |
| 5.4 | Output 2D raffigurante la divisione in cluster geografici delle attività in esame | 56 |
| 5.5 | Schermata Home, in cui è possibile inserire i selettori | 62 |
| 5.6 | Schermata dell'elenco attività, in cui è possibile visionare le singole attività o filtrarle ulteriormente per poi generare l'esperienza unica . | 64 |
| 5.7 | Schermata della generazione, in cui viene mostrato il risultato finale | 65 |
| 5.8 | Schermata delle attività personali, in cui sono mostrate le attività generate che sono state salvate | 66 |
| 5.9 | Schermata della Mappa, in cui è possibile effettuare la ricerca tramite una mappa interattiva | 67 |

Capitolo 1

Introduzione

1.1 Contesto

La digitalizzazione nel concetto di Turismo 4.0 è un argomento cardine della missione 1 del PNRR [1]. Rispetto agli ultimi decenni e soprattutto post pandemia la fruibilità dei servizi turistici online è esponenzialmente aumentata.

Ciò ha fatto sì che i player internazionali offrissero digitalmente pacchetti standardizzati che però ha incrementato il turismo di massa creando fenomeni di over-tourism nelle principali mete turistiche e, di conseguenza, spopolando altre zone in cui si è invece verificato una situazione di under-tourism.

In tale contesto si inserisce il progetto di tesi. In particolare sarà parte del più ampio progetto TEIA (Turismo Esperienziale generato dall'Intelligenza Artificiale) la cui proposta cerca di contrastare questo fenomeno tramite la creazione di un'offerta turistica opposta, e quindi ad alta qualità, oltretutto in grado di valorizzare un territorio locale specifico in modo automatico e digitale. Difatti, attualmente ciò è possibile solamente attraverso una pratica di creatività manuale, frammentata e non digitalizzata dalla quale il turismo internazionale ha una scarsa visibilità online.

Gli operatori turistici con l'obiettivo di valorizzare offerte turistiche locali ed originali incontrano difficoltà economiche ed operative nell'individuazione di tali contesti. Per tale motivo il progetto TEIA punta ad inserire nel settore turistico una Key Enabling Technology quale l'Intelligenza Artificiale capace di risolvere la scalabilità della creazione di esperienze turistiche di qualità, creando un nuovo modello di gestione delle destinazioni basate sulla conoscenza già esistente online e attraverso l'analisi di big data. TEIA creerà un nuovo approccio digitale Data Driven per la creazione di esperienze autentiche in grado di massimizzare il ritorno per le comunità locali.

L'utilizzo dell'intelligenza artificiale cercherà quindi di creare automaticamente

una nuova offerta turistica promuovendo l'unicità di un territorio e della sua cultura, mettendo in evidenza le realtà medio piccole (MPMI) che lavorano per mantenere vive le tradizioni e le bellezze locali, facilitando i flussi turistici verso zone di valore anche meno conosciute (impattando positivamente sui problemi di over- e under-tourism), promuovendo l'importanza della relazione umana tra professionisti e viaggiatori.

Altro aspetto fondamentale è l'impatto ambientale: Il progetto TEIA rispetta il principio del DNSH (Do Not Significant Harm) e non arreca danno significativo all'ambiente.

In uno studio del 7 maggio 2018 pubblicato dalla rivista *Journal Nature Climate Change* [2], si stima che il turismo sarebbe responsabile dell'8% delle emissioni di anidride carbonica dell'economia globale. Tale cifra supera di tre volte la soglia di sostenibilità. Vengono considerati, oltre ai trasporti, anche l'energia utilizzata nel supportare le infrastrutture turistiche, tra le quali bevande, cibo e servizi. Il turismo può difatti essere trattato come un'industria, pesante ed in crescita esponenziale in tutto il mondo. Secondo il *World and Travel Tourism Council (WTTC)*, costituisce il 10,1% del PIL mondiale accrescendo sempre più la propria importanza, essendo poi inquinante ed impattante.

Il progetto TEIA si basa sull'introdurre una nuova tecnologia abilitante, quale l'AI, con lo scopo di creare una nuova offerta turistica basata su modelli di gestione delle destinazioni a loro volta basate sulla conoscenza, la sostenibilità e la massimizzazione dell'impatto sociale sulle comunità locali.

Ciò si traduce in un ulteriore obiettivo, quello di ridurre l'impatto sociale, ambientale e di non recare alcun danno all'ambiente ma bensì di ridurre l'impatto complessivo, grazie anche al supporto scientifico dell'Università degli Studi dell'Insubria e i relativi professori esperti in Economia del Turismo Sostenibile. Il progetto adotterà una logica di turismo responsabile in cui è valutato principalmente l'impatto sulla popolazione locale e sullo sviluppo economico e sociale, unitamente al concetto di turismo sostenibile, il quale si occupa di valutare anche l'impatto del turismo sull'ambiente e il relativo inquinamento.

1.2 Obiettivo

L'obiettivo principale del progetto di tesi è quindi quello di creare un modello di AI che, una volta ricevuti dati adeguati a una o più determinate regioni, possa generare un'attività specifica con la relativa descrizione.

In particolare, si pensa anche ad un'interfaccia con la quale l'utente possa interagire inserendo delle keyword: dove, quando, tag primario, tag secondario. Se inseriti tutti e quattro si avrà un elenco di attività che rispecchiano tale input. Nel

caso ne venissero forniti una parte, il modello completerà prima la scheda di ricerca e secondariamente si potrà avviare la ricerca.

Ad ogni modo l'elenco di attività sarà ordinato in base ad un parametro di "unicità", valore che indicherà quanto quest'ultima sia unica e quindi più di valore per contrastare l'over-tourism. L'utente potrà poi selezionare una o più attività che potranno quindi essere combinate in modo da formarne una unica.

Considerato tale obiettivo, si è potuto definire i compiti cui dovrà sopperire il modello:

- **Raccolta dati**, sul quale costruire un dataset che rappresenti la conoscenza del modello.
- **Analisi dati**, trovare pattern che leghino i dati ottenuti.
- **Generazione attività**, in base alle analisi effettuate e dato un input di richiesta specifico, si ottenga un'esperienza e la sua descrizione.

Sarà quindi necessario capire come potranno essere risolte tali tasks. La prima fase sarà quella di analizzare per ogni segmento le tecniche e tecnologie a disposizione. Dopo aver condotto tale studio avendo evidenziato pro e contro, verranno scelte delle soluzioni per ogni compito adattandoli inoltre per poter operare in sequenza formando una vera e propria pipeline.

Principalmente, si cercherà di adottare soluzioni Open Source in modo tale da ottenere flessibilità e sicurezza ma anche rispettando la necessità del progetto di avere costi ridotti ed essere indipendenti da terze parti.

1.3 Struttura della tesi

La tesi seguirà in ordine cronologico lo sviluppo del modello stesso. Dividendo tra studi fatti per definire l'architettura con annessa descrizione teorica, per poi mostrare l'implementazione finale ed eventuali risultati. Specificatamente sull'architettura, partendo dalla creazione della conoscenza su internet tramite scraping, si procederà con la vera e propria parte di analisi dei dati tramite clustering ed attribuzione di un qualche score alle attività. Infine, vi sarà un modulo generativo che produrrà un output.

Vi sarà quindi un capitolo di studio del dominio ed analisi cui seguirà il capitolo dove i moduli saranno implementati, fatta eccezione per l'ultimo dove verranno tratte le conclusioni. Più in dettaglio:

- Capitolo 2 "Background": studio del dominio e background teorico dei vari moduli che motivino ogni scelta architettonica.

- Capitolo 3 “Progettazione del sistema”: capitolo nel quale viene illustrato l’approccio utilizzato nella progettazione del sistema.
- Capitolo 4 “Scelte Tecnologiche”: capitolo in cui si motiveranno le scelte effettuate per ogni modulo.
- Capitolo 5 “Implementazione”: tecnologie utilizzate per l’implementazione dei suddetti moduli.
- Capitolo 6 “Risultati”: capitolo in cui saranno illustrati i risultati per ogni modulo.
- Capitolo 7 “Conclusioni”: ultime considerazioni, punti di forza e limitazioni, sviluppi futuri.

Capitolo 2

Background

Riprendendo il contesto in cui viene inserito TEIA, il punto di partenza è la creazione di esperienze turistiche innovative e che non riguardino le località principali. In virtù di ciò si vorrebbe favorire l'under-tourism sull'over-tourism senza però avere grande impatto sull'ambiente.

La creazione di un'esperienza turistica è finora un compito prettamente manuale; nel caso in cui se ne volessero offrire in una regione diversa da quella di origine del fornitore, bisogna prima creare una conoscenza su quel luogo. Solo dopo è possibile quindi valutare le opportunità ed agire di conseguenza.

Un obiettivo del progetto TEIA è quello di automatizzare questo procedimento, la raccolta delle informazioni quindi non dovrebbe essere più svolta da un operatore umano andando a velocizzare tale operazione. Sarà inoltre richiesto che tale attività sia più unica possibile oltrechè adatta ad un determinato target di persone.

Per poter sviluppare un sistema adatto a soddisfare gli obiettivi definiti, è stata necessaria una fase di studio delle tecniche disponibili e delle relative tecnologie. Il sistema sarà quindi suddiviso in moduli in modo tale che ognuno sopperisca ad ogni compito.

In particolare, per ogni modulo si è compreso quali obiettivi dovesse raggiungere e di conseguenza cosa era necessario utilizzare. Chiaramente si sono confrontate più soluzioni analizzando pro e contro di ognuna in modo da selezionare la più adatta in base al contesto.

2.1 Scraping

Il primo modulo dell'architettura proposta per il progetto è formato da un blocco di Scraping [3]. Infatti, per poter soddisfare l'obiettivo principale, bisognerà essere a disposizione di un gran numero di informazioni che andranno a formare il dataset di partenza su cui i moduli successivi svolgeranno le loro analisi creando le attività.

In particolare, l'utilizzo specifico di questo modulo sarà quello di

- **Raccolta di dati turistici:** automatizzare la raccolta di informazioni riguardo le attrazioni turistiche, eventi, recensioni.
- **Analisi delle tendenze di mercato:** monitorare e analizzare i trend di mercato per adattare le offerte turistiche alle preferenze dei clienti.
- **Creazione di database informativi:** costruire un database di informazioni aggiornate e dettagliate per supportare le decisioni strategiche, creando le fondamenta della conoscenza da dare in input alla pipeline IA prevista dal progetto.

Ad ogni modo, prima di poter iniziare a trattare il tema dello scraping è necessario definire le fonti su cui si effettuerà. Infatti, nel contesto dell'innovazione nel settore turistico, l'accesso ad informazioni dettagliate ed aggiornate è fondamentale per sviluppare esperienze personalizzate e soddisfare quindi le sempre più specifiche esigenze dei turisti. Per tale motivo, l'uso strategico delle fonti web riveste un ruolo cruciale.

Tali fonti includeranno siti web di esperienze turistiche ma anche social network e portali istituzionali. Ciò farà sì che le informazioni estratte forniscano non solo una panoramica dettagliata delle offerte disponibili sul mercato ma anche preferenze e feedback degli utenti.

Con questo approccio si semplifica quindi la comprensione delle tendenze di mercato e si potranno identificare opportunità di personalizzazione tramite l'intelligenza artificiale. Ad ogni modo le informazioni estratte dovranno essere compatibili e sfruttabili dai moduli successivi.

La prima attività è stata dunque quella di individuare le keywords di indagine, connesse ad un business di riferimento ed alla domanda di mercato. Una prima è stata definita in base all'area geografica di riferimento, secondo le prospettive di sviluppo dell'azienda (Spagna > Andalusia). Le altre riguardano la fruizione del prodotto turistico nelle sue varie declinazioni (viaggio, vacanza, esperienza).

Inoltre, in accordo con l'azienda partner, si era deciso di fare una prima selezione solo per la lingua italiana. Sarà comunque possibile applicare le medesime keywords anche per l'inglese e lo spagnolo. Le keywords definite sono state utilizzate nella ricerca e selezione delle principali fonti web. Sono stati considerati i maggiori siti fornitori di esperienze per valutare il ventaglio tipologico dell'offerta.

Parallelamente, sono stati selezionati i maggiori social media per studiare le conversazioni degli utenti in modo da rilevare i commenti sulle esperienze vissute ed i "desiderata" in relazione alle esperienze potenziali.

Sulla base di una prima individuazione, sono stati selezionati i siti web che presentavano le minori problematiche tecniche legate all'estrazione dei dati (che tratteremo in seguito). Inoltre, sono stati selezionati mediante ricerca su Google i

principali siti istituzionali legati all'informazione turistica dell'area di interesse. Le scelte sono ricadute sui seguenti siti raggruppati per categoria:

- Siti fornitori di esperienze come Tripadvisor ed Airbnb.
- Siti istituzionali e blog come “ilmiodiariodiviaggio.com” o “GetYourGuide”
- Social media come Facebook, cercando in gruppi di viaggio più per ricevere riscontri su attività piuttosto che l'attività in sé.

Chiaramente per ogni fonte sarà necessaria una soluzione specifica che sarà trattata nei seguenti sottocapitoli.

2.1.1 Arricchimento della conoscenza basata sul Web

La raccolta di informazioni è un'attività fondamentale e comune a molti business; In un contesto turistico, è necessario conoscere in modo approfondito la/le regione/i nella quale si vuole proporre qualcosa e le attività ivi disponibili. Quest'ultime potrebbero ad esempio dipendere anche dal periodo dell'anno o da altre variabili.

Per i locali, tali informazioni potrebbero essere scontate, ma un esterno che volesse inserirsi come fruitore potrebbe riscontrare maggiori difficoltà e rischiare di essere scontato o non valido nelle proposte, soccombendo ad altri fornitori.

Fino ad oggi quest'attività di raccolta doveva necessariamente essere fatta a mano da operatori affidandosi a lunghe ricerche online o, nel caso si avessero conoscenze, chiedendo a locali. Ciò poteva essere molto costoso in termini economici ma anche di tempo, inoltre un buon risultato non era garantito.

Oggi giorno si può far affidamento al “web scraping” (o “data scraping”), una tecnica di ultima generazione automatica dove, comunque, vi sarà una supervisione umana che però si limita alla verifica dei risultati e che quindi potrà non prestarsi più attivamente alla ricerca in sé.

Con “web scraping” si intende quindi l'attività di raccolta di una grande mole di informazioni tramite l'ausilio di appositi tool di estrazione dati a partire dalla conoscenza presente su internet. Questa pratica permette di automatizzare la raccolta di dati a partire da siti web, trasformando il contenuto presente nelle pagine web in dati strutturati che possono essere comodamente analizzati e utilizzati per vari scopi.

Negli ultimi anni, l'utilizzo del web scraping è cresciuto notevolmente grazie all'aumento dell'accessibilità delle tecnologie e disponibilità di strumenti e librerie open-source, alla portata di tutti. Il web scraping trova applicazioni in diversi settori e per differenti scopi: aziende, ricercatori e sviluppatori indipendenti lo utilizzano per una vasta gamma di applicazioni, che spaziano dall'analisi di mercato e dal monitoraggio dei prezzi, alla raccolta di dati per progetti di machine learning ed alla creazione di database di informazioni pubblicamente disponibili.

Il web scraping ha quindi introdotto numerosi vantaggi, tra i principali vi sono sicuramente:

- **Automazione:** permette di automatizzare la raccolta di dati, sostituendo attività manuali che richiederebbero tempo e risorse significative.
- **Efficienza:** consente di raccogliere grande quantità di dati in tempi relativamente brevi, migliorando l'efficienza complessiva del processo di raccolta dati.
- **Accesso a dati diversificati:** offre la possibilità di raccogliere dati da una vasta gamma di fonti online, garantendo una visione più completa delle informazioni disponibili. Questo tipo di approccio permette di aggregare dati da diverse prospettive, migliorando la qualità e la profondità delle analisi.

2.1.2 Tecnologie e librerie

Lo scraping può essere effettuato tramite diversi tool, ognuno con i propri vantaggi e svantaggi; di seguito verranno analizzate le diverse tecnologie.

BeautifulSoup

Libreria Python progettata per semplificare l'analisi e la manipolazione di documenti HTML e XML [4]. Tramite l'utilizzo di questa tecnologia, è possibile convertire il markup (tendenzialmente complesso) in un albero di oggetti Python che possono essere facilmente navigati e manipolati. Questo rende semplice estrarre dati specifici da una pagina web, come titoli, link, paragrafi, url delle immagini e altri elementi HTML.

Pro:

- **Facilità d'uso:** offre un'interfaccia Python intuitiva e facile da imparare, rendendo l'analisi dei documenti accessibile anche a chi non ha dimestichezza con l'argomento.
- **Flessibilità:** consente di cercare, filtrare e manipolare facilmente i tag e il contenuto del documento, adattandosi alle esigenze specifiche di estrazione dei dati.
- **Open Source:** essendo open-source, è liberamente disponibile e supportata da una comunità attiva. Questo vuol dire che gli utenti stessi possono contribuire al suo sviluppo, segnalare bug e migliorare la libreria nel tempo. Inoltre, è possibile utilizzarla in maniera gratuita per qualsiasi scopo.

- Integrazione con altre librerie: si integra molto bene con tre librerie Python, come ad esempio Requests per il download delle pagine web e Pandas per l'analisi dei dati estratti.
- Documentazione dettagliata: ha una documentazione chiara e dettagliata, con numerosi esempi che aiutano gli utenti a comprendere e utilizzare correttamente le sue funzionalità.

Contro:

- Mancato supporto per JavaScript: non è in grado di eseguire JavaScript, il che significa che non può gestire il contenuto dinamico generato da script, rendendolo meno adatto per il web scraping di pagine web con contenuti dinamici.
- Velocità di esecuzione: può essere più lento rispetto ad altre librerie di scraping più specializzate, specialmente quando si analizzano pagine web molto grandi e/o complesse.
- Parsing inefficace: in alcuni casi, il parsing del markup può risultare inefficiente, specialmente se il documento è estremamente complesso e/o mal formattato, rallentando l'analisi complessiva.
- Manutenzione: richiede un certo grado di manutenzione per adattarsi ai cambiamenti nella struttura delle pagine web, poiché le modifiche del markup possono influenzare la corretta estrazione dei dati.

Scrapy

Framework per il web scraping in Python, progettato per estrarre, elaborare e memorizzare dati da pagine web in modo efficiente [5]. Utilizza un'architettura asincrona per gestire grandi volumi di richieste simultanee, offrendo flessibilità con il supporto per selettori CSS, XPath e pipelines di dati.

Pro:

- Efficienza e velocità: è progettato per essere veloce ed efficiente, sfruttando delle tecniche asincrone per scaricare ed elaborare molte pagine web contemporaneamente.
- Framework completo: offre una soluzione completa per il web scraping, incluso il download delle pagine, l'estrazione dei dati, la gestione delle richieste e la pulizia dei dati.
- Gestione automatica delle richieste: gestisce in automatico le richieste HTTP e il crawling delle pagine, facilitando il rispetto delle norme di scraping.

- Supporto per XPath e CSS Selectors: consente di utilizzare sia XPath sia selettori CSS per discretizzare gli elementi delle pagine e i relativi dati da estrarvi, offrendo grande flessibilità nell'analisi dei documenti.
- Estendibilità: avendo una struttura modulare, permette agli sviluppatori di personalizzare il comportamento di scraping attraverso middleware e plugin.
- Integrazione con pipeline di dati: supporta pipelines di dati per pulire, elaborare e memorizzare i dati estratti in vari formati, come JSON, CSV e database SQL/NoSQL.
- Open Source: essendo open-source, è liberamente disponibile e supportato da una comunità attiva. Questo vuol dire che gli utenti stessi possono contribuire al suo sviluppo, segnalare bug e migliorare la libreria nel tempo. Inoltre, è possibile utilizzarlo in maniera gratuita per qualsiasi scopo.
- Documentazione dettagliata: ha una documentazione chiara e dettagliata, con numerosi esempi che aiutano gli utenti a comprendere e utilizzare correttamente le sue funzionalità.

Contro:

- Curva di apprendimento ripida: essendo un framework complesso, ha una curva di apprendimento più ripida rispetto a strumenti più semplici (come ad esempio BeautifulSoup, sopracitato), richiedendo più tempo per imparare ad utilizzarlo ma soprattutto per configurarlo in maniera corretta.
- Configurazione iniziale: configurare un progetto Scrapy è un'operazione da eseguire in maniera preliminare rispetto all'avvio dei lavori di scraping, la quale può richiedere parecchio tempo.
- Mancato supporto per JavaScript: non è in grado di eseguire JavaScript, il che significa che non può gestire il contenuto dinamico generato da script, rendendolo meno adatto per il web scraping di pagine web con contenuti dinamici.

Selenium

Non è progettato specificamente come strumento di web scraping, bensì come suite di strumenti per l'automazione dei browser web [6]. Non è quindi utilizzabile di per sé per lo scraping ma può essere utilizzato in combinazione con le librerie precedenti aumentando quindi il grado di automazione. È particolarmente impiegato per il testing delle applicazioni web, permettendo di simulare interazioni utente come il click, inserimento di testo e navigazione. Supporta i principali web browser e

può essere utilizzato con diversi linguaggi di programmazione (tra cui Python e JavaScript). Tornando al web scraping, risulta particolarmente utile per le pagine che caricano contenuto dinamico tramite JavaScript.

Pro:

- Automazione completa del browser: permette di automatizzare tutte le interazioni con il browser, come il click, inserimento di testo e navigazione.
- Open Source: essendo open-source, è liberamente disponibile e supportato da una comunità attiva. Questo vuol dire che gli utenti stessi possono contribuire al suo sviluppo, segnalare bug e migliorare la libreria nel tempo. Inoltre, è possibile utilizzarlo in maniera gratuita per qualsiasi scopo.
- Gestione del JavaScript dinamico: è in grado di gestire contenuti dinamici generati da JavaScript, funzionalità che molti strumenti non sono in grado di coprire (come visto, contro delle tecnologie sopracitate).

Contro:

- Prestazioni: è più lento rispetto ad altri strumenti di web scraping, poiché simula un browser completo.
- Configurazione complessa: la configurazione e l'utilizzo possono essere più complessi rispetto ad altri strumenti di scraping.
- Manutenzione: richiede frequenti aggiornamenti e manutenzione per rimanere compatibile con le ultime versioni dei browser.
- Consumo di risorse: eseguendo un browser completo e simulandone il comportamento, consuma più risorse di sistema, rendendolo meno efficiente per operazioni di scraping su larga scala.

Apify

Piattaforma cloud per l'automazione del web scraping e l'estrazione dei dati, che permette agli utenti di creare, eseguire e gestire web scraper, crawler e automazioni web [7]. Offre un'infrastruttura scalabile che permette di raccogliere dati da qualsiasi sito web configurato, gestendo anche i contenuti dinamici generati da JavaScript. Gli utenti possono creare scraper personalizzati utilizzando JavaScript o scegliere tra numerosi scraper predefiniti nella sua libreria. Apify fornisce anche strumenti per la gestione delle richieste HTTP, il proxying, la rotazione degli IP e il salvataggio dei dati estratti in vari formati (JSON, CSV).

Inoltre, dispone di un'interfaccia utente web per configurare e monitorare facilmente i propri scraper, senza l'utilizzo di codice esplicito (a differenza dei metodi

sopra citati, che consistono essenzialmente in librerie Python e dove si richiede mandatoriamente la scrittura di codice) Vi è comunque la possibilità di integrare le API utilizzandole in un codice in modo da avere un utilizzo più specifico.

Pro:

- Scalabilità: offre un'infrastruttura cloud scalabile che può gestire grandi volumi di dati, rendendola ideale per progetti di scraping su larga scala.
- Gestione dei contenuti dinamici: supporta le operazioni di web scraping di pagine web con contenuti dinamici generati da JavaScript, grazie all'integrazione con strumenti con Puppeteer e Playwright.
- Facilità d'uso: la piattaforma fornisce un'interfaccia utente intuitiva, scraper predefiniti e una vasta libreria di template per configurare e monitorare le operazioni di scraping senza dover scrivere codice complesso.
- Integrazione: può salvare i dati estratti in vari formati come JSON o CSV, e si integra facilmente con altre applicazioni e servizi tramite API.
- Supporto per proxies e rotazione IP: fornisce funzionalità di proxying e rotazione degli IP per evitare blocchi da parte dei siti web e migliorare l'affidabilità dello scraping.
- Documentazione e comunità: ha una documentazione chiara e dettagliata, con numerosi esempi che aiutano gli utenti a comprendere e utilizzare correttamente le sue funzionalità.

Contro:

- Costi: essendo una piattaforma cloud, i costi possono diventare significativi per volumi elevati di scraping o per l'uso di funzionalità avanzate, rappresentando un fattore limitante per budget ridotti.
- Curva di apprendimento: nonostante l'interfaccia user-friendly, la creazione di scraper personalizzati richiede una notevole familiarità con JavaScript. Inoltre, la scelta degli scraper presenti sul mercato richiede parecchio tempo, in quanto bisogna studiare quali tra questi si presta meglio a seconda del contesto di utilizzo, attraverso vari test.
- Limitazioni sui siti protetti: come altri strumenti di scraping, può incontrare difficoltà con siti web che utilizzano misure di protezione come CAPTCHA, rilevamento di bot e restrizioni IP.
- Dipendenza dal cloud: l'affidabilità del servizio dipende dalla disponibilità e dalle prestazioni della piattaforma cloud di Apify, il che può costituire un potenziale single point of failure.

- **Performance variabile:** le prestazioni dei singoli web scraper possono variare a seconda della complessità delle pagine web che sono oggetto di analisi e in base alla configurazione delle risorse cloud.

2.2 Preprocessing

I dati raccolti allo step precedente, provenendo da siti differenti, potrebbero avere formattazioni diverse oltre a dati mancanti; è quindi necessario adattarli prima di poterli utilizzare. Il preprocessing si suddivide in diverse fasi, ciascuna mirata a risolvere specifiche problematiche.

In questa fase di studio si è cercato di definire quali fossero le più adatte da realizzare per il caso di interesse. Nel seguente elenco si elencano le principali:

- **Pulizia dei dati:** in questa fase, vengono rimossi eventuali dati duplicati, incongruenti o incompleti. Le informazioni mancanti vengono gestite con tecniche di imputation, come la sostituzione con medie, modelli di regressione o semplici valori predefiniti.
- **Standardizzazione:** dato che le fonti possono presentare formati differenti sia internamente (ad esempio, date in formati diversi, valute espresse in unità differenti, ecc.) che nel formato del file (JSON oppure testuale), i dati vengono uniformati secondo uno standard comune per facilitare le successive analisi. Inoltre potrebbe essere necessario l'adozione di un'unica lingua e quindi tradurre eventuali dati.
- **Normalizzazione e trasformazioni:** alcuni campi potrebbero richiedere normalizzazione (es. riduzione della scala di variabili numeriche) o trasformazioni logaritmiche per migliorare la distribuzione dei dati e la performance degli algoritmi.
- **Tokenizzazione e pulizia del testo:** per i dati testuali è poi possibile eseguire operazioni di tokenizzazione, rimozione di stopwords e stemming/-lemmatizzazione, trasformando il testo grezzo in un formato più facilmente processabile dagli algoritmi di machine learning.
- **Rilevazione e gestione dei dati anomali:** un ultimo step è quello di individuare e trattare eventuali outlier o valori anomali che potrebbero distorcere i risultati delle analisi.

2.2.1 Teoria Preprocessing

Tra quelle elencate, tre sono le tecniche atte a favorire l'efficienza degli algoritmi di machine learning o comunque di reti neurali che opereranno sui dati di input:

- **Tokenizzazione:** la tokenizzazione è il processo di suddivisione di un testo in unità più piccole, chiamate token. I token possono essere parole, frasi o persino caratteri, a seconda del livello di dettaglio desiderato. Questo passaggio è cruciale per la maggior parte delle applicazioni di elaborazione del linguaggio naturale (NLP), poiché i computer trattano meglio le parole o simboli isolati piuttosto che un blocco di testo continuo.

Ad esempio, dato il testo: “Il gatto salta sul tavolo”, la tokenizzazione a livello di parola lo suddividerà in: [“*Il*”, “*gatto*”, “*salta*”, “*sul*”, “*tavolo*”].

Questo processo permette di analizzare il testo in modo più semplice e di applicare successivamente tecniche di analisi come il conteggio delle occorrenze, la costruzione di vettori di caratteristiche o la creazione di modelli predittivi.

- **Stemming:** lo stemming è una tecnica che riduce le parole alla loro radice o forma base, chiamate *stems*. L’obiettivo è ridurre le varianti morfologiche di una parola (ad esempio, plurali, coniugazioni o derivazioni) a una forma comune per semplificare l’elaborazione del testo. Questa tecnica è particolarmente utile quando si desidera trattare parole correlate come equivalenti, in modo da non contare più volte parole che significano la stessa cosa ma sono rappresentate in forme diverse.

Ad esempio, per la parola “correre” e varianti, lo stemming ridurrebbe: “*correvo*”, “*corrono*”, “*corriamo*” → “*corr*”. Tuttavia, è importante notare che lo stemming può talvolta produrre radici non esattamente corrette da un punto di vista linguistico (come nel caso di “*corr*” che non è una parola vera). Questo perché lo stemming applica regole meccaniche, senza tenere conto del significato, vi è quindi un’alternativa più raffinata chiamata lemmatizzazione, che cerca di ridurre le parole al loro lemma, ossia la forma base corretta, tenendo conto anche del contesto grammaticale.

- **Lemmatizzazione:** la lemmatizzazione è un processo che, come lo stemming, mira a ridurre le varianti morfologiche di una parola, ma lo fa in modo più sofisticato e preciso. La lemmatizzazione riduce le parole al loro lemma, che è la forma base corretta di una parola secondo il vocabolario di una lingua.

A differenza dello stemming, che tronca semplicemente una parola seguendo regole meccaniche, la lemmatizzazione considera il contesto e la parte del discorso per scegliere la forma base giusta. Questo significa che la lemmatizzazione tiene conto di come la parola viene usata grammaticalmente in una frase per identificare la sua forma standard. Ad esempio:

Le forme “*correvo*”, “*corrono*”, “*corriamo*” diventerebbero tutte “*correre*” (il lemma corretto).

La lemmatizzazione è particolarmente utile quando si cerca di preservare il significato originale delle parole, poiché garantisce che la radice restituita sia una parola esistente e valida nella lingua. Per eseguire la lemmatizzazione, è spesso necessario conoscere la parte del discorso (ad esempio, verbo, sostantivo) di ogni parola, il che rende questo processo più complesso ma anche più accurato rispetto allo stemming.

Per ricapitolare, la tokenizzazione è un'operazione necessaria a dividere le parole per poterle poi analizzare indipendentemente mentre stemming e lemmatizzazione riducono il numero di varianti con lo stesso significato. In particolare lo stemming applica regole semplici per rimuovere prefissi o suffissi da una parola, portando a radici spesso non riconoscibili come parole reali, la lemmatizzazione cerca di ridurre la parola a una forma valida e grammaticalmente corretta.

2.3 Clustering dei Dati e Tagging

Dopo aver ottenuto i dati di input questi ultimi vengono passati al modulo successivo della pipeline, ovvero il modulo di “Analisi e Clustering dei dati”. Più in dettaglio, utilizzando algoritmi di clustering avanzati, il sistema categorizza le sue attrazioni in base a vari attributi come la vicinanza geografica, i tipi di servizi che offrono e le valutazioni degli utenti. Questa categorizzazione aiuta a creare itinerari di viaggio coerenti e logici che raggruppano attività simili che rispecchino le richieste dell'utente o per fornire suggerimenti alternativi.

Nel contesto progettuale, questo modulo ha quindi un obiettivo fondamentale: quello di analizzare i dati in modo da trovare pattern comuni in essi e poterli poi raggruppare a seconda di questi ultimi. In particolare, si definiscono quattro elementi principali in grado di discriminare le suddette attività:

- **Luogo geografico**, in modo da circoscrivere le attività ad una ristretta zona geografica.
- **Periodo dell'anno**, per poter distinguere attività stagionali e selezionare le più indicate in base alla stagione.
- **Tag primario**, essenzialmente la categoria principale cui appartiene l'attività.
- **Tag secondario**, sottocategoria in modo da poter specificare maggiormente l'attività.

In ottica di utilizzo si prevedono due modalità. La prima, dove inserendo tutti e quattro gli elementi verrà ritornato un insieme di attività (in numero piccolo) che corrispondano appunto a quelli inseriti. L'utilizzatore potrà poi sceglierne una o più in modo da analizzarle meglio ed eventualmente formare il “pacchetto”.

Secondariamente si potrà inserire un numero minore di filtri: sarà quindi il sistema a completare il quadro secondo i pattern trovati. In questo modo sarà possibile proseguire come nella casistica precedente.

Si era poi parlato di attribuire un punteggio alle attività analizzate; quest'ultimo, durante un'ipotetica ricerca, sarebbe infatti necessario ad ordinare le attività di interesse mostrando quindi per prime le più rilevanti (nel caso considerato, le più singolari).

2.3.1 Teoria del Clustering

Il clustering è una tecnica di machine learning non supervisionato utilizzata per raggruppare dati simili in insiemi (chiamati cluster) sulla base di caratteristiche comuni. Ogni cluster contiene dati che sono più simili tra loro rispetto a quelli in altri cluster. Per clusterizzare dati relativi ad attività di natura turistica e categorizzarle sulla base di caratteristiche comuni (luogo, descrizione o altro), si possono utilizzare diversi algoritmi di clustering.

Durante questa fase, l'obiettivo è raggruppare le attività in base a vari attributi e trovare un pattern di similarità tra di esse. Generalmente tale attività è eseguita su dati numerici. La maggior parte dei dati che saranno utilizzati sono però in forma testuale. Sarà quindi necessario effettuare una conversione dal formato testuale ad una rappresentazione numerica.

Si andrà ad effettuare la vettorizzazione dei dati, ossia la conversione di dati testuali in vettori multidimensionali che rappresentano le parole all'interno del documento.

Tale vettorizzazione può essere effettuata utilizzando diverse tecniche:

- **TF-IDF**: converte in un formato numerico il corpus facendo emergere termini specifici. I termini molto rari o molto comuni ricevono un punteggio basso, in quanto meno significativi per l'analisi. TF sta per Term Frequency, mentre IDF sta per Inverse Document Frequency. Il valore TF-IDF aumenta proporzionalmente al numero di volte che una parola appare nel documento ed è compensato dal numero di documenti nel corpus che contengono quella parola.
- **Bag-of-words**: è una rappresentazione semplificata utilizzata nell'elaborazione del linguaggio naturale e nel recupero delle informazioni per descrivere il contenuto testuale dei documenti. In questo modello, un documento viene rappresentato come un insieme (chiamato *bag*) di parole, senza tener conto della grammatica e dell'ordine delle parole, ma solo della loro frequenza. Ogni valore del vettore rappresenta dunque il conteggio della parola specifica nel documento.
- **Word2Vec**: questo metodo prevede l'elaborazione di un corpus di testi per apprendere le associazioni tra le parole. Si basa sull'ipotesi che le parole che

appaiono vicine in un testo abbiano somiglianze semantiche. Aiuta a mappare parole semanticamente simili in vettori di embedding che sono geometricamente vicini nello spazio vettoriale.

- **GloVe**: a differenza di Word2Vec, che crea embeddings di parole utilizzando il contesto locale. GloVe si concentra sul contesto globale per creare gli embeddings e la relazione semantica tra le parole è ottenuta utilizzando una matrice di co-occorrenza.
- **BERT**: utilizzando questo modello, si possono estrarre gli embeddings dagli ultimi layer del modello. Questo processo è possibile tramite delle librerie predefinite di Hugging Face (piattaforma leader nell'elaborazione del linguaggio naturale, facilita la ricerca e l'implementazione di modelli avanzati di machine learning, promuovendo la democratizzazione dell'IA). È un metodo che cattura meglio il significato delle parole, ma è molto costoso in termini di risorse computazionali.

Dopo aver scelto il modo di rappresentare le informazioni testuali (nel nostro caso TF-IDF) è possibile effettuare il vero e proprio clustering.

2.3.2 Tecnologie per il Clustering

Vi sono diversi algoritmi di clustering, ognuno con i propri pregi e difetti.

K-Means

Il primo analizzato è il K-Means: algoritmo basato sulla partizione, che suddivide i dati in un numero predefinito di cluster (parametro k), minimizzando la distanza tra i punti all'interno di ogni cluster ed il centroide (punto medio) del cluster stesso. Nel caso di dati testuali si utilizzano tecniche di word embedding per trasformare i testi in vettori numerici che rappresentano le parole o frasi in uno spazio vettoriale.

Pro:

- Noto per la sua semplicità concettuale ed implementativa, rendendolo accessibile anche a chi ha conoscenze di base di machine-learning.
- Efficiente computazionalmente ed è in grado di gestire grandi volumi di dati in modo rapido, rendendolo adatto per applicazioni su larga scala.

Contro:

- Richiede la predefinitone del numero di cluster (parametro k).

- È sensibile agli *outliers*, ossia i punti dati che si discostano significativamente dalla maggior parte dei dati, influenzando negativamente la formazione dei cluster.
- Il risultato dell'algoritmo dipende fortemente dal posizionamento iniziale dei centroidi, che può dunque portare a soluzioni diverse in base alla scelta iniziale.
- È necessario convertire i dati testuali in vettori numerici mediante delle tecniche di word embedding, il che può introdurre problemi di sparsità e perdita di informazioni durante la rappresentazione vettoriale.

DBSCAN

Segue il DBSCAN (Density-Based Spatial Clustering of Applications with Noise): è un algoritmo di clustering che identifica cluster basati sulla densità dei punti, riuscendo a rilevare cluster di forma arbitraria e a distinguere il rumore dei dati validi. Funziona definendo punti centrali (*core points*), punti di confine (*border points*) e punti di rumore (*noise points*), raggruppando i punti che sono vicini in termini di distanza.

È particolarmente utile per dati con distribuzioni spaziali complesse e non richiede di specificare il numero di cluster in anticipo. Tuttavia, la scelta dei parametri ϵ (distanza massima tra due punti per essere considerati nel medesimo cluster) e MinPts (numero minimo di punti per formare un cluster) è cruciale per le sue prestazioni.

Pro:

- È in grado di identificare cluster di forme non necessariamente sferiche e con densità variabile, adattandosi bene a strutture complesse nei dati.
- Riconosce e gestisce efficacemente gli outliers come punti non assegnati a nessun cluster, contribuendo a una segmentazione più accurata dei dati.
- Può utilizzare metriche di similarità come la cosine similarity (misura di similarità tra due vettori nel contesto della rappresentazione vettoriale) per calcolare le distanze tra punti, consentendo di raggruppare insieme recensioni simili in cluster di attrazioni o altri contesti.

Contro:

- Può mostrare una diminuzione delle prestazioni quando applicato a insiemi di dati particolarmente grandi, richiedendo più tempo computazionalmente per completare l'analisi.
- La performance dipende fortemente dalla correlazione dei parametri, come la densità minima e la distanza tra i punti, il che può essere non intuitivo e richiedere esperimenti iterativi per ottenere risultati ottimali.

Clustering gerarchico

Vi è poi il Clustering gerarchico: algoritmo di clustering in cui viene costruita una gerarchia di cluster, partendo dai punti singoli e aggregandoli gradualmente in base alle loro somiglianze, fino al livello di granularità desiderato. Per dati testuali si applica inizialmente una trasformazione tramite TF-IDF o embeddings.

Pro:

- Fornisce una struttura ad albero che rappresenta le relazioni di similarità tra i dati, consentendo una visualizzazione intuitiva e dettagliata della struttura dei cluster.
- Particolarmente adatto per esplorare e comprendere le relazioni complesse tra le attrazioni o altri elementi di interesse, permettendo di identificare cluster a diversi livelli di dettaglio.
- Non è necessario specificare in anticipo il numero dei cluster, poiché il numero dei cluster può essere determinato analizzando la gerarchia risultante.

Contro:

- L'analisi gerarchica può essere onerosa dal punto di vista computazionale quando applicata a grandi volumi di dati, richiedendo risorse significative per eseguire le operazioni di clustering.
- L'interpretazione della struttura gerarchica può essere complessa, specialmente quando la gerarchia è profonda o contiene molti cluster e sottocluster.
- Può essere più lento nell'esecuzione, soprattutto se non ottimizzato correttamente. La scelta del metodo di linkage utilizzato per unire i cluster influisce sui risultati e sulle prestazioni dell'algoritmo.

Ottimizzazioni

Indipendentemente dalla tecnologia scelta, è possibile adottare la seguente ottimizzazione: in caso di molti dati si può ricorrere all'analisi delle componenti principali, o PCA. È un metodo di riduzione della dimensionalità di grandi insiemi di dati, trasformando insieme di variabili in uno più piccolo che contiene ancora la maggior parte delle informazioni nell'insieme di grandi dimensioni. La riduzione del numero di variabili di un set di dati va naturalmente a discapito dell'accuratezza, ma set di dati più piccoli sono più facili da esplorare e visualizzare e rendono l'analisi dei dati molto più semplice e veloce per gli algoritmi di apprendimento automatico senza variabili superflue da elaborare.

2.3.3 Tagging

Con il clustering è quindi possibile raggruppare in gruppi i dati forniti (in questo contesto le attività) in base a quanto esse sono simili tra di loro. In questo modo però viene ancora a mancare la tipologia che accomuna le attività in un gruppo, ovvero il tag. Quest'ultimo deve essere definito a posteriori a seconda delle attività presenti.

Si è considerato quindi un secondo approccio sfruttando una rete neurale già esistente, BART:

- **BART** (Bidirectional and Auto-Regressive Transformers) è un modello di linguaggio sviluppato da Facebook AI, ed è disponibile attraverso la libreria di Hugging Face [8]. È progettato per essere versatile e potente per una vasta gamma di compiti di elaborazione del linguaggio naturale (NLP), tra cui riassunto, traduzione, generazione di testo.

Tale modello è in grado di, ricevuta la descrizione di un'attività ed un insieme di descrittori, indicare quale tra quelli forniti è più adatto a descrivere l'attività.

In questo caso è quindi possibile far classificare le varie attività una volta che si è definito tale insieme. Il vantaggio è che adesso è possibile definire non solo tag primari o secondari ma anche ad esempio la stagionalità. Inoltre, essendo la rete già allenata, essa è pronta all'utilizzo fin da subito.

2.3.4 Punteggio di unicità

Un'ultima attività prevista dal modulo corrente è l'attribuzione di un punteggio alle attività. Essendo uno degli obiettivi del progetto, quello di favorire l'under-tourism, si è quindi riflettuto su quali elementi potessero distinguere le attività tra loro e soprattutto quando un'attività potesse essere più indicata al completamento di tale obiettivo.

Si sono considerati i risultati del clustering applicati in un contesto tematico piuttosto che geografico. In questo modo, infatti, le attività che appartengono al medesimo cluster dovrebbero essere relativamente affini tra loro e quindi appartenere alla stessa categoria.

A questo punto, per ogni attività, si è cercato di computare un valore numerico, rappresentativo del punteggio, che stimasse quanto l'attività fosse unica: difatti si è ipotizzato che meno elementi fossero presenti in un cluster, più quest'ultimo fosse particolare; inoltre si è supposto che più il singolo elemento fosse lontano dal centroide, e quindi marginale, più quest'ultimo potesse essere rilevante.

La formula risultante è la seguente:

$$score = \frac{(distance_weight \cdot distance)}{(size_weight \cdot cluster_size)} \quad (2.1)$$

Dove *distance* rappresenta la distanza della singola attività dal centroide, *cluster_size* è il numero di elementi nel cluster, mentre *distance_weight* e *cluster_weight* rappresentano dei pesi assegnati in modo tale da ottenere un punteggio compreso tra 0 ed 1. Nonostante i risultati ottenuti risultassero interessanti, dove attività reputabili più comuni (ad esempio visite guidate in monumenti famosi) avessero punteggi più bassi rispetto ad attività considerabili più particolari e meno conosciute, si è preferito non procedere per il momento con l'implementazione in modo tale da, eventualmente, effettuare maggiori ricerche a riguardo ed ottenere una formula più consistente. Potrà sicuramente esser considerato un punto di partenza per sviluppi futuri.

2.4 Copywriting generativo

Una volta che si sono analizzati tutti i dati aggiungendo le informazioni necessarie, sarà possibile selezionare le attività che soddisfano criteri specifici e, dopo averne scelta una o più, le si potrà combinare in un'unica esperienza.

Questo compito sarà svolto quindi dall'ultimo modulo della pipeline, formato dalla IA Generativa: Con IA Generativa si intende una classe di algoritmi che creano nuovi contenuti (in questo caso testuali), basati su una serie di dati di input. Nello specifico, apprendono la distribuzione dei dati di input e utilizzano la conoscenza appresa per generarne di nuovi che somiglino a quelli di addestramento.

2.4.1 Tecnologie

Le tecniche di Generative AI in questo contesto sono utili per poter generare itinerari turistici personalizzati basandosi sulle preferenze degli utenti, la durata del viaggio, il budget e gli interessi dell'utente.

Lo studio delle differenti tipologie di reti adatte a soddisfare tale richiesta si è focalizzato su due soluzioni:

- **Reti neurali ricorrenti (RNN):** le RNN sono progettate per gestire sequenze di dati, come il testo o le serie temporali. Hanno un ciclo che consente alle informazioni di persistere, passando da un passo temporale al successivo. Questo ciclo può essere considerato come una memoria che mantiene informazioni precedenti.

Le RNN sono soggette a problemi quali la scomparsa o l'esplosione del gradiente, che rendono difficile l'apprendimento di dipendenze a lungo termine nei dati, perciò ci sono varianti più avanzate di RNN, come le LSTM (Long Short-Term Memory) e le GRU (Gated Recurrent Units) che sono state sviluppate per mitigare questi problemi e sono in grado di catturare dipendenze a lungo termine meglio delle RNN standard.

- **LSTM**: sono in grado di ricordare meglio le informazioni su sequenze più lunghe. Le LSTM utilizzano un meccanismo di “celle di memoria” e “gate” per controllare il flusso di informazioni, consentendo loro di ricordare informazioni importanti per periodi di tempo più lunghi.
- **GRU**: utilizzano sempre il meccanismo di “celle di memoria” e “gate”, ma hanno una struttura più semplice rispetto alle LSTM, con meno parametri, il che le rende più facili da addestrare e meno costose da un punto di vista computazionale, ma anche meno flessibili.

Pro:

- Ottime per catturare relazioni a lungo termine e dipendenze tra gli elementi di una sequenza.
- In grado di apprendere automaticamente rappresentazioni significative dei dati di input.
- Numerosi modelli pre-addestrati su cui fare fine tuning disponibili su Hugging Face Transformers, Tensorflow o Pytorch.

Contro:

- Nonostante il problema del vanishing gradient sia parzialmente risolto, non sono adatte a sequenze troppo complesse.
 - Alto rischio di overfitting.
 - Molto costose da addestrare rispetto ad altre alternative.
- **Generative Pre-trained Transformer (GPT)**: è una serie di modelli [9] sviluppati da OpenAI e pre-addestrati su un enorme corpus di testo per apprendere al meglio le probabilità delle sequenze di parole. Viene addestrato utilizzando l'apprendimento per rinforzo da feedback umano (RLHF). Questo processo di formazione prevede una fase iniziale di messa a punto supervisionata, in cui gli vengono fornite conversazioni, seguita da una fase di apprendimento per rinforzo, in cui il modello viene premiato per le risposte che gli utenti valutano come le migliori.

Può essere vantaggioso fare fine-tuning su dati di viaggio per generare degli itinerari creativi. Più il prompt in ingresso è preciso, migliori saranno i risultati. Difatti progettare il sistema in modo tale che la generazione sia il più guidata possibile (tramite form per esempio) aiuterebbe a creare degli itinerari più personalizzati, permettendo un focus sui dettagli fondamentali alla corretta generazione.

In particolare, della famiglia GPT, i più performanti sono GPT-3.5 e GPT-4, poiché in grado di generare testo molto vicino al linguaggio umano e sono

capaci di adattare il testo in vari stili e formati, in base alle richieste. GPT- 4 può gestire sequenze testuali più lunghe del suo predecessore, e addirittura lavorare con input multimodali.

Pro:

- Genera testi coerenti e contestualmente rilevanti.
- Buone performance anche senza fine-tuning

Contro:

- Richiedono risorse computazionali significative.
- Potenziali costi API elevati poiché accessibili solo tramite OpenAI, che ha tariffe di tipo usage-based.

Capitolo 3

Progettazione del sistema

Dopo aver analizzato i moduli singolarmente, si è reso necessario comprendere come integrarli per consentire un funzionamento coordinato.

È stato fondamentale stabilire i collegamenti tra i moduli prima di decidere le tecnologie da adottare per ciascuno, in modo da poterli inserire in un flusso di esecuzione fluido, in cui l'output di un modulo diventasse l'input di quello successivo. Questa pianificazione ha permesso di garantire una transizione coerente dei dati tra i vari moduli, facilitando il processo complessivo.

3.1 Suddivisione pipeline

Ricapitolando, grazie alla definizione di contesto ed obiettivi oltre ai principali compiti da svolgere, si è proposta finalmente l'architettura per tale modello, composta da quattro moduli principali:

- **Scraping**, tecnica informatica di estrazione di dati da internet utilizzata per creare il dataset su cui basare la conoscenza del modello.
- **Preprocessing**, i dati raccolti provengono da siti differenti e potrebbero avere formattazioni diverse oltreché dati mancanti; è quindi necessario adattarli prima di poterli utilizzare.
- **Clustering dei Dati**, in tale sezione vi è la vera e propria analisi dei dati; l'obiettivo è quello di trovare pattern comuni per poterli raggruppare. Si utilizzeranno metodi di raggruppamento basati su criteri geografici e contestuali, inoltre si cerca di attribuire ad ogni attività un punteggio di "unicità".
- **Generazione**, tramite IA generativa, dopo aver selezionato più di un'attività da combinare si genera la descrizione; nell'interfaccia finale verranno visualizzate anche altre informazioni.

Di conseguenza, come già anticipato nel sottocapitolo “Obiettivo”, si è poi legato ogni modulo ad una delle principali task. La raccolta dati sarà affidata al modulo di scraping che da fonti su internet immagazzinerà le esperienze; chiaramente queste esperienze andranno uniformate nello stile e ci si affiderà al modulo di preprocessamento cui seguirà la fase di analisi, affidata al modulo di clustering e tagging, ed infine la generazione della descrizione a seconda delle attività da combinare.

Il tutto sarà inserito in un'interfaccia web che avrà l'obiettivo di mantenere innanzitutto il database con le informazioni e di fare poi da tramite tra il modello ed un utilizzatore. Sarà quindi un "wrapper" degli algoritmi e restituirà la risposta ad una determinata interrogazione.

3.2 Metodologia e flusso di esecuzione

Inizialmente, si era ipotizzata un'applicazione in tempo reale: selezionata una regione di interesse, si sarebbe proceduto alla costruzione della conoscenza relativa, effettuando lo scraping seguito dalle varie analisi per poi ottenere direttamente le esperienze raccomandate.

Questa modalità non è però possibile a causa delle tempistiche. Difatti sia lo scraping che la fase di analisi sono attività time-consuming che devono quindi essere effettuate prima dell'utilizzo vero e proprio. Esse possono richiedere anche diverse ore.

Avendo scartato quindi la soluzione di un unico flusso di esecuzione, si propone di dividere in due fasi:

- **Raccolta ed analisi**, dove si crea il database con le varie informazioni, arricchito dalle analisi necessarie. Eventualmente effettuata periodicamente ad intervalli mensili per aggiornare la conoscenza.
- **Interrogazioni**, si pone al modello la domanda, si vedrà in seguito secondo che modalità, e quest'ultimo darà in output la risposta.

Considerando quanto detto, diventa essenziale la presenza di un'interfaccia strutturata che offra prima di tutto la possibilità di gestire un database contenente i dati e, secondariamente, presenti due sezioni distinte per le due fasi.

Tale interfaccia sarà presentata successivamente. Tuttavia, considerando che il progetto di tesi si inserisce nel contesto del più ampio progetto TEIA, è opportuno precisare che ci si è concentrati sull'implementazione della seconda fase, mentre la prima è stata lasciata per la prosecuzione futura del progetto.

3.3 Portale web

I moduli fin ora trattati devono ancora essere inseriti in un flusso di esecuzione unico; difatti finora essi venivano considerati singolarmente per valutarne efficacia e prestazioni ma in un contesto di utilizzo si prevede che un utilizzatore inserisca i tag per ottenere le attività senza interruzioni.

Per ottenere questo comportamento si propone un'interfaccia web che faccia da wrapper per i vari algoritmi di IA e che interagisca con un database gestito nel backend del sistema. Più in dettaglio, si separa quindi logicamente lo scraping col processamento e le analisi dalla generazione dell'attività; la raccolta ed analisi verranno effettuate all'inizio, ed eventualmente ad intervalli regolari a distanza di mesi per mantenere aggiornate le informazioni, andando a creare il database su cui si baseranno le analisi.

Dopo aver fatto ciò sarà possibile interfacciarsi con l'applicazione come già detto precedentemente. L'utilizzatore potrà inserire un sottoinsieme dei tag per averne generati i rimanenti e poi lanciare la ricerca o direttamente inserire i quattro descrittori per ottenere i risultati. Dopo aver ottenuto questi ultimi, verranno scelte le attività da combinare e lanciando l'esecuzione si otterrà una scheda descrittiva.

3.3.1 Human-AI Interaction (HAAI)

Con "HAAI" si fa riferimento alla Human-AI Interaction, ovvero alla relazione e all'interazione tra gli esseri umani e i sistemi basati sull'intelligenza artificiale. Questo campo di studio si occupa di come le persone interagiscono con le tecnologie intelligenti e di come queste interazioni possano essere progettate per essere intuitive ed efficaci per gli utenti finali. L'obiettivo è creare esperienze utente che siano naturali, comprensibili e che massimizzino l'efficienza e la soddisfazione degli utenti nell'uso dei sistemi AI.

Sono stati definiti i requisiti utente per una web application che sfrutti l'interazione Human-AI, essi saranno richiesti nella prima schermata. La web application accetta come input una combinazione di massimo quattro parametri, organizzata in una matrice:

- **Dove:** presumibilmente la città.
- **Tag primario:** tipo di attività.
- **Tag secondario:** ulteriori dettagli per diversificare e caratterizzare l'attività.
- **Quando:** la stagione o il periodo dell'anno.

Il tag primario indica il tipo di attività, mentre i tag secondari forniscono ulteriori informazioni per meglio caratterizzare l'attività. Ad esempio, una combinazione

di tag primario e secondario potrebbe essere “escursione - cavallo” o “escursione - montagna”.

Lanciando la ricerca verrà ottenuto un elenco di attività sotto forma di schede consultabili. Cliccandone una, essa sarà mostrata in dettaglio con le seguenti informazioni:

- Luogo.
- Foto.
- Durata.
- Prezzo.
- Descrizione dell'attività.
- Tag primario e secondario.

Infine, sono previste le seguenti funzionalità:

- **Combinazione delle attività:** tramite l'elenco ottenuto, è possibile selezionare delle attività. In questo modo si otterrà una scheda unica dove prezzo e durata saranno la somma delle singole mentre la descrizione verrà ottenuta tramite il modello IA generativo (copywriting).
- **Editing dell'elenco:** gli utenti possono marcare attività come “non rilevanti” o eliminarle. Possono anche modificare le descrizioni delle attività, permettendo al modello di apprendere tramite reinforcement learning e correggere eventuali errori di associazione.
- **Salvataggio delle attività:** gli utenti possono salvare le attività ritenute migliori in una sezione dedicata, ad esempio “Le mie esperienze”, mentre scorrono l'elenco delle attività.

3.3.2 Studio dell'interfaccia

Per supportare il flusso di esecuzione proposto, è stata progettata un'interfaccia web che funge da punto di contatto tra l'utente e il sistema. Seguendo le linee guida selezionate, che saranno illustrate nel capitolo successivo, e tenendo in considerazione la separazione delle fasi operative (raccolta ed interrogazione), si è proceduto alla progettazione dell'interfaccia dell'applicazione. L'obiettivo principale era garantire un'interazione intuitiva e fluida per l'utente, con un'interfaccia che riflettesse chiaramente le due fasi precedentemente descritte: una sezione per la gestione del database e una per l'interrogazione del modello.

A tal proposito, è stato utilizzato Figma, uno strumento di design collaborativo che ha permesso di creare un prototipo dell'applicazione, fornendo una chiara visualizzazione dell'esperienza utente prevista. Il prototipo formulato avrebbe dovuto presentare le seguenti caratteristiche:

- **Sezione di gestione del database:** un'interfaccia che consente agli amministratori di visualizzare, modificare e aggiornare i dati raccolti. È possibile impostare intervalli di aggiornamento periodici per mantenere il database allineato con le analisi più recenti.
- **Sezione di interrogazione del modello:** pensata per essere semplice ed efficace. L'utente seleziona la regione di interesse e invia una richiesta, ottenendo in risposta le raccomandazioni personalizzate.

Ad ogni modo, per mancanza di tempo, la fase di sviluppo si è concentrata sulla seconda parte del prototipo, ossia la sezione relativa alle interrogazioni del modello. Questa sezione rappresenta un elemento fondamentale dell'applicazione, poiché permette all'utente finale di ottenere le raccomandazioni basate sull'analisi dei dati raccolti in precedenza.

Riguardo la prima sezione, si lascia la prototipazione e lo sviluppo a studi futuri. Il database è stato popolato lanciando gli script di scraping, preprocessing ed analisi manualmente tramite terminale.

Tornando alla progettazione della seconda sezione, non avendo punti di partenza, si è definito da zero i componenti grafici e logici dell'interfaccia verificando in seguito se essi fossero conformi a linee guida selezionate ad hoc (illustrate nel capitolo successivo).

Prima di analizzare in dettaglio le varie schermate proposte si discutono alcune scelte grafiche generiche:

- **Cromia:** l'uso di un gradient viola-arancione in alto fornisce un aspetto moderno e accattivante, mentre elementi chiave come i bottoni hanno uno sfondo scuro con testo bianco, garantendo un contrasto elevato e maggior leggibilità.
- **Layout:** gli elementi fondamentali sono stati disposti in modo da garantire una struttura chiara a colonne, permettendo la gestione visiva di un ampio volume di informazioni senza sovraccaricare l'utente, con l'obiettivo di migliorare l'usabilità complessiva.

Di seguito sono presentate le schermate essenziali dell'applicazione, ciascuna corredata da una breve descrizione. Queste immagini illustrano le principali funzionalità dell'interfaccia utente che saranno poi implementate in modo più completo:



Figura 3.1: Schermata di selezione tramite quattro selettori

- Selezione tramite quattro selettori (figura 3.1): questa schermata consente all'utente di filtrare le esperienze raccomandate utilizzando quattro diversi selettori, ognuno dei quali rappresenta una variabile chiave.
- Selezione tramite mappa (figura 3.2): schermata alternativa alla ricerca tramite

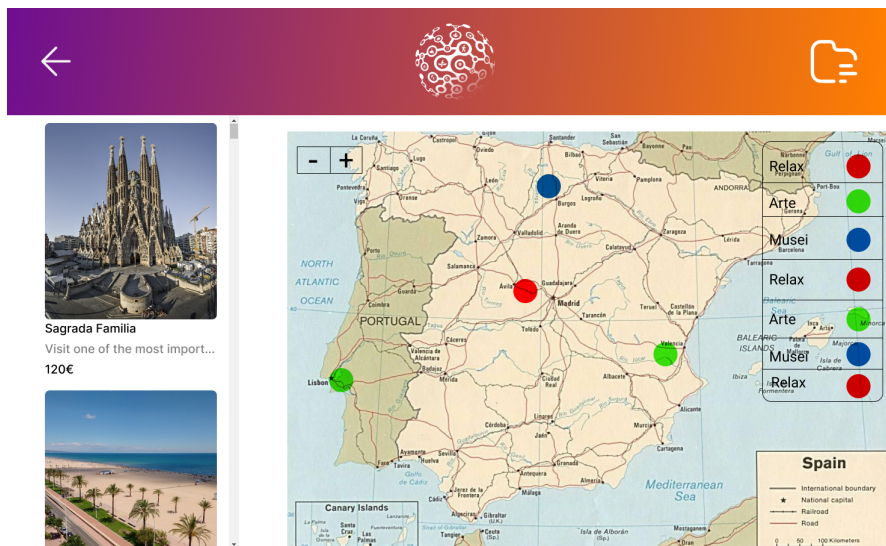


Figura 3.2: Schermata di selezione tramite mappa

selettori e raggiungibile tramite link apposto nella prima schermata. Gli

utenti possono visualizzare le opzioni disponibili su una mappa interattiva, disegnando la regione di interesse in modo più visivo e intuitivo sotto forma di area circolare.

- Risultati ottenuti dal database (figura 3.3): qui, l'utente può visualizzare i

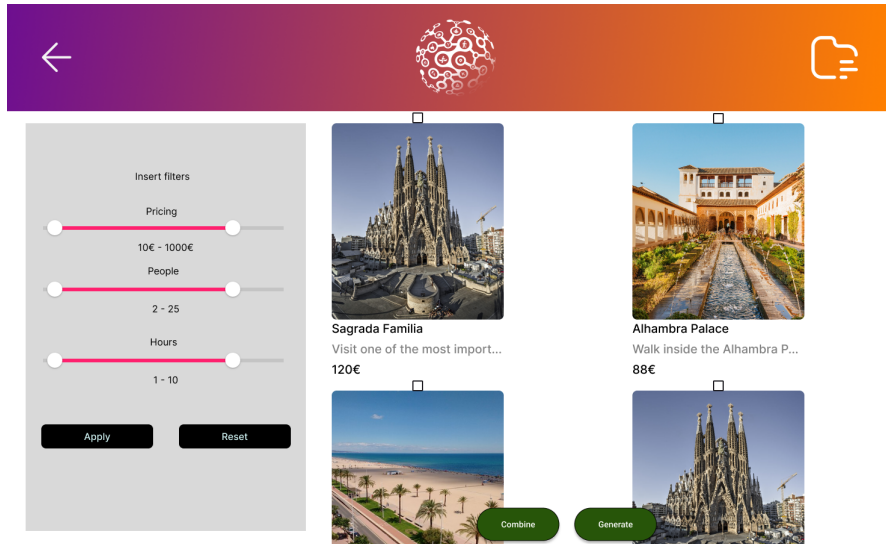


Figura 3.3: Schermata dei risultati ottenuti dal database, con possibilità di combinare o generare le attività

risultati ottenuti dal database; è innanzitutto possibile l'opzione di applicare ulteriori filtri all'elenco delle attività ottenute nonchè visionare in dettaglio una singola attività. Dopodichè sarà possibile generare un'esperienza unica a partire da un sottoinsieme dell'elenco risultante.

- Risultato della generazione (figura 3.4): questa schermata mostra il risultato

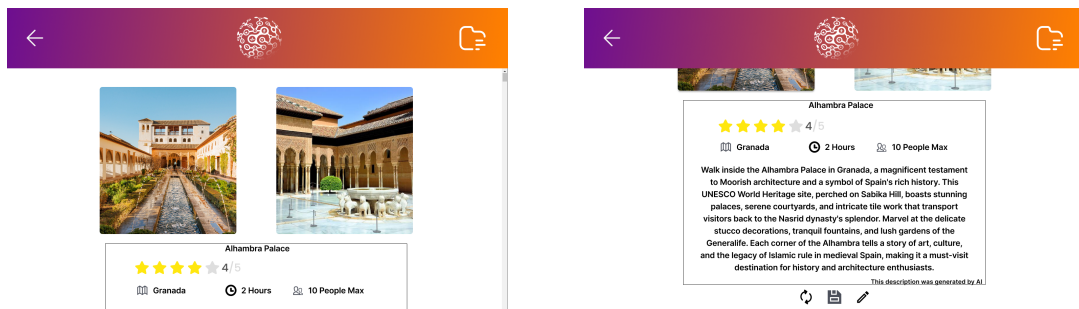


Figura 3.4: Schermate del risultato della generazione

finale generato dall'applicazione, fornendo all'utente tutte le varie informazioni.

Sarà poi possibile salvare il risultato ottenuto o far rigenerare la descrizione aggiungendo richieste specifiche.

In parallelo alla definizione dell'interfaccia si è studiato cosa servisse per implementare i moduli e quali fossero le scelte più convenienti.

Capitolo 4

Scelte Tecnologiche

Dopo aver valutato i pro ed i contro delle tecnologie disponibili nel background teorico ed aver progettato il sistema per garantire un flusso operativo efficiente, è stata necessaria una selezione ponderata degli strumenti da impiegare nel progetto. Le scelte tecnologiche sono state guidate da criteri specifici, quali la necessità di garantire efficienza, facilità di integrazione nei flussi di esecuzione rispettivi e compatibilità con i moduli esistenti.

In questo capitolo verranno discusse le motivazioni che hanno portato alla selezione delle tecnologie utilizzate, evidenziando come queste abbiano contribuito al raggiungimento degli obiettivi del progetto. Ogni sezione avrà le proprie metodologie specifiche che saranno illustrate prima della relativa scelta.

Dove possibile, per avvalorare ulteriormente le scelte si è deciso di aggiungere una valutazione utilizzando un modello derivato dal SQO-OSS Quality Model [10]. Quest'ultimo è un modello di valutazione della qualità del software open source, basato su alcune misurazioni. SQO-OSS (Software Quality Observatory for Open Source Software) è un'iniziativa volta a creare un framework di strumenti e metriche per valutare la qualità dei progetti open source in modo sistematico.

Questo modello si basa su vari parametri di qualità come il codice sorgente, la documentazione, il processo di sviluppo e la comunità di supporto. Utilizzando strumenti automatici, il modello misura questi aspetti per fornire una valutazione oggettiva e comparativa del software open source, facilitando la scelta di soluzioni di alta qualità. Ciò ha permesso di verificare che le tecnologie selezionate fossero adeguate non solo in termini di funzionalità, ma anche di sostenibilità ed affidabilità, garantendo una solida base per il sistema progettato. Chiaramente, è stato necessario adattarlo sia ad ogni caso specifico sia per permettere il confronto con tecnologie non propriamente open source come Apify o GPT.

Si utilizza una tabella rappresentante una valutazione qualitativa di criteri di analisi del software, che possono essere utili per confrontare diverse tecnologie o approcci in termini di:

- **Analyzability** (Analizzabilità): quanto è facile analizzare il codice. Più in dettaglio si considerano:
 - Cyclomatic Number, basato sul grafo del flusso di controllo di un programma, un grafo che rappresenta le diverse strade che il codice può percorrere in base alle condizioni e alle decisioni che contiene (ad esempio, le istruzioni if, while, for, switch, ecc.).
 - Numero di istruzioni e dimensione media, guardando puramente la lunghezza del codice.
 - Documentazione, ovvero se la libreria considerata ha o meno chiarificazioni sull'utilizzo e guide.
- **Changeability** (Cambiabilità): quanto è facile apportare modifiche al codice. In particolare, valutando:
 - Dimensione media delle istruzioni, intesa come facilità nel modificare le istruzioni nel codice.
 - Frequenza del vocabolario, considerata come la quantità di componenti che devi conoscere per usare lo strumento.
- **Stability** (Stabilità): quanto il software è stabile in termini di struttura interna. In questo caso si considera:
 - Numero di livelli nidificati e di nodi di ingresso/uscita, verificando quindi la profondità del codice.
- **Testability** (Testabilità): quanto è facile testare il codice. Prendendo in considerazione:
 - Numero di uscite da strutture condizionali e cicli, dove si misura la complessità del ciclo in ottica di testing
- **Pricing**: è stato necessario inserire un parametro che tenesse in conto del grado di gratuità della tecnologia dovendo paragonare anche elementi non totalmente open source.

Per tali metriche, un valore minore porta ad un giudizio più alto (“less is better”). Discorso diverso per la “Documentazione”, dove un valore più alto sarebbe più indicato. Per mantenere coerenza si è preferito quindi riportare il valore reciproco in modo tale da avere sempre il miglior giudizio per il numero minore (che però ha il denominatore più grande).

4.1 Scraping

Il primo modulo per cui si illustrano le scelte effettuate è lo scraping. In particolare si parte da criteri specifici concludendo poi col modello derivato dal più generale SQO-OSS.

4.1.1 Criteri iniziali

Il processo di valutazione degli strumenti di web scraping adottato si basa su diversi criteri, ciascuno dei quali contribuisce a determinare l'idoneità di uno strumento per un dato compito. Il primo criterio utilizzato per valutare tali strumenti è la facilità d'uso, dove si considerano essenzialmente tre elementi: Innanzitutto la curva di apprendimento, ovvero quanto effort (tempo e sforzo) è necessario al fine di imparare ad utilizzare lo strumento. Segue la disponibilità e qualità della documentazione a disposizione sul web (documentazione ufficiale). Infine, si quantifica il supporto della comunità, quindi la disponibilità di forum, tutorial, risorse online a disposizione sul web che possano essere d'aiuto agli utenti (W3CSchools, Stack Overflow, GeeksforGeeks e altri).

Il secondo criterio si basa sulla flessibilità e personalizzazione. Più specificatamente si sono considerati: la capacità di estrazione, quanto è flessibile lo strumento nel selezionare ed estrarre dati specifici (elementi HTML, selettori CSS e altri); il supporto per diversi formati, capacità di gestire vari formati di output (JSON, CSV, XLS, XML e altri); l'automazione, possibilità di automatizzare processi di scraping complessi.

Il terzo principio valuta le performance. La valutazione delle prestazioni di uno strumento di web scraping si basa su diversi fattori chiave. La velocità di scraping è fondamentale, misurando quanto rapidamente lo strumento può raccogliere dati utili dal web. L'efficienza delle risorse è altrettanto importante, includendo il consumo di CPU, memoria e batteria durante il processo di scraping. La scalabilità dello strumento determina la sua capacità di gestire grandi quantità di dati e numerose richieste simultanee.

Vi è poi l'aspetto sulle compatibilità e sul supporto tecnico. Difatti un valido strumento di web scraping deve supportare vari linguaggi di programmazione, offrendo librerie o API per linguaggi come Python, JavaScript e altri. La frequenza degli aggiornamenti e la risoluzione dei bug sono cruciali per garantire l'affidabilità e l'efficienza dello strumento nel tempo.

Fondamentale anche la presenza di funzionalità avanzate come la capacità di gestire CAPTCHA, per superare le barriere che possono impedire l'accesso ai dati. Lo strumento dovrebbe anche essere in grado di rilevare e gestire pop-up e altri elementi interattivi che potrebbero ostacolare il processo di scraping. La robustezza

nella gestione degli errori e la capacità di affrontare pagine non disponibili sono essenziali per un'operazione di scraping senza interruzioni.

Procedendo, si è valutata la sicurezza e la conformità. In particolare, bisogna rispettare la conformità legale, aspetto cruciale poiché lo strumento deve rispettare le leggi e i regolamenti sul web scraping e sulla privacy dei dati, oltreché l'anonimizzazione, attraverso l'uso di proxy e altre tecniche, che è importante per proteggere l'identità dell'utente durante l'operazione di scraping.

Infine, il costo dello strumento è un fattore decisivo. È importante considerare se lo strumento è disponibile gratuitamente, a pagamento unico o se offre un modello di pricing basato su abbonamento. Il rapporto qualità-prezzo valuta le funzionalità offerte in relazione al costo dello strumento, determinando la sua convenienza economica.

Concludendo sul processo di valutazione, l'aspetto legale è molto importante: difatti lo scraping è regolamentato da vari enti e leggi che possono anche differire a seconda della zona geografica (per l'UE è normato dal GDPR). Questo fa sì che il suo utilizzo sia libero per scopi di ricerca ma molto limitato per intenti economici.

Inoltre, compagnie di rilievo con molta disponibilità economica e computazionale adottano diversi metodi per bloccare o, comunque, limitare fortemente lo scraping sui loro siti (generalmente le Big Corporation come Amazon, Tripadvisor e simili).

Dopo un attento studio considerando i precedenti criteri, si è optato per due diverse soluzioni scegliendo una o l'altra a seconda che la fonte su cui fare scraping fosse o meno limitata. Di seguito saranno spiegate in modo esaustivo, non prima di aver ulteriormente avvalorato la scelta delle librerie tramite il modello SQO-OSS. Ad ogni modo si userà BeautifulSoup e Selenium in contesti più semplici, sennò ci si affiderà ad Apify.

4.1.2 Valutazione SQO-OSS

Si sono confrontate le due librerie BeautifulSoup e Scrapy con la piattaforma Apify, mentre non si è considerato Selenium in quanto, quest'ultimo, è un tool di supporto che può essere integrato indipendentemente da quanto scelto come tecnologia.

Nella tabella 4.1 sono illustrati i risultati della valutazione:

- **Analyzability:** dai valori ottenuti, è possibile dedurre che Scrapy risulti più complesso da imparare e utilizzare rispetto agli altri strumenti. Questo è dovuto alla sua struttura a classi, che lo rende più articolato; nonostante ciò, insieme ad Apify, presenta una migliore documentazione di BeautifulSoup.
- **Changeability:** riguardo la possibilità di modificare il codice, Scrapy risulta migliore rispetto ai competitor nonostante la sua complessità. Grazie alla sua modularità è possibile selezionare solo una determinata parte e cambiarla lasciando più o meno intatto il resto del codice. Ad ogni modo proprio la

presenza di tali classi fa sì che la conoscenza del vocabolario necessaria sia maggiore per Scrapy.

- **Stability:** BeautifulSoup ed Apify sono più semplici da implementare rispetto Scrapy dove la presenza di classi e quindi di una struttura gerarchica aumenta la profondità del codice (in termini di livelli nidificati).
- **Testability:** analogamente al caso precedente, il testing su BeautifulSoup ed Apify risulta più immediato.
- **Pricing:** riguardo il pricing, è evidente come Apify sia l'unico ad offrire API a pagamento al contrario dei primi due, totalmente gratuiti.

| Metrica | Criterio | Beautiful-Soup | Scrapy | Apify | Scala |
|---------------|---|----------------|---------------|---------------|----------------|
| Analyzability | Cyclomatic Number | 5 | 8 | 3 | Less is better |
| | Numero di istruzioni e dimensione media | 4 | 6 | 4 | Less is better |
| | Documentazione | $\frac{1}{3}$ | $\frac{1}{5}$ | $\frac{1}{5}$ | Less is better |
| Changeability | Dimensione media delle istruzioni | 7 | 5 | 6 | Less is better |
| | Frequenza del vocabolario | 6 | 7 | 7 | Less is better |
| Stability | Numero di livelli nidificati e di nodi di ingresso/uscita | 3 | 6 | 4 | Less is better |
| Testability | Numero di uscite da strutture condizionali e cicli | 3 | 6 | 4 | Less is better |
| Pricing | | 0 | 0 | 5 | Less is better |

Tabella 4.1: Matrice per la valutazione di tecnologie scraping

Ciò avvalorava la scelta finale di basarsi su BeautifulSoup ed Apify. Nel grafico a ragnatela sottostante 4.1 si può verificare ancora una volta quanto detto. Dove un'area minore è preferibile ad una maggiore.

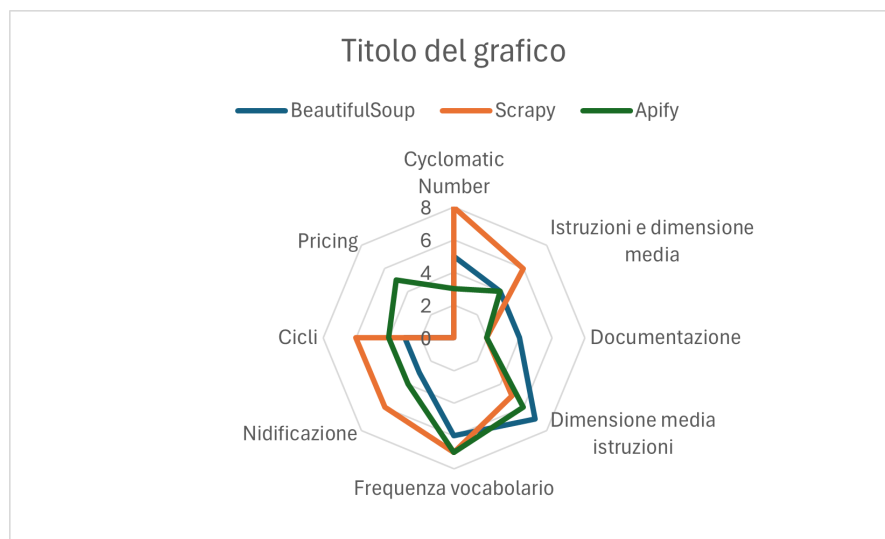


Figura 4.1: Diagramma relativo al modello SQO-OSS utilizzato per la valutazione delle scelte tecnologiche per lo scraping

4.1.3 Scelta effettuata

Dopo le varie valutazioni, si illustreranno in dettaglio le due soluzioni.

La prima soluzione consiste nella combinazione tra BeautifulSoup e Selenium. Il primo tool è stato selezionato in quanto permette di estrarre facilmente dati da siti ben strutturati e senza troppe limitazioni commerciali. Nonostante sia necessario ideare uno scraper per sito analizzato, si presta molto bene in quanto molto personalizzabile a seconda delle esigenze necessarie. Riguardo Selenium, è stato selezionato in quanto permette di interagire con le pagine web dinamiche che utilizzano JavaScript per caricare contenuti. Più in dettaglio, permette di navigare nelle varie pagine web, andando ad emularne l'interazione (click, scroll), fornendo così la possibilità di aggirare alcune limitazioni (cookies, avvisi della privacy, pop-up pubblicitari).

La combinazione di Beautiful Soup e Selenium risulta estremamente efficace per il web scraping. BeautifulSoup è ideale per gestire siti ben strutturati, permettendo un'estrazione dei dati semplice e personalizzabile. Selenium, invece, offre la capacità di interagire con contenuti dinamici, navigando e interagendo con le pagine come farebbe un utente reale. Insieme, queste tecnologie coprono un ampio spettro di esigenze, permettendo di superare limitazioni tecniche e normative dei siti web.

Questa sinergia consente di creare scraper robusti e flessibili, capaci di adattarsi a diversi scenari e requisiti, garantendo un processo di raccolta dati completo ed efficiente.

Sebbene la combinazione di queste due tecnologie sia potente e versatile, come

anticipato precedentemente, sono presenti delle limitazioni significative quando si tratta di siti commerciali con restrizioni più severe. Questi siti spesso implementano misure di sicurezza avanzate, come CAPTCHA, rilevamento di bot, limitazioni di frequenza di accesso e complesse strutture dinamiche per impedire l'estrazione automatizzata dei dati.

BeautifulSoup, pur essendo efficace per siti ben strutturati, non è progettato per aggirare queste misure di sicurezza. Selenium, pur potendo emulare interazioni umane, può essere facilmente rilevato e bloccato dai sistemi di protezione dei siti commerciali. Inoltre, l'uso di Selenium può essere limitato dalle risorse hardware, in quanto l'automazione del browser richiede un consumo significativo di CPU e memoria.

Pertanto, per siti commerciali con restrizioni elevate, si è preferito considerare una seconda soluzione più avanzata e specifica, seppur a pagamento: gli scraper messi a disposizione su Apify.

Quest'ultimo è stato selezionato per sopperire alle limitazioni delle tecnologie sopracitate; dunque, il suo utilizzo sarà relegato esclusivamente allo scraping su quelle fonti web di cui risulta difficile estrarre le informazioni tramite gli strumenti open source. Offre una vasta gamma di funzionalità aggiuntive che permettono di ottimizzare il processo.

La facilità d'uso e la flessibilità offerta da Apify lo rendono una scelta ideale per le operazioni di scraping di siti commerciali come, ad esempio, TripAdvisor o gruppi Facebook pubblici, dove è cruciale ottenere dati accurati e aggiornati in modo efficiente e affidabile, senza incorrere in limitazioni tecniche e/o commerciali.

4.2 Preprocessing

Per il secondo modulo, il processo di valutazione è stato modificato rispetto al modulo precedente. Difatti non si dovevano confrontare diverse tecnologie quali librerie da utilizzare, bensì definire quali tecniche fossero rilevanti per l'efficienza del processo e l'accuratezza dei risultati finali, metriche condivise con gli steps precedenti e successivi.

Si è inoltre omessa la valutazione tramite modello SQO-OSS.

4.2.1 Valutazione delle scelte

Si è innanzitutto optato per operazioni basilari ma essenziali, come la standardizzazione dei dati e la rimozione di eventuali valori mancanti, assicurandosi che i dati fossero coerenti e pronti per le fasi successive della pipeline.

Tuttavia, si è deciso di omettere tecniche più avanzate di trasformazione del linguaggio naturale, come stemming e lemmatizzazione. Tale scelta, presa dopo lo studio delle fasi successive (analisi e generazione), è motivata dal fatto che,

come vedremo, i moduli seguenti si baseranno su architetture neurali avanzate, come BART e GPT, che gestiscono internamente e con metodi più efficienti la comprensione ed il trattamento del linguaggio naturale senza richiedere pre-elaborazioni.

4.3 Clustering e Tagging

Analogamente al modulo precedente, la valutazione per il modulo di analisi risulta essere più semplice ed intuitiva rispetto al modulo di Scraping.

4.3.1 Valutazione delle scelte

In virtù delle possibilità studiate, si è ritenuto opportuno combinare il clustering con l'utilizzo del modello BART, utilizzando uno o l'altro a seconda del contesto.

Innanzitutto, si effettuerà un algoritmo di clustering per ottenere dalle coordinate geografiche l'appartenenza di un'attività ad una suddetta regione geografica. In particolare, dopo aver eseguito tale algoritmo ed aver formato i gruppi, si valuta la città principale o regione che lo racchiude e si aggiunge tale informazione ad ogni attività.

Per i descrittori tag primario e secondario oltreché per il periodo, si utilizza il modello NLP BART previa formazione degli insiemi delle labels. Quest'ultima attività, scontata per le quattro stagioni dell'anno, ha richiesto un'osservazione delle categorie già definite sul sito dell'azienda partner in modo da essere allineati con questi ultimi.

La valutazione per la scelta ottimale dell'algoritmo di clustering da utilizzare, invece, è stata sostanzialmente basata sull'unione di due criteri:

- **Efficienza**, si è cercato un algoritmo che non richiedesse un elevato consumo di tempo e risorse.
- **Accuratezza**, la ricerca di ottimizzazione non doveva sacrificare la correttezza dei risultati.

L'algoritmo che meglio rispondeva a tale connubio si è rilevato essere il K-Means.

4.4 Copywriting generativo

Anche in questo caso, come nel modulo precedente, i criteri valutati per selezionare la tecnologia più adatta sono stati efficienza ed accuratezza dell'output. In particolare, si è considerata l'unione dei due criteri.

Per una maggiore chiarezza, come nel caso dello scraping, si è utilizzato infine il modello SQO-OSS.

4.4.1 Valutazione delle scelte

Fin da subito, è stata evidente la superiorità di GPT: nonostante sia necessario utilizzare delle chiamate API per interfacciarsi col modello di OpenAI, i risultati sono molto migliori e si ha anche un vantaggio in termini di prestazioni non essendo obbligatorio un training come invece per una rete RNN (anche nel caso in cui si volesse effettuare il fine-tuning, quest'ultimo richiede qualche ora ed è a carico computazionale di OpenAI, richiedendo solo l'invio di un dataset).

A questo punto si sono confrontate due diverse possibilità mosse dal fine-tuning: quest'ultimo è il collo di bottiglia di tale modulo essendo la parte più lenta e quindi dispendiosa. Inoltre, rappresenta un'operazione da eseguire più volte nel tempo; infatti, aggiornare il dataset delle attività vorrebbe dire rieseguire il fine-tuning in modo da considerare anche le nuove informazioni.

Quindi scegliendo come modello base GPT-4 che, nonostante abbia un costo maggiore, porta vantaggi in termini di prestazione e tempistici, si è valutato:

- **Modello fine-tunato:** scelto il suddetto modello, quest'ultimo viene riallenato considerando il dataset passato. Tale dataset viene formattato in modo da avere l'elenco delle attività e per ognuna dei campi associati come descrizione, prezzo o durata, tutto in formato JSON. Una volta finito tale allenamento, il modello è pronto per l'uso.

Pro:

- Il modello riceve le attività un'unica volta prima del fine-tuning. Nell'interrogazione è possibile indicare solo il nome identificativo.
- Prompt di interrogazione ristretto il che è vantaggioso dato che la sua lunghezza incide sul costo della chiamata API.

Contro:

- È necessario fornire il dataset contenente le attività per intero, inoltre se quest'ultimo viene aggiornato bisogna rieseguire il fine-tuning.
- Il fine-tuning è comunque un'attività dispendiosa in termini economici e soprattutto temporali.

- **Modello base con prompt engineering:** si utilizza il modello base senza effettuare l'allenamento, l'enfasi va sul prompt passato per l'interrogazione. Si passano solamente le attività da combinare nell'esperienza formattate come descritto precedentemente, dopodiché si descrive dettagliatamente cosa ci si aspetta che il modello faccia e cosa non, in modo chiaro.

Pro:

- Basta fornire solamente le attività su cui si basa l'interrogazione e non più l'intero insieme.
- Non è più necessario attendere che il modello finisca il fine-tuning ma può subito essere utilizzato per le interrogazioni.

Contro:

- Ad ogni interrogazione bisogna fornire le attività da combinare con le varie informazioni oltre al comportamento che il modello deve adottare. Ciò rende parecchio ingombrante il prompt.

4.4.2 Valutazione SQO-OSS

Quanto detto può essere poi comprovato ulteriormente nella tabella 4.2 (dove si considera per il modello RNN una rete definita tramite la libreria tensorflow) grazie al modello che si era definito per lo scraping:

- **Analyzability:** GPT mette a disposizione le proprie API, mentre per utilizzare una RNN bisogna scendere a più basso livello aumentando il grado di difficoltà. Specificatamente a GPT, il caso con fine-tuning è più complesso in quanto aggiunge un task.
- **Changeability:** la rete RNN presenterà diversi livelli di gerarchia (layers che compongono la rete) che potrebbero dover essere adattati al caso specifico, mentre per GPT tale livello è gestito internamente.
- **Stability:** la profondità nel caso di una rete RNN è maggiore rispetto ad una GPT per la diversa gestione dei fornitori rendendo più semplice l'implementazione di questi ultimi.
- **Testability:** analogamente al caso precedente, il testing su GPT risulta più immediato.
- **Pricing:** unico punto a favore di una RNN in quanto totalmente gratuita, a differenza dei modelli GPT.

Per ricapitolare, valutando pregi e difetti dei due modelli, ci si è resi conto che la seconda opzione è più vantaggiosa: nonostante la singola interrogazione risulti più costosa a causa dell'allungamento della chiamata, il costo è pressoché bilanciato dalla mancata esecuzione del fine-tuning. Vi sono inoltre vantaggi in termini temporali e si è notato anche un incremento di realismo nelle risposte ottenute.

Anche per questa casistica si riporta il grafico a ragnatela nella figura 4.2.

| Metrica | Criterio | RNN | GPT FT | GPT P.E. | Scala |
|---------------|---|---------------|---------------|---------------|----------------|
| Analyzability | Cyclomatic Number | 7 | 4 | 2 | Less is better |
| | Numero di istruzioni e dimensione media | 6 | 4 | 2 | Less is better |
| | Documentazione | $\frac{1}{5}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | Less is better |
| Changeability | Dimensione media delle istruzioni | 7 | 4 | 2 | Less is better |
| | Frequenza del vocabolario | 7 | 4 | 2 | Less is better |
| Stability | Numero di livelli nidificati e di nodi di ingresso/uscita | 6 | 3 | 3 | Less is better |
| Testability | Numero di uscite da strutture condizionali e cicli | 5 | 3 | 3 | Less is better |
| Pricing | | 0 | 5 | 4 | Less is better |

Tabella 4.2: Matrice per la valutazione di tecnologie copywriting, dove per GPT FT si intende la versione con fine-tuning, mentre con P.E. quella con prompt engineering

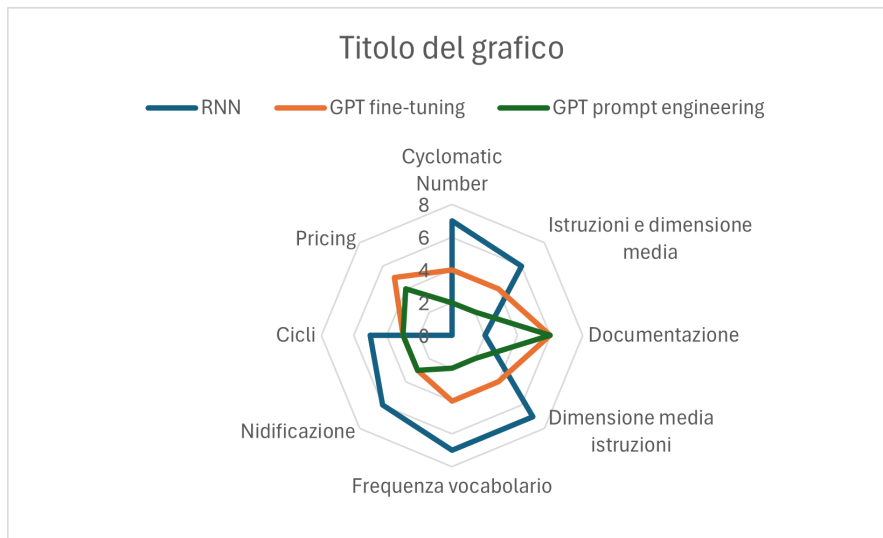


Figura 4.2: Diagramma relativo al modello SQO-OSS utilizzato per la valutazione delle scelte tecnologiche per il copywriting

4.5 Portale web

Molto differenti le considerazioni effettuate per la decisione su come implementare il portale web. Di fondamentale importanza era la scelta delle linee guida da seguire, su cui si è poi modellata la struttura dell'interfaccia.

4.5.1 Valutazione scelte

Le linee guida per le applicazioni web sono cruciali per garantire che queste siano efficienti, sicure, user-friendly e conformi agli standard del settore. Nell'ambito dell'HCI, queste linee guida sono fondamentali per progettare interfacce che non solo rispondano alle esigenze funzionali degli utenti, ma che offrano anche un'esperienza d'uso intuitiva e gratificante. Integrando principi di usabilità, accessibilità e design visuale coerente, l'HCI contribuisce a creare applicazioni web che promuovono l'interazione fluida e positiva tra l'utente e il sistema, migliorando così la soddisfazione e l'efficacia complessiva dell'esperienza utente. Nel contesto dell'HMAI, le linee guida sono fondamentali per progettare e sviluppare sistemi che rispettino le specifiche di cui sopra, integrandone l'intelligenza artificiale.

Queste linee guida mirano a garantire che l'interazione tra gli esseri umani e i sistemi basati su AI sia ottimizzata per la massima efficienza e soddisfazione dell'utente finale. L'interfaccia grafica per la web application del progetto TEIA verrà progettata cercando di rispettare il più possibile le linee guida e le best practice relative alla Human-AI Interaction (HMAI). Questo impegno garantirà che

l'interazione tra gli utenti e i sistemi basati sull'intelligenza artificiale sia ottimizzata per la massima efficienza, sicurezza e soddisfazione dell'utente finale.

Dopo attente ricerche si è scelto di adottare le seguenti linee guida:

- **PAIR Guidebook** [11]: Il PAIR Guidebook è un sito creato da Google nell'ambito del PAIR (People + AI Research) Initiative. È progettato per essere una risorsa educativa e pratica per designer e sviluppatori che lavorano con l'intelligenza artificiale (AI). Il sito fornisce linee guida, casi studio, strumenti e risorse per supportare la progettazione responsabile e l'implementazione etica dell'AI. Include anche approfondimenti su temi come l'interpretazione e l'interpretabilità dei modelli AI, l'etica dell'AI e l'impatto sociale delle tecnologie basate sull'AI.
- **Microsoft HAX Toolkit** [12]: Il Microsoft HAX Toolkit è una piattaforma fornita da Microsoft per supportare gli sviluppatori nella creazione di applicazioni di realtà mista e aumentata utilizzando HoloLens e altre tecnologie. Include strumenti, documentazione, esempi di codice e risorse per aiutare gli sviluppatori a comprendere, creare e distribuire esperienze di realtà aumentata e mista con le tecnologie Microsoft. La piattaforma mira a facilitare lo sviluppo di applicazioni innovative e immersive che sfruttano le capacità avanzate di HoloLens e altre piattaforme Microsoft.

In questo modo ci si assicurerà che l'interfaccia sia intuitiva, trasparente, accessibile e conforme agli standard di sicurezza e privacy. L'obiettivo è creare un sistema che non solo risponda alle esigenze degli utenti, ma che lo faccia in modo piacevole e intuitivo, contribuendo a un'esperienza utente positiva e coinvolgente.

Proseguendo su questa linea, il team di progetto lavorerà alla realizzazione di un prototipo ad alta fedeltà che integri questi principi, assicurando che ogni decisione sia guidata dalle migliori pratiche in ambito HAAI. Questi ultimi non solo hanno orientato la progettazione iniziale, ma continueranno ad essere un riferimento per eventuali miglioramenti futuri, come l'introduzione di nuove funzionalità o l'adattamento dell'interfaccia a diversi contesti d'uso. La scalabilità delle linee guida selezionate le rende ideali per accompagnare il progetto in un percorso di crescita ed innovazione.

4.6 Altre tecnologie

Dopo aver effettuato le scelte per ogni modulo, è stato necessario definire le tecnologie ed i linguaggi di programmazione più adatti per l'implementazione. Non avendo alcun punto di partenza, la prima problematica è stata quella di definire il linguaggio di programmazione. Considerando sia la semplicità d'utilizzo ma anche la grande disponibilità di librerie, la scelta è ricaduta su Python. Quest'ultimo

sarà anche utilizzato per implementare il server, mentre il frontend sarà scritto in JavaScript, in particolare sfruttando la libreria React.

4.6.1 Python

Python [13] è un linguaggio di programmazione ad alto livello, interpretato e orientato agli oggetti, noto per la sua sintassi semplice e leggibile. È ampiamente utilizzato in vari ambiti, tra cui sviluppo web, automazione, analisi dei dati, intelligenza artificiale e sviluppo di applicazioni. Python offre una vasta gamma di librerie e framework, che rendono facile la creazione di soluzioni complesse con meno codice rispetto ad altri linguaggi. Grazie alla sua versatilità e alla forte comunità di supporto, è una scelta popolare sia tra principianti che professionisti.

4.6.2 JavaScript

JavaScript [14] è un linguaggio di programmazione interpretato, utilizzato principalmente per sviluppare funzionalità dinamiche nelle pagine web. Inizialmente creato per il lato client, consente di interagire con il DOM (Document Object Model) per aggiornare il contenuto delle pagine web senza doverle ricaricare. È un linguaggio versatile, utilizzato anche nel backend (con Node.js) per sviluppare server e applicazioni complete. JavaScript supporta la programmazione orientata agli oggetti, funzionale e procedurale, ed è uno dei pilastri dello sviluppo web moderno insieme a HTML e CSS.

React

React [15] è una libreria JavaScript open-source sviluppata da Facebook per creare interfacce utente, in particolare per applicazioni web a pagina singola (SPA). Il suo approccio si basa sulla gestione di componenti, blocchi riutilizzabili di codice che rappresentano parti dell'interfaccia utente. React utilizza un "Virtual DOM" per aggiornare in modo efficiente la pagina web, applicando solo i cambiamenti necessari senza ricaricare l'intera pagina. La sua flessibilità e prestazioni lo rendono una delle librerie più popolari per lo sviluppo frontend moderno.

Capitolo 5

Implementazione

In questo capitolo si farà un'analisi dei vari moduli all'interno dell'applicativo, a partire dalla tecnologia scelta per poi trattarne l'implementazione ed il funzionamento logico.

Prima di procedere con l'implementazione vera e propria, si presenta, esemplificato, il funzionamento del sistema che riprende i due flussi di esecuzione precedentemente illustrati:

- **Flusso 1: Scraping, Preprocessing e Clustering/Tagging.** In questo primo flusso, i dati turistici vengono inizialmente raccolti dai siti di riferimento prescelti attraverso il modulo di Scraping. Quest'ultimo produce un JSON grezzo, ovvero contenente le informazioni delle attività così come possono essere ottenute senza alcuna pulizia o processamento. Questi dati vengono quindi passati al modulo di Preprocessing, dove vengono puliti e standardizzati per ottenere un JSON formattato. Successivamente, il modulo di Clustering e Tagging arricchisce questi dati, associando ogni attività ad un cluster specifico in base a criteri geografici e a tag descrittivi, generando infine un JSON arricchito. Questo flusso permette di strutturare e arricchire il dataset, pronto per essere utilizzato nel secondo flusso. Inoltre questo processo potrà essere effettuato ogni volta che si vorrà aggiornare il database con altre attività.
- **Flusso 2: Copywriting.** Il secondo flusso sfrutta i dati strutturati e arricchiti prodotti nel Flusso 1. L'utente può selezionare manualmente le attività da includere, oppure il sistema può selezionarle automaticamente per poi generare nuove attività turistiche attraverso il modulo di Copywriting. Indipendentemente dalla modalità con la quale generare l'attività, sarà necessario interagire con tale modulo che prende i dati precedenti come input per creare contenuti personalizzati e dettagliati, risultando in una attività generata pronta per la fruizione finale.

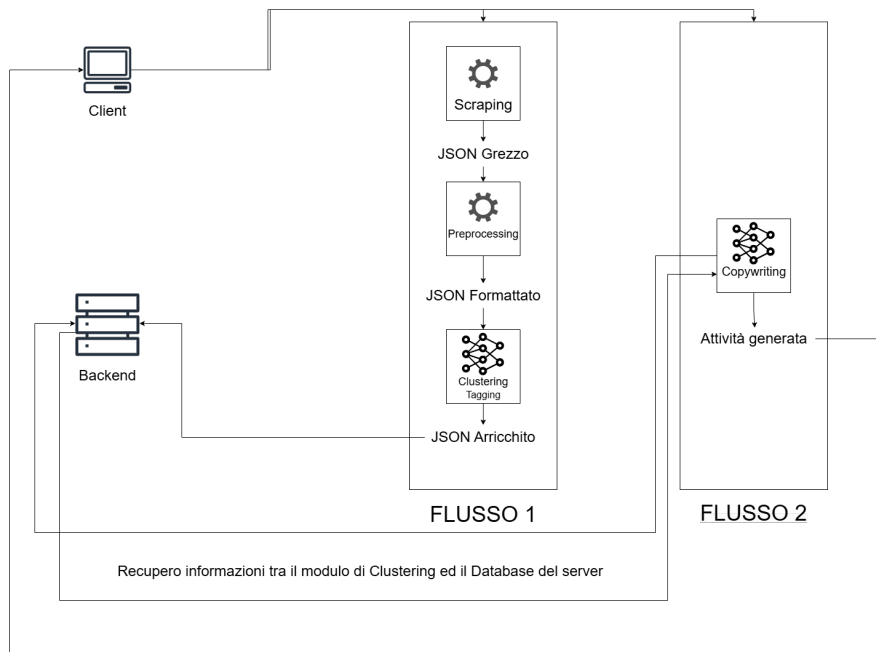


Figura 5.1: Diagramma illustrante il funzionamento del sistema

Quanto detto finora può essere ulteriormente visionato nel diagramma in figura 5.1.

5.1 Scraping

Lo scraping ha svolto un ruolo essenziale nel popolamento del database su cui effettuare le analisi e le future interrogazioni. Come già stabilito nel capitolo precedente, si è deciso di affidarsi a due differenti soluzioni a seconda delle fonti sulle quali operare.

5.1.1 BeautifulSoup e Selenium

Le librerie utilizzate per la prima soluzione del modulo corrente sono *Selenium* e *BeautifulSoup* che, come precedentemente spiegato, si occupano di navigazione su browser automatica e scraping. L'obiettivo consisterà nell'ottenere un elenco delle attività con le principali informazioni, possibilmente in formato JSON, partendo da una fonte determinata a priori.

Analizzando i punti principali si ha:

- **Inizializzazione e Configurazione:** dopo aver importato le librerie necessarie, viene avviato il browser Chrome tramite Selenium. Si definiscono poi dei parametri:
 - Parametri di controllo per il ciclo, ovvero `start_parameter` e `max_parameter` che determinano da dove iniziare e dove fermarsi nel processo di scraping.
 - URL base, rappresentante l'URL della pagina da cui iniziare lo scraping, in questo caso "www.spain.info".
 - esperienze, una lista di esperienze dove accumulare tutte le esperienze turistiche estratte.
- **Navigazione:** viene avviata la navigazione all'interno del sito gestendo eventualmente anche i cookie. L'azione principale è quella di estrarre tutti i possibili links su cui si potranno poi trovare le attività.
- **Scraping e Salvataggio dei Dati:** il cuore del codice dove ne è contenuta la logica. Innanzitutto si naviga in tutti i links ottenuti gestendo eventuali cookie; dopodichè viene creato un dizionario vuoto, chiamato `esperienza`, per ogni esperienza turistica. Le informazioni come titolo, descrizione, prezzo, lingue disponibili, luogo, durata, ecc., vengono estratte dalla pagina e salvate nel dizionario.

Il dizionario `esperienza`, rappresentante la singola attività, viene aggiunto alla lista `esperienze`, che contiene tutte le esperienze estratte. Alla fine, tutti i dati raccolti vengono salvati in un file JSON. A questo punto, la scheda aperta per il link viene chiusa e il browser ritorna alla scheda principale per continuare il ciclo.

```
1 for link_element in links:
2     link = link_element.get_attribute('href')
3     driver.execute_script("window.open('');")
4     window_handles = driver.window_handles
5     driver.switch_to.window(window_handles[-1])
6     driver.get(link)
7
8     inner_cookie_banner = soup.find("div", class_="cookie-
notice")
9
10    if inner_cookie_banner:
11        try:
12            inner_rifiuta_tutti_button = WebDriverWait(
driver, 10).until(
13                EC.element_to_be_clickable((By.CLASS_NAME,
"cn-decline")))
14            )
```



```
15         inner_rifiuta_tutti_button.click()
16     except:
17         pass
18
19     soup = BeautifulSoup(driver.page_source, "html.parser"
20 )
21     esperienza = {
22         "titolo": "",
23         "descrizione": "",
24         "prezzo": "",
25         "luogo": "",
26         "durata": ""
27     }
28
29     title_experience = soup.find("h1", class_="titolo-
30 sencillo")
31
32     if title_experience:
33         esperienza["titolo"] = title_experience.text
34         p_next_to_h1 = title_experience.find_next_sibling(
35 "p", class_="text")
36         if p_next_to_h1:
37             description_experience = p_next_to_h1.text
38
39             esperienza["descrizione"] =
40 description_experience
41
42             wrapper_info = p_next_to_h1.find_next_sibling(
43 "div")
44             span_titles = wrapper_info.find_all("span",
45 class_="title")
46             span_texts = wrapper_info.find_all("span",
47 class_="text")
48
49             for title, text in zip(span_titles, span_texts
50 ):
51                 if title.text == "Prezzo":
52                     esperienza["prezzo"] = text.text
53                 else:
54                     print("Nessuna descrizione trovata")
55                 else:
56                     print("Nessun titolo trovato")
57
58     esperienze.append(esperienza)
```

```
56     driver.close()
57
58     driver.switch_to.window(window_handles[0])
59
60     start_parameter += 36
61
62
63 with open("scraping_spain_info.json", "w", encoding="utf-8")
64     as file:
65     json.dump(esperienze, file, ensure_ascii=False, indent=4)
66
```

Ricapitolando, questo script automatizza il processo di estrazione di dati da una serie di pagine web, raccogliendo informazioni sulle esperienze turistiche.

Output

Come accennato nel capitolo teorico di background e come vedremo meglio in seguito nel capitolo dedicato ai risultati, i dati raccolti tramite web scraping possono variare notevolmente a seconda della fonte da cui provengono e del metodo utilizzato. Questa variabilità può presentarsi sia nel formato dei dati sia nella struttura delle informazioni estratte. Affrontare questa complessità richiede l'adozione di pratiche standardizzate, finalizzate a semplificare la gestione e l'analisi dei dati. In tal senso, definire uno standard di output per i dati ottenuti tramite scraping si è rivelata una scelta strategica, poiché ha consentito una maggiore interoperabilità tra i diversi moduli dell'applicazione e una più agevole elaborazione successiva.

La prima scelta fondamentale è stata quella di utilizzare il formato JSON per codificare il dataset. Questo formato, largamente utilizzato in ambito di sviluppo software, offre vantaggi significativi, in particolare:

- **Struttura gerarchica:** I dati possono essere organizzati in una gerarchia che rispecchia le relazioni tra le diverse informazioni estratte dai siti.
- **Compatibilità:** La struttura JSON si integra perfettamente con le librerie Python, in particolare la libreria json, facilitando la manipolazione e il parsing dei dati.
- **Flessibilità:** JSON permette di rappresentare dati di diversa natura (stringhe, numeri, array, oggetti complessi) mantenendo una leggibilità elevata e una struttura chiara.

Ad ogni modo, si è incappati in un problema ricorrente durante l'attività di scraping riguardante le fonti che non strutturano le informazioni in maniera ordinata o prevedibile. Per alcuni siti, le informazioni sono contenute in formati più complessi o sparsi, come tabelle non omogenee o sezioni dinamiche caricate tramite JavaScript. In questi casi, estrarre le informazioni rilevanti e codificarle in JSON può risultare difficile. Attualmente, per queste fonti, i dati vengono raccolti in file di tipo .txt, che successivamente richiedono un intervento manuale o semi-automatizzato per essere uniformati.

Nel presente, le analisi saranno condotte sui file JSON. Per formare il dataset finale è quindi necessario far sì che gli elementi contengano almeno un set di informazioni necessarie per le analisi (luogo, descrizione, prezzo, durata, nome).

Sarà quindi fondamentale il compito del preprocessing, ovvero il modulo seguente: dopo aver ottenuto le informazioni dalle diverse fonti, questo modulo si occuperà di unirle in un unico file JSON contenenti quelle informazioni minime che verranno poi passate al modulo successivo.

5.1.2 Apify

La seconda soluzione si basa su Apify, che come già ampiamente detto, mette a disposizione un market di tool chiamati "Actors", creati da terze parti che si occupano di effettuare automaticamente lo scraping una volta impostati dei parametri di interesse. Questi strumenti si sono dimostrati efficaci nello scraping di siti di grandi compagnie, come TripAdvisor, che utilizzano meccanismi avanzati per bloccare o limitare fortemente tali operazioni.

Vi sono due modalità operative: la prima prevede l'esecuzione tramite GUI, attraverso il sito web. In questa modalità, è possibile impostare i vari filtri e avviare l'esecuzione direttamente dall'interfaccia. Il risultato verrà restituito e potrà essere scaricato sul proprio PC. La seconda modalità riguarda l'utilizzo delle API, specifiche per ogni attore, che consentono una maggiore integrazione con il proprio codice.

Attualmente si è scelto di utilizzare la prima opzione, più semplice ed immediata. Si lascia a studi futuri la possibilità di integrare tali chiamate nel modulo dello scraping.

In particolare, è possibile osservare come avviare l'esecuzione. Nella figura 5.2, i parametri vengono impostati manualmente modificando i campi corrispondenti, mentre nella figura 5.3, che mostra un'opzione a più basso livello, si fornisce direttamente un file JSON.

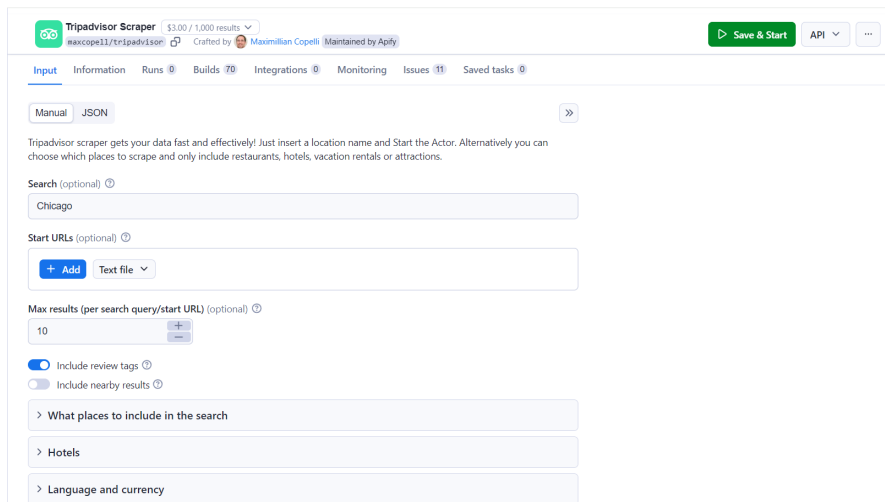


Figura 5.2: Scraping tramite Apify, i parametri sono impostati in modo manuale

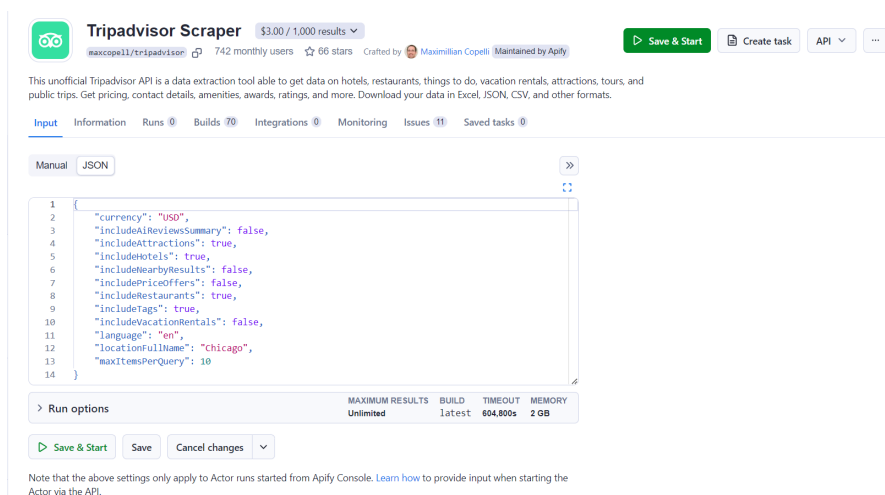


Figura 5.3: Scraping tramite Apify, i parametri sono impostati con un file JSON

Output

L'output ottenuto, in linea con quanto avviene per la prima soluzione, sarà in formato JSON. I campi sono ovviamente diversi da quelli ottenuti nella prima soluzione, l'insieme minimo è però presente: sarà compito del modulo successivo quello di uniformare le due soluzioni in un file unico selezionando solo le informazioni necessarie.

5.2 Preprocessing

Come già stabilito, le tecnologie si basano sempre sul linguaggio Python; per questo modulo in particolare si utilizzeranno le librerie *json* e *csv*. Infatti, l'obiettivo di questo modulo sarà quello di uniformare i risultati dello scraping in un unico file JSON, con il quale si effettueranno le varie analisi, ed affiancargli un file Excel da utilizzare come backup o alternativa nel caso in cui alcune operazioni non possano essere eseguite direttamente sul file JSON. In seguito, si analizzeranno le caratteristiche principali:

- Inizializzazione: innanzitutto è necessario caricare il file JSON per poterlo poi utilizzare, dopo aver importato le librerie.
- Si sono definiti poi i campi di interesse, sottoinsieme degli originali, da cui saranno filtrati.

Nella versione attuale i campi scelti risultano essere:

- Id,
- Nome,
- Descrizione,
- Luogo,
- Latitudine e longitudine,
- Durata,
- Prezzo

- Seguendo, vengono definite delle funzioni ognuna con il proprio scopo.

```

1 def filter_function(obj):
2     filtered_obj = {campo: obj.get(campo) for campo in fields
3     if campo in obj}
4     filtered_obj['city'] = obj.get('addressObj').get('city')
5     return filtered_obj
6
7 def filter_rating(obj):
8     if obj["rating"] >= 2.5:
9         return obj
10
11 def for_excel(data):
12     vector = []
13     for d in data:
14         ob = (d["description"], d["name"], d["latitude"], d["
15             longitude"], d["city"])
16         if ob[0] != "":
17             vector.append(ob)

```

```

16     return vector
17
18 def create_excel(data):
19     csv_file = "main4.csv"
20     with open(csv_file, 'w', newline='', encoding='utf-8') as
21         f:
22         writer = csv.writer(f)
23         writer.writerow(['Desc', 'title', 'latitude', 'longitude',
24             'City'])
25         writer.writerows(data)
26     print("Data has been written to", csv_file)

```

- *Filter_function* gestisce l'opzione di filtraggio, azione principale del modulo di preprocessing necessaria ad uniformare gli elementi. Tale filtraggio avviene in base ai campi specificati precedentemente.
 - *Filter_rating* utilizzata per selezionare solo le attività con un rating superiore a 2.5 stelle su 5. In questo modo viene già effettuata una preselezione scartando attività scadenti.
 - *For_excel* crea il vettore delle attività con solamente i campi di interesse che saranno poi utilizzati per creare il file excel
 - *Create_excel* gestisce finalmente la creazione del file excel
- Funzionamento ed utilizzo: infine vi è l'esecuzione del modulo. In questo caso si considerano solo attività spagnole

```

1 for item in data:
2     try:
3         if item["addressObj"]["country"] == "Spain":
4             spain_data.append(item)
5             filtered_data = list(map(filter_function,
6                 spain_data))
7     except:
8         pass
9
10 print(filtered_data)
11 spain1 = for_excel(filtered_data)
12 create_excel(spain1)

```

Come si può vedere i file saranno salvati sia in formato Excel che in formato JSON in modo tale da utilizzare uno o l'altro in base al contesto.

È importante notare come nell'effettiva implementazione manchino elementi che erano stati definiti nella progettazione: si è preferito avere un insieme minimo di elementi per ogni modulo che permettesse un funzionamento dell'intera pipeline piuttosto che dedicarsi fin da subito all'intera funzionalità. Si prevede comunque l'integrazione di tali elementi negli studi futuri.

5.3 Clustering e tagging

In linea con i casi precedenti, la base sarà sempre Python. Cambiano però le librerie utilizzate, per il modulo in questione le principali utilizzate sono:

- **Sklearn** (Scikit-learn): È una libreria Python utilizzata per il machine learning. Fornisce strumenti efficienti per il data mining e l'analisi dei dati. Alcuni componenti principali includono algoritmi di classificazione, regressione, clustering (come KMeans per raggruppare dati simili), e preprocessing (come StandardScaler, che normalizza i dati per migliorarne la gestione nei modelli di machine learning).
- **Transformers**: È una libreria sviluppata da Hugging Face che facilita l'uso di modelli di deep learning, in particolare quelli basati su architetture transformer come BERT, GPT, T5, e altri. Viene usata per compiti come il natural language processing (NLP), la traduzione, la generazione di testo, e la sentiment analysis. Il metodo pipeline permette di applicare facilmente questi modelli a vari task senza configurazioni complesse.

Tornando al modulo, esso avrà essenzialmente due compiti principali: il clustering geografico ed il tagging. Il clustering geografico, responsabile dell'inserimento dell'attività in uno dei cluster predefiniti, organizza le attività in gruppi basati sulla loro posizione geografica. Questo approccio consente di creare raggruppamenti significativi e intuitivi per l'utente finale, facilitando la creazione di pacchetti turistici ottimizzati per specifiche aree, come città o regioni. Difatti attività vicine spazialmente saranno parte dello stesso cluster e, di conseguenza, saranno più adatte a formare un pacchetto di esperienze.

Il tagging, invece, aggiunge i tag primario e secondario, oltre al tag del periodo stagionale, per arricchire le informazioni di ciascuna attività.

In entrambi i casi, tali informazioni saranno inserite come campi aggiuntivi all'interno del file JSON.

È possibile quindi dividere il modulo corrente in due parti ed ognuno svolgerà un compito, analizzato più in dettaglio di seguito.

5.3.1 Clustering geografico

- Inizializzazione: in primis vengono caricate le attività dal file Excel, in seguito si eliminano dati nulli e si prendono le sole coordinate per la clusterizzazione. È necessario anche inizializzare uno scaler per standardizzare queste ultime; infine, si stabilisce il numero di clusters.
- Algoritmo: si inizializza poi l'algoritmo KMeans. In particolare, viene istanziato un oggetto che tramite il metodo "fit" analizza le coordinate inserendole in un cluster. Tale informazione viene poi aggiunta all'interno del dataset originale.

```

1 kmeans = KMeans(n_clusters=num_clusters)
2 kmeans.fit(coordinates_scaled)
3 cluster_labels = kmeans.labels_
4
5 data['Cluster'] = cluster_labels
6

```

- Output: Dopo aver eseguito l'algoritmo, è possibile visualizzarne i risultati tramite plot in un grafico 2D. Ogni elemento rappresenta il luogo di un'attività, esso espone oltre alle coordinate, anche il nome della località.

Di seguito, si riporta nella figura 5.4 un esempio di output basato su attività del sud della Spagna:

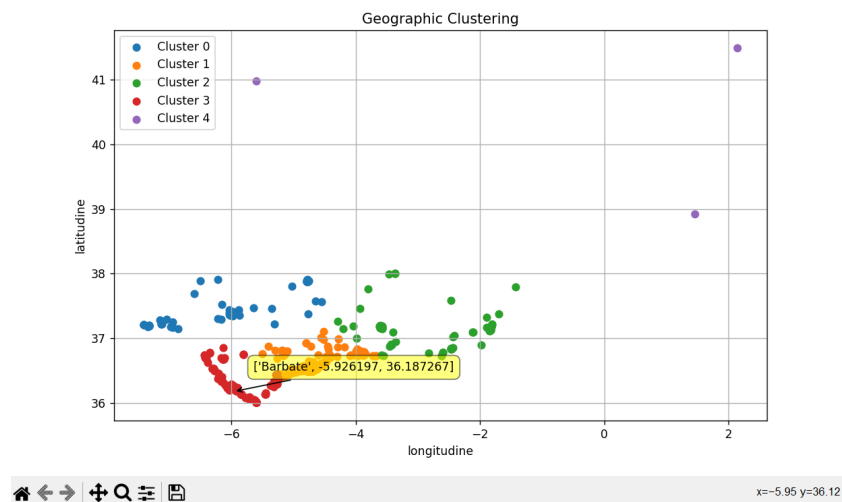


Figura 5.4: Output 2D raffigurante la divisione in cluster geografici delle attività in esame

5.3.2 Tagging

Svolto in sequenza sui tag e sul periodo dell'anno. Il tagging viene effettuato per arricchire le informazioni di ciascuna attività, aggiungendo informazioni sia sulla categoria che sulla stagionalità. Questo approccio consente di differenziare meglio le attività, permettendo una classificazione più dettagliata che facilita la ricerca e la creazione di pacchetti turistici personalizzati. L'aggiunta sequenziale di questi tag permette di includere informazioni chiave che migliorano la precisione nella selezione delle attività. Sarà quindi possibile associare le attività in base a specifici criteri selezionati, come ad esempio l'affinità tra le categorie o la stagionalità.

L'esecuzione è affidata al modello BART di hugging face: il principale punto di forza è che, essendo un modello già allenato con ottime prestazioni e risultati, è necessario solamente specificare un insieme di descrittori senza preoccuparsi di altro. Sarà poi il modello ad analizzare per ogni attività quanto ogni descrittore sia adatto a descriverlo, attribuendogli inoltre un punteggio. Una volta effettuata la suddetta analisi, verrà preso quello a maggiore affinità. In dettaglio:

- Inizializzazione: è necessario istanziare il modello che si occuperà del tagging ed anche in questa fase bisogna caricare i dati.

In particolare, il tagging si baserà sulle descrizioni testuali.

- Prima di procedere con il tagging, bisogna definire le classi di interesse. A seconda dell'analisi effettuata, tale insieme sarà composto da elementi diversi: ad esempio, nel caso della stagionalità le classi saranno le quattro stagioni (Inverno, Primavera, Estate, Autunno). Mentre nel caso del tag primario, ad esempio, ne vengono definite quindici basandosi sul sito di Cicero, tra cui "Hiking", "Gastronomia", "Foraging", "Cooking", "Degustazione", "Escursione", "Natura" ed altre.
- Utilizzo del modello: iterando su tutte le descrizioni presenti, viene applicato l'algoritmo su tutte le attività analizzando ogni elemento delle "labels". In particolare, si imposta l'opzione `multi_label` a `vero`: in questo caso, ogni label sarà considerata come a sé stante ed avrà punteggio tra 0 ed 1, mentre, nel caso fosse impostato a `falso`, allora la somma dei punteggi delle labels dovrà essere uguale ad 1, sconsigliato per un gran numero di labels. Ad ogni attività verrà quindi aggiunto un campo contenente le labels, ordinate in base al punteggio assegnato, ed il relativo punteggio. In particolare, per ciascuna attività si manterrà un numero limitato di etichette. Tale numero è attualmente impostato ad uno in modo tale da associare un'etichetta per ogni descrittore.

```
1 activities_labeled = []
2 for i, item in enumerate(data_json, start=1):
```

```
3     if (item['descrizione']):
4         result = classifier(item['descrizione'],
5                               candidate_labels, multi_label=True)
6
7         labeled_item = {
8             "id": i,
9             "sequence": item['descrizione'],
10            "labels": result['labels'],
11            "scores": result['scores']
12        }
13
14        activities_labeled.append(labeled_item)
```

Dopo aver applicato il modello per completare i vari compiti, si ottiene un file JSON contenente i descrittori geografico, di stagionalità e tag primario e secondario. Ciò fornirà un dataset di base da interrogare, eventualmente caricato su un server online.

5.4 Copywriting generativo

Rimanendo coerenti con i moduli precedenti, si continua ad utilizzare Python. L'unica libreria adottata, però, è OpenAI permettendo di interfacciarsi direttamente con uno dei modelli della suddetta organizzazione. In questo modo, il grosso del carico viene eliminato lasciando tutto a carico di OpenAI.

Avendo scelto di saltare il fine-tuning limitandosi al prompt engineering, l'unica difficoltà sarà proprio questa.

- **Inizializzazione:** si richiede di inizializzare un client che si incaricherà di inoltrare le richieste ad OpenAI e sostanzialmente di verificare che si disponga del credito necessario. Inoltre, come già detto, è necessario mandare la richiesta al modello prescelto. Come sarà spiegato in seguito, vi saranno due possibili richieste: la scelta da parte del modello di un sottoinsieme di attività da quello inviato, in base a criteri di affinità, o la combinazione delle attività al suo interno. Questa fase di scelta sarà effettuata da un utilizzatore tramite l'interfaccia.

In entrambi i casi l'elemento principale, presente in entrambe le richieste, è l'insieme delle attività con le relative informazioni che sarà mantenuto in un apposito vettore. I dati, che rappresentano le attività, sono stati precedentemente standardizzati durante la fase di preprocessing e quindi, come vedremo nel capitolo dei risultati tramite un esempio, con i campi di interesse scelti.

- **Prompt engineering:** è la pratica di progettare e ottimizzare i prompt, ovvero le istruzioni o i testi di input forniti a modelli di linguaggio AI come GPT, per ottenere risposte desiderate e utili. Questo processo implica la formulazione precisa e strategica del testo di input per influenzare il modo in cui il modello genera il suo output, massimizzando la qualità e la rilevanza delle risposte. Il prompt engineering è cruciale per migliorare l'efficacia dei modelli AI in una vasta gamma di applicazioni, dalla generazione di testo alla risoluzione di problemi complessi.

Durante l'implementazione, grazie all'utilizzo del prompt engineering, si è palesata una possibile funzionalità aggiuntiva: la generazione di un'unica attività partendo da un insieme specifico prevede che la scelta venga effettuata comunque da un operatore umano. È però possibile far effettuare tale scelta sempre alla rete. Nel caso si volesse utilizzare questa variante, vi sarà una prima chiamata API che effettui la scelta cui seguirà la chiamata di generazione vera e propria. Il prompt utilizzato per la fase della scelta è il seguente:

```
1 prompt = (  
2     "Dalle seguenti attività, seleziona un sottoinsieme di  
3     massimo 3 attività che siano fortemente correlate tra loro  
4     in termini di descrizione, tipo o altre caratteristiche  
5     rilevanti. Le attività selezionate devono essere  
6     complementari, tali da poter essere proposte come parte di  
7     un pacchetto combinato coerente. Analizza le seguenti  
8     caratteristiche per identificare l'affinità. "  
9     "1. Tipologia di attività  
10    2. Parole chiave nella descrizione  
11    3. Durata e costo simili  
12    4. Target di utenti simili (ad esempio, numero di persone  
13    o fascia d'età)  
14    5. Contesto o location"  
15    "Non aggiungere dettagli esterni alle attività e ritorna  
16    solo quelle selezionate in un JSON. Il JSON deve contenere  
17    un vettore chiamato 'selected_activities' nel medesimo  
18    formato dei dati ricevuti. Il risultato deve essere preciso  
19    e rispettare il limite di 3 attività affini:\n"  
20    + "\n".join(json.dumps(card, ensure_ascii=False) for card  
21    in data['cards'])  
22 )
```

Di seguito il prompt utilizzato invece per la combinazione finale (nel caso in cui si volessero scegliere manualmente le attività da combinare, la chiamata API precedente è omessa):

```
1 prompt = (  
2     "Combina le seguenti attività in una singola attività.  
   Mantieni tutte le informazioni fornite e non aggiungere  
   dettagli esterni. Genera una nuova descrizione che includa  
   gli elementi salienti di tutte le descrizioni originali. "  
3     "Il prezzo e la durata devono essere ottenuti come la  
   somma dei prezzi e durate forniti nelle rispettive attività  
   , inoltre forma un nuovo titolo dalla combinazione dei  
   precedenti e ritorna il tutto sotto forma di un json  
   formattato come i dati ricevuti e ritorna due immagini in  
   due campi distinti di nome imgUrl1 ed imgUrl2 :\n"  
4     + "\n".join(json.dumps(card, ensure_ascii=False) for card  
   in data['cards'])  
5 )  
6
```

In entrambi i casi, come anticipato, si includono le attività direttamente nel prompt.

- **Chiamata API:** successivamente si utilizza il client per mandare la richiesta ad OpenAI. I parametri passati saranno il modello prescelto (in questo caso gpt-4) ed il prompt precedentemente scritto. Dopo aver ottenuto la risposta, è possibile trovare quest'ultima all'interno di content. Si vedrà nella sezione dei risultati un possibile output.

5.5 Portale web

Affrontiamo adesso l'implementazione del portale web che, come già anticipato in precedenza, sarà necessario a racchiudere i vari moduli ed a permettere l'interazione tra questi ultimi ed un utilizzatore.

Tale portale risulterà logicamente diviso in due sezioni che seguono le due fasi definite:

- **Scraping e analisi:** svolto da una sezione di backoffice dove, inserendo la regione o comunque un'indicazione geografica, saranno eseguiti i suddetti compiti creando infine un database dove salvare tali informazioni.
- **Generazione attività:** sezione principale del sito nella quale interrogando il database si otterranno un'insieme di attività con la quale generare un'esperienza unica.

In particolare verrà analizzata la seconda sezione, mentre la prima è lasciata a sviluppi futuri.

Ad ogni modo il portale è stato sviluppato utilizzando il framework React per il frontend, insieme a Flask per la gestione del backend e la comunicazione con il database. I linguaggi utilizzati sono quindi JavaScript per il frontend e Python per il backend. In questo modo si ha continuità con i moduli della pipeline semplificando la gestione delle chiamate API ai vari componenti di IA. In dettaglio, saranno illustrate le *routes* presenti con i principali componenti ed infine, lato server, le varie API implementate. Inoltre, l'implementazione del prototipo ha seguito il design definito nel documento Figma, assicurando coerenza tra la progettazione del prototipo ed il suo sviluppo.

5.5.1 Gestione Routes

In un'applicazione web basata su React, le routes (rotte) sono utilizzate per gestire la navigazione tra diverse viste o pagine dell'interfaccia utente senza ricaricare l'intera pagina. Questo viene gestito tramite React Router, che consente il routing client-side. In questa sezione vengono descritte le principali rotte utilizzate nel portale web, insieme alla loro funzione.

- **Route principale** (“/”): la rotta principale serve come punto di accesso all'applicazione. Quando l'utente visita l'URL base dell'applicazione, viene reindirizzato alla schermata iniziale. Questa pagina include un componente di selezione delle attività basato su SelectorComponent, che permette agli utenti di filtrare e cercare attività specifiche in base alla posizione, al periodo dell'anno ed ai tag. Avviando la ricerca si passa alla route seguente.
- **Route delle attività** (“/activities”): questa rotta visualizza una lista di attività selezionate dall'utente tramite la pagina principale. Il componente ContentView visualizza le attività come card scrollabili, dove l'utente può cliccarne una specifica per una visualizzazione dettagliata. È poi possibile selezionarne più di una per far generare l'esperienza, eventualmente filtrando maggiormente il sottoinsieme ottenuto tramite gli *sliders* sulla sinistra. In alternativa, è possibile far effettuare la scelta delle attività da combinare al modello stesso. In entrambi i casi, avviando la generazione si arriverà alla route sottostante.
- **Route della generazione** (“/results”): tale route mostra il risultato del processo generativo basato sulle attività selezionate. Vi è il componente GenerativeViewer che, ottenute le attività selezionate, genera il contenuto personalizzato basato su di esse. Sarà possibile, inoltre rigenerare l'attività specificando ulteriori richieste tramite un form, o salvare l'attività creata.

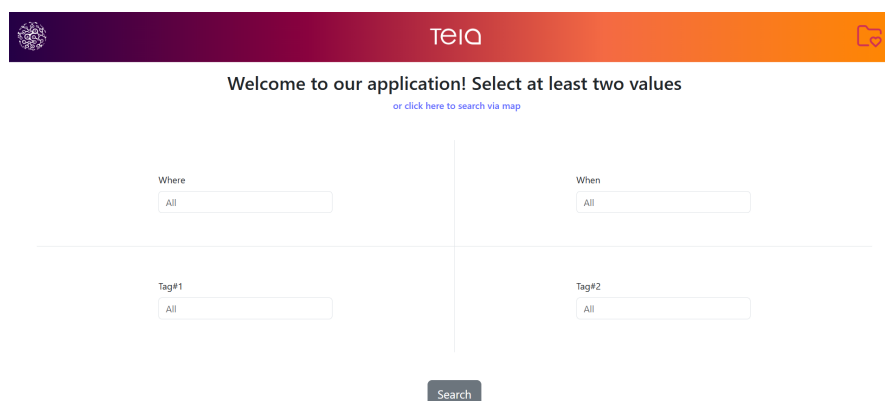
- **Route delle attività personali** (“/myactivities”): questa rotta consente agli utenti di visualizzare le attività personali che sono state generate e successivamente salvate. Il componente PersonalContent gestisce la visualizzazione delle attività specifiche per ogni utente.
- **Route mappa** (“/map”): visualizza una mappa che rappresenta la posizione delle attività nella regione di interesse e rappresenta un’alternativa alla ricerca tramite selettori: infatti, anzichè inserire uno o più degli elementi richiesti, è possibile tracciare l’area di interesse su cui poi visualizzare le attività. Questo permette all’utente di visualizzare graficamente dove si trovano le attività selezionate. Come per la route principale, dopo la ricerca si arriverà nella route delle attività.
- **Route di fallback** (“*”): questa rotta gestisce tutti gli URL che non corrispondono a quelli definiti. Quando un utente tenta di accedere a una pagina non esistente, viene visualizzato un messaggio di errore.

È inoltre sempre presente l’*header* del portale tramite cui è possibile una navigazione più veloce nelle varie sezioni del sito, tra cui le attività personali, o il ritorno alla schermata precedente.

5.5.2 Componenti Frontend

Proseguiamo con una descrizione dei principali componenti illustrandone scopo e funzionamento.

Selector Component



The screenshot shows a web application interface. At the top, there is a dark purple header with a logo on the left, the text 'TEIQ' in the center, and a red icon on the right. Below the header, a message reads 'Welcome to our application! Select at least two values' with a link 'or click here to search via map'. The main content area is divided into four quadrants by a vertical and a horizontal line. Each quadrant contains a selector: 'Where' (top-left), 'When' (top-right), 'Tag#1' (bottom-left), and 'Tag#2' (bottom-right). Each selector has a dropdown menu with 'All' selected. At the bottom center, there is a 'Search' button.

Figura 5.5: Schermata Home, in cui è possibile inserire i selettori

Il componente Selector (5.5) ha il compito di fornire un'interfaccia utente per filtrare le attività in base a diversi criteri come la posizione (where), il periodo dell'anno (when) e i tag (tag1, tag2). In particolare, questi quattro elementi sono gestiti tramite lo *useState* per tracciare la selezione dell'utente.

La schermata è stata suddivisa in 4 sezioni: in ognuna di esse è presente un form dove inserire l'elemento indicato. Si è usato il componente Typeahead, di react-bootstrap, per fornire all'utente opzioni di selezione per ogni filtro. Ogni campo permette all'utente di selezionare un'opzione, con la possibilità di cancellare la selezione. Di seguito uno dei quattro form:

```

1 <Form.Group>
2   <Form.Label>Tag#1</Form.Label>
3   <div style={{ position: 'relative', width: '300px' }}>
4     <Typeahead
5       id="selections-example-1"
6       labelKey="tag#1"
7       onChange={setTag1}
8       options={props.tags1}
9       placeholder="All"
10      selected={tag1}
11      style={{ width: '100%', boxSizing: 'border-box' }}
12      clearButton={false}
13    />
14    {tag1.length > 0 && (
15      <button
16        type="button"
17        onClick={() => setTag1([])}
18      >
19        x
20      </button>
21    )}
22  </div>
23 </Form.Group>

```

Quando l'utente preme il pulsante di ricerca, viene costruita una query string usando le selezioni correnti. Quindi, viene eseguita una chiamata fetch per ottenere le attività filtrate, che vengono passate al componente della prossima schermata. Il componente ha quindi una funzione chiara: permette all'utente di filtrare i risultati tramite un'interfaccia semplice, gestisce la selezione e invia una richiesta per recuperare le attività corrispondenti ai criteri scelti.

Content Component

Il componente ContentView (figura 5.6) rappresenta una pagina che permette all'utente di visualizzare delle schede (card) di attività, applicare filtri sui risultati

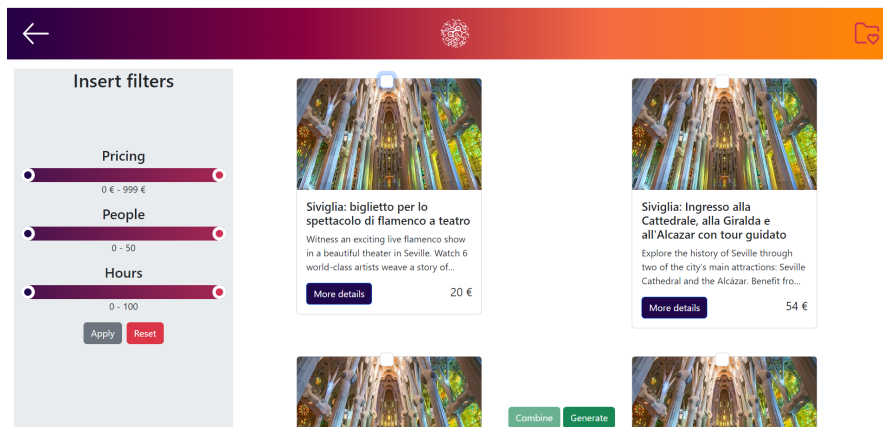


Figura 5.6: Schermata dell'elenco attività, in cui è possibile visionare le singole attività o filtrarle ulteriormente per poi generare l'esperienza unica

e visualizzare maggiori dettagli in un *modal*. Inoltre, consente di selezionare alcune cards per procedere alla generazione di un'esperienza unica finale.

Il componente gestisce i seguenti stati:

- *showModal* e *showedCard*: questi stati controllano la visibilità e il contenuto del modal. Quando l'utente clicca su una card, la card selezionata viene salvata in *showedCard* e viene aperto il modal.
- *price*, *people*, *hours*: Sono gli stati usati per gestire i filtri aggiuntivi per prezzo, numero di persone e durata dell'attività.
- *filteredCards*: rappresenta la lista di cards mostrate nella schermata e filtrate secondo i parametri scelti. Inizialmente è popolata con tutte le cards ricevute tramite props.

Le funzionalità principali sono:

- Applicazione / Reset filtri: vi sono due funzioni specifiche che si occupano di impostare o eventualmente ripristinare al valore iniziale i filtri.

```

1  const applyFilters = () => {
2    const filtered = props.cards.filter((card) => {
3      const isPriceValid = card.price >= price[0] && card.
price <= price[1];
4      const n_people = card.n_people.match(/\d+/g);
5      const isPeopleValid = n_people[0] >= people[0] &&
n_people[1] <= people[1];
6      const isHoursValid = card.hours >= hours[0] && card.
hours <= hours[1];
7      return isPriceValid && isPeopleValid && isHoursValid;

```



```

8     });
9     setFilteredCards(filtered);
10  };
11
12  const resetFilters = () => {
13    setFilteredCards(props.cards)
14    setPrice([0, 999])
15    setPeople([0, 50])
16    setHours([0, 50])
17  }
18

```

- Selezione attività per generazione: la funzione “handleCheckboxChange” si occupa di salvare in un vettore le attività selezionate per la combinazione, mentre la “handleCombine” semplicemente naviga nella schermata successiva in cui vi è la vera e propria generazione dell’attività. Nel caso in cui si volesse far effettuare la scelta al modello, si userà invece la “handleGenerate” che richiede la scelta alla rete e naviga autonomamente nella prossima schermata per la generazione.

In sintesi, il componente ContentViewer offre una schermata dove l’utente può esplorare attività, applicare filtri per prezzo, numero di persone e ore, visualizzare dettagli delle singole attività e selezionare quelle che preferisce. Una volta selezionate almeno due attività, l’utente può generarne una univoca. Può anche lasciare la scelta al modello per una maggiore automazione.

Generative Component



Figura 5.7: Schermata della generazione, in cui viene mostrato il risultato finale

Il componente GenerativeViewer (figura 5.7) è progettato per visualizzare un'attività combinata, generata a partire da più cards selezionate precedentemente. Consente inoltre di aggiungere questa attività alla lista personale.

Presenta quindi l'attività con le varie informazioni ed offre due funzionalità interagendo con i relativi bottoni: la prima riguarda la rigenerazione dell'attività che consente di generare nuovamente l'attività, specificando eventuali modifiche o aggiunte/rimozioni da effettuare. Inoltre, è possibile effettuare il salvataggio dell'attività: se l'attività ottenuta è di relativo interesse, essa può essere salvata per essere consultata in un secondo momento, all'interno di una sezione dedicata.

Personal Component

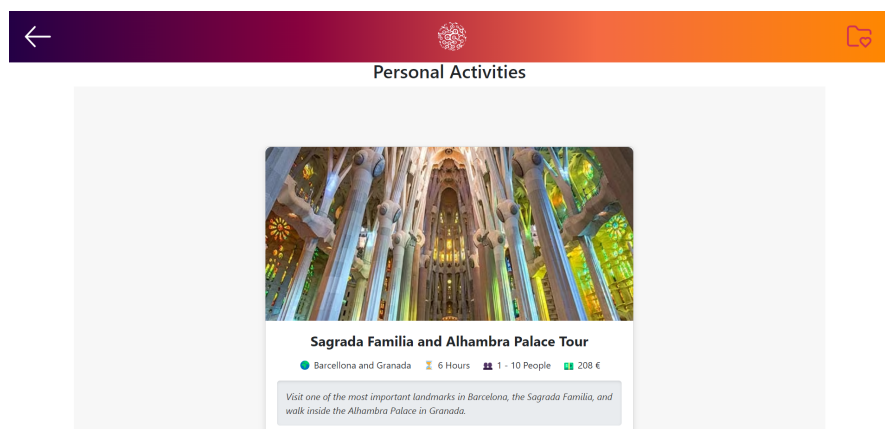


Figura 5.8: Schermata delle attività personali, in cui sono mostrate le attività generate che sono state salvate

Il componente PersonalContent (5.8) visualizza un elenco di attività personali dell'utente. Nel caso in cui non ve ne fossero, mostra un messaggio informativo tramite un *Alert*. Altrimenti, itera sulle attività ricevute e per ciascuna rende una card dettagliata utilizzando un componente ad hoc. Quest'ultimo mostra i dettagli di un'attività specifica in una card, includendo un'immagine, un titolo, luogo, durata, numero di partecipanti e prezzo, insieme a una descrizione stilizzata.

Map Component

Il componente MapViewer (5.9) rappresenta un'alternativa al Selector Component per avviare una ricerca diversamente. Difatti, è una mappa interattiva che utilizza Leaflet per visualizzare attività geolocalizzate in Spagna. Nel caso in esempio, viene creata una mappa centrata in Spagna; inoltre, si aggiunge un livello di cluster per raggruppare i marker delle attività oltre il livello città. Questo approccio

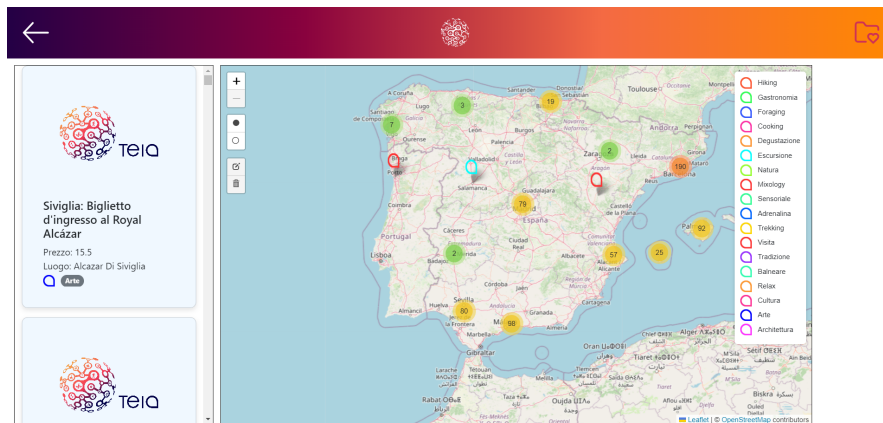


Figura 5.9: Schermata della Mappa, in cui è possibile effettuare la ricerca tramite una mappa interattiva

consente di raggruppare le attività vicine in un singolo cluster, semplificando la visualizzazione e la navigazione nella mappa.

Vi è comunque una doppia visualizzazione. Infatti lo schermo risulta diviso verticalmente in due parti:

- **Elenco attività:** sulla sinistra, tramite cui è possibile vedere l'elenco di tutte le attività.
- **Mappa:** sulla porzione di schermo restante, in cui i marcatori fissano le attività sulle coordinate specifiche.

Le due visualizzazioni sono interscambiabili: cliccando su un'attività nell'elenco, il relativo marcatore sulla mappa viene selezionato; viceversa, cliccando su un marcatore nella mappa, viene selezionata la corrispondente attività nell'elenco.

Infine è possibile tracciare un'area sulla quale verrà effettuata la seconda fase della ricerca (sempre tramite il Content Component).

5.5.3 Backend

Il server è stato creato con Flask, microframework web per Python. Grazie ad esso è stato possibile mantenere in linguaggio Python il codice precedentemente sviluppato in modo da doverlo solamente integrare nelle API. In particolare, serve come backend per un'applicazione web, fungendo da intermediario tra il frontend, in React, ed il database. Consente di effettuare richieste HTTP dal frontend per recuperare, elaborare o inserire dati, utilizzando diverse rotte API.

Configurazione del server

Innanzitutto è stato abilitato il CORS (Cross-Origin Resource Sharing) per consentire richieste provenienti solo dal frontend React (in esecuzione su `http://localhost:5173`). Secondariamente, si è stabilita la connessione al database PostgreSQL. Esso è un sistema di gestione di database relazionale open-source, avanzato e altamente personalizzabile. È conosciuto per la sua robustezza, affidabilità e conformità agli standard SQL. Supporta sia dati strutturati (tabelle con righe e colonne) sia funzionalità avanzate come la gestione di dati non relazionali (JSON, XML).

Principali API

Le API necessarie al corretto funzionamento del sistema si dividono essenzialmente in due categorie:

- **Gestione Database:** per permettere il caricamento e la visualizzazione delle attività o dei loro campi:
 - Ottenimento selettori (“`/api/get_data_input`”): recupera valori unici per tre campi (Where, Tag1, e Tag2) dalla tabella Activities. Questi valori unici sono utilizzati per riempire i menù a tendina nell’interfaccia utente.
 - Ottenimento attività (“`/api/activities`”): consente di recuperare un elenco di attività dal database in base a filtri opzionali, come luogo (where), data (when), e due tipi di tag (tag1 e tag2). In alternativa vi è anche “`/api/activities`” per ottenere tutte le attività (utilizzata nella visuale mappa).
 - Ottenimento attività personali (“`/api/personal-activities`”): come per la precedente, questa API restituisce tutte le attività personali memorizzate nella tabella MyActivities.
 - Salvataggio attività (“`/api/add-activity`”): consente di aggiungere una nuova attività nella tabella MyActivities. I dati dell’attività sono forniti nel body della richiesta sotto forma di JSON, che include titolo, descrizione, prezzo, durata, luogo, tag ed URL delle immagini.
- **API generative ad OpenAI:** consentono il dialogo con i server di OpenAI in modo tale da poter generare le esperienze:
 - Combinazione attività (“`/api/combine-cards`”): combina due o più attività (chiamate *cards*) in una singola attività utilizzando GPT-4. Questo endpoint utilizza l’API di OpenAI, già illustrata nel paragrafo dedicato, per creare una descrizione e altri dettagli combinati delle attività.

- Selezione attività automatica (“*/api/generate-activities*”): alternativa alla selezione manuale da parte di un utente, permette di utilizzare la rete per effettuare la scelta, in base a criteri specificati, delle migliori attività tra quelle passate come parametro. Ritorna quindi le attività selezionate che saranno poi combinate in un'altra chiamata al modello utilizzando l'API precedente.

In sintesi, tale server Flask è responsabile della gestione del flusso di dati tra il database PostgreSQL e il frontend React. Gestisce sia richieste per recuperare dati, come attività e filtri, sia per inserirli o elaborarli, ad esempio combinando due attività o aggiungendone di nuove. L'uso delle API GPT-4 rende il server capace di automatizzare la generazione di contenuti per le attività, espletando così il suo ruolo primario.

Capitolo 6

Risultati

In questo capitolo vengono presentati i risultati ottenuti dall'applicazione dei diversi moduli sviluppati nel progetto. Ogni modulo, ovvero scraping, preprocessing, analisi tramite intelligenza artificiale e generazione finale con GPT, ha contribuito alla pipeline completa, elaborando dati in modi specifici e fornendo output intermedi che hanno permesso l'evoluzione progressiva dei risultati finali.

6.1 Scraping

Il primo modulo si occupa di recuperare tutte le varie attività dalle varie fonti fornite. Nel capitolo implementativo è largamente spiegato come tali risultati ottenuti possono variare di molto nella forma a seconda dal sito di provenienza e dal metodo utilizzato.

Laddove possibile, si è quindi cercato di ottenere un output in formato JSON, strutturato e già pronto all'uso, così da ottimizzare la fase di preprocessing successiva.

Durante il processo di scraping, sono state raccolte complessivamente 13499 attività da 3 diverse fonti: 11349 da TripAdvisor, 900 da GetYourGuide e 1250 da Spain.info. Questo ha consentito di ottenere un insieme di dati sufficientemente ampio e variegato per le fasi di analisi successive. Nonostante la varietà delle fonti, il sistema è riuscito a gestire correttamente la maggior parte dei dati, pur affrontando alcune sfide, come la formattazione incoerente dei campi descrittivi in TripAdvisor e la gestione dei dati multilingua su GetYourGuide. Infatti si è preferito aggiungere la traduzione delle descrizioni in inglese in modo tale da migliorare il processo di analisi. Queste problematiche sono state risolte in larga parte nel modulo di preprocessing, garantendo la qualità e la coerenza dei dati.

Inoltre, il numero significativamente più alto di attività raccolte da TripAdvisor rispetto a GetYourGuide e Spain.info è dovuto alla vasta copertura di TripAdvisor

nel settore turistico. Essendo uno dei portali principali in tale ambito, TripAdvisor offre un database più ampio di attività, che copre una gamma estesa di attrazioni e servizi in molteplici destinazioni. In confronto, piattaforme come GetYourGuide tendono a concentrarsi su attività specifiche o esperienze curate, mentre Spain.info è maggiormente orientata a informazioni ufficiali su destinazioni spagnole, limitando il numero totale di attività disponibili.

Di seguito è fornito un esempio di dati estratti dal sito GetYourGuide:

```

1 {
2   "id": 1,
3   "titolo": "Siviglia: Ingresso alla Cattedrale, alla Giralda e all'
4     Alcazar con tour guidato",
5   "descrizione": "Esplora la storia di Siviglia attraverso due delle
6     principali attrazioni della città: la Cattedrale di Siviglia e
7     l'Alcázar. Approfitta dell'ingresso prioritario e scopri di pi
8     ù sui monumenti dalla tua guida esperta. Inizia visitando la
9     Cattedrale di Siviglia e la sua bellissima architettura. Goditi
10    l'ingresso prioritario e scopri l'affascinante storia
11    multiculturale di questo sito sacro. Sali sulla Giralda, il
12    campanile della Cattedrale. Visita poi l'Alcázar e scopri la
    diversità di culture ed eredità storiche esistente a Siviglia.
    Dopo l'ingresso prioritario, la tua guida ti condurrà in una
    piacevole passeggiata attraverso alberi di arancio e mirto e ti
    aiuterà a comprendere la profondità della storia e della
    cultura della città.",
13  "prezzo": "54 euro",
14  "lingue": [ "Spagnolo", "Italiano", "Inglese", "Francese" ],
15  "luogo": "Siviglia",
16  "durata": "2.5 ore",
17  "fornitore": "Seville inside, sito web: https://sevillainside.com/en/",
18  "valutazione": "4.6 su 5",
19  "numero_recensioni": 5834
20 }

```

Poiché le informazioni provengono da fonti diverse, è naturale che ci siano variazioni nei dati e, di conseguenza, nei file JSON di output. Un altro esempio di output è il seguente, in questo caso ottenuto da Spain.info:

```

1 {
2   "id": 5,
3   "titolo": "Ferrata route: Vidosa in Asturias",

```

```

4 | "descrizione": "A Ferrata route is a vertical and horizontal
   | route equipped with different material: nails, staples, dams,
   | railings, chains, suspension bridges and zip lines that allow
   | you to safely reach areas difficult to access for hikers or not
   | accustomed to climbing. The security is in charge of a cable
   | of steel installed in all the way and the harness provided of a
   | heatsink and special carabiners of via Ferrata that assure in
   | case of fall. When? We carry out the activity throughout the
   | year, except in meteorological or other adverse conditions.
   | With prior booking. Place and Schedule The departures are made
   | at 9:30 am from our facilities at the Finca la Fundición de
   | Coviella in Arriondas. Duration time The duration of the
   | activity is approximately two hours, plus the transfer in our
   | vehicles. Where In the gorge of Beyos (Ponga), in Vidosa.",
5 | "prezzo": "A partire da: 55.0 euro a persona (tasse comprese)",
6 | "lingue": "Spanish, English",
7 | "luogo": "Cangas de Onís (Asturias)",
8 | "durata": "2 ore",
9 | "fornitore": "Escuela Asturiana de Piragüismo - Ranasella",
10 | "valutazione": "4.4 su 5",
11 | "categorie": "Sport e avventure",
12 | "target": "Adults without children, Young people, LGBTQI+,
   | Senior"
13 | }

```

Infine, oltre ai risultati precedenti derivati dalla soluzione BeautifulSoup e Selenium, si riporta anche un esempio di attività su TripAdvisor ottenuta da Apify:

```

1 | {
2 |   "id": "5828389",
3 |   "type": "attraction",
4 |   "category": "attraction",
5 |   "subcategories": [
6 |     "Tours",
7 |     "Boat Tours & Water Sports",
8 |     "Outdoor Activities"
9 |   ],
10 |   "name": "Andalucia Activities",
11 |   "locationString": "Cadiz, Costa de la Luz, Andalucia",
12 |   "description": "An outdoors activities company offering action
   | 365 days a year. Located at the port of Sotogrande all year and
   | port of La Duquesa from June to September.",
13 |   "image": "https://media-cdn.tripadvisor.com/media/photo-o/0e/8a/
   | ce/f1/sup-para-toda-la-familia.jpg",
14 |   "photoCount": 14,
15 |   "awards": [],
16 |   "rankingPosition": 18,
17 |   "rating": 3.5,

```



```
18 | "rawRanking": 2.738262176513672 ,
19 | "phone": "+34 633 53 89 30",
20 | "address": "Cadiz Spain",
21 | "localName": "Andalucía Activities",
22 | "email": "info@andaluciaactivities.com",
23 | "latitude": 36.292343 ,
24 | "longitude": -5.273738
25 | }
```

Saranno file di questo formato a progredire negli steps successivi della pipeline. Nel caso in cui si ottenessero invece file testuali, problema più volte incontrato, dovrà essere necessario un ulteriore passo nel preprocessamento che però attualmente non è stato affrontato lasciandolo a casi futuri.

Per completezza, un esempio di output, troncato, in formato testuale:

```
1 | Le tappe del viaggio:
2 | MALAGA - GRANADA - CORDOBA - SIVIGLIA - CADICE + TAPPA A
   | GIBILTERRA - MALAGA - RONDA - MALAGA"
3 | "ANDALUSIA: 7a Tappa Malaga / 8a Tappa Ronda": "7a tappa:
   | Malaga
4 |
5 | 23 maggio
6 | Di nuovo nell'albergo delizioso della 1a tappa, il Bahia.
7 | Anche a Malaga usiamo la formula molto speciale "hop on hop
   | off" del "City-Sightseeing" che ci permette di godere di
   | tutte le sue attrazioni godendo della maggiore libertà e
   | che regala viste dall'alto dei suoi autobus, a due piani
   | e scoperti, viste panoramiche eccezionali di Malaga con
   | un biglietto valido 24 ore il cui prezzo è di 29E e che
   | comprende alcune facilitazioni compresa una piccola
   | crociera nella baia. Potremo salire e scendere a nostro
   | piacimento alle fermate situate lungo il percorso in base
   | ai nostri gusti. Prima tappa di oggi l'Alcazaba.
8 |
9 | L'Alcazaba è un edificio impressionante costruito tra il VII
   | e XI sec, dove prima sorgeva una città romana, con le
   | funzioni di fortezza e di palazzo, dove hanno vissuto i
   | governanti della Malaga musulmana. Dalla pianta molto
   | irregolare, tutti gli ambienti sono disposti a diversi
   | livelli per adattarsi alla conformazione del terreno e
   | sono concentrati in due recinti di mura.
```

10 Il Palazzo occupa tre cortili consecutivi. Il primo, chiamato de "Los Surtidores" (riferito agli zampilli della fontana), conserva degli archi del periodo del califfato e conduce ad un ambiente che a sua volta conduce alla "Torre de la Armadura Mudejar" che ha un soffitto in legno decorato e poi alla "Torre Maldonado" con bellissime colonne ed archi e con uno straordinario belvedere sulla città. Continuando si accede al "Cortile degli aranci" e all'Alberca (piscina). Una veloce visita al museo con varie testimonianze storiche ed un plastico del sito. Questa fortezza-palazzo è uno dei punti di riferimento della città, un luogo molto visitato che combina storia e bellezza nello stesso luogo.

11 Ai piedi dell'Alcazaba è stato ritrovato nel 1951 un teatro romano. Per molti secoli, il teatro romano di Malaga è rimasto nascosto sotto terra. La costruzione risale al I secolo, sotto l'impero di Augusto. Fu in uso fino al III secolo. Gli arabi utilizzarono alcuni elementi di questa costruzione, tra cui i capitelli e i fusti delle colonne, a beneficio della Alcazaba, come sostegno per gli archi a ferro di cavallo delle porte di questo edificio. Le dimensioni sono 31 metri di raggio, 16 metri di altezza.

12 Torniamo in centro per visitare la cattedrale.

13 ...

6.2 Preprocessing

Il modulo di preprocessing ha svolto un ruolo cruciale nell'uniformazione dei dati raccolti da fonti diverse. In molti casi, le informazioni provenivano da siti con formattazioni differenti e/o con dati mancanti. L'obiettivo del preprocessing è stato quello di convertire i dati in un formato coerente e standard, principalmente JSON, eliminando le incongruenze e gestendo eventuali dati mancanti.

Infatti, è emerso che circa il 15% dei dati iniziali presentava informazioni essenziali mancanti come, ad esempio, nome o locazione dell'attività. Per garantire coerenza e qualità nei risultati finali, questi dati sono stati scartati dal set principale. Infine, si sono aggiunte le informazioni riguardo latitudine e longitudine ricavandole dal luogo dell'attività.

È stata inoltre aggiunta la possibilità di salvare gli stessi dati anche in formato Excel in modo da non aver limitazioni durante le analisi.

Dopo l'elaborazione, i dati risultano uniformati e pronti per la fase successiva di analisi tramite intelligenza artificiale. Un esempio del risultato ottenuto dopo il preprocessing è il seguente:

```
1 {  
2   "id": 5,  
3   "name": "Ferrata route: Vidosa in Asturias",  
4   "description": "A Ferrata route is a vertical and horizontal  
   route equipped with different material: nails, staples, dams,  
   railings, chains, suspension bridges and zip lines that allow  
   you to safely reach areas difficult to access for hikers or not  
   accustomed to climbing. The security is in charge of a cable  
   of steel installed in all the way and the harness provided of a  
   heatsink and special carabiners of via Ferrata that assure in  
   case of fall. When? We carry out the activity throughout the  
   year, except in meteorological or other adverse conditions.  
   With prior booking. Place and Schedule The departures are made  
   at 9:30 am from our facilities at the Finca la Fundición de  
   Coviella in Arriendas. Duration time The duration of the  
   activity is approximately two hours, plus the transfer in our  
   vehicles. Where In the gorge of Beyos (Ponga), in Vidosa.",  
5   "price": "55.0 euro",  
6   "location": "Cangas de Onís (Asturias)",  
7   "latitude": 43.35,  
8   "longitude": -5.13,  
9   "duration": "2 ore"  
10 }
```

Attraverso questo processo, i dati sono stati puliti e standardizzati, permettendo la loro analisi da parte del modulo successivo in modo più efficiente.

6.3 Clustering e Tagging

Il modulo di clustering e tagging ha prodotto un output consistente con quanto ci si attendeva, arricchendo con nuove informazioni derivanti dall'analisi geografica e dal modello di classificazione BART i dati di input. Queste informazioni aggiuntive includono il cluster geografico di appartenenza dell'attività, il tag primario e secondario e la stagione associata all'attività.

- L'algoritmo di clustering geografico, basato su KMeans, ha assegnato ogni attività a uno specifico cluster in base alle sue coordinate geografiche (latitudine e longitudine). Questo processo ha permesso di categorizzare le attività in gruppi omogenei dal punto di vista geografico, permettendo visualizzazioni più mirate.

L'output finale del clustering geografico si riflette in un nuovo campo, denominato **Cluster**, aggiunto ad ogni attività nel file di input. Esso indica il gruppo geografico cui appartiene.

- Per ogni attività è stato applicato un modello BART per il tagging. Il modello ha analizzato le descrizioni delle attività per assegnare i tag primario e secondario, oltre a indicare il periodo dell'anno più adatto (in termini di stagione) per lo svolgimento dell'attività.

L'output del processo di tagging si traduce nell'aggiunta di tre nuovi campi nel file JSON per ogni attività: **Tag1** (tag primario), **Tag2** (tag secondario) e **Periodo** (stagione associata).

Di seguito un esempio del risultato ottenuto applicando contemporaneamente gli algoritmi di clustering e tagging:

```

1 {
2   "id": 5,
3   "name": "Ferrata route: Vidosa in Asturias",
4   "description": "A Ferrata route is a vertical and horizontal
   route equipped with different material: nails, staples, dams,
   railings, chains, suspension bridges and zip lines that allow
   you to safely reach areas difficult to access for hikers or not
   accustomed to climbing. The security is in charge of a cable
   of steel installed in all the way and the harness provided of a
   heatsink and special carabiners of via Ferrata that assure in
   case of fall. When? We carry out the activity throughout the
   year, except in meteorological or other adverse conditions.
   With prior booking. Place and Schedule The departures are made
   at 9:30 am from our facilities at the Finca la Fundición de
   Coviella in Arriondas. Duration time The duration of the
   activity is approximately two hours, plus the transfer in our
   vehicles. Where In the gorge of Beyos (Ponga), in Vidosa.",
5   "price": "55.0 euro",
6   "duration": "2 ore",
7   "Cluster": 3,
8   "Tag1": "Hiking",
9   "Tag2": "Adventure",
10  "Periodo": "Summer"
11 }
```

Ad ogni modo è interessante notare qualche risultato quantitativo:

- **Clustering**: si sono definiti dodici clusters che rappresentano grossomodo le regioni spagnole.
- **Tagging**: l'analisi dei risultati del tagging ha evidenziato alcune tendenze interessanti. Tra i diciotto tag identificati, i primi tre (Escursione, Cultura

e Visite) coprono rispettivamente il 20%, il 14,3% e il 9,6% delle attività, rappresentando insieme il 43,9% del totale. Questa distribuzione riflette una preferenza marcata verso attività che coinvolgono esplorazione del territorio, esperienze culturali e visite guidate, in linea con le caratteristiche principali delle offerte turistiche in Spagna.

Per quanto riguarda la stagionalità, emerge che la maggior parte delle attività è associata alla Primavera (48,47%) e all'Estate (31,8%), periodi tipicamente di alta affluenza turistica. Questo risultato è coerente con il flusso stagionale del turismo, che tende a concentrarsi nei mesi con condizioni climatiche più favorevoli, in particolare quello spagnolo.

Questi dati, in particolare quelli ottenuti dal Tagging, suggeriscono che il modello utilizzato ha catturato correttamente le tendenze prevalenti nelle descrizioni delle attività turistiche, contribuendo ad una categorizzazione significativa ed utile per gli utenti finali.

6.4 Copywriting automatico

In questa sezione, si mostrano i risultati dell'implementazione del modulo di copywriting automatico che utilizza l'intelligenza artificiale per generare esperienze uniche. Questo modulo sfrutta le API di OpenAI per analizzare le attività fornite e produrre una descrizione coerente e persuasiva.

Il compito principale è quello di combinare le attività ricevute in una unica, ma vi è la possibilità di lasciare la scelta alla rete che poi le unisce.

Nel caso si volesse procedere così facendo vi saranno due interazioni con GPT:

- Scelta schede: si chiede al modello di scegliere un sottoinsieme di attività che siano complementari, o comunque affini, per essere combinate a partire dall'insieme di quelle che erano state precedentemente filtrate. Di seguito un esempio di output risultante da attività estive, le descrizioni sono state abbreviate:

```
1 {
2   "selected_activities": [
3     {
4       "id": 19,
5       "title": "Los Cristianos: Crociera Eco-Yacht per
6       avvistare le balene con bagno",
7       "description": "Embark on an unforgettable whale
8       watching cruise along the coast of Los Cristianos in
9       Tenerife...",
10      "location": "Tenerife",
11      "time": "Summer",
12      "tag1": "Relax",
```

```
10     "tag2": "Degustazione",
11     "price": 15,
12     "n_people": "[2-8]",
13     "hours": 2,
14     "imgUrl": "https://www.spain.info/export/sites/segtur/.
content/imagenes/reportajes/cataluna/sagrada-familia-
interior.jpg"
15   },
16   {
17     "id": 21,
18     "title": "Ibiza: crociera lungo la costa con
paddleboarding, cibo e bevande",
19     "description": "Take a cruise from Port de Sant Antoni
de Portmany and enjoy the breeze along Ibiza's dazzling
coastline. Benefit from informative commentary from your
guide on this 6-hour cruise...",
20     "location": "Ibiza",
21     "time": "Summer",
22     "tag1": "Relax",
23     "tag2": "Degustazione",
24     "price": 69,
25     "n_people": "[2-8]",
26     "hours": 6,
27     "imgUrl": "https://www.spain.info/export/sites/segtur/.
content/imagenes/reportajes/cataluna/sagrada-familia-
interior.jpg"
28   },
29   {
30     "id": 22,
31     "title": "Palma di Maiorca: crociera in catamarano di 5
ore con pranzo e nuotata",
32     "description": "Climb aboard a catamaran and sail from
Palma de Mallorca to the coastal resorts of Portals Vells
or Cala Vella, depending on wind conditions. See cruise
ships and VIP yachts as you leave the port of Palma and set
sail along the island's magnificent southern coast...",
33     "location": "Maiorca",
34     "time": "Summer",
35     "tag1": "Relax",
36     "tag2": "Arte",
37     "price": 56,
38     "n_people": "[2-8]",
39     "hours": 5,
40     "imgUrl": "https://www.spain.info/export/sites/segtur/.
content/imagenes/reportajes/cataluna/sagrada-familia-
interior.jpg"
41   }
42 ]
43 }
```

- Esperienza generata tramite combinazione: dopo il primo step, viene chiesta la generazione di un'attività unica partendo da quelle fornite. Quest'ultima sarà ritornata in un formato analogo a quelle di partenza, JSON:

```

1 {
2   'description': "Immergetevi in una serie di avventure
marittime con partenza da tre diverse località nelle isole
spagnole. Iniziate con una crociera eco-yacht da Los
Cristianos a Tenerife, dove avrete l'opportunità di
avvistare balene e altri animali marini. In seguito,
partite dal Port de Sant Antoni de Portmany per una
crociera di 6 ore lungo la costa di Ibiza con possibilità
di fare paddleboarding, degustare cibi e bevande, e nuotare
in meravigliose zone di mare. Infine, salite a bordo di un
catamarano a Palma de Mallorca e raggiungete le località
balneari di Portals Vells o Cala Vella. Fate due soste per
nuotare in acque cristalline e approfittate di un delizioso
pranzo a buffet con vino e sangria. In tutto, il viaggio
dura 13 ore e offre un'esperienza memorabile."
3   'hours': 13
4   'imgUrl1': "https://www.spain.info/export/sites/segur/.
content/imagenes/reportajes/cataluna/sagrada-familia-
interior.jpg"
5   'imgUrl2': "https://cdn.britannica.com/56/140856-050-
C408FFB6/Patio-de-los-Arrayanes-Alhambra-Spain-Granada.jpg"
6   'location': "Tenerife, Ibiza, Maiorca"
7   'n_people': " [2-8]"
8   'price': 140
9   'tag1': "Relax"
10  'tag2': "Nature"
11  'time': "Summer"
12  'title': "Viaggio completo: Eco-Yacht, crociera lungo la
costa con pranzo e nuotata in catamarano"
13 }
14

```

Nel caso in cui invece le attività venissero scelte direttamente da un'operatore umano, vi sarà solamente il secondo step con la loro combinazione.

Ad ogni modo, dall'ultimo output si evince come prezzo e durata siano la somma delle singole attività, inoltre le informazioni si limitano a quanto contenuto nelle attività specificate, come indicato. Si può anche notare come tale risposta sembri coesa e naturale, rivelando una buona adattabilità del modello alle varie richieste.

Infine è interessante notare la durata dei tempi di esecuzione: l'intero processo di selezione e generazione delle attività è stato progettato per massimizzare l'efficienza, bilanciando però con la qualità della generazione finale. In media, il tempo necessario per completare il primo step di selezione delle attività, se effettuato automaticamente, è di circa 1 minuto; mentre la generazione dell'attività combinata richiede circa 30 secondi. Questi risultati dimostrano la velocità con cui il sistema è in grado di processare richieste complesse, garantendo un'esperienza utente fluida ed immediata oltrechè un risultato realistico.

6.5 Valutazioni Finali

Dopo l'implementazione del prototipo, è stato eseguito un test di usabilità per valutarne l'efficacia e osservare come gli utenti interagiscono con il sistema. L'obiettivo principale era ottenere un feedback chiaro sulle scelte di design e identificare eventuali miglioramenti o problematiche nell'interfaccia.

Il test ha seguito una fase preliminare di pianificazione, seguita dall'esecuzione dei test e dall'analisi dei risultati ottenuti. La fase di pianificazione è stata fondamentale per stabilire le fasi del test ed in particolare le attività da svolgere. Da tale fase è scaturito lo script da seguire durante i test, necessario ad indicare le attività all'utente.

6.5.1 Pianificazione

Per condurre i vari test, è stato necessario definire uno script da seguire. Prima di poter fare quest'attività dovevano essere definiti gli obiettivi dei test oltrechè il target cui farli eseguire.

Target

Partendo dal target, l'interfaccia non è prevista per un utilizzo da parte di privati e quindi come prodotto finale bensì pensata per le aziende fornitrici di pacchetti turistici. Si è quindi contattato i partner per effettuare la valutazione.

Una volta definito la categoria dei partecipanti al test, si è dovuto stabilire il numero di quest'ultimi. Affidandosi ad importanti ricerche in merito, tra cui quella di Jakob Nielsen [16], si può dimostrare che un numero sufficiente di partecipanti risulti essere cinque poichè, con tale numero, è possibile identificare circa l'85% dei problemi di usabilità più comuni. Ad ogni modo ci si riferirà ai partecipanti con i codici U1 ad U5 per garantirne la privacy.

Obiettivi

A questo punto, è stato possibile procedere con l'identificazione degli obiettivi da raggiungere tramite tali test; essenzialmente, ne sono stati definiti due:

- **Valutazione dell'usabilità:** l'obiettivo principale era valutare se l'interfaccia risultasse usabile e capire come gli utenti reali avrebbero interagito con il sistema, assicurandosi che interpretassero correttamente i contenuti. Inoltre, si voleva verificare che il flusso introdotto per la generazione delle attività fosse intuitivo e facile da seguire. Durante i test, si cercavano anche suggerimenti per migliorare l'interfaccia o aggiungere nuove funzionalità non precedentemente considerate.
- **Individuazione di problematiche:** secondariamente si aveva l'intenzione di individuare eventuali problemi non scovati durante lo sviluppo o situazioni non attese nell'esecuzione con l'intento di correggerli.

6.5.2 Attività

Dopo la pianificazione, per la valutazione dell'interfaccia si è pensato di definire un insieme di attività (riassunte nella tabella 6.1) da eseguire:

- **Attività 1:** nella prima attività, viene richiesto all'utente di selezionare gli elementi chiave necessari per avviare la ricerca delle attività. L'utente può interagire con i menù a tendina per scegliere le caratteristiche rilevanti per ciascuno dei quattro filtri (ad esempio, luogo, periodo o tag) non essendo però obbligato ad impostarli tutti. L'attività si considera "completata con successo" se l'utente riesce a impostare i filtri e avviare la ricerca, prestando attenzione alla possibilità di lasciare un input vuoto per esplorare tutte le opzioni.
- **Attività 2:** per la seconda attività, l'utente deve interagire con gli elementi della schermata relativa all'elenco di attività ottenute, applicando filtri aggiuntivi tramite gli sliders (per impostare durata, costo massimo e numero di persone per l'attività) e selezionando in dettaglio un'attività.
- **Attività 3:** in questa attività si richiede all'utente di utilizzare una delle due funzionalità principali della piattaforma: generazione automatica delle attività o combinazione manuale. Se l'utente sceglie la generazione automatica, il sistema combinerà automaticamente le attività in un'unica proposta. Se opta per la combinazione manuale, dovrà selezionare le attività tra quelle filtrate e unirle. In particolare, si terrà in considerazione quale opzione sarà scelta dall'utente.

- **Attività 4:** in questa attività, l'utente si trova nella schermata finale, dove può visualizzare i dettagli dell'attività proposta. L'utente può far rigenerare la descrizione dell'attività o salvarla nelle attività personali tramite i bottoni dedicati. Dopo eventuali rigenerazioni, si chiederà all'utente di salvare l'attività.
- **Attività 5:** per l'ultima attività, l'utente è invitato a navigare nella sezione delle attività personali. Qui potrà visualizzare tutte le proposte salvate.

6.5.3 Svolgimento

Per lo svolgimento, i test sono stati condotti online, fornendo un computer con il prototipo funzionante accessibile tramite browser. Inoltre, per facilitare il processo, ogni partecipante è stato affiancato da un facilitatore, il cui compito era agevolare gli utenti nel compimento delle attività, oltre a registrarne comportamenti e ragionamenti. Eventualmente avrebbe avuto anche il compito di intervenire se un utente avesse incontrato problemi nello svolgimento del test.

Alle cinque attività definite si è poi associata una metrica specifica per valutare matematicamente il successo o il fallimento del test. Al termine, a ciascun partecipante è stato somministrato un questionario SUS (System Usability Scale) [17] insieme a domande aggiuntive mirate a identificare eventuali problemi, criticità o funzionalità da introdurre.

Il questionario SUS è uno strumento largamente diffuso per valutare la percezione degli utenti riguardo all'usabilità di un sistema, prodotto o interfaccia. Comprende domande standardizzate che misurano l'usabilità attraverso valutazioni soggettive. Le domande del questionario riguardano vari aspetti dell'usabilità, come la facilità di apprendimento, l'efficienza d'uso, la memorabilità delle funzioni e la soddisfazione complessiva dell'utente.

Le risposte vengono poi analizzate per ottenere un punteggio complessivo che indica il livello di usabilità del sistema o prodotto.

6.5.4 Risultati

- **Attività 1:** tutti i partecipanti sono riusciti ad inserire gli elementi per i vari selettori. La maggior parte ne ha lasciato qualcuno vuoto facendo sì che venisse impostato su "All".
- **Attività 2:** anche in questa attività i partecipanti non hanno avuto problemi nell'applicare ulteriori filtri oltreché nel selezionare una singola attività per la visualizzazione. In particolare U1 ed U4 hanno notato l'ulteriore funzionalità di reset dei filtri.

- **Attività 3:** giunti in quella che è risultata essere l'attività più complessa, solamente tre partecipanti sono riusciti a far partire la generazione dell'attività scegliendo indipendentemente una o l'altra modalità; al contrario, i rimanenti hanno chiesto delucidazioni sulle due modalità e sul come eseguirle.
- **Attività 4:** a questo punto, tutti i partecipanti non hanno incontrato problemi nel far rigenerare l'attività o nel salvarla. In dettaglio, quattro hanno semplicemente lanciato nuovamente la generazione mentre solamente un utente ha specificato un'ulteriore richiesta.
- **Attività 5:** anche l'attività finale è stata eseguita senza grosse difficoltà. I partecipanti hanno navigato, tramite l'apposito bottone, nella sezione delle attività personali per poi visualizzarne il contenuto.

Nel complesso, i test hanno dato esiti favorevoli. Tutti i partecipanti, seppur con diverse difficoltà, sono stati in grado di completare con successo tutte le funzionalità previste dall'applicazione.

6.5.5 Eventuali modifiche

In base ad i test effettuati e ai riscontri dei partecipanti, si evince che l'interfaccia è nel complesso intuitiva e di facile utilizzo. Nonostante ciò, vi è spazio per ulteriori miglioramenti, emersi dai feedback degli utenti:

- Nella fase di generazione sarebbe necessaria maggiore chiarezza, in particolare andrebbero illustrate meglio le due diverse modalità (anche tramite un popup cliccabile).
- Una volta generata l'esperienza, potrebbe essere utile offrire la possibilità di modificare i campi manualmente.
- Eventualmente, si potrebbe implementare una sezione personale completa di funzionalità di login. Questa caratteristica è stata momentaneamente esclusa per ragioni di tempo, ma risulta sicuramente da includere nelle fasi future del progetto.

| TITOLO | DESCRIZIONE | CRITERI DI SUCCESSO | METRICA |
|------------|---|---|---|
| Attività 1 | Selezione elementi chiave per la ricerca | Il partecipante riesce ad inserire almeno due selettori e ad avviare la ricerca | Successo (1) o fallimento (0) |
| Attività 2 | Selezione filtri aggiuntivi e visualizzazione delle singole attività in dettaglio | Il partecipante riesce ad inserire i filtri aggiuntivi e a selezionare un'attività per la visualizzazione | 1 punto per ogni interazione, per un massimo di 2 |
| Attività 3 | Generazione esperienza unica partendo da un sottoinsieme | Il partecipante riesce a scegliere una delle due modalità presenti e a far partire la generazione | Successo (1) o fallimento (0) |
| Attività 4 | Rigenerazione della descrizione e salvataggio dell'attività ottenuta | Il partecipante riesce a far rieseguire la generazione per poi salvare l'attività | 1 punto per ogni interazione, per un massimo di 2 |
| Attività 5 | Navigazione nell'elenco di attività salvate | Il partecipante riesce a navigare nella suddetta sezione | Successo (1) o fallimento (0) |

Tabella 6.1: Attività per la valutazione dell'interfaccia web

| Partecipante | A1 | A2 | A3 | A4 | A5 | Successo medio partecipante |
|-------------------------|------|------|-----|------|------|-----------------------------|
| U1 | 1 | 2/2 | 1 | 2/2 | 1 | 100% |
| U2 | 1 | 2/2 | 1 | 2/2 | 1 | 100% |
| U3 | 1 | 2/2 | 0 | 2/2 | 1 | 80% |
| U4 | 1 | 2/2 | 1 | 2/2 | 1 | 100% |
| U5 | 1 | 2/2 | 0 | 2/2 | 1 | 80% |
| Successo medio attività | 100% | 100% | 60% | 100% | 100% | |

Tabella 6.2: Tabella dei risultati: esiti delle attività per la valutazione

Capitolo 7

Conclusioni

Il progetto di tesi ha avuto come obiettivo lo sviluppo di un sistema in grado di creare esperienze personalizzate e, soprattutto, uniche.

Il lavoro è iniziato con una fase di ricerca e studio del dominio, al fine di comprendere innanzitutto il contesto di utilizzo e le esigenze degli utenti. Si sono poi definiti i problemi da risolvere e quali moduli potessero occuparsene, per poi definire quali fossero le tecnologie più adatte a risolverli.

Una volta deciso come procedere, è stato progettato il sistema che potesse sfruttare tecnologie di scraping ed analisi dati, al fine di raccogliere e processare informazioni relative alle attività disponibili online. Il modulo di scraping è stato implementato per estrarre attività da fonti eterogenee, mentre l'analisi è stata finalizzata a identificare e categorizzare i dati in modo accurato. Dopo tali moduli, ne è stato integrato uno di generazione, basato su modelli GPT, per creare le esperienze uniche combinando le attività selezionate in modo automatico.

A questo punto si è definita ed implementata l'interfaccia che permettesse l'interazione tra un utilizzatore ed il sistema.

Dopo l'implementazione del sistema, sono stati condotti test con un gruppo di utenti reali per valutarne l'usabilità e l'efficacia. I partecipanti hanno interagito con la piattaforma, testando sia la generazione automatica delle esperienze che la combinazione manuale delle attività. Il feedback raccolto è stato complessivamente positivo, evidenziando la semplicità d'uso e l'utilità della piattaforma, con alcune modifiche suggerite per migliorare l'esperienza utente.

Infine, è stato somministrato un questionario SUS per valutare la percezione degli utenti sull'usabilità del sistema. I risultati hanno indicato un livello di soddisfazione generale, confermando il livello positivo del prodotto ottenuto.

Concludendo, si può affermare che il sistema sviluppato riesce ad adempiere agli obiettivi minimi prefissati per tale progetto. Può quindi essere considerato un buon punto di partenza per sviluppi futuri atti a migliorarne la qualità.

7.1 Sviluppi futuri

Nonostante i risultati positivi ottenuti a seguito della conclusione del progetto, ci sono ampi margini di miglioramento e di espansione delle funzionalità. Molte delle caratteristiche trattate a livello teorico, infatti, non sono poi state implementate nella versione finale per concentrarsi su un insieme minimo che garantisca la piena operatività del sistema. Tra le migliorie possibili, si potrebbero considerare:

- Miglioramento del preprocessing dei dati, includendo tecniche avanzate di pulizia e arricchimento delle informazioni estratte.
- Implementazione della sentiment analysis sulle recensioni degli utenti per arricchire l'analisi qualitativa delle attività proposte.
- Completamento del sistema di valutazione basato su un punteggio di unicità, che permetterebbe di quantificare quanto le esperienze generate siano effettivamente nuove e irripetibili.
- Estensione della seconda modalità di utilizzo, dove l'utente, inserendo un sottoinsieme dei selettori, ottiene prima il completamento dei rimanenti e, solo successivamente, l'avvio della ricerca.
- Implementazione della prima sezione del portale, nella quale vengono effettuati scraping ed analisi delle attività. Tali operazioni possono essere avviate manualmente da terminale, indipendentemente l'una dall'altra, ed è necessaria l'integrazione nel prototipo.

Queste integrazioni, se realizzate, contribuirebbero a rendere TEIA un prodotto ancora più completo, capace inoltre di offrire esperienze uniche ed ottimizzate in modo ancor più preciso sulle esigenze degli utenti.

Appendice A

Script per la valutazione del prototipo

A.1 Presentazione

Buongiorno! Grazie per aver accettato di partecipare a questo studio di usabilità. Prima di iniziare, vorrei fornirti una breve panoramica sul contesto e lo scopo del prototipo che stai per utilizzare.

Il progetto che ho sviluppato ha come obiettivo la volontà di offrire agli utenti un'esperienza guidata nella creazione di attività uniche partendo da altre già esistenti. L'obiettivo principale è quello di permettere agli utenti di esplorare e selezionare una serie di attività basate su vari criteri, come descrizione, tipo e altre caratteristiche rilevanti, e infine combinarle in un'esperienza unica tramite il supporto dell'intelligenza artificiale.

Il sistema ti permetterà di selezionare attività tramite un'interfaccia composta da filtri, mappe e altre opzioni, per poi generare una combinazione finale. La proposta mira a facilitare l'interazione con il sistema in modo intuitivo e guidato.

Durante il test, ti verranno richiesti alcuni compiti specifici da completare, ma sei libero di esplorare l'applicazione come preferisci. (Ci sarà un facilitatore pronto ad aiutarti nel caso tu incontrassi delle difficoltà). Non c'è alcuna pressione, vorrei solo raccogliere le tue opinioni e osservazioni mentre navighi attraverso il sistema.

A.2 Istruzioni

Quello che stai per utilizzare è un prototipo funzionale del sito sviluppato, quindi alcune aree potrebbero essere ancora in fase di sviluppo, ma le funzionalità principali dovrebbero funzionare correttamente.

Per iniziare, apri il link che ti è stato fornito e condividi il tuo schermo, così che possa seguirti mentre esplori l'applicazione. Durante la sessione, ti invito a descrivere ad alta voce i tuoi pensieri, cosa ti aspetti di vedere e cosa provi mentre navighi tra le varie funzionalità. Questo ci aiuterà a comprendere meglio la tua esperienza.

Al termine del test, ti chiederò anche di completare un questionario SUS (System Usability Scale), composto da dieci domande per raccogliere la tua valutazione sull'usabilità del sistema.

Sei pronto per iniziare?

A.3 Attività

A.3.1 Attività 1: Selezione degli elementi chiave per la ricerca

In questa prima attività, ti chiedo di selezionare gli elementi chiave necessari per avviare la ricerca delle attività. Puoi interagire con i menù a tendina per scegliere le caratteristiche rilevanti per ciascuno dei quattro filtri (ad esempio, luogo, periodo o tags). Una volta impostati gli elementi di interesse, avvia la ricerca.

Vedere se il soggetto nota la possibilità di lasciare un input vuoto, avviando la ricerca su tutte le opzioni per quel campo.

Al termine, passare alla prossima attività.

A.3.2 Attività 2: Filtri aggiuntivi e visualizzazione singole attività

Ora, come seconda attività, ti chiedo di interagire con gli elementi presenti nella nuova schermata relativi all'elenco di attività ottenute. Infatti potrai ulteriormente filtrare l'elenco tramite gli sliders impostando anche durata e/o costo massimo come anche il range di persone per l'attività. Una volta applicati i filtri, seleziona una singola attività cliccando sulla card corrispondente per visualizzarne i dettagli completi.

Al termine, passare alla prossima attività.

A.3.3 Attività 3: Generazione esperienza unica

Adesso ti verrà data la possibilità di utilizzare una delle due funzionalità principali della piattaforma: la generazione automatica delle attività o la combinazione manuale.

- Caso 1 - Generazione automatica: Se scegli questa opzione, il sistema genererà automaticamente una serie di attività combinandole in un'unica proposta.
- Caso 2 - Combinazione manuale: Se invece preferisci procedere manualmente, ti invito a selezionare le attività tra quelle ottenute dai filtri precedenti e utilizzare l'opzione di combinazione. Successivamente, potrai vedere come le attività si uniscono in un'unica proposta personalizzata.

Una volta completata la procedura, otterrai una proposta che ti chiedo di valutare.

Vedere se il soggetto ha difficoltà nel capire come procedere, inoltre prestare attenzione alla scelta effettuata tra le due casistiche.

Al termine, passare alla prossima attività.

A.3.4 Attività 4: Rigenerazione e salvataggio attività

Adesso ci troviamo nella schermata in cui viene mostrata l'attività finale. Oltre a visualizzarne i dettagli è possibile far rigenerare la descrizione o salvarla nelle attività personali. Ti chiedo quindi di provare ad effettuare queste due azioni tramite gli appositi bottoni.

Vedere se il soggetto ha difficoltà nell'espletare le azioni, inoltre notare se, nella rigenerazione, specifica eventuali richieste tramite l'apposito form.

Al termine, passare alla prossima attività.

A.3.5 Attività 5: Attività personali

Per quest'ultima attività, ti chiedo di navigare nella sezione dedicata alle tue attività personali, dove potrai visualizzare tutte le proposte salvate. Assicurati di poterle scorrere e visualizzare correttamente.

A.4 Debriefing

Siamo arrivati alla fine del test. Ti ringrazio per aver completato tutte le attività richieste. Come menzionato all'inizio, ora ti chiedo gentilmente di compilare il questionario SUS e rispondere a qualche breve domanda per raccogliere il tuo feedback.

Ecco alcune domande che ti farò al termine del questionario:

1. Come ti è sembrato il test?
2. Ritieni che ci siano funzionalità dell'app che potrebbero essere migliorate?
3. Pensi che dovrebbero essere aggiunte ulteriori funzionalità?

4. Il processo di selezione e combinazione delle attività ti è sembrato intuitivo o cambieresti qualcosa?
5. Hai domande o chiarimenti riguardo al questionario?

Grazie ancora per il tuo prezioso contributo. Il tuo feedback sarà fondamentale per migliorare il prodotto finale. Buona giornata!

Appendice B

Questionario SUS

In questa appendice si riporta un questionario SUS compilato da un partecipante a rappresentazione del numero totale. Si calcola poi su di ognuno un punteggio indicante la valutazione complessiva dell'usabilità del prototipo.

| Domande | | Voto del partecipante | Punteggio risultante |
|---------------------------|---|-----------------------|----------------------|
| 1. | Penso che mi piacerebbe utilizzare questa interfaccia frequentemente | 4 | 3 |
| 2. | Ho trovato l'interfaccia poco intuitiva | 2 | 3 |
| 3. | Ho trovato l'applicazione molto semplice da usare | 4 | 3 |
| 4. | Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'interfaccia | 1 | 4 |
| 5. | Ho trovato le varie funzionalità dell'interfaccia ben integrate | 4 | 3 |
| 6. | Ho trovato incoerenze tra le varie funzionalità dell'interfaccia | 2 | 3 |
| 7. | Penso che la maggior parte delle persone potrebbero imparare ad utilizzare facilmente l'interfaccia | 4 | 3 |
| 8. | Ho trovato l'interfaccia lenta e meccanica | 2 | 3 |
| 9. | Ero a mio agio durante l'utilizzo dell'interfaccia | 5 | 4 |
| 10. | Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'interfaccia | 1 | 4 |
| Valutazione totale | | | 82,5 |

Tabella B.1: Valutazione SUS per il partecipante U1

Bibliografia

- [1] AgendaDigitale. *PNRR, verso il turismo 4.0: ecco le misure e gli obiettivi*. <https://www.agendadigitale.eu/cultura-digitale/competenze-digitali/pnrr-verso-il-turismo-4-0-ecco-le-misure-e-gli-obiettivi/>. 2022 (cit. a p. 1).
- [2] Manfred Lenzen, Ya-Yen Sun, Futu Faturay, Yuan-Peng Ting, Arne Geschke e Arunima Malik. «The carbon footprint of global tourism». In: *Nature Climate Change* 8.6 (2018), pp. 522–528. ISSN: 1758-678X. DOI: 10.1038/s41558-018-0141-x (cit. a p. 2).
- [3] Rabiya Diouf, Edouard Ngor Sarr, Ousmane Sall, Babiga Birregah, Mamadou Bouso e Sény Ndiaye Mbaye. «Web scraping: state-of-the-art and areas of application». In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. 2019, pp. 6040–6042 (cit. a p. 5).
- [4] Crummy. *Beautiful Soup Documentation*. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. 2023 (cit. a p. 8).
- [5] Scrapy. *Scrapy Documentation*. <https://docs.scrapy.org/en/latest/>. 2023 (cit. a p. 9).
- [6] Selenium. *Selenium dev*. <https://www.selenium.dev/>. 2023 (cit. a p. 10).
- [7] Apify. *Apify: Full-stack web scraping*. <https://apify.com/>. 2023 (cit. a p. 11).
- [8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov e Luke Zettlemoyer. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL]. URL: <https://arxiv.org/abs/1910.13461> (cit. a p. 20).
- [9] Gokul Yenduri et al. «Gpt (generative pre-trained transformer)—a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions». In: *IEEE Access* (2024) (cit. a p. 22).

- [10] Ioannis Samoladas, Georgios Gousios, Diomidis Spinellis e Ioannis Stamelos. «The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation». In: *Open Source Development, Communities and Quality*. A cura di Barbara Russo, Ernesto Damiani, Scott Hissam, Björn Lundell e Giancarlo Succi. Boston, MA: Springer US, 2008, pp. 237–248. ISBN: 978-0-387-09684-1 (cit. a p. 32).
- [11] Google. *Google AI Guidebook*. <https://pair.withgoogle.com/guidebook/> (cit. a p. 44).
- [12] Microsoft. *Microsoft HAX Toolkit*. <https://www.microsoft.com/en-us/haxtoolkit/> (cit. a p. 44).
- [13] *Python*. <https://docs.python.org/3/> (cit. a p. 45).
- [14] *JavaScript*. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (cit. a p. 45).
- [15] *React*. <https://react.dev/learn> (cit. a p. 45).
- [16] Jakob Nielsen. *Why You Only Need to Test with 5 Users*. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (cit. a p. 80).
- [17] *Questionario SUS*. <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/> (cit. a p. 82).