Università degli Studi di Salerno

Progetto di Fondamenti d'Intelligenza Artificiale

ExploreWorld

Partecipanti:

Nome	Matricola
Alfieri Riccardo	05121 16533
Mascolo Francesco	05121 17352

Link repository GitHub:
https://github.com/Riccardoalfieri2003/ExploreWorld

Sommario

Introduzione

Quante persone vorrebbero viaggiare? Quante di queste persone effettivamente viaggiano? Quante di queste viaggiano l'intero pianeta? Secondo gli studi e analisi di NBC Bay Area, all'incirca solo 400 persone nella storia hanno visitato tutti gli Stati del mondo.

Da un punto di vista tecnico, la superficie complessiva della Terra è di 509 600 000 km^2.
Volendo effettuare una separazione, la superficie "calpestabile" è di 148 326 000 km^2, mentre la superficie ricoperta da acqua è di 361 740 000 km^2.

Supponendo che una persona viva in media 80 anni, vorrebbe dire che per esplorare ogni angolo della terra emersa, si dovrebbe viaggiare ad una velocità di circa 211.65 km/h per tutta la vita (volendo attraversare anche solo uno dei chilometri quadrati in linea retta).
Ovviamente ciò è impossibile.

Come risolvere il problema? Seguire la nostra pagina Instagram @ExploreWorld.

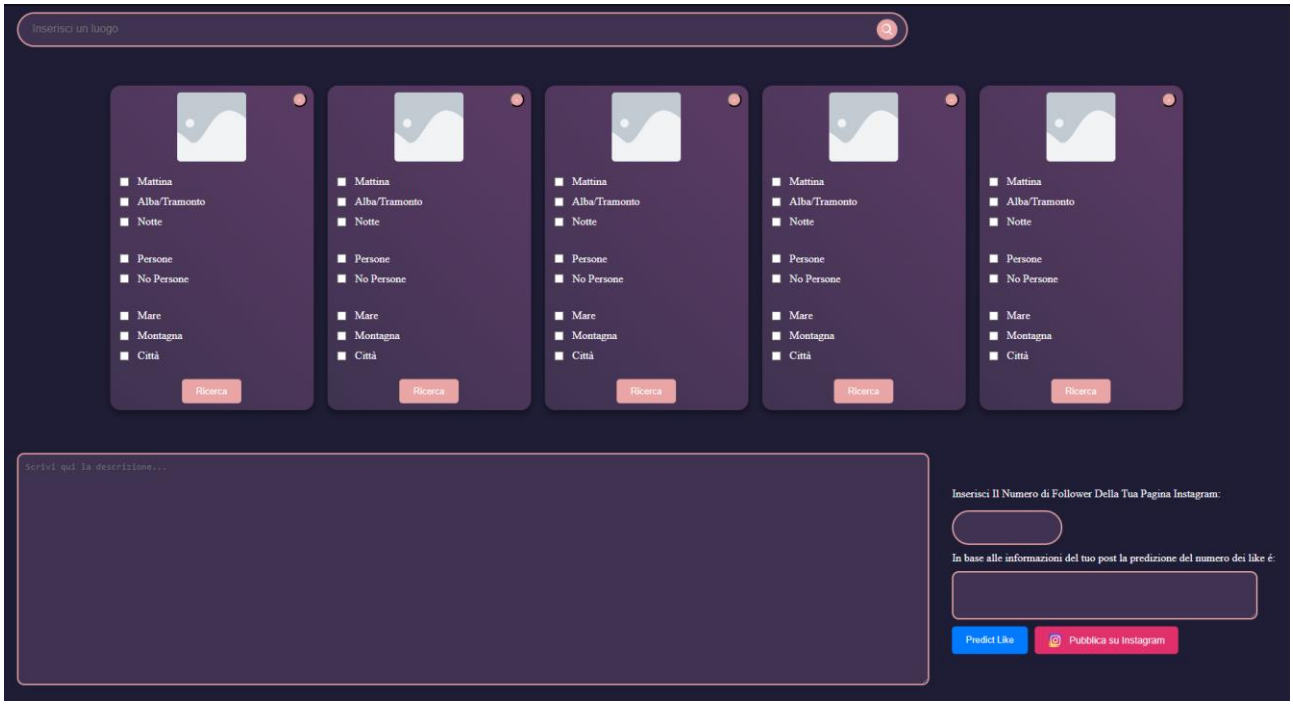
ExploreWorld nasce per diversi motivi:
- Condividere con altre persone luoghi che abbiamo visitato
- Far conoscere luoghi particolari e poco conosciuti ad altri utenti
- Mettere sotto sfida la propria conoscenza a livello geografico

L'obiettivo finale? Esplorare tutto il mondo.

ExploreWorld è una piccola applicazione che permette di effettuare automaticamente i post su Instagram con poche istruzioni.
Una volta deciso il luogo, automaticamente vengono fornite delle foto, e una volta "setacciate" tramite l'intelligenza artificiale è possibile postarle su Instagram.
Inoltre, per assicurarci che il nostro post raggiunga più persone possibili, è possibile anche avere una stima sul numero di like che il post potrebbe ottenere.

Link all'articolo di NBC Bay Area:
https://www.nbcbayarea.com/news/local/bay-area-proud/193-countries-san-jose-2nd-grade-teacher-becomes-rare-traveler-to-visit-every-country-in-the-world/3521511/#:~:text=Although%20there%20is%20no%20official,have%20gone%20into%20outer%20space.

Interfaccia e funzionamento



Il luogo che si desidera si deve inserire all'interno della barra di ricerca e successivamente si preme la lente di ingrandimento per ottenere le immagini. Una volta ottenute apparirà un simbolo di check.

Per aggiungere le foto successivamente postate si deve premere sul simbolo "+", con un massimo di 5 foto. Per ogni foto si può scegliere fra le varie opzioni:
- Presenza di persone o non
- Tempo della giornata
- Presenza di Mare/Montagna/Città

Nella parte sottostante è presente un'area di testo per aggiungere la descrizione al post. In modo da poter ottenere una stima dei like al post, è presente un pulsante che permette di effettuare tale azione. Per ottenere una stima più precisa è possibile inserire anche il numero di follower della propria pagina.

Infine è presente il pulsante che permette la condivisione del post, sulla pagina Instagram @ExploreWorld, delle foto selezionate con la descrizione inserita.

Struttura del progetto

Il progetto è strutturato in diverse cartelle.

1.1. Datasets

Il progetto ExploreWorld possiede due tipi di dataset: dataset di immagini per il riconoscimento di elementi e dataset contenente stringhe per il calcolo di like tramite informazioni di post Instagram.

1.1.1. Datasets immagini

Ogni dataset di immagini è suddiviso in 2 sottocartelle: training e test.
Per effettuare la suddivisione in queste 2 cartelle si è utilizzata una suddivisione 80-20 circa per tutti i dataset.

Dataset binari
All'interno di ogni cartella sono presenti 2 gruppi di immagini: uno per far riconoscere l'elemento in questione, e il secondo non possiede tale elemento all'interno delle immagini.

Dataset multiclasse
All'interno di ogni cartella, a differenza della classificazione binaria, sono presenti un determinato numero di cartelle per quante sono le classi.

Tutte le immagini all'interno di questi dataset sono di dimensioni 244x244 px.
L'obbiettivo principale di questi dataset è quello di riconoscere determinati elementi all'interno di foto.

PhotoOrNotDataset
Il primo step per il riconoscimento di elementi, è assicurarsi che un'immagine sia una foto.
Sia all'interno della cartella test che train sono presenti due cartelle:
0_photos
1_other

Nella cartella 0_photos sono contenute immagini che rappresentano foto. Per ottenerle si è effettuato lo scraping di immagini ricercando:
photos, aniaml photos, nature photos, people photos, New York, tropical cities
e utilizzando foto ricavate dallo scraping per altri dataset

Nella cartella 1_other sono contenute immagini relative principalmente a disegni, includendo disegni a colori, disegni in bianco e nero, mappe e grafici (che possono essre considerati dei disegni). È stata esclusa la categoria delle immagini CGI.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
Drawings: drawimgs, drawings color, disegni e disegni a colori, black and white drawings
Maps: maps, game maps
Charts: charts

PeopleOrNotDataset
All'interno di questo dataset sono presenti immagini per il riconoscimento di persone all'interno delle foto.
Sia all'interno della cartella test che train sono presenti due cartelle:
0_noPeople
1_people

Questo dataset, oltre allo scraoing effettuato, è stato "sostenuto" da un dataset già esistente, ripreso da Kaggle al seguente link
https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset

Nella cartella 0_noPeople sono contenute immagini che non contengono esseri umani, e che potrebbero ricondurre ad esseri umani (a causa del sottodimensionamento delle immagini).
Per ottenerle si è effettuato lo scraping di immagini ricercando:
nature photos, city photos, aniaml photos, trees, pillars, tall vases, exotic trees, palm trees, observatories

Nella cartella 1_people sono contenute immagini rappresentanti esseri umani, di diversa etnia, età e in posizioni diverse.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
white people photo, black people photo, indian people photo, chinese people photo, people in the background, children, people from behind, side profile person, crowds
e utilizzando foto ricavate dallo scraping per altri dataset

TimeDataset
Dataset multiclasse utilizzato per il riconoscimento del periodo della giornata.
Le classi presenti: Giorno, Alba/Tramonto, Notte e Aurora boreale.
Sono utilizzati dataset contenenti "tempi" normali, come giorno, alba/tramonto e notte.
E sono utilizzati anche dataset per "tempi" rari, come ad esempio l'aurora boreale, che ricade nel caso "Notte".

Nella cartella Giorno sono contenute immagini rappresentanti mattinate con il Sole.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
foto in mattinata, *sunny city photos*, *sunny nature photos*, *sunny towns*, *sunny weather*, *sunny mountains*

Nella cartella Alba/Tramonto sono contenute immagini rappresentante tale pate della giornata.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
sunset photos, *city sunset*, *sunset nature*, *sunset pictures*, *crepuscolo, citta crepuscolo*

Nella cartella Notte sono contenute immagini rappresentanti tale pate della giornata. Per ottenerle si è effettuato lo scraping di immagini ricercando:
Night, city at night, foto di notte, night nature photos, starry sky

Nella cartella Aurora Boreale sono contenute immagini rappresentanti tale fenomeno atmosferico. È stato deciso di implementare anche questa classe siccome, in modelli precedenti non contenenti l'aurora boreale, il modello restituiva predizioni errate.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
Aurora Borealis, Lapponia aurora borealis, red aurora borealis

MareDataset, MontagneDataset e CittaDataset
Tutti e 3 sono dataset binari, ognuno contenente immagini del relativo elemento da riconoscere.

Nel dataset **MareDataset** sono contenute immagini rappresentanti il mare e oceano.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
mare, sea, spiaggia, beaches

Nel dataset **MontagneDataset** sono contenute immagini rappresentanti montagne.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
Mountains, montagne, green mointains, montagne senza neve

Nel dataset **CittaDataset** sono contenute immagini rappresentanti città e villaggi.
Per ottenerle si è effettuato lo scraping di immagini ricercando:
Cities, towns, snowy towns, towns at night, cities at night

Per ottenere tali dataset è stata effettuata una combinazione delle stesse immagini. Includendo ed escludendo immagini per i differenti dataset in base ai contenuti delle immagini delle ricerche stesse. È stato ricercato:
Cities on beaches, *Mountains on sea*, *City squares, cities on mountains*

1.1.2. Dataset Post Instagram

Nel progetto vengono utilizzati due dataset strettamente collegati, ma con finalità diverse.

Il primo, chiamato **Instagram_Posts_Full**, è una versione completa che raccoglie informazioni dettagliate sui post di Instagram.
Ogni riga del dataset rappresenta un post e contiene variabili come il nome dell'account, il luogo associato al post (se presente), il numero di "Mi piace" ricevuti, il testo completo della descrizione, la data e l'orario di pubblicazione, il numero di follower al momento del post, e altri dettagli relativi al contenuto.
Ad esempio, include il numero di immagini presenti, il conteggio delle parole, il numero di emoticon utilizzati, e il numero di menzioni e hashtag. Inoltre, il dataset riporta informazioni sull'associazione del post a un luogo, specificando se è esplicitamente indicato e quante volte viene menzionato nel testo.
Questo dataset, fornisce una panoramica ricca e completa, ideale per analisi esplorative e per costruire nuove caratteristiche utili al modello.

Il secondo dataset, denominato **Instagram_Posts**, rappresenta una versione modificata del primo. In questa versione, sono stati inclusi solo i dati strettamente necessari per addestrare il modello di predizione del numero di "Mi piace". Sono state rimosse informazioni superflue come il nome dell'account e la descrizione completa, mentre sono state mantenute le variabili utili, tra cui il luogo, i "Mi piace", i dettagli sul contenuto del post (numero di parole, emoticon, menzioni, hashtag) e il numero di follower.
Questa riduzione permette al dataset di essere più leggero e mirato, ottimizzando le prestazioni del modello.

Questi due dataset, pur avendo lo stesso numero di osservazioni, rispondono a necessità diverse: uno offre un quadro completo per analisi approfondite, mentre l'altro è strutturato per essere direttamente utilizzabile nel processo di addestramento del modello di predizione.

La relazione tra i due dataset evidenzia un passaggio naturale dalla raccolta e analisi iniziale dei dati alla loro preparazione per l'applicazione pratica.

1.2. GoogleScraper
Tale cartella contiene file dedicati allo scraping di Google Maps e Immagini tramite Selenium.
Le funzioni presenti all'interno dei file sono utilizzate per:
- Ottenere il luogo specifico da ricercare
- Ottenere le coordinate del luogo
- Ottenere gli URL delle immagini di tale luogo

1.3. InstagramScraper
Tale cartella contiene file dedicati allo scraping di dati su Instagram tramite Selenium.
Le differenti funzioni raccolgono dati che verranno successivamente manipolati per essere gestiti al meglio all'interno del dataset.

Le informazioni raccolte riguardano:
- Numero di follower della pagina che ha postato
- Il luogo, se presente, allegato a tale post
- Il numero di immagini presenti all'interno del post
- La descrizione
- Il numero di like, che verrà successivamente usato come dato da predire
- Giorno e ora di quando il post è stato pubblicato

1.4. Models
All'interno di questa cartella sono presenti i modelli utilizzati all'interno del progetto. Dalla natura dei dataset stessi, sono presenti due tipi di modelli.

Modelli per riconoscimento di immagini
Questi modelli sono addestrati sui dataset di immagini per riconoscere determinati elementi all'interno delle foto.

Per addestrare i modelli sono state utilizzate delle reti neurali con caratteristiche simili.
Per dataset con elementi più difficili da riconoscere, come quello delle persone, è stato utilizzato un modello che va a lavorare con 20 epoche. I restanti modelli lavorano con solo 10 epoche.

Cambiano anche come i modelli vanno ad addestrarsi sui dati. Seguendo il ragionamento precedente, modelli più complessi vengono addestrati su un batch size di dati pari a 16, poiché cercano di rilevare feature più complesse e precise. Nei restanti modelli, il bathc size è pari a 32.

Ulteriore differenza si ha tra i modelli binari e multiclasse con l'activation.
Nel caso dei modelli binari si è utilizzato una activation "sigmoid", nei casi dei modelli multiclasse, l'activation risulat essere "softmax".

Altra differenza si ha nel modo in cui sono stati gestiti i dati. Per tutti i modelli è stata utilizzata la tecnica della data augmentation, per ottenere un dataset più ampio nel momento di training e testing.
In generale, sono state applicate tecniche di:

- rescale=1.0/255: Normalizza i valori dei pixel tra 0 e 1
- rotation_range=10: Rotazioni casuali
- width_shift_range=0.2: Traslazioni orizzontali
- height_shift_range=0.2: Traslazioni verticali
- shear_range=0.2: Trasformazioni angolari
- zoom_range=(0.8,1.2): Zoom casuale
- fill_mode='nearest': Riempie i pixel dopo la trasformazione
- horizontal flip=True: Ribaltamenti orizzontali
- brightness_range=[0.8, 1.2]: Cambi di luminosità

Per ogni modello è poi presente un file che permette la valutazione del modello, per conoscere le sue prestazioni.

Per ogni modello, inoltre, è anche presente un file che permette di recuperare la classe predetta. Alcuni di questi modelli presentano, per migliorare le prestazioni stesse, modifiche della probabilità delle predizioni effettuate.

Modelli per predizione like
Questi modelli sono addestrati su Dataset di informazioni relative ad un post in modo da poter rendere ogni informazione utile nella predizione dei like. Per affrontare in modo accurato le differenze tra account con molti follower e account con follower ridotti, il sistema è stato progettato con due modelli distinti, ciascuno ottimizzato per una specifica fascia di follower.

Per account con più di 47600 follower, viene utilizzato un modello di **Random Forest Regressor** ovvero un modello di ensemble basato su alberi di regressione decisionali. È stato stabilito che la "Foresta" deve contenere al massimo 100 alberi decisionali con la variabile nestimators=100 in modo da non aumentare i costi computazionali di un modello già complesso. Il modello è stato addestrato con l'80% dei dati e testato con il restante 20%.

Per account con 46700 follower o meno, il sistema si avvale invece di un modello di **Gradient Boosting Regressor**. Questo modello utilizza un approccio iterativo in cui alberi decisionali vengono costruiti in sequenza per correggere gli errori dei modelli precedenti.
La configurazione prevede l'uso di 200 alberi decisionali con un learning rate di 0.1, bilanciando la velocità di apprendimento e la stabilità del modello. Per migliorare ulteriormente le predizioni, il target (numero di "Like") è trasformato logaritmicamente da una funzione di

compensazione durante il preprocessing, e le caratteristiche numeriche sono standardizzate tramite uno StandardScaler.
Questo modello si dimostra particolarmente efficace nel catturare relazioni non lineari e nel gestire la variabilità dei dati.
Inoltre, include una compensazione specifica per account con pochi follower, migliorando la robustezza delle predizioni anche in casi complessi.

I due modelli sono integrati in un unico file, che coordina l'intero processo di predizione. La funzione principale, seleziona dinamicamente quale modello utilizzare in base al numero di follower forniti nel dato di input. Se il numero di follower è superiore a 47600, viene utilizzato il modello Random Forest; altrimenti, viene selezionato il modello Gradient Boosting. Questa funzione garantisce una gestione flessibile e ottimizzata delle predizioni, adattandosi automaticamente al caso specifico.

1.5.Interfaccia
L'interfaccia è realizzata tramite un'applicazione Flask in python.
La struttura seguita è caricare contenuti statici, come lo stile, in una cartella che prende il nome *static* e la pagina che deve essere mostrata e modificata, nella cartella *templates*. L'applicazione app.py gestisce entrambe permettendo il collegamento delle funzioni python per utilizzarle in html/js.

Allo startup dell'applicazione, viene deployata sul server localhost la pagina index.html, con relativo script js, e stile style.css

1.6.Map
Questa porzione del codice è dedicata a mostrare i luoghi che sono stati postati su Instagram, e testare le competenze a livello geografico del gestore della pagina.

Per gestire i luoghi, essi vengono memorizzati all'interno di un file testuale. In particolare, vengono memorizzati:
- Nome del luogo
- Coordinate
- Link delle immagini che sono state postate su @ExploreWorld.

Quando sono recuperati tutti i luoghi e relative informazioni, è possibile utilizzare librerie per convertire le coordinate in punti della mappa.
Utilizzando un algoritmo density-base, è possibile inserire all'interno di un unico cluster i punti vicini tra di loro.
Ogni punto sulla mappa è "esplorabile", facendo vedere il luogo e le foto ad esso relativo.

Relazioni col modello Crisp-DM

Il progetto è stato realizzato seguendo la struttura del modello Crisp-DM.

Business Understanding

All'interno di tale fase vengono presi in considerazione obiettivi e requisiti per sviluppare ExploreWorld.

Il funzionamento di ExploreWorld è basato su una parte di ricerca delle foto e di una seconda parte riguardante il post su Instagram.
Di conseguenza, come business success criteria, c'è la necessita rispettivamente di

- Dare la possibilità all'utente dell'applicazione di scegliere un luogo e le foto ad esso associate, provenienti da Google Maps e Google Immagini
- Effettuare il post delle foto e descrizione scelta su Instagram, tramite un account dedicato

Il tutto deve essere sostenuto da modelli di intelligenza artificiale per aiutare l'utente.
In particolare, devono essere presenti modelli per la scelta della foto e modelli per predire quale possa essere il riscontro di altri utenti su Instagram del post effettuato.

Data Understanding

Lavorando con due tipi di modelli, c'è la necessita di prendere dati di nature differenti.

Per il modello che lavora ad immagini, bisogna prendere immagini riguardanti elementi da riconoscere all'interno delle foto.
Gli elementi che si è deciso si debbano riconoscere sono:
Persone, mare, montagne, città e tempo della giornata. Un ulteriore modello è presente per riconoscere se un'immagine è una foto o no.
Di conseguenza si deve effettuare la ricerca di tali immagini su Google Immagini.

Per il modello per predizione di like, bisogna recuperare dei post su Instagram per effettuare le operazioni.
Gli elementi recuperati da ogni post sono:
Nome dell'utente, numero di follower dell'utente, like, descrizione, luogo e data del post.

Data Preparation

Seguendo la linea di due tipi di modelli differenti, i dati sono preparati in due modi diversi

Per i modelli per riconoscimento di elementi all'interno di immagini, è stato effettuato lo scraping di Google Immagini tramite il file scraping_for_dataset.py
Tale file permette di inserire l'oggetto della ricerca, e verranno prese tutte le immagini (che il programma riesce a prendere non generando eccezioni).
Ognuna di queste immagini verrà successivamente ridimensionata a 244x244 px, dimensione di immagini con cui si andrà a lavorare con i modelli.
Una volta recuperate tutte le immagini, sono rimosse quelle con estensioni "particolari", come .webp, .avif e altre. I formati accettati sono .jpg, .jpeg e .png.

I dati recuperati, per ogni dataset, sono suddivisi in dati di train e test, rispettando una suddivisione in media dell'80-20 %.

Per i modelli per la regressione dei like vengono pre-elaborati, ovvero, sono stati inclusi solo i dati strettamente necessari per addestrare il modello di predizione del numero di "Mi piace". Sono state rimosse informazioni superflue come il nome dell'account e la descrizione completa, mentre sono state mantenute le variabili utili, tra cui il luogo, i "Mi piace", i dettagli sul contenuto del post (numero di parole, emoticon, menzioni, hashtag) e il numero di follower.

Questa riduzione permette al dataset di essere più leggero e mirato, e soprattutto permette di rendere i dati pronti per la modellazione.

Modeling

Per i modelli mirati al riconoscimento di elementi in immagini, sono state utilizzati delle reti neurali convoluzionali (CNN) simili tra di loro.

Tutti i dati sono soggetti a trasformazioni di data augmentation, per aumentare le prestazioni dei modelli stessi. Alcuni dei dati di training possiedono un batch size differente in base alla difficoltà del riconoscimento dell'elemento stesso.

I modelli, addestrati su tali dati, lavorano in moda con 10 epoche. I modelli che lavorano con immagini più complesse lavorano anche con 20 epoche.

Per i modelli di regressione orientati alla predizione dei like, i dati vengono suddivisi in 80% training set e 20% testing set.

Per i post che possiedono il numero di follower (dell'utente che ha pubblicato) che supera i 47600 viene utilizzano il modello RandomForestRegression.
Nel caso in cui il numero di follower sia inferiore o uguale a 47600 viene utilizzato GradientBoostingRegression. Questo perché il numero di follower rappresenta l'attributo con più potenza predittiva e porterebbe ad ottenere valori "nulli" quando si ha a che fare con pochissimi follower.

Evaluation

Per ogni modello basato su immagini, è presente un file a parte, mirato solo ed esclusivamente all'evaluation del modello.
Tale valutazione viene effettuata sui dati di testing. All'interno di tale file è presente la confusion matrix, con il proprio classification report.

Di sotto sono elencate le accuracy dei modelli:
Accuracy dei modelli:
- CittaModel: 0.8
- MareModel: 0.83
- MontagneModel: 0.77
- PeopleModel: 0.79
- PhotoOrNotModel: 0.87
- TimeModel: 0.86

All'interno di ogni modello, quindi sia nel caso di utilizzo del modello di regressione Gradient Boosting che nel caso di utilizzo del modello di regressione Random Forest sono state applicate le seguente misure di valutazione:

- MAE (Mean Absolute Error): 1640.318
- RMSE (Root Mean Squared Error): 2415.69

Deployment

Una volta che tutti i modelli sono funzionanti, e assicurateci che l'interfaccia è ben configurata, è possibile utilizzare l'applicazione.

Allo startup dell'applicazione vengono caricati tutti i modelli, e le funzioni saranno richiamate solo quando necessario.