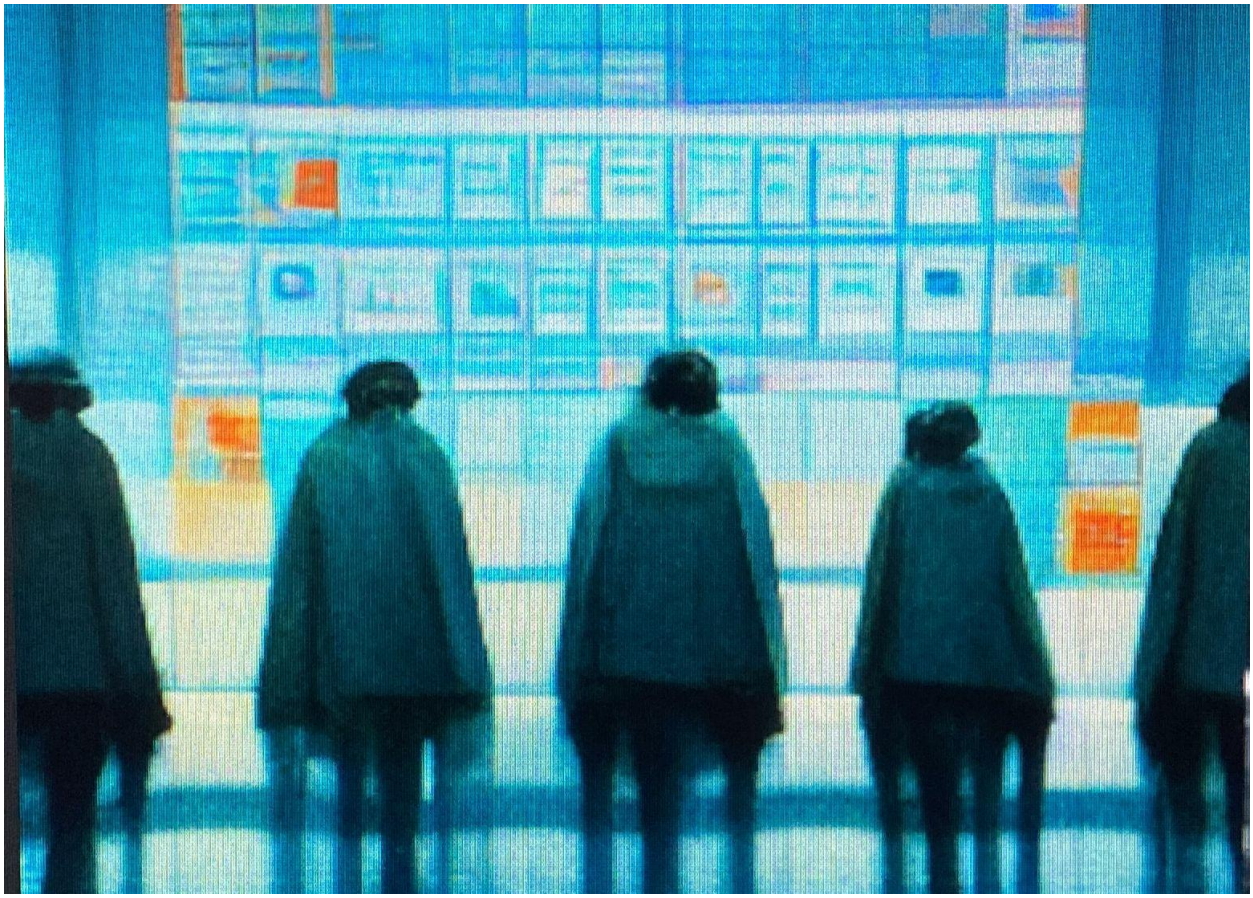
Progetto Tripadvisor
ANALISTI ANONIMI



Introduzione

Questo logo rappresenta gli "***Analisti Anonimi***". E' stato generato grazie ad un' Intelligenza Artificiale chiamata Midjourney che permette di generare immagini dando in input delle caratteristiche.

Descrizione delle modalità di lavoro

L'organizzazione del progetto è partita da un ***brainstorming*** iniziale di tutti i componenti degli "***Analisti Anonimi***", andando a definire come sarebbe stato svolto il progetto.

Abbiamo successivamente creato un ***server Discord***, dove abbiamo avuto modo di coordinarci tramite riunioni vocali e presentazioni video per la realizzazione del progetto. Inoltre, per rimanere in costante contatto tra di noi, abbiamo creato anche un ***gruppo Whatsapp***, dove ci siamo tenuti quotidianamente aggiornati sull'andamento del progetto.

Per la maggior parte del progetto abbiamo lavorato in sincronia, tramite ***Google Colab***, che ci ha permesso di lavorare simultaneamente sullo stesso codice mentre eravamo collegati sul server ***Discord***. Quando è stato possibile, ci siamo anche incontrati di persona, a casa di qualcuno di noi o nelle aule studio dell'Università.

Anche per la creazione del file ***Word*** e della presentazione ***PowerPoint***, abbiamo lavorato in sincronia coordinandoci tramite il server Discord. Abbiamo utilizzato anche in questo caso dei servizi in Cloud offerti da Google, rispettivamente: ***Google Docs*** e ***Google Slides.***

Nella pratica il progetto è stato suddiviso in due fasi: nella prima abbiamo utilizzato ***Selenium*** per fare ***Web Scraping*** dal sito ***www.tripadvisor.it ,*** in modo da scaricarci i file html dei 5 ristoranti. Successivamente abbiamo abbandonato Selenium e siamo passati a ***Beautiful Soup.*** Questa libreria ci è servita per estrarre informazioni dai file html che abbiamo scaricato in precedenza ed analizzarli.

Importazione Librerie

Abbiamo utilizzato varie librerie. Le andiamo ad elencare fornendo una breve descrizione:

1. ***Selenium*** serve per l'automazione del browser. Permette di controllare, tramite un driver, un browser web in maniera automatizzata.
2. ***ChromeDriverManager*** facilità l'installazione e l'aggiornamento automatico dei driver Chrome necessari per controllare il browser tramite Selenium.
3. ***By*** è una classe in Selenium che fornisce metodi per identificare gli elementi della pagina web in base a diversi criteri come l'ID, il nome o la classe.
4. ***re*** è il modulo delle espressioni regolari in Python, che fornisce uno strumento per la manipolazione e il controllo di stringhe di testo.
5. ***cleantext*** è una libreria python per la pulizia di testo. Fornisce funzioni per rimuovere caratteri indesiderati. Nel nostro caso le emoji dalle recensioni.
6. ***WordCloud*** serve per generare "nuvole di parole" in cui la dimensione delle parole rappresenta la loro frequenza.
7. ***Matplotlib*** è stata utilizzata in concomitanza della libreria WordCloud, dato che ci ha permesso di visualizzare la "nuvola di parole", che è in tutti i sensi un grafico.
8. ***Time*** è una libreria Python per gestire il tempo, in questo caso è stata utilizzata per creare dei piccoli intervalli nei processi di web-scraping in modo da non dare sospetti allo "snort" di TripAdvisor.
9. ***Pandas*** è una libreria multi-funzione, in questo progetto è stata utilizzata per creare dei DataFrame contenenti i dati dei ristoranti e delle recensioni per essere poi trasferiti nel database. Inoltre è stata utilizzata per creare il DataFrame delle parole con il relativo conteggio, che poi è stato utilizzato per la creazione delle WordCloud.
10. **Plotly.express** è una sotto-componenete della libreria Plotly che permette la creazione di grafici interattivi che danno la possibilità di visualizzare meglio dei dati. In particolare è stata utilizzata per mostrare la percentuale di ogni rating (1,2,3,4,5 stelle) delle recensioni per ogni ristorante.
11. ***SQLite3*** è stata utilizzata per la creazione del Database. Essa permette di creare, gestire e interrogare database SQLite utilizzando il linguaggio SQL.

Start Driver e Download Pagine HTML

La funzione **start_driver()** utilizza una componente della libreria Selenium e si occupa dell'attivazione del driver web. Nel nostro caso utilizziamo come browser web Chrome e con la libreria driver manager ci scarichiamo il driver aggiornato alla corrispettiva versione del nostro browser

La funzione **get_html(driver)** prende come parametro il driver web inizializzato in precedenza. Definiamo poi due liste, la prima contiene il numero massimo di recensioni estraibili per ogni ristorante (comunque limitato superiormente a 400), parametro che andrà in una stringa formattata successivamente.

la seconda contiene i codici univoci di ogni ristorante, sempre da inserire nei link per aprire le pagine dei vari ristoranti.

A questo punto cicliamo su tutti e 5 i ristoranti. Per ognuno, prima apriamo la pagina iniziale in cui troviamo già 10 recensioni e che quindi ci scarichiamo con il metodo pagesource di Selenium.. Nel frattempo, se si presenta il pulsante per accettare i cookies cerchiamo di accettarlo. Dopodichè cicliamo da 10 al numero massimo di recensioni per ogni ristorante a intervalli di 10. Inseriamo questo valore nella stringa formattata sottostante che tramite il metodo get raggiunge i link con le varie pagine delle recensioni e le scarica una ad una. Infine le varie pagine sono unite con la funzione join e il file hmtl unito di ogni ristorante viene salvato nella directory sempre in formato html. Concludiamo chiudendo il driver.

Estrazione dei dati delle recensioni

La funzione **get_all_reviews** si occupa di raccogliere i dati delle recensioni di un ristorante a nostra scelta, indicato con n (n che va da 1 a 5).

Procediamo aprendo il corrispettivo file e analizzandolo con beautiful soup. Con quest'ultimo, tramite il metodo find_all troviamo tutti i contenitori con il tag div che contengono le informazioni di ogni recensione. Useremo questi contenitori per cercarci all'interno gli altri elementi tramite le funzioni di acquisizioni successive.

A questo punto, per ogni contenitore delle recensioni, creiamo un dizionario che conterrà come chiavi, Id del ristorante, autore della recensione, rating, data della recensione, titolo e testo della recensione, e come valori gli output delle corrispettive funzioni di acquisizione. Appendiamo il dizionario di ogni recensione alla lista dei dizionari e quando il ciclo è finito e quindi le recensioni sono state analizzate tutte, chiudiamo anche l'html del ristorante n e ritorniamo in output proprio la lista di dizionari. questa ci servirà in seguito sia per la realizzazione delle wordcloud sia per la creazione dei database.

→ **get_rate(r)**: Questa funzione restituisce il rating (numero di stelle) associato a una recensione r. La funzione cerca un elemento con una classe specifica all'interno della recensione. A seconda della classe trovata, viene assegnato un valore numerico corrispondente al numero di stelle (da 1 a 5). Se la classe non viene trovata, viene restituita una stringa vuota.

→ **get_title(r)**: Questa funzione restituisce il titolo di una recensione r. La funzione cerca un elemento con una classe specifica che contiene il titolo della recensione. Se l'elemento non viene trovato, viene restituita una stringa vuota. Altrimenti, viene estratto il testo contenuto nell'elemento, pulito dagli spazi iniziali e

finali e dalle emoji utilizzando la libreria cleantext. Il titolo pulito viene restituito come output della funzione.

→ **get_date(r)**: Questa funzione restituisce la data associata a una recensione r. La funzione cerca un elemento con una classe specifica che contiene la data della recensione. Se l'elemento non viene trovato, viene restituita una stringa vuota. Altrimenti, viene estratto il testo contenuto nell'elemento e viene tenuta solo la parte relativa alla data (presumendo che ci sia una formattazione fissa). La data viene restituita come output della funzione.

→ **get_review(r)**: Questa funzione estrae il testo della recensione r. La funzione cerca un elemento con una classe specifica che contiene il testo della recensione. Il testo viene estratto come semplice testo all'interno dell'elemento, inclusa l'eventuale parte nascosta delle recensioni troppo lunghe. Viene rimossa la stringa "Più" che potrebbe essere presente per espandere le recensioni lunghe. Infine, il testo viene pulito dalle emoji utilizzando la libreria cleantext e viene restituito come output della funzione.

→ **get_author(r)**: Questa funzione restituisce il nome dell'autore associato a una recensione r. La funzione cerca un elemento con una classe specifica che contiene il nome dell'autore della recensione. Se l'elemento non viene trovato, viene restituita una stringa vuota. Altrimenti, viene estratto il testo contenuto nell'elemento, pulito dagli spazi iniziali e finali, e viene restituito come output della funzione.

Queste funzioni possono essere utilizzate all'interno del codice principale per estrarre le informazioni desiderate dalle recensioni dei ristoranti e crearne dei dizionari che contengono tutte le informazioni associate a ciascuna recensione.

Estrazione dei dati dei ristoranti

Funzione generale

La funzione **get_info_rist** si occupa di raccogliere i dati dei ristoranti scelti, tramite un ciclo for. Inizialmente creiamo una lista vuota (all_info_rist) che conterrà tutte le informazioni che andremo ad estrarre dalle pagine tripadvisor, tramite gli ID dei ristoranti, estratti dalle pagine web. Procediamo creando un ciclo “for” in “range” da 1 a 6 che viene utilizzato per iterare attraverso i file HTML dei ristoranti, e ottenere le informazioni desiderate da ciascun file, automatizzando così il processo di estrazione delle informazioni per più ristoranti. A questo punto, creiamo un dizionario che conterrà come chiavi, Id del ristorante, nome, fascia di prezzo, tipo di cucina ed indirizzo. Appendiamo il dizionario di ogni ristorante alla lista dei dizionari e quando il ciclo è finito e quindi i dati dei ristoranti sono stati estratti dalle box, chiudiamo anche l’html del ristorante e ritorniamo in output proprio la lista di dizionari. questa ci servirà in seguito sia per la realizzazione delle wordcloud sia per la creazione dei database.

Funzioni di acquisizione

In seguito abbiamo definito le quattro funzioni contenute nel dizionario della funzione sopra descritta.

Partendo dalla funzione **get_name_rest(r):** per estrarre il nome del ristorante ed in seguito per la funzione **get_price(r):** per estrarre il prezzo dei vari ristoranti abbiamo preso un input un oggetto r, ed utilizzato il metodo “find” per estrarre su “r” il primo elemento delle box, esplicitando la classe del tipo di oggetto contenuto nel file html.

Analogamente abbiamo definito la funzione **get_address():** tramite una “find_all” sull’oggetto “r” per trovare tutti gli elementi di tipo “a”. E poiché l'indirizzo desiderato sembra essere il secondo elemento nella lista, viene estratto il testo dall'elemento address[1] utilizzando il metodo text.

In seguito abbiamo iterato la funzione **get_typecook()** utilizzando un ciclo if-else, così se la lista typecook_list è vuota (ovvero nessun elemento è stato trovato), viene assegnata una stringa vuota alla variabile type cook. Infine iterata la lista typecook_list e per ogni elemento, viene estratto il suo testo e aggiunto alla lista type cook, che verrà poi convertito in una stringa tramite un join.

WordCloud

Per WordCloud si intende una vera e propria ***“nuvola di parole”***. Tramite queste, è stato possibile visualizzare nuvole di parole contenenti le ***recensioni dei ristoranti***, suddivise per ogni ristorante con ***rating che va da 1 a 3 e da 4 a 5***. Il meccanismo di funzionamento delle word clouds è che vengono inserite le ***parole che vengono scritte più frequentemente***, e la loro grandezza all’interno della nuvola varia in base a quante volte sono state incontrate (***più la parola è grande più volte è stata contata***).

È stato possibile crearle grazie alla ***libreria WordCloud, MatPlotLib, Pandas, CleanText e re***. Le prime due sono essenziali, mentre le ultime altre tre sono servite per rendere il tutto più semplice e pulito, sia da realizzare che da visualizzare.

Andando più nel dettaglio, abbiamo creato la funzione ***generate_wordcloud*** che prende come input il parametro n, ovvero il numero che identifica il ristorante. Successivamente richiamiamo la funzione ***get_all_reviews*** che ci serve per prenderci tutte le recensioni del ristorante. Convertiamo tutte le recensioni in un

DataFrame Pandas, andando poi a dividerlo secondo le recensioni di rating >= a 4 e < di 4. Abbiamo poi concatenato in una lista le recensioni, separandole con lo spazio. Per rimuovere caratteri indesiderati (come ad esempio le ***emoji***) e prenderci solo le parole escludendo anche le punteggiature e gli spazi è stata utilizzata la libreria ***re***, che ci ha permesso di farlo utilizzando le espressioni regolari.

In seguito è stato utilizzato il file delle ***stopwords*** italiane, che contiene un insieme di parole che non aggiungono particolare significato al testo, che di conseguenza possono essere rimosse dalle recensioni.

Tramite dei cicli for andiamo ad "appendere" la lista di "***parole pulite***" che andremo poi ad utilizzare per generare le word clouds. Queste liste verranno poi convertite in set (perchè sappiamo che il set è una sequenza di elementi unici non ordinati).

Successivamente vengono creati due dizionari, in cui ***la chiave è una parola unica e il valore è il conteggio di quante volte appare quella parola*** nelle recensioni corrispondenti. Poi abbiamo convertito i dizionari in DataFrame, per poi rinominarli con la colonna "count".

In seguito andiamo a generare le word clouds per le recensioni con rating da 1 a 3 e da 4 a 5, e lo generiamo a partire dalle frequenze delle parole nel DataFrame.

Infine, i due DataFrame contenenti i 10 termini più frequenti per ciascun rating vengono restituiti come output della funzione, oltre i grafici che abbiamo generato in precedenza.

Commento e Analisi delle WordClouds per le recensioni con rating da 1 a 3 e da 4 a 5:

Word Clouds con le recensioni del rating da 1 a 3:

1. ***Hachi Ristorante Giapponese:***

Come possiamo vedere nell'immagine, nonostante queste parole provengano dalle recensioni con rating 1-2-3, non sono poi tutte così malvagie. Tuttavia ci sono parole come ***"glutine", "contaminata" o "giovincello"***. Questo può significare per esempio che un cliente celiaco è andato a mangiare nel locale senza però aver passato una buona esperienza, o anche che il personale non è stato professionale.

2. ***Owap Pizzeria Foria:***

Nel caso di questo ristorante ci sono critiche riguardanti il ***"prezzo"*** ed il fatto che la ***"pizza"*** sia ***"piccola"***. I clienti sembrano essere stati ***"delusi"***. Anche in questo caso però non mancano le parole con accezione positiva.

3. ***Pizza & Sfizi:***

Nel caso di questo ristorante, invece, troviamo una nuvola di parole più negative. Infatti si capisce subito, data la grandezza delle parole, che probabilmente i clienti abbiano aspettato molti ***"minuti"*** per avere una piccola ***"porzione".*** Anche il locale è stato criticato, di fatti "***l'ambiente"*** risulta ***"rumoroso"***,

4. ***Bellini 9:***

Nel quarto ristorante sembra che siano stati criticati principalmente gli ***ingredienti e le pietanze***, ma anche il prezzo ***"pagato"*** Inoltre anche il prezzo che è stato pagato risulta non conforme a ciò che i clienti hanno mangiato.

5. ***Tavernetta Colauri:***

Nella word cloud del quinto ristorante, nel complesso, i clienti sembrano aver apprezzato il ristorante definendolo nella ***"media",*** anche se la ***"qualità"*** dei prodotti e i ***"prezzi"*** non hanno superato le aspettative.

Word Clouds con le recensioni del rating da 4 a 5:

1. ***Hachi Ristorante Giapponese:***

In questa prima immagine le parole che saltano subito all'occhio sono ***"qualità", "servizio", " sushi", "ottimo", "personale" e "Giorgio",*** che molto probabilmente è il nome del proprietario. Nel complesso quindi possiamo farci un'idea di questo ristorante positiva, dove si mangia sushi di qualità, con personale gentile, professionale ed un ottimo servizio.

2. ***Owap Pizzeria Foria:***

Nel secondo ristorante, si percepisce come vengano servite delle ***"ottime" "pizze"*** con un ***"impasto" "leggero"*** ed ***"ingredienti"*** di ***"qualità"***. Anche il ***"personale"*** ed il ***"servizio"*** risultano valutati positivamente.

3. ***Pizza & Sfizi:***

Qui il ristorante offre ***"pizza"*** di ***"qualità"*** con ***"prezzi"*** giusti. Anche il ***"personale"*** ed il ***"servizio"*** risultano rispettivamente ***"gentile"*** e ***"veloce".***

4. ***Bellini 9:***

In questo ristorante invece i prodotti di ***"qualità"*** sono i ***"panini"***. Le parole ***"casa"***, ***"tradizione"*** e ***"signora"*** ci fanno intendere che in questo ristorante si mangia la vera ***"cucina" "napoletana".*** Sembrano essere apprezzate molto le ***"polpette"*** e la ***"provola"***.

5. *Tavernetta Colauri:*

Nel quinto ed ultimo ristorante spicca subito l'apprezzamento da parte dei clienti verso la ***"carne"*** di ***"qualità"***. Non mancano i complimenti al ***"locale"***, al ***"servizio"***, al ***"personale"*** e a ***"Nicola"***, il proprietario.

Database

Abbiamo in seguito creato un database che si occuperà di memorizzare i dettagli dei ristoranti e delle recensioni.

Per elaborare questa struttura ci siamo avvalsi della libreria **Sqlite3** che dà la possibilità di creare basi di dati leggerissime e molto veloci.

Inoltre abbiamo definito, durante la scrittura delle query **CREATE TABLE**, la condizione **IF NOT EXISTS** che implica l'esecuzione della query solamente se le tabelle in questione non esistono già.

Il db è composto da due tabelle: ristoranti e recensioni.

- **Ristoranti**: questa tabella è deputata a mantenere le informazioni di ogni ristorante. Queste sono:
    - ID Ristorante → L'ID univoco del ristorante assegnato da TripAdvisor;
    - Nome → Il nome del ristorante;
    - Prezzo → Il prezzo medio in € del ristorante;
    - Genere Cucina → La tipologia di cucina del ristorante, esempio: italiana, americana, thailandese, indiana, ecc.
    - Indirizzo → L'indirizzo del ristorante.
- **Recensioni**: questa tabella si occupa di mantenere le informazioni di ogni recensione. In tutto quelle scaricate sono 1577. Gli attributi di questa tabella sono:
    - ID Ristorante → L'ID univoco del ristorante assegnato da TripAdvisor;
    - Autore → Il nickname TripAdvisor dell'autore della recensione;
    - Titolo → Il titolo della recensione;
    - Rating → La valutazione da 1 a 5 stelle (visualizzate come pallini);
    - Data della Recensione → La data in cui è stata pubblicata la recensione su TripAdvisor;

- Testo → Il testo intero della recensione (comprendente anche eventuali parti nascoste dal bottone “Più”).

Per raccogliere tutti i dati prima di immetterli nella base dati abbiamo utilizzato un DataFrame pandas e in seguito ne abbiamo rinominato gli attributi per farli coincidere con quelli del db.

Punto importante da sottolineare è che l’inserimento nelle tabelle dei rispettivi dati è stato eseguito tramite il metodo “to_sql” della libreria pandas.

La funzione deputata alla creazione del database infine ritorna come output i due dataframe precedenti (dfRistoranti, dfRecensioni).

Grafici Recensioni

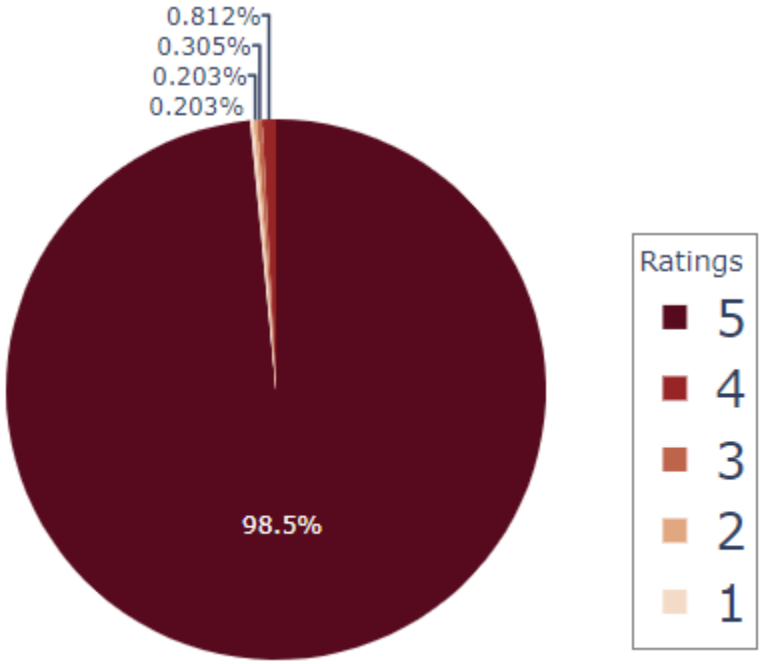
Per vedere la distribuzione delle valutazioni delle recensioni per ogni ristorante abbiamo deciso di creare, sfruttando la libreria plotly.express un grafico interattivo che possa mostrare la percentuale di ogni categoria di rating per ogni ristorante.

Ad esempio: qual è la percentuale di recensioni a 5 stelle in confronto a quelle a 4, 3 e così via.

In più sarà possibile passando con il mouse sopra il grafico visualizzare il numero di recensioni con quella particolare valutazione.

In più abbiamo sfruttato una particolare palette di colori che è integrata nella libreria.

E’ presente anche una legenda indicante le stelle del rating e il rispettivo colore.



Questo grafico è ripetuto per 5 volte (il numero di ristoranti presi in considerazione per l'analisi).

Ognuno di questi ha delle caratteristiche particolari che forniscono informazioni sul ristorante stesso. Ad esempio:

- Il ristorante in 4° posizione non presenta recensioni da 1 o 2 stelle, ma solo superiori.
- Il ristorante in 3 posizione presenta recensioni fino a 2 stelle, ma non inferiori.

Questi due dati presi come esempio possono essere molto utili ad un turista interessato alla scelta di un ristorante dove mangiare in quanto "assicurano" entro un certo limite un servizio di buona qualità.