UNIVERSITÀ DEGLI STUDI
DI MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Corso di Laurea in Informatica

Anno accademico 2021-2022

Raccolta ed analisi di dati relativi a conferenze e luogo in cui sono state svolte

Candidato:
Marco Lupis

Relatore:
Prof. Riccardo Martoglia

Correlatori:
Prof. Luca Bedogni
Prof. Giacomo Cabri
Prof. Francesco Poggi

Indice

Introduzione

In questo elaborato verrà descritta l'attività di tirocinio svoltasi presso l'Università di Modena e Reggio Emilia. L'ambito di cui fa parte è la data analytics, un settore che negli anni recenti sta prendendo sempre più piede in diversi contesti: nel mercato del lavoro ad esempio il numero di persone da impiegare nelle imprese sta esponenzialmente crescendo, così come sono in aumento le figure professionali derivanti dal settore, ad esempio il data analyst, il data scientist, data engineer e così via; figure tra l'altro non facili da trovare dato l'elevato numero di competenze richieste ma soprattutto la multidisciplinarietà presente in queste competenze.

Non soltanto il mercato del lavoro è stato coinvolto dalla diffusione di questa materia: anche la ricerca scientifica, settore inarrestabile e in continua evoluzione, si è adattata al cambiamento. Questa tesi nasce infatti proprio dall'esigenza di alcuni ricercatori di analizzare grandi moli di dati; essi però vanno prima ottenuti, dopodiché avranno sicuramente bisogno di essere processati, infine andranno analizzati per ottenere il risultato sperato, o comunque per avere una risposta ai quesiti che ci si è posti all'inizio della ricerca.

Le domande principali di questa ricerca sono le seguenti: esiste una correlazione tra il successo di una conferenza scientifica e il luogo in cui si è tenuta ? Se c'è, quali sono i fattori che permettono di affermare ciò ?

Verrebbe spontaneo rispondere affermativamente alla prima domanda, in quanto molte delle attività che svolgiamo quotidianamente in un determinato luogo vengono fatte proprio perché la "qualità" del luogo ci spinge a farle, che sia un'attività di svago, lavorativa o altro. Chiaramente per una ricerca non basta esprimere un parere basato sull'esperienza quotidiana, ma bisogna seguire dei metodi scientifici per dimostrare un determinato fatto. Per questo motivo sono stati effettuati tutti i processi di gestione dei dati che la data analysis prevede, per arrivare ad una conclusione e rispondere alle domande che ci si è posti inizialmente.

L'elaborato prevede cinque capitoli: nel primo si scenderà nel dettaglio nell'introdurre dei concetti di base come data analysis, data analytics e data science, dopodiché verrà descritto il punto di partenza della ricerca e quali dati sono stati utilizzati per arrivare alle conclusioni. Il secondo capitolo prevede una descrizione di tutti i software utilizzati ai fini della raccolta e dell'analisi dei dati, il terzo capitolo invece è il fulcro della tesi, perché elenca punto per punto tutto ciò che è stato fatto per arrivare ai risultati, che verranno invece presentati nel capitolo 5.

Nel quarto capitolo invece sono presenti degli screenshot di alcune parti di codice scritto durante il tirocinio che merita osservazioni e commenti particolari.

Parte I – scenario iniziale

1. Contesto dell’attività svolta

Nel seguente capitolo verranno descritti gli ambiti nel quale le attività del tirocinio possono essere collocate: data science, data analytics e data analysis. Successivamente verrà introdotto il progetto al quale le attività svolte fanno capo.

1.1. Data science

La data science è una scienza che coinvolge diverse discipline, il cui obiettivo è l’estrazione di valore da un insieme di dati. Il settore è attualmente in una fase di enorme espansione, e non per un motivo casuale: ciò è causato dall’aumento esponenziale del volume dei dati in circolazione. Si stima infatti che nel 2014 circolassero più di 650 Exabyte di dati [1]; considerando che nel 2007 ne circolavano “soltanto” 65, e che l’avanzare della tecnologia ha permesso un accrescimento dell’archiviazione di informazioni [2], possiamo dedurre che la quantità sia ancora cresciuta esponenzialmente fino ad oggi.

La crescita del volume dei dati sicuramente non è l’unico motivo per il quale la data science sta avendo successo: l’attenzione che il mondo imprenditoriale sta avendo nei confronti di questa scienza permette ad essa di svilupparsi, e permette allo stesso tempo lo sviluppo delle aziende che si interessano alla materia. Esse infatti riescono a prendere decisioni che mirano ad alzare gli utili dell’impresa, ma non solo. Ad esempio:

- riescono ad analizzare dati di processi interni all’azienda,come ad esempio i tempi necessari ad un impianto che modella un determinato prodotto; quest’analisi può portare ad un miglioramento dei processi produttivi
- studiano il comportamento degli utenti sui loro canali di vendita, per capire gusti,abitudini del cliente, ed eventualmente prevedere futuri acquisti.
- in ambito sanitario si possono migliorare le diagnosi dei pazienti analizzando i risultati degli esami clinici
- in ambito finanziario si possono rilevare frodi rilevando comportamenti sospetti degli utenti sulle piattaforme di acquisto e vendita di titoli finanziari.

Ci sono sicuramente tanti altri esempi, che variano anche in base all’ambito in cui l’azienda opera.

In generale, la data science opera seguendo determinati passaggi[3]:

1. *Capire il problema aziendale*: è la fase iniziale, nella quale ci si pone una domanda alla quale dare una risposta o si evidenzia un problema al quale bisogna dare una soluzione; ad esempio un'azienda può chiedersi come ridurre gli sprechi durante la produzione di un bene.
2. *Raccogliere e integrare i dati grezzi*: in questa fase si deve avere una visione di quali dati si hanno a disposizione e soprattutto di quali non si hanno, per far sì che si riescano ad ottenerli, pulirli in modo da averli in un formato leggibile e adatto alle operazioni successive da eseguire.
3. *Esplorare, trasformare, pulire e preparare i dati:* avviene un'elaborazione che porta ad avere dati completi, corretti e non ridondanti. Si impiegano inoltre strumenti per avere visualizzazioni più efficienti dei dati (tipicamente vengono rappresentati dei grafici)
4. *Creare e selezionare modelli basati sui dati:* si utilizzano modelli di machine learning, deep learning o elaborazione del linguaggio naturale, che aiuteranno a risolvere il problema iniziale
5. *Testare, mettere a punto e distribuire i modelli:* ci si assicura che il modello funzioni come dovrebbe, inoltre i suoi output (previsioni, proiezioni, anomalie) vengono inseriti nei sistemi aziendali
6. *Monitorare, testare, aggiornare e governare i modelli:* il modello può sempre essere revisionato e/o migliorato, questo è possibile grazie al monitoraggio e al testing di quest'ultimo, anche con dati nuovi in input.

Come già detto all'inizio del paragrafo, la scienza dei dati coinvolge varie discipline: le più importanti sono senza dubbio l'informatica, la matematica e la statistica; come sottodomini delle materie appena citate abbiamo il data mining, machine learning e intelligenza artificiale, basi di dati e visualizzazione dati; in secondo piano abbiamo anche le scienze umane, la biologia e l'economia.

1.2. Data analytics e data analysis

Se volessimo suddividere la data science in settori, uno di questi sarebbe sicuramente quello della data analytics: possiamo definire quest'ultima infatti come il processo che permette di rispondere a delle domande, di ricavare informazioni e/o trend partendo da dati grezzi, che non hanno valore se vengono presi così come sono e non subiscono un'elaborazione.

Siamo quindi in un contesto più ristretto, dove il focus principale è la vera e propria gestione del dato; in riferimento all'elenco di azioni che fanno parte di un progetto di data science appena citato nel paragrafo 1.1, la data analytics coinvolge i passaggi n. 2 e 3.

Un'altra precisazione da fare riguarda la data analysis: essa infatti viene spesso confusa con la data analytics,ma anch'esse sono due materie distinte, sebbene siano molto correlate. La data analysis infatti è quella ramificazione della data analytics che si occupa di :

- pulire
- trasformare
- modellare
- interrogare

i dati per trovare informazioni utili.

Si notano quindi delle differenze,anche se sottili.

Possiamo schematizzare i tre concetti nel seguente modo:

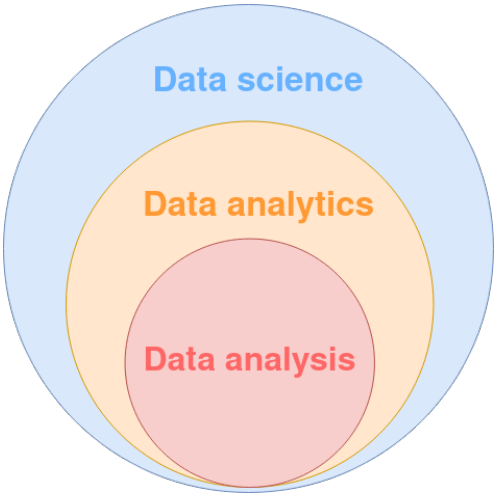

Figura 1.2: schema "a matrioska" della struttura della data science

Possiamo collocare il lavoro descritto in questa tesi nell'ambito della data analytics, ed in particolare nella data analysis.

1.3. Il progetto

Le attività che verranno presentate nei prossimi capitoli sono un contributo ad una ricerca svolta da dei ricercatori dell'Università di Modena e Reggio Emilia. Lo scopo della ricerca è trovare una correlazione tra l'impatto di una conferenza scientifica e il luogo in cui si è tenuta[4]. Per fare ciò, sono stati raccolti dei dati che possiamo suddividere in due categorie:

1. *dati bibliometrici:* un indicatore che riconosce l'importanza di una conferenza è il numero di citazioni degli articoli di quella conferenza. Sono stati quindi raccolti metadati di oltre 2 milioni di articoli, tra cui, oltre alle citazioni, anche il DOI, il luogo e l'anno in cui si è svolta la conferenza, e ovviamente la conferenza a cui appartiene l'articolo.

2. *dati turistici:* definire la "turisticità" di un luogo non è una questione da sottovalutare. Non è facile trovare degli indicatori precisi che descrivano perfettamente quanto un luogo è attraente, perché ci sono diversi fattori, anche soggettivi, che determinano quanto sia turistica una città o uno Stato.

   Per questo progetto, per il quale è già stato pubblicato un articolo, sono stati usati indici diversi per la turisticità di uno Stato e di una città:

   - per uno Stato sono stati estratti degli indicatori da un report del World Economic Forum, il Travel and Tourism Competitiveness Report. Il documento mira a descrivere l'impatto che il settore del turismo e dei viaggi ha sullo sviluppo di un Paese. Sono disponibili i dati di 140 nazioni.

     Il report contiene decine di indicatori numerici, ognuno dei quali delinea un aspetto che contribuisce allo sviluppo economico del Paese, ad esempio la presenza di infrastrutture per il trasporto aereo, oppure la sostenibilità ambientale.

     Nella prima fase della ricerca sono stati utilizzati 4 di questi indici, che riguardano:

     - i servizi a disposizione per i turisti
     - le risorse culturali, tra cui i siti UNESCO
     - infrastrutture stradali e ferroviarie.

       È presente anche un indice che racchiude tutti gli aspetti che il report propone.

   - per una città sono stati utilizzati due parametri molto diversi tra loro:
     - Tourist Arrivals: indica l'arrivo di turisti nella città. Il dato è riferito all'anno 2018
     - Size Wikipedia Page (SWP): esprime la dimensione in byte della pagina di Wikipedia relativa alla città. Ci si aspetta che maggiore è l'importanza della città maggiori saranno le informazioni presenti sulla pagina Web.

Questi sono i dati usati nell'articolo già pubblicato. Per quanto riguarda il lavoro descritto in questa tesi invece sono state individuate nuove fonti da cui attingere, per avere dei risultati aggiornati e più precisi, ma sono anche state riutilizzate quelle vecchie per recuperare indici che non erano stati sfruttati precedentemente. Nel dettaglio:

1. *per i dati bibliometrici:* anche qui abbiamo raccolto informazioni su articoli di conferenze. La novità sta nella fonte dalla quale sono state scaricate: se prima erano stati utilizzati gli archivi di OpenCitations e DBLP, dai quali poi sono stati presi solo gli elementi in comune, stavolta invece abbiamo aggiunto quelli di Microsoft Accademic Graph (MAG), un'altra fonte autorevole contenente milioni di record. Nel dataset usato poi per l'analisi dei dati infatti abbiamo oltre 3 milioni di record, e quindi informazioni su oltre 3 milioni di articoli.

   Un'altra novità è la presenza di informazioni più dettagliate sulla conferenza, nello specifico sui loro ranking, cioè su delle valutazioni fornite da diversi consorzi/associazioni che attestano la qualità di una conferenza. Per l'analisi sono stati utilizzati due ranking provenienti da:

   - consorzio GRIN: consorzio italiano che usa un algoritmo particolare per attribuire un rating ad ogni conferenza. Al di là dei diversi rating esistono comunque tre classi (1,2 e 3) nelle quali vengono raggruppate le conferenze in base alla qualità
   - associazione CORE: un'associazione di dipartimenti universitari di informatica dell'Australia e della Nuova Zelanda [[5]] che attribuisce alle conferenze una classe (A*,A,B,C) in base alla loro qualità.

2. *per i dati turistici:* anche in questa nuova fase è stata mantenuta la distinzione tra indici di una città e di uno Stato.

   - indici di Stato: oltre a conservare quattro dei cinque parametri già utilizzati, ne sono stati aggiunti molti altri. In particolare:
     - sono stati recuperati degli indici del report del World Economic Forum che non erano stati considerati nella prima fase della ricerca, per arrivare quindi ad un totale di 19 parametri
     - sono stati aggiunti gli arrivi di turisti (TA), per gli anni compresi dal 1995 al 2020.
   - indici di una città: non sono stati riutilizzati i precedenti ma ne sono stati aggiunti tre nuovi.
     - risultati Google: corrisponde al numero di risultati che una ricerca Google restituisce se cerchiamo la città in questione

- risultati Booking: corrisponde al numero di strutture ricettive presenti nella città e registrate su booking.com
- risultati TripAdvisor: corrisponde al numero di attrazioni di una città presenti su TripAdvisor. Quando si parla di attrazioni ci si riferisce a diverse tipologie di luoghi: musei, piazze, teatri, parchi e riserve naturali, ponti ma anche pub e ristoranti.

Le modalità con cui è stato utilizzato questo materiale verranno descritte a partire dal capitolo 3.

2. Tecnologie e strumenti utilizzati

In questo capitolo vengono descritti gli strumenti informatici utilizzati durante il tirocinio, a partire da Python, per poi passare alle librerie Python importate nel codice e sfruttate in diversi modi.

2.1. Python

Python è un linguaggio di programmazione dinamico, multi-paradigma e open source. Tra le caratteristiche principali, che hanno sicuramente contribuito alla sua popolarità e diffusione, abbiamo:

- facilità di apprendimento: la sintassi semplice favorisce uno studio che permette di scrivere programmi non banali in poco tempo. Questa caratteristica permette ad un neofita della programmazione di avere subito un linguaggio sul proprio curriculum
- presenza di un'enorme quantità di librerie di vario tipo
- possibilità di utilizzo in vari contesti: sviluppo Web, data science, bioinformatica, sviluppo di videogiochi ecc
- presenza di documentazione ben dettagliata e di grandi community online

Nella data science è sicuramente uno dei linguaggi più importanti e popolari, in quanto fornisce diverse funzioni riguardanti la matematica e la statistica, ma anche librerie di machine learning, deep learning, visualizzazione dati o comunque gestione di dati in generale.

Ci sono altri linguaggi usati nella data science: il più diffuso, dopo Python, è sicuramente R, ma anche Java, SQL, Scala, C++, Matlab [[6]]. Ognuno ha delle caratteristiche per le quali vengono usati in questo ambito, ma difficilmente sono paragonabili alla completezza di Python, che possiede tutte le caratteristiche appena descritte.

2.2. Pandas

Pandas è probabilmente la libreria più popolare di data science per Python, il suo scopo principale infatti è la facile gestione del dato strutturato e l'analisi dei dati.

La sua struttura dati principale è il DataFrame, una sorta di tabella composta da righe e colonne chiamate Series. Possiamo vedere le righe e le colonne come degli array monodimensionali; entrambe hanno delle "etichette" che identificano in modo univoco l'array.

Una funzione molto importante del software è l'importazione di file, di solito in formato csv, Excel o veri e propri database; questo permette di sfruttare dataset provenienti da fonti esterne e quindi evitare di inserire manualmente dati nei DataFrame. È anche possibile esportare un DataFrame in un file: ciò permette di utilizzare i dati su cui si lavora anche fuori da un ambiente di programmazione; l'utilità sta anche nel "congelare" dati su cui si sta lavorando, per poi reimportarli in un secondo momento e terminare il lavoro. In questo modo si evita di rieseguire il codice che aveva prodotto quei dati presenti nel file.

Un altro aspetto caratterizzante della libreria è la gestione dei dati mancanti, sia in fase di creazione di strutture dati, sia in fase di manipolazione:

- in fase di creazione, ad esempio di un DataFrame, nel caso in cui non si dichiarino Series delle stesse dimensioni, verranno inseriti automaticamente dei valori. Esempio:

  ```
  df=pd.DataFrame({ 'col1' :{ 'a' : 1 , 'b' : 1 }, 'col2' :{ 'a' : 1 }})
  ```

  Il codice sopra riportato produrrà il seguente output:

  ```
  >>> df
     col1  col2
  a     1   1.0
  b     1   NaN
  ```

  Il valore inserito automaticamente è NaN, Not a Number. Da notare che la presenza di questo valore ha trasformato il tipo di dato della colonna in un float64, l'equivalente del float di Python; il valore della prima riga infatti è 1.0, nonostante nell'istruzione di creazione del DataFrame sia stato inserito 1.

- in fase di manipolazione abbiamo diverse funzioni che tengono conto della presenza di valori NaN, per citarne alcune:
  - *dropna():* elimina le righe di un DataFrame con valori mancanti

- *isna():* applicato ad una struttura, ne restituisce una delle stesse dimensioni ma con dei booleani al posto dei valori originari. I booleani indicano se nella struttura dati sono presenti dei NaN in quelle determinate posizioni
- *fillna(value):* sostituisce nel DataFrame i NaN con il valore passato come parametro.

La quantità e la qualità delle numerose funzionalità offerte rendono quindi Pandas uno strumento indispensabile per mestieri come il data scientist, il data analyst o semplicemente per chi si approccia alla materia per scopi non lavorativi.

2.3. Numpy

Numpy è una libreria per il calcolo algebrico e matriciale, spesso usata con Pandas in quanto la struttura tabellare dei DataFrame è paragonabile a quella di una matrice. Lo stesso discorso vale per le Series, che, come detto nel paragrafo 2.1, è equiparabile ad un array monodimensionale, e quindi ad una colonna di una matrice.

Viene spesso associato a Matlab per le funzionalità che condividono, ma anche per la sintassi: gli utilizzatori di entrambi i software infatti trovano molte somiglianze tra i due[7].

Data la sua naturale capacità di operare con numeri e matrici, viene spesso usata in ambiti come il quantum computing, l'image processing, l'analisi matematica e le geoscienze[8].

Nel codice scritto durante le ore di tirocinio, Numpy è stata utilizzata principalmente per due motivi:

1. inserimento di valori nulli nei DataFrame, con l'attributo **np.nan**
2. utilizzo del metodo **np.where(),** che ritorna indici di un array o gli elementi dell'array stesso, in base ad una condizione data come parametro.

2.4. Jupyter Notebook

Jupyer Notebook è un'applicazione Web che permette di scrivere documenti al cui interno possono esserci elementi eterogenei tra loro. Si distingue da un editor di testo come Word perché è possibile inserirvi del codice Python, ma soprattutto è possibile eseguirlo e vedere il suo output: questa è una grande comodità quando si lavora con strutture dati come quelle di Pandas o si vogliono

visualizzare dei grafici/immagini prodotte dal codice. È anche possibile scrivere in Markdown, un linguaggio di markup che permette di scrivere del testo ben formattato; questa funzionalità è molto utile quando si vogliono apporre dei commenti al codice appena scritto, ed evitare quindi di usare le funzionalità builtin di Python dei commenti, che spesso risultano antiestetici e poco comodi da leggere.

Il documento è strutturato in celle, facilmente manipolabili sotto diversi punti di vista: possiamo infatti spostarle facilmente in alto o in basso nel documento, tagliarle, copiarle ed incollarle, possiamo scegliere la tipologia del contenuto (linguaggio di programmazione, Markdown, testo semplice), possiamo eseguire il codice di una singola cella o di un loro insieme, evitando quindi di eseguire tutto ciò che c'è nel documento.

Un'altra caratteristica riguardante la strutturazione della pagine in celle è la presenza di due modalità: edit e command mode. La prima è la modalità che si attiva automaticamente quando si digita del testo in una cella; si potranno quindi usare le classiche scorciatoie da tastiera utili durante la scrittura di testo (taglia, copia incolla, seleziona tutto ecc). La command mode invece permette di usare le stesse combinazioni di tasti che si userebbero in edit mode, ma con fini diversi; le scorciatoie sono personalizzabili ma le opzioni di default favoriscono le azioni di manipolazione delle celle, citate poc'anzi.

Ultima ma non meno importante opzione è l'esportazione del documento nei formati più conosciuti per documenti: pdf, LaTex ma anche HTML, Markdown, o semplicemente come script[9].

2.5. Requests

Requests è una libreria che permette di eseguire richieste HTTP in Python. È ormai uno standard de facto, il numero di download settimanali infatti supera i 30 milioni, che lo rendono uno dei package Python più utilizzati[10]. La popolarità è dovuta sicuramente alla sua estrema semplicità di utilizzo: per fare un esempio, basta una sola riga di codice per eseguire una richiesta HTTP:

```
x = requests.get('https://www.unimore.it')
```

il comando su riportato restituisce un Response Object, contenente tutti i dati della risposta restituita dal server interrogato. Questi dati possono essere visualizzati anche uno ad uno tramite gli attributi dell'oggetto Response. Esempio:

```
>>> x.status_code
200
```

Nelle attività di tirocinio questa libreria è stata fondamentale nella fase di scraping, dove è stato necessario fare richieste ai server dei siti Web da cui sono stati scaricati dei dati. È stato

opportunamente utilizzato l'attributo *raise_for_status*, per verificare che non ci siano stati errori durante la fase di richiesta HTTP.

2.6. BeautifulSoup

BeautifulSoup è stato uno degli strumenti più importanti per la raccolta dei dati di questa ricerca. Molte delle informazioni necessarie sono state prese online, da diverse pagine Web. La libreria consente di effettuare il parsing di pagine HTML e XML, sia partendo da un file già presente sul dispositivo in uso, sia prendendo in input un oggetto Response ottenuto da una richiesta effettuata tramite la libreria Requests. Il parsing viene effettuato solitamente dal parser incluso nella libreria di Python, ma se ne possono usare anche altri[11].

BeautifulSoup si occupa di creare un "albero" composto da diversi tipi di oggetti[12], i quali saranno poi facilmente utilizzabili per trovare ed estrarre i contenuti della pagina HTML/XML analizzata. Ad esempio, se volessimo trovare tutti i link presenti in una pagina Web, con il metodo *find_all()* riusciremmo a trovare tutte le ancore HTML presenti nel documento, da cui poi potremo estrarre i link.

A volte il codice HTML potrebbe presentare degli errori di sintassi o semplicemente potrebbe non essere ben pulito: per quanto possa sembrare un problema, BeautifulSoup riesce comunque a creare l'albero e a navigare tra i tag. Potrebbe essere necessario però utilizzare altri strumenti o funzioni built in di Python per ottenere le informazioni desiderate: solitamente si usano i metodi di manipolazione di stringhe o di liste, come *split()*, *replace()*, *strip*(), o a volte anche le espressioni regolari. Come vedremo nel paragrafo 2.7, si integra bene anche con altre applicazioni che navigano su pagine Web, ad esempio Selenium.

2.7. Selenium

Selenium è un insieme di tool e librerie che mirano all'automazione dei browser. Nasce con lo scopo di effettuare test, e mette a disposizione diversi strumenti, tra cui un IDE apposito per il testing, un'estensione per browser ed un'API, SeleniumWebDriver. Quest'ultima è stata utilizzata in questo progetto, non per del testing ma per automatizzare delle operazioni di raccolta dati. È stato necessario scaricare un driver di un browser, cioè lo strumento che invia i comandi ad esso e recupera i risultati elaborati[13]. Tramite il driver quindi è stato possibile effettuare operazioni, non

di testing ma di vera e propria navigazione nel Web: accedere a determinate pagine, digitare nelle caselle di testo e cliccare su dei bottoni.

Durante l'esecuzione del codice viene aperta una finestra del browser, nella quale vengono effettuate le operazioni inserite nel codice; si assiste quindi ad una vera e propria "robotizzazione" del dispositivo in uso, che esegue del lavoro come se ci fosse una persona che compie delle operazioni manualmente.

A volte i server a cui si fa richiesta tramite strumenti come Selenium non restituiscono la risorsa desiderata, ma rispondono con un messaggio di errore. Questo succede perché avviene un controllo anti bot, che può essere però aggirato tramite lo User Agent, una stringa testuale che il client inoltra tramite l'header della richiesta. Il server riceve le informazioni presenti nella stringa, nella quale solitamente sono riportati il sistema operativo e il browser dal quale è partita la richiesta, e restituisce una risposta adatta al dispositivo con quelle caratteristiche.

La scelta di questa libreria è stata dettata dalla necessità di scaricare dei dati che appaiono in una pagina Web solo dopo aver interagito con il sito (click del mouse, digitazione di testo), quindi l'utilizzo esclusivo di BeautifulSoup non è stato sufficiente.

2.8. Matplotlib e Seaborn

Nella data science un mezzo molto usato per la presentazione dei risultati ottenuti dalle analisi è sicuramente la visualizzazione tramite grafici. Essi permettono una comprensione più immediata di un concetto, ancor di più se il soggetto a cui viene presentato il lavoro non è esperto del settore in cui si è svolta l'attività da illustrare. In generale, il cervello umano memorizza meglio ciò che apprende tramite immagini; leggere del testo o tabelle con i numeri potrebbe non far emergere dettagli che, con la tipologia giusta di grafico, risultano evidenti.

Si deduce facilmente quindi che per l'attività svolta in questo tirocinio sono stati utilizzati dei grafici. Nell'immensità di librerie disponibili per Python ne abbiamo due che spiccano fra le altre per la qualità delle loro prestazioni: parliamo di Matplotlib e Seaborn.

La prima è la più datata, consente di creare grafici in 2D, anche animati e interattivi; la sua filosofia è il creare grafici semplici con pochi comandi. La seconda, anche se arrivata molto più tardi, ha guadagnato molta popolarità, in quanto più moderna nelle rappresentazioni[14]; anche la sintassi più snella favorisce un uso più confortevole.

Entrambe hanno a disposizione diversi tipi di grafici, tra i più interessanti, come possiamo notare nella figura 2.8, abbiamo i classici grafici a linee e/o a punti (lineplot, scatterplot), diagrammi a torta, istogrammi, ma anche la heatmap, utile nella data science perché adatta a mostrare correlazioni.

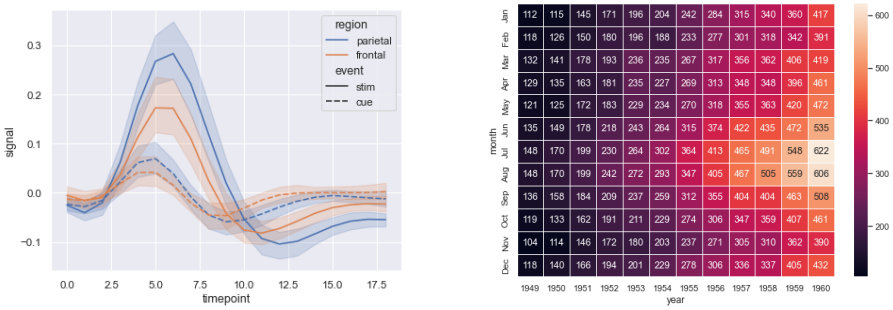

Figura 2.8: esempi di lineplot ed heatmap

2.9. Altri strumenti

Qui di seguito sono elencati alcuni strumenti usati meno rispetto a quelli appena descritti, ma che hanno avuto comunque un'utilità nello sviluppo del progetto:

- *Github*: forse lo strumento di version control più popolare, nel quale è stata creata una repository dove archiviare tutto il lavoro svolto, per consentire anche a chi ha contemporaneamente lavorato al progetto di scaricare i file necessari
- *Visualizzatore di fogli di calcolo*: alcuni dati sono stati presi da dei fogli di calcolo. Un visualizzatore come LibreOffice Calc è stato necessario per un'esaminazione preliminare di ciò che poi è stato portato in strutture come i DataFrame
- *Espressioni regolari*: Python possiede un package chiamato **re**, che permette di utilizzare espressioni regolari. Come già accennato nel paragrafo 2.6, si rendono molto utili per la manipolazione di porzioni di testo, nello specifico per la sostituzione e ricerca di caratteri
- *PyPDF*: libreria Python per la manipolazione di documenti, è stata utile per l'estrazione di indici turistici da un file PDF.

Parte II – sviluppo dell’attività e risultati

3. Progettazione

Nel primo paragrafo di questo capitolo verranno illustrati gli obiettivi di questo progetto, nei paragrafi successivi invece verranno descritti tutti i passaggi avvenuti per arrivare ai risultati finali del lavoro, evidenziando i metodi con cui sono state compiute determinate operazioni, le problematiche emerse durante il lavoro e le soluzioni adottate per risolverle.

3.1. Obiettivi

Nel paragrafo 1.3 è stato affermato che lo scopo della ricerca è di trovare una correlazione tra il successo di una conferenza scientifica e il luogo in cui si è svolta. Avendo già a disposizione i dati riguardanti le conferenze, uno dei primi obiettivi della nuova fase è stato cercare nuovi dati turistici che esprimessero quanto un luogo è popolare dal punto di vista del turismo.

Una volta trovati i dati bisogna capire se sono ottenibili facilmente o no. Essi infatti potrebbero semplicemente trovarsi in un file liberamente scaricabile, oppure è necessario avere abbonamenti a qualche servizio che li fornisce, o magari sono dati non raccolti in un file ma presenti nel corpo di pagine Web, e quindi hanno bisogno di essere scaricati tramite operazioni di scraping. Nei prossimi paragrafi verranno dettagliate tutte le modalità con le quali sono stati ottenuti gli indici turistici necessari per l'analisi.

L'obiettivo successivo è stata la preparazione dei dati per l'analisi di questi ultimi. Come detto nel paragrafo 1.2, una delle fasi che un processo di data science prevede è quella della pulizia, integrazione e preparazione dei dati, per far sì che i risultati finali siano i più corretti possibile. In questa fase si coglie anche l'occasione per controllare che il download dei dati sia avvenuto correttamente.

Nell'ultima parte del lavoro si persegue l'obiettivo principale del progetto, cioè trovare una correlazione tra i dati bibliometrici e quelli turistici; con il termine "correlazione" non ci si riferisce soltanto al significato statistico della parola ma anche alla definizione più generica, intesa quindi come relazione reciproca tra due elementi.

3.2. Ricerca di nuovi indici turistici

La ricerca di nuovi parametri turistici si è basata sul voler avvicinarsi il più possibile a racchiudere in un valore numerico la potenzialità di un luogo di attrarre turisti. Questa potenzialità è sicuramente dettata da tantissimi fattori, ad esempio la presenza di luoghi d'interesse, i collegamenti ferroviari/stradali interni all'area di interesse, i collegamenti con altre zone turistiche, la presenza di servizi per turisti come infopoint o abbonamenti speciali per mezzi pubblici, e così via. La pluralità di fattori non permette sicuramente di racchiudere tutto in un solo parametro, per questo i primi tentativi di ricerca si sono indirizzati verso una suddivisione in ricerche per categorie, come cultura, infrastrutture, servizi, vita notturna.

Si è partiti da una ricerca Google, un tentativo banale e scontato in quanto il passare del tempo online ci ha abituati a chiedere qualsiasi cosa al motore di ricerca; ma data la sua ormai nota potenza, la possibilità di trovare dei risultati più o meno soddisfacenti non era bassa.

Le ricerche sulla piattaforma in sé non hanno portato direttamente ai risultati previsti, ma hanno fatto comparire delle pagine Web che hanno dato spunti per indirizzarsi verso un altro metodo di ricerca. Dato che uno degli obiettivi era trovare indici turistici riguardanti uno Stato, ed avendo nei dati bibliometrici una lista di 126 nazioni, non sarebbe stato di certo producente cercare 126 fonti diverse (o comunque qualche decina) che restituissero risultati soddisfacenti. Lo scopo da questo punto in poi è stato quindi trovare un'unica fonte da cui estrarre del materiale, ed essendo il numero di nazioni così elevato, è stata fatta una ricerca di organizzazioni/associazioni riconosciute a livello mondiale per la loro autorevolezza e affidabilità, che potessero fornire informazioni utili. Questa via ha portato a fare un elenco di questi enti, con lo scopo di racimolare dei dati tramite i loro canali ufficiali.

Sono stati analizzati siti di autorità come ONU, Organizzazione mondiale del commercio, Fondo monetario internazionale, Unione Europea, Organizzazione per la cooperazione e lo sviluppo economico, Organizzazione mondiale del turismo. Le fonti che sono tornate utili sono le ultime due citate nell'enunciato precedente. Sono state trovate diverse tipologie di dati, in particolare sul portale dell'Organizzazione mondiale del turismo, dove ad esempio si avevano a disposizione il numero di turisti arrivati in uno Stato categorizzati per il mezzo di trasporto con il quale sono arrivati, oppure il numero di turisti che hanno soggiornato almeno per una notte, suddivisi in base al tipo di struttura in cui hanno alloggiato (Hotel, B&B, ecc..). I dettagli sul download di questi dati verranno forniti nel paragrafo 3.3.1.

Per quanto riguarda gli indici delle città invece, la ricerca è stata un'impresa ben più ardua. L'immensità del Web non è stata sufficiente per avere dei dataset pronti per essere utilizzati o comunque con una quantità contenute di dati da estrarre. Per questo motivo si è deciso di "inventare" dei parametri che comunque abbiano un significato; inventare non vuol dire assegnare dei numeri casualmente alla città, bensì trovare dei valori che siano collegati in qualche modo con il

luogo. Questi valori sono stati estrapolati da pagine online tramite tecniche di Web scraping, e sono stati già nominati e descritti nel paragrafo 1.3; a partire dal paragrafo 3.3.2 verrà mostrato come sono stati scaricati questi indici.

3.3. Estrazione dei nuovi indici turistici

La fase di estrazione è stata la più impegnativa del progetto, per diversi motivi: per la quantità di tempo impiegato, per la quantità di strumenti utilizzati, per la lunghezza delle operazioni da eseguire, prima fra tutte la connessione alle migliaia di pagine Web a cui accedere e prelevare contenuti.

Nei prossimi paragrafi verranno mostrate le procedure con le quali si è arrivati ad avere dei dati pronti per la fase finale di analisi, esaminando caso per caso tutti gli indici ottenuti.

3.3.1. Arrivi di turisti in uno Stato

Come già detto nel paragrafo 3.2, una delle fonti più utili è stata l'Organizzazione mondiale del turismo, dalla quale si possono ottenere informazioni molto dettagliate; per fare un altro esempio tra le variabili possibili c'è la permanenza media negli hotel, espressa in numero di notti.

Nonostante l'elevata categorizzazione di informazioni potenzialmente utili, si è optato per scaricare dati più generici, per rimanere fedeli al principio di esprimere la turisticità di un luogo nella maniera più completa possibile. Scegliere di usare dati come ad esempio il numero di lavoratori impiegati nel settore turistico avrebbe potuto trascurare aspetti più importanti del turismo. Infine per questa fonte sono stati scaricati gli arrivi di turisti in uno Stato, suddivisi per anno.

Un altro ente che ha a disposizione del materiale utile è l'Organizzazione per la cooperazione e lo sviluppo economico (OCSE). Anch'essa rende possibile scaricare informazioni sugli arrivi di turisti, ma anche sulle partenze, suddivise per anno, tipologia di sistemazione, tempo di permanenza nel Paese. Questa fonte è stata però scartata, in quanto la quantità di dati utili non è comparabile a quella dell' Organizzazione mondiale del turismo, che raccoglie dati di oltre 200 Paesi, mentre la prima soltanto 60; inoltre quest'ultima possiede informazioni per una fascia temporale maggiore (dal 1995 al 2020, l'OCSE invece dal 2008 al 2018).

I dati scaricati erano in un file Excel, importato poi nel notebook e manipolato per estrarre soltanto la parte necessaria. Nella figura 3.3.1 viene mostrato un esempio di alcuni dati relativi ad uno Stato:

si può facilmente notare che la maggior parte del contenuto delle celle non è utile ai fini dell'analisi; per questo molte delle colonne e delle righe sono state eliminate, conservando solo il nome dello Stato, i valori degli arrivi e la riga delle "etichette" con gli anni di riferimento.

			1. INBOUND TOURISM: Arrivals						
C.	S.	C. & S.	Basic data and indicators	Units	Notes	Series	1995	1996	1997
204	0	204-0	BENIN						
			Arrivals						
204	1.1	204-1.1	Total arrivals	Thousands		VF	580	516	541
204	1.2	204-1.2	Overnights visitors (tourists)	Thousands		TF	138	143	148
204	1.3	204-1.3	Same-day visitors (excursionists)	Thousands			..	..	..
204	1.4	204-1.4	of which, cruise passengers	Thousands			..	..	..

Figura 3.3.1: righe di esempio del file Excel contenente gli arrivi di turisti in uno Stato, suddivisi per anno.

Fonte: Organizzazione mondiale del turismo

L'ultima operazione sul DataFrame è stata la sostituzione di alcuni valori nelle colonne degli arrivi. Quando un dato non è disponibile, nel foglio elettronico sono stati inseriti due punti, che vengono letti da Pandas come una stringa: quest'ultima è stata rimpiazzata con un NaN in tutte le celle dove era presente. Quest'operazione è quasi necessaria, in quanto stiamo gestendo valori numerici, e combinarli con le stringhe porta sicuramente a generare negli errori nella fase di analisi dei dati.

Il risultato finale è quello che possiamo vedere nelle righe di esempio del DataFrame 3.3.1.

Paesi	1995	1996	1997	1998	1999	2000
AFGHANISTAN	NaN	NaN	NaN	NaN	NaN	NaN
ALBANIA	304.0	287.0	119.0	184.0	371.0	317.0
ALGERIA	520.0	605.0	635.0	678.0	749.0	866.0
AMERICAN SAMOA	NaN	NaN	NaN	NaN	NaN	NaN

DataFrame 3.3.1: arrivi di turisti in uno Stato, suddivisi per anno.

3.3.2. Indici del World Economic Forum

Tra i primi indici utilizzati nella ricerca, ancor prima che il lavoro descritto in questo elaborato venisse svolto, figurano quelli prelevati da un report del World Economic Forum riguardante il turismo e i viaggi. I risultati ottenuti da questi parametri hanno portato a riutilizzare i parametri stessi, adattandoli ai nuovi dati bibliometrici, ma anche a scaricarne di nuovi, che erano presenti nel report ma che non erano stati selezionati per la prima analisi dei dati. Si è provveduto quindi a recuperare il documento, disponibile in formato PDF, il quale è stato scansionato opportunamente con la libreria nominata nel paragrafo 2.9, PyPDF.

Il report suddivide gli indicatori in:

- *TTCI*: è un unico indice, corrisponde alla somma algebrica di 4 valori. Concettualmente è quello che racchiude tutti gli aspetti che il report tiene in considerazione per definire la competitività di uno Stato in ambito turistico
- *4 indici*
- *14 "pillars"*, sotto indici dei 4 indici sopra citati
- *90 indicatori generici[15].*

Possiamo immaginare le tipologie sopra elencate come strati di una piramide: per costruire ognuno di essi infatti ci si è basati su quello successivo, eseguendo calcoli algebrici tra i diversi valori.

I valori dei 90 indicatori generici non sono annotati nel report, per questo sono stati scaricati soltanto gli altri 19, un numero comunque molto importante che permette di fare riflessioni durante la fase di analisi.

Oltre al TTCI già commentato poco fa, gli altri indici e sotto indici sono:

Nome indice/sottoindice	Cosa esprime
EE - Enabling Environment	Condizioni generali necessarie per operare in una nazione
BE - Business Environment	Adozione di politiche favorevoli allo sviluppo di imprese
SS - Safety and Security	Qualità dei servizi di polizia
HH - Health and Hygiene	Accesso ad acqua potabile, presenza di letti di ospedale, presenza di malattie
HRLM - Human Resources and Labour Market	Capacità del Paese di sviluppare competenze attraverso l'istruzione e capacità di distribuire queste competenze nei settori lavorativi giusti
TTPEC - T&T Policy and Enabling Conditions	Politiche e aspetti strategici che impattano l'industria del turismo
PTT - Prioritization of Travel & Tourism	Priorità che i governi danno al settore turistico
IO - International Openness	Disponibilità a stringere rapporti commerciali con l'estero
PC - Price Competitiveness	Costo della vita, concentrandosi su spese sostenute da turisti come le tasse sui biglietti aerei
ES - Environmental Sustainability	Adozione di politiche ambientali
INF - Infrastructure	Disponibilità e qualità delle infrastrutture fisiche
ATI - Air Transport Infrastructure	Presenza di aeroporti e voli
GPI - Ground and Port Infrastructure	Presenza di infrastrutture stradali, ferroviarie e portuali
TSI - Tourist Service Infrastructure	Disponiiblità di strutture ricettive
NCR - Natural and Cultural Resources	Risorse naturali e culturali. Secondo il report sono i motivi principali per viaggiare
NR - Natural Resources	Disponibilità di risorse naturali
CRBT - Cultural Resources and Business Travel	Presenza di siti UNESCO, di grandi stadi

Tabella 3.3.2: indici raccolti dal report del World Economic Forum

Le righe bianche della tabella contengono i 4 indici, le righe grigie successive invece i relativi sotto indici. Tutti i valori vanno da un minimo di 1 ad un massimo di 7.

L'estrazione dal file PDF degli indici non è stata particolarmente complessa, ma ha richiesto dell'attenzione nella fase iniziale. Una volta estratto il testo, è risultato utile portarlo in un file .txt. Possiamo vedere nella figura 3.3.2 come il testo del documento si è distribuito uniformemente sulle righe del nuovo file. L'unico ostacolo è stato il dover pulire i file (uno per ogni pagina di pdf) prima dell'estrazione dei dati, perché contenevano delle righe vuote e dei caratteri non utili. È bastato poi il metodo *readline()* per estrarre una riga, che è stata inserita come valore in un DataFrame.

Rank	Economy	Score
1	Spain	5.4
2	France	5.4
3	Germany	5.4
4	Japan	5.4
5	United States	5.3
6	United Kingdom	5.2

```
1
Spain
5.4

2
France
5.4

3
Germany
5.4

4
Japan
5.4
```

Figura 3.3.2: a sinistra, un estratto di una pagina del PDF, a destra il testo estratto dalla pagina, riportato su un file di testo.

La struttura del DataFrame con tutti gli indici è la seguente:

Stato	TTCI	EE	TTPEC	INF	NCR	BE
Albania	3.6	5.0	4.3	3.1	2.0	4.0
Algeria	3.1	4.6	3.6	2.3	2.1	3.9
Angola	2.7	3.4	3.7	2.1	1.7	3.5
Argentina	4.2	4.9	4.0	3.4	4.3	3.3
Armenia	3.7	5.3	4.4	3.2	2.0	5.0

DataFrame 3.3.2: indici turistici di uno Stato. Fonte: World Economic Forum

3.3.3. Risultati Google

Anche il download di questo primo indice relativo alle città non è stato complesso. Siamo sicuramente ad un livello di difficoltà superiore rispetto all'estrazione di testo da un PDF, perché sono stati coinvolti strumenti di manipolazione di dati di pagine Web come BeautifulSoup e Requests.

I passi per arrivare al risultato finale sono stati pochi: dopo aver richiesto al server la pagina desiderata, dalla risposta si è estratto il codice HTML, e nello specifico il testo del tag HTML dove era contenuto il numero di risultati della ricerca. A quel punto è bastata un'espressione regolare che "catturasse" dei caratteri in mezzo a due parole, corrispondenti al numero (in formato stringa) a cui si voleva arrivare.

Dopo l'esecuzione dello script è doveroso controllare che abbia operato correttamente. Scorrendo tra le righe del DataFrame sembra che non ci siano errori, quindi si passa al controllo successivo, cioè verificare se per qualche città non si è riusciti ad ottenere il risultato sperato. Per questo si sfrutta il metodo *isna()* per vedere se ci sono dei NaN nella colonna dei valori. Da qui arriva la conferma che lo script ha fatto il suo dovere, tranne che per 3 città su 2403; si è proceduto quindi a riempire quei pochissimi valori nulli a mano, eseguendo una ricerca "manuale" tramite browser.

Il DataFrame finale è cosi strutturato:

citta	risultati
Austin	848.000.000
Wrocław	105.000.000
Innsbruck	117.000.000
Villefranche-sur-Saône	11.400.000
Zakopane	25.900.000
...	...
Veneto	213.000.000
Bastia	30.500.000
Laramie	33.700.000
Longyearbyen	15.300.000
Shijiazhuang City	9.810.000

DataFrame 3.3.3: numero di risultati di una ricerca Google per una determinata città.

3.3.4. Risultati Booking

In questo caso è stata eseguita una procedura quasi identica a quella adottata per i risultati Google. La differenza principale è nel fatto che una ricerca su Booking di default prevede anche l'inserimento di una data di check in e di check out dalla struttura ospitante, se questo dato non viene inserito allora il sistema imposterà dei valori di default. Inizialmente lo script non prevedeva la ricerca con delle date prescelte, quindi venivano effettuate ricerche con valori inseriti in automatico. Una ricerca con delle date in input è sbagliata a prescindere per i nostri scopi, perché restituisce il numero di strutture disponibili nel periodo selezionato, e non il numero di quelle presenti nella città. Per questo si è cercata una strada alternativa per ottenere i dati di cui si aveva bisogno: il sito è stato navigato in lungo e in largo, con il fine di trovare una sezione alternativa da cui prendere le informazioni. Purtroppo questa sezione non esiste, quindi si è tornati indietro per trovare una soluzione al problema sopra descritto, e fortunatamente questa soluzione è stata trovata. Il trucco era nel visitare la pagina di ricerca giusta, che permettesse di cercare un alloggio senza inserire date di alcun tipo; è bastato modificare il link della pagina alla quale fare richiesta per ottenere finalmente il dato desiderato.

Successivamente si è passati alla fase di "controllo dati": in questo caso le ricerche che non hanno restituito il numero di strutture sono state diverse decine. Leggendo l'elenco di città per le quali non si avevano dei risultati è emerso che molte di queste sono russe, e nel momento in cui lo script è stato eseguito il sito non permetteva prenotazioni per quelle destinazioni. Molte altre città invece avevano nel nome la parola "City": provando a fare una ricerca online di queste città si è scoperto che in realtà il vero nome non contiene quel termine, per questo si è provveduto ad eliminarlo e a rieseguire la ricerca con il nome modificato. In questo modo sono stati ottenuti 25 nuovi risultati, facendo diminuire il numero di valori nulli da 234 a 209.

L'ultimo controllo sulla correttezza dei risultati è stato fatto eseguendo delle ricerche di prova dalla home page del sito, inserendo manualmente alcune città. È stato riscontrato che lo script ha lavorato in maniera ottimale.

Il DataFrame finale è così strutturato:

citta	risultati
Austin	639
Wrocław	970
Innsbruck	288
Villefranche-sur-Saône	16
Zakopane	2,227

DataFrame 3.3.4: numero di risultati di una ricerca su Booking per una determinata città.

3.3.5. Risultati TripAdvisor

L'ultimo indice relativo alle città corrisponde al numero di attrazioni TripAdvisor, già descritte nel paragrafo 1.3. In questa fase di scraping la complessità è stata maggiore rispetto alle altre due, per diversi motivi: il primo è sicuramente l'utilizzo di uno strumento aggiuntivo oltre a BeautifulSoup, vale a dire Selenium; secondariamente perché la struttura del codice HTML del sito è tutt'altro che semplice, e questo ha richiesto un'attenta esplorazione del codice stesso. Inoltre l'utilizzo di un tool di automazione prevede l'interazione con la pagina web, che cambia "forma" in base al tipo di azione eseguita (clic del mouse, digitazione di testo ecc…) , e questo porta anche ad un cambiamento del codice HTML della pagina stessa; bisogna quindi prestare attenzione a come si naviga nell'albero generato da BeautifulSoup e da quali tag HTML vengono prelevati i dati, per evitare di prendere quelli sbagliati.

La molteplicità di possibili cambiamenti della pagina ha portato a dover prevedere la nascita di possibili errori; sono stati inseriti quindi dei blocchi try/except nel programma, che gestiscono tutti i casi nei quali si può ricadere.

Nonostante la diversità del lavoro da svolgere in questa fase di scraping rispetto alle altre due, si è riuscito comunque a riutilizzare una buona parte del codice già scritto per le fasi precedenti, in quanto abbiamo delle operazioni molto simili che fanno parte di un lavoro del genere, cioè l'instaurazione di una connessione con un server e l'eventuale gestione di errori in caso di riposta negativa, l'analisi della pagina ottenuta, l'estrazione del testo desiderato. Verranno mostrati gli snippet di codice più importanti nel capitolo 4.

Il DataFrame di questa fase ha una struttura praticamente identica ai precedenti:

citta	risultati
Austin	1,165
Wrocław	439
Innsbruck	306
Villefranche-sur-Saône	28
Zakopane	203

DataFrame 3.3.4: numero di risultati di una ricerca su TripAdvisor, nella sezione "attrazioni", per una determinata città

3.3.6. Riepilogo indici turistici

Di seguito sono riportati tutti gli indici di cui si è parlato in questo capitolo, per avere una visione completa dei dati turistici a disposizione:

Indici di Stato	Indici di città
Numero di arrivi di turisti in uno Stato (TA) *26 valori, dall'anno 1995 al 2020*	***Risultati Google*** *numero di risultati di una ricerca effettuata sul motore di ricerca*
Indici del World Economic Forum: *19 parametri, ognuno esprime un concetto relativo al turismo del Paese di riferimento*	***Risultati Booking*** *numero di strutture presenti sul sito Booking.com*
	Risultati TripAdvisor *numero di attrazioni turistiche registrate su TripAdvisor*

Tabella 3.3.6: schema riassuntivo degli indici turisti raccolti per l'analisi.

3.4. Preparazione dei dati per l'analisi

Una volta ottenuti tutti i dati necessari, verrebbe spontaneo pensare di passare subito all'analisi dei dati. Questo è un passo azzardato, in quanto durante la fase di raccolta, anche se si opera con diligenza, qualche errore potrebbe essere stato commesso. Per questo motivo si è deciso di dare un'ultima occhiata al materiale appena prodotto, anche con lo scopo di preparare dei dataset con una struttura che avrebbe poi facilitato l'analisi.

3.4.1. Indici relativi ad uno Stato

Il punto di partenza è stato il controllo degli indici relativi agli Stati: aprendo i due DataFrame si nota subito che in uno dei due i nomi delle nazioni sono scritti in maiuscolo, nell'altro invece soltanto le iniziali sono maiuscole. Dato che nei record del dataset degli articoli (dati bibliometrici) i nomi dei luoghi hanno soltanto le iniziali maiuscole, si è deciso di uniformare tutte le stringhe a

quest'ultimo metodo di scrittura, per favorire un futuro join tra questi attributi (e anche per una miglior leggibilità delle stringhe).

Successivamente è stata fatta un'altra osservazione: nella fase di raccolta gli indici relativi agli Stati sono stati presi da dei file, senza essere filtrati in alcun modo; questo vuol dire che sono stati salvati quelli di tutte le nazioni presenti nei file, senza guardare quali Paesi ci sono tra i dati bibliometrici in possesso. È stato quindi eseguito un filtraggio per mantenere solo quelli necessari; gli altri sono comunque conservati nei file originari, in quanto potranno rendersi utili in eventuali ricerche future.

L'ultima considerazione fatta è forse la più importante: controllando quali Stati ci fossero in comune tra i vari dataset, si è notato che mancavano nazioni relativamente grandi o importanti a livello mondiale, un fatto che ha destato dei dubbi in quanto si hanno a disposizione dati di oltre 3 milioni di articoli, e la probabilità che Paesi come la Russia non comparissero nei dataset degli indici era abbastanza bassa. Andando a leggere l'elenco dei Paesi non in comune, si è notato che alcuni Stati hanno dei nomi leggermente differenti nelle varie liste. Esempi:

- Czechia = Czech Republic
- Iran = Iran, Islamic Rep.
- Russia = Russian Federation
- Slovakia = Slovak Republic.

Per questo motivo si è provveduto subito ad uniformare i nomi, in modo da recuperare dati preziosi che sarebbero andati persi nelle fasi di join.

Infine gli indici sono stati inseriti in un unico dataset, al quale oltre alla Series contenente gli Stati è stata aggiunta una colonna con il luogo completo (città, regione, Stato), per favorire un eventuale join futuro con i dati bibliometrici.

3.4.2. Indici relativi ad una città

Per quanto riguarda gli indici delle città il processo è stato molto più semplice, principalmente perché a differenza degli indici di Stato non sono stati presi da dei file ma è stato necessario effettuare delle ricerche su dei siti online, ai quali ho dovuto inviare io stesso la stringa da cercare. Non ci sono quindi stati i problemi descritti poco fa: è bastato creare un nuovo DataFrame dove inserire una colonna con i nomi delle città e le colonne con tutti i parametri.

3.4.3. Altri dati

Fino ad ora il focus è stato sugli indici turistici, tralasciando la parte bibliometrica del progetto. L'analisi si concentrerà infatti perlopiù sull'incrocio tra i due tipi di dati, evidenziando come il turismo impatta sulle conferenze, e non il contrario. Per dare continuità a quello che è stato già fatto nella prima pubblicazione di questa ricerca si è deciso di replicare un'analisi prettamente riguardante i dati bibliometrici, di cui si parlerà nel paragrafo 3.5. Per fare ciò è stato necessario preparare due dataset con uno scheletro praticamente identico: sulle righe, come indice, avremo i nomi delle conferenze (quello che nei dataset è stato nominato *ConferenceSeriesNormalizedName*), sulle colonne invece avremo degli anni. Come possiamo notare nel DataFrame 3.4.3, il contenuto delle celle equivale al numero di citazioni di una conferenza in un determinato anno; nel primo dataset ci saranno le citazioni totali che gli articoli di quella conferenza hanno ottenuto, nella seconda invece le citazioni medie (citazioni totali/numero di articoli).

	2006	2007	2008	2009
accv	4.670103	8.077348	NaN	3.726744
ace	6.916667	7.463415	5.327434	3.366071
aces	NaN	NaN	NaN	NaN
acfie	NaN	NaN	1.187500	NaN

DataFrame 3.4.3: esempio di dataset contenente il numero di citazioni (in questo caso medie), suddivise per conferenza e per anno.

3.5. Analisi dei dati

L'obiettivo finale di questo progetto è raggiungibile solo passando per la fase di analisi dei dati, forse la più importante perché è quella che restituisce i risultati. L'aver processato bene i dati di cui si è discusso fino ad ora è sicuramente propedeutico ad una buona analisi e di conseguenza alla produzione di risultati corretti.

In questo paragrafo verranno discusse e motivate le tecniche con le quali sono state eseguite le diverse analisi, i risultati verranno poi commentati nel capitolo 5.

Avendo a disposizione due fonti di dati bibliometrici, si è deciso di eseguire la stessa analisi su entrambe le fonti.

La prima operazione ha riguardato soltanto i dati bibliometrici: sono state messe a confronto diverse citazioni di conferenze, sia medie che totali. Nello specifico sono state prese quelle per le quali avevamo il ranking CORE e GRIN, descritti nel paragrafo 1.3. Sono stati prodotti diversi grafici a linee, in ognuno dei quali erano presenti conferenze appartenenti alla stessa classe, e sono stati fatti dei commenti riguardanti l'andamento delle citazioni negli anni, per evidenziare come il successo di una conferenza può cambiare nel tempo.

Successivamente si è passati alla prima integrazione con i dati turistici, nello specifico gli arrivi di turisti in uno Stato, suddivisi per anno. Anche qui si è cercato di rimanere in linea con i risultati pubblicati in precedenza, per questo è stato fatto un tentativo di riprodurre un grafico a linee che mostrava l'andamento delle citazioni medie di una conferenza negli anni , confrontato all'indice SWP dei luoghi dove si era svolta la conferenza in quel determinato anno. Come possiamo vedere nella figura 3.5 le due linee seguono un andamento quasi identico, si potrebbe quindi intuire che una correlazione tra i due esista. Con i nuovi dati, gli arrivi di turisti hanno sostituito il SWP, mentre l'altra linea continua a rappresentare le citazioni medie di una conferenza.

Le conferenze di cui possediamo dati sono oltre 5000, e per provare a riprodurre ciò che è stato appena descritto sono stati fatti decine e decine di tentativi. Purtroppo non è stato trovato alcun esempio fedele a ciò che era stato fatto in precedenza; questo non esclude che possa esserci comunque qualche correlazione tra andamento di citazioni e il numero di turisti di uno Stato.

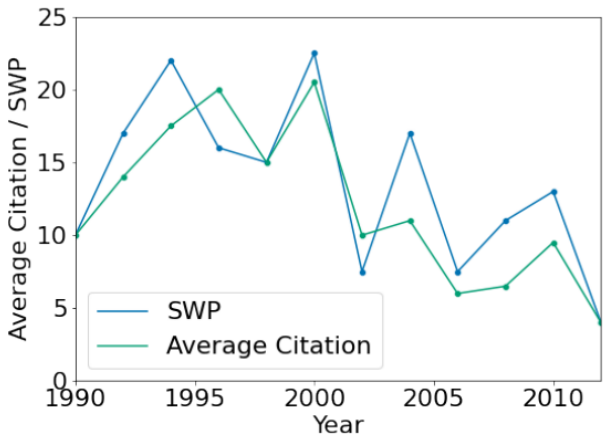

Figura 3.5: andamento di citazioni medie di una conferenza e indice turistico SWP

Andando avanti con l'analisi, ci si avvicina sempre di più a concetti di statistica. I risultati che verranno descritti nelle prossime righe infatti riguardano due tecniche che studiano la relazione tra

due variabili: la regressione lineare e la correlazione. Nel primo caso è stato fondamentale l'uso di Seaborn, che mette a disposizione una funzione che, dati in input le due variabili da mettere in relazione, costruisce automaticamente il grafico di regressione. Per questo tipo di studio si è deciso di confrontare i valori degli indici del World Economic Forum (asse x del grafico) con le citazioni medie di tutti gli articoli di conferenze tenutesi in un determinato Stato (asse y). È stato generato un grafico per ogni indice: abbiamo quindi 19 grafici, che hanno presentato i primi risultati positivi dell'analisi.

L'ultimo utilizzo degli indici di Stato è avvenuto calcolando la sopra citata funzione di correlazione con gli stessi dati di input usati per i grafici di regressione. Sono state utilizzate funzioni di Pandas che permettono di scegliere anche il metodo di correlazione. Ne sono presenti tre: Pearson, Kendall e Spearman; è stato deciso di non escludere nessuno dei tre, in modo da fare un confronto con i risultati della prima fase di ricerca dove erano stati utilizzati tutti. Per presentare i risultati ottenuti sono state utilizzate delle heatmap, una per ogni metodo di correlazione. Anche qui i risultati ottenuti sono stati soddisfacenti, come vedremo nel capitolo 5.

L'ultimo grafico generato è identico al precedente, la differenza sta nell'impiego degli indici delle città in sostituzione di quelli di Stato. Anche le citazioni medie non sono le stesse, bensì sono riferite agli articoli di conferenze che si sono svolte in determinate città.

I risultati sono stati meno soddisfacenti se confrontati allo stesso schema con i dati relativi agli Stati: il coefficiente di correlazione infatti, il cui range va da -1 a 1, è nella maggior parte dei casi vicino allo 0. Si conferma ancora una volta quindi la difficoltà a trovare degli indicatori che permettano di dimostrare un collegamento tra il turismo di una città e il successo di una conferenza.

4. Implementazione

In questo capitolo verranno mostrate le parti di codice più interessanti e/o più rilevanti del progetto, seguite da commenti e spiegazioni utili per una più facile comprensione del codice stesso. Il capitolo è suddiviso in paragrafi, ognuno dei quali riporta il codice di una delle fasi descritte nel capitolo 3.

4.1. Arrivi di turisti in uno Stato

```
paesi = pd.read_excel('unwto-inbound-arrivals-data.xlsx', usecols=[3])
paesi.drop([0,1], inplace=True)
# mantengo solo le righe con i nomi degli Stati
paesi = paesi.iloc[::6]
paesi.drop([1340,1346], inplace=True)



arrivi = pd.read_excel('unwto-inbound-arrivals-data.xlsx', header=2,
                       usecols=range(11,37))
arrivi = arrivi.iloc[2::6]
arrivi.drop([1340], inplace=True)
arrivi.replace('..',np.NaN, inplace=True)

# aggiungo gli Stati al dataframe dei valori
arrivi['Paesi'] = paesi
```

Script 4.1: estrazione da un file excel di arrivi di turisti di uno Stato, suddivisi per anno

Per produrre un dataset ordinato con tutti gli arrivi di turisti in uno Stato, nel codice è stato importato lo stesso file per due volte. Per quanto possa sembrare poco ottimale, è risultato più comodo ripetere l'operazione, in quanto la struttura articolata del file avrebbe portato ad eseguire diverse operazioni di pulizia che avrebbero soltanto allungato i tempi. Si è preferito quindi sfruttare il parametro *usecols* che la funzione *read_excel()* mette a disposizione per importare solo le colonne necessarie; successivamente con il metodo *iloc()* si è provveduto a conservare soltanto le righe utili, mentre altre sono state eliminate grazie alla funzione *drop()*, passandole gli indici delle righe da buttare via. Infine sono stati inseriti dei NaN con la funzione *replace()* dove non avevamo dati disponibili ed è stata aggiunta la colonna delle nazioni estratta nelle prime righe dello script al resto dei dati.

4.2. Indici del World Economic Forum

```
reader = PdfReader('WEF_TTCR_2019.pdf')

for page in list(range(80,99)): # sono le pagine del pdf
    page_content = reader.pages[page]
    print(page_content.extract_text())

    f = open(f'text_to_clean_{page}.txt', 'w')
    f.write(page_content.extract_text())
    f.close()
```

Script 4.2.1: trasposizione del testo presente nel PDF in dei file di testo.

Dopo aver aperto il documento contenente gli indici tramite la libreria PyPDF, si esamina il suo contenuto. Come detto nel paragrafo 3.3.2 la "trasposizione" del testo dal file ad un oggetto della libreria ha fatto sì che comparissero dei caratteri non presenti nel report originale, quindi c'è stato bisogno di fare una pulizia per eliminare stringhe non necessarie ai fini della raccolta dati. Per comodità è stato trasposto tutto su dei file, creandoli e aprendoli in scrittura con la funzione *open(),* per poi scrivervi dentro il contenuto delle pagine del PDF , poi chiuderli con il metodo *close()*.

Successivamente, aprendo un file alla volta, sono stati estratti sia le nazioni sia i relativi indici, chiamando più volte *readline()* durante l'inserimento dei valori in un DataFrame, ma anche al di fuori, per far sì che venissero lette le righe vuote e si potesse quindi far avanzare il puntatore alle righe con i dati utili. Una volta arrivati alla fine del documento, esso può essere eliminato in quanto non più utile, e i dati appena salvati in un DataFrame di appoggio possono essere trasferiti in un unico DataFrame finale, che era stato creato in precedenza e che ha come colonne i nomi degli indici.

```
for index,page in zip(indexes, list(range(80,99))):
    df = pd.DataFrame(columns=['Stato', f'{index}'])
    with open(f'text_to_clean_{page}.txt', 'r') as f:
        while True:
            df = df.append({'Stato': f.readline().replace('\n', ''),
                            f'{index}': f.readline().replace('\n', '')},
                            ignore_index=True)
            f.readline()
            if f.readline()==='': # controllo se sono alla fine del file
                break

    os.remove(f'text_to_clean_{page}.txt')
    df = df.sort_values(by=['Stato']).reset_index(drop=True)

    df_final[f'{index}'] = df[f'{index}']
```

Script 4.2.2: estrazione dai file di testo degli indici e inserimento in un DataFrame.

4.3. Risultati Google

```
def start_soup(link):
    response = requests.get(link, headers=headers)
    response.raise_for_status()
    soup = BeautifulSoup(response.text, 'html.parser')
    return soup

headers = {
    'User-agent':
    'Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0'

for x in citta:
    base_link = f"https://www.google.com/search?q={x}"

    try:
        results = start_soup(base_link).find("div",
                                             {"id": "result-stats"}).text
        # estrazione del numero dal tag HTML
        reg = re.search('Circa (.*) risultati', results).group(1)
        # inserisco il numero nel dataframe, in corrispondenza della città
        google_results['risultati'] = np.where(google_results['citta']!=x,
                                               google_results['risultati']
                                               reg)
    except:
        google_results['risultati'] = np.where(google_results['citta']!=x,
                                               google_results['risultati']
                                               np.nan)
```

Script 4.3: scraping di Google per scaricare il numero di risultati ottenuti effettuando una ricerca di una città.

Per ogni città presente nei dati bibliometrici effettuiamo una richiesta ai server di Google per ottenere indietro la pagina dei risultati. Questo viene effettuato nella funzione *start_soup()*, che provvede inoltre a generare un oggetto della libreria BeautifulSoup necessario a navigare nella pagina. Una volta che la funzione ritorna questo oggetto (sempre se non sono emersi errori nella ricezione della risposta) possiamo estrarvi il numero dei risultati, andando prima a cercare il testo presente nel tag * *, e successivamente usando il metodo *search()* della libreria delle espressioni regolari. Infine inseriamo il valore nel DataFrame, oppure inseriamo NaN se la ricerca o la fase di connessione al server ha prodotto degli errori.

4.4. Risultati Booking

Il codice di scraping è pressoché identico a quello di Google, escludendo ovviamente il link di connessione e la stringa dell'espressione regolare. L'unica peculiarità che vale la pena mostrare è la manipolazione fatta ai nomi di alcune città per far sì che le relative ricerche non generino errori e quindi si riesca ad ottenere il risultato sperato.

```
def cut(citta):
    if citta.endswith('City'):
        return re.sub(r"City$", '', citta)
    return citta

booking_results['citta'] = booking_results['citta'].map(cut)
```

Script 4.4: manipolazione della Series tramite la funzione map()

La funzione *cut()*, come si può intuire grazie all'ottima leggibilità di un linguaggio di programmazione come Python, controlla che nel nome della città non ci sia il suffisso "City"; nel caso lo sia allora viene eliminato (più precisamente viene rimpiazzato da una stringa vuota). Un altro dettaglio da non far passare inosservato è l'uso di *map()*, una funzione molto potente di Pandas che consente di applicare una funzione ad ogni elemento di una Series.

4.5. Risultati TripAdvisor

Anche in questo caso è stato sfruttato il codice già scritto per lo scraping di Google e Booking; le modifiche però sono state varie, ed è stato necessario integrare la parte di automazione di cui si è già discusso nel paragrafo 3.3.5. La prima modifica ha riguardato l'inserimento di una funzione che estragga l'id di una città (un numero assegnato da TripAdvisor) da del codice HTML: dopo aver digitato del testo in una casella, il programma va in pausa per un paio di secondi con la funzione *sleep()*, in attesa che compaiano i suggerimenti di ricerca. Dopo aver controllato che ci siano effettivamente dei suggerimenti (in caso contrario la funzione termina) cerchiamo l'id nel codice HTML di questi ultimi. Eliminiamo poi il contenuto della casella di testo col metodo *clear()*, per far sì che la ricerca successiva avvenga senza errori. L'id servirà nel link necessario a connettersi alla pagina da cui scaricare il numero di attrazioni.

```python
def search_in_text_box (city):
    # digita il nome della città nella textbox
    driver.find_element(By.NAME, 'q').send_keys(city)
    sleep(2)
    # controllo che siano usciti dei suggerimenti di ricerca.
    #Se non ci sono, passo alla città successiva
    suggestions = re.findall('<a class="GzJDZ w z _S _F Wc Wh Q B- _G"',
                             driver.page_source)
    if len(suggestions) ==1:
        return ''
    match = re.findall('-g\d+', driver.page_source)
    driver.find_element(By.NAME, 'q').clear()
    return match[0]
```

Script 4.5.1: funzione che digita il nome di una città in una casella di testo e ritorna l'id di una città.

Le altre modifiche al codice invece sono state fatte dopo essersi connessi alla pagina desiderata. Il testo da scaricare non sempre si trovava nello stesso tag HTML, quindi è stata prevista una gestione di questo caso aggiungendo un except Attribute Error nel blocco try/except già presente, all'interno del quale è stato inserito un altro costrutto per la gestione degli errori. Questa scelta si è resa necessaria per non perdere l'opportunità di raccogliere dati da inserire nel DataFrame e per gestire al meglio i possibili errori. Nel caso in cui non si riesca ad arrivare al numero di attrazioni (possibilità molto remota) si provvederà ad inserire un NaN nel dataset.

```python
def scraping (df):
    for x in df['citta']:
        city_id = search_in_text_box(x)
        base_link = f"https://www.tripadvisor.com/Attractions" + city_id \
        + "-Activities-a_allAttractions.true"

        try:
            results = start_soup(base_link)
            number_of_attractions = results.find('div', class_='Ci').text.split()[-1]
            # inserisco il numero nel dataframe, in corrispondenza della città
            df['risultati'][df['citta'] == x] = number_of_attractions
        except AttributeError:
            try:
                number_of_attractions = results.find('div',
                                                     class_='aTSjG').text.split()[0]
                df['risultati'][df['citta'] == x] = number_of_attractions
            except:
                df['risultati'][df['citta'] == x] = np.nan
        except Exception as e:
            df['risultati'][df['citta'] == x] = np.nan

    return df
```

Script 4.5.2: Scraping di TripAdvisor per scaricare il numero di risultati ottenuti effettuando una ricerca di una città.

4.6. Pulizia di dati per l'analisi

Nel paragrafo 3.4 si è parlato di quanto è importante arrivare alla fase di analisi con dei dati pronti ad essere già utilizzati per la costruzione di grafici o comunque per esprimere delle considerazioni. In questo paragrafo verrà mostrato un esempio di pulizia e preparazione di un dataset, usato poi per disegnare un grafico come quello della figura 3.5.

```
df_MAG['Country'] = df_MAG['ConferenceLocation'].str.split(',').str[-1].str.lstrip()

conf_country_year = df_MAG.loc[:, ['Year','Country']]
conf_country_year = conf_country_year.reset_index().drop_duplicates()
conf_country_year.set_index('ConferenceSeriesNormalizedName', inplace=True)
conf_country_year = conf_country_year[(conf_country_year['Year'] >= 1995) &
                                      (conf_country_year['Year'] <= 2020)]
```

Script 4.6: preparazione di un dataset per l'analisi.

Il DataFrame df_MAG contiene i dati bibliometrici di cui si è già ampiamente parlato. Ogni record della tabella corrisponde ad un articolo.

La prima riga dello script 4.6 produce una nuova colonna nel DataFrame, contenente la nazione nella quale si è svolta la conferenza di cui fa parte quell'articolo. Vengono eseguite delle operazioni di manipolazione di stringhe, nello specifico per ogni istanza viene creata una lista dove ogni elemento è una parte della stringa originaria e dalla quale viene preso solo l'ultimo elemento. Dopodiché si rimuovono eventuali spazi presenti nella stringa estratta. Possiamo riassumere questo processo nel seguente schema, accompagnato da un esempio:

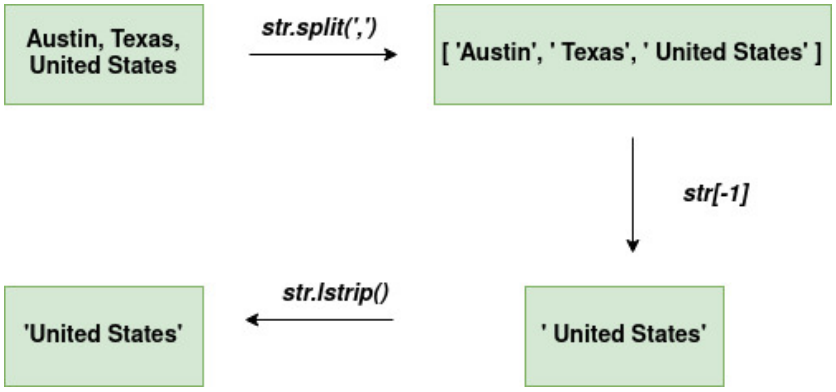

Figura 4.6: processo di estrazione dello Stato dalla stringa del luogo.

Dato che lo scopo è ottenere un DataFrame contenente esclusivamente il nome della conferenza con l'anno e il Paese in cui si è svolta, sono state conservate solo le colonne che possiedono quest'informazione con il metodo *iloc()*, dopodiché sono state eliminate le righe che si ripetevano. Per fare ciò basterebbe il metodo di Pandas *drop_duplicates()*, in questo caso non è stato sufficiente in quanto il metodo non tiene conto degli indici della tabella, che nel DataFrame in questione sono rappresentati dai nomi delle conferenze. È stato quindi effettuato un reset per non considerare la colonna dei nomi come una colonna di indici, per poi applicare il metodo citato poco fa. Successivamente sono stati reimpostati gli indici con il metodo *set_index()*.

Sono anche state eliminate le righe di conferenze svoltesi prima del 1995 e dopo il 2020 con il boolean indexing[16] di Pandas, in quanto l'analisi prevede un confronto con gli arrivi di turisti di quegli anni.

Il risultato finale è quindi il seguente:

ConferenceSeriesNormalizedName	Year	Country
disc	2014	United States
esa	2014	Poland
enter	2013	Austria
dexa	2002	France
icaisc	2006	Poland

DataFrame 4.6: conferenze, anno e nazione in cui si sono svolte.

5. Risultati

In quest'ultimo capitolo saranno presentati i risultati ottenuti dalle analisi dei dati: verranno mostrati i grafici prodotti e i relativi commenti. Ogni tipo di analisi è stata eseguita sui dati di entrambe le fonti a disposizione, vale a dire OpenCitations e MAG; verranno quindi mostrati i risultati di entrambe.

5.1. Analisi bibliometriche

Come già detto nel paragrafo 3.5, sono state svolte delle analisi "preliminari" dove non vengono inclusi i dati turistici, e dove vengono analizzati gli andamenti delle citazioni totali e medie negli anni.

In questo paragrafo e nel successivo, verranno mostrati due grafici accostati l'un altro: il primo proviene dall'analisi delle citazioni provenienti da MAG, il secondo invece da OpenCitations. In ogni caso verranno messe a confronto tre conferenze.

Prendendo delle conferenze di classe B secondo il ranking CORE, vediamo come alcuni andamenti sono pressoché identici, come possiamo notare nel grafico a sinistra della figura 5.1.1; nel grafico a destra invece c'è una somiglianza a partire dal 2014 per quanto riguarda il numero di citazioni totali.

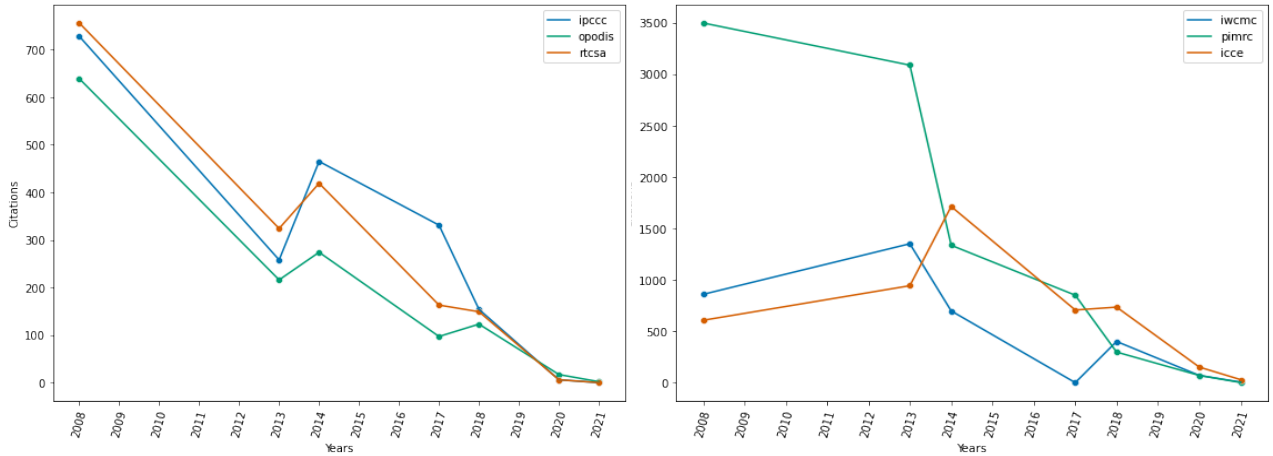

Figura 5.1.1: andamenti di citazioni totali di conferenze di classe B, ranking CORE.

Nella figura 5.1.2 abbiamo invece dati di conferenze di classe 1 secondo il ranking GRIN, con citazioni totali. In questo caso osserviamo come l'andamento può cambiare in pochissimo tempo: abbiamo infatti drastiche variazioni nel numero di citazioni, sia in aumento sia in diminuzione, anche da un anno all'altro.

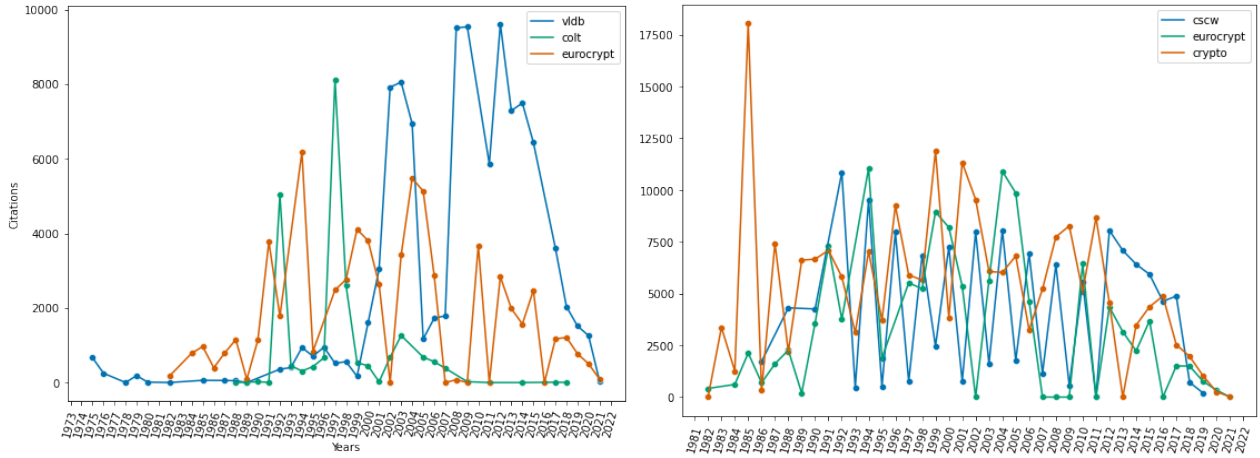

Figura 5.1.2: andamenti di citazioni totali di conferenze di classe 1, ranking GRIN.

Anche per le citazioni medie possiamo fare osservazioni interessanti: nei grafici della figura 5.1.3 abbiamo ancora conferenze di cui possediamo la valutazione del consorzio GRIN, ma stavolta appartenenti alla classe 3, quindi ritenute dal consorzio qualitativamente inferiori rispetto alle precedenti appartenenti alla classe 1. Notiamo come fino ai primi anni 2000, in entrambi i casi, abbiamo diversi aumenti radicali di citazioni, seguiti da diminuzioni altrettanto notevoli. Più ci si avvicina ai giorni attuali, più si osserva una costante diminuzione; questo è un fenomeno abbastanza prevedibile in quanto la "giovinezza" di un articolo potrebbe impedirgli di avere visibilità rispetto a pubblicazioni che hanno avuto tempo e modo di diffondersi nella comunità scientifica.

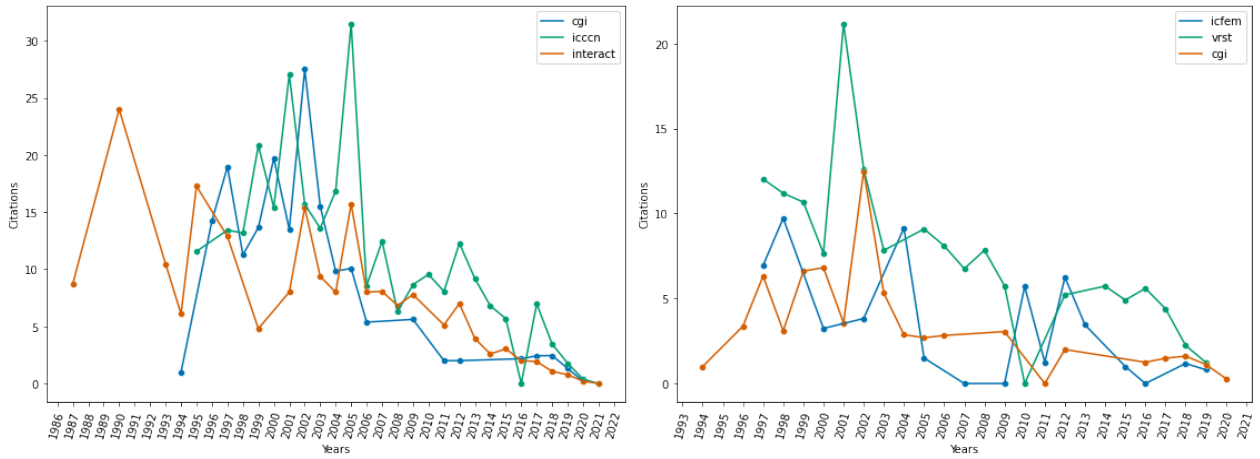

Figura 5.1.3: andamenti di citazioni medie per conferenze di classe 3, ranking GRIN.

5.2. Confronto tra citazioni medie e arrivi di turisti

Non sempre i processi di analisi restituiscono quanto sperato: i risultati di questa sezione dell'elaborato infatti non confermano quanto emerso dal lavoro pubblicato nel paper di questa ricerca. Il tentativo di far notare un legame tra le citazioni medie delle conferenze tenutesi in un determinato Stato e il numero di arrivi di turisti nella nazione non ha portato alla conclusione sperata. Come possiamo notare dai grafici di esempio di due conferenze diverse riportati nella figura 5.2, le due linee, che rappresentano i due dati appena nominati, non seguono minimamente lo stesso trend, anzi hanno comportamenti completamente diversi.

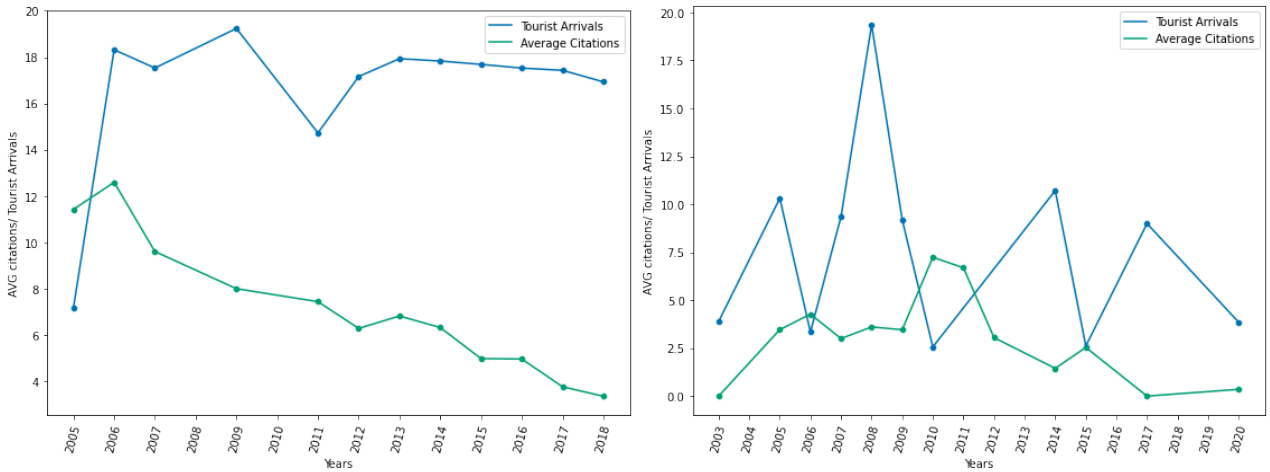

Figura 5.2: andamenti di citazioni medie negli anni, comparate al numero di arrivi di turisti nello Stato in cui si è tenuta la conferenza. Il numero di arrivi di turisti è espresso in decine di milioni.

Come rimarcato già nel paragrafo 3.5, non essendo stati testati tutti i dati delle conferenze a disposizione (sono oltre 5000, da moltiplicare per 2 se consideriamo che abbiamo due fonti diverse e quindi numeri di citazioni diversi) non si può affermare con certezza assoluta che non esista un nesso tra le citazioni e gli arrivi.

5.3. Analisi di regressione

A differenza di quello che è stato appena descritto nel paragrafo 5.2, i risultati emersi da queste ultime fasi che verranno illustrati nelle prossime righe, hanno portato a rispondere positivamente alla domanda posta all'inizio della ricerca, sulla correlazione tra citazioni e turismo di un luogo.

Nei grafici di regressione delle figure 5.3.1 e 5.3.2, rappresentanti rispettivamente i risultati ottenuti partendo dai dati di MAG e OpenCitations, possiamo notare come i punti nel piano, che rappresentano gli Stati, riescono a formare una retta che nella quasi totalità dei casi esprime una relazione lineare diretta tra le due variabili, in quanto all'aumentare della variabile x (l'indice di riferimento) aumenta anche la variabile y (citazioni medie delle conferenze svoltesi nello Stato).

Facendo un paragone tra le due figure, spicca all'occhio la differenza che c'è nella pendenza delle rette: nella figura 5.3.1 infatti i valori tendono ad andare verso l'alto più velocemente, un buon segno per il tipo di grafico che stiamo analizzando. Nella seconda immagine invece c'è un "appiattimento" generale, che però non va visto come un segnale negativo: la relazione tra le due variabili rimane comunque buona, e questo consente di affermare che esiste un nesso tra i concetti che le variabili esprimono.

L'ultimo dettaglio da commentare riguarda l'unico grafico che non ha dato un riscontro positivo in entrambi i casi: l'indice PC, ossia Price Competitiveness, ha espresso il concetto contrario degli altri, e cioè una relazione inversa tra le citazioni e il parametro stesso. Dalle informazioni presenti nella Tabella 3.3.2 sappiamo che questo indicatore misura il costo della vita, nello specifico il costo dei carburanti, tasse aeroportuali e sui biglietti aerei, costo delle strutture ricettive. La controtendenza di questo indice però non stravolge troppo ciò che è stato affermato fino ad ora, esso infatti è l'unico dei 19 indici riguardanti uno Stato che ha restituito un risultato negativo.

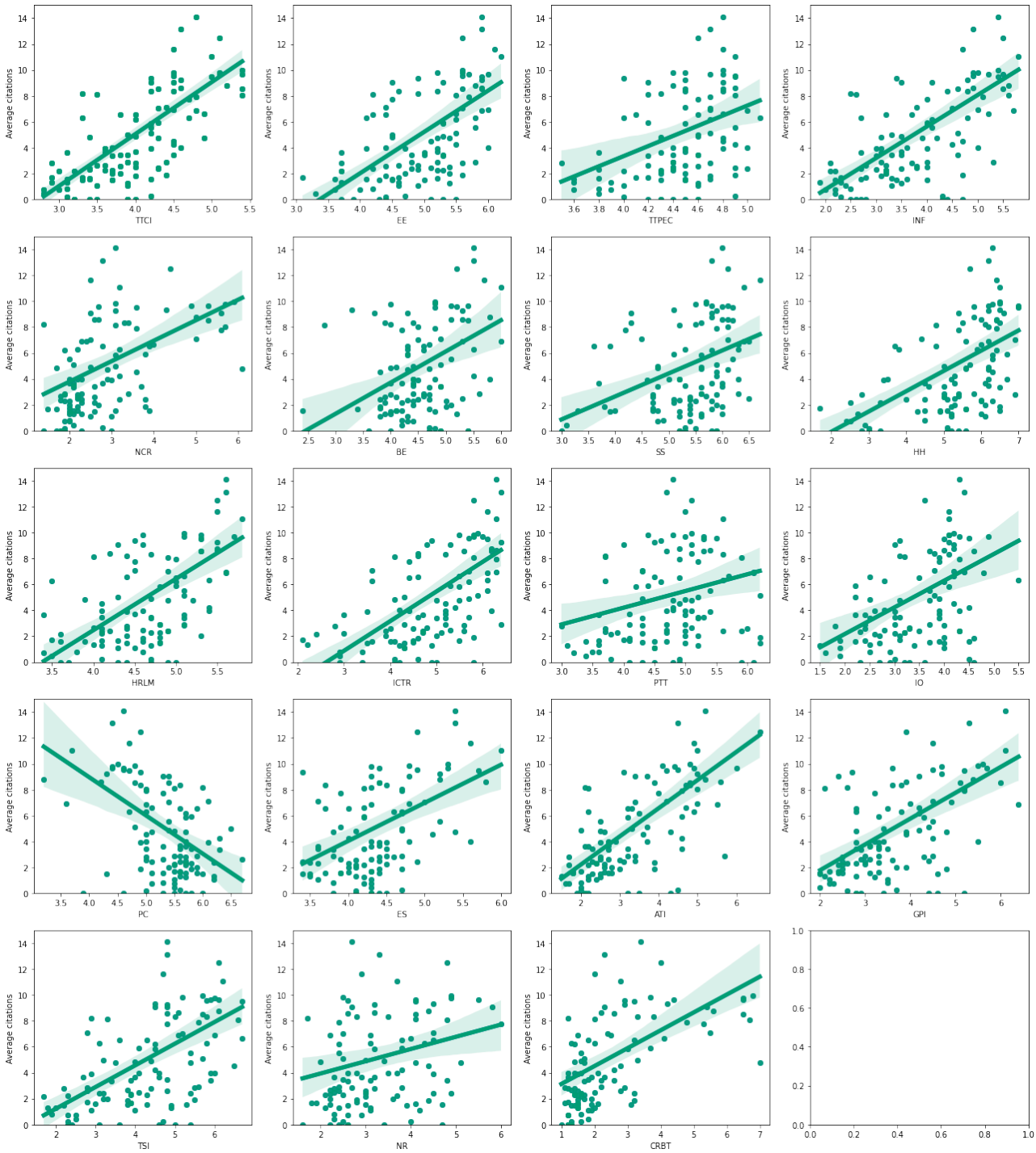

Figura 5.3.1: diagrammi di dispersione, con dati bibliometrici provenienti da MAG. Sull'asse x abbiamo gli indici del World Economic Forum (valori compresi tra 1 e 7), sull'asse y il numero di citazioni medie.

L'ultimo piano cartesiano è stato lasciato intenzionalmente vuoto, in quanto la libreria con cui sono stati costruiti i grafici permette di inserirli in una struttura matriciale per le quali vanno dichiarate il numero di righe e colonne. Il numero di grafici da disegnare era 19 quindi non si sarebbe potuta evitare in alcun modo la presenza di almeno un grafico extra.

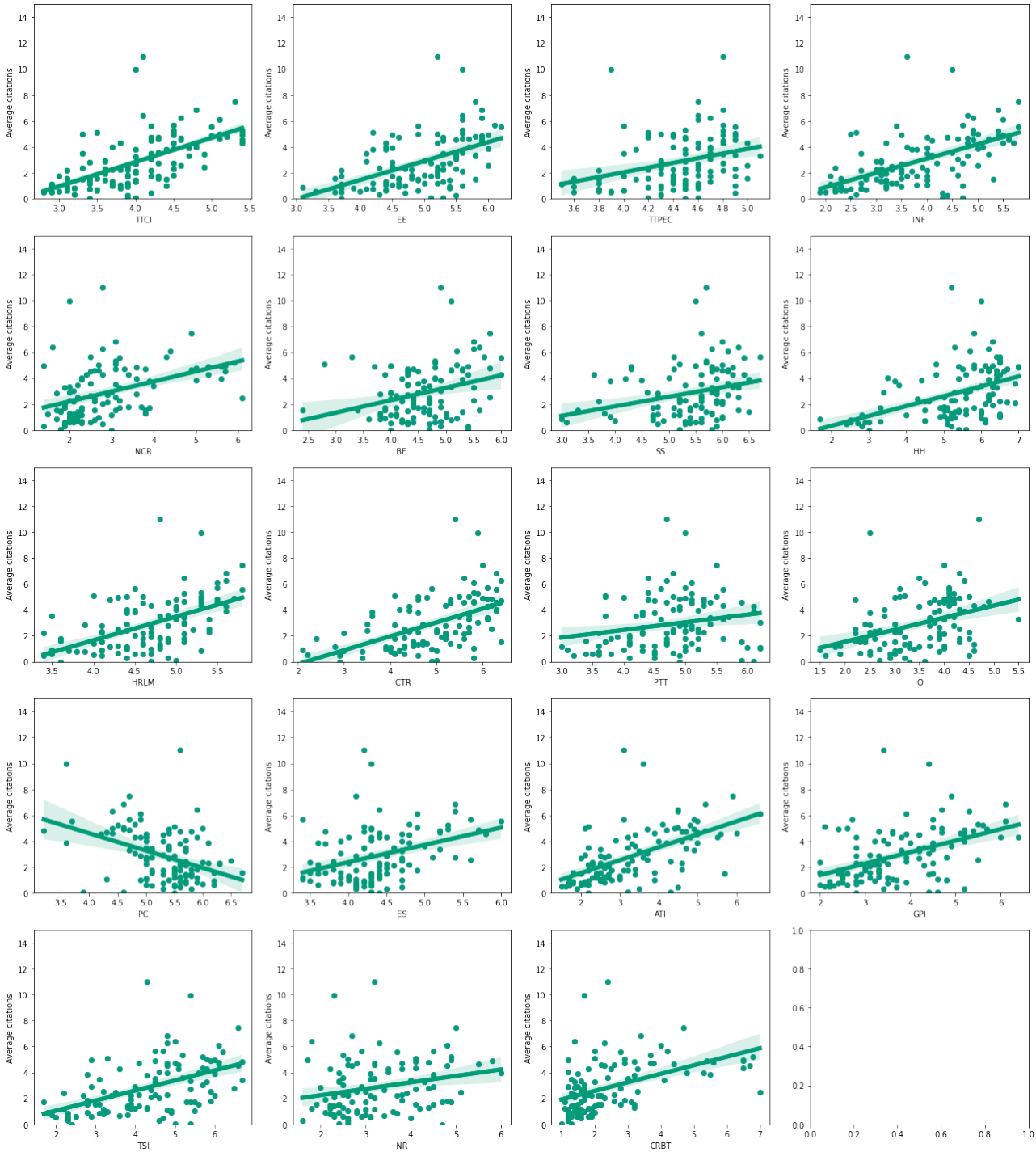

Figura 5.3.2: diagrammi di dispersione, con dati bibliometrici provenienti da OpenCitations. Sull'asse x abbiamo gli indici del World Economic Forum (valori compresi tra 1 e 7), sull'asse y il numero di citazioni medie.

L'ultimo piano cartesiano è stato lasciato intenzionalmente vuoto, in quanto la libreria con cui sono stati costruiti i grafici permette di inserirli in una struttura matriciale per le quali vanno dichiarate il numero di righe e colonne. Il numero di grafici da disegnare era 19 quindi non si sarebbe potuta evitare in alcun modo la presenza di almeno un grafico extra.

5.4. Analisi di correlazione

Quest'ultima fase di analisi è quella che probabilmente rende più chiaro il risultato ottenuto: sono stati generati infatti dei grafici che descrivono un concetto sia tramite numeri, sia tramite le varie sfumature di colori delle celle appartenenti ad essi. Come si può notare in tutte le figure di questo paragrafo, sulla destra abbiamo una barra con diversi colori e gradazioni, accompagnata da un intervallo di valori che va da -1 a 1. Questo range non è casuale, infatti corrisponde all'insieme di valori che il coefficiente di correlazione può assumere[17].

I grafici sono così strutturati: si hanno tre colonne, ognuna delle quali rappresenta un metodo di calcolo della correlazione (Pearson, Kendall, Spearman). All'interno delle celle di ogni colonna abbiamo i coefficienti di correlazione, ed ognuno si riferisce al rapporto tra le citazioni, medie o totali, e le metriche riportate alla sinistra del grafico.

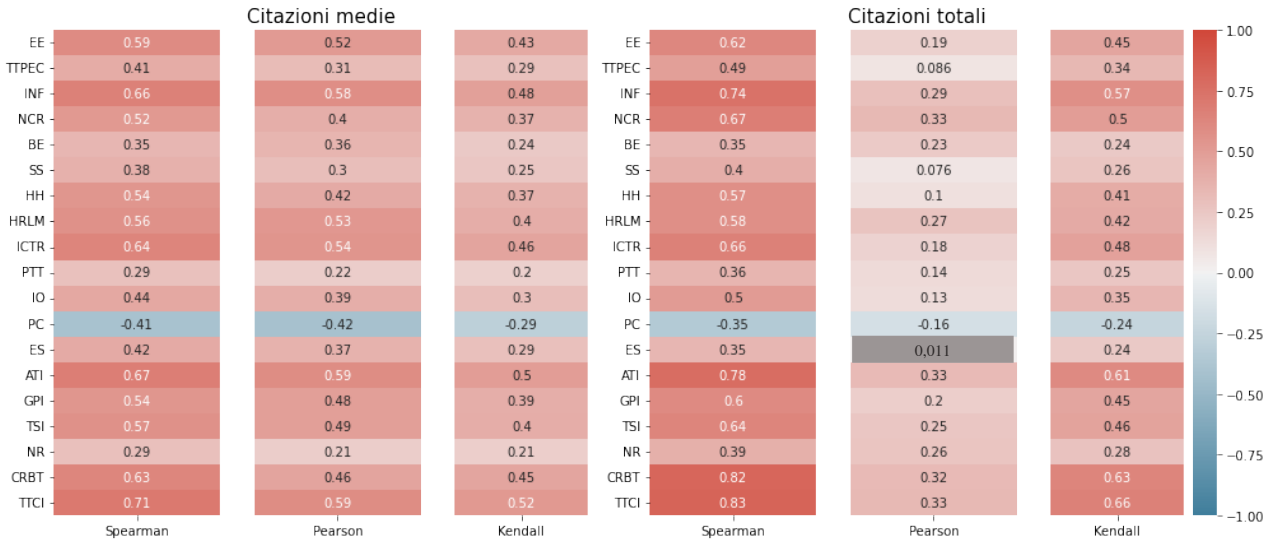

Figura 5.4.1: schemi di correlazione tra citazioni medie e totali e indici turistici di Stato.
La cella in grigio contiene un valore non significativo per lo studio,in quanto il relativo p-value è superiore a 0,05.
Fonte dati bibliometrici: MAG

Nella figura 5.4.1 sono raffigurati i grafici riguardanti le citazioni MAG e gli indici di Stato, gli stessi presenti nell'analisi di regressione. Possiamo innanzitutto confermare ciò che era emerso da quest'ultima analisi: escluso l'indice PC, mettere in relazione questi parametri porta ad avere dei

riscontri positivi, pur cambiando la funzione statistica usata per rapportare le due variabili. In questo caso possiamo ribadire il concetto anche con le citazioni totali, non utilizzate nell'analisi precedente.

Se nel grafico delle citazioni medie non notiamo particolari differenze tra le tre tipologie diverse di correlazione, nelle citazioni totali si può notare il metodo di Pearson che calcola valori molto più bassi rispetto agli altri due.

Non si può neanche rimanere indifferenti al valore di alcuni coefficienti, soprattutto nelle citazioni totali: diversi indici nella colonna di Spearman infatti si avvicinano molto al massimo valore raggiungibile. Gli stessi indici (TTCI, CRBT, ATI) hanno ottimi valori anche nelle citazioni medie, e anche se non sono alti quanto i corrispondenti nelle citazioni totali, sono comunque tra i più alti nel grafico di citazioni medie.

Dando uno sguardo anche alle corrispondenti heatmap di OpenCitations mostrate in figura 5.4.2 sembra quasi di aver fatto una fotocopia della figura 5.4.1: tutti i valori infatti, se confrontati con i "gemelli" dell'altra fonte, sono quasi identici. La differenza massima possiamo trovarla nelle citazioni medie nella colonna di Spearman, indice BE (Business Environment), dove la differenza tra i due valori è di 0.06, una quantità decisamente irrilevante considerando l'intervallo di valori nel quale ci muoviamo.

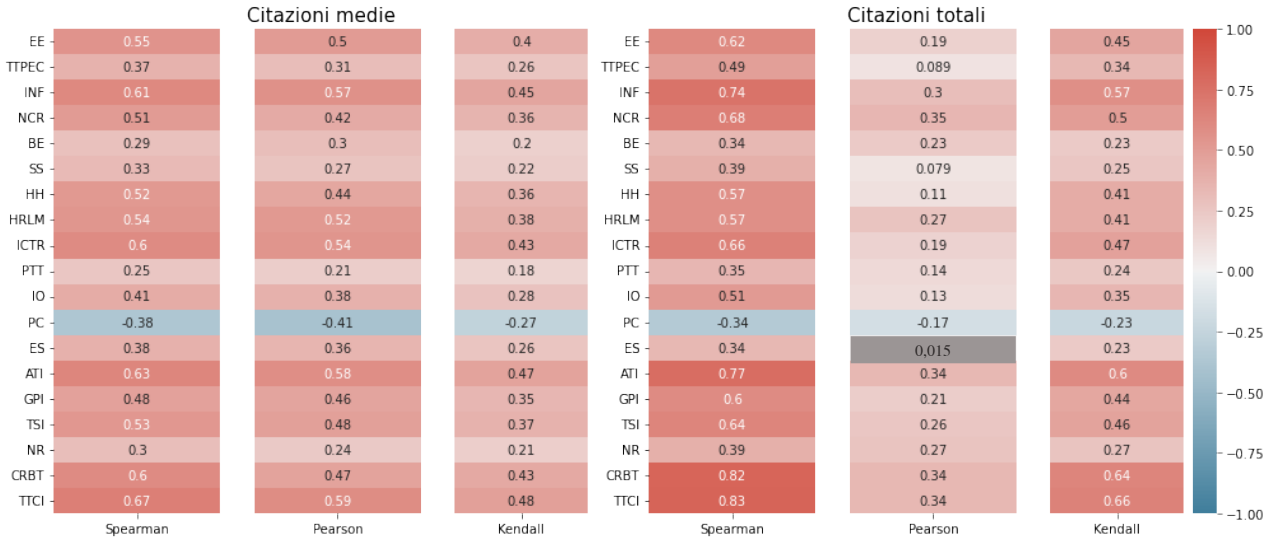

Figura 5.4.2: schemi di correlazione tra citazioni medie e totali e indici turistici di Stato.
La cella in grigio contiene un valore non significativo per lo studio,in quanto il relativo p-value è superiore a 0,05.
Fonte dati bibliometrici: OpenCitations

Passando agli indici relativi alle città, gli esiti sono stati ben diversi se confrontati con quelli appena commentati. Come possiamo notare nelle figure 5.4.3 e 5.4.4 , corrispondenti rispettivamente alle correlazioni calcolate partendo dai dati bibliometrici di MAG e OpenCitations, il valore più alto ottenuto è 0.35, nel rapporto tra citazioni totali e i risultati di ricerca Google. Spicca molto la vicinanza al colore bianco della maggior parte delle celle, che corrisponde a valori vicinissimi allo 0: fortunatamente sono tutti positivi, escluso quello relativo alle citazioni medie rapportate all'indice Booking. Questo dato non stravolge esageratamente quanto riscontrato fino ad ora: anche se i coefficienti sono molto inferiori rispetto a quelli relativi agli Stati, abbiamo comunque dei valori positivi che rivelano un legame esistente tra le variabili messe a confronto.

Nonostante la presenza di nuovi parametri turistici rispetto alla prima fase della ricerca, anche stavolta è emersa la difficoltà a trovare degli indicatori che esprimano una forte correlazione tra il turismo di una città e il successo delle conferenze: questo può essere dovuto alla vera e propria mancanza di dati del genere. Analizzare un aspetto come il turismo di un luogo relativamente piccolo probabilmente non è nell'interesse di enti/organizzazioni che svolgono questo tipo di ricerche; trovare quindi dei dataset già pronti potrebbe essere un'utopia. Questa considerazione però non deve scoraggiare la ricerca o la nascita di idee che potrebbero portare a definire dei nuovi indici, come è stato descritto in questo elaborato.

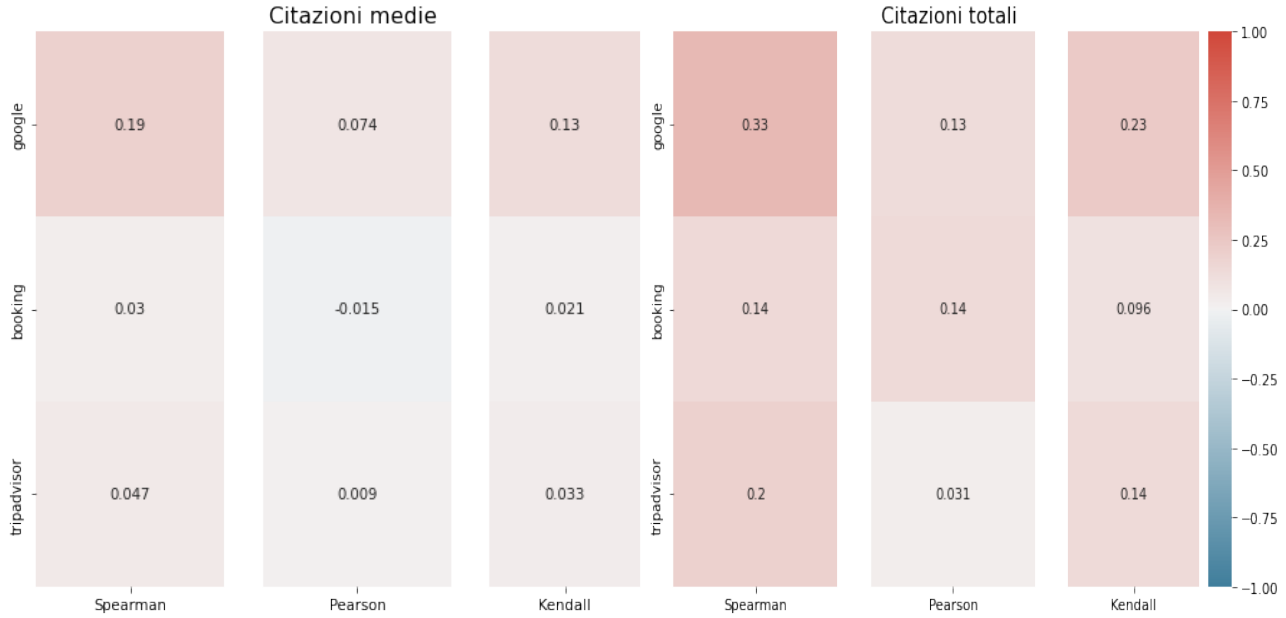

Figura 5.4.3: schemi di correlazione tra citazioni medie e totali e indici turistici di città.
Fonte dati bibliometrici: MAG

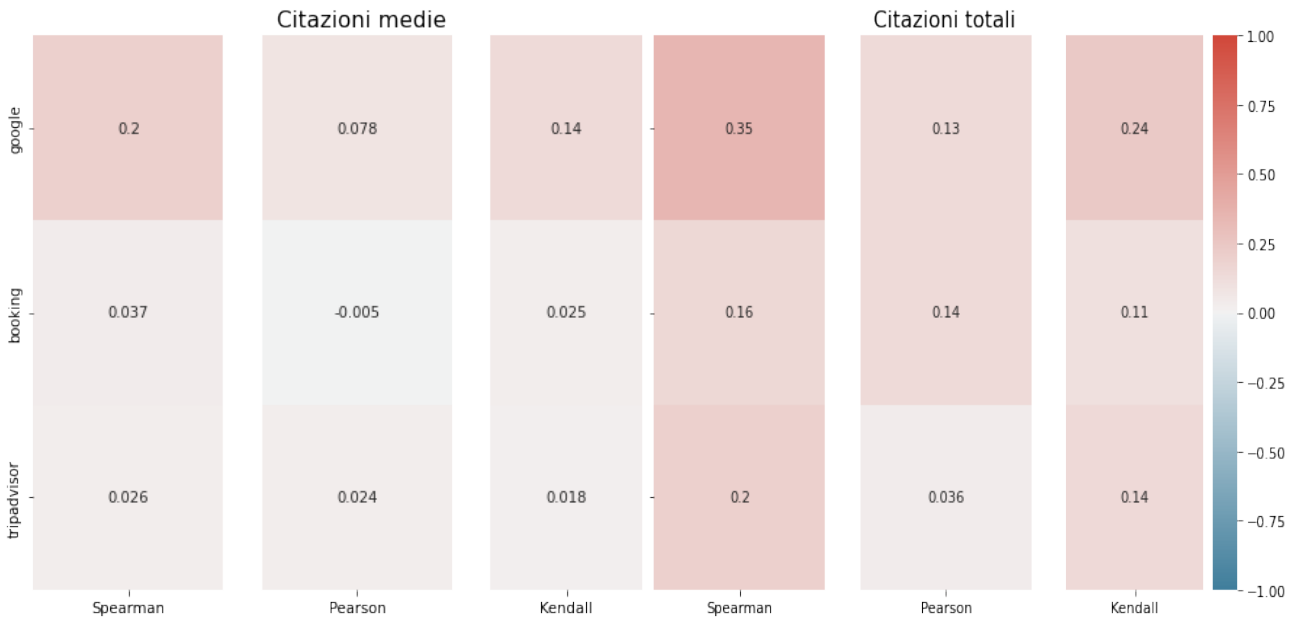

Figura 5.4.4: schemi di correlazione tra citazioni medie e totali e indici turistici di città.
Fonte dati bibliometrici: OpenCitations

Conclusione

Nell'introduzione dell'elaborato erano state poste delle domande, che sono poi state il punto di riferimento da non perdere durante la svolgimento delle attività descritte fino ad ora. Dopo aver effettuato ricerche, manipolato strutture dati, scansionato e prelevato dati da pagine online, si è arrivati alla fase finale di analisi che ha finalmente permesso di dare delle risposte soddisfacenti. Lo studio dei dati, esaminati nei diversi modi e presentati tramite schemi di vario tipo, ha consentito di affermare con certezza che esiste una correlazione tra il successo di una conferenza e il luogo in cui si è tenuta. Questo aspetto è più evidente se quando parliamo di luogo intendiamo lo Stato di appartenenza; non va comunque trascurato il nesso trovato anche con il turismo della città.

Ci sono però aspetti riguardanti la location che non sono stati tenuti in considerazione durante l'attività: per fare un esempio pratico, una conferenza che si svolge in una cittadina non classificata come grande meta turistica avrà dei valori negli indici turistici molto bassi, (le strutture su Booking saranno sicuramente poche, lo stesso vale per TripAdvisor) ma se questa cittadina si trovasse vicino ad un centro urbano molto frequentato da turisti si potrebbe affermare che abbia più probabilità di accogliere una conferenza rispetto ad una cittadina lontana da zone turistiche. Un altro caso molto simile riguarda paesini vicini a grandi città che, anche se non sono mete turistiche gettonate, hanno comunque una rilevanza di un certo tipo e quindi afflussi di migliaia di persone. Un esempio appartenente a questa tipologia di città potrebbe essere Bruxelles, centro geografico d'Europa e sede delle istituzioni principali dell'Unione Europea. In una fase successiva della ricerca quindi si potrebbe tener conto di questo fattore, magari creando un nuovo indice che misuri la vicinanza di una città ad una meta turistica.

Se ci si volesse concentrare maggiormente sul lato bibliometrico invece, si potrebbe modificare il criterio con il quale viene misurata la popolarità della conferenza: si potrebbero sfruttare altri dati riguardanti gli articoli, che non siano le citazioni, oppure si potrebbero utilizzare dati non riguardanti gli articoli ma le conferenze in generale.

Bibliografia

1 Big data- Science: https://www.science.org/doi/full/10.1126/science.1200970

2 Data science – Oracle: https://www.oracle.com/it/what-is-data-science/

3 Fasi di un processo di data science: https://www.tibco.com/it/reference-center/what-is-data-science

4 Does the Venue of Scientific Conferences Leverage their Impact? A Large Scale study on Computer Science Conferences – Luca Bedogni, Giacomo Cabri, Riccardo Martoglia, Francesco Poggi: https://arxiv.org/abs/2105.14838

5 Associazione CORE: https://www.core.edu.au/

6 Linguaggi più utilizzati nella data science: https://www.geeksacademy.it/articolo-31/i-linguaggi-piu-utilizzati-per-le-data-science/

7 Numpy: https://it.emcelettronica.com/data-analysis-ed-intelligenza-artificiale-in-python-interpretare-dati-reali-con-numpy-pandas-e-scikit-learn

8 Numpy: https://numpy.org/

9 Jupyter Notebook: https://jupyterlab.readthedocs.io/en/stable/user/export.html

10 Requests: https://pypi.org/project/requests/

11 Parser in BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/#installing-a-parser

12 Oggetti in BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/#kinds-of-objects

13 Selenium : https://www.selenium.dev/documentation/webdriver/getting_started/

14 Seaborn: https://www.html.it/pag/405220/seaborn/

15 Indici del Travel&Tourism Competitiveness Report- World Economic Forum: https://reports.weforum.org/travel-and-tourism-competitiveness-report-2019/about-the-ttcr/

16 Boolean indexing in Pandas: https://pandas.pydata.org/docs/user_guide/indexing.html#boolean-indexing

17 Correlazione: https://datascience.eu/it/matematica-e-statistica/qual-e-il-coefficiente-di-correlazione/