Università
Ca'Foscari
Venezia

DIPARTIMENTO DI SCIENZE AMBIENTALI,
INFORMATICA E STATISTICA

Corso di Laurea in Informatica

Tesi di Laurea

Analisi dei pattern di mobilità dei turisti nella città di Venezia

Relatore
Prof.ssa Alessandra Raffaetà

Correlatore
Prof. Claudio Lucchese

Laureando
Filippo Zanatta
863433

Anno Accademico
2017-18

Abstract

Il numero di turisti che visitano la città di Venezia negli ultimi anni è in continua e forte crescita. Questo fenomeno richiede un'attenta analisi per consentire di avere un più puntuale e preciso profilo di chi sono questi turisti, come e quando si muovono. Il lavoro presentato in questa tesi si inserisce in questo contesto e, attraverso l'analisi dei dati forniti dall'azienda di trasporto pubblico ACTV S.p.a, vuole studiare la mobilità dei turisti attraverso le validazioni dei biglietti all'interno del servizio di trasporto. L'azienda ha fornito i dati relativi alle validazione di due mesi e questi sono stati studiati attraverso tecniche di maching learning che hanno portato all'identificazione di diverse tipologie di turisti, descrivendo tratti identificativi di questi ultimi come il punto di origine, le tratte percorse e l'orario di percorrenza. Altri elementi che sono stati studiati attraverso gli stessi dati sono: il tempo di permanenza in determinate zone, i punti di accesso alla città e le distribuzioni temporali delle visite.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Gli ultimi anni vedono un continuo e forte aumento del numero di turisti che visitano la città di Venezia. L'obiettivo del lavoro di questa tesi è studiare la mobilità di un sottoinsieme ben definito di queste persone identificando come e quando si muovono all'interno del centro storico e delle isole. L'analisi è svolta attraverso i dati forniti dall'azienda di trasporto pubblico ACTV, che contengono le validazioni degli utenti che hanno usufruito del servizio di trasporto tra i mesi di settembre e ottobre 2018. Di ogni utente si conosce la successione di fermate e gli orari in cui è transitato, e questa traccia spazio-temporale rappresenta la sua traiettoria.

L'obiettivo primario di questo studio è l'identificazione di pattern frequenti dei turisti. Un pattern frequente è una sequenza di fermate che un numero sufficientemente elevato di utenti ha percorso Un pattern può corrispondere a un'intera traiettoria oppure a una sottosequenza di questa. Perciò potranno esserci pattern di lunghezza diversa, da poche fermate a quelli più lunghi ma attraversati da un numero più limitato di utenti.

La ricerca dei pattern ha presentato immediatamente alcune problematiche legate, non tanto alle tecniche e strumenti usati per lo studio, ma ai dati disponibili. I maggiori problemi sono tre e solo uno di questi può essere risolto in misura significativa. Il primo è legato al funzionamento del servizio di trasporto, che non prevede in uscita la validazione. Quindi, non essendo possibile sapere dove l'utente scende, questa informazione può essere solo stimata. Inoltre i dati disponibili indicano solo la fermata di validazione e non la linea, automobilistica o di navigazione, di cui l'utente si serve. Per questo motivo le possibili fermate di discesa aumentano in modo considerevole all'aumentare delle linee che transitano per quella fermata. Il secondo problema riguarda gli utenti del servizio. Nonostante sia obbligatorio, molti turisti non sempre validano il biglietto. Perciò i dati legati ad alcuni turisti potrebbero essere incompleti ed è impossibile riconoscere quelli esatti o meno. Un esempio tipico sono i titoli di viaggio con durata 24 ore, in quanto una grossa percentuale di utenti ha validato una sola fermata. Tuttavia questo comportamento è contrario al buon senso, in quanto se un turista dovesse effettuare una sola validazione, acquisterebbe un biglietto di corsa singola. Quindi con grande probabilità questi utenti non hanno validato tutte le fermate in cui sono transitati. L'ultimo problema è legato sia agli utenti che ai dati forniti, in quanto sono presenti validazioni duplicate. Queste duplicazioni, molto probabilmente, dipendono dal fatto che

gli utenti per vari motivi validano il biglietto più volte a distanza di pochi minuti. Questo problema è l'unico che è stato possibile correggere attraverso una pulizia e un filtraggio con criteri automatici che hanno consentito di eliminare la maggior parte delle validazioni duplicate.

L'analisi compiuta si è svolta in varie fasi che hanno richiesto tecniche e strumenti diversi. Visto che l'interesse dello studio è rivolto ai turisti, dal dataset contenente le validazioni sono state selezionate quelle relative ai titoli di viaggio con durata maggiore o uguale alle 24 ore. Infatti questa tipologia di biglietto è acquistata dai turisti. Successivamente sono stati creati e testati alcuni script per la pulizia automatica dei dati. Le operazioni precedenti hanno avuto come scopo rielaborare i dati per renderli corretti e pronti ad essere analizzati attraverso strumenti di maching learning.

L'analisi vera è propria è stata svolta provando diversi algoritmi di clustering con diversi parametri. Il clustering è un processo che partiziona un insieme in sottoinsiemi tali che gli elementi di ogni sottoinsieme siano simili tra loro e distanti da quelli degli altri gruppi. Il passaggio delicato, che ha richiesto anche la maggior parte del tempo di lavoro, è stato determinare cosa significasse per due turisti essere simili o distanti. Questa fattore è l'elemento fondamentale per poter eseguire gli algoritmi di clustering. Sono state testate diverse misure comunemente utilizzate e ne è stata definita una di nuova più adatta alle esigenze di questo lavoro.

Oltre a dover determinare la misura di distanza migliore per questo problema, si è dovuto lavorare in parallelo per ricercare ottimizzazioni del software. Nonostante l'utilizzo di una macchina virtuale molto performante, i programmi e gli script iniziali non potevano essere eseguiti per due motivi: la saturazione della memoria e i tempi di esecuzione non accettabili. Le ottimizzazioni hanno richiesto la lavorazione con diversi linguaggi di programmazione e librerie, la ricerca una nuova rappresentazione dei dati e la sincronizzazione di diversi programmi tra loro.

Le modalità, le tecniche e gli strumenti descritti fino a questo momento hanno prodotto dei risultati significativi, identificando diversi cluster di turisti che si caratterizzano per le tratte percorse. Inoltre per ciascun cluster è stato effettuato un secondo livello di partizionamento per determinare le fasce orarie degli spostamenti degli utenti

Il Capitolo 2 introduce il contesto in cui si inserisce questa tesi, mostrando in dettaglio i numeri del turismo e descrivendo brevemente il servizio di trasporto e il tipo di analisi che si è svolta.

Il Capitolo 3 descrive gli strumenti sia hardware che software che sono stati usati per svolgere questo lavoro.

Il Capitolo 4 illustra la fase di preparazione dei dati. Inizialmente descrive in maniera dettagliata i dati disponibili, come sono stati filtrati, puliti e rielaborati per poter essere facilmente accessibili e manipolabili. L'ultima parte presenta una panoramica dei dati attraverso statistiche, grafici e tabelle che ne permettono una facile comprensione.

Il Capitolo 5 espone in dettaglio l'analisi compiuta. Descrive inizialmente il supporto teorico su cui si base lo studio e infine mostra l'analisi compiuta presentando i risultati ottenuti.

Infine nel Capitolo 6 vengono proposte le conclusioni, richiamando brevemente i contenuti ed esponendo le possibili continuazioni di questo studio.

Capitolo 2

Descrizione del problema

La città di Venezia, negli ultimi anni, registra un andamento crescente dei flussi di turisti che visitano il centro storico e le principali isole. Le stime più recenti mostrano un afflusso di oltre 33 milioni di turisti a fronte di una popolazione di appena 50 mila residenti. L'annuario del turismo[1], una raccolta di dati commissionata dal Settore Turismo della Città di Venezia, offre un insieme ragionato di dati certi e quantificabili relativi ai principali indicatori turistico-culturali della Città di Venezia e della sua area metropolitana. Alla data odierna è disponibile la settima versione riferita all'anno 2017[2]. Un aspetto interessante è legato alla forte crescita della vendita dei biglietti turistici AVM/ACTV dall'anno 2013 al 2017, che evidenzia un netto aumento nell'ultimo anno. La tabella 2.1 e il grafico 2.2 riassumono questa tendenza.

Il forte aumento delle vendite rende necessario lo studio del comportamento degli utenti. La presente tesi si inserisce in questo contesto e si pone l'obiettivo di analizzare la mobilità di una porzione dei turisti attraverso l'analisi dei dati rilasciati dall'azienda ACTV S.p.a.

ACTV (Azienda del Consorzio Trasporti Veneziano) è la principale azienda di trasporto pubblico nel comune di Venezia e delle zone limitrofe. La rete di trasporto è composta dalle seguenti tre componenti:

- la rete di navigazione, che collega il centro storico di Venezia alle isole della laguna;
- la rete tranviaria urbana che unisce il comune di Venezia, Chioggia e le isole;
- la rete automobilistica extraurbana che si estende fino alle province di Padova, Treviso e Rovigo.

Per usufruire del servizio è necessario validare il proprio titolo di viaggio nelle obliteratrici poste all'interno degli autobus o negli imbarcaderi per il servizio di navigazione. All'uscita del servizio non è necessario una timbratura di conferma, questo significa che non è possibile sapere dove sono dirette le persone, ma solo stimare la loro destinazione.

I dati forniti da ACTV riguardano tutti i tipi di biglietto esclusi gli abbonamenti, questi sono:

- biglietti ordinari, con durata limitata, tipicamente 75 minuti;

Tipologia biglietto	2013	2014	2015	2016	2017	Var. % 16-17
***Biglietto 12 ore** *	879.454	561.703				
Biglietto 24 ore	340.039	762.117	1.430.782	1.574.126	1.715.037	9,0%
di cui Venezia Metropolitana 24 **					47.411	
***Biglietto 36 ore** *	208.211	136.340				
Biglietto 48 ore	184.020	276.494	423.230	442.091	510.345	15,4%
Biglietto 72 ore	319.922	303.642	304.660	279.077	350.835	25,7%
Biglietto 7 giorni	136.848	117.694	120.290	96.956	144.062	48,6%
Biglietto ordinario* **	5.000.593	5.247.791	4.855.459	4.847.236	5.812.556	19,9%
TOTALE	**7.069.087**	**7.405.781**	**7.134.421**	**7.239.486**	**8.532.835**	**17,9%**

* Non più emesso da agosto 2014.
** Introdotto dal 21 giugno 2017.
*** Incluso ordinario traghetto, ordinario a bordo e aventi origine e/o destinazione l'aeroporto "Marco Polo". La nuova aggregazione modifica i dati, che pertanto si scostano da quelli riportati nella medesima tabella nelle precedenti edizioni. Il dato pubblicato nel 2016 è stato corretto,

Fonte: Comune di Venezia – Settore Traffico Acqueo, Mobilità e Trasporti

Figura 2.1: Principali tipologie di biglietti turistici AVM / ACTV: vendite anni 2013-2017

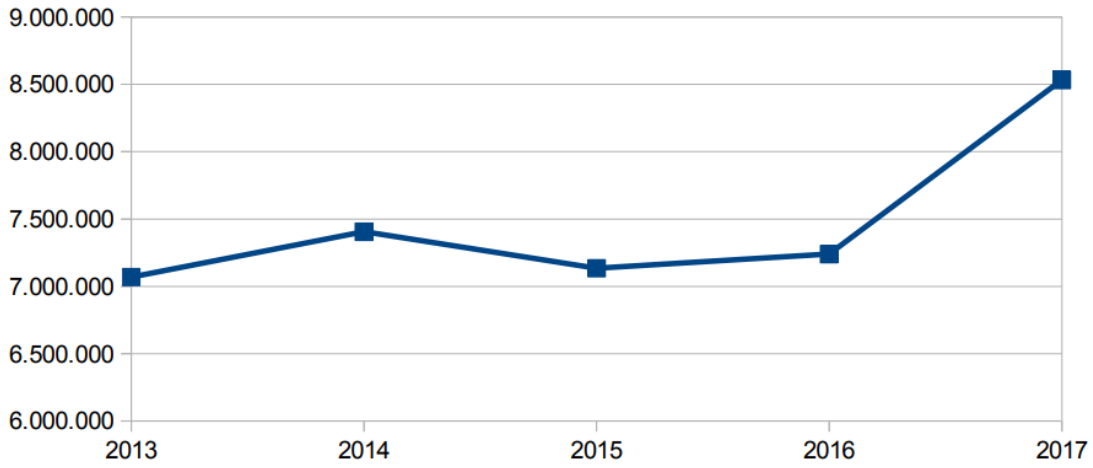

Figura 2.2: AVM/ACTV: totale vendite principali titoli di viaggio anni 2013-2017

- biglietti a tempo, con validità di 1,2,3,7 giorni.

Si può ragionevolmente assumere che i titoli con validità di più giorni siano per la maggior parte acquistati da turisti. Residenti, lavoratori e studenti è più probabile che utilizzino abbonamenti mensili o annuali o biglietti con corse singole.

L'analisi si concentra principalmente su due aspetti legati ai turisti:

1. lo studio dei percorsi più comuni all'interno del centro storico e nelle isole;

2. lo studio della permanenza media dei turisti in determinate zone.

L'analisi di seguito presentata potrà essere utile all'ACTV in quanto potrebbe costituire la base per una razionalizzazione del trasporto pubblico. Una conoscenza più approfondita consente il miglioramento del servizio di navigazione erogato per le principali tratte percorse dai turisti e, al tempo stesso, lo studio della permanenza media in determinate aree, potrà garantire una presenza dei mezzi più sensibile alle esigenze dei turisti.

Inoltre questo studio potrà aiutare la gestione della manutenzione dei battelli, ogni barca, infatti, richiede un periodo di 5 mesi per la propria manutenzione. Una schedulazione che ignora i flussi del turismo rischia di rendere inattivi troppi battelli in periodi in cui sono più richiesti. Quindi compromettere il livello di qualità del servizio non assicurando un numero sufficiente e adeguato di mezzi per fasi dell'anno più o meno attive per i turisti.

I casi di studio devono tenere conto della privacy e dell'anonimato dei soggetti interessati. L'analisi svolta in questo lavoro garantisce la salvaguardia della privacy di ogni utente in quanto non esiste alcun modo di associare i dati in possesso ad una persona specifica.

I dati analizzati in questa tesi sono stati forniti da ACTV e sono suddivisi nelle seguenti categorie:

1. dati specifici forniti per l'analisi di questa tesi;

2. dati pubblici accessibili da chiunque nel formato GTSF, General Transit Feed Specification.

2.1 Dati forniti da ACTV

L'azienda ACTV ha fornito le validazioni di tutti i titoli di viaggio nel periodo compreso tra 11 settembre 2018 e 12 novembre 2018. Le validazioni si riferiscono all'utilizzo di biglietti ordinari, non degli abbonamenti, e ammontano a circa 5 milioni.

2.2 Dati pubblici GTFS

La seconda fonte di dati è distribuita secondo il formato GTFS [3] come open data al seguente link: `http://actv.avmspa.it/sites/default/files/attachments/opendata/`.

Il formato GTFS static è uno standard introdotto da Google per definire un formato comune per gli orari dei trasporti pubblici e le loro informazioni geografiche. È nato per consentire agli sviluppatori di creare diverse applicazioni che possano lavorare con questi dati in modo intercambiabile.

GTFS prevede una serie di file di testo compressi in un unico file zip, ognuno di questi file descrive un particolare aspetto del trasporto: le fermate, le linee, le tratte e ulteriori dati.

Capitolo 3

Tool di sviluppo

La tesi prevede l'analisi dei dati forniti da ACTV e una loro presentazione grafica che ne permette una veloce e chiara comprensione. Questo capitolo espone e descrive i linguaggi e gli strumenti che sono stati impiegati per l'analisi.

3.1 Strumenti hardware

L'analisi è stata svolta, principalmente, attraverso il servizio di cloud engine offerto da Google.

Google Compute Engine è un servizio che offre macchine virtuali ospitate negli innovativi data center di Google. I vantaggi principali sono l'avvio rapido, l'archiviazione su disco permanente e le prestazioni sempre costanti. Il servizio permette di personalizzare la propria macchina virtuale scegliendo tra un set di macchine predefinite o personalizzabile per i propri carichi di lavoro. I parametri personalizzabili variano dai più intuitivi, come il numero di virtual CPU richieste, la quantità di ram richiesta e la dimensione del disco, fino alla scelta dell'area geografica e della zona richiesta per la sede fisica della macchina. La macchina virtuale creata per questa tesi ha le seguenti caratteristiche:

- 8 virtual CPU, che consentono un discreto livello di parallelizzazione per eseguire diverse operazioni;
- 52 GB di ram, che permettono di lavorare con l'intero dataset caricato in memoria e gestire in modo veloce matrici di dimensione elevata;
- 50 GB di disco permanente, per la memorizzazione del dataset e di dati intermedi anche tra sessioni differenti;
- sistema operativo Ubuntu 16 server.

3.2 Linguaggi utilizzati

Per l'analisi sono stati utilizzati i seguenti linguaggi:

- Python, per lo studio e l'analisi dei dati;

- C, per operazioni computazionalmente pesanti che richiedono velocità maggiore.

Il linguaggio python, uno dei più diffusi per l'analisi e lo studio dei dati, è stato scelto per la capacità di gestire velocemente grandi quantità di dati, il supporto di molte librerie di machine learning e la velocità di scrittura di script per l'analisi.

Il linguaggio C è stato usato in misura minore rispetto a python. In quanto linguaggio compilato e non interpretato il suo ruolo è stato computare matrici molto grandi ed effettuare altre operazioni che python non avrebbe gestito con la stessa velocità.

3.3 Strumenti software

- Jupyter Notebook, come ambiente di sviluppo per python;
- Visual Studio Code, IDE per scrivere più comodamente il codice C;
- Overleaf, per la stesura in LaTeX della presente tesi.

Jupyter Notebook è una applicazione web che consente di creare documenti (noti con il nome di notebook) che contengono codice e testo formattato in markdown. Il vantaggio principale è di essere sia documenti facilmente leggibili e interpretabili sia documenti che possono essere eseguiti per compiere l'analisi sui dati.

3.4 Librerie software utilizzate

Per facilitare la gestione dei dati e la loro analisi sono state usate le seguenti librerie di python:

- Numpy [4], la libreria fondamentale per l'analisi matematica in python. Contiene tra le altre cose, la gestione di matrici N-dimensionali, funzioni di algebra lineare, molte funzioni avanzate e strumenti per l'integrazione di programmi in C/C++ e Fortran;
- Sklearn [5], una libreria di machine learning che implementa strumenti semplici ed efficienti per il data mining e il data analysis;
- Matplotlib [6], una libreria per la realizzazione di grafici di varie tipi;
- Jupyter-gmaps [7], una comoda libreria per lavorare con le mappe di Google all'interno dell'ambiente di jupyter-notebook. Per il suo utilizzo è necessaria l'autenticazione attraverso una API key, ottenibile gratuitamente.

Nei programmi C, invece, sono state necessarie solo due librerie per la gestione del multithread:

- pthread.h, per la creazione e gestione dei thread;

- semaphore.h, per la gestione della concorrenza e degli accessi multipli alla stessa risorsa.

Il multithreading migliora le prestazioni dei programmi consentendo l'esecuzione parallela di diverse operazioni.

3.5 Permanenza dei dati

Il salvataggio dei dati è stato effettuato creando dei file in formato json. La scelta è motivata dalla facilità di interagire con questi file e dalla possibilità di convertire direttamente questi ultimi in variabili utilizzabili dal linguaggio python. L'ulteriore vantaggio è che l'intero dataset è caricato completamente in memoria, perciò modalità non è necessario accedere continuamente in memoria per il reperimento delle informazioni, come generalmente accade interrogando un comune database SQL. I dettagli e le caratteristiche dei file generati saranno analizzati e descritti nel Capitolo 4.

Capitolo 4

Preelaborazione dei dati

Questo capitolo descrive come sono stati organizzati i dati grezzi e la loro prima rielaborazione. Prima di tutto saranno descritti i dati GTFS in dettaglio e la loro riorganizzazione. Poi saranno presentati i dati legati alle validazioni, i metodi utilizzati per la pulizia e il loro filtraggio, infine alcuni grafici e mappe che esprimono e interpretano questi dati.

4.1 Dati GTFS

Come accennato nel Capitolo 2, GTFS [3] è uno standard introdotto da Google per definire un formato comune per gli orari dei trasporti pubblici e le relative informazioni geografiche.

Un feed GTFS è una collezione di file di testo compressi in un archivio zip. Ciascun file, che risulta essere in formato CSV[1], descrive e contiene i dati di una tabella di un database relazionale SQL, il nome della tabella corrisponde al nome del file, ignorando l'estensione. Alcuni di questi file sono opzionali in quanto possono essere omessi. Non sono incluse le informazioni riguardanti il tempo reale dei mezzi. Di seguito sono elencati e descritti i principali file:

- agency.txt, descrive l'agenzia che fornisce il servizio di trasporto;
- stops.txt, individua i luoghi in cui i veicoli possono far scendere o salire i passeggeri;
- routes.txt, descrive un insieme di percorsi (trip) che appartengono alla stessa linea;
- trips.txt, descrive un percorso come una sequenza di due o più fermate con i relativi orari;
- stop_times.txt, l'orario in cui una linea si ferma e parte da ciascuna fermata;
- calendar.txt, le date per i servizi durante la settimana;

[1]Il Comma-Separated Values (abbreviato in CSV) è un formato di file utilizzato per l'importazione ed esportazione di una tabella di dati, generalmente ogni riga rappresenta un record e ciascun campo del record è diviso da un carattere speciale, tipicamente una virgola.

- calendar_dates.txt, le eccezioni alla normale programmazione settimanale;
- shapes.txt, le regole per disegnare il tracciato di un percorso nelle mappe.

ACTV fornisce due diversi feed GTSF: uno legato al servizio di navigazione e l'altro legato a quello automobilistico. Entrambi sono frequentemente aggiornati per correggere piccoli errori o aggiornare il servizio offerto. Durante il periodo analizzato sia il servizio di navigazione che quello automobilistico sono stati aggiornati più volte. Per motivi di completezza, sono stati uniti i dati relativi alle versioni dei feed GTSF dalla 375 alla 378 per quanto riguarda la navigazione e dalla 436 alla 444 per il servizio automobilistico.

Da questi dati sono state prelevate le informazioni relative a tutte le fermate contenute nel file stop.txt.

Stop.txt è un file csv composto da 12 campi. Gli unici campi necessari a questa analisi sono:

- *stop_id*, l'identificativo della fermata;
- *stop_name*, il nome della fermata;
- *stop_lat*, la latitudine geografica;
- *stop_lon*, la longitudine geografica.

Il procedimento descritto nel prossimo paragrafo è lo stesso adottato per entrambe le tipologie di servizio.

Il primo passo è stato caricare tutte le fermate di tutte le versioni. Per ogni fermata, le informazione utili sono: id, nome, latitudine e longitudine. Le fermate sono state raggruppate per l'id in modo da evitare i duplicati, per i campi rimasti latitudine e longitudine sono rimasti invariati, mentre per il nome è stata scelta l'ultima versione. È probabile che la maggior parte delle modifiche alle differenti versioni abbia avuto effetto su altri aspetti del servizio piuttosto che sulle fermate.

Il risultato è stato l'identificazione di 2364 fermate automobilistiche distribuite principalmente nella città metropolitana di Venezia e 142 di navigazione presenti nel centro storico, al Lido, a punta Sabbioni e nelle isole vicine.

Gli identificativi delle fermate del servizio di navigazione e automobilistico costituiscono insiemi disgiunti, per cui non è necessario modificarli per poter aggregare insieme i due gruppi di fermate.

Il passo successivo è stato raggruppare queste fermate per riuscire a razionalizzare meglio la loro presenza sul territorio. Le fermate degli autobus, per ovvie ragioni, sono in numero molto maggiore rispetto a quelle di navigazione ma allo stesso tempo sono estremamente sottoutilizzate dai turisti. Le due tipologie di fermate sono state gestite con due diverse modalità che i prossimi paragrafi andranno a descrivere.

4.1.1 Fermate servizio automobilistico

Per quanto riguarda il servizio automobilistico sono state individuate cinque zone chiave e di interesse, che permettono di identificare caratteristiche importanti dei nostri utenti:

- *Lido*, in quanto descrive il servizio automobilistico dell'isola del Lido;
- *Aeroporto*, in quanto descrive gli arrivi o ritorni dall'aeroporto;
- *Piazzale Roma*, in quanto descrive le fermate degli autobus che arrivano a Venezia;
- *Stazione di Mestre*, in quanto descrive un punto di riferimento della città;
- *Terraferma non specificata*, che racchiude tutte le altre fermate.

Un candidato ad essere una zona di interesse è stata Chioggia, ma il numero davvero limitato di validazione non lo ha reso necessario.

Per ognuna delle prime quattro zone sono state fissate le coordinate geografiche e un raggio di interesse che descrive una circonferenza. Tutte le fermate automobilistiche all'interno di una di queste circonferenze vengono associate alla corrispondente zona. Tutte le fermate escluse vengono inserite nell'ultima zona. Per uniformità con le altre zone, *Terraferma non specificata* è identificata dalle coordinate del centro di Mestre. La tabella seguente mostra quante fermate sono state associate a ciascuna zona.

Zona	Numero fermate
Lido	100
Aeroporto	6
Piazzale Roma	19
Stazione di Mestre	9
Mestre centro	2230

4.1.2 Fermate servizio di navigazione

L'aggregazione delle fermate del servizio di navigazione comporta problemi maggiori rispetto a quello automobilistico. Infatti due fermate vicine dal punto di vista spaziale possono presentare differenze sostanziali.

La prima, la più semplice e intuitiva, è legata alla geografia della città: due fermate vicine possono essere divise da un canale. Per cui fermate vicine nello spazio potrebbero essere molto distanti da raggiungere.

La seconda è legata al significato delle fermate, fermate vicine potrebbero avere un contenuto informativo molto diverso. Il caso più esemplificativo è relativo alle fermate della stazione e di piazzale Roma. Queste fermate sono raggiungibili in meno di 5 minuti a piedi, tuttavia rappresentano due tappe completamente diverse: quella della stazione rappresenta generalmente quella di un turista arrivato in treno, quella di piazzale Roma, di un turista che tendenzialmente ha alloggiato nella terraferma ed è arrivato in autobus o taxi. Questa sottile differenza rappresenta un fattore importante nell'individuazione dei comportamenti dei turisti. Per queste ragioni, l'aggregazione delle fermate non può estendersi alla creazione di zone di interesse come avvenuto per le fermate automobilistiche, ma si deve limitare a raggruppare tra loro quelle che appartengono allo stesso imbarcadero o gruppo di fermate.

L'esempio seguente riassume come è stata eseguita l'aggregazione delle fermate per il servizio di navigazione, mostrando le sei fermate di San Zaccaria e le informazioni disponibili. La mappa in Figura 4.1 mostra le posizioni nei rispettivi imbarcaderi. Dal momento che appartengono allo stesso gruppo, queste fermate sono raggruppate insieme.

Id	Nome	Latitudine	Longitudine
5009	S. Zaccaria (Pieta') "A"	45.433395	12.344964
5011	S. Zaccaria (Jolanda) "D"	45.433456	12.342809
5012	S. Zaccaria (Jolanda) "C"	45.433491	12.343048
5013	S. Zaccaria (Danieli) "F"	45.433342	12.342007
5014	S. Zaccaria (Danieli) "E"	45.433384	12.342215
5076	S. Zaccaria (M.V.E.) "B"	45.433514	12.344095

Figura 4.1: Le sei fermate di San Zaccaria

L'aggregazione è stata fatta con un processo automatico che valuta il nome delle fermate e la loro distanza. Se due fermate condividono lo stesso nome che al massimo differisce per la lettera identificatrice dell'imbarcadero o per pochi altri caratteri e la loro distanza geografica è entro un certo limite fissato, allora sono aggregate insieme. Il rappresentate del gruppo è scelto come la fermata mediamente più vicina alle altre, il suo nome è pulito da eventuali caratteri di contorno, come l'identificativo dell'imbarcadero.

Il risultato finale è stato di ridurre le fermate da 142 a 69. La mappa in Figura 4.2 mostra queste fermate.

4.1.3 Salvataggio dei dati

I dati precedentemente elaborati sono stati salvati in modo permanente in quattro file JSON separati, la tabella 4.1 illustra la dimensione e la numerosità di questi file.

- mare_stop.json, descrive le fermate originali di navigazione. Consiste in una lista di tuple, una tupla descrive una fermata e in ordine contiene: l'id, il nome, la latitudine, la longitudine;

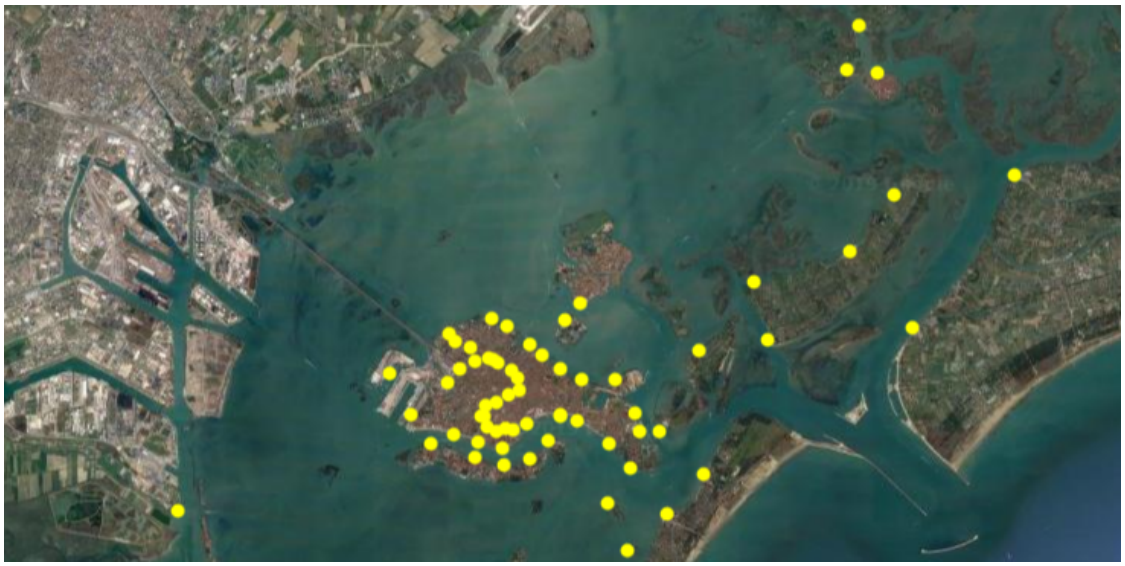

Figura 4.2: Le fermate di navigazione escluse quelle verso Chioggia

- terra_stop.json, come per mare_stop.json, descrive le fermate del servizio automobilistico;
- stop.json, composto come il file mare_stop.json, descrive le nuove fermate che sono nate dall'aggregazione di quelle originali, contiene sia quelle di navigazione che quelle automobilistiche;
- stop_function.json, questo file contiene un dizionario che mappa l'id di ogni fermata di terra e di mare originale in quella nuova.

L'utilizzo dei file stop.json e stop_function.json rende l'analisi compiuta parametrica rispetto al raggruppamento, una diversa scelta di raggruppamento può essere facilmente integrata.

File	Dimensione file	Numero elementi lista/dizionario
mare_stop.json	7.8 KB	142
terra_stop.json	131 KB	2364
stop.json	3.7 KB	74
stop_function.json	30 KB	2506

Tabella 4.1: Elenco dei file prodotti

4.2 Dati sulle validazioni

Le validazioni rilasciate da ACTV si presentano come un file csv composto dalle seguenti colonne:

1. *DATA_ VALIDAZIONE*: La data e l'ora in cui la validazione è stata effettuata, nel formato gg/mm/aaaa hh:mm;

2. *SERIALE*: L'identificativo univoco di un singolo biglietto;
3. *FERMATA*: L'identificativo univoco della fermata in accordo con gli identificativi delle fermate GTFS;
4. *DESCRIZIONE*: La descrizione della fermata;
5. *TITOLO*: Il codice del titolo di viaggio a cui si riferisce questo biglietto;
6. *DESCRIZIONE_ TITOLO*: La descrizione testuale del titolo di viaggio.

Ogni riga di questo file descrive una singola validazione, le validazioni di un utente sono tutte identificate dallo stesso *SERIALE*. Il file contiene 4876779 di righe, di cui la prima è l'intestazione delle colonne CSV.

Il range temporale di queste validazioni parte dal 2018-09-11 00:01:00 e arriva al 2018-11-12 02:48:00.

I primi numeri mostrano:

- 1672 fermate distinte in cui gli utenti hanno effettuato almeno una validazione;
- 1751985 utenti distinti tra loro;
- 163 diversi titoli di viaggio con diverse durate.

4.3 Titoli di viaggio

La prima analisi effettuata per le validazioni riguarda i titoli di viaggio. Per poter individuare i turisti è stato necessario identificare il loro tipo di biglietto. I titoli di viaggio totali sono 163 e sono riportati in appendice nella Tabella A.1. La Tabella 4.2 ne mostra alcuni.

Id	Nome titoli	Numero validazioni
12315	Ferry11-autocarri+35q.	864
13003	Cav -Trep + Actv 24H	9479
14126	Extra tratta 6	1718
14122	Extra tratta 2	35062
10040	Bus+People mover online	488
11136	75'-Tpl 6,30-ComVe1,20	5062
11552	72H RVenice+aerop.CS online	4704
11302	24ore online no aerobus	27181
11551	72 ore R.Venice online	36675
12104	Bigl.Mestre/Lido 75' a bordo	12796

Tabella 4.2: Esempio di titoli di viaggio

Attraverso l'analisi del nome sono stati facilmente individuati i titoli di viaggio di interesse, cioè quelli con una validità da 1 a 7 giorni, e esclusi quelli di durata inferiore alle 24 ore. La Tabella 4.3 mostra la loro numerosità. Inoltre molti titoli

Durata titolo	Numero di titoli
Meno di un giorno	116
1 giorno	21
2 giorni	6
3 giorni	14
7 giorni	6

Tabella 4.3: Distribuzione dei titoli rispetto alla durata

Identificativo	Nome
11251	24ore online aerobus AR
11236	24hAerAR-Tpl26,9-CVe5,1

Tabella 4.4: Esempio di titolo con versione online e non

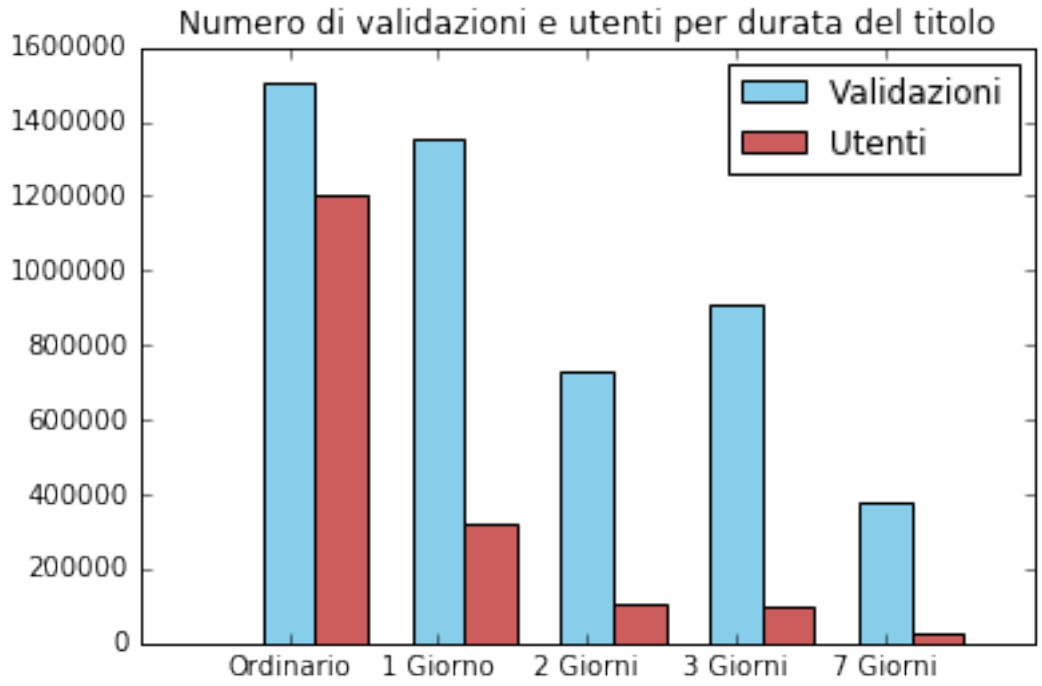

Figura 4.3: Distribuzione delle validazioni e degli utenti rispetto alla durata del titolo di viaggio

di viaggio si presentano doppi, con l'indicazione di un acquisto online oppure no, come mostra la Tabella 4.4. In totale i titoli turistici online sono 19.

Il grafico in Figura 4.3 mostra come le validazioni e gli utenti sono distribuiti rispetto alle durate dei titoli. Come ci si può aspettare il rapporto tra validazioni e utenti aumenta all'aumentare della durata del titolo. Questo risultato previsto è importante in quanto significa che i turisti hanno effettuato abbastanza validazioni per definire un loro tracciato.

Raggruppare i titoli solo per la durata rischia di eliminare alcune differenze importanti tra gli utenti, per questo motivo gli originali 47 titoli di viaggio di interesse sono stati raggruppati in 21 titoli distinti, principalmente aggregando i

titoli con la loro controparte online. Il risultato è descritto nella Tabella 4.5.

Id	Titolo	Durata	Titoli originali
0	Cav -Trep + Actv 24H	1	[13003]
1	Jesolo + Actv 24H	1	[13006]
2	T.Fusina Ve+ACTV 24 ore	1	[16103]
3	24H metropolitano ORD	1	[10033, 10023]
4	24H metropolitano ORD+1	1	[10024, 10034]
5	24H metropolitano ORD+2	1	[10035, 10025]
6	24H metropolitano RES.	1	[10020, 10021, 10022]
7	Ville Venete solo linea 53 24H	1	[14145, 14150, 14140]
8	24ore aerobus AR	1	[11251, 11236]
9	24ore aerobus CS	1	[11341, 11226]
10	24ore no aerobus	1	[11302, 11105]
11	48ore no aerobus	2	[11304, 11107]
12	48ore aerobus CS	2	[11343, 11228]
13	48ore aerobus AR	2	[11253, 11238]
14	Cav - Trep + Actv 72H	3	[13004]
15	T.Fusina Ve+ACTV 72 ore	3	[16104]
16	72ore no aerobus	3	[11305, 11109, 11108, 11551]
17	72ore aerobus CS	3	[11344, 11229, 11230, 11552]
18	72ore aerobus AR	3	[11254, 11240, 11553, 11239]
19	7 days no aerobus	7	[11306, 11149]
20	7 days aerobus AR	7	[11255, 11241]
21	7 days aerobus CS	7	[11345, 11231]

Tabella 4.5: Elenco dei titoli turistici aggregati

4.4 Pulizia e filtraggio dei dati

Individuati i titoli di viaggio significativi, l'obiettivo è diventato filtrare e correggere i dati per renderli coerenti rispetto ai vincoli imposti.

Il primo livello di filtraggio è stato rimuovere gli esatti duplicati. Il dataset di partenza infatti presenta alcune validazioni perfettamente identiche, dove il seriale del biglietto, fermata validata, data, ora e minuti sono uguali. Questa pulizia ha eliminato 205470 validazioni, passando da 4876778 a 4671308 validazioni totali.

La rimozione di questo numero significativo di validazioni è giustificata da due principali fattori. Il primo è l'utilizzo sbagliato da parte degli utenti, che convalidano il biglietto nella stessa stazione a differenza di pochi secondi. Il secondo è il dataset fornito da ACTV, in quanto contiene i dati raccolti senza nessuna pulizia e filtraggio.

Eliminate queste validazioni non corrette, il passo successivo è stato estrarre le validazioni dei titoli di interesse, cioè appartenenti a uno dei 21 titoli di viaggio individuati. Le validazioni per i titoli turistici, cioè con durata minima di un giorno, sono 3183618.

Successivamente il filtraggio è stato effettuato a un livello più legato al significato dei dati. È stato verificato se la distanza temporale tra le validazioni di uno stesso utente rispettano la tipologia del titolo di viaggio. In particolare tutte le validazioni oltre la durata massima del titolo sono state rimosse. La Tabella 4.6 ne mostra un esempio, l'ultima validazione non è consentita dal titolo di viaggio perché avviene dopo oltre 24 ore dalla prima validazione, quindi viene rimossa. Questo tipo di filtraggio ha rimosso 7888 validazioni.

Data	Fermata
22 settembre 10:22	5001 Lido
22 settembre 11:30	5038 Rialto
22 settembre 15:02	5063 Murano
22 settembre 16:30	5068 Burano
22 settembre 18:30	5009 San Zaccaria
23 settembre 10:53	5001 Lido

Tabella 4.6: Esempio di vincolo temporale non rispettato per un utente con titolo da 24 ore

L'ultimo filtraggio risolve il problema opposto a quello sopracitato, cioè la rimozione di validazioni troppo vicine tra loro. Queste sono principalmente causate dagli utenti che sbagliano a validare o validano più volte. Il tipico errore osservato è la validazione consecutiva di fermate nello stesso imbarcadero come mostrato nell'esempio in Tabella 4.7. La correzione è avvenuta rimuovendo tutte le validazioni che sono seguite da un'ulteriore validazione entro 5 minuti. Questo filtraggio ha rimosso 117929 validazioni. Questo numero è particolarmente grande e un lasso di tempo maggiore avrebbe rimosso un numero molto più elevato di validazioni. È stato scelto 5 minuti come tempo di riferimento perché in alcune zone è possibile fare dei cambi con questa frequenza, come ad esempio con il Tram di Mestre o il passaggio tra l'isola del cimitero di San Michele e l'isola di Murano.

Data	Fermata
22 settembre 11:30	5038 Rialto
22 settembre 15:02	5063 Murano
22 settembre 18:22	5009 San Zaccaria
22 settembre 18:23	5009 San Zaccaria
22 settembre 18:25	5009 San Zaccaria

Tabella 4.7: Esempio di validazioni duplicate di un utente

4.5 Salvataggio dei dati

I dati precedentemente elaborati sono stati salvati in modo permanente in 2 file json distinti:

- titoli.json, un dizionario che contiene i 21 titoli identificati nella tabella 4.5, la chiave del dizionario è l'id, mentre il valore è una tupla che contiene in

ordine: nome del titolo di viaggio, durata in giorni, tupla degli id dei titoli originari che contiene;

- validazioni.json, il dizionario che contiene tutte le validazioni dei titoli turistici. È strutturato nel seguente modo: la chiave è l'identificativo di un titolo di viaggio, il valore è la lista degli utenti di quel titolo, ogni utente è una lista ordinata di validazioni, ogni validazione è una coppia, il cui primo elemento è la data, mentre il secondo è l'id della fermata.

File	Dimensione file
titoli.json	1.2 KB
validazioni.json	83 MB

Tabella 4.8: File delle validazioni

4.6 Presentazione dei dati filtrati

Le operazioni precedenti hanno avuto come risultato il ridurre le validazioni totali a 3057801 e il numero di utenti a 551962. Sono ripartiti tra le varie durate dei titoli di viaggio come mostrato nel grafico in Figura 4.4.

Le validazioni all'interno di questo periodo si distribuiscono come indicato nel grafico in Figura 4.5, il grafico mette in risalto come l'andamento è fondamentalmente costante, con una diminuzione nelle ultime due settimane analizzate. Questa forte riduzione è spiegata dal maltempo che ha colpito il Veneto alla fine di ottobre 2018 [8] e l'acqua alta eccezionale che ha colpito la città [9].

Un altro punto di riferimento importante è la mappa di calore delle validazioni che mostra come ciascuna fermata è frequentata. La mappa in Figura 4.6 e la Tabella 4.9 mostrano come il 79% di tutte le validazione è concentrata in appena 15 fermate.

La Tabella 4.9 mostra un risultato atteso, cioè come alcune fermate svolgono un ruolo più importante rispetto ad altre, in particolare:

- terra, p.le Roma e ferrovia, sono il punto d'accesso alla città per chi proviene dalla terraferma;
- Murano e Burano, in quanto le principali isole turistiche;
- S. Zaccaria, in quanto una delle fermate più connessa alle linee di trasporto e punto strategico per il collegamento con le isole e punta sabbioni;
- Rialto, in quanto snodo per il cuore della città.

La mappa e la tabella inoltre mostrano come l'aeroporto, il servizio automobilistico del Lido e i collegamenti con Chioggia siano estremamente rari per i turisti.

Nei paragrafi seguenti vengono elencate le statistiche più dettagliate e complete per le varie durate di titolo di viaggio.

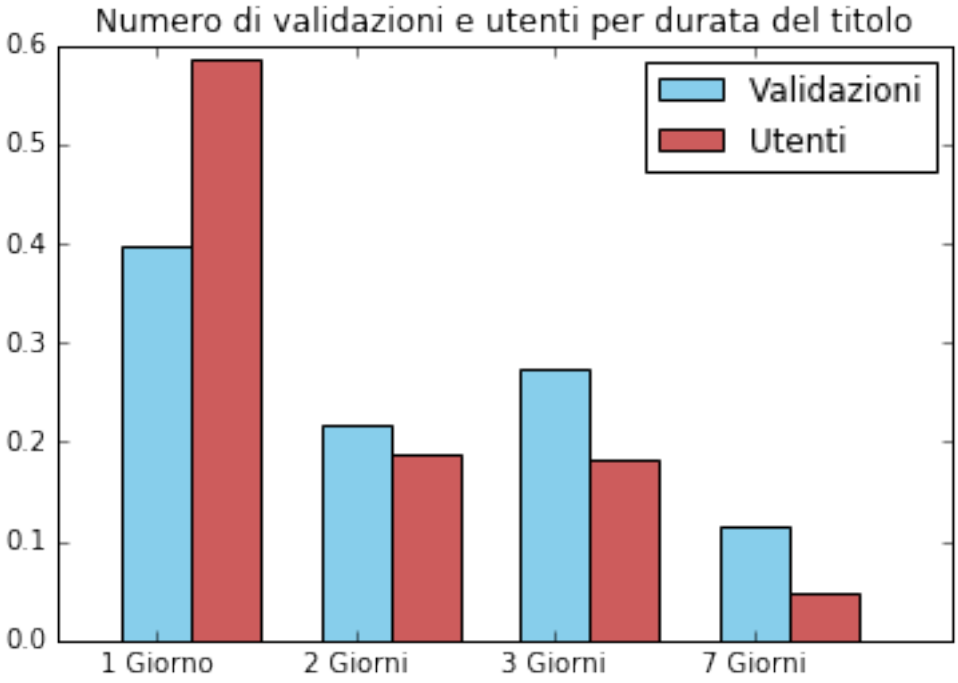

Figura 4.4: Distribuzione rispetto alla durata del dati filtrati

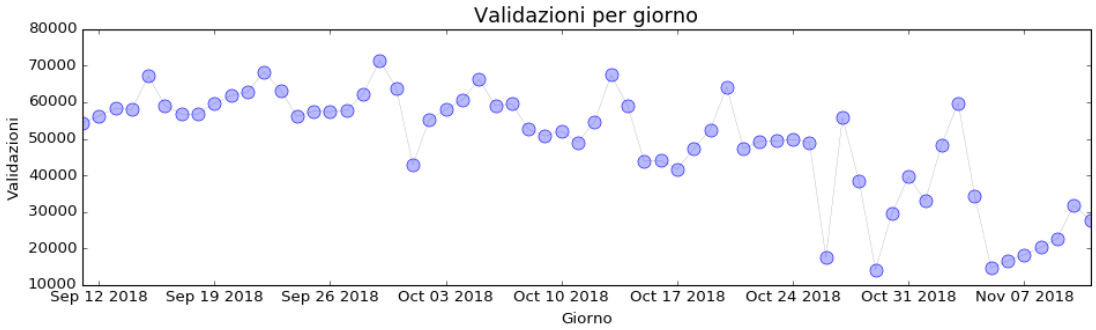

Figura 4.5: Distribuzione delle validazioni nel periodo di tempo studiato

4.6.1 Statistiche titolo da 1 giorno

Gli utenti e le validazioni di titoli da 1 giorno, come mostra il grafico in Figura 4.7, assumono lo stesso andamento delle validazioni totali.

Come si può evincere dalla Figura 4.8 e dalla Tabella 4.10, ogni utente in media fa 3.7 fermate, gli utenti con più di 10 fermate non sono completamente attendibili in quanto descrivono percorsi poco affidabili, come mostra l'utente riportato in Tabella 4.11. I criteri automatici descritti in precedenza non hanno corretto queste validazioni. In ogni caso il loro numero è estremamente basso e non causa problemi negli algoritmi usati.

Le validazione sono ripartite tra i servizi nel seguente modo:

Validazioni servizio automobilistico	105172	9%
Validazioni servizio di navigazione	1106272	91%

Figura 4.6: Mappa di calore delle validazioni, la massima intensità è posta pari al 50% della fermata maggiormente validata, circa il 5% delle validazioni totali

Id	Fermata	Percentuale validazioni
5009	s. zaccaria	12.00%
5063	murano	9.54%
5032	ferrovia	8.10%
5038	rialto	7.60%
5068	burano	7.10%
-1	TERRA	6.98%
5501	p.le roma	6.54%
15060	f.te nove	4.94%
5001	lido	4.90%
5053	s. marco	3.03%
-2	Piazzale Roma	1.92%
5021	s. giorgio	1.67%
5043	s. toma	1.66%
5035	s. marcuola-casino	1.61%
5049	zattere	1.59%

Tabella 4.9: Elenco delle 15 fermate maggiormente validate e della loro percentuale

Inoltre possiamo distinguere i turisti in base all'utilizzo del biglietto in due giorni distinti, infatti i biglietti a tempo hanno validità dalla prima timbratura fino alla loro durata massima e questa può essere a cavallo tra più giorni.

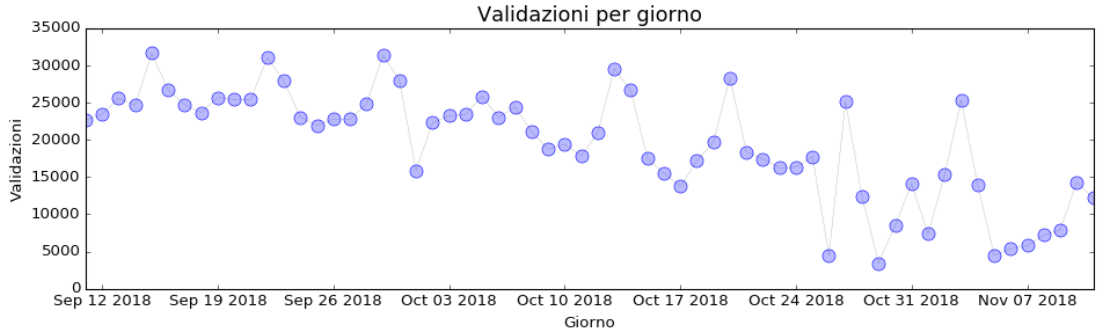

Figura 4.7: Distribuzione delle validazioni di titoli da 24h nel periodo di tempo studiato

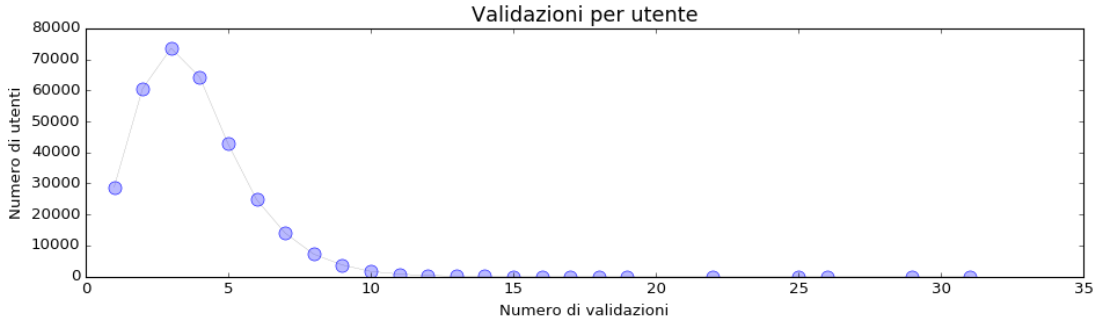

Figura 4.8: Distribuzione degli utenti di titoli di 24h rispetto al numero di validazioni

Numero di validazioni	Numero di utenti	Percentuale
1	28796	9%
2	60601	19%
3	73629	23%
4	64333	20%
5	42734	13%
6	24767	8%
7	14000	4%
8	7206	2%
9	3718	1%
10	1638	1%

Tabella 4.10: Distribuzione degli utenti di titoli di 24h rispetto al numero di validazioni in termini assoluti e relativi

Validazioni nello stesso giorno	240661	75%
Validazioni nel giorno successivo	82071	25 %

Il grafico in Figura 4.9 mostra la distribuzione oraria e risulterà importante per l'analisi dei pattern, infatti come sarà spiegato nel prossimo capitolo, gli utenti saranno prima divisi per traiettoria, successivamente ogni traiettoria sarà studiata rispetto al tempo. L'analisi della distribuzione oraria e di una sua corretta divisione

Data	Fermata
2018-09-12 00:10:00	p.le roma
2018-09-12 00:16:00	p.le roma
2018-09-12 09:13:00	ferrovia
2018-09-12 09:31:00	ferrovia
2018-09-12 10:19:00	murano
2018-09-12 10:43:00	murano
2018-09-12 13:48:00	burano
2018-09-12 17:25:00	murano
2018-09-12 19:07:00	ferrovia
2018-09-12 20:31:00	s. toma
2018-09-12 20:57:00	s. toma
2018-09-12 21:04:00	s. toma
2018-09-12 21:11:00	s. toma
2018-09-12 22:50:00	s. zaccaria
2018-09-12 23:21:00	ferrovia
2018-09-12 23:45:00	ferrovia

Tabella 4.11: Utente non correttamente pulito dai criteri automatici

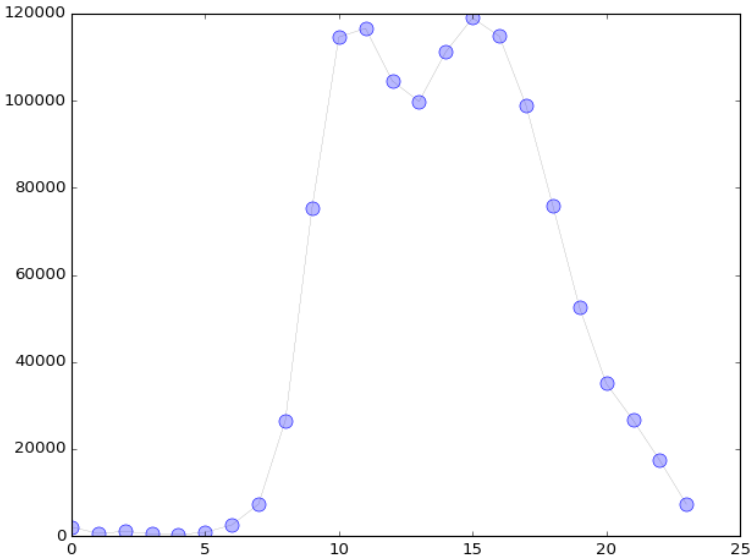

Figura 4.9: Distribuzione oraria delle validazioni dei titoli da 24h

in fasce orarie diventa quindi necessaria per poter identificare i pattern dei turisti.

4.6.2 Statistiche titolo da 2 giorni

Come per i titoli con durata pari a un giorno vengono riproposti alcuni grafici, in particolare il grafico in Figura 4.10 mostra un andamento simile a quello globale, il grafico in Figura 4.11 e la Tabella 4.12 mostrano come la moda degli utenti è di effettuare tra le 4 e 7 validazioni, e per ultima la Tabella 4.13 mostra come queste siano equamente distribuite tra due giorni.

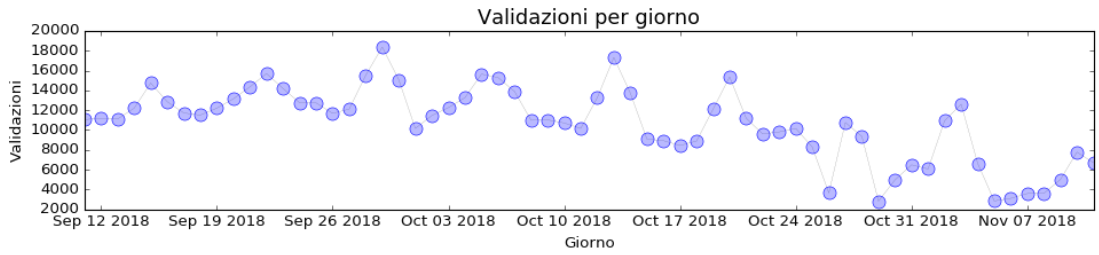

Figura 4.10: Distribuzione delle validazioni di titoli da 48h nel periodo di tempo studiato

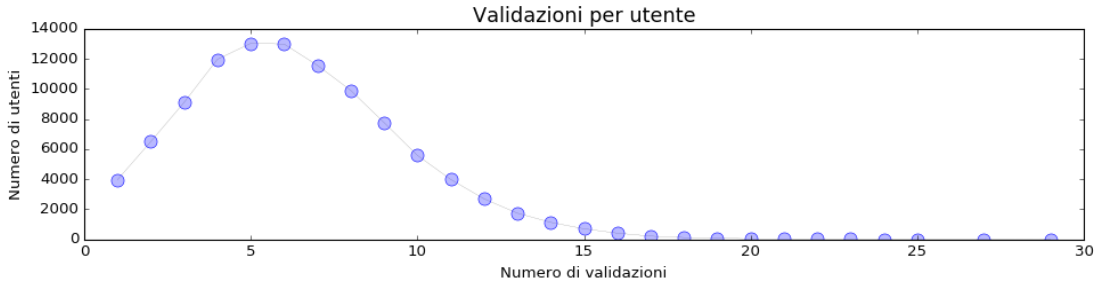

Figura 4.11: Distribuzione degli utenti di titoli da 48h rispetto al numero di validazioni

Numero di validazioni	Numero di utenti	Percentuale
1	3956	4%
2	6493	6%
3	9104	9%
4	11976	12%
5	13037	13%
6	12996	13%
7	11578	11%
8	9913	10%
9	7769	8%
10	5588	5%
11	3985	4%
12	2693	3%
13	1738	2%
14	1128	1%
15	698	1%

Tabella 4.12: Distribuzione degli utenti di titoli da 48h rispetto al numero di validazioni in termini assoluti e relativi

Giorno	Validazioni	Percentuale	Media utente
0	289554	0.44	2.8
1	309028	0.47	2.98
2	61471	0.09	0.59

Tabella 4.13: Distribuzione delle validazioni di titoli da 48h rispetto al giorno di utilizzo

4.6.3 Statistiche titolo da 3 giorni

Come per gli altri titoli sono riproposti i grafici e le tabelle più significative, in particolare il grafico in Figura 4.12 mostra lo stesso andamento degli altri titoli, il grafico in Figura 4.13 e la Tabella 4.14 mostrano come la moda degli utenti è di effettuare tra le 6 e 9 validazioni, e infine la Tabella 4.15 mostra come queste siano equamente distribuite tra i primi 3 giorni e che l'ultimo sia fondamentalmente nullo rispetto agli altri, come per i precedenti titoli.

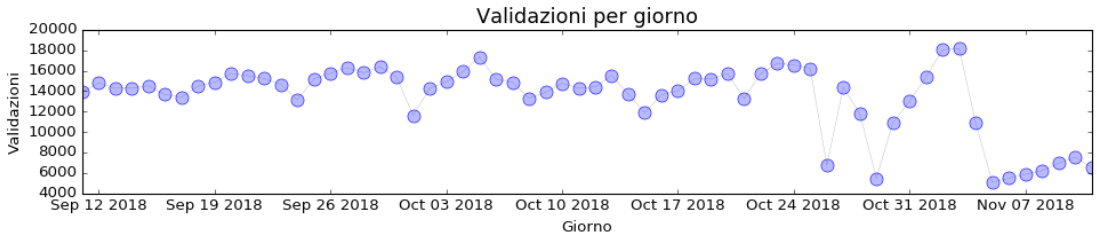

Figura 4.12: Distribuzione delle validazioni di titoli da 72h nel periodo di tempo studiato

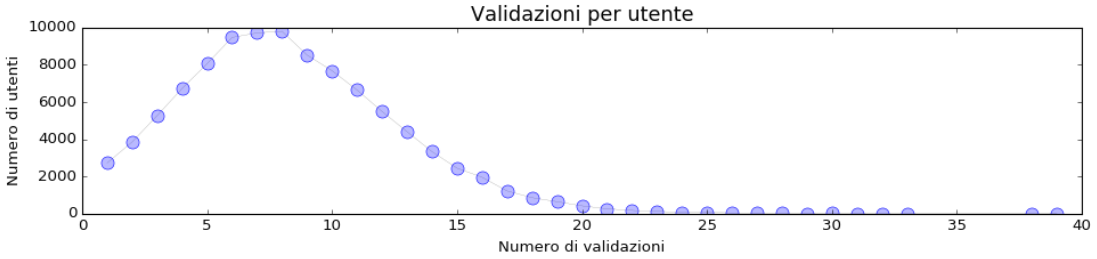

Figura 4.13: Distribuzione degli utenti di titoli da 72h rispetto al numero di validazioni

Numero di validazioni	Numero di utenti	Percentuale
1	2715	3%
2	3845	4%
3	5263	5%
4	6729	7%
5	8062	8%
6	9481	9%
7	9722	10%
8	9813	10%
9	8533	9%
10	7689	8%
11	6650	7%
12	5495	5%
13	4405	4%
14	3331	3%
15	2440	2%
16	1953	2%
17	1213	1%
18	852	1%
19	633	1%
20	410	0.5%

Tabella 4.14: Distribuzione degli utenti di titoli da 72h rispetto al numero di validazioni in termini assoluti e relativi

Giorno	Validazioni	Percentuale	Media utente
0	250600	0.3	2.51
1	288993	0.35	2.89
2	237726	0.29	2.38
3	52679	0.06	0.53

Tabella 4.15: Distribuzione delle validazioni di titoli da 72h rispetto al giorno di utilizzo

4.6.4 Statistiche titolo da 7 giorno

Come per gli altri titoli sono riproposti i grafici e le tabelle più significative, in particolare il grafico in Figura 4.14 mostra lo stesso andamento degli altri titoli nell'arco dei due mesi studiati, il grafico in Figura 4.15 e la Tabella 4.16 mostrano che gli utenti più frequentemente timbrano il biglietto tra le 10 e le 15 volte, e infine la tabella 4.17 mostra come queste siano distribuite tra i possibili 8 giorni disponibili. Interessante è notare come i primi 5 giorni abbiano più o meno le stesse percentuali, con un lieve calo nell'ultimo, mentre dal sesto in poi ci sia un crollo e mediamente meno di una validazione per utente.

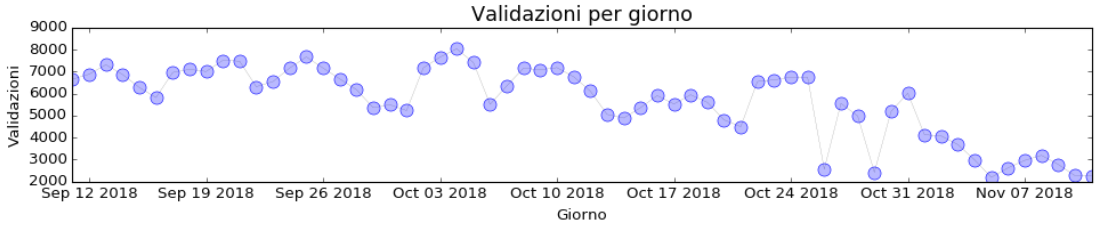

Figura 4.14: Distribuzione delle validazioni di titoli da 7g nel periodo di tempo studiato

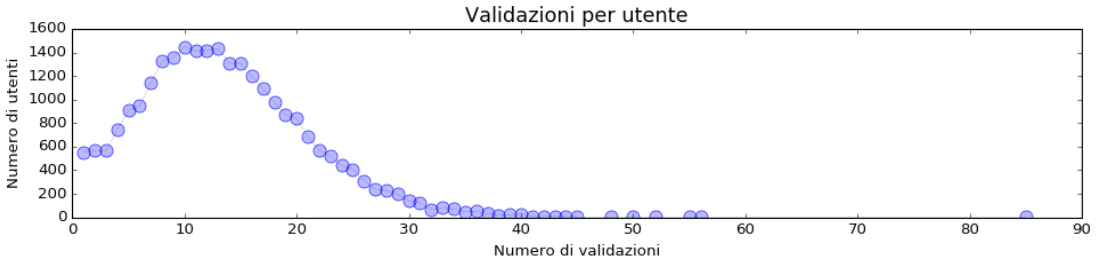

Figura 4.15: Distribuzione degli utenti di titoli da 7g rispetto al numero di validazioni

Numero di validazioni	Numero di utenti	Percentuale
1	554	2%
2	566	2%
3	565	2%
4	746	3%
5	907	4%
6	945	4%
7	1147	4%
8	1325	5%
9	1358	5%
10	1448	6%
11	1411	5%
12	1419	6%
13	1430	6%
14	1310	5%
15	1308	5%
16	1197	5%
17	1090	4%
18	974	4%
19	869	3%
20	840	3%
21	683	3%
22	569	2%
23	520	2%
24	442	2%
25	402	2%
26	311	1%
27	239	1%
28	227	1%
29	198	1%
30	139	1%

Tabella 4.16: Distribuzione degli utenti di titoli da 7g rispetto al numero di validazioni in termini assoluti e relativi

Giorno	Validazioni	Percentuale	Media utente
0	57229	0.16	2.22
1	63993	0.18	2.49
2	65075	0.19	2.53
3	62308	0.18	2.42
4	48503	0.14	1.89
5	31151	0.09	1.21
6	17739	0.05	0.69
7	4165	0.01	0.16

Tabella 4.17: Distribuzione delle validazioni di titoli da 7g rispetto al giorno di utilizzo

Capitolo 5

Analisi dei pattern

Questo capitolo descrive l'analisi che è stata effettuata per determinare pattern sul movimento dei turisti. Le prime sezioni elencano gli algoritmi di data mining usati, le misure di qualità, le varie misure di distanza provate e le implementazioni e ottimizzazioni adottate. L'ultima sezione entra nel pieno del merito dell'analisi e descrive come sono stati usati gli strumenti precedentemente presentati e i risultati ottenuti.

5.1 Algoritmi di Clustering

Il clustering [10] è un processo di partizionamento di un insieme di osservazioni in sottoinsiemi. Ognuno di questi sottoinsiemi si chiama cluster ed è creato in modo tale che ogni elemento all'interno sia molto simile agli altri, ma al tempo stesso molto diverso dagli elementi degli altri cluster. Differenti algoritmi di cluster possono generare diversi cluster a partire dallo stesso set di dati. Il clustering rientra tra gli strumenti di unsupervised learning in quanto non si conosce a priori la classe di ogni elemento e, per questo motivo, il clustering è utile in quanto può portare alla scoperta di gruppi non conosciuti all'interno dei dati. Gli aspetti e i requisiti che si richiedono ad un algoritmo di clustering sono svariati. Di seguito sono elencate le caratteristiche più adatte per l'analisi svolta:

- scalabilità, molti algoritmi lavorano bene solo su piccole quantità di dati che non superano le poche centinaia di osservazioni. Tuttavia per questa analisi bisogna analizzare gruppi che variano dai 30 mila ai 300 mila utenti, per questo motivo gli algoritmi ricercati richiedono scalabilità;

- capacità di gestire diversi tipi di attributi. Anche se molti algoritmi sono studiati per lavorare con intervalli numerici, questo studio richiede la capacità di lavorare con dei dati molto particolari, ovvero sequenze spazio-temporali;

- capacità di trovare cluster di forma arbitraria. Molti algoritmi di clustering utilizzano come distanza quella Euclidea o di Manhattan, che tendono a creare cluster di dimensione sferica, mentre un cluster dovrebbe poter assumere qualsiasi forma;

- capacità di gestire dati rumorosi. I dataset reali contengono spesso outlier, valori mancanti o dati errati e il dataset delle validazioni non fa eccezione. Algoritmi troppo sensibili a questi dati rumorosi possono portare a risultati di bassa qualità, per questo motivo c'è bisogno di un algoritmo robusto rispetto al rumore e agli errori.

5.1.1 K-means

K-means è un algoritmo di clustering basato sulla partizione, in quanto suddivide gli elementi in gruppi mutuamente esclusivi, e sulla tecnica del centroide perché rappresenta ogni cluster con il suo elemento centrale. Il centroide può essere definito in vari modi e generalmente corrisponde alla media degli elementi che appartengono al cluster. Per questo motivo non necessariamente è un elemento reale, ma potrebbe essere fittizio. L'algoritmo richiedere che il numero K di cluster sia fornito a priori. Il livello di qualità dei cluster ottenuti è misurato dalla somma dei quadrati delle distanze di tutti gli elementi rispetto al proprio centroide, minore è questa sommatoria, maggiore sarà la qualità del cluster complessivo. È dimostrato che una ricerca esaustiva del miglior partizionamento è un problema NP-hard anche sotto le migliori condizioni. Per superare il costo proibitivo dell'algoritmo, la soluzione maggiormente utilizzata è un approccio greedy.

L'algoritmo greedy generale è rappresentato nello schema seguente.

Algorithm 1: Algoritmo greedy k-means

```
Data: D: dataset, K: numero di cluster
scegli k oggetti da D come centri iniziali dei cluster;
while ci sono cambiamenti do
    (re)assegna ogni elemento al cluster il cui centro è più vicino;
    aggiorna i centri del cluster, ricalcolando la media;
end
```

Questo algoritmo non garantisce la convergenza al migliore assoluto, per cui è buona norma ripetere l'algoritmo più volte con differenti centri iniziali. La complessità è lineare rispetto al numero di cluster e di elementi del dataset e per questo motivo l'algoritmo è generalmente scalabile. I vantaggi principali sono la complessità e la facilità di interpretazione. I limiti di questo algoritmo sono: la forma generalmente sferica che assumono i cluster, la sensibilità agli outlier, i rappresentanti che non necessariamente appartengono agli elementi iniziali, la necessità di specificare il numero di cluster a priori e la misura di distanza che spesso è troppo generale.

5.1.2 DBscan

Molti algoritmi di clustering sono studiati per creare cluster di dimensione sferica e trovano molta difficoltà nel generare cluster di forma arbitraria. Gli algoritmi di clustering basati sulla densità (density-based) cercano di superare questo limite definendo regioni dense nello spazio dei dati. La densità di un oggetto può essere

definita come il numero di elementi a lui vicini. DBscan (Density-Based Spatial Clustering of application with noise) definisce gli oggetti densi come core object, successivamente connette i core object e i loro vicini per formare regioni dense che corrispondono a cluster. Un core object è definito da due parametri:

- k, ovvero il numero minimo di elementi nel vicinato;
- ε, ovvero il raggio entro cui i vicini devono essere presenti.

Algorithm 2: Algoritmo DBscan

```
Data: D: dataset, k: il numero minimo di elementi nel vicinato, ε: il raggio
      entro cui i vicini devono essere presenti
while ci sono core object non visitati do
    estrai un core object non visitato e crea un cluster che contiene solo
     questo elemento;
    while il cluster può espandersi do
        aggiungi tutti i vicini dei core object presenti;
    end
end
```

L'algoritmo per come è strutturato, automaticamente esclude gli outlier in quanto, non essendo vicini a nessun core object, non saranno mai aggiunti a nessun cluster. I vantaggi sono la creazione di cluster di forma qualsiasi e l'esclusione automatica degli outlier, gli svantaggi sono legati alla scelta dei due parametri. Una scelta non accorta rischia di "avvicinare" troppo tutti i core object: il caso limite è la costruzione di un cluster che raccoglie tutti, o quasi, gli elementi.

5.1.3 Clustering gerarchico

Il clustering gerarchico raggruppa gli elementi del dataset a diversi livelli creando una gerarchia, assimilabile ad un albero. L'approccio gerarchico può essere agglomerativo o divisivo. La tecnica divisiva utilizza una strategia top-down: inizialmente ogni elemento appartiene alla stesso cluster e ricorsivamente divide il cluster in cluster più piccoli. Questa analisi utilizza algoritmi agglomerativi che utilizzano un approccio bottom-up: tipicamente inizia creando per ogni elemento del dataset il proprio cluster e iterativamente unisce i cluster per formarne uno più grande. L'algoritmo termina quando tutti gli elementi appartengono allo stesso cluster oppure se determinate condizioni di terminazione sono raggiunte. L'algoritmo sceglie di unire due cluster in base alla loro distanza, questa può essere espressa in diversi modi:

- distanza minima, che corrisponde alla distanza minima possibile tra un elemento del primo cluster e un elemento del secondo;
- distanza massima, che corrisponde alla distanza massima possibile tra un elemento del primo cluster e un elemento del secondo;
- distanza media, che calcola la distanza tra gli elementi centrali dei due cluster;

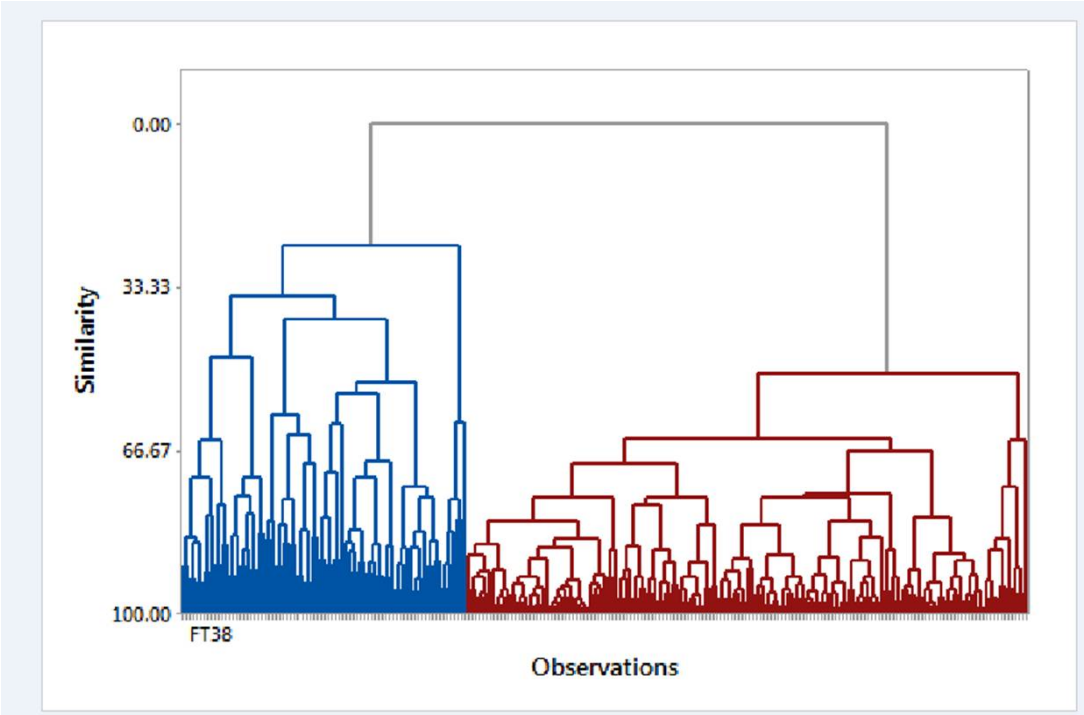

Figura 5.1: Esempio di dendrogramma

- media delle distanze, che calcola tutte le possibili distanze tra gli elementi del primo cluster e del secondo e ne effettua la media.

Il dendrogramma è una struttura ad albero comunemente usata per rappresentare il processo di clustering gerarchico. Mostra passo per passo come il clustering è effettuato. La Figura 5.1 ne mostra un esempio.

5.2 Coefficiente di silhouette

Il coefficiente di silhouette è una misura intrinseca per valutare la qualità di un cluster che basa la sua misura su quanto i cluster sono compatti e quanto separati tra loro. Per ogni oggetto o del dataset appartenente al cluster C_i calcola due quantità:

- $a(o)$, ovvero la distanza media di o dagli oggetti dello stesso cluster

$$a(o) = \frac{\sum_{o' \in C_i} dist(o, o')}{|C_i|}$$

- $b(o)$, ovvero la distanza minima media tra o e gli oggetti dei cluster a cui non appartiene

$$b(o) = \min_{i \leq j \leq k, j \neq i} \frac{\sum_{o' \in C_j} dist(o, o')}{|C_i|}$$

Il coefficiente di silhouette di o è definito come:

$$s(o) = \frac{b(o) - a(o)}{max\{a(o), b(o)\}}$$

Il valore che può assumere è compreso tra -1 e 1. Più il valore si avvicina all'uno, più il cluster è compatto e o è distante dagli altri cluster. Il coefficiente dell'intero dataset è calcolato come la media dei coefficienti di tutti gli oggetti appartenenti al dataset.

5.3 Misure di distanza

Un aspetto fondamentale degli algoritmi di clustering è l'utilizzo della misura di similarità o distanza. Una misura non significativa può compromettere completamente la qualità del risultato. Ricordiamo il dominio e le proprietà di cui deve godere una misura di distanza definita su un insieme X:

$$d : X \times X \longrightarrow \mathbb{R}$$

$$d(x, y) \geq 0$$

$$d(x, y) = 0 \iff x = y$$

$$d(x, y) = d(y, x)\ (simmetria)$$

$$d(x, y) \leq d(x, z) + d(z, y)\ (disuguaglianza\ triangolare)$$

Di seguito saranno illustrate in dettaglio le varie misure di distanza o similarità utilizzate e testate durante l'analisi.

5.3.1 Jaccard

La similarità di Jaccard è un indice definito come il rapporto tra la dimensione dell'intersezione e la dimensione dell'unione di due insiemi.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

L'indice di similarità Jaccard è compreso tra 0 e 1, la distanza non è altro che:

$$Dj(A, B) = 1 - J(A, B)$$

Questa misura è stata usata per definire la distanza tra due utenti senza considerare il fattore tempo, per cui ogni utente è rappresentato da una sequenza ordinata di fermate. L'applicazione della Jaccard sugli insiemi composti dalle fermate validate da un utente presenta un limite notevole nel fatto che l'ordine perde il suo significato.

$$A = (f1, f2)$$

$$B = (f2, f1)$$

$$J(A, B) = 1$$

In questo esempio due utenti A e B sono passati per le stesse fermate $f1$ e $f2$, ma in ordine inverso, tuttavia l'indice di Jaccard è pari a 1, per cui A e B sono simili al massimo.

Si è cercato di risolvere questo problema rappresentando gli utenti diversamente. Ogni utente è rappresentato come insieme di coppie di fermate. La jaccard è valutata su queste coppie. Ogni coppia di fermate mantiene l'ordine di validazione, essendo costruite ponendo come primo elemento una fermata e come secondo tutte le fermate validate dopo la prima. L'esempio seguente mostra meglio la costruzione.

$$A = (f1, f2) => [(f1, f2)]$$
$$B = (f2, f1) => [(f2, f1)]$$
$$J(A, B) = 0$$

$$C = (f1, f2, f3, f4) => [(f1, f2), (f1, f3), (f1, f4), (f2, f3), (f2, f4), (f3, f4)]$$
$$D = (f1, f2, f4, f3) => [(f1, f2), (f1, f4), (f1, f3), (f2, f4), (f2, f3), (f4, f3)]$$
$$J(C, D) = \frac{5}{7}$$

Il risultato è una misura che mantiene un certo livello di informazione dell'ordine delle fermate. In generale potrebbe essere applicata anche con n-uple di lunghezza arbitraria per ricercare una severità maggiore nella misura.

Nello studio di questi tragitti è possibile che alcune fermate possano essere duplicate, per cui è necessario lavorare con i multi-insiemi piuttosto che con gli insiemi normali. Esiste una variante della similarità di Jaccard che è definita sui multi-insiemi, la differenza è nel range di valori che può assumere l'indice, che varia da 0 a 0.5.

5.3.2 Euclidea

La distanza euclidea è la misura della lunghezza del segmento passante tra due punti. Ogni punto è definito da n coordinate nello spazio, la distanza è:

$$D(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

con A_i e B_i le i-esime coordinate.

5.3.3 Edit

La distanza di edit, o Levenshtein, è definita generalmente sulle stringhe e conta il numero di operazioni elementari necessarie per trasformare una stringa A nella stringa B. Le operazioni elementari sono:

- inserimento di un carattere;

- sostituzione di un carattere;
- eliminazione di un carattere.

Ad esempio le parole *cane* e *pane* hanno distanza 1: l'unica operazione elementare necessaria da fare è la sostituzione del primo carattere.

Questa misura assume valore almeno pari alla differenza della lunghezza tra le due stringhe e al massimo valore pari alla lunghezza della stringa maggiore.

Tra due stringhe R e S con numero di caratteri rispettivamente n e m è calcolata nel seguente modo:

$$EDR(R,S) = \begin{cases} n & se\ m = 0 \\ m & se\ n = 0 \\ min \begin{cases} EDR(Rest(R), Rest(S)) + c \\ EDR(Rest(R), S) + 1 \\ EDR(R, Rest(S)) + 1 \end{cases} & altrimenti \end{cases}$$

$$c = \begin{cases} 0 & se\ r_1 = s_1 \\ 1 & altrimenti \end{cases}$$

$$Rest(R) = R_2 \dots R_n$$

Questa distanza è facilmente applicabile a una generica sequenza di elementi. Nel nostro caso di studio la sequenza di elementi è pari alla sequenza ordinata di fermate validate, senza considerare il fattore tempo.

$$A = (f1, f2)$$

$$B = (f2, f1)$$

$$Edit(A,B) = 2$$

$$C = (f1, f2, f3)$$

$$D = (f1, f2, f4, f5)$$

$$Edit(C,D) = 2$$

Come mostra l'esempio, la distanza di edit è da valutare in funzione della lunghezza delle stringhe, rapportandola alla lunghezza maggiore delle due sequenze.

$$Edit(A,B) = \frac{2}{2}$$

$$Edit(C,D) = \frac{2}{4}$$

Come si evince dall'analisi prodotta in questa tesi, la distanza di edit applicata a questo contesto soffre molto della differenza della lunghezza delle sequenze.

5.3.4 Sottosequenza più lunga

LCSS (longest common subsequence) è una misura di similarità che conta la lunghezza della sottosequenza più lunga contenuta tra le due sequenze. Sono state provate due diverse versioni:

- sottosequenza esatta;
- sottosequenza con buchi.

La prima versione (nota come substring) richiede che elementi consecutivi nella sottosequenza siano consecutivi anche nelle sequenze originali. Al contrario la sottosequenza con buchi (nota solitamente come subsequence) non deve rispettare questa proprietà. Il problema della subsequence può essere visto come il problema della distanza di edit con solo due operazioni elementari consentite: inserimenti e cancellazioni.

Tra due sequenze R e S, con numero di elementi rispettivamente n e m, la subsequence è definita nel seguente modo:

$$LCSS_{\text{b}}(R,S) = \begin{cases} 0 & se\ m = 0\ o\ n = 0 \\ LCSS_{\text{b}}(Rest(R), Rest(S)) + 1 & se\ r_1 = s_1 \\ max \begin{cases} LCSS_{\text{b}}(R, Rest(S)) \\ LCSS_{\text{b}}(Rest(R), S) \end{cases} & altrimenti \end{cases}$$

$$Rest(R) = R_2 \dots R_n$$

Diversamente LCSS esatta non è facilmente definibile attraverso una formulazione matematica, ma può essere calcolata attraverso la programmazione dinamica. L'idea è cercare la lunghezza del più lungo suffisso per tutte le sottosequenze di entrambe le sequenze e salvarle in una tabella.

La Funzione 5.1 mostra l'esempio di codice C per calcolare la lunghezza della sottostringa comune tra due stringhe, riceve in input i puntatori alle stringhe e le loro lunghezze.

```
int LCSS_esatta(char *X, char *Y, int m, int n){
    int LCSuff[m+1][n+1];
    int result = 0;

    for (int i=0; i<=m; i++){
        for (int j=0; j<=n; j++){
            if (i == 0 || j == 0)
                LCSuff[i][j] = 0;
            else if (X[i-1] == Y[j-1]){
                LCSuff[i][j] = LCSuff[i-1][j-1] + 1;
                result = max(result, LCSuff[i][j]);
            }
            else
                LCSuff[i][j] = 0;
        }
    }
    return result;
}
```

Listing 5.1: LCSS esatta [11]

L'esempio mostra la differenza tra le due versioni.

$$A = (f1, f2, f3, f4, f5)$$
$$B = (f1, f2, f3, f5)$$
$$LCSS_{\mathrm{e}}(A, B) = 3$$
$$LCSS_{\mathrm{b}}(A, B) = 4$$

La sottosequenza esatta è una misura più severa rispetto alla sottosequenza con buchi, ma è molto suscettibile a eventuali elementi di rumore, come mostra il seguente esempio.

$$A = (f1, f2, f3, f4, f5, f6, f7, f8, f9, f10)$$
$$B = (f1, f2, f4, f5, f7, f8, f10)$$
$$LCSS_{\mathrm{e}}(A, B) = 2$$
$$LCSS_{\mathrm{b}}(A, B) = 7$$

Come per la misura di Edit, anche LCSS deve essere valutata in funzione della lunghezza delle sequenze e complementata per ottenere la distanza.

5.3.5 LCSS bilanciata

LCSS bilanciata è una misura di similarità ideata per questa tesi. Il suo obiettivo è riuscire a pesare quanto la sottosequenza comune è continua. È calcolata come il prodotto tra la LCSS con buchi e quella esatta.

$$LCSS_{\mathrm{bil}}(A, B) = LCSS_{\mathrm{b}}(A, B) * LCSS_{\mathrm{e}}(A, B)$$

L'esempio seguente mostra i limiti di LCSS con i buchi.

$$A = (f1, f2, f3, f4)$$
$$B = (f1, f4, f5, f6)$$
$$C = (f1, f2, f5, f6)$$
$$LCSS_{\mathrm{b}}(A, B) = 2$$
$$LCSS_{\mathrm{b}}(A, C) = 2$$

L'esempio mostra come B e C siano ugualmente simili ad A, tuttavia A e C condividono esattamente le prime due fermate, mentre A e B risultano ad una rapida osservazione maggiormente diverse. LCSS bilanciata prova a correggere questi casi bilanciando la sottosequenza più lunga con la più lunga esatta. L'esempio mostra come B e C non sono più ugualmente simili ad A.

$$LCSS_{\mathrm{bil}}(A, B) = 2 * 1 = 2$$
$$LCSS_{\mathrm{bil}}(A, C) = 2 * 2 = 4$$

Come per le altre misure di similarità, anche questa necessita di essere normalizzata, dividendola per il quadrato della lunghezza maggiore, e complementata per essere usata come distanza.

Questa misura tuttavia non è propriamente una misura di distanza in quanto non rispetta la disuguaglianza triangolare. L'algoritmo di clustering gerarchico non sfrutta questa proprietà durante l'esecuzione per cui questa misura è stata adottata ugualmente.

5.4 Implementazioni

In questa sezione saranno illustrate le varie tecniche usate per l'implementazione dei vari algoritmi di clustering con le differenti misure di similarità, i problemi riscontrati durante le implementazioni, le soluzioni e le ottimizzazioni trovate.

Gli algoritmi di clustering utilizzati sono disponibili nella libreria python di sklearn. Oltre i parametri specifici di ogni algoritmo, bisogna specificare la misura di distanza. Questo parametro può essere:

- metrica di default, cioè una delle misure standard di distanza, come ad esempio Euclidea, Coseno, Manhattan;
- una callable, cioè una funzione che riceve in input due elementi e ne calcola la distanza, generalmente usata per misure personalizzate;
- la stringa 'precomputed', cioè indica all'algoritmo che al posto del vettore degli elementi viene fornita una matrice quadrata simmetrica tale per cui la cella di coordinate (i, j) rappresenta la distanza tra l'elemento i e l'elemento j. Fornire la matrice diminuisce il tempo di computazione, ma aumenta lo spazio in memoria richiesto.

I primi cluster ricercati utilizzavano o una metrica di default o una callable. I tempi di computazione erano insostenibili per un numero di elementi sufficientemente grande.

Il tempo eccessivo ha obbligato a testare il clustering con la matrice delle distanze precomputata. I tempi di esecuzione degli algoritmi sono enormemente migliorati, ma il tempo di computazione della matrice si è rivelato non praticabile su un insieme modesto di utenti. Tuttavia una volta creata la matrice come struttura dati numpy, questa può essere salvata nella memoria permanente e ricaricata nel programma o nello script in tempi praticamente nulli.

Python, in quanto linguaggio interpretato e non compilato, è estremamente lento nel calcolo di matrici molto grandi rispetto a linguaggi compilati come il C. La soluzione migliore ricercata è stata quindi la seguente:

1. calcolo della matrice di distanza in un programma ottimizzato in C;
2. salvataggio della matrice in un file;
3. caricamento del file in uno script Python;

4. conversione del file in una matrice numpy;

5. salvataggio della matrice numpy con i metodi offerti dalla libreria.

Questa serie di operazioni consente di generare un file contenente la matrice di distanza che risulta essere estremamente veloce da caricare in un qualsiasi programma Python.

Il programma C è stato implementato ponendo molta attenzione alle ottimizzazioni e al multithreading, ottenendo ottimi risultati dal punto di vista del tempo di esecuzione: per una matrice di 70 mila utenti impiega circa 2 minuti per il calcolo e 2 minuti per il salvataggio su file.

Tuttavia la soluzione appena descritta soffre di un grosso problema di prestazioni: è apparso un collo di bottiglia che vincola la maggior parte del tempo di esecuzione a un singolo componente della catena descritta.

Il caricamento del file del punto 3 è lento ed è dovuto principalmente ai seguenti fattori:

- passaggio per la memoria secondaria. Come risaputo la velocità di un disco è estremamente inferiore di quella della RAM;

- dimensione del file. La matrice completa richiede, supponendo di usare numeri a 4 Byte, $4 * n^2$ Byte, con n il numero di utenti su cui si calcola. Questo significa che, con appena 10000 utenti, il file, supponendo sia binario, peserà come minimo 400 MegaByte, per 30000 utenti 3.6 GigaByte;

- il linguaggio Python non offre meccanismi veloci per una lettura personalizzata di un file, per cui operazioni di questo tipo avvengono con molte istruzioni che nel complesso risultano lente.

Non è possibile controllare i problemi legati alla velocità di lettura di Python, per cui le ottimizzazioni ricercate sono rivolte ai primi due punti.

La prima problematica affrontata è stata la dimensione del file. Sono stati individuati tre miglioramenti:

- file binario. Il salvataggio in un file binario piuttosto che di testo, svincola il file a contenere solo sequenze di caratteri riconducibili a quelli leggibili, evitando di dover eseguire conversioni in fase di scrittura e lettura;

- matrice simmetrica. La matrice di distanza per definizione è simmetrica in quanto, ricordando le proprietà delle misura di distanza, $d(A, B) = d(B, A)$. Per questo motivo è necessario salvare soltanto metà matrice, con il conseguente dimezzamento della dimensione del file.

- rappresentazione dei numeri. Generalmente i numeri interi e con la virgola sono rappresentati con 4 Byte. Supponendo che una approssimazione dei valori non compromettesse la qualità della misura, le varie distanze sono state rappresentante con un singolo Byte. La questione è legata al range di valori che può assumere la distanza. Con un singolo Byte sono rappresentabili tutti i numeri interi compresi tra 0 e 255, mentre con 4 quelli compresi tra 0 e 65536.

Il passaggio da una rappresentazione a 4 Byte a quella ad uno singolo, non necessariamente significa approssimazione. Se la misura di distanza era già per tutti i valori un numero naturale compreso entro 255, allora il passaggio è automatico senza perdita di informazione. Per i numeri reali invece la conversione potrebbe approssimare il valore. Infatti le distanze sono state convertite nel seguente modo: sono normalizzate così da essere comprese tra 0 e 1, moltiplicate per 255 e arrotondate all'intero più vicino. Così facendo tutte le distanze diventano comprese tra 0 e 255, in modo da poter essere espresse in un Byte.

Tuttavia sotto determinate condizioni anche i numeri con la virgola potrebbero non perdere precisione. Poniamo l'esempio di LCSS applicata a tutti gli utenti con al massimo 6 validazioni. La similarità LCSS tra due sequenze A e B è calcolata come il rapporto tra la lunghezza della sottosequenza comune e la lunghezza massima tra le le sequenze A e B, quindi la similarità sarà un valore tra 0 e 1. Il denominatore di questo rapporto potrà essere solo un numero compreso tra 1 e 6. Per questo motivo, se si moltiplica la misura trovata per il minimo comune multiplo $mcm(1,2,3,4,5,6) = 60$, si otterrà un numero compreso tra 0 e 60 e inoltre questo sarà un numero intero. Infatti il denominatore si semplificherà con l'mcm, come mostra l'esempio seguente

$$A = (f1, f2, f3, f4, f5)$$
$$B = (f1, f2, f3, f5)$$
$$C = (f1, f2, f4)$$
$$D = (f1, f3)$$
$$LCSS(A,B) = \frac{4}{5} = 0.8$$
$$LCSS(A,C) = \frac{3}{5} = 0.6$$
$$LCSS(A,D) = \frac{2}{5} = 0.4$$
$$LCSS(C,D) = \frac{1}{3} = 0.333$$

$$LCSS(A,B) * 60 = \frac{4 * 60}{5} = 48$$
$$LCSS(A,C) * 60 = \frac{3 * 60}{5} = 32$$
$$LCSS(A,D) * 60 = \frac{2 * 60}{5} = 24$$
$$LCSS(C,D) * 60 = \frac{1 * 60}{3} = 20$$

Inoltre la moltiplicazione non altera la qualità della misura in quanto è solo un cambio di scala. Per ottenere la distanza infine basta sottrarre a 60 la similarità. Queste operazioni permettono di rientrare nel primo caso descritto, cioè di un numero intero compreso tra 0 e 255. Questo esempio mostra come, sotto

determinate condizioni e accorgimenti, la diversa rappresentazione numerica non comporta perdita di precisione.

I tre miglioramenti insieme contribuiscono a ridurre la dimensione del file di 8 volte, la Tabella 5.1 mostra la riduzione della dimensione a fronte della possibile perdita di precisione.

Numero utenti	Originale	Migliorie
10000	0.4 GB	0.05 GB
20000	1.6 GB	0.2 GB
30000	3.6 GB	0.45 GB
70000	19.6 GB	2.45 GB

Tabella 5.1: Confronto della dimensione dei file prima e dopo le ottimizzazioni

Il problema del passaggio per il disco è stato affrontato con due modalità diverse: la prima usare una RAM disk e la seconda evitare il disco sfruttando il meccanismo della pipeline.

Una RAM disk è l'allocazione di una parte della memoria RAM come memoria secondaria per garantire prestazioni estremamente superiori. Infatti la velocità di accesso in lettura e scrittura di questa allocazione sarà di qualche ordine di grandezza superiore rispetto al disco normale. Ovviamente la RAM disk ha dei limiti, il principale è la sua volatilità: se manca l'alimentazione il contenuto è perso. Questo limite tuttavia non è un problema per lo scopo ricercato, in quanto il file da salvare è solo temporaneo, in attesa di essere convertito dallo script di Python.

Inaspettatamente l'utilizzo della RAM disk per salvare il file generato dal programma C non ha portato alcun vantaggio: i tempi con questa soluzione e senza sono i medesimi. Ritengo che l'utilizzo di una macchina virtuale penalizzi questo meccanismo e non ne consenta il corretto funzionamento.

Nei sistemi operativi Unix-like la pipeline è un meccanismo per la comunicazione tra processi distinti. Una pipeline è un insieme di processi concatenati insieme tali per cui gli standard stream sono collegati, lo standard output di un processo diventa lo standard input di quello successivo. Un limite di questo meccanismo è il tipo di stream consentito: è permesso solo uno stream di caratteri e non binario.

In questa analisi il programma C e lo script Python sono stati collegati con questa tecnica, rendendo lo standard output del primo lo standard input del secondo. Sono stati testati due approcci diversi:

- calcolo completo della matrice in C e successivo invio al programma Python;
- ogni cella calcolata viene inviata immediatamente al programma Python.

Il calcolo della matrice completo e il successivo invio si è dimostrata fallimentare. L'esecuzione non era ancora terminata dopo 3 ore dall'avvio. Il possibile problema che ho individuato è la dimensione del buffer degli stream, il programma C ha un flusso in uscita maggiore di quello che la pipe o Python possono gestire.

L'invio continuo delle celle calcolate invece ha introdotto un problema differente. Il programma C è multithreading e le celle della matrice sono calcolate in

parallelo. Tuttavia i metodi di scrittura dello standard output non sono thread-safe, per cui, se più thread provano a scrivere in contemporanea è possibile che i loro dati si sovrappongono. La soluzione è stata l'introduzione di un semaforo POSIX, che impedisce ai thread di accedere nello stesso momento allo stream di output. Il semaforo ha limitato la velocità di esecuzione del programma C, ma ha aumentato quella complessiva in quanto il programma python carica la propria matrice continuamente e non solo alla fine dell'esecuzione del programma C.

Usando il meccanismo della pipe in ogni caso si è deciso di continuare ad usare la misura approssimata di distanza. Questo accorgimento è stato imposto in quanto se la matrice fosse di numeri a 4 Byte piuttosto che a uno singolo, gli algoritmi di clustering non riuscirebbero ad essere eseguiti per problemi di memoria.

Riassumendo il calcolo della matrice ottimizzato con i miglioramenti descritti diventa:

- programma C: calcola la matrice di distanza dimezzata in multithreading. Ogni volta che una cella è calcolata, le sue coordinate e il suo valore approssimato sono scritte nello standard output;
- script Python: legge continuamente dallo standard input coordinate e valore corrispondente, inserendoli nella matrice numpy. Al termina salva la matrice con i metodi di libreria;
- tecnica della pipeline tra programma C e Python.

Questa soluzione è stata adottata per tutte le matrici sufficientemente grandi (maggiori di 6000 elementi), per insiemi più ristretti di elementi questo meccanismo non è necessario in quanto il programma scritto in python riesce a gestire autonomamente la creazione della matrice di distanza in tempi utili.

La Tabella 5.2 mostra i risultati della creazione della matrice svolto su 30000 utenti con i diversi meccanismi, da ricordare che la complessità è quadratica rispetto al numero di utenti.

Implementazione	Tempo totale creazione matrice
Calcolata in python	$\sim$2 ore
Calcolata in C e salvata su file	$\sim$1.30 ora
Calcolata in C e passata attraverso pipe intera	$>$3 ore, non terminato
Calcolata in C e passata attraverso pipe cella per cella	$\sim$25 minuti

Tabella 5.2: Tempi di esecuzione della creazione della matrice con le diverse scelte implementative

5.5 Analisi compiuta sui titoli da 24 ore

Nei paragrafi precedenti sono stati introdotti gli algoritmi di clustering e le misure di distanze, questa sezione descriverà la tecnica adottata e i risultati ottenuti dall'analisi delle validazioni dei turisti. Allo stato attuale l'analisi è stata svolta per i titoli da 24 ore, ma è facilmente adattabile ai titoli con durata maggiore.

L'analisi prevede due fasi distinte principali:

- clustering degli utenti basato solo sulle sequenze di fermate;
- clustering successivo degli utenti di ogni cluster basato sull'ora di percorrenza.

5.5.1 Clustering delle sequenze

Il clustering degli utenti basato solo sulle sequenze degli utenti si è svolta testando le diverse misure di distanza con i diversi algoritmi di clustering a disposizione.

Gli utenti totali sono 300000, ma escludendo il fattore tempo le possibili sequenze di fermate trovate diminuiscono a 70000, questo significa che molti utenti sono duplicati esatti.

I primi tentativi hanno creato matrici di distanza in questo insieme ridotto di utenti, per motivi di complessità, e testato diversi algoritmi di clustering. La qualità del risultato è stata molto bassa: il problema fondamentale è l'incapacità delle varie misure di distanza di valutare utenti con lunghezza di sequenze di fermate molto diverse tra loro.

La lunghezza delle sequenze è stata quindi usata per dividere a priori gli utenti su cui effettuare clustering.

Gli utenti che hanno validato una sola fermata sono 32251 e il 70% è concentrato in appena 10 fermate. Come mostra la Tabella 5.3 gli utenti con una sola validazione possono essere raggruppati direttamente senza dover eseguire algoritmi di clustering.

Id	Fermata	Numero utenti	Percentuale	% cumulata
5032	ferrovia	5974	18.52	18.52
5501	p.le roma	4136	12.82	31.34
5009	s. zaccaria	4050	12.56	43.9
5063	murano	2212	6.86	50.76
5038	rialto	2020	6.26	57.02
-1	TERRA	1972	6.11	63.13
15060	f.te nove	1599	4.96	68.09
5068	burano	1445	4.48	72.57
5053	s. marco	1148	3.56	76.13
5001	lido	1072	3.32	79.45

Tabella 5.3: Fermate più frequentate dagli utenti con una sola validazione

Allo stesso modo gli utenti che hanno validato esattamente due fermate sono raggruppabili senza dover eseguire algoritmi di clustering. Sono 65510 utenti totali che possono essere descritti da 1270 tratte distinte. Il 50% di questi utenti può essere descritto da appena 15 coppie di fermate come mostra la Tabella 5.4.

Gli utenti che hanno validato esattamente 3 fermate sono 76410, come per gli utenti precedenti sono stati identificati tutte le possibili tratte che risultano essere 7207. Le 15 tratte più frequenti descrivono il 25% degli utenti, le 115 più frequenti arrivano al 50%. Nella tabella 5.5 sono riportate le 15 tratte più frequenti.

Id	Fermata	Numero utenti	Percentuale	% cumulata
(5063, 5068)	murano - burano	4071	6.21	6.21
(15108, 5009)	sabbioni - s. zaccaria	3626	5.54	11.75
(5032, 5038)	ferrovia - rialto	2702	4.12	15.87
(5032, 5009)	ferrovia - s. zaccaria	2430	3.71	19.58
(5032, -1)	ferrovia - TERRA	2280	3.48	23.06
(5009, 5068)	s. zaccaria - burano	2261	3.45	26.51
(15060, 5068)	f.te nove - burano	2245	3.43	29.94
(5501, 5038)	p.le roma - rialto	2008	3.07	33.01
(5032, 5063)	ferrovia - murano	1999	3.05	36.06
(5009, 5063)	s. zaccaria - murano	1753	2.68	38.74
(5501, 5009)	p.le roma - s. zaccaria	1722	2.63	41.37
(15060, 5063)	f.te nove - murano	1430	2.18	43.55
(5501, -1)	p.le roma - TERRA	1414	2.16	45.71
(5032, 5053)	ferrovia - s. marco	1218	1.86	47.57
(5501, 5063)	p.le roma - murano	1144	1.75	49.32

Tabella 5.4: Percorsi più frequenti per gli utenti che hanno validato solo due fermate

Id	Fermata	Utenti	%	% cumulata
(5032, 5063, 5068)	ferrovia - murano - burano	3836	5.02	5.02
(15060, 5063, 5068)	f.te nove - murano - burano	3591	4.7	9.72
(5009, 5063, 5068)	s. zaccaria - murano - burano	2738	3.58	13.3
(5501, 5063, 5068)	p.le roma - murano - burano	1831	2.4	15.7
(15060, 5068, 5063)	f.te nove - burano - murano	1703	2.23	17.93
(15108, 5038, 5009)	sabbioni - rialto - s. zaccaria	869	1.14	19.07
(5009, 5068, 5063)	s. zaccaria - burano - murano	784	1.03	20.1
(15108, 5068, 5063)	sabbioni - burano - murano	712	0.93	21.03
(5032, 5009, 5063)	ferrovia - s. zaccaria - murano	680	0.89	21.92
(5063, 5068, 5063)	murano - burano - murano	602	0.79	22.71
(5063, 5068, 5038)	murano - burano - rialto	538	0.7	23.41
(15108, 5068, 5009)	sabbioni - burano - s. zaccaria	527	0.69	24.1
(5032, 5063, 5009)	ferrovia - murano - s. zaccaria	504	0.66	24.76
(5063, 5068, 5009)	murano - burano - s. zaccaria	494	0.65	25.41
(15060, 5068, 5038)	f.te nove - burano - rialto	461	0.6	26.01

Tabella 5.5: Percorsi più frequenti per gli utenti che hanno validato esattamente 3 fermate

Questo tipo di approccio ovviamente non è scalabile, in quanto l'incremento della lunghezza delle sequenze aumenta in modo esponenziale le possibili tratte distinte. Tuttavia permette di escludere, almeno per il momento, gli utenti che hanno validato al massimo 3 fermate. In questo modo oltre a garantire che le misure di similarità funzionino meglio, il numero di utenti si riduce a quasi la metà e consente di calcolare le matrici di distanza e eseguire gli algoritmi in tempi minori. Gli utenti rimasti con più di 3 validazioni sono circa 150000.

Per evitare i problemi legati alla differenza di lunghezza dei loro tracciati e

comunque mantenere una porzione significativa degli utenti, gli algoritmi di clustering sono stati testati solo agli utenti con lunghezza pari a 4 o 5. Gli utenti con questo numero di validazioni sono 105000 sui 150000 rimasti, ma quelli distinti sono appena 30000.

Sono state calcolate le matrici di distanza usando le seguenti metriche:

- Jaccard su coppie di due fermate
- Longest Common subsequence con buchi (LCSS)
- Edit
- LCSS bilanciata

DBscan è stato il primo algoritmo utilizzato, ma ha evidenziato subito un limite. Comunque si cambino i parametri e la distanza i risultati possibili sono sempre due:

- quasi l'interezza degli utenti è classificata nello stesso cluster. Questo fenomeno avviene quando i core object sono troppi e troppo vicini tra loro, l'algoritmo li concatena tutti e in questo modo viene generato un unico cluster con quasi tutti gli utenti;
- quasi l'interezza degli utenti è classificata come outlier e quindi non inserita in nessun cluster. Questo fenomeno è l'opposto, il raggio e il numero di elementi del vicinato è troppo piccolo per riuscire a identificare qualche core object. Essendo pochi, la maggior parte degli utenti rimane fuori dai cluster.

Il clustering gerarchico invece si è dimostrato molto più appropriato. Come descritto nel paragrafo 5.1.3, questo algoritmo, oltre alla misura di distanza tra singoli elementi, richiede anche una misura di distanza tra cluster. Quella media non può essere usata in quanto le tratte su cui si esegue l'algoritmo non contengono l'informazione sulla loro numerosità, per cui la media risulterebbe distorta. Al contrario la distanza minima e massima non sono penalizzate dall'assenza della numerosità. La distanza minima è stata testata, ma soffre dello stesso problema di DBscan, cioè aggrega troppo facilmente i cluster. La distanza massima invece si è dimostrata essere quella che ha prodotto i migliori risultati, come descritto nei prossimi paragrafi.

Il livello di qualità dei clustering eseguito è stato valutato introducendo una nuova proprietà. Una proprietà in grado di esprimere se ogni singolo cluster sia significativo, cioè se rappresenta un insieme di utenti coeso che hanno percorso lo stesso tracciato.

Un cluster significativo è stato definito attraverso queste proprietà:

- un numero minimo di utenti presenti;
- un numero minimo di coppie di fermate validate consecutivamente;
- una percentuale minima di utenti che devono essere passati per ogni coppia del punto precedente.

L'esempio seguente descrive meglio questa proprietà. La Tabella 5.6 contiene gli utenti di un ipotetico cluster Z e la Tabella 5.7 mostra le coppie di fermate validate consecutivamente e la percentuale di utenti di Z che le hanno percorse.

Utente	Percorso
A	(f1, f2, f3, f4, f5)
B	(f1, f2, f3, f5)
C	(f2, f3, f5)
D	(f1, f3, f4, f5)
E	(f1, f2, f3, f5, f4)

Tabella 5.6: Utenti di un ipotetico cluster Z

Coppie	Utenti	Percentuale
(f1,f2)	3	60%
(f1,f3)	1	20%
(f2,f3)	4	80%
(f3,f4)	2	40%
(f3,f5)	3	60%
(f4,f5)	2	40%
(f5,f4)	1	20%

Tabella 5.7: Coppie di fermate consecutive con le percentuali di passaggi degli utenti del cluster Z

Ipotizziamo di fissare come parametri per definire un cluster significativo i seguenti: 3 utenti, 2 coppie di fermate e il 70% come percentuale. Il cluster risulterebbe non significativo, infatti *(f2, f3)* è l'unica coppia percorsa da almeno il 70% degli utenti. Tuttavia abbassando la soglia della percentuale al 60%, il cluster risulta significativo in quanto tutti e tre i vincoli vengono rispettati.

I parametri scelti per definire un cluster significativo sono stati: 200 utenti con almeno due tratte percorse come minimo dal 60% degli utenti. Il clustering migliore risulta essere quello che con i suoi cluster significativi descrive il maggior numero di utenti.

Il risultato del cluster gerarchico effettuato con le diverse unità di misura è descritto nella Tabella 5.8 e nel grafico in Figura 5.2. L'implementazione dell'algoritmo di sklearn richiede che sia specificato anche il numero di cluster ricercati.

Distanza	250 cluster		500 cluster		750 cluster		1000 cluster	
Edit	11	9668	26	16471	37	21226	42	21998
Jaccard	13	9460	18	11410	22	12479	23	12681
LCSS bil	16	25665	19	32873	54	45680	70	49589
LCSS	19	16529	25	19882	42	21479	52	25425
Distanza	1250 cluster		1500 cluster		1750 cluster		2000 cluster	
Edit	46	23242	48	23314	50	23652	51	23251
Jaccard	27	14482	29	16102	31	16599	33	17586
LCSS bil	77	51101	75	50804	77	49392	75	48918
LCSS	62	27091	63	27202	63	27115	63	27091

Tabella 5.8: Risultati dei cluster effettuati. Per ogni numero di cluster ricercato sono riportati il numero di cluster significativi trovati e il numero di utenti che descrivono

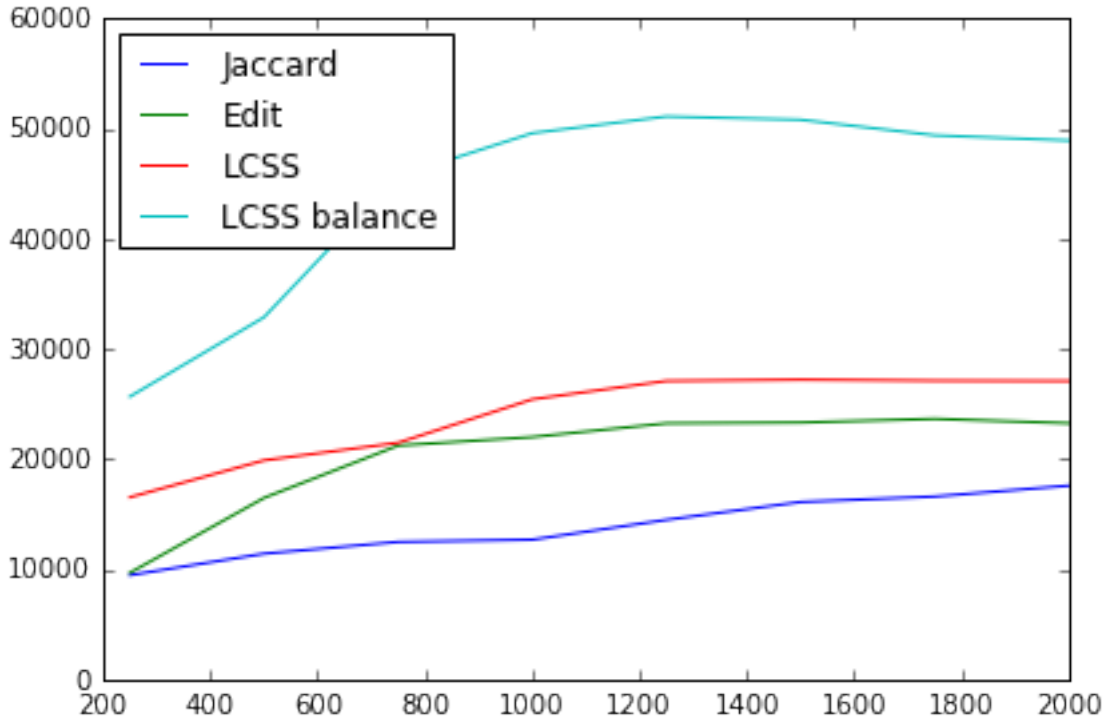

Figura 5.2: Grafico degli utenti descritti dai cluster significativi, sull'asse x il numero di cluster ricercati

Il risultato migliore è stato trovato con la distanza LCSS bilanciata con 1250 cluster, gli utenti totali che riesce a descrivere con i 77 cluster significativi sono 51101 su 104808, quasi il 50%.

Il grafico in Figura 5.2 mostra un risultato atteso conforme alla definizione data di cluster significativo. Ha un andamento crescente, una fase stabile e una lieve tendenza a decrescere alla fine. Infatti un numero troppo piccolo di cluster ricercati tende a raggruppare maggiormente gli utenti e quindi ridurre la percentuale di tratte comuni percorse. Un numero troppo grande, invece, distribuisce maggior-

mente gli utenti in cluster diversi, per cui molti avranno un supporto di utenti non sufficiente per essere definito significativo.

Trovati i cluster significativi è necessario l'individuazione di un rappresentante per ognuno che riesca a comprime e descrive il comportamento degli utenti all'interno del cluster. Il rappresentante inizialmente è stato scelto come l'utente mediamente meno distante dagli altri utenti. Questa scelta tuttavia presenta un problema come dimostra la mappa in Figura 5.3. La mappa contiene le informazioni legate alla visita delle fermate attraverso la mappa di calore e il tracciato del rappresentante attraverso la linea spezzata. Il rappresentante ha validato le fermate nell'ordine indicato dalla Tabella 5.9.

Figura 5.3: Esempio di rappresentante poco significativo

Ordine	Id	Fermata
1	5032	Ferrovia
2	15060	Fondamenta Nove
3	5063	Murano
4	5009	San Zaccaria

Tabella 5.9: Percorso del rappresentante

La mappa mostra come l'utente più simile agli altri percorre le fermate più significative, ma transita anche per fermate molto meno frequentate. Questo fenomeno è frequente e si spiega immaginando diversi utenti che provengono da zone

diverse, si uniscono e percorrono la stessa tratta centrale e infine si dividono ancora. Il diagramma Sankey[1] in Figura 5.4 ne mostra un esempio.

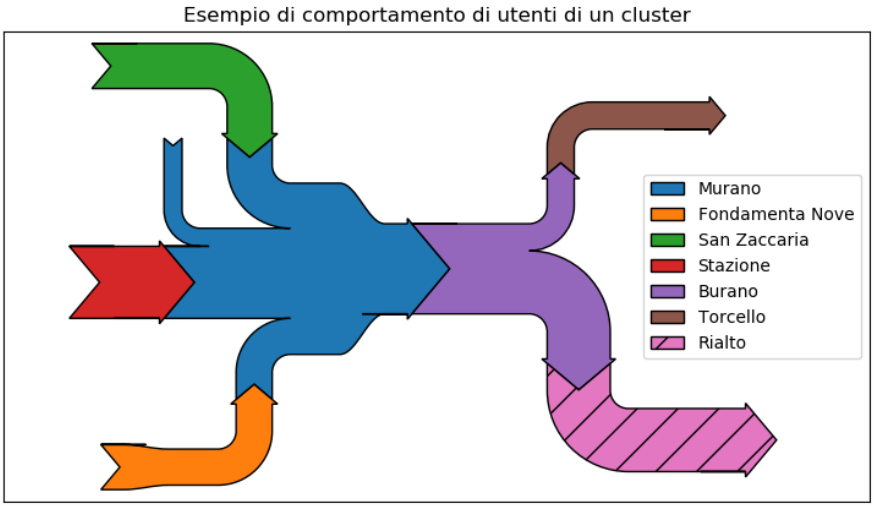

Figura 5.4: Esempio di utenti di un cluster: diversi origini, stesso percorso centrale, diverse destinazioni

La soluzione è stata accettare solo quelle fermate del rappresentante che siano state validate da almeno una percentuale degli utenti del cluster. Il risultato applicato all'esempio precedente è descritto nella mappa in Figura 5.5. La percentuale richiesta è fissata allo stesso valore del parametro dei cluster significativi, il 60%.

[1]Il diagramma di Sankey è un particolare tipo di diagramma di flusso in cui l'ampiezza delle frecce è disegnata in maniera proporzionale alla quantità di flusso

Figura 5.5: Esempio di rappresentante corretto

La tabella 5.10 mostra i risultati ottenuti specificando il cluster, la sua numerosità e il suo rappresentante.

Tabella 5.10: Elenco dei cluster significativi

Id	Numero utenti	Rappresentante
124	6493	('f.te nove', 'murano', 'burano')
597	3479	('s. zaccaria', 'murano', 'burano')
328	2980	('p.le roma', 'murano', 'burano')
1214	2623	('ferrovia', 'murano', 'burano')
696	1977	('murano', 'burano', 'murano')
270	1901	('murano', 'burano', 'f.te nove')
138	1736	('f.te nove', 'burano', 'murano')
400	1655	('ferrovia', 'murano', 'burano', 's. zaccaria')
1203	1421	('murano', 'burano', 'rialto')
486	1346	('sabbioni', 'burano', 'murano', 's. zaccaria')
607	1323	('ferrovia', 'murano', 'burano', 'murano')
647	912	('p.le roma', 's. zaccaria', 'murano')
112	868	('ferrovia', 's. zaccaria', 'murano')
148	794	('s. zaccaria', 'murano', 's. zaccaria')
648	737	('murano', 'burano', 'TERRA')
953	652	('f.te nove', 'burano', 'rialto')

Continua nella pagina seguente

Tabella 5.10 – continua dalla pagina precedente

Id	Numero utenti	Rappresentante
1118	640	('s. zaccaria', 'burano', 'murano')
568	624	('f.te nove', 'burano', 'f.te nove')
829	618	('TERRA', 'murano', 'burano')
350	587	('f.te nove', 'murano', 's. zaccaria')
1096	583	('f.te nove', 'burano', 'murano', 's. zaccaria')
559	575	('murano', 'torcello', 'burano')
644	503	('murano', 'burano', 'lido')
485	487	('ferrovia', 'rialto', 'ferrovia')
944	467	('ferrovia', 'f.te nove', 'burano', 'f.te nove')
333	453	('TERRA', 'p.le roma', 's. zaccaria')
718	445	('f.te nove', 'burano', 's. zaccaria')
721	440	('rialto', 's. zaccaria', 'murano')
697	429	('sabbioni', 'burano', 'torcello', 'murano')
665	419	('TERRA', 'p.le roma', 'rialto')
171	417	('f.te nove', 'murano', 'f.te nove')
567	407	('murano', 'burano', 'ferrovia')
231	397	('murano', 'burano', 'f.te nove', 's. zaccaria')
1164	378	('rialto', 'ferrovia', 'murano')
595	375	('s. zaccaria', 'burano', 's. zaccaria')
592	367	('f.te nove', 'torcello', 'burano')
53	352	('rialto', 'TERRA', 'rialto')
1102	320	('TERRA', 'ferrovia', 'murano')
1133	319	('rialto', 'murano', 'burano')
810	319	('murano', 'ferrovia', 'rialto')
205	319	('f.te nove', 'cimitero s. michele', 'murano')
311	315	('s. zaccaria', 'lido')
650	313	('murano', 'f.te nove', 'burano')
49	309	('p.le roma', 'rialto', 'p.le roma', 'rialto')
722	302	('murano', 'burano', 'torcello')
632	300	('ferrovia', 'burano', 'murano')
653	296	('f.te nove', 'burano', 'torcello', 'murano')
682	293	('ferrovia', 'murano', 'ferrovia')
1144	290	('ferrovia', 's. zaccaria', 'burano')
1098	282	('s. zaccaria', 'ferrovia', 'murano')
817	278	('p.le roma', 'rialto', 'p.le roma')
1050	277	('s. zaccaria', 'burano', 'rialto')
792	270	('lido', 'burano', 'murano')
1100	267	('TERRA', 'p.le roma', 'murano')
929	263	('s. zaccaria', 'burano', 'f.te nove')
738	263	('s. zaccaria', 'lido', 'burano')
770	260	('murano', 'burano', 's. marco')
50	260	('murano', 'burano', 'Rialto Mercato')
733	235	('chioggia', 'TERRA', 'lido')
748	232	('p.le roma', 's. zaccaria', 'p.le roma')

Continua nella pagina seguente

Tabella 5.10 – continua dalla pagina precedente

Id	Numero utenti	Rappresentante
523	232	('s. zaccaria', 'sabbioni', 'burano')
749	231	('p.le roma', 's. zaccaria', 'burano')
891	229	('TERRA', 's. zaccaria', 'murano')
576	228	('ferrovia', 'rialto', 's. zaccaria')
264	228	('s. zaccaria', 'murano', 's. zaccaria', 'rialto')
761	220	('sabbioni', 'rialto', 'p.le roma', 's. zaccaria')
313	214	('s. zaccaria', 'murano', 'rialto')
238	212	('f.te nove', 'burano', 'TERRA')
781	211	('f.te nove', 'burano', 'ca oro')
574	211	('rialto', 'ferrovia', 's. zaccaria')
152	211	('ferrovia', 's. zaccaria', 'lido')
542	210	('burano', 'murano', 'f.te nove')
61	208	('arsenale', 'murano', 'burano')
613	206	('burano', 'torcello', 'f.te nove')
34	205	('rialto', 'ferrovia', 'rialto')
495	203	('f.te nove', 'burano', 'murano', 'TERRA')
223	200	('rialto', 's. zaccaria', 's. giorgio')

Gli utenti che hanno validato esattamente 3 fermate possono essere facilmente associati a questi cluster. Infatti è sufficiente che un utente corrisponda perfettamente al rappresentante di un cluster per essere associato ad esso. Usando questa tecnica il 33% degli utenti con 3 validazioni (circa 25000 su 75000) può essere associato ai cluster significativi trovati.

Di seguito saranno rappresentati e descritti con più dettagli due cluster significativi e rappresentativi.

5.5.2 Cluster Ferrovia-Murano-Burano

Questo cluster racchiude 2623 utenti e descrive uno delle pattern che più ci si può aspettare: il turista che arriva alla stazione, prende il battello, e compie il giro delle isole, muovendosi prima a Murano e poi a Burano. La mappa in Figura 5.7 mostra come sono distribuite nello spazio le tratte di questo cluster, mentre il diagramma Sankey in Figura 5.6 mostra con una maggiore facilità il flusso degli utenti. La frequenza di ogni tratta è specificata nella Tabella 5.11.

Tratta		Percentuale
Ferrovia	Murano	0.97%
Murano	Burano	0.97%
Burano	Rialto	0.36%
Burano	Ferrovia	0.12%
Burano	Torcello	0.1%
Burano	Lido	0.08%
Burano	S. Marco	0.07%

Tabella 5.11: Frequenza delle varie tratte del cluster Ferrovia-Murano-Burano

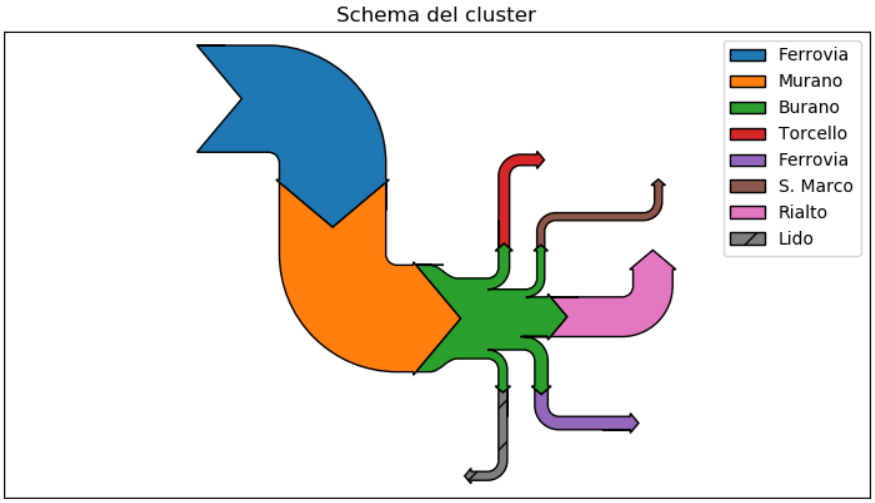

Figura 5.6: Flusso degli utenti all'interno del cluster Ferrovia-Murano-Burano

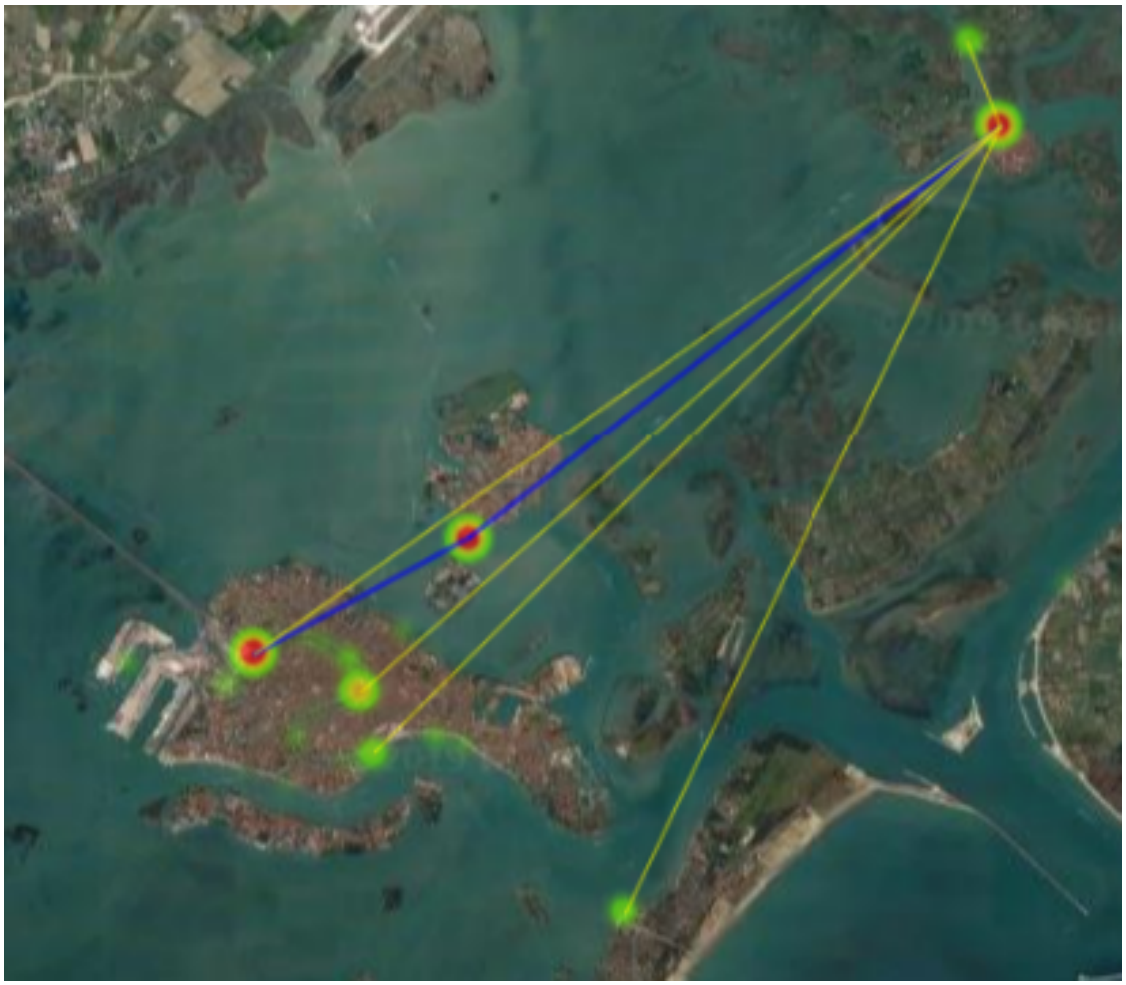

Figura 5.7: Mappa delle traiettorie del cluster Ferrovia-Murano-Burano, in blu quella principale, in giallo quelle secondarie

5.5.3 Cluster Murano-Burano-Terra

Questo cluster descrive 737 utenti. Merita un approfondimento in quanto presenta alcuni punti interessanti. Come per il cluster descritto in precedenza sono riportate la mappa in Figura 5.9, il diagramma di Sankey in Figura 5.8 e la Tabella delle frequenze 5.12.

Tratta		Percentuale
Terra	P.Le Roma (vaporetto)	0.1%
Terra	Piazzale Roma (bus)	0.06%
P.Le Roma	Murano	0.46%
Ferrovia	Murano	0.45%
Murano	Burano	0.99%
Burano	Terra	0.98%

Tabella 5.12: Frequenza delle varie tratte del cluster Murano-Burano-Terra

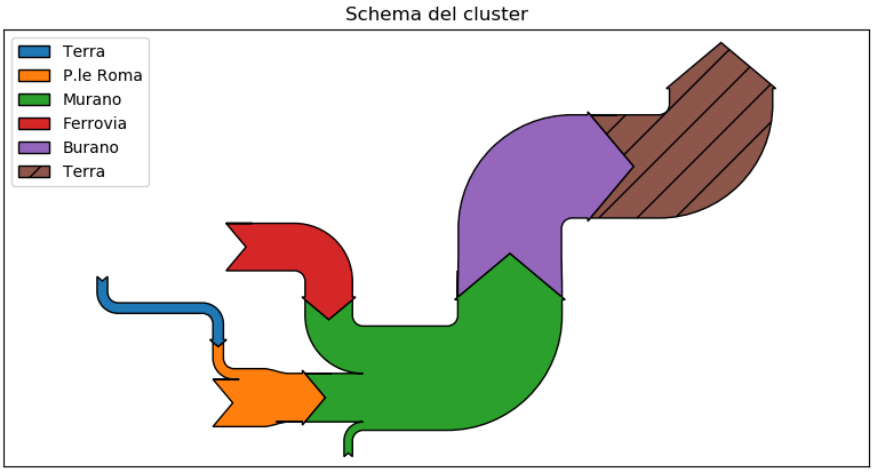

Figura 5.8: Flusso degli utenti all'interno del cluster Murano-Burano-Terra

Figura 5.9: Mappa delle traiettorie del cluster Murano-Burano-Terra, in blu quella principale, in giallo quelle secondarie

Il primo fatto interessante è come il cluster sia riuscito a descrivere utenti che iniziano le loro tratte in due fermate distinte (p.le Roma e Ferrovia) e che terminano sulla terraferma. Per cui non è difficile credere che il luogo di origine di questi turisti sia per la maggior parte la terraferma, sia per chi ha iniziato dalla stazione sia per chi ha iniziato da p.le Roma.

L'altro fatto di interesse è la mancanza della fermata automobilistica di piazzale Roma, che dovrebbe essere validata per poter timbrare una di TERRA. Le spiegazioni più probabili sono le seguenti:

- gli utenti non sono obbligati a validare e hanno saltato questa fermata;

- le validazioni del servizio automobilistico sono associate a una fermata valutando le coordinate geografiche del bus all'ora della validazione e le fermata del percorso più vicina. Per questo motivo se un utente non valida appena salito sul mezzo, ma durante il tragitto e da metà del ponte della libertà, potrebbe succedere che la sua validazione venga associata alla fermata TERRA che racchiude quasi tutte quelle comprese nella terraferma.

5.5.4 Clustering temporale

Determinati i cluster significativi, l'analisi è proseguita individuando come ogni utente si muove rispetto al tempo. È stato effettuato un secondo livello di clustering suddividendo ulteriormente gli utenti di ogni cluster in base all'orario di percorrenza.

Il primo passo è stata la determinazione di alcune fasce orarie significative, ciò ha permesso di discretizzare l'orario consentendo di effettuare più facilmente il clustering degli utenti .

Le fasce orarie sono state determinate eseguendo ancora una volta algoritmi di clustering. L'algoritmo scelto è K-means, perché è in grado di determinare cluster che dispongono di un elemento centrale ben definito a cui intorno ruotano gli altri elementi. K-means è stato eseguito ricercando 8 cluster e usando come metrica di distanza quella euclidea, il dataset che gli è stato fornito è composto da tutti gli orari delle validazioni rappresentati come minuti dall'inizio della giornata. Sono stati fissati 8 cluster in quanto si ritiene che riescano a descrivere in modo esauriente i diversi momenti della giornata.

Il risultato ottenuto è descritto dalla Tabella 5.13, tuttavia per garantire una maggiore facilità di comprensione le fasce sono state approssimate con quelle contenute nella Tabella 5.14

Id fascia	Validazioni	Inizio	Fine	Durata
1	5005	0:0	5:37	5:37
2	135708	5:38	10:12	4:34
3	218990	10:13	12:05	1:52
4	195135	12:06	14:00	1:54
5	222112	14:01	15:56	1:55
6	211107	15:57	17:53	1:56
7	147399	17:54	20:15	2:21
8	75988	20:16	23:59	3:43

Tabella 5.13: Fasce orarie originali trovate dall'algoritmo

Per poter essere processato da un algoritmo di clustering, ogni utente deve essere rappresentato in modo omogeneo e significativo rispetto agli altri. La rappresentazione scelta è quella di un vettore di lunghezza pari a quella del rappresentante del cluster. La posizione i-esima del vettore rappresenta la fascia oraria in cui l'utente ha validato la i-esima fermata del rappresentante. Può succedere che un utente non abbia validato la j-esima fermata, in questo caso la fascia oraria è stimata valutando gli orari precedenti e successivi della sue validazioni.

Id fascia	Inizio	Fine	Durata
1	00:00	06:00	6:00
2	06:00	10:15	4:15
3	10:15	12:00	1:45
4	12:00	14:00	2:00
5	14:00	15:45	1:45
6	15:45	17:45	2:00
7	17:45	20:00	2:15
8	20:00	23:59	4:00

Tabella 5.14: Fasce orarie approssimate

Di seguito è presentato un esempio. Gli utenti A e B appartengono al cluster che ha come rappresentante R.

$$R = (f1, f2, f3)$$

$$A = [(f4, 09:00),\ (f1, 09:30),\ (f5, 10:45),\ (f2, 12:15),\ (f3, 17:00)]$$

$$B = [(f1, 08:15),\ (f5, 09:30),\ (f2, 17:15),\ (f3, 19:00),\ (f4, 21:00)]$$

Ogni orario è trasformato nella corrispondente fascia oraria:

$$A = [(f4, 2),\ (f1, 2),\ (f5, 3),\ (f2, 4),\ (f3, 6)]$$

$$B = [(f1, 2),\ (f5, 2),\ (f2, 6),\ (f3, 7),\ (f4, 8)]$$

Gli utenti sono rappresentati come vettori di 3 elementi, che identificano le fasce orarie delle fermate del rappresentante:

$$A = [2, 4, 6]$$

$$B = [2, 6, 7]$$

Sono stati testati diversi algoritmi con diversi parametri e distanze su diversi cluster. È stato scelto il migliore algoritmo valutando il coefficiente di silhouette e quello che mediamente ha fornito il coefficiente più alto è stato K-means.

Gli utenti di ogni cluster quindi sono stati ulteriormente divisi applicando l'algoritmo K-means con distanza euclidea. Il numero di centroidi ricercati dipende dalla numerosità del cluster su cui si applica, comunque cercando di ottenere che mediamente ogni cluster contenesse tra il 10% e il 20% degli utenti.

Questo meccanismo applicato al cluster Murano-Burano-Terra descritto al nella Sezione 5.5.3 produce i risultati contenuti in Tabella 5.15 e nel grafico in Figura 5.10. I cluster ricercati sono 5, il coefficiente di silhouette ha valore 0.47 e il numero di orari che sono stati stimati sono 22, cioè l'1%. A differenza di quanto ci si aspetterebbe, le fasce orarie dei cluster non sono tutte crescenti. Il quarto cluster infatti ha l'ultima fermata che avviene nella fascia oraria 3, mentre quella precedente in quella 6. Significa che l'ultima validazione è stata effettuata il giorno seguente, infatti il titolo di viaggio può avere validità tra giorni distinti.

Numero utenti	Percentuale	Murano	Burano	Terra
69	0.09	4	5	6
184	0.25	3	4	6
211	0.29	5	6	7
70	0.09	5	6	3
203	0.28	4	5	7

Tabella 5.15: Cluster di secondo livello per Murano-Burano-Terra. La colonna delle fermate identifica in che fascia oraria gli utenti della corrispondente riga hanno validato tale fermata

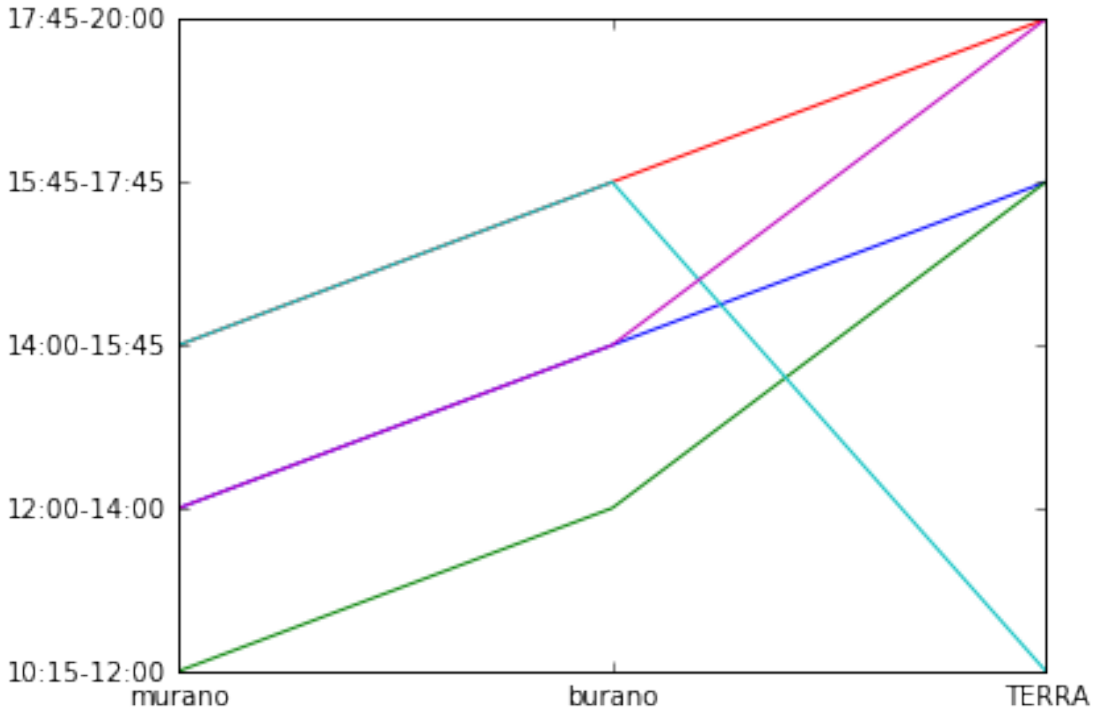

Figura 5.10: Rappresentazione grafica dei vari cluster temporali

5.5.5 Analisi dei punti di accesso

Questa sezione descrive il punto di partenza dei turisti e, in particolare, prova a stimare quanti di questi turisti provengono dalla città vera e propria di Venezia e quanti dalla terraferma e Lido.

Questa analisi è stata svolta contando le prime fermate validate di ciascun utente, focalizzando l'attenzione su quelle che possono essere un punto di accesso alla città. La Tabella 5.16 mostra il risultato. Il 57% degli utenti rientra in questi gruppi. Ferrovia, Piazzale Roma e Tronchetto, nonostante appartengono alla città, sono stati presi in esame in quanto rappresentano un possibile punto di accesso per un utente che arriva dalla terraferma, a differenza di fermate come Rialto che si trova nel cuore della città.

Fermata	Utenti	Percentuale
ferrovia	67996	21%
p.le roma	45969	14%
TERRA	23849	7%
sabbioni	21911	7%
tronchetto	11562	4%
lido	6266	2%
Stazione Mestre	2808	1%
chioggia	2308	1%

Tabella 5.16: Punti di accesso esterni alla città

5.5.6 Analisi della permanenza media

Questa sezione presenta una stima sulla permanenza dei turisti nelle isole di Murano, Burano e Torcello.

Allo stato attuale, il tempo di permanenza di ogni utente è stato calcolato come la differenza tra l'orario della validazione della fermata nell'isola e quello della validazione immediatamente precedente. Questa, tuttavia, è una sovrastima in quanto non viene considerato il tempo necessario per il viaggio.

La Tabella 5.17 mostra la permanenza nelle isole espresse in minuti. Si noti che la colonna con etichetta "Utenti" raccoglie tutti gli utenti che hanno effettuato una validazione nell'isola, con "Validi" quelli per cui è stato possibile stimare il tempo. Un utente non è valido se la prima delle sue fermate è esattamente quella dell'isola.

Isola	Utenti	Validi	Min	Max	Media	Mediana	1 Quartile	3 Quartile
Murano	134600	120333	6	1438	159	131	88	188
Burano	117871	113873	6	1436	165	147	113	190
Torcello	9068	8957	6	1404	104	86	61	124

Tabella 5.17: Studio della permanenza nelle principali isole

Capitolo 6

Conclusioni

La presente tesi ha analizzato il comportamento dei turisti di Venezia che hanno usufruito dei mezzi di trasporto pubblico nei mesi di settembre e ottobre 2018. Prima di tutto l'analisi ha riguardato la pulizia e il filtraggio dei dati, Capitolo 4, successivamente attraverso gli strumenti e la metodologia descritti nel Capitolo 5 sono stati identificati diversi cluster di utenti che descrivono a un primo livello le tratte percorse e a un secondo livello l'orario rispetto al quale le percorrono. L'analisi allo stato attuale è limitata ai titoli da un giorno con al massimo 5 validazioni. Si ottengono 127 cluster che descrivono circa 160000 su 300000 utenti, quasi il 57%, che sale al 64% se si considera che circa 50000 utenti non sono rientrati nell'analisi in quanto hanno un numero maggiore di validazioni rispetto a quelli studiati.

Il primo passo nello sviluppo futuro di questa tesi, sarebbe confrontare i cluster trovati con un test set per verificare se i comportamenti individuati siano effettivamente quelli reali oppure se sono troppo legati ai dati disponibili e le analisi compiute hanno commesso overfitting.

Ovviamente la seconda attività da sviluppare è continuare l'analisi dei titoli con durata maggiore di un giorno. È stata testata la stessa metodologia, tuttavia gli stessi parametri usati per identificare i cluster significativi non sono più appropriati ed è necessario provarne di diversi e meno vincolanti. Infatti, applicando gli stessi parametri usati per i titoli da 24 ore, al clustering dei titoli da 48 ore, nessun cluster significativo è stato individuato.

Oltre ai risultati ottenuti questa tesi ha creato una metodologia per analizzare questo tipo di dati. Sono stati prodotti diversi script e programmi che possono essere facilmente eseguiti per effettuare un'analisi su dataset differenti. Questi script hanno un grande vantaggio, ovvero sono parametrici rispetto a diversi fattori e proprietà. Gestione delle fermate, raggruppamento dei titoli, filtraggio delle validazioni, metrica di distanza e algoritmi di clustering sono facilmente personalizzabili e adattabili per effettuare lo stesso studio con vincoli e principi differenti.

In particolare la misura di distanza è fondamentale per la buona riuscita di questo tipo di analisi. Un possibile sviluppo e perfezionamento della tesi consiste nella ricerca di una nuova misura di distanza che sia più affidabile di quelle testate. Il problema maggiore riscontrato nelle misure usate è la sensibilità alla differenza della lunghezza delle sequenze, come descritto più in dettaglio nel Paragrafo 5.5.1. Inoltre la misura che ha ottenuto i risultati migliori non gode della disuguaglianza triangolare. Una misura che soddisfa la disuguaglianza triangolare sarebbe da

ricercare e privilegiare perché, oltre a garantire una base teorica più ampia, può essere applicata con maggiori garanzie a qualsiasi algoritmo di clustering.

Ulteriori sviluppi della tesi possono riguardare la creazione di un sistema di raccomandazione che suggerisca agli utenti il percorso migliore per la loro visita alla città in base al tempo di permanenza. L'integrazione con altri dati legati ai punti di interesse della città, infine, potrebbe aiutare a comprendere in misura maggiore le dinamiche descritte da questi cluster, completandoli con l'informazione delle visite ai musei, chiese e altre attrazioni della città.

Bibliografia

[1] Annuario del turismo. https://www.comune.venezia.it/it/content/studi.

[2] Annuario del turismo 2017. https://www.comune.venezia.it/sites/comune.venezia.it/files/immagini/Turismo/ANNUARIO%202017%20Ver%202.8.1%20cover.pdf.

[3] Standard gtfs. https://developers.google.com/transit/gtfs/.

[4] Numpy. http://www.numpy.org/.

[5] Scikit-learn. https://scikit-learn.org.

[6] Matplotlib. https://matplotlib.org/.

[7] Jupyter-gmaps. https://jupyter-gmaps.readthedocs.io.

[8] Maltempo veneto 26-30 ottobre 2018. https://it.wikipedia.org/wiki/Maltempo_sul_Triveneto_del_26-30_ottobre_2018.

[9] Acqua alta 29 ottobre 2018. http://www.veneziatoday.it/attualita/acqua-alta-venezia-oggi-29-ottobre-2018.html.

[10] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining Concepts and Techniques*. Elsevier, 3 edition, 2012.

[11] Longest common substring. https://www.geeksforgeeks.org/longest-common-substring-dp-29/.

Appendice A

Tabella dei titoli trovati

Tabella A.1: Elenco dei titoli di viaggio disponibili

Id	Nome titolo	Numero di validazioni
12315	Ferry11-autocarri+35q.	864
12400	SOSTITUTIVO Pass Imob	1
13003	Cav -Trep + Actv 24H	9479
14126	Extra tratta 6	1718
10020	24H metropolitano RES.	26
11113	Calcio ospiti A/R	6
11137	Traghetto Gratuito	4
14202	Big.extra AR t 2 TICKET NA	4
14122	Extra tratta 2	35062
10040	Bus+People mover online	488
7017	ARRIVA Extra tr. 5-6-7 BORDO	14
11136	75'-Tpl 6,30-ComVe1,20	5062
11552	72H RVenice+aerop.CS online	4704
10021	24H metropolitano RES+1	30
1109	Mens. cose animali RETE UNICA	10
11302	24ore online no aerobus	27181
7015	ARRIVA Extra tr.1 BORDO	29
11551	72 ore R.Venice online	36675
12104	Bigl.Mestre/Lido 75' a bordo	12796
13011	Bibione-S.Marco AR	37
201	bordo multip aeroporto autisti	147
12355	L.17-auto "C"da 4,01 a 4,50 mt	5969
11438	Gruppi e scuole online TVM cs	39
11186	CartaEventi 5 giorni	4
12365	Ferry17-autocarri+35q.	3354
11440	Gruppiescuole online TERMINAL	30
11175	Ev3-Tpl 26,50-C.Ve1,50	942
11231	7ggAerCS-Tpl49,6-CVe16,4	9612
85	VENDITA A BORDO 75' ORD.	557
11220	Bagaglio CartaVenezia	700

Continua nella pagina seguente

Tabella A.1 – continua dalla pagina precedente

Id	Nome titolo	Numero di validazioni
7001	ARRIVA VENETO tratta 1	133
7007	ARRIVA VENETO tratta 7	31
11254	72ore online aerobus AR	11626
11455	Bicicletta "concessionari"	513
12360	L.17-auto "D" oltre metri 4,50	11442
7020	ARRIVA Integ.Aerop. BORDO	7
7004	ARRIVA VENETO tratta 4	72
11241	7ggAerAR-Tpl55,6-CVe16,4	30592
5	75'-Tpl 6,30-ComVe1,20	24204
14124	Extra tratta 4	13805
13006	Jesolo + Actv 24H	15234
11305	72ore online no aerobus	97343
7019	ARRIVA Aeroporto BORDO	6
14125	Extra tratta 5	4891
12335	Bicicletta "Palmare"	2135
12330	Ferry11-Trasporti pericolosi	6
14145	Ville Venete solo linea 53 24H	7
11304	48ore online no aerobus	46298
11240	72 ore R.Venice+aeroporto AR	9510
11239	72hAerAR-Tpl45,4-CVe6,60	18823
16104	T.Fusina Ve+ACTV 72 ore	1798
13608	Extra tratta 8-9-10 TVM	2
14128	Extra tratta 8-9-10	945
13005	Jesolo - S.Marco AR	8488
14140	Ville Venete+24H actv urb+nav	207
10024	24H metropolitano ORD+1	19028
14121	Extra tratta 1	9626
11451	Ciclomotore fino 50cc	1302
11107	48h-Tpl 24,90-ComVe5,10	655150
12300	L.11-auto "AeB" fino a 4 metri	946
11253	48ore online aerobus AR	1925
15104	Bigl.urbano CHIOGGIA bordo	9
11180	Tragh-Tpl 4,10-C.Ve0,90	8648
12106	Bigl Aer-Venezia TSC	42491
11226	24hAerCS-Tpl20,9-CVe5,1	11477
11185	CartaEventi 3 giorni	1580
13601	Extra tratta 1 TVM	217
11181	Tragh-Tpl 7,60-C.Ve2,40	30067
11138	75'-Tpl 12,60-ComVe2,40	197263
12350	L.17-auto "AeB" fino a 4 metri	5187
14130	Extra tratta 1 BORDO	733
14123	Extra tratta 3	20906
11178	Ev12-Tpl 57,00-C.Ve3,00	162
11238	48hAerAR-Tpl36,9-CVe5,1	6215

Continua nella pagina seguente

Tabella A.1 – continua dalla pagina precedente

Id	Nome titolo	Numero di validazioni
11104	12h-Tpl 13,40-ComVe4,60 NA	8
13604	Extra tratta 4 TVM	70
14132	Extra tratte 2-3-4 BORDO	1357
200	bordo multip BUS urb. autisti	847
11174	Prenotazione Veicolo ABBONATO	80
11306	7 days online no aerobus	58437
7003	ARRIVA VENETO tratta 3	29
11114	Biglietto bici per turisti NA	1
206	bordo multip tr2-3-4 autisti	79
14203	Big.extra AR t 3 TICKET NA	2
15103	Bigl.urbano CHIOGGIA	462
24316	Omnibus+Actv tratta 3 T.	10
208	bordo multip tr8-9-10 autisti	1
12375	Ferry17-AUTOBUS	173
13603	Extra tratta 3 TVM	123
11431	Biglietto Soc. Sportive	651
11245	Aer+boat-Tpl12,50-C.Ve1,50	8070
13602	Extra tratta 2 TVM	410
11550	75'-Tpl 12,60-CVe2,40 online	1301
11345	7 days online aerobus CS	4736
10035	24H metropolitano ORD+2 online	172
7002	ARRIVA VENETO tratta 2	25
11108	72h-Tpl 33,40-ComVe6,60	497555
13605	Extra tratta 5 TVM	10
14127	Extra tratta 7	623
10023	24H metropolitano ORD.	268
11452	Ciclomotore oltre 50cc	105
11343	48ore online aerobus CS	3819
11307	75'-Tpl 6,00-ComVe1,50	3027
11346	Aeroporto-Venezia CS ONLINE	167
12320	Ferry11-carri+35q.rim.	62
10034	24H metropolitano ORD+1 online	95
207	bordo multip tr5-6-7 autisti	17
14136	Extra tratte 8-9-10 BORDO	18
15102	Carnet 10c. TICKET fs NA	3
12394	Prenotaz OCCASIONALE si barra	1573
12380	Ferry17-Trasporti pericolosi	75
12370	Ferry17-carri+35q.rim.	1071
11121	SpiaggeAR-Tpl 11,75-ComVe1,25	31
11256	Aeroporto-Venezia AR ONLINE	819
14134	Extra tratte 5-6-7 BORDO	61
11230	72 ore R.Venice+aeroporto CS	16098
13606	Extra tratta 6 TVM	5
10025	24H metropolitano ORD+2	12388

Continua nella pagina seguente

Tabella A.1 – continua dalla pagina precedente

Id	Nome titolo	Numero di validazioni
11344	72ore online aerobus CS	6800
10033	24H metropolitano ORD online	134
7016	ARRIVAExtra tr.2-3-4 BORDO	26
12340	Biglietto MOTO FINO 50 cc	94
11149	7gg-Tpl 43,60-ComVe16,40	255210
11341	24ore online aerobus CS	1468
13004	Cav - Trep + Actv 72H	1559
12101	Bigl.Aut.75'Mestre/Lido-tsc	343933
11246	Aeroporto-Venezia AR	4976
11222	Tariffa Agevolata	3121
11228	48hAerCS-Tpl30,9-CVe5,1	15154
11437	Gruppi e scuole online TVM ar	4951
12325	Ferry11-AUTOBUS	7
14150	Ville Venete l.53 24H online	4
80	VENDITA A BORDO 75' CV	437
11430	Gruppi e Scuole	8088
7010	ARRIVA VENETO AEROPORTO	78
11101	75'-Tpl 6,30-ComVe1,20	641556
11103	Traghetto ordinario NA	6
16103	T.Fusina Ve+ACTV 24 ore	3780
205	bordo multip tratta1 autisti	10
7008	ARRIVA VENETO tratta 8-9-10	1322
11436	Gruppi e Scuole AR-SM	2379
11434	Gruppi Organizzati CS	716
11116	biglietto merci C.Semplice	131
10022	24H metropolitano RES+2	25
12305	L.11-auto "C"da 4,01 a 4,50 mt	845
12310	L.11-auto "D" oltre metri 4,50	988
11229	72hAerCS-Tpl39,4-CVe6,60	17798
11553	72H R.Venice+aerop.AR online	3887
11099	CartaVenezia bordo TSC	5562
11255	7 days online aerobus AR	22008
11109	Biglietto 72 ore Roll. Venice	185833
12111	C Aut.10 corse 75' TICKET NA	12
7006	ARRIVA VENETO tratta 6	28
11439	Gruppi e scuole online 2viaggi	113
11176	Ev5-Tpl 33,50-C.Ve1,50	789
11236	24hAerAR-Tpl26,9-CVe5,1	1391
11105	24h-Tpl 14,90-Com.Ve5,10	1247921
15114	Carnet CHIOGGIA 10c. TICKET	706
13002	Cav-Trep - S.Marco AR	3623
7018	ARRIVA Extra tr.8-9-10 BORDO	51
11450	Bicicletta "biglietteria"	7395
7005	ARRIVA VENETO tratta 5	8

Continua nella pagina seguente

Tabella A.1 – continua dalla pagina precedente

Id	Nome titolo	Numero di validazioni
11251	24ore online aerobus AR	327

Ringraziamenti

Desidero fare i miei più sinceri ringraziamenti alla Prof.ssa Alessandra Raffaetà per avermi proposto questo interessante progetto che mi ha permesso di mettere alla prova molte delle competenze acquisite in questo percorso di studi. Tuttavia, la voglio ringraziare in misura maggiore per il continuo sostegno, aiuto e disponibilità che mi ha sempre dimostrato e fornito. Ringrazio inoltre il Prof. Claudio Lucchese per i validi consigli e la pazienza con cui mi ha sempre risposto ogni qualvolta avevo dubbi su come affrontare una determinata fase di questa tesi. Ringrazio ovviamente la mia famiglia. I miei genitori che hanno sempre creduto in me e mi hanno fornito tutti gli strumenti per poter affrontare questo percorso con tranquillità e serenità. Mio fratello per tutte le indicazioni e idee che mi ha suggerito per affrontare al meglio la realtà universitaria. E i miei nonni che sono da sempre disponibili ad aiutarmi ogni volta che ho bisogno. Infine, per ultimi ma non per importanza, ringrazio i miei amici più stretti e i miei compagni di corso che hanno avuto un contributo fondamentale per il raggiungimento di questo importante obiettivo, senza di loro questi ultimi tre anni non sarebbero stati così semplici. Vorrei ringraziarli uno ad uno, ma queste poche righe non me lo permettono.