

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

Visualizzazione di Open Data sulla Mobilità: Un Caso di Studio sul Comune di Bologna

Relatore:
Prof. Giovanni Delnevo

Presentata da:
Claudio Verazza

Sessione II
Anno Accademico 2022-2023

*A mio padre Fausto
e mia madre Cristina*

Introduzione

Gli Open Data sono diventati parte fondamentale del lavoro di ricerca dei giorni nostri, in quanto, grazie ad essi, è possibile effettuare studi e avanzamenti tecnologici senza dover dipendere da grandi aziende che producono dati, che potrebbero essere utilizzati solo sotto pagamento di una somma di denaro, una volta sola o tramite sottoscrizioni ricorrenti (abbonamenti). Grazie ad essi, anche persone singole, senza ingenti disponibilità economiche, possono condurre studi e ricerche riguardanti argomenti più o meno conosciuti, allargando il proprio bagaglio culturale e aiutando a farsi domande e darsi delle risposte. Inoltre, scoperte a seguito dell'utilizzo di questi dati possono essere utilizzate per eseguire degli ulteriori studi a riguardo, entrando così a far parte di una catena potenzialmente infinita di scoperte che non fa altro che aiutare la società attuale nella sua crescita, sia tecnologica, sia intellettuale. Questo non si applica solo ai dati, ma anche a come questi vengono usati, ovvero una totale apertura anche dei metodi di ricerca che ha portato a determinate conclusioni. Nonostante molte scoperte vengono effettuate in merito alla casualità, vi sono alcuni casi in cui esse vengono alla luce grazie a dei nuovi ed innovativi procedimenti che potranno venire utilizzati da lì in avanti come punto di riferimento.

Un altro concetto importante sta nel come queste nuove informazioni e scoperte vengano convogliate al pubblico, la cui stragrande maggioranza è interessata, ma non competente o totalmente formata in materia. Dopotutto, l'obiettivo della ricerca è proprio migliorare la vita quotidiana, che sia per una piccola porzione di persone o per tutti, ed è fondamentale che essa venga

compresa e capita da coloro a cui è rivolta. Il processo tecnico e di implementazione che è stato seguito per arrivare alla conclusione a volte può essere tralasciato, tuttavia è importante che i risultati raggiunti vengano visualizzati e presentati in modo da poter essere facilmente capibili, e, anche senza lunghe descrizioni testuali, che diano un'interpretazione di essi a primo impatto, fornendo diverse chiavi di lettura, ad esempio tramite immagini, grafici e colori.

L'obiettivo di questo progetto di tesi è cercare di coniugare i due concetti appena descritti: utilizzare dati liberi e gratuiti per effettuare e produrre un lavoro di ricerca riguardante il traffico della città di Bologna, ed essere visualizzati in modo che possano essere comprensibili da tutti coloro che lo vogliono utilizzare, indipendentemente dalle loro conoscenze pregresse a riguardo. Fornire il giusto livello di dettaglio nella presentazione del lavoro è un altro punto a favore: troppe informazioni visualizzate allo stesso tempo rischiano di avere l'effetto contrario, ovvero la confusione dell'utente, che non saprà che chiave di lettura dare a ciò che vede, e dovrà attivare un processo mentale aggiuntivo per scartare tutti i dati superflui. Anche quest'ultimo è stato considerato come uno scopo importante del progetto implementato, ovvero fornire una visualizzazione abbastanza semplice che possa dare all'utente chiare informazioni per poter essere utilizzata, anche da qualcuno che non vi ha mai avuto a che fare, senza l'ausilio di manuali particolari, non tralasciando però la parte più importante, ovvero non deve essere troppo semplice da ignorare anche dati utili.

Il contenuto del volume di tesi è così organizzato:

- **Contesto e stato dell'arto:** qui si fornirà il contesto iniziale sotto il quale il progetto di tesi è stato implementato. Esso è molto importante per capire su quali concetti ci si è basati, e sulla motivazione del loro utilizzo. In particolare, nella prima parte si tratterà di Open Data, cioè tutti quei dati gratuiti e liberi per essere utilizzati in qualsiasi modo. In questo campo si parlerà non solo di cosa consistono, ma anche come essi vengono utilizzati, in quali ambiti, delle limitazioni

nel loro utilizzo e di tutti quegli aspetti concernenti alla privacy delle persone da cui questi dati sono stati raccolti. Nella seconda parte di questo capitolo si parlerà, invece, di Data Visualization, ovvero quelle tecniche di visualizzazione dei dati, dandogli una chiave di lettura ed elaborandoli di modo che siano di più facile analisi ed interpretazione. Si parlerà, quindi, dei tipi di grafici e diagrammi esistenti, dei significati delle forme e dei colori, ma anche dell'accessibilità ai dati per persone affette dai diversi tipi di disabilità. Infine, si parlerà, con il prospetto del progetto di tesi, di Data Visualization applicata ai dati di mobilità, grazie ai quali è possibile ottenere informazioni riguardanti il traffico e alle zone della città più congestionate, aiutando anche a prevedere eventi futuri partendo da quelli passati;

- **Tecnologie:** in questo secondo capitolo ci si concentrerà sull'ambiente, linguaggi e librerie con i quali è stato implementato il progetto. In particolare, inizialmente si tratterà di HTML e CSS, cioè i linguaggi che consentono di dare struttura e grafica ad una pagina web, JavaScript, che permette di fornire dinamicità e consentire all'utente di interagire liberamente con l'applicazione, e PHP, che consente di generare il contenuto della pagina in base a chi la visualizza, oltre che a fornire un supporto di salvataggio permanente dei dati inseriti dall'utente. Successivamente, si parlerà delle principali librerie utilizzate, ovvero Leaflet, che fornisce la mappa geografica Open Source sulla quale verranno visualizzati i dati elaborati, e Vue, un framework di JavaScript che mira alla creazione di interfacce utente dinamiche in modo più rapido;
- **Progetto:** nel terzo, ed ultimo, capitolo si parlerà del progetto stesso, implementato seguendo i principi discussi. Non si discuterà di come le funzionalità offerte sono state implementate nello specifico: tranne in un caso particolare, in cui si discuterà della velocità di elaborazione dei dati, verranno descritte le possibilità di interazione con la mappa visibili all'utente, ovvero come egli può cambiarne la visualizzazione ed

i dati mostrati grazie ad una serie di controlli, attivabili e disattivabili a piacimento, e dalle informazioni che vi possono essere ricavate.

Indice

1	Contesto e stato dell'arto	11
1.1	Introduzione	11
1.2	Definizione di Open Data	12
1.2.1	Gli utilizzi degli Open Data	13
1.2.2	Linked Open Data	15
1.2.3	Costi degli Open Data	16
1.2.4	Open Data, FAIR Data e Open Source	17
1.2.5	Licenze	18
1.2.6	Sorgenti di Open Data	19
1.2.7	Privacy	20
1.2.8	Il valore degli Open Data	21
1.3	Data Visualization	22
1.3.1	Definizione di Data Visualization	22
1.3.2	Importanza della corretta visualizzazione	22
1.3.3	Tipi di visualizzazione	24
1.3.4	Dati animati	25
1.3.5	Colori	26
1.3.6	Accessibilità	28
1.4	Data Visualization per dati di mobilità	31
1.4.1	Utilità	31
1.4.2	Smart Transportation	33
1.4.3	Produzione dati	33

2	Tecnologie	35
2.1	HTML	35
2.2	CSS	38
2.2.1	Animazioni	40
2.2.2	Visualizzazione per piccoli schermi	41
2.3	JavaScript	41
2.3.1	DOM: Document Object Model	43
2.3.2	JSON	44
2.3.3	AJAX e jQuery	44
2.4	PHP	46
2.4.1	Sessioni	48
2.4.2	Connessione ai database	49
2.5	Leaflet.js	50
2.5.1	La mappa	50
2.5.2	Segnalini, popups e coordinate geografiche	51
2.5.3	Layers	52
2.5.4	Plugins	53
2.5.5	Heatmap.js	54
2.6	Vue.js	54
2.6.1	Single-File Components	55
2.6.2	Binding	56
3	Progetto: Visualizzazione del traffico di Bologna tramite spire	57
3.1	Introduzione	57
3.2	Dati dei sensori	58
3.3	Controlli	59
3.3.1	Giorni di visualizzazione	60
3.3.2	Ore di visualizzazione	61
3.3.3	Rotazione giorni	63
3.3.4	Ottimizzazione della ricerca e aggregazione dei dati	65
3.3.5	Segnalini	69

3.3.6	Animazioni	70
3.3.7	Mappa di calore	74
3.4	Errori di visualizzazione	77

Capitolo 1

Contesto e stato dell'arto

In questo capitolo verrà presentato il contesto in cui il progetto di tesi è stato redatto. In particolare, ci si concentrerà su concetti come Open data e Data visualization, e di come questi svolgono un ruolo di colonna portante nella realizzazione di applicazioni odierne.

1.1 Introduzione

Il primo programma software fu realizzato il 21 giugno 1948 da parte di Tom Kilburn, e aveva solamente il compito di effettuare operazioni matematiche. Infatti, per diversi anni, i computer vennero impiegati per la comunicazione e come strumento di ausilio ai calcoli scientifici, ancora svolti su carta e penna, in quanto non avevano né memoria, né potenza, né tecnologia abbastanza avanzata per poter eseguire programmi complessi. Inoltre, fu impossibile produrli su larga scala a causa dei costi [1].

A mano a mano che il tempo passava, la tecnologia si evolveva con passi da gigante, i programmi divennero sempre più complessi, e si presentò la necessità di utilizzare grandi quantità di dati per poterli sfruttare e migliorare. Questi ultimi possono essere facilmente prodotti ed usati in grandi aziende, ma diventa un problema per aziende piccole e medie, progetti no-profit e in particolare ricerche scientifiche, che non dispongono di tali volumi di dati, e a

causa di ciò, il loro progresso viene notevolmente rallentato. A tal proposito, enti pubblici ed alcune aziende private hanno messo a disposizione di tutti i propri dati gratuitamente, aprendo il mondo degli Open Data [2].

1.2 Definizione di Open Data

Il termine "Open Data" risale solamente al 1995 in una rivista scientifica, ma venne discusso per la prima volta nel 2007, in una riunione riguardante dati climatici e geografici. Ciò significa che è un termine relativamente recente, di circa 20 anni fa [3].

Esso viene definito come "Qualsiasi dato che chiunque può liberamente accedere, fruire, modificare e condividere per qualsiasi scopo" [4]. Ciò significa che i dati devono essere affiancati da una licenza che ne permetta libera modifica e condivisione, e che non limiti il contesto del loro utilizzo, né se li si integra con programmi, né con altri dati. Inoltre, devono essere forniti in modo da essere facilmente modificabili e attingibili. Le uniche limitazioni che essi possono avere sono l'obbligo di citarne l'attribuzione e l'obbligo, ove venissero utilizzati, di rilasciare il prodotto che ne fa uso sotto la loro stessa licenza. Quindi, essi devono essere:

- Legalmente aperti, ovvero dotati di licenze appropriate che ne garantiscono riuso e modifica;
- Tecnicamente aperti, ovvero il cui formato può essere compreso da tutti i calcolatori e non solo quelli proprietari, e che siano accessibili e fruibili mettendoli a disposizione, ad esempio, su un sito internet pubblico. I formati possono includere, ad esempio: XML, JSON, CSV.

In generale, l'obiettivo degli Open Data è lo stesso di ogni movimento Open, ovvero la diffusione della conoscenza libera, senza ostacolarne la fruizione in qualsiasi modo. Occorre far notare che, in ogni caso, tutti i dati distribuiti non comprendono dati sensibili di persone, aziende ed enti pubblici [5].

1.2.1 Gli utilizzi degli Open Data

L'utilizzo principale, e motivo della nascita degli Open Data, è fondamentalmente uno: la ricerca scientifica. Quest'ultima ha potuto fare passi da gigante grazie alla libera fruizione dei dati, che siano usati per esperimenti, statistiche, o predizione di eventi futuri [6]. La ricerca è portata avanti non solo da grandi laboratori con innumerevoli scienziati, ma anche da persone singole o piccoli gruppi indipendenti: senza sufficienti dati, infatti, le loro ricerche sarebbero rese difficili, se non impossibili; non solo a causa della mole di informazioni, ma anche del loro costo di mantenimento. Inoltre, la facilità d'accesso alle grandi banche di dati costituisce una colonna portante per la scienza, in quanto se i dati fossero liberi ma difficili da reperire o analizzare, costituirebbero un altro ostacolo [7].

Le ricerche basate su Open Data hanno un ulteriore vantaggio: possono essere riprodotte e verificate da chiunque. Siccome le sorgenti di studio sono pubbliche, esse possono essere analizzate da potenzialmente milioni di persone e, se vi dovessero essere delle inconsistenze, sarebbero più facilmente trovate e risolte. Questo porta ad un circolo vizioso, in cui la ricerca su dati pubblici porta a dei risultati che saranno a loro volta usati come fonte per ulteriori ricerche, anche in ambiti diversi. Le persone che andranno ad utilizzare questi dati saranno più propense a segnalare incertezze o illeggibilità, che porterà ad una loro correzione e, di conseguenza, all'aumento della loro qualità [8]. Senza riproducibilità, questi studi diventano privi di fondamento, e quindi di credibilità. Più informazioni vengono rese pubbliche, più sono le persone che possono utilizzare e mettere a confronto, e più sono le possibilità di scoperte o sviluppi interessanti. Inoltre, esse possono essere utilizzate per citazioni in documenti scientifici ufficiali: basti solamente pensare che non sarebbe stato possibile redigere questo documento senza l'ausilio di Open Data.

Gli Open Data vengono utilizzati anche per ispirare fiducia, inclusione e trasparenza. Si pensi ad esempio a tutti quei governi ed istituzioni che mettono a disposizione, attraverso i portali ufficiali, statistiche e informazioni riguar-

danti il proprio territorio. Se queste non venissero rilasciate, la popolazione verrebbe lasciata all'oscuro dei problemi del proprio paese, e non potrebbe fare nulla per cercare di migliorarli, soprattutto se essi non riguardassero la scala nazionale (molto spesso trattata dai telegiornali), ma solamente la propria città, o il proprio quartiere. Dopotutto, si potrebbe controbattere che sono proprio i cittadini a produrre questi dati, ed è loro diritto conoscerne i risultati. Offrendoli gratuitamente a tutti, invece, anche se il paese risultasse in declino, ispirerebbe fiducia tra l'istituzione e le persone, e fornirebbe un altro modo a queste ultime di partecipare attivamente alle problematiche del paese piuttosto che farlo in modo passivo; non solo con il proprio lavoro, ma anche dando suggerimenti, proponendo soluzioni o miglioramenti, o richiedendo più informazioni riguardanti il problema. La trasparenza riduce il rischio della corruzione delle persone, e fa sì che le informazioni siano facilmente accessibili ed utilizzabili [5].

Oltre alla ricerca in sé, anche se una persona non è interessata a svolgere attività sperimentali, questi dati possono comunque costituire fonte di studio autonomo per appassionati del settore, senza dover attingere a costosi corsi specifici per ampliare le proprie conoscenze. La scienza infatti nasce proprio così: da un'idea scaturita da una domanda, dalle critiche, dal desiderio di conoscere di più e, di conseguenza, di scoprire di più. Questo fa nascere dibattiti che, grazie ai diversi punti di vista, possono aprire le porte per nuove scoperte. Una popolazione intelligente è più propensa a farsi domande a darsi delle risposte. E proprio da questo nasce il concetto di Open Knowledge, che viene definita come "Cosa gli Open Data diventano quando sono utili, utilizzabili, e vengono utilizzati", che si presagisce come l'ingresso ad un futuro più libero dagli ostacoli della ricerca ed educazione privata.

Gli Open Data hanno contribuito alla standardizzazione dei dati. Questa infatti è fondamentale nel caso in cui i dati vengano utilizzati da persone diverse in contesti diversi, e soprattutto nel caso si vogliano mettere insieme dati appartenenti a dataset differenti. Senza standardizzazione, ogni programma dovrebbe avere anche il compito di implementare un livello logico

aggiuntivo che permetta la ricerca di informazioni utili per ogni tipo di dataset; tuttavia essi possono essere potenzialmente innumerevoli, e diventerebbe ancora più difficile la loro unione per, ad esempio, fini statistici, economici, sociali, o solamente per trarre delle conclusioni dalla loro analisi [9]. Il punto fondamentale, quindi, è l'interoperabilità semantica tra sistemi, ovvero strutturare i dati pubblici in modo che possano essere letti e compresi anche da macchine e programmi non proprietari.

1.2.2 Linked Open Data

Linked Open Data è un tipo di Open Data che ne collega degli altri, in un formato più efficiente da analizzare rispetto che avere tanti dati diversi e separati. Per far sì che questo collegamento sia sempre comprensibile, bisogna standardizzarlo. Per farlo è stato creato un formato chiamato RDF, o *Resource Description Framework*, che modella un grafo a nodi e archi il quale, partendo da una risorsa, la collega a tutte le altre. Ogni elemento di questo grafo è rappresentato da un URI, preferibilmente HTTP, formando la tripletta sorgente-collegamento-destinazione. In questo modo, si può identificare univocamente ogni singolo elemento del grafo, ed è il miglior modo per gli umani per rappresentarlo graficamente e memorizzarlo. In questo modo, viene facilitato molto il processo di aggiunta di nuovi collegamenti, dato che bisogna semplicemente aggiungere un nuovo nodo e il relativo arco, e la loro modifica senza che la struttura venga cambiata. Inoltre, libera dai vincoli dei formati delle risorse collegate, ed è meno costoso in termini di dimensione di un database relazionale, mantenendo la possibilità di eseguire interrogazioni semantiche tramite inferenza, ovvero derivare nuovi dati partendo da quelli già presenti [10].

È bene che il contenuto delle risorse collegate sia non solo leggibile dalle persone (come pagine HTML), ma anche dalle macchine, utilizzando dei formati appositi, come API RESTful o download. Questo non è obbligatorio, ma vi sono dei grandi benefici nel caso venga presentato. Bisogna far notare che

non tutti gli Open Data sono Linked Open Data: infatti, possono essere rilasciati dati che non forniscono alcun collegamento ad altri, oppure quando lo forniscono non sono pubblicamente accessibili (e vengono denominati Linked Data). Proprio in merito a ciò, bisogna sempre assicurarsi di rilasciare una licenza adatta per l'utilizzo dei dati collegati.

1.2.3 Costi degli Open Data

Nonostante gli Open Data siano accessibili e fruibili gratuitamente, non è vero che la loro produzione e mantenimento lo sia altrettanto. Infatti, scienziati, ricercatori, programmatori, analisti, statisti e tutte le altre persone che contribuiscono alla pubblicazione di dati hanno comunque bisogno di mezzi per mantenersi durante la loro produzione, senza contare la necessità di fornire risultati attendibili, comprensibili e preferibilmente privi di errori. In più, se i dati venissero pubblicati dopo essere stati analizzati, elaborati e puliti piuttosto che grezzi, aumenta ancora di più il loro costo. Questi dati andranno poi mantenuti utilizzando delle strutture apposite, come server, librerie o altro, che per funzionare hanno anche loro bisogno di manutenzione, elettricità e personale. Gli Open Data sono gratuiti per i consumatori, ma non per produttori ed editori, ed è il motivo principale che ne frena la diffusione, in quanto richiederebbe troppo lavoro per potenzialmente nessun ricavo economico [11].

La produzione di tutto ciò è sostenuta in gran parte da finanziamenti esterni, come enti pubblici o sponsorizzazioni, dal lavoro di volontari, e da donazioni. La maggior parte del denaro però viene utilizzato per il mantenimento delle infrastrutture, per cui non può costituire una fonte di guadagno stabile ed assicurata nel caso di grandi gruppi di ricerca.

Arrivati ad avere dati pronti e puliti per essere distribuiti, devono essere caricati su piattaforme che ne consentono una facile consultazione e utilizzo. Questo si traduce nell'avere la necessità di costruire siti internet interi per la loro condivisione, che andranno poi aggiornati, migliorati e resi pronti per potenziali successive modifiche. Più dati vengono caricati, più aumentano i

costi di mantenimento delle macchine che li ospitano. Una nota importante va fatta anche per quanto riguarda la privacy. Se venissero manipolati certi tipi di dati, come quelli geospaziali, i ricercatori devono impiegare molto tempo a verificare che, una volta pubblicati, non compromettano alcuna informazione privata, come ad esempio basi militari, abitazioni di persone ad alta visibilità e vulnerabili. Quindi, si devono implementare dei sistemi per l'oscuramento di dati sensibili, che però non vadano ad impattare significativamente l'utilità di quelli messi a disposizione [12].

1.2.4 Open Data, FAIR Data e Open Source

Occorre fare un appunto per quanto riguarda termini simili, ma concettualmente diversi, che si possono sentir nominare quando si discute di questi argomenti.

Open Source viene definito come un software che può essere acceduto, modificato e condiviso da chiunque, offrendo completa trasparenza anche al suo codice sorgente. Entrambi i termini quindi rappresentano qualcosa di gratuitamente fruibile da tutti. Tuttavia, la prima differenza sta nel fatto che i dati, da soli, sono inutili se non vengono utilizzati: essi possono essere coordinate geografiche, statistiche, dati finanziari, atmosferici, e tanto altro. I software invece sono proprio quelle applicazioni che non solo usano, ma producono questi dati, rendendoli effettivamente utili. Si può quindi dire che uno si appoggia sull'altro. L'altra differenza, invece, risiede nella privacy: mentre i codici sorgenti dei programmi Open Source sono visibili e modificabili da tutti, gli Open Data potrebbero prima passare attraverso un filtro che elimina dati sensibili, prima di renderli disponibili.

FAIR Data invece segue sempre i principi della conoscenza libera per tutti, espandendoli ulteriormente. Questo acronimo indica "trovabilità, accessibilità, interoperabilità e riusabilità". Open Data si focalizzano su dati liberi per tutti, mentre FAIR data si concentra anche sul fatto che i essi devono essere facilmente trovabili e compresi dai software. Inoltre fa riferimento ai metadati, che servono a identificare univocamente i dati, e non solo a questi

ultimi. Come ultimo, gli Open Data mirano solamente a rendere i dati pubblici, mentre i principi FAIR menzionano anche la necessità di proteggerne l'accesso [13].

1.2.5 Licenze

Le licenze Open si riferiscono a tutti quei documenti legali che consentono il riutilizzo e ridistribuzione di contenuti senza alcuna restrizione. Esse sono normalmente conosciute come Open Data Commons, Creative Commons e licenze GNU. Tutti i dati che si vogliono rendere gratuitamente fruibili o di pubblico dominio devono essere accompagnati da esse, che stabiliscono cosa si può o non si può fare. Infatti, nonostante siano di libero utilizzo, vi sono numerosi tipi di licenze che possono imporre degli obblighi, o che liberano il fornitore da responsabilità. Ad esempio [14]:

- Obbligo di rilasciare il prodotto che fa uso dei dati sotto la stessa licenza di questi ultimi;
- Obbligo di citare l'attribuzione della fonte di studio e/o del suo autore;
- Obbligo di fornire un collegamento diretto al sito da cui i dati sono stati forniti;
- Libertà o meno di creare dati derivati da quelli forniti;
- Possibilità di utilizzare i dati per scopi commerciali o di lucro;
- Menzioni al fair-use, ovvero l'utilizzo dei contenuti per scopi che non vanno ad infrangere il Copyright di altri.

Di solito, esse offrono una versione riassuntiva, che spiega brevemente com'è possibile utilizzare quei dati, e una versione estesa, che fornisce dettagliatamente tutte le informazioni legali. Alternativamente, il fornitore di questi dati può attribuirgli una qualsiasi licenza da lui redatta, purché rispetti i principi degli Open Data.

Uno dei problemi della definizione di Open Data risiede nel fatto che molti rapporti di studi, ritenuti liberi nell'utilizzo da parte del suo autore, vengono pubblicati senza menzionare alcun tipo di licenza, dando per scontato che entrino a far parte del pubblico dominio, e quindi sotto licenza libera. Questo provoca molte ambiguità, in quanto gli utilizzatori non riescono a capire se quello studio sia fruibile o meno. In pratica, purtroppo, lavori senza di esse sono da considerarsi protetti dal diritto d'autore, e non è possibile accederli, utilizzarli o condividerli liberamente. Per legge, per rendere dei dati effettivamente liberi, serve menzionarlo esplicitamente [15].

1.2.6 Sorgenti di Open Data

A livello nazionale, i dati sotto le licenze Open vengono solitamente distribuiti tramite siti governativi, e molto spesso gli open dataset sono inseriti all'interno di database pubblici che appartengono direttamente allo stato, regione o città di riferimento. Ad esempio Stati Uniti, Olanda, Italia e molti altri paesi forniscono un sito web ufficiale in cui sono disponibili centinaia di migliaia di open dataset relativi alle statistiche del proprio territorio: lavoro, trasporti, educazione, ambiente, salute e molto altro. Questi dati provengono sia da analisi e stime fatte dal governo, sia da aziende private che mettono a disposizione i propri dati caricandoli sui quei portali, facendone previa richiesta [16].

Andando più nello specifico, le aziende stesse sono libere di esporre al pubblico i propri dati. Google, ad esempio, mette a disposizione un'intera sezione del proprio dominio per la diffusione dei suoi dati. Però, in casi dove l'azienda non è conosciuta, bisogna comunque fare attenzione che i dati che espone siano accurati e non falsi (ad esempio per motivi di tornaconto personale).

In generale, la sorgente può essere qualsiasi. Anche una persona che pubblica i dati relativi alla sua vita privata, come la dimensione della sua abitazione, numero di stanze ed età dei suoi inquilini senza richiedere nulla in cambio può essere considerata una sorgente.

1.2.7 Privacy

Con tutte queste informazioni aperte a tutti, un punto di interesse per molti può essere proprio la privacy. Nonostante esse abbiano importanti benefici, c'è il rischio che si condividano dati riguardanti persone che non sono di comune conoscenza, in quanto molto spesso la linea che separa il dato utile dal dato sensibile non è ben definita. C'è da fare molta attenzione a ciò che si rende pubblico, in quanto è un'arma a doppio taglio:

- Dati molto grezzi sono i più apprezzati, in quanto consentono analisi e studi più approfonditi, ma aumentano i rischi di rivelare qualcosa che dovrebbe essere mantenuto nascosto;
- Dati elaborati consentono di mantenere private più informazioni, ma dal lato opposto non ammettono un'analisi approfondita di essi.

Vi è inoltre il pericolo dell'”effetto mosaico”, in cui dati apparentemente non collegati tra, loro presi da diversi dataset e banche dati, se uniti possono portare a dei risultati che vanno ad identificare nello specifico alcuni individui. Considerata la velocità a cui le informazioni vengono diffuse, i danni difficilmente possono essere rimediati.

Purtroppo, in questo caso, non vi è una vera e propria soluzione generale per garantire sufficiente protezione, in quanto nessuno si può immaginare quali informazioni derivate è possibile estrarre da una collezione di informazioni più o meno grande, e a volte non è possibile determinare con chiarezza quali informazioni possono essere considerate private. Si devono, in generale, valutare il rapporto beneficio-rischio nel rilascio dei dati, e minimizzare il secondo fattore. Si dovrebbero memorizzare e condividere meno informazioni personali possibili, anche se in alcuni casi è necessario farlo proprio per mettere in atto dei meccanismi che garantiscono maggiore privacy. Se dati sensibili dovessero venire scoperti, si potrebbe perdere fiducia in colui che li ha messi a disposizione. Ciò però non significa che bisognerebbe smettere di pubblicare dati, in quanto essi pongono comunque dei grandi benefici a livello di società [17].

1.2.8 Il valore degli Open Data

In questo campo si possono intendere vari tipi di valore: economico, sociale e di sviluppo.

A livello economico è difficile dare un valore preciso, perché tante aziende che fanno uso di Open Data non sono ancora pienamente lanciate sul mercato. Tuttavia, nel 2016 è stato stimato che il valore degli Open Data è superiore a quello dei dati forniti sotto pagamento dello 0.5%. Questo perché grazie ad essi è possibile adottare nuove strategie lavorative, conoscere l'andamento del mercato e agire di conseguenza. Gli Open Data aumentano ulteriormente di valore quando è lo stato stesso che ne finanzia e incentiva le iniziative, perché fa nascere nuove aziende interessate sia a produrne sia a farne uso e, dopo che esse guadagnano, anche lo stato lo fa, sotto forma di tasse [18].

A livello sociale, contribuiscono alla conoscenza libera per tutti. Si pensi ad esempio ai dati medici, atmosferici e ambientali, che possono dare all'utilizzatore o lettore un'idea di cosa fare in caso di emergenza e come, ad esempio mostrando le vie di evacuazione o mettendo in guardia delle zone a rischio incendi. Dati a pagamento, anche se forniscono un guadagno quasi istantaneo a chi li fornisce, verranno usati da meno persone e meno frequentemente rispetto a quelli gratuiti. In più, forniscono trasparenza sui dati che vengono raccolti, e le persone si sentono meno tenute all'oscuro delle informazioni che, inconsciamente o non, introducono nella società o, nel caso di siti internet, nel sistema.

A livello di sviluppo, infine, tutti ne beneficiano perché i dati possono essere usati per qualunque scopo. Dati relativamente irrilevanti per uno potrebbero essere fondamentali per la ricerca di un altro. In più, essendo idealmente liberi da licenze restrittive, essi diventano una raccolta messa insieme da tutti coloro che vi contribuiscono [19].

1.3 Data Visualization

In questa sezione si parlerà del concetto di visualizzazione dei dati, ovvero avendo a disposizione le informazioni opportune, il modo in cui visualizzarle nel modo più intuitivo e facilmente memorizzabile possibile.

1.3.1 Definizione di Data Visualization

Per Data Visualization si intende la rappresentazione delle informazioni utilizzando grafici, diagrammi, funzioni e/o animazioni. L'obiettivo è la visualizzazione di informazioni complesse e i loro collegamenti in modo che siano facili da capire. Viene usata per riuscire ad identificare più facilmente sequenze e andamenti di ciò che si sta studiando, e per trasmettere conoscenza a tutte le persone che non sono esperte o tecniche della materia in esame. Anche delle semplici mappe geografiche sono un esempio di ciò, in cui si mettono insieme tutte le informazioni del territorio per darne una vista semplificata a tutti, utilizzando schemi e colori per renderla più riconoscibile [20]. Essa è particolarmente importante quando si ha a che fare con Big Data, per cui l'enorme mole di informazioni renderebbe difficile la sua analisi senza aiuti ulteriori.

1.3.2 Importanza della corretta visualizzazione

Il cervello umano è molto più facilitato a riconoscere colori e forme piuttosto che righe e righe di dati, in quanto nella nostra società certi oggetti vengono già associati a determinati significati. Si pensi alla lampadina accesa, che rappresenta un'idea, il punto esclamativo, associato a pericolo o per richiamare attenzione, o il disco floppy, che nei programmi di scrittura di documenti rappresenta il salvataggio dei dati. Anche senza dare alcuna informazione testuale, queste immagini trasmettono informazioni a chi le vede sul loro significato, se usate nel contesto opportuno. Un obiettivo molto simile lo ricoprono i colori: il rosso, il giallo, il verde, l'azzurro ricoprono dei ruoli importanti, che danno una chiave di lettura aggiuntiva ai dati, richiamando

la gravità o meno di essi [21]. Questo perché le informazioni rappresentate in formato grafico sono più coese e chiare, dando la possibilità di trarre conclusioni senza dover leggere documenti che, per la natura della lingua, potrebbero essere ambigui.

Oltre a ciò, anche la dimensione stessa delle forme, usata correttamente, dà informazioni sulla loro quantità. Delle forme grandi vengono adottate per rappresentare un altrettanto grande numero di dati concentrati in una certa zona o per un certo argomento, mentre al contrario, una forma piccola dà poca importanza o minor numero di dati ad un determinato aspetto del problema in analisi.

Con una corretta visualizzazione, è più facile identificare mode e sequenze di un certo insieme di dati piuttosto che vedere tanti numeri. Si pensi ad esempio ad un'azienda che vuole conoscere quali sono i prodotti più venduti in uno specifico periodo di tempo. Con un grafico a torta o un istogramma, specie se calcolati automaticamente, il risultato sarà più chiaro e più facilmente memorizzabile, piuttosto che semplicemente mostrare un aggregato del numero di prodotti venduti [22]. Inoltre, è possibile identificare più facilmente gli outliers, ovvero quei dati che si discostano totalmente dagli altri e/o che potrebbero causare problemi nella qualità dei prodotti e servizi offerti.

In più, la società è arrivata ad un punto in cui anche l'occhio vuole la sua parte: le persone sono più incentrate ad utilizzare qualcosa di più esteticamente piacevole e semplice, ma magari meno utile, rispetto a elenchi di informazioni dettagliate. Grandi liste di numeri sono noiose e possono intimidire chi legge, dando magari una percezione del problema più complicata o difficile di quello che in realtà è.

Un aspetto importante da ricordare è il fatto che, nonostante in ogni diagramma si possano inserire tutte le informazioni che si vuole, inserire troppe chiavi di lettura e simboli avrà l'effetto opposto, cioè una visualizzazione che confonderà chi la andrà a vedere, in quanto non saprà più su cosa concentrare l'attenzione. Dopotutto, la semplicità è proprio ciò che dovrebbe distinguere

i grafici da enormi tabelle di dati grezzi. Pochi colori e un moderato numero di simboli sono ideali per mantenere una visione chiara e concisa di ciò che si vuole mostrare [20].

Potrebbe inoltre capitare il caso in cui si vuole trarre delle conclusioni o idee non ancora note a partire dai dati. Mettendo questi ultimi in modo da fornire un impatto visuale interessante, è nettamente più facile analizzarli e discuterne insieme ad altre persone, e intraprendere delle decisioni basate su di essi. Questo tipo di visualizzazione solitamente non è raffinata e completa, ma centra in pieno il suo obiettivo: fornire dei punti di vista che, senza di essi, sarebbe difficile notare [23].

1.3.3 Tipi di visualizzazione

Non solo la presenza di diagrammi e disegni è importante, ma è altrettanto fondamentale avere un'idea di cosa si va a mostrare e dell'obiettivo. Infatti, in base a questi, vi sono tipi di grafici più adatti rispetto ad altri, ed il misuse di uno può portare ad una trasmissione di informazioni inefficace. Di seguito si elencano alcuni principali tipi di visualizzazione:

- Grafico a torta: viene utilizzato quando si vuole mostrare vari tipi di dati dividendoli in sezioni, che rappresentano una frazione dell'insieme di cui fanno parte. La dimensione di queste sezioni viene mostrata in percentuale, e la somma di tutte equivale al 100%. Utile soprattutto quando si vuole mettere a confronto un'informazione sia rispetto alle altre, sia rispetto al totale di tutte quelle disponibili.
- Istogramma: rappresenta la distribuzione dei dati in un raggio di valori più o meno grande, utilizzando l'asse delle ascisse per rappresentare i valori assumibili, e l'asse delle ordinate per misurare la quantità di informazioni che assumono un certo valore. Utile quando si mira ad identificare gli outliers.
- Mappa di calore: viene utilizzata soprattutto quando si utilizzano delle coordinate geografiche per mostrare la concentrazione dei dati in una

determinata zona piuttosto che un'altra. Solitamente si ha uno spettro di colori limitato per la visualizzazione, che viene usato per rappresentare i punti di poca concentrazione (di solito il verde), fino ad arrivare ai punti con concentrazione massima (di solito rosso), e i colori intermedi per gli altri valori.

Da questo piccolo elenco salta all'occhio che non c'è una rappresentazione utile per mostrare qualsiasi tipo di dato, ma si deve valutare in base alla situazione. Ad esempio, è inutile utilizzare una mappa di calore per rappresentare i risultati dei lanci di una moneta per capire se è truccata. Invece, un diagramma a torta è ideale, al fine di capire le percentuali di uscita di ciascun risultato.

1.3.4 Dati animati

Uno spazio importante va dedicato alle animazioni. Anche se l'idea di animare dei grafici statici il cui unico compito è mostrare delle informazioni, e quindi è importante che siano il più semplice e chiari possibile, possa sembrare solamente una perdita di tempo e un aumento di probabilità di confondere il lettore, in realtà ha notevoli benefici se usata correttamente. Vedere qualcosa che solitamente è statico prendere vita fa sì che il pubblico ne sia maggiormente attratto, e quindi può convogliare informazioni ad un numero maggiore di persone. Dopotutto, i dati sono inutili se nessuno ne è interessato o non ne trae conoscenza. Questo succede perché le animazioni danno l'impressione che la creazione di questi dati sia stata curata maggiormente, dato che per impostarle serve più tempo [24].

Oltre a ciò, in base a come gli oggetti si muovono, essi suscitano emozioni. Ad esempio, un oggetto che piano piano si ingrandisce e uno che invece fa la stessa cosa ma in maniera molto più rapida possono suscitare un sentimento di calma e di fretta rispettivamente, e fanno sì che tutte le persone che vedono l'animazione se la immaginino allo stesso modo, senza ambiguità. Si riesce a ricordare le informazioni in modo più significativo se ad esse viene aggiunto

un elemento emotivo che colpisce, per cui successivamente si potranno, anche inconsciamente, associare le due cose.

Le animazioni vengono utilizzate per mostrare cambiamenti nel tempo in modo più comprensibile. Si pensi ad una classifica di una gara: piuttosto che avere due immagini della situazione del campionato prima e dopo la gara, dove l'alternarsi da una all'altra avviene istantaneamente, se vi è una lenta animazione che mostra come le posizioni dei vari atleti sono cambiate evita anche che il lettore debba cercare la nuova, e più aggiornata, posizione di una persona, dato che non dovrà fare altro che seguire con gli occhi dove il suo nome si andrà a posizionare alla fine. Lo svantaggio è che chiaramente il convogliamento delle informazioni sarà più lento, tuttavia vi è il grande beneficio che sia di più facile lettura e memorizzazione [25].

Infine, si pensi a quanto sia più semplice identificare qualcosa di dinamico all'interno di un ambiente dove nulla si muove. Richiama immediatamente l'attenzione e permette di processare l'informazione più velocemente.

1.3.5 Colori

Come già precedentemente menzionato, i colori ricoprono un ruolo fondamentale nella creazione di grafici in quanto servono per dare un'ulteriore chiave di lettura. Una buona scelta di colori riesce a porre l'accento sui fattori importanti, mentre una cattiva scelta rischia di confonderli o nasconderli. La ruota di colori RGB è un buon inizio per capire quali sono quelli a nostra disposizione, mostrando quelli simili e complementari. Tuttavia, come fare per capire se quelli che scegliamo sono adatti o no? In generale dipende da cosa si vuole mostrare, in quanto essi hanno intrinsecamente associati dei significati. Presi singolarmente, qualsiasi scelta va bene, tuttavia vi sono delle combinazioni che rendono i dati difficili da leggere. In questo caso, ci vengono in aiuto le cosiddette palette.

Una palette può essere definita come una combinazione di più colori di facile distinzione o lettura, e ve ne sono di tre tipi principali: qualitative, sequenziali e divergenti [26].

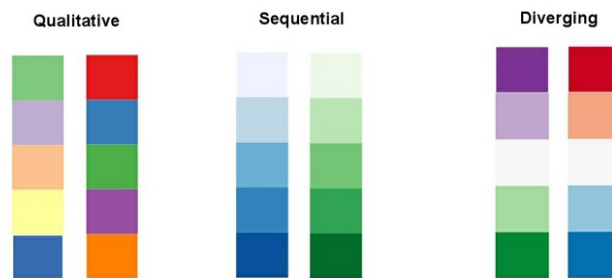


Figura 1.1: I tre tipi di palette di colori. Da sinistra a destra: qualitativa, sequenziale, divergente.

- **Qualitativa**: usata quando le informazioni rappresentano delle categorie senza un ordine particolare, ad esempio per elezioni amministrative, o il sesso delle persone. In questo caso è importante che i colori scelti siano molto diversi tra loro (ad esempio complementari) e non variare semplicemente la saturazione o luminosità nel caso ve ne siano molti.
- **Sequenziale**: le informazioni sono omogenee ma la loro distribuzione o importanza non lo è. Un esempio è il livello di profondità dei mari.
- **Divergente**: le informazioni sono omogenee ma il loro valore può essere minore di zero. Solitamente, il colore centrale è il bianco o il grigio, che rappresentano proprio il valore neutrale. Un esempio è la distribuzione delle temperature in un territorio.

In ogni caso, bisognerebbe cercare di utilizzare colori già di significato noto, come blu-arancione per le temperature, al fine di non creare confusione nel lettore, e tonalità non troppo accese per non affaticare la vista. In più, è necessario scegliere solo pochi colori, altrimenti la visualizzazione risulterà poco chiara. Questi infatti devono essere usati per dare enfasi a determinate caratteristiche e non per abbellire i grafici. Nel caso vi siano più grafici che rappresentano le stesse informazioni ma da punti di vista diversi (ad esempio pesi ed altezze medi di donne e uomini), è importante essere consistenti con

i colori usati.

1.3.6 Accessibilità

L'accessibilità è un aspetto da tenere in considerazione, dato che circa il 4% della popolazione mondiale è daltonico, e queste persone vengono escluse dall'interpretazione di tutte quelle visualizzazioni che fanno uso di colori per rappresentare le informazioni. Per non parlare di persone non vedenti, non udenti, o che soffrono di epilessia. Per tutte queste categorie di persone vi sono altrettante soluzioni, che si andranno ad analizzare di seguito, che permettono di far utilizzo delle visualizzazioni dei dati in maniera alternativa.

Daltonismo

Riguardo le persone daltoniche, dato che vi sono diversi tipi di daltonismo, bisognerebbe evitare delle combinazioni di colori che sono comunemente problematiche, come rosso-verde, marrone-verde, e verde-blu. Fortunatamente le combinazioni adatte sono state già esplorate da tante persone, ed esistono degli strumenti su Internet per capire, scelti i colori, come sono visti dalle persone daltoniche, ed avere un'idea se le informazioni sono ben visibili per tutti. In alternativa, si deve dare la possibilità di poter cambiare la colorazione dei dati in vari modi, di cui almeno uno sia compatibile per ogni persona. Si potrebbe, ad esempio, far scegliere la tinta tramite un menù a scelta esclusiva (radio button o menù a tendina), ricordandosi comunque che il problema potrebbe sussistere nel caso in cui i colori da distinguere sono molti. Questa seconda alternativa è più indicata per chi è affetto da acromatopsia totale, ovvero ha una visione monocromatica del mondo (in bianco e nero). Un'ulteriore possibilità da considerare, utilizzabile in concomitanza con quelle già elencate, è quella, soprattutto nei grafici dove i dati sono rappresentati da pallini o linee, di variare le forme e modello dei dati, usando

anche quadrati, triangoli, stelle e così via [27]. Quindi, in generale, non basare la visualizzazione solo sui colori, ma anche sulle forme.

Cecità

Per le persone non vedenti, invece, le soluzioni elencate non funzionano, in quanto non riusciranno a distinguere né forme né colori. Questo si applica non solo ai dati, ma a tutte le pagine Internet e come queste possano essere accedute e navigate. Esse infatti si basano su tre dimensioni: struttura, navigazione, e descrizione. La struttura detta come le informazioni vengono impaginate, ovvero come sono disposti i pannelli e le figure. Per navigazione si intende invece la disposizione dei menù, come scorrere da una pagina all'altra, e come cambiare livello di dettaglio. La descrizione invece è come le informazioni della pagina vengono presentate all'utente. In tutti questi casi viene in aiuto il supporto audio: tramite dei dispositivi appositi o un'impostazione del proprio computer, è possibile far sì che, tramite la conversione da testo a parlato (text-to-speech), si possa accedere alle informazioni normalmente. Per quanto riguarda le immagini e tutto ciò che non è testo, invece, se rappresenta una parte importante delle informazioni, va opportunamente accompagnato da una didascalia che lo descriva. Ciò che viene usato per abbellire la pagina può essere ignorato, ma l'importante è che il contenuto principale sia fruibile da tutti [28]. In aiuto, il linguaggio HTML offre un'opzione per le immagini e contenuti visuali che, se impostata, permette ai lettori di schermo di descriverli.

Difficoltà motorie

Riguardo agli utenti con difficoltà motorie, il primo passo da fare è rendere tutto il contenuto e funzionalità accessibili utilizzando solo la tastiera, senza avere la necessità di premere tasti a tempo se non dove il contenuto è mirato a farlo. Ad esempio, per cambiare il tipo di visualizzazione dei dati alterandone

la posizione, tipo, animazioni e così via tramite un menù, si deve progettare la possibilità di poterlo fare non solo con il puntatore del mouse. Questo non significa però che non si deve dare la possibilità ad altri dispositivi di input di operare, ma che semplicemente deve essere possibile fare tutte le operazioni possibili anche utilizzando solo la tastiera [29].

Sordità

Nonostante la maggior parte dei contenuti sia solo visuale, come appunto grafici, diagrammi e rappresentazione dei colori per le informazioni, si deve pensare a metodi di comunicazione alternativi nel caso essi siano video o audio. La parte fondamentale è la scelta di immagini ed inquadrature che siano autoesplicative, senza la necessità che per essere comprensibili devono essere accompagnate dal testo. Anche se non parlano, queste possono comunque essere interpretate e possono suscitare emozioni in chi le guarda, ed a supporto di esse devono essere aggiunti testi e sottotitoli. L'ordine, la velocità, i colori e i soggetti principali sono ciò che viene convogliato a primo impatto, in quanto sono immediati e, muovendosi, attraggono l'attenzione. Certo, i testi restano comunque ciò che le persone non vedenti andranno poi a leggere per avere un'interpretazione univoca di ciò che viene raccontato nell'audio o video, ma bisogna ricordare che, come precedentemente menzionato, è più facile essere attirati da qualcosa che si muove (il video) in un contesto in cui tutto è fermo (testo). Inoltre, si può dare più importanza ad un certo aspetto anche in base al tempo usato dal video per documentarlo [30].

Epilessia

Circa l'1% della popolazione mondiale soffre di epilessia, e per questo motivo non è possibile ignorarla. A tal proposito è necessario visualizzare i dati di modo che non scatenino delle reazioni avverse in chi li guarda. Questo si traduce nel non far sì che essi non lampeggino (flash) per più di tre volte al secondo, e che le animazioni di movimento generate dall'interazione dell'utente con i dati possono essere disattivate se non strettamente necessarie

al convogliamento delle informazioni [31]. Molto spesso, infatti, si possono notare delle avvertenze all'inizio di pagine, video e programmi che contengono dati che lampeggiano, che avvisano proprio che è possibile che alcuni utenti possano avere delle reazioni indesiderate dalla loro visione, ed è bene farlo anche se i dati da mostrare rispettano le regole indicate dal WCAG.

1.4 Data Visualization per dati di mobilità

Nel contesto del progetto di tesi, è utile parlare della visualizzazione dei dati nel mondo della mobilità. Questo è diventato un aspetto più rilevante soprattutto negli ultimi anni a causa della diffusione e ampliamento delle Smart City, ovvero quelle città tecnologicamente avanzate che usano il suolo pubblico per raccogliere dati, analizzarli ed impiegarli per migliorare la vita dei propri cittadini. Anche se il termine include il raccoglimento di molti tipi di dati (e tutt'ora non pienamente definito) [32], in questo contesto ci si limiterà a parlare di dati relativi alla viabilità del traffico.

1.4.1 Utilità

Come tutti gli strumenti di visualizzazione di dati, l'utilità principale è la ricerca di mode e sequenze ricorrenti del passaggio dei veicoli, nonché della segnalazione degli eventuali outliers. Queste ricerche possono essere effettuate per diversi periodi di tempo: ore, giorni, settimane, mesi, anni, possono essere usati per dare una stima delle statistiche dei veicoli che circolano in quei momenti. Ciò si traduce nell'aiuto allo smaltimento della congestione del traffico: il numero di automobili iniettate nel mercato ha un aumento rapido, mentre l'innovazione dell'industria dei trasporti non segue di pari passo. Il traffico, in generale, è generato da tre concetti chiave:

- Eventi inaspettati: ad esempio incidenti e condizioni atmosferiche come pioggia e vento;
- Eventi programmati: come manifestazioni, cantieri o concerti;

- Capacità fisica dell'arto stradale.

Il raccoglimento dei dati serve per aiutare la città nello smaltimento del traffico, ad esempio fornendo percorsi alternativi tramite segnali stradali, e ai conducenti per valutare se conviene transitare per una determinata zona piuttosto che un'altra, evitando lunghe code. Inoltre, serve ai governi per riuscire a meglio identificare i tratti di strada problematici, ad esempio quali sono particolarmente soggetti a congestioni e valutare se investire denaro pubblico per migliorarli e potenziarli.

Soprattutto se l'analisi viene effettuata a tempo reale, sono stati studiati dei metodi, che possono essere implementati nei sistemi tramite veri e propri algoritmi, per il redirezionamento del traffico, traducendo il manto stradale come se fosse un grafo con nodi e archi. Questi metodi possono essere utilizzati dai veicoli stessi, spesso dotati di dispositivi GPS o navigatori di vari tipi, che riusciranno, forniti i dati di partenza, a visualizzare il percorso alternativo più adatto. Non solo, ma questa scelta potrà essere fatta anche in base al tipo di veicolo guidato [33].

Il passaggio di più o meno veicoli, e il tipo di strada su cui transitano, può dare una stima riguardo l'inquinamento ambientale che essi provocano. L'emissione di biossido di carbonio (CO_2), infatti, non è uguale ovunque a parità di automobili che circolano: dipende anche dall'inclinazione della strada, dal tipo di manto, nonché da ciò che vi è attorno, ad esempio la presenza di boschi e foreste. La qualità dell'aria è un aspetto fondamentale nella decisione di vari aspetti della vita quotidiana. Se un'area è inquinata a causa del traffico, infatti, risulterà meno probabile che le persone ci vadano a vivere. Di conseguenza, potrebbe portare a pensare a soluzioni che permettano di renderla più pulita, come l'installazione di zone a traffico limitato (ZTL), o il divieto per certe tipologie di veicoli di circolarvi. Oppure, sfruttando proprio i dati raccolti, pensare a percorsi alternativi che distribuiscano in un'area più grande l'inquinamento.

Anche la sicurezza è un aspetto da non sottovalutare. Non tutte le strade sono sicure allo stesso modo: un piccolo vicolo di un centro storico è meno

soggetto, in linea di massima, ad incidenti stradali rispetto ad un'autostrada o superstrada, anche se venissero rispettati tutti i limiti di velocità e le norme standard di circolazione. La visualizzazione di questi dati permette di dare una stima sulla pericolosità del percorso, in termini di incidenti e incidenti fatali, ed agire di conseguenza.

1.4.2 Smart Transportation

Il concetto di Smart City, unito alla raccolta e visualizzazione di dati relativi al traffico, fa nascere un altro concetto: lo Smart Transportation, o Smart Mobility. Esso permette di risolvere molti problemi riguardanti la viabilità in modo automatico, analizzando i dati in tempo reale. Questo è nato dal fatto che ormai qualsiasi persona o veicolo possiede un dispositivo mobile capace di inviare e ricevere informazioni.

Non è raro in tempi odierni, soprattutto in città popolate, di dover cercare parcheggio per diverso tempo in quanto tutti i posti sono pieni, oppure di dover aspettare con impazienza un autobus in ritardo perché bloccato. Grazie a questa tecnologia è possibile visualizzare in tempo reale una stima di quanti minuti i trasporti pubblici subiranno ritardi, oppure la probabilità di trovare un parcheggio libero considerato il luogo, giorno della settimana e concentrazione di automobili nella zona, sempre che essi non siano già dotati di sensori appositi per farlo. Oppure, grazie a questi dati e alle possibili segnalazioni manuali dei guidatori, identificare gli incidenti ancor prima che le autorità pubbliche siano intervenute per evitare lunghe code [34].

1.4.3 Produzione dati

Nel mondo della viabilità, i dati possono essere introdotti nei sistemi in vari modi. Di seguito se ne elencano alcuni esempi:

- Direttamente dalle automobili, in modo indipendente, che inviano la propria posizione ai server opportuni tramite dei dispositivi IoT installati su di esse (come GPS);

- Da torri di ricezione, che richiedono informazioni alle auto, entro un determinato raggio attorno a loro, riguardo alla propria posizione;
- Da sensori posizionati direttamente sopra o sotto il manto stradale (come fotocellule o sensori a pressione) che, al passaggio di un'auto, ne registrano la presenza e inviano periodicamente i dati ad un server centrale;
- Da telecamere che impiegano tecnologie di visione artificiale.

I sensori stradali sono solitamente poco impiegati a causa del loro costo di dispiegamento e manutenzione, mentre ormai sono molto diffusi i sensori mobili che inviano dati anonimi. Queste informazioni possono essere di varia natura, come posizione, velocità e direzione di marcia, da cui possono esserne derivate delle altre, come appunto la congestione di un tratto e i percorsi alternativi possibili. I sensori mobili più comuni sono proprio i telefoni cellulari dei guidatori: una buona parte di essi ha sempre il sensore GPS attivato, per cui è facile per le applicazioni estrapolarne i dati. In più, le automobili semplici o vecchie non dispongono di navigatori o dispositivi di posizione integrati, e anche se possono essere normalmente utilizzati, non possono tutt'ora considerarsi come il mezzo primario da cui attingere dati [35].

Capitolo 2

Tecnologie

In questo capitolo si parlerà delle principali tecnologie e librerie utilizzate per implementare il progetto di tesi. Esse sono tutte pubbliche e da considerarsi liberamente utilizzabili da chiunque, seguendo i principi Open discussi nel capitolo precedente. Si forniranno informazioni riguardo al loro funzionamento e come possono essere impiegate in contesti anche al di fuori di questa tesi. Il progetto è stato svolto completamente in PHP, Javascript, HTML e CSS, di cui se ne parlerà nelle sezioni seguenti.

2.1 HTML

HTML, acronimo di *HyperText Markup Language*, è un linguaggio di Markup, ovvero ha la funzione di costruire la struttura di base di una pagina web e del suo contenuto per poter essere facilmente leggibile dagli umani, e, potenzialmente, anche da altri calcolatori. I file di questo tipo solitamente hanno estensione *.html* o *.htm*. È un linguaggio interpretato, ciò significa che quando il browser riceve il file che contiene la pagina da visualizzare, ne viene letta una riga per volta ed interpretata sul momento, al contrario dei linguaggi compilati (o di programmazione), il cui codice sorgente deve essere prima trasformato in un insieme di file binari, tramite un processo di compilazione, prima di poter effettivamente rendere operativo il programma che

ne fa uso. Un linguaggio interpretato come questo, generalmente, richiede più tempo di elaborazione rispetto ad un programma già compilato, tuttavia permette di saltare completamente la parte di compilazione, e la lettura di un file HTML viene delegata interamente al browser. Inoltre è un linguaggio di markup descrittivo, ovvero descrive la struttura del documento, separando la parte strutturale da quella grafica e logica. Al momento, la versione del linguaggio più aggiornata è la 5.2, ed è quella che si terrà come riferimento sia durante questo capitolo, sia durante l'illustrazione del progetto. È un linguaggio lato client, ovvero l'interpretazione dei file HTML avviene direttamente sul calcolatore dell'utente visitatore della pagina web senza nessun onere ulteriore per il server a cui si fa richiesta della pagina, a parte il suo invio.

Un documento HTML è formato da una moltitudine di elementi HTML, che sono tutti quei costrutti che permettono di dare forma alla pagina. Un elemento HTML è formato da tre parti:

- Tag di apertura, che è il nome dell'elemento da visualizzare, oltre il quale il contenuto della pagina subirà dei cambiamenti strutturali;
- Contenuto, cioè ciò che risiede all'interno dell'elemento, che può essere testo, audio, video, o altro;
- Tag di chiusura, ovvero il limite dopo il quale le modifiche strutturali cessano di avere effetto.

I tag di apertura e chiusura sono sempre racchiusi da parentesi angolari, per effettivamente separare il contenuto dal tag. Inoltre, un tag di chiusura deve essere accompagnato da uno slash (/) dopo la parentesi angolare aperta, per segnalare che è il punto in cui si concludono gli effetti del tag di apertura precedente e non è l'inizio di un elemento innestato al primo. Ogni tag può (ed in particolari casi deve) avere degli attributi, ovvero delle opzioni in più che ne permettono un'ulteriore modifica del comportamento o della visualizzazione grafica. Un esempio può essere l'aggiunta di un attributo ad

un campo di inserimento testuale per disattivarlo, ovvero impedire all'utente di cambiare il testo visualizzato.

```
<section name="traffico"> Esempio di elemento HTML. </section>
```

Figura 2.1: Un esempio di un elemento HTML.

In figura 2.1, è possibile vedere un comune elemento di HTML: *section* è il nome del tag, *name* è il nome dell'attributo, *traffico* è il valore dell'attributo, ed il testo che è compreso tra i due tag di apertura e chiusura è il contenuto. Se ci si dimentica di specificare dove un elemento finisce, non verrà lanciato nessun errore: l'interprete del browser proverà comunque a visualizzare gli elementi, tuttavia vi saranno dei risultati non prevedibili. Inoltre, per lo stesso motivo, è importante che, quando si innestano elementi dentro ad altri, non si sovrappongano tra loro e sia chiaramente distinguibile quale elemento sia dentro all'altro: le tag di chiusura vanno poste in ordine inverso a quelle di apertura.

Possono esservi degli elementi che, però, non richiedono alcun tag di chiusura: questi sono chiamati *elementi void* e sono caratterizzati da avere lo slash di chiusura alla fine del tag di apertura piuttosto che averne uno separato. Questi elementi sono, quindi, privi di contenuto, e possiedono solo nome ed attributi che, per questi casi specifici, sono obbligatori.

```
<link rel="stylesheet" href="./css/main.css"/>
```

Figura 2.2: Un esempio di un elemento void HTML senza tag di chiusura, ma con uno slash finale.

In figura 2.2 vi è un tipico caso di elemento void, ovvero quello che serve per includere un file esterno all'interno di un documento HTML. In questo caso, l'attributo *href*, che indica il percorso del file da includere, non è opzionale, ma obbligatorio. Questi tipi di elementi si dicono anche *self closing* (che si chiudono da soli), ovvero hanno bisogno solamente dello slash finale per essere chiusi.

Il linguaggio HTML ha già una vasta serie di tag che offre ed è in grado di riconoscere nativamente, come quelli già esposti nelle immagini, che permettono di definire la struttura di una pagina web in modo ottimale. Si potrebbe però avere la necessità di utilizzarne degli altri, per dare una forma ancora più specifica al documento. Anche questo è possibile utilizzando un altro linguaggio di markup chiamato XML, acronimo di *eXtensible Markup Language*, che permette di definire elementi con nomi ed attributi personalizzati, che modificano il contenuto nella maniera desiderata. La sintassi di base di XML è standardizzata, quindi al trasferimento di dati tra macchine diverse, queste riusciranno comunque a cercarli, trovarli e comprenderli. In ogni caso, questi elementi dovranno sempre seguire la semantica di HTML, ovvero chiusi, innestati e strutturati correttamente.

2.2 CSS

Il linguaggio HTML viene usato solamente per strutturare la pagina web, come ad esempio dividerla in sezioni e paragrafi, impostare i metadati, usati dai motori di ricerca per trovare la pagina web su Internet, i titoli e le immagini, ed inserire l'effettivo corpo della pagina con il testo. Tuttavia, essa risulterà difficile da leggere con facilità, perché può capitare che tutto il contenuto sia uno sotto l'altro, compresi i menù, le immagini, i video, e così via. Senza delle regole precise che dettano come questi devono essere visualizzati, se ne delega il compito al browser. Tuttavia, non è detto che tutti i browser mostrino il contenuto tutti allo stesso modo: qualcuno può avere un font diverso, dimensione dei caratteri, colori, contrasti, e posizioni dei vari elementi che non sempre uguali. Così facendo risulta anche difficile riuscire a ritrovare lo stesso contenuto su due browser differenti.

Il linguaggio CSS, o *Cascading Style Sheets*, viene in soccorso proprio per questo obiettivo: se HTML gestisce la struttura, CSS ne dirige la presentazione grafica. Diversamente da HTML, è un linguaggio di stile e non di markup, ovvero serve per dare un impatto grafico diverso agli elementi delle pagine

web. È standardizzato dal WCAG, il che fornisce la possibilità di visualizzare la pagina sempre allo stesso modo, indipendentemente dal browser che si utilizza, oltre che a riuscire a realizzare un documento la cui impostazione garantisce un buon ritmo di lettura e linearità del suo contenuto. I file che definiscono questi aspetti hanno estensione *.css*. L'ultima versione di CSS è CSS3, tuttavia questo concetto è diventato con il tempo sempre più irrilevante: ciò che è importante, invece, è sapere se una determinata funzionalità è supportata in un certo browser, in quanto tutti supportano i fogli di stile, ma non allo stesso modo. La versione, al giorno d'oggi, viene solo utilizzata per capire quanto tempo fa una certa funzionalità è stata introdotta. Come l'HTML, è un linguaggio lato client.

È possibile variare la presentazione di una pagina web cambiandone la visualizzazione dei suoi elementi in modo più o meno specifico.

```
html, body{  
  height: 100%;  
  font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;  
  background-color: ■ antiquewhite;  
}
```

Figura 2.3: Un esempio di alcuni attributi che è possibile dare agli elementi della pagina per cambiarne la presentazione.

Nella figura 2.3, si può vedere un esempio del cambiamento di presentazione di alcuni elementi. I primi due nomi, e unici fuori dalle parentesi graffe, indicano quali sono gli elementi della pagina a cui quello stile è rivolto. Nel caso della figura, i cambiamenti sono rivolti a tutta la pagina HTML. All'interno delle parentesi graffe, separati da un punto e virgola e un'andata a capo, vi sono gli aspetti grafici che devono essere cambiati, in questo caso la grandezza della pagina, il font ed il colore dello sfondo. La quantità di stili che è possibile modificare è innumerevole, come anche i margini destri e sinistri, l'interlinea tra le righe di testo, o la trasparenza dei vari elementi. Tutto ciò, oltre che a migliorare l'aspetto estetico per renderlo più piacevole ed elegante da vedere, ha anche un altro vantaggio: il miglioramento dell'accessibilità, ad esempio ingrandendo la dimensione dei testi per le persone

con bassa acutezza visiva, o il cambiamento dei colori visualizzati per persone daltoniche.

La grafica di visualizzazione, scritta in CSS, può essere integrata ad un documento HTML in tre modi:

- In linea, inserendo l'attributo *style* all'interno di un tag;
- Internamente, utilizzando l'elemento `<style>` all'interno di un file HTML;
- Esternamente, creando un file *.css* a parte.

Nella stragrande maggioranza dei casi è consigliato l'utilizzo dei file esterni, in quanto separano effettivamente la struttura dalla grafica, oltre che, in tal modo, possono essere inclusi in più documenti, senza dover eseguire un processo di copia ed incolla del codice in ognuno, abituando così l'utente ad uno stile di visualizzazione uniforme in tutte le pagine web.

2.2.1 Animazioni

Le pagine web più semplici il cui unico scopo è quello di mostrare testi ed immagini, come i giornali, non hanno bisogno di stili particolarmente elaborati, e l'importante è che la lettura delle informazioni sia facilitata, come dal loro posizionamento, dal font, dai colori, e così via. Molto spesso, però, nelle pagine web più elaborate, sono presenti delle animazioni: queste possono comprendere una foto che quando ci si sposta il puntatore sopra si allarga, oppure un elemento della pagina che scompare e ricompare lentamente in un ciclo, o oggetti che si muovono all'interno di essa.

Questi comportamenti vengono resi possibili proprio grazie al linguaggio CSS, che permette di realizzare queste animazioni utilizzando delle parole chiave. Queste possono poi essere applicate agli elementi della pagina, dandole un nome e specificandolo negli attributi che definiscono come quell'elemento andrà visualizzato.

Le animazioni servono soprattutto per rendere una pagina più dinamica e piacevole da vedere: a patto che non siano estremamente invasive, danno l'idea di una pagina più curata nei dettagli e possono attirare l'attenzione su determinate zone della pagina piuttosto che su altre, cambiandone l'importanza senza menzionarlo esplicitamente. È un aspetto a cui si dà erroneamente poca importanza nella progettazione, a cui però l'utente finale presterà, anche inconsciamente, molta attenzione.

2.2.2 Visualizzazione per piccoli schermi

Nonostante le pagine web vengano realizzate su dei calcolatori che dispongono di grandi schermi, esse vengono, per una decisa maggioranza, accedute tramite dispositivi mobili, come telefoni e tablet, i cui schermi sono decisamente più piccoli. Ciò significa che anche se nel primo caso la grafica è adatta per una visualizzazione chiara e piacevole, nel secondo caso i testi potrebbero risultare troppo piccoli, rendendo difficile anche solamente accedere ad altre pagine tramite i menù posti sopra o a lato della pagina. Proprio per questo, in CSS, vengono solitamente realizzati degli stili ulteriori che servono ai siti web per essere navigati facilmente anche da questi dispositivi più piccoli. Si pensi, ad esempio, ai pulsanti che, una volta toccati, mostrano un menù che si sovrappone alla pagina principale per metà o più. Per schermi grandi questo non è necessario, in quanto si ha spazio a sufficienza per la visualizzazione dell'intero menù in una piccola parte del monitor, ma è fondamentale per i telefoni, che non riuscirebbero a visualizzarlo abbastanza grande per poter essere utilizzabile.

2.3 JavaScript

Una pagina che utilizza solamente HTML e CSS si può considerare, in un certo senso, una pagina statica: serve solo per mostrare del testo, indipendentemente da ciò che l'utente vuole fare. È ottimo per pagine semplici il cui unico scopo è convogliare informazioni, come quotidiani, riviste scientifiche,

dizionari, enciclopedie, eccetera. Tuttavia, non è sempre così: si potrebbe avere la necessità che, cliccando su un bottone, diventino visibili delle funzionalità che prima non lo erano. E questo, utilizzando solamente i due linguaggi appena descritti, è qualcosa che non è possibile fare. Serve un ulteriore pezzo che dia ulteriore dinamicità alla pagina.

Ed è qui che entra in gioco *JavaScript*, spesso abbreviato in JS. È il linguaggio che gestisce la logica degli elementi della pagina, e che cosa devono fare quando subiscono delle interazioni da parte dell'utente. È un linguaggio di scrittura (o di *scripting*) generalmente interpretato, ma in alcuni casi può anche essere compilato (come nel caso di NodeJS). Come tutti quelli già descritti, è anch'esso lato client ed è debolmente tipizzato, ovvero non è necessario dichiarare il tipo delle variabili quando vengono create. Inoltre, insieme ad HTML e CSS, costituisce il terzo ed ultimo elemento che viene considerato il cuore delle pagine web. Infatti, proprio in virtù di ciò, è standardizzato attraverso tutti i browser, anche se, come con CSS, alcuni possono supportare delle funzionalità che altri non supportano. Viene utilizzato in quasi la totalità delle pagine web presenti su Internet per dare quell'aspetto di dinamicità che, senza di esso, mancherebbe. Inoltre, esso non è associato a nessun elemento specifico della pagina: elementi di natura diversa possono essere associati allo stesso codice JavaScript

Esso può essere incluso all'interno di una pagina HTML in due modi:

- Internamente, utilizzando il tag `<script>` all'interno della pagina, e inserendo, nel contenuto, il codice;
- Esternamente, creando un file a parte con estensione `.js`, ed includendolo nella pagina web aggiungendo l'attributo `src` all'interno del tag `<script>`.

Similmente ai file CSS, è sempre altamente consigliato il secondo modo di inclusione, sempre per gli stessi motivi. A differenza degli altri linguaggi, i browser danno la possibilità di disattivare l'esecuzione degli script JavaScript. Questo succede perché essi sono fonte piuttosto comune di malware,

cioè codici dannosi che possono far trapelare informazioni personali dell'utente, o far forzatamente scaricare dei file sospetti che, una volta aperti, portano all'installazione di programmi altrettanto dannosi. Tuttavia, le pagine altamente dinamiche che dipendono da questi codici cesseranno di funzionare, parzialmente o interamente, fino a che non vengono riattivati.

JavaScript è un linguaggio object-oriented, ovvero supporta l'esistenza e dichiarazione di oggetti, che al loro interno possono avere delle proprietà con un loro valore e, potenzialmente, metodi. Avendo anche dei tipi primitivi (come stringhe e numeri) non lo porta ad essere object-based. Una particolarità di questo linguaggio sta nell'aggiunta di un ulteriore tipo primitivo, *undefined*, il cui unico valore assumibile è proprio undefined. Una variabile dichiarata ma non assegnata avrà tipo e valore undefined, perciò provare ad utilizzarla, come stamparne a schermo il contenuto, non causerà errori. Questo differisce dalla maggior parte dei linguaggi di programmazione, nei quali una variabile non inizializzata avrà valore nullo, con conseguente ed inevitabile lancio di un'eccezione se si prova ad utilizzarla. In ogni caso, per controllare se una variabile non è definita, occorre sempre far riferimento al suo tipo piuttosto che al contenuto.

2.3.1 DOM: Document Object Model

JavaScript fa un forte utilizzo del DOM del documento, acronimo di *Document Object Model*. Esso è un'interfaccia che rappresenta come la pagina web può essere acceduta, per poter essere manipolata da un qualsiasi linguaggio di scrittura (ce ne sono diversi, come il Perl, ma, in questo contesto, ci si concentrerà solo su JavaScript). Si può immaginare il DOM come un albero: in testa vi è il documento stesso, il cui livello sottostante è la pagina HTML, che si divide nei rami principali Head e Body, che a loro volta avranno dei rami e così via, fino ad arrivare alle foglie dell'albero, ovvero quegli elementi che non hanno figli. Per accedere ad un particolare elemento della pagina, quindi, si utilizza il DOM, percorrendo la sua struttura in discesa fino

ad arrivare a selezionare l'elemento desiderato. In questo modo, JavaScript può, potenzialmente, alterare del tutto sia la struttura della pagina web, sia la visualizzazione grafica, cambiando ciò che è già presente, ma anche aggiungendo o togliendo altri elementi e i relativi attributi e comportamenti. Il DOM è stato standardizzato, per cui la sua logica e modi di utilizzo sono gli stessi su tutti i browser.

2.3.2 JSON

Un altro elemento che JavaScript usa, ed è particolarmente rilevante, è il JSON. Acronimo di *JavaScript Object Notation*, è un formato per la serializzazione e condivisione di dati. È un formato fondato sulla sintassi JavaScript, anche se alcune sue regole cambiano. Esso è utilizzato non solo da questo linguaggio, ma anche da altri programmi, per condividere efficientemente le proprie informazioni (un esempio è MongoDB, un database non relazionale le cui tabelle interne sono tutte in JSON). È molto utile in quanto un oggetto JSON è facilmente deserializzabile in una stringa o oggetto JavaScript e viceversa. A parte la sintassi stessa, i suoi elementi non devono seguire alcuna struttura particolare: possono essere di qualsiasi tipo, compresi altri oggetti JSON, e questi ultimi possono anche avere lunghezze e campi differenti gli uni dagli altri. È un formato estremamente modellabile, che può essere acceduto sia come stringa, sia come vettore normale, sia come vettore associativo.

2.3.3 AJAX e jQuery

JavaScript riesce ad integrare l'aspetto dinamico all'interno di uno statico. Questo avviene interamente all'interno del browser dell'utente, in quanto la pagina ha già le informazioni pronte, ma le visualizza e si comporta in base a come egli vi interagisce. Se l'interazione ne comporta la modifica, ed il contenuto da mostrare non è già presente al suo interno, essa viene ricaricata con il nuovo, ed aggiornato, contenuto. Per quanto sia indubbiamente

comodo, da questo derivano degli svantaggi: la pagina deve essere ricaricata da zero, per cui il web server, ovvero colui a cui si richiedono i dati relativi alla pagina da visualizzare, la deve rinviare per intero, compresi tutti i contenuti che non hanno subito cambiamenti, comportando un inutile spreco di trasferimento di dati. In più, l'utente vedrà l'effetto della pagina che viene ricaricata, ovvero verrà riportato in cima ad essa e un lampeggiamento della stessa dovuto alle informazioni non ancora pienamente caricate.

A questo proposito è stata creata un insieme di tecnologie che prende il nome di AJAX, acronimo di *Asynchronous Javascript And XML*. La sua funzionalità principale è quella di poter richiedere al web server l'invio non di intere pagine, bensì solo di alcuni frammenti di esse che, a seguito delle interazioni dell'utente, hanno subito cambiamenti, e di poterli visualizzare senza che la pagina web venga ricaricata per intero, alterando solo la parte che è stata modificata. Questo porta, oltre che a un minor trasferimento di dati inutili, anche il fatto che c'è bisogno di meno memoria per ciò che è accaduto precedentemente alla pagina. Se una parte di essa fu precedentemente modificata, un cambiamento ulteriore ad una parte diversa, dato che non viene ricaricata da zero, la farà rimanere invariata, senza doversi preoccupare di mantenere in memoria tutte le interazioni che l'utente ha eseguito che ha portato a una determinata visualizzazione.

Queste richieste possono essere eseguite con qualsiasi metodo HTTP, meglio noti come *CRUD*. Tuttavia, nella maggior parte dei casi, vengono usati:

- HTTP GET: le informazioni ricevute sono visibili all'utente. In alcuni casi, viene modificato l'URL della pagina;
- HTTP POST: le informazioni ricevute sono trasparenti, e l'utente vedrà solo come la pagina verrà cambiata alla fine del caricamento delle nuove informazioni.

Siccome la sintassi di AJAX è piuttosto verbosa, e quindi lunga da implementare, solitamente si utilizza jQuery insieme ad AJAX.

2.4 PHP

Fino ad ora, tutti i linguaggi descritti vengono interpretati direttamente dal browser dell'utente. Ciò significa che l'unico compito del web server è quello di inviare, dopo averne ricevuto richiesta, tutti i dati relativi alla pagina, che sarà poi elaborata e visualizzata sempre allo stesso modo indipendentemente da chi la va ad usare. AJAX e jQuery daranno una maggiore interattività della pagina, tuttavia la stessa serie di input darà sempre lo stesso risultato. Questo può essere un aspetto voluto, ma non necessariamente è sempre così: vi sono dei casi in cui il contenuto di una pagina deve essere personalizzato, in base all'utente. Si pensi, ad esempio, ad un sito di commercio elettronico: l'utente, aprendo la pagina relativa alla cronologia dei suoi acquisti, si aspetta di vedere l'elenco di prodotti che ha acquistato in passato che si aggiorni a mano a mano che egli continui ad utilizzare il sito, e che quindi cambi anche quando il browser viene chiuso e riaperto. Serve quindi un modo per rendere le pagine web ancora più dinamiche, nel senso che è necessario un ulteriore strumento che reperisca dati diversi a seconda delle circostanze di navigazione.

Da qui nasce il PHP, acronimo di *PHP: Hypertext Processor*, un linguaggio di scrittura Open Source usato nella programmazione di siti web. Dato che logica, grafica e struttura sono già ricoperti dagli altri linguaggi, l'obiettivo di PHP è essere general-purpose, ovvero non ha un compito preciso, ma può essere usato per tutte le applicazioni, riunendo vari principi degli altri linguaggi. Praticamente tutti i siti delle grandi aziende (tra cui Google, Facebook e WordPress) usano PHP al loro interno per mostrare pagine diverse a parità di indirizzi URL, proprio in base a chi le visita. A differenza dei linguaggi già descritti, è un linguaggio lato server, ovvero non viene analizzato direttamente dal browser dell'utente, che, al contrario, è invisibile alla sua sintassi ed elaborazione. Invece, questo viene processato direttamente dal web server tramite un interprete PHP, che costruisce la pagina web da visualizzare, e solo successivamente viene inviata all'utente. Questo comporta che l'utente non potrà vedere il processo che ha portato alla visualizzazione della

pagina, ma solo il prodotto finale, composto da HTML, CSS e JavaScript. Non essendo interpretato dal browser, significa che si avrà sempre lo stesso risultato indipendentemente da esso. È inoltre cross-platform, ovvero il sistema operativo su cui viene fatto eseguire non ha importanza e funzionerà sempre allo stesso modo. La versione attuale più aggiornata, al momento della stesura di questo documento, è la 8.2, tuttavia quella più utilizzata, ed usata come base anche per il progetto, è la versione 7.

Un file PHP, per essere riconosciuto ed elaborato come tale, deve avere estensione *.php*. Però non è detto che al suo interno deve necessariamente avere del codice PHP: se ne è privo, il web server non eseguirà alcuna elaborazione ulteriore ed invierà la pagina all'utente così com'è. Infatti, al suo interno, può contenere HTML, CSS, JavaScript e testo normale.

La dinamicità delle pagine web fornita dal PHP è data anche dal fatto che permette l'elaborazione dei dati inviati attraverso un form HTML. Quest'ultimo è una sezione della pagina dentro cui è possibile inserire delle informazioni che, una volta premuto il relativo pulsante di conferma invio dei dati, verranno inviate al web server ed elaborate, fornendo una risposta diversa in base a ciò che è stato inserito. Dei chiari esempi sono quelle sezioni all'interno di un sito web in cui bisogna inserire il proprio nome utente e password per accedere al proprio profilo. Queste informazioni sono inserite dentro un form, che poi verrà controllato, per verificare che le credenziali inserite appartengono effettivamente ad un profilo esistente e, nel caso, accettare o rifiutare il tentativo di accesso. I metodi principali di invio dei dati sono gli stessi discussi per quanto riguarda la libreria jQuery di JavaScript, ovvero GET e POST. In questo caso, però, il modo in cui essi si inviano diventa rilevante:

- Il metodo POST è quello normalmente più utilizzato, in quanto non fa vedere all'utente le informazioni trasferite, soprattutto quando queste sono sensibili (come le credenziali);
- Il metodo GET, invece, visualizza i parametri inviati all'interno dell'URL della pagina come se fossero delle variabili. Viene utilizzato quando la stessa pagina web viene utilizzata per più scopi (ad esempio

sia l'elenco sia la visualizzazione specifica del contenuto delle email, oppure una pagina che visualizza un'immagine diversa in base ai parametri inviati) e quindi si vuol dare all'utente la possibilità di poterla salvare tra i preferiti (bookmark) in modo da poterci tornare in seguito senza dover inserire un'altra volta tutti i dati.

Questi concetti sono talmente importanti all'interno di PHP che i dati ottenuti da queste richieste vengono inseriti all'interno di variabili speciali, chiamate *superglobali*.

2.4.1 Sessioni

Una volta che vengono cambiate le informazioni di una pagina, però, quando si decide di visitarne un'altra, esse vengono inevitabilmente perse, in quanto il protocollo HTTP, usato per navigare su Internet, è *stateless*, ovvero non mantiene in memoria ciò che è accaduto precedentemente, ma si occupa solo di trasferire i dati richiesti. Ciò significa che se un utente accedesse al proprio profilo, quando cambierà pagina egli risulterebbe nuovamente come un utente non registrato e dovrebbe effettuare nuovamente l'accesso. Questo comporta una quasi totale inutilità dell'invio di dati al web server. Si deve quindi trovare un modo per memorizzare, almeno temporaneamente, le azioni intraprese durante la visita di un sito web.

PHP mette a disposizione una funzionalità mirata a questo problema, ovvero le sessioni. Una sessione permette di memorizzare, almeno fino a che il browser non viene chiuso, le informazioni relative alla sessione di navigazione attuale. Visitando un'altra pagina, quindi, l'utente che ha effettuato l'accesso al sito continuerà a rimanere acceduto, anche se la pagina venisse chiusa e decidesse di visitare un altro sito. Al contrario dei cookies, ciò che è memorizzato con le sessioni, come tutto il PHP, è invisibile all'utente: anche se dati sensibili dovessero venire memorizzati, non sarebbe capace in alcun modo di visualizzarli, in quanto vengono salvati lato server, ed associati ad un codice identificativo che rappresenta la sessione. Ogni volta che l'utente

cambia pagina, invia al server il codice della sua sessione, in modo che quest'ultimo possa distinguere quale sessione appartiene a chi. Anche in questo caso, le sessioni sono una parte fondamentale del PHP, ed in merito a ciò si utilizza una variabile superglobale per gestirle. Per cancellare la sessione attuale, ad esempio a seguito della disconnessione dell'utente, basta distruggerla con un'apposita funzione, che si impegnerà anche di eliminare tutte le variabili ad essa associate.

2.4.2 Connessione ai database

Le sessioni sono indubbiamente comode per quanto riguarda la memorizzazione dei dati durante la navigazione di una pagina web, permettendo di mantenere le modifiche fin tanto che la si utilizza. Però vi è un ulteriore problema: la memorizzazione di questi dati è temporanea, e una volta che il tempo di vita della sessione scade, o si chiude totalmente il browser, anche questi dati vengono persi. Quindi, ad una visita successiva della pagina, l'utente verrà trattato come se fosse la prima volta che vi accede, e tutta la cronologia delle sue visite passate non sarà più visibile.

Vengono fornite così una serie di funzionalità per la connessione e manipolazione di banche dati, per salvare le azioni intraprese dall'utente in modo permanente, su un supporto esterno al browser. Usando queste funzionalità, è possibile utilizzare il linguaggio SQL per effettuare interrogazioni al database, che possono includere aggiunte, modifiche ed eliminazioni. Il database a cui ci si connette può essere sia sulla stessa macchina che ospita il web server (utilizzando come indirizzo quello del *localhost*), oppure su macchine diverse, specificandone sempre indirizzo IP e porta. Anche il tipo di database può essere qualunque: in questo contesto si è utilizzato MySQL Workbench per motivi di inserimento dati, tuttavia è possibile utilizzarne molti altri, tra cui PHPMyAdmin e MongoDB.

2.5 Leaflet.js

Leaflet è una libreria Open Source di Javascript lato client che consente di visualizzare la mappa geografica ed eseguire tutte le operazioni grafiche per visualizzare i dati interessanti. Essa è estremamente leggera da un punto di vista di dimensioni (molto minore di 1 megabyte), e permette di costruire facilmente applicazioni e siti web mobile-first e mobile-friendly. È supportata da tutti i browser più utilizzati e, essendo Open Source, chiunque può contribuire al suo miglioramento. È basata su un'altra libreria, OpenStreetMap: mentre quest'ultima fornisce la mappa sotto una licenza di Open Data, Leaflet fornisce un wrapper Javascript attorno ad essa, tramite il quale è possibile fare molte operazioni di visualizzazione dati, che si andranno a discutere in seguito. Anche OpenStreetMap è Open Source, e i suoi contributori possono tenerla sempre accuratamente aggiornata. Questi cambiamenti si rifletteranno direttamente anche su Leaflet senza dover effettuare alcuna modifica, se non dover scaricare la nuova versione se si dovesse operare la libreria in locale. Tuttavia, come molte altre librerie HTML, è possibile utilizzare un'API online, includendola nella pagina in cui la si vuole mostrare, facendo sì che l'utente che la visualizza non la debba scaricare nuovamente nel caso in cui abbia visitato un altro sito che ne mostra una simile.

2.5.1 La mappa

La mappa viene considerata un vero e proprio oggetto, che deve essere istanziato prima di essere manipolato. È l'elemento fondamentale dell'intera libreria: tutte le operazioni saranno fatte su di essa, quindi va istanziata subito dopo aver strutturato la pagina in cui andrà visualizzata. I dati da visualizzare e la mappa stessa vanno forniti dallo sviluppatore, in quanto quest'ultima non deve necessariamente essere quella di OpenStreetMap, ma solamente una compatibile con le funzioni che Leaflet mette a disposizione. La mappa base deve essere una *tile map*, ovvero un'insieme di piccole immagini, come se fossero le tessere di un puzzle, che insieme compongono la

mappa finale. Quella più utilizzata, ed impiegata anche in questo progetto, è quella di OpenStreetMap, in quanto Open Source, e quindi libera di licenze restrittive e abbonamenti. Tuttavia, non è l'unica: vi sono diversi altri fornitori di mappe simili, compatibili con Leaflet. In questo caso, bisogna fare molta attenzione alle loro condizioni di utilizzo: infatti, è comune che all'interno della loro licenza di Copyright, sia necessaria fornire l'attribuzione ai creatori e/o un collegamento diretto al loro sito. Ogni frammento di una tiled web map è un'immagine quadrata, di solito di formato PNG o JPEG, ed ognuno viene caricato separatamente rispetto agli altri, un compito facilmente parallelizzabile e più veloce da eseguire rispetto che visualizzare una sola grande immagine.

La visualizzazione della mappa può essere cambiata in diversi modi, ad esempio ingrandendola o rimpicciolendola con lo zoom, limitando quest'ultimo, spostandola trascinando il mouse o usando il display e, se necessario, limitare il movimento facendo sì che un determinato pixel sia sempre visualizzato, come un perno. Inoltre, offre la possibilità di localizzare con più o meno precisione l'utente utilizzando il suo sensore GPS, utile se si volesse cercare un determinato luogo entro un certo raggio.

2.5.2 Segnalini, popups e coordinate geografiche

Sulla mappa possono essere posti dei segnalini, o Marker, che servono per denotare dei punti di interesse al suo interno. La posizione dei segnalini non è prefissata e possono essere posizionati in qualsiasi punto, purché vengano fornite le sue coordinate geografiche. Viene fornito anche un modo per renderli accessibili, come una descrizione che può essere interpretata dai lettori di schermo o la possibilità di poter essere selezionati utilizzando il pulsante per la tabulazione della tastiera. Di base, l'icona dei segnalini è molto simile a quella usata su Google Maps, tuttavia è possibile facilmente cambiarla con una personalizzata, e cambiarne anche la dimensione. È bene che non sia troppo grande per non ostacolare la vista dei restanti frammenti di mappa. Se ne viene data la possibilità, essi possono essere spostati, trascinandoli in

una nuova posizione, e far scattare un evento per decidere cosa deve succedere quando la loro posizione viene modificata. I segnalini possono anche non essere punti precisi, bensì aree: per questo, è possibile definire dei poligoni o linee di forma arbitraria la cui area all'interno rappresenta un unico grande segnalino.

Ad ogni segnalino può essere associato un popup, ovvero un messaggio che verrà visualizzato quando si effettua un click su di esso. Questo popup può contenere qualsiasi testo, e viene solitamente utilizzato per fornire informazioni sul luogo in cui risiede il segnalino. La loro dimensione può essere limitata e ad ognuno può essere associata una classe CSS che ne descrive le caratteristiche, come il font, colore, dimensione del testo, effetti grafici e così via. Un popup può anche non essere associato ad alcun segnalino, però bisogna fornire sempre le coordinate della mappa in cui esso verrà mostrato. Anche in questo caso, alla chiusura di un popup può essere associato un evento per modificare il comportamento della pagina.

Per quanto riguarda le coordinate geografiche che vanno associate ai vari elementi della mappa, ne sono supportati diversi tipi, tra cui il più popolare GeoJSON. In alternativa, non avendo a disposizione nessun formato particolare, si può utilizzare una coppia di valori latitudine-longitudine, con un numero a piacere di cifre decimali, che verrà interpretata dalla libreria correttamente. Quest'ultima fornisce anche una funzione per poter convertire un dato GeoJSON nella coppia latitudine-longitudine se fosse necessario.

2.5.3 Layers

La mappa è strutturata a layers, ovvero strati logici che è possibile modificare senza che uno impatti il funzionamento dell'altro. Vi può essere un numero arbitrario di strati, e l'ultimo aggiunto si andrà a sovrapporre a tutti quelli già presenti. Solitamente, il primo strato è quello della tiled map, a cui sopra si andranno ad impilare tutti gli altri. Tutto ciò che si andrà ad aggiungere ad essa è un layer, compresi segnalini e popup, e ad ognuno si possono associare determinati eventi, simili a quelli già discussi in preceden-

za.

Questi strati indipendenti possono essere raggruppati in un unico insieme tramite un'apposita funzione. Quando si aggiunge un elemento ad uno di questi gruppi, non viene creato un nuovo strato, ma l'elemento viene solamente aggiunto a quel gruppo. Questo è utile quando si vuole raggruppare elementi simili (ad esempio tutti i segnalini, tutti i popup, eccetera) che devono avere almeno un evento in comune, o quando li si debba gestire tutti allo stesso modo senza utilizzare dei cicli, che possono diventare lunghi se gli elementi aggiunti alla mappa sono molti. In ogni caso, accedendo al gruppo si possono andare a controllare e modificare tutti gli elementi che ne fanno parte in modo indipendente, tuttavia se essi devono essere gestiti tutti allo stesso modo, li si può modificare tutti insieme senza dispendio di tempo.

2.5.4 Plugins

Leaflet fornisce un insieme di funzionalità semplici, dando al programmatore tutto il necessario al proprio lavoro e mantenendo una velocità di elaborazione rapida grazie alla sua leggerezza. Tuttavia, se queste funzionalità non dovessero bastare, esistono i plugins, ovvero delle estensioni che ampliano i modi di visualizzazione della mappa. I plugins possono essere sviluppati da chiunque, in quanto Leaflet stesso è Open Source, e sono tutti pubblicamente disponibili. Essi migliorano la visualizzazione della mappa in vari modi, ad esempio permettendo ad essa di essere aperta a schermo intero, oppure raggruppare tutti i segnalini di una certa zona in uno unico nel caso ce ne fossero troppi, e visualizzarli individualmente solo nel caso la mappa fosse vista con un certo livello di zoom, e molti altri, che forniscono una quantità innumerevole di modi per la visualizzazione dei dati, nonché l'aggiunta di altre interazioni con la mappa. In questo contesto, si discuterà del plugin utilizzato per mostrare una mappa di calore: `Heatmap.js`.

2.5.5 Heatmap.js

Heatmap.js è l'estensione utilizzata per visualizzare una mappa di calore. Come la mappa base, anch'essa è un oggetto che deve essere istanziato prima dell'utilizzo. Le informazioni per determinare i colori delle varie zone possono essere aggiunte sia gradualmente sia tutte in una volta, a seconda del dataset che il programmatore ha a disposizione. Queste informazioni, per essere aggiunte, devono essere in formato JSON, e devono contenere le coordinate del punto centrale, in cui i dati sono concentrati, ed il valore che rappresenta quanti dati vi sono in quel punto. Per capire di quale colore tingere le varie zone della mappa, si può procedere sia preimpostando un valore che rappresenta la concentrazione massima (estremo globale), sia usando il valore massimo tra quelli forniti come riferimento per la concentrazione massima (estremo locale). Oltre a ciò, si devono fornire alcune impostazioni di base alla mappa di calore prima di poterla usare, come il *range* dei punti del dataset, la trasparenza dei colori e il nome dei campi del dataset che rappresentano latitudine, longitudine e valore dei dati. Il numero di dati può essere arbitrario, dato che è una libreria molto efficiente in termini di memorizzazione di essi.

2.6 Vue.js

Vue.js (pronunciato come *view*) è un framework di Javascript pensato per la costruzione delle interfacce utente. Tramite esso, è possibile associare a qualsiasi elemento HTML degli eventi, che vengono scatenati ogni qual volta che l'utente vi interagisce. Ciò è naturalmente possibile farlo anche senza questo framework, tuttavia la differenza risiede nel fatto che richiede molto meno tempo, e il codice risulta più elegante, pulito e comprensibile grazie ai suoi costrutti. Nonostante sia nato per Javascript, Vue viene usato anche per strutturare codice HTML e CSS, come se fosse un'estensione che racchiude tutti e tre. Anche se Vue è un framework avanzato, con innumerevoli fun-

zionalità e capace di dar vita ad applicazioni complesse, si discuterà solo dei concetti più semplici ed utilizzati.

Come con Leaflet, anch'esso è possibile utilizzarlo scaricandolo in locale o utilizzando un'API esterna, includendola nell'head della pagina o applicazione come script, per garantire che sia sempre aggiornato. Tuttavia, se si utilizza la prima opzione, è possibile utilizzare una funzionalità offerta da Vue che altrimenti non sarebbe possibile: i Single-File Components, che verranno descritti più in dettaglio di seguito.

2.6.1 Single-File Components

Un Single-File Component, o SFC, è un file con estensione *.vue* che viene usato per incapsulare il modello, logica e grafica di un elemento della pagina o applicazione web. Infatti, proprio per il primo aspetto, viene introdotto un nuovo tag HTML: `<template>`, che racchiude proprio il modello dell'elemento, ovvero il suo codice HTML. All'interno di un SFC, divisi dai rispettivi tag di apertura e chiusura, vengono memorizzati anche la grafica e la logica, usando rispettivamente CSS e Javascript con i costrutti di Vue. L'idea alla base è che nonostante siano linguaggi diversi, l'importante è che siano separati solamente gli strati che compongono un elemento, e non i linguaggi stessi. Ciò significa che anziché memorizzare un elemento in tre file diversi, divisi per linguaggio, è più utile ed intuitivo che vi sia un unico file che detti la rappresentazione e le funzionalità di un elemento. Così facendo, se esso non dovesse comportarsi come previsto, non bisogna fare altro che accedere al relativo, ed unico, file che lo identifica.

L'unico riguardo di questi tipi di file è che essendo che sono nativi di Vue, per poterli utilizzare occorre prima passare attraverso un processo di build, in cui, utilizzando uno strumento apposito, il file viene precompilato e separato nei linguaggi che lo compongono. Questo permette che non vi sia nessun processo di compilazione a run-time, non sacrificando così del tempo, avendo comunque un unico file a livello di sviluppo.

2.6.2 Binding

Il binding è una funzionalità che permette di cambiare dinamicamente interfaccia e comportamento dell'applicazione web, sincronizzandola con i dati in input e output dell'utente. Questo viene ottenuto applicando delle direttive speciali agli elementi della pagina, rendendoli reattivi ai cambiamenti e all'interazione con essa. Il binding può essere applicato a quasi tutti gli elementi, tra cui testi, tag HTML e i loro attributi, sia in modo statico (ad esempio mostrando un contatore) sia in modo dinamico (ad esempio rendendo un testo inserito dall'utente con testo, dimensione e comportamento decisi da lui). Permette anche di richiamare funzioni, cicli ed eventi che modificano il comportamento della pagina. Si vuole notare che questo processo è invisibile all'utente finale: esso vedrà una normale pagina web composta solo da costrutti nativi di Javascript, CSS e HTML, in quanto i costrutti di Vue verranno precompilati per essere correttamente interpretati nativamente dall'applicazione.

Capitolo 3

Progetto: Visualizzazione del traffico di Bologna tramite spire

In questo terzo capitolo si parlerà del progetto redatto seguendo i principi e tecnologie presentate nei precedenti capitoli. Se ne presenterà un quadro generale nella sezione seguente, insieme agli obiettivi. In seguito, se ne mostreranno le funzionalità, e le decisioni di maggiore importanza prese durante lo sviluppo che hanno dato un notevole impatto sulle sue prestazioni.

3.1 Introduzione

Sotto il manto stradale della città di Bologna sono situate delle spire, ovvero dei sensori che, al passaggio di un'auto, ne registrano il transito, e, periodicamente, vengono inviate ad un server centrale, appartenente al Comune di Bologna, le informazioni relative alla quantità di auto transitate in un certo periodo di tempo. Il numero di spire è elevato: ve ne possono essere anche più di una all'interno della stessa strada, soprattutto se questa è lunga o ricca di incroci, per dare un'idea migliore delle auto che la usano. Sono più concentrate verso il centro della città, e diventano più rare a mano a mano che ci si allontana da esso. Date queste informazioni, l'obiettivo del progetto è una visualizzazione del traffico generale dell'intera città usando i dati rac-

colti dalle spire, mostrare le aree più trafficate, ed eventualmente il sensore che ha catturato più veicoli in transito. Siccome la visualizzazione è limitata dal numero e dalla posizione delle spire, non significa che nelle altre aree non vi sia traffico, ma semplicemente vi è una quantità di dati insufficiente per darne una stima.

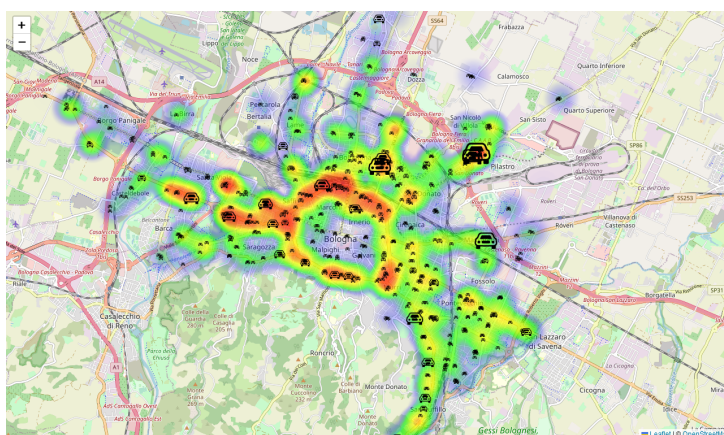


Figura 3.1: Una vista della mappa

In figura 3.1 viene mostrata la mappa la prima volta che viene caricata. Se ne darà una spiegazione più approfondita nelle sezioni seguenti, insieme ad altre immagini, per mostrarne le funzionalità più interessanti. Tuttavia, questo è utile per dare un'idea di ciò che si andrà a presentare.

3.2 Dati dei sensori

Ogni spira ha dei propri dati fissi, indipendentemente dal traffico. Questi comprendono, ad esempio, il proprio codice identificativo, le coordinate geografiche della posizione, il codice e nome della via su cui sono state installate, e così via. I dati più dinamici ed importanti, invece, sono proprio relativi al numero di veicoli che hanno attraversato il sensore. Essi sono suddivisi per giorno e per ora, per cui, per ogni giorno di funzionamento, vi saranno almeno 24 campi da considerare. Si vuol far notare che siccome questi provengono da

sensori fisici, è possibile che non siano privi di errori: un malfunzionamento casuale potrebbe compromettere l'affidabilità dei dati registrati.

Tutti i dati vengono caricati circa mensilmente all'interno del sito degli Open Data di Bologna, suddivisi per anno, da cui è possibile scaricare interamente o parzialmente i dataset, offerti in diversi formati tra cui JSON, GeoJSON e CSV, con tutte o parte delle informazioni relative a ciascuna spirale. Viene inoltre fornita una API per attingere ad essi senza scaricare alcun file. Tuttavia, siccome questo progetto mira all'utilizzo di molti dati, potenzialmente anche di anni diversi, si è deciso di scaricare in locale i dataset ed inserirli in un database per non sovraccaricare inutilmente il server di Bologna di richieste. Questo, inoltre, consente di evitare l'impossibilità di utilizzare la pagina web nel caso in cui i server principali di Bologna siano sotto manutenzione. Si è deciso di utilizzare i dati dall'anno 2020 al 2022, in quanto i relativi dataset di cui fanno parte sono già completi e non riceveranno aggiornamenti futuri, al contrario del 2023, ancora in formazione.

3.3 Controlli



Controlli

Giorno iniziale di cui visualizzare le informazioni: 30 / 10 / 2022

Giorno finale di cui visualizzare le informazioni: 30 / 10 / 2022

☒ Giorno intero

☒ Giorno singolo

☒ Mostra mappa di calore

☐ Animazioni segnalini

Figura 3.2: I controlli disponibili per la visualizzazione dei dati sulla mappa

In figura 3.2, sono presentati tutti i possibili controlli disponibili per visualizzare i dati sulla mappa. Alcuni di essi sono invisibili o disattivati, per evitare che l'utente modifichi informazioni che possano portare a delle in-

consistenze di visualizzazione. Essi verranno mostrati solamente quando una determinata serie di opzioni viene attivata o disattivata, in base al tipo di controllo che offrono. Si vuol far notare che non è presente alcun pulsante per avviare l'aggiornamento della mappa: questo perché essa verrà aggiornata automaticamente ogni qualvolta che qualsiasi controllo viene modificato, per evitare dei click aggiuntivi inutili da parte dell'utente. Infatti, si è supposto che egli vorrà vedere subito i cambiamenti, oltre che per dare un'idea di maggior interattività con la pagina web. Si discuterà di tali controlli nelle sezioni seguenti, presentandone nel dettaglio le funzionalità e i loro effetti sulla mappa. Tutti i controlli sono indipendenti gli uni dagli altri e possono essere sovrapposti a piacere, fin tanto che non provochino inconsistenze nella visualizzazione. In tal caso, essi vengono automaticamente disattivati ed ignorati.

3.3.1 Giorni di visualizzazione

In testa all'elenco di controlli vi sono quelli per modificare i giorni di cui si è interessati a conoscerne il traffico. Facendo click su di essi, verrà proposto un menù a tendina, simile ad un calendario, da cui poter scegliere visivamente i giorni senza doverli necessariamente scriverli a mano con la tastiera. Il primo controllo è sempre attivo, e rappresenta il giorno iniziale, o unico, di cui mostrare le informazioni. Il secondo controllo, invece, rappresenta l'ultimo giorno di visualizzazione. Normalmente è disattivato, in quanto se l'utente è interessato a vedere il traffico di un solo giorno specifico, non c'è motivo per fargli scegliere la data finale, in quanto essa e quella iniziale saranno sempre le stesse. Infatti, mentre è disattivato, ogni volta che la data iniziale viene cambiata, quella finale verrà anch'essa modificata di conseguenza per rispecchiare la scelta dell'utente. Questo ha solo un impatto estetico, in quanto, in ogni caso, non verrà presa in considerazione.

Quando l'utente, invece, è intenzionato a conoscere il traffico non di un giorno, ma di un arco temporale che ne comprende molteplici, disattivando l'opzione *Giorno singolo*, il controllo sulla data finale si attiverà e, finché è attivo,

non verrà cambiato alla modifica della data iniziale. In questo caso, tutti i giorni tra quello iniziale e quello finale, questi ultimi compresi, verranno considerati per la visualizzazione delle informazioni, ed i relativi dati saranno mostrati. Non vi è limite alla quantità di giorni visualizzabili. L'unico vincolo è che le date di inizio e di fine siano comprese tra gli anni 2020 e 2022, in quanto per i restanti non vi sono dati.

Se si cerca di mostrare dei giorni che cadono al di fuori di questi anni, verrà mostrato a schermo un avviso. Allo stesso modo, nel caso in cui l'utente inserisca una data di fine che accade prima della data d'inizio, la mappa risulterà vuota, e verrà stampato in cima ai controlli un testo, avvertendo l'utilizzatore della problematica.

In ogni caso, si vuol far notare che vi possono essere dei casi in cui in un periodo una zona risulti poco trafficata, mentre in quello seguente lo è molto. Questo perché i dati utilizzati contengono solo il numero di veicoli transitati in una determinata strada, ma non considerano la possibilità che vi siano dei motivi straordinari per quella determinata visualizzazione. Ad esempio, se una strada solitamente molto trafficata è soggetta a lavori di manutenzione per un certo periodo di tempo, verrà poi considerata come zona non congestionata. Anche se, effettivamente, quell'informazione non è sbagliata, potrebbe portare a pensare all'utente che controlla la mappa che essa sia anche transitabile, fatto che poi non si rivelerà veritiero. Oppure una zona normalmente tranquilla su cui però si verifica un incidente e, a causa delle lunghe code, per una determinata ora risulti insolitamente trafficata. Purtroppo, questa è una limitazione intrinseca dei dati, in quanto non è possibile conoscere le circostanze riguardo al traffico, ma è possibile solamente effettuare delle deduzioni a riguardo.

3.3.2 Ore di visualizzazione

Normalmente, se non diversamente specificato, la visualizzazione di un giorno o arco di giorni avviene considerandoli nella loro interezza, ovvero verranno mostrati di dati a partire dalla mezzanotte del giorno iniziale fino

alla mezzanotte seguente del giorno finale. Così facendo, si potranno conoscere le aree e vie più trafficate di quei periodi.

Tuttavia, non sempre si può essere interessati a giorni completi: è possibile che si voglia conoscere il traffico solamente in una particolare ora o periodo del giorno, ad esempio per capire quali sono le strade più congestionate di quel momento e, di conseguenza, pensare a percorsi alternativi per evitarle. Oppure per il motivo opposto, cioè cercare le vie meno trafficate per raggiungere la propria destinazione, dato che spesso i navigatori GPS installati sul proprio veicolo cercano solamente il percorso più corto senza tener conto delle code che vi possono essere.

A tal proposito, si è implementata la possibilità di visualizzare il traffico solo per determinate ore della giornata. Disattivando l'opzione *Giorno intero*, infatti, appariranno sopra di essa due nuovi campi, attivi e modificabili: il primo rappresenta l'ora di inizio da cui partire nella visualizzazione dei dati, e il secondo rappresenta l'ora finale. Così facendo, il traffico presente prima dell'ora iniziale e dopo l'ora finale non verrà considerato. È possibile, quindi, che la distribuzione del traffico visualizzata sulla mappa cambi, nonostante ci si trovi nella stessa giornata. Questo perché è possibile che nelle ore diurne e notturne le aree più trafficate della città non siano le stesse, ad esempio a causa di strade chiuse o, in generale, meno veicoli in circolazione.

La visualizzazione di solo una parte delle ore della giornata funziona bene anche per periodi di tempo che comprendono molteplici giorni. Disattivando anche l'opzione *Giorno singolo* e selezionando una data finale differente da quella iniziale, verrà mostrato il traffico di quel periodo come di consueto, ma verranno considerate solo le ore selezionate per ogni giorno di quel periodo di tempo.

Nonostante vi sia la possibilità di modificare anche i minuti e non solo l'ora, essi non verranno considerati. Questo nasce da una limitazione dei dati, che forniscono la divisione del traffico solo per ore intere, e non viene ulteriormente suddiviso in base ai minuti. Inoltre, se nell'ora finale viene digitata l'ora 00:00, essa verrà considerata come 24, ovvero come la mezzanotte del

giorno seguente. Anche questo viene fatto per scavalcare una limitazione del tipo di campo utilizzato per il controllo dell'ora, in quanto ammette solo ore a partire da 00:00 fino a 23:59.

Riattivando l'opzione *Giorno intero*, le ore torneranno nuovamente invisibili, e la mappa verrà aggiornata considerando nuovamente l'intera giornata o periodo temporale.

3.3.3 Rotazione giorni

Fino ad ora ci si è concentrati sulla possibilità di mostrare dati riguardanti giorni e ore differenti, anche per periodi di tempo notevolmente prolungati, come mesi ed anni. Questi vengono ottenuti sommando tutti i vari dati appartenenti ai giorni e ore selezionati, e visualizzati come un aggregato di essi. Nonostante questa è una funzionalità utile, l'utente non può capire come il traffico sia variato nel corso dei giorni di quel periodo, e come quindi abbia portato quei risultati, se non con un lavoro di molti click per cambiare di volta in volta il giorno da visualizzare per capire il decorso nel tempo.

Per questo, si è data la possibilità per cui, dato un periodo di tempo, si visualizzano le informazioni non tutte allo stesso tempo, ma un po' per volta, alternandone la visualizzazione in mappa ogni pochi secondi, proprio come una rotazione delle stesse. Questo si traduce nella possibilità per l'utente di avere un'idea, nel periodo da lui selezionato, quale sia stato il giorno (e potenzialmente anche l'ora) e la zona più trafficati. Infatti, per ogni giorno alternato, la mappa si aggiornerà automaticamente di conseguenza e, supponendo che l'utente stia concentrando la sua attenzione su di essa, vedrà in modo più chiaro il modo in cui il traffico si è evoluto nel corso del tempo. Questa opzione, normalmente invisibile, diventerà visibile e selezionabile solo nel caso in cui l'opzione *Giorno singolo* sia disattivata, ovvero è possibile attivarla solo nel caso in cui si vogliano conoscere le informazioni relative ad un intervallo di tempo che comprende molteplici giorni.

Attivando l'opzione *Rotazione giorni*, dopo che essa è diventata visibile, comparirà, sotto di essa, un menù a tendina, invitando l'utente a scegliere il tipo

di rotazione dei giorni. Infatti, è possibile visualizzare le informazioni divise per giorno, settimana e mese. Nel primo caso, sulla mappa verranno mostrati i dati relativi al giorno singolo preciso. Nel secondo caso, la rotazione avverrà per intervalli di sette giorni. Si vuol far notare che una settimana non è stata intesa come il periodo lunedì-domenica, ma essa dipende dalla data iniziale di visualizzazione (ovvero se il primo giorno è un giovedì, una settimana sarà giovedì-mercoledì, similmente per gli altri). Inoltre, è possibile che nella visualizzazione dell'ultima settimana, aggiungere sette giorni superi la data di fine inserita per l'arco di giorni da mostrare. Questa è una funzionalità intesa e l'ultima settimana non verrà accorciata di proposito per rispettare il limite. Si è pensato che l'utente voglia vedere sempre periodi lunghi sette giorni e, nel caso fosse necessario, viene lasciato a lui il compito di verificare che l'ultimo giorno dell'ultima settimana coincida con la data finale.

Per il mese, invece, il numero del giorno della data di inizio e di fine dell'intervallo di visualizzazione saranno ignorati, e verranno considerati solo il loro mese e anno. Di conseguenza, la rotazione avverrà mostrando i dati relativi a tutti i giorni dei mesi considerati (ad esempio se la data di inizio è il 13 luglio e quella di fine il 23 settembre, verranno mostrati i dati di tutto luglio, tutto agosto e tutto settembre, anche se i giorni precisi da cui iniziare e finire non sono precisamente quelli di inizio e fine mese). Questo è dato da un'ulteriore limitazione di efficienza, di cui se ne discuterà nella sezione immediatamente successiva. Essa, però, non è presente nel caso in cui la rotazione sia disattivata: in quel caso, infatti, verrà mostrato correttamente solo l'intervallo temporale che l'utente decide di vedere.

Anche in questo caso, disattivando l'opzione *Giorno intero* e selezionando una qualunque ora di inizio e di fine, la rotazione avverrà all'interno dell'arco temporale desiderato, mostrando solamente i dati compresi nelle ore stabilite.

Questo ciclo di giorni, settimane e mesi, dell'arco temporale desiderato, verrà accompagnato da un titolo, che comparirà sopra alla mappa e sotto ai controlli, che indicherà il giorno, settimana o mese di cui si stanno attualmente

mostrando i dati, comprese le ore di inizio e di fine del transito dei veicoli da considerare. Dopo qualche secondo, quando è il momento di mostrare il periodo di tempo successivo, insieme alle informazioni sulla mappa, cambierà anche il titolo di conseguenza. Questo serve per far sì che l'utente non debba ricordarsi che giorno, settimana o mese sta venendo visualizzato, dato che sarebbe facilmente dimenticabile se ci si vuole concentrare anche sui dati che stanno venendo mostrati. La formattazione del titolo è divisa in tre modi, a seconda del tipo di rotazione:

- Rotazione giornaliera: viene mostrato il giorno preciso di cui si stanno venendo visualizzati i dati;
- Rotazione settimanale: vengono mostrati gli estremi del periodo che sta venendo visualizzato;
- Rotazione mensile: viene mostrato il nome del mese e l'anno che attualmente sta venendo presentato.

Una volta che viene raggiunta o superata la data finale della visualizzazione, il ciclo di rotazione ricomincerà da capo, tornando alla data iniziale, e continuando a cambiare i dati che stanno venendo mostrati ogni pochi secondi, fino a che qualsiasi controllo non venga nuovamente modificato. In tal caso, la rotazione si interromperà, e la mappa verrà rinnovata con i nuovi cambiamenti, ricominciando nuovamente il ciclo nel caso la relativa opzione continuasse ad essere attivata.

3.3.4 Ottimizzazione della ricerca e aggregazione dei dati

Come precedentemente menzionato, tutti i dati relativi al traffico, disponibili gratuitamente per tutti sul sito ufficiale del comune di Bologna, sono stati scaricati e vengono mantenuti in un database locale. Questo comprende i dati relativi a tutti i giorni dal 2020 al 2022, componendo così un dataset di dimensioni piuttosto grandi, che include una quantità di record altrettanto

numerosa. Nella selezione del periodo da visualizzare, devono essere reperiti i dati relativi a tutti i giorni di quel periodo (si può considerare il caso in cui si voglia visualizzare un solo giorno come un periodo di lunghezza 1). Più questo periodo diventa lungo, più dati dovranno essere recuperati dal database, elaborati e mostrati sulla mappa. Sotto il manto stradale di Bologna vi sono installate tra le 800 e le 900 spire, ed i dati che esse registrano vengono suddivisi nel database in base al giorno. Per ogni giorno vi sono 24 campi per ogni ora, senza contare tutti quelli relativi all'identificativo della spira, che permette di distinguerla dalle altre, dalla loro posizione geografica, dal livello registrato di accuratezza nella registrazione dei dati, e così via. Questo comporta che, supponendo che un utente voglia visualizzare i dati per un arco di tempo di un solo mese, comprendendo tutte le ore del giorno, il database dovrà reperire, nel migliore dei casi, $800 * 30 = 24'000$ record, ed il programma dovrà analizzare ben $800 * 30 * 24 = 576'000$ dati relativi al traffico di tutte le ore. Come si può immaginare, questo comporta un tempo di elaborazione che, per periodi lunghi, non è più possibile trascurare, in quanto diventerebbe talmente lungo che darebbe l'impressione che la pagina si sia bloccata, o che abbia dato un errore nella visualizzazione. Si è cercato, quindi, di pensare a soluzioni che potessero essere utili per la velocizzazione di questo reperimento ed elaborazione dei dati.

Il primo passo è stato un'ottimizzazione della ricerca per quanto riguarda il linguaggio SQL: si è cercato di reperire tutti i dati utilizzando meno interrogazioni possibili, in quanto si è potuto constatare che risulta più efficiente effettuare poche interrogazioni che reperiscono una grossa porzione di dati, piuttosto che molteplici che ne ottengono di meno. Questo ha portato, infatti, ad un leggero miglioramento delle prestazioni, però rimaneva comunque presente quando il periodo desiderato comprendeva diversi mesi, o anche anni. Il miglioramento, infatti, incideva ed era visibile solamente sul reperimento di, relativamente, piccoli archi temporali, di al più 3-4 mesi. Quando questo arco diventava più lungo, questa soluzione non risultava particolarmente efficace, e si è dovuto pensare ad un'ulteriore miglioramento per accorciare

ancora di più il tempo di attesa per la visualizzazione.

La soluzione adottata, che ha cambiato radicalmente le attese e che si considera ora come un problema risolto, è stata quella di aggregare i dati relativi a più giorni, della stessa spira, in un unico record, che viene memorizzato direttamente nel database. A tal proposito, è stata creata un'ulteriore tabella, contenente, per ogni spira, tutti i suoi dati registrati relativi al traffico, suddivisi, questa volta, non più per giorno, ma per mese. Ciò significa che un unico record di questa tabella contiene i dati di un intero mese di passaggio di veicoli, che equivale a circa 30 record della tabella in cui i dati sono suddivisi per giorni, diminuendo così drasticamente il numero di informazioni sia da reperire, sia da analizzare per essere mostrati in mappa. Se nel periodo di tempo da visualizzare c'è almeno un mese *completo*, ovvero di cui si devono recuperare i dati relativi a tutti i suoi giorni, si potranno utilizzare questi dati aggregati piuttosto che sommare le informazioni relative a tutti i giorni singoli. Si può riassumere questo processo in pochi punti:

- Se nel periodo di visualizzazione c'è almeno un mese completo, si reperiscono i relativi dati aggregati;
- Per tutti gli altri giorni che appartengono a mesi non completi, vengono reperiti i dati giornalieri come di consueto;
- Si sommano tutti i dati ottenuti dai precedenti due punti e si visualizzano in mappa.

Questa aggregazione ha permesso di ridurre le attese di circa il 75%, rispetto allo stesso identico processo, che però usa solo i dati giornalieri. Nonostante questa ottimizzazione, però, un leggero tempo di attesa resta inevitabile, ma si è considerato tale periodo come un tempo accettabile, e che quindi non causerà eccessiva frustrazione nell'utente che desidera conoscere il dati del traffico relativi ad un arco di tempo, anche molto esteso.

Limitazioni

Relativamente all'ottimizzazione dei tempi di ricerca e visualizzazione delle informazioni, se ne è parlato supponendo che esse debbano essere mostrate tutte allo stesso momento, ovvero quando l'opzione *Rotazione giorni* è disattivata. Non si è discusso di quando, invece, il ciclo dei giorni è attivato, e le informazioni mostrate devono cambiare ogni pochi secondi. Infatti, in questo caso, non è possibile usare questa soluzione al pieno delle sue potenzialità per un motivo importante: nel caso della rotazione giornaliera e settimanale, anche se il periodo di visualizzazione avesse almeno un mese completo, i dati in mappa devono essere relativi al giorno (o settimana) specifico da visualizzare e non un aggregato del mese di cui fanno parte. Di conseguenza, per questi due tipi di rotazione, non è possibile utilizzare i dati aggregati per mese, ma si deve obbligatoriamente usare quelli divisi per giorno. Questo porta ad un totale annullamento dell'efficacia della strategia di riduzione di tempi di attesa per periodi di visualizzazione lunghi, e non è possibile adottare strade alternative, se non quella di utilizzare interrogazioni SQL più semplici possibile.

L'unica soluzione trovata è stata quella di limitare il periodo di tempo entro il quale è possibile effettuare la rotazione per giorno e settimana. Se questo vincolo non ci fosse, infatti, avendo l'impossibilità di ridurre i tempi di calcolo, si tornerebbe ad avere il problema di partenza, in cui si dovrebbe aspettare molto tempo prima di vedere i dati richiesti visualizzati, dando anche l'impressione che la pagina abbia smesso di funzionare. Il periodo massimo è stato impostato a 60 giorni, dato che si è considerato abbastanza lungo per fornire una risposta in tempi accettabili e non frustranti per l'utente. Nel caso di superamento di questo limite, verrà mostrato un avviso a schermo dando notizia all'utente del problema, e i nuovi dati non verranno né caricati, né mostrati. Per cui, la mappa rimarrà invariata.

Questo limite non è presente per quanto riguarda la rotazione mensile: infatti, dato che in ogni caso vengono mostrate le informazioni relative a ciascun mese completo, verranno sempre utilizzati i dati aggregati piuttosto che quelli

giornalieri, e i tempi di attesa non cresceranno così tanto al punto da causare evidenti problemi.

3.3.5 Segnalini

Una volta ottenute le informazioni, e finita la loro elaborazione, sono ora pronte per essere visualizzate. Si deve pensare ad un modo facile ed intuitivo per poterle mostrare, facendo in modo che l'utente possa trarne delle informazioni a primo impatto, senza andare necessariamente ad analizzarle una per una, dato che, come precedentemente menzionato, essendo installate tra le 800 e 900 spire, risulterebbe in un dispendioso spreco di tempo, e l'utente non sarebbe invogliato ad utilizzare l'applicazione per condurre ulteriori analisi del traffico. Anzi, senza un modo facile per la consultazione, sarebbe meno probabile che venga utilizzata per lunghi periodi di tempo, in quanto una mappa che convoglia poche informazioni è una mappa inutile, e fallisce nel suo obiettivo.

Su di essa, quindi, sono stati posizionati numerosi segnalini, che indicano il luogo esatto dove risiede ogni spira che rappresentano. Possono esservi dei giorni in cui sono di più, ed altri in cui sono di meno: questo perché vengono rimossi quando il sensore registra zero veicoli transitati in quel periodo: si è pensato che esso sia dovuto a dei problemi con il sensore stesso, e che quindi sia di poca utilità, se non dannoso mostrarlo, in quanto non farebbe altro che aumentare il numero di segnalini visualizzati. Essi sono interattivi: cliccandone su uno, nelle vicinanze si aprirà un messaggio, che mostrerà le informazioni che la relativa spira ha registrato nel periodo di tempo che sta venendo mostrato. In particolare, vengono visualizzati:

- Nome della strada su cui risiedono;
- La direzione di marcia dei veicoli: utile soprattutto se la strada ha più corsie e diversi sensi di marcia. Viene mostrata solo l'iniziale della direzione, o le iniziali di ognuna se essa è obliqua (es. Nord-Est);
- Numero di veicoli transitati nel periodo di tempo considerato;

- Media giornaliera del numero di veicoli transitati: utile se si vuol dare una stima del traffico di quella via se l'arco temporale considerato comprende molteplici giorni.

All'apertura di un secondo messaggio cliccando su un altro segnalino, il primo verrà automaticamente chiuso, per evitare click aggiuntivi inutili da parte dell'utente e per non saturare la pagina di informazioni. In alternativa, si possono chiudere manualmente, cliccando sull'apposito bottone all'interno del messaggio.

Già dal primo impatto, appena la pagina viene caricata, si può notare che non tutti i segnalini sono delle stesse dimensioni, facendo pensare che ce n'è qualcuno più importante degli altri. Questo perché la loro grandezza varia in base al traffico: più veicoli hanno transitato attraverso un sensore, più sarà grande il segnalino che lo rappresenta, fino ad arrivare ad una grandezza massima, che rappresenta quello in cui si è registrato più traffico. La dimensione dei rimanenti segnalini, invece, varia in base alla differenza tra il numero di automobili transitate registrate dal segnalino considerato e di quello con traffico maggiore. Quindi, in generale, potrebbe capitare che se una spira registra un numero di veicoli molto maggiore rispetto a tutte le altre, tutti i segnalini saranno piccoli, tranne uno. Per evitare che vi siano degli indicatori delle spire troppo piccoli, attraverso cui ci passano relativamente poche auto, come ad esempio nelle strade di periferia, si è deciso di inserire un limite inferiore nella loro dimensione, di modo che siano chiaramente visibili e possano essere cliccati con facilità.

3.3.6 Animazioni

La visione statica dei segnalini permette di fornire già un'idea della situazione riguardante il traffico all'interno di certe zone di Bologna. Come precedentemente menzionato, soprattutto per strade lunghe o affollate, è possibile che vi siano più spire lungo la stessa carreggiata, così da poter verificare non solo quale sia la strada, ma anche il tratto maggiormente trafficato. Però, una strada può avere più di un senso di marcia a parità di nome, e diventa

meno intuitivo capire, senza andare a cliccare singolarmente tutti i segnalini che ne fanno parte per aprirne il relativo messaggio e visualizzarne le informazioni, quale spira appartenga a quale direzione. In più, è possibile che, a seguito di un incrocio, una via cambi nome, anche se la struttura di essa rimane la stessa (ad esempio rimane sempre dritta, sia prima sia dopo un incrocio, ma il suo nome cambia). Una mappa statica, per quanto possa essere ricca di informazioni, come descritto nel precedente capitolo, rimane comunque piuttosto monotona da osservare, in quanto non c'è nulla che si distingua particolarmente, a prima vista, dal restante delle informazioni, se non la dimensione dei segnalini. Serve quindi un elemento in più che richiami attenzione e renda ancora più piacevole la visualizzazione.

Si è deciso, quindi, di fornire, all'interno dei controlli, un'ulteriore impostazione. Attivando l'opzione *Animazioni segnalini*, la mappa verrà aggiornata, fornendo ad essi delle animazioni che li facciano muovere al suo interno. Il loro movimento non è casuale: infatti, in base al senso di marcia registrato dalle spire, i segnalini si sposteranno lungo quella direzione (ad esempio, se la direzione è verso Nord, l'indicatore si muoverà dal basso verso l'alto). In questo modo, verranno raggruppati tutti i segnalini di una determinata via e senso di marcia in uno unico, che si sposterà andando a toccare tutti i punti in cui è segnalata la presenza di un sensore. Per questo motivo, è possibile notare che il numero di indicatori, all'interno della mappa, sia diminuito. Questo non è un errore, in quanto, come descritto prima, se all'interno della stessa via e direzione ve ne sono molteplici, essi vengono raggruppati in uno unico. Facendo click su una di queste icone, e aprendo il relativo messaggio, quest'ultimo non rimarrà fermo: anch'esso si muoverà, rimanendo sempre in prossimità dell'icona di cui rappresenta i dati. Dopo qualche secondo, terminato il percorso del segnalino, esso si riposiziona al punto iniziale, e l'animazione ripartirà da capo, in un ciclo infinito, fino a che le animazioni non vengono disattivate nuovamente. Siccome esse sono sempre sincronizzate, ovvero ripartono tutte allo stesso momento, la velocità di movimento di un segnalino dipende dalla lunghezza della strada che rappresenta: se il

senso di partenza e di arrivo sono distanti tra loro, l'indicatore che li rappresenta si muoverà più rapidamente all'interno della mappa. Al contrario, se la distanza è breve, la sua velocità sarà molto ridotta. Effettuando lo zoom sulla mappa, i segnalini manterranno le stesse dimensioni, ma la loro velocità verrà nuovamente variata per rispettare il nuovo livello di zoom: infatti, ingrandendola, essi si sposteranno più velocemente, in quanto, dovendo comunque percorrere la strada dall'inizio alla fine, questa sarà vista con un maggior livello di dettaglio, e la distanza tra i due punti iniziali e finali aumenterà visivamente. Al contrario, rimpicciolendo la visualizzazione, le icone si muoveranno più lentamente per il motivo esattamente inverso. Nel caso in cui su una certa strada e direzione vi è una sola spira, si è deciso di rappresentare comunque l'indicatore, ma esso rimarrà statico e non verrà animato.

Siccome più segnalini vengono raggruppati in uno unico, si introduce un problema per quanto riguarda il calcolo del numero dei veicoli che transitano in quella strada. Infatti, normalmente, non tutte le spire risulteranno avere lo stesso numero di automobili circolate registrato, e potrebbe esserci un tratto della via che è decisamente più trafficato rispetto ad un altro. Chiaramente, fare la somma diretta tra questi numeri non è possibile, in quanto le auto che transitano lungo tutta la via, verranno considerate tante volte quanto il numero di spire che attraversano. Si avrebbe, quindi, la necessità di contare quante sono le automobili univoche che vi circolano, ad esempio tramite la targa. Tuttavia sarebbe una chiara invasione della riservatezza di una persona: infatti, conoscendo la targa, sarebbe possibile tracciare, senza fatica, tutti gli spostamenti di chiunque. In assenza di particolari informazioni aggiuntive, la soluzione che è stata adottata è stata quella più semplice, ovvero calcolare il rapporto tra la somma dei veicoli transitati registrati da tutti i sensori ed il numero di sensori stesso. Di conseguenza, un'auto potrebbe essere considerata meno di una volta nel caso passasse attraverso una spira e non quella successiva, ma capiterebbe anche che un'altra auto passi attraverso quest'ultima spira e non attraverso la prima, bilanciando il calcolo.

L'unico caso in cui un veicolo venga correttamente considerato è quello in cui percorre la via nella sua interezza, ma per i restanti (e più comuni) casi, si può solo eseguire un calcolo approssimativo, che ne stimi la congestione.

Calcolo del percorso del segnalino

La decisione del percorso che un segnalino dovrà seguire per essere animato si può riassumere in:

1. Raccolta di tutti i punti che appartengono alla stessa via con la stessa direzione;
2. Costruzione di una linea invisibile che, in base alla direzione, collega tutti i punti raccolti;
3. Inizio dell'animazione, facendo percorrere all'indicatore corrispondente la linea invisibile appena costruita in un ciclo infinito.

Si vuol far notare che, all'interno di questa descrizione, non si menziona in alcun modo la forma fisica della strada: tutte le informazioni che si hanno a disposizione, infatti, sono le coordinate geografiche delle spire, e non è possibile sapere con chiarezza la forma precisa della via in cui sono state installate.

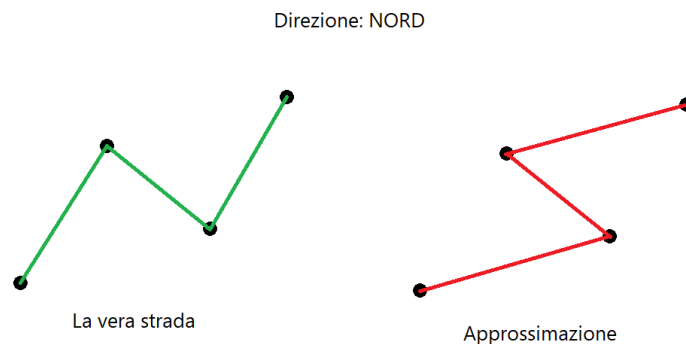


Figura 3.3: Un esempio di erronea rappresentazione della strada.

In figura 3.3 viene mostrato un esempio. I punti neri rappresentano le coordinate delle spire. La linea verde, a sinistra, è come fisicamente è strutturata la strada, tuttavia, non avendo alcun modo per saperlo, viene approssimata con la linea rossa, a destra, in quanto la direzione registrata è NORD, per cui il segnalino dovrà muoversi dal basso verso l'alto, e si cercherà di ordinare i punti di passaggio di conseguenza. Per cui, il segnalino si muoverà seguendo quest'ultima, piuttosto che la sua vera forma. Questa è l'unica grande limitazione della rappresentazione dei percorsi delle animazioni: si può dare solo una stima di come il manto stradale è stato costruito.

3.3.7 Mappa di calore

La dimensione dei segnalini fornisce, a primo impatto, informazioni sul traffico di un tratto di strada preciso, dando la possibilità di capire quale, tra tutti quelli presenti sulla mappa, è il più transitato all'interno del periodo desiderato. Questo è possibile farlo non per un'unica via, ma per tutte quelle i cui sensori hanno registrato un numero particolarmente alto di veicoli rispetto agli altri. Anche le animazioni danno il loro contributo: grazie ad esse è possibile stimare il traffico non solo di un tratto, ma di un'intera strada e dei suoi specifici sensi di marcia più facilmente, che, normalmente, è proprio ciò che interessa ad un guidatore che vuole capire in quali periodi della giornata conviene percorrere una via piuttosto che un'altra. Le informazioni già presenti sono senz'altro utili, tuttavia non sempre si è interessati a quali strade specifiche siano maggiormente congestionate. Si potrebbe avere la necessità di conoscere la viabilità di zone intere della città, considerando non solo le vie precise, ma a quante automobili circolano all'interno dell'area di cui fanno parte, per valutare, ad esempio, qual è la probabilità di rimanere bloccati nel traffico o evitare intense assunzioni di smog.

Per questo, è stato implementato un ulteriore controllo. Attivando l'opzione *Mostra mappa di calore*, infatti, è possibile visualizzarne una che, facendo affidamento sui dati che le spire hanno registrato, genera, relativamente all'arco temporale che sta venendo visualizzato, una mappa di calore, che colorerà

le varie zone di essa in base a quanto traffico è presente. Infatti, è possibile notare, tramite questa funzionalità, che le spire in cui viene segnalato il maggior transito di veicoli, ovvero quelle in cui l'icona che le rappresentano è di dimensioni maggiori, non sempre sono all'interno delle aree più trafficate. Anzi, solitamente, ne sono proprio al di fuori. Senza mappa di calore, invece, si potrebbe pensare che siano queste ultime ad essere le zone più congestionate, quando invece si può mostrare che non è così. La spira che registra il maggior numero di veicoli transitati farà parte di almeno una piccola area che verrà visualizzata come trafficata, tuttavia potrebbe trattarsi solamente di un tratto di strada mal organizzato che funge da collo di bottiglia, oppure una congiunzione per entrare ed uscire dalla città.

Attivando le animazioni, la mappa di calore non si muoverà con esse: rimarrà sempre fissa, segnalando, così, i tratti di strada più congestionati, piuttosto che tutta la via. Così facendo, nonostante il nuovo segnalino ne raggruppi molteplici, rendendo più difficoltoso riuscire a capire quale tratto stradale sia potenzialmente da evitare, la mappa di calore prenderà in considerazione ogni singola spira prima del suo raggruppamento, al fine di tingere la mappa con i colori corretti. Si potrà, quindi, distinguere quale zona specifica dell'intera via sia più trafficata rispetto alle restanti.

Lo spettro di colori utilizzato per visualizzare la situazione del traffico è quello classico usato, normalmente, per le mappe di calore: i punti in cui viene registrato il passaggio di pochissime auto vengono tinti di un leggero azzurro, per poi passare al verde, giallo, arancione e, infine, le zone con più concentrazione di automobili verranno colorate di rosso. Per far sì che i colori utilizzati non siano troppo invasivi con la visualizzazione della mappa geografica, si è data una leggera trasparenza ad essi, di modo che siano chiaramente visibili, ma è possibile comunque capire, guardando attentamente, i nomi, forme delle strade e punti di riferimento sui quali sono stati posizionati certi tipi di colori. Come tutto il resto, cambiando il periodo di visualizzazione della mappa e, potenzialmente, anche le ore interessate, la mappa di calore si aggiornerà di conseguenza, modificando i colori in base ai nuovi dati.

Ogni spira che risulta avere più di zero veicoli che vi hanno passato attraverso, viene considerata nel calcolo dei colori di cui tingere la mappa e, come precedentemente menzionato, più è grande il numero registrato, più sarà intenso il colore che la rappresenta. Prese singolarmente, quindi, ognuna avrà un colore che la distingue. Tuttavia, in una mappa come questa, esse possono essere vicine le une alle altre, per cui i loro colori si possono intersecare. Infatti, questi ultimi si influenzano l'uno con l'altro: due o più spire vicine che registrano un numero medio di veicoli transitati avranno, singolarmente, un certo colore, che, intersecandosi, ne daranno vita ad un altro, più intenso, in base alla somma del numero di auto registrate e alla distanza fisica dei sensori. In questo modo, sarà possibile rilevare delle zone piuttosto trafficate che, se considerate le singole spire, potrebbero non risultare. Però, questo approccio porta con sé un problema, ovvero quello della decisione del raggio di influenza. Si consideri ogni spira sulla mappa come un punto con il proprio colore ed un certo raggio di influenza, all'interno del quale, se si interseca con un raggio appartenente ad un'altra spira, essi si fondono in un nuovo, potenzialmente più acceso, colore. Non sempre potrebbe risultare, però, in una mappa le cui informazioni diventano attendibili. Infatti:

- Un raggio di influenza troppo grande significa che tutti i punti si influenzano l'un l'altro, anche se appartengono a strade e tratti stradali lontani tra loro. Questo significa che tutta la mappa verrà tinta arancione e rosso, risultando così come completamente trafficata;
- Un raggio di influenza troppo piccolo significa che nessun punto ha a che fare con gli altri. Questo comporta che la mappa diventi solamente un insieme di piccoli puntini colorati sparsi in giro, e, oltre che ad essere esteticamente spiacevole da vedere, diventa poco utile, in quando vi sono delle spire che effettivamente si devono influenzare tra loro, come, ad esempio, quelle posizionate ai lati di un incrocio.

Si è cercato quindi di adottare una soluzione intermedia, ovvero l'impostazione di un raggio di influenza tale da considerare i punti vicini, escludendo

quelli troppo lontani. La visualizzazione risultante si è considerata, quindi, accettabile: tuttavia, per quanto si è provato a ridurre gli effetti indesiderati, tentando diverse possibilità di raggi di influenza, non necessariamente è priva di inconsistenze. Il numero di queste ultime, però, è notevolmente ridotto.

3.4 Errori di visualizzazione

La mappa cerca di fornire tutte le informazioni necessarie per quanto riguarda il traffico passato, sia per vederne l'andamento nel tempo, sia per trarne conclusioni, sia per provare a predire degli eventi futuri, come la strada più trafficata in un certo periodo di tempo. È anche utile nel caso l'utente sia semplicemente curioso di conoscere, avendo i dati alla mano, il numero preciso di veicoli che circolano all'interno della città, in quanto è difficile dare anche solamente una stima di essi se non si è soliti circolare tutto il giorno. Tuttavia, come tutte le pagine web e programmi, non si può dire che essa sia totalmente esente da errori di visualizzazione. Si è cercato di mostrare e dare una spiegazione sulle limitazioni che le soluzioni ai problemi adottati possono presentare, ma questi errori, a volte, non dipendono da come i dati vengono visualizzati. In alcuni casi, dipendono proprio dai dati stessi, che per qualche motivo sono stati ottenuti e pubblicati con delle problematiche più o meno gravi. Si cercherà quindi di descrivere gli errori più comuni riscontrati durante un'attenta analisi sia dei dataset forniti dal portale del comune di Bologna, sia come questi possono influenzare la visualizzazione sulla mappa. Il primo errore è dato dall'affidabilità dei dati. Infatti, durante il caricamento, viene fornita una stima della precisione dei dati che una spira ha raccolto, da 0 a 100%, invisibile all'utente. Mentre nella maggior parte dei casi essi vengono dichiarati come totalmente accurati, si è notato che in altri non è così, andando dal 90% fino ad un minimo dello 0%. Dati con quest'ultimo fattore di precisione appartengono con tutta probabilità a sensori rotti o difettosi, incapaci di operare correttamente e far sì che i dati visualizzati siano accurati. Si è comunque deciso di mostrare tutti i dati indistintamente, in

quanto questo problema, per quanto presente, è una vasta minoranza rispetto alla precisione di tutti gli altri dati raccolti, e si è pensato che non possa causare grossi problemi di visualizzazione, né che possa compromettere l'affidabilità dell'intera mappa.

Un altro problema che è importante riconoscere è il posizionamento delle spire. Esse sono concentrate in particolar modo nel centro della città ed i suoi dintorni, ed è chiaramente possibile distinguere, tra queste zone, quelle maggiormente trafficate. Però, a mano a mano che ci si allontana dalle aree più abitate, la concentrazione di spire diventa sempre più rarefatta, fino ad arrivare ad un limite, oltre il quale cessano di essere presenti. In queste aree, che sono sempre appartenenti al comune di Bologna, è impossibile dare una stima della circolazione dei veicoli in quanto non vi sono informazioni a riguardo. Ciò non significa che, però, quelle strade siano completamente vuote: vi è sicuramente un passaggio di veicoli, con alta probabilità che sia molto inferiore rispetto al centro, ma, non essendo stati installati dei sensori sotto di esse, si possono solo fare delle stime mentali, considerando la zona in cui sono poste (ad esempio se rurale o industriale) ed ai dati rilevati dalle spire più vicine ad esse.

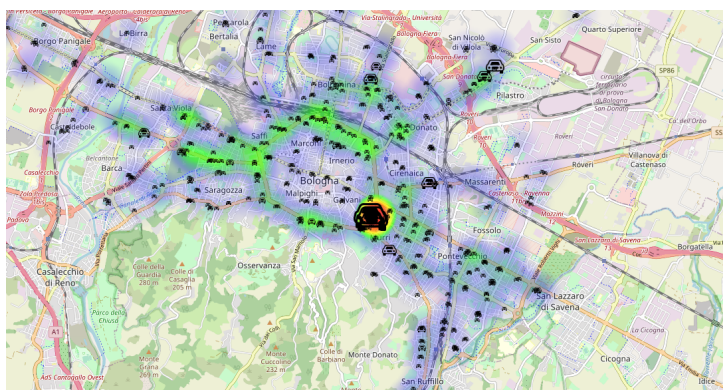


Figura 3.4: Mappa in data 1 Dicembre 2022. Questa visualizzazione rappresenta un problema, in quanto tutte le auto sono concentrate in un unico punto. Questa non è l'unica data in cui è presente l'errore, è solamente una delle diverse.

In figura 3.4, è possibile vedere l'ultimo problema più rilevante, ed è quello che maggiormente influenza la visualizzazione dei dati. Vi sono alcuni giorni in cui i dati vengono rappresentati in questo modo, ovvero sono tutti concentrati in un unico punto, ed il resto della città sembra disabitato. Questo non è un problema legato all'applicazione web, che ha solo il compito di visualizzare i dati già presenti. Questo errore è relativo direttamente ai dati: a causa di motivi sconosciuti, quelle spire hanno registrato un traffico insolitamente elevato, mentre le altre, se non insolitamente basso, che non si avvicina minimamente al traffico presente in quei specifici punti. Questo tipo di visualizzazione è decisamente anomala, ed è solamente possibile fare delle ipotesi a riguardo:

- Una parte dei dati raccolti da tutte le spire, per qualche motivazione, è stata aggiunta non al sensore che li ha rilevati, ma ad alcuni specifici a causa di errori;
- Sono state aggiunte le auto più di una volta al sensore che le ha registrate;
- Si tratta di outliers, ovvero degli scenari in cui è possibile che vi siano zone decisamente più trafficate rispetto alle altre, come, ad esempio, casi in cui artisti famosi, come cantanti, visitano la zona per effettuare i propri concerti, per cui un numero insolitamente alto di veicoli transiteranno attraverso un determinato tratto di strada.

Conclusioni

Il progetto, basato su dati geografici, mira ad un facile utilizzo ed interpretazione per fornire all'utente una base per conoscere la situazione del traffico di Bologna. Si è pensato anche alla portabilità dei dati, e ad un modo per rendere la loro visualizzazione in mappa accattivante, con modi ed interazioni interessanti che attraggano l'attenzione, senza confondere l'utente riempiendolo di informazioni. A tal proposito, si può dire che si sono raggiunti dei risultati soddisfacenti, soprattutto grazie alla grande mole di dati a cui è stato possibile fruire gratuitamente per condurvi uno studio approfondito. Dato che essi forniscono molte informazioni, suddivise non solo in base al giorno, ma anche alle ore specifiche, oltre che a dati relativi ai sensori stessi che li hanno prodotti, ne è stato fatto un utilizzo intensivo, seguendo i principi degli Open Data e della facile visualizzazione. Sono state seguite le tecnologie attualmente più diffuse in ambito web, come linguaggi di Markup e di scrittura, per dar vita ad un progetto piacevole da vedere, i cui dati che vi sono inseriti siano di facile reperimento. Grazie all'applicazione implementata, visualizzabile direttamente sul proprio browser, è possibile attingere a queste informazioni ovunque ci si trovi e, potenzialmente, possono essere facilmente aggiornate con delle nuove. Si è cercato di dare diverse chiavi di lettura alla mappa presentata, dando la possibilità all'utente di vederla e riuscire ad analizzarla in modi diversi, in base alle sue esigenze. Proprio in base a ciò si è fatto particolare attenzione ai metodi di visualizzazione ed i modi in cui i dati sono presentati. Il risultato è stata la visualizzazione solamente delle parti più interessanti, dando molta importanza non al testo,

che comunque resta un aspetto da non sottovalutare, ma a colori e icone. In questo modo è possibile trarre informazioni rilevanti a prima vista, spostando l'attenzione dell'utente verso zone precise della mappa senza menzionarlo esplicitamente. Anche le animazioni, che sembrano dare solamente un ulteriore impatto estetico, sono risultate essere efficaci nell'indiretta trasmissione di informazioni a chi le visualizza, evidenziando ancora di più la congestione delle singole corsie.

Per il raggiungimento degli obiettivi, sono state utilizzate tecnologie e librerie aggiuntive in conformità ai principi inizialmente menzionati per l'aggiunta di funzionalità non banali, ponendo sempre l'utente in prima posizione, favorendo la facile interazione con l'applicazione, accompagnata da un menù intuitivo con voci il più possibile autoesplicative e di semplice modifica, al fine di focalizzare l'attenzione al contenuto principale, ovvero le informazioni visualizzate sulla mappa, piuttosto che a come interagirvi.

In questo ambito, ci si è concentrati solamente sul traffico della città di Bologna. Tuttavia, se si intende pensare al futuro, questa applicazione non è stata progettata per essere specifica per essi. Infatti, essa è in grado di elaborare qualsiasi tipo di dato, fermo restando che quest'ultimo contenga i campi fondamentali dai quali recuperare le informazioni. Si potrebbe, quindi, pensare ad un'espansione dell'area considerata, che comprenda non solo la città di Bologna, ma anche altri territori, che possono essere sia vicini e confinanti, ma anche zone completamente separate. Si è cercato, infatti, di implementare un'applicazione che sia la più generale possibile e ridurre al minimo i cambiamenti necessari per l'aggiunta di funzionalità. Riguardati queste ultime, se si intendesse aggiungerne di nuove, siccome nessuna dipende dalle altre, non si avrebbe la necessità di modificare il comportamento di alcuna di esse per accomodare gli aggiornamenti, se non per la correzione degli eventuali errori passati inosservati. Le applicazioni simili attualmente disponibili sono per la maggior parte proprietarie e richiedono dei costi non indifferenti per essere utilizzate. Nonostante esse siano chiaramente ricche di funzionalità, quest'applicazione mira ad essere semplice, funzionale e gratuita, ed in tutti i

casi non ne è stata trovata nessun'altra che offre una funzionalità di rotazione dei giorni simile a quella presente. Il limite dell'implementazione di nuove ed uniche interazioni con la mappa che possono essere offerte all'utente è solo dato dalla creatività.

Si è data la priorità alla visualizzazione della mappa per la stragrande maggioranza di utenti in grado di poter attingere alle informazioni contenute in essa con facilità. Si potrebbe pensare anche ad una visualizzazione migliore per tutti gli utenti con qualsiasi tipo di disabilità: anche se si è cercato il più possibile di fornire una grafica con colorazione e modalità di accesso il più semplice possibile, rendendola più accessibile supportando l'utilizzo di un lettore di schermo, un'ulteriore funzionalità sviluppabile può essere la libertà per gli utenti di poter cambiare i colori della mappa di calore, decisione della grandezza dei testi, o fornire altri modi per ridurre la quantità di segnalini sullo schermo, per potersi concentrare meglio su quelli più importanti (ad esempio quelli più trafficati). In merito a ciò si può pensare ad una più stretta collaborazione con gli utenti finali che andranno di fatto ad utilizzare l'applicazione. Ciò può essere fatto eseguendo dei test insieme a loro, prendendo nota delle loro critiche e suggerimenti per migliorare l'esperienza d'utilizzo, implementando le modifiche e ripetendo nuovamente i test, fino a che essi non si sentano soddisfatti con il prodotto.

Bibliografia

- [1] Sanford L. Moskowitz. *Advanced Materials Innovation: Managing Global Technology in the 21st century*. Wiley, 2016. ISBN 9780470508923.
- [2] Viviane Callier. The open data explosion. *The Scientist*, 2019. URL <https://www.ncbi.nlm.nih.gov/search/research-news/6880/>.
- [3] Simon Chignard. A brief history of open data. *ParisTech review*, 2013. URL <https://www.paristechreview.com/2013/03/29/brief-history-open-data/>.
- [4] Open Knowledge Foundation group. Open definition 2.1 - defining open in open data, open content and open knowledge. *Open Knowledge Foundation*, 2022. URL <https://opendefinition.org/od/2.1/en/>.
- [5] Albert Meijer Erna Ruijter, Stephan Grimmelikhuijsen. Open data work: understanding open data usage from a practice lens. *International Review of Administrative Sciences*, 2020. URL <https://doi.org/10.1177/0020852317753068>.
- [6] Charles W. Uth. Data management and open data. *Illinois Tech*, 2023. URL <https://guides.library.iit.edu/data>.
- [7] Stephanie Jamilla Melissa Edmiston, Stephanie Coker. The pros and cons of open data. *MERL Center*, 2022. URL <https://merlcenter.org/guides/pros-and-cons-of-open-data/>.

-
- [8] World Bank Group. Open data essentials. *The World Bank*, 2019. URL <https://opendatatoolkit.worldbank.org/en/essentials.html>.
- [9] Open Data Enterprise. Data standards. *U.S. Open Data Toolkit*, 2019. URL <https://www.usopendatatoolkit.org/best-practices/data-standards>.
- [10] W3C. Rdf. Technical report, W3C, 2023. URL <https://www.w3.org/RDF/>.
- [11] Peter Suber. Open access overview. 2006. URL <https://web.archive.org/web/20070519103647/http://www.earlham.edu/~peters/fos/overview.htm>.
- [12] Peter A. Johnson. The cost(s) of geospatial open data. *Wiley Online Library*, 2017. URL <https://onlinelibrary.wiley.com/doi/full/10.1111/tgis.12283>.
- [13] Sara Nodehi Putu Hadi Purnama Jati, Yi Lin. Fair versus open data: A comparison of objectives and principles. *MIT Press Direct*, 2022. URL <https://direct.mit.edu/dint/article/4/4/867/112737/FAIR-Versus-Open-Data-A-Comparison-of-Objectives>.
- [14] Open Knowledge Foundation group. Open data commons attribution license (odc-by) v1.0 - open data commons: Legal tools for open data. *Open Knowledge Foundation*, 2022. URL <https://opendatacommons.org/licenses/by/1-0/>.
- [15] European Commission. Open data licensing. *Data Europa*, 2023. URL <https://data.europa.eu/en/open-data-licensing>.
- [16] U.S. General Services Administration. About data.gov. Technical report, U.S. Government, 2023. URL <https://data.gov/about/>.

-
- [17] Ekblaw Ariel Green Ben, Cunningham Gabe. *Open Data Privacy*. Cyber Harvard Library, 2017. <https://dash.harvard.edu/bitstream/handle/1/30340010/OpenDataPrivacy.pdf>.
- [18] Tinhold Dinand. *The Open Data Economy - Unlocking Economic Value by Opening Government and Public Data*. Capgemini, 2013. https://www.capgemini.com/us-en/wp-content/uploads/sites/4/2017/08/the_open_data_economy_unlocking_economic_value_by_opening_government_and_public_data.pdf.
- [19] Open Data Institute. Research: The economic value of open versus paid data. *Open Data Institute*, 2016. URL <https://theodi.org/article/research-the-economic-value-of-open-versus-paid-data/>.
- [20] IBM. What is data visualization? Technical report, IBM, 2023. URL <https://www.ibm.com/topics/data-visualization>.
- [21] James T. Guthrie. Psychology and perception of colour and shape. *ResearchGate*, 2003. URL https://www.researchgate.net/publication/241059972_Psychology_and_perception_of_colour_and_shape.
- [22] SAS Institute. Data visualization: What it is and why it matters. Technical report, SAS Institute, 2023. URL https://www.sas.com/en_us/insights/big-data/data-visualization.html.
- [23] Scott Bernato. Visualizations that really work. *Harvard Business Review Magazine*, 2016. URL <https://hbr.org/2016/06/visualizations-that-really-work>.
- [24] Josh Taylor. Why we should use animation in data visualization. *Towards Data Science*, 2022. URL <https://towardsdatascience.com/why-we-should-use-animation-in-data-vizualisation-6004fb400014?gi=fc23974c5605>.

-
- [25] Andy Cotgreave. The benefits of animating your data visualisations. *Information Age*, 2020. URL <https://www.information-age.com/benefits-animating-data-visualisations-15602/>.
- [26] Berkley Advanced Media Institute. Color theory basics for data visualization. *Berkley Advanced Media Institute*, 2023. URL <https://multimedia.journalism.berkeley.edu/tutorials/data-visualization-color/>.
- [27] CR Ferreira. Two simple steps to create colorblind-friendly data visualizations. *Towards Data Science*, 2020. URL <https://towardsdatascience.com/two-simple-steps-to-create-colorblind-friendly-data-visualizations-2ed781a167gi=43902fc3f576>.
- [28] W3C. How to meet wcag (text). Technical report, W3C, 2023. URL <https://www.w3.org/WAI/WCAG21/quickref/#text-equiv>.
- [29] W3C. How to meet wcag (keyboard). Technical report, W3C, 2023. URL <https://www.w3.org/WAI/WCAG21/quickref/#keyboard-accessible>.
- [30] Marin P. Allen. Picture this! *National Institutes of Health*, 2021. URL <https://www.nih.gov/about-nih/what-we-do/science-health-public-trust/perspectives/picture>.
- [31] W3C. How to meet wcag (seizures and physical reactions). Technical report, W3C, 2023. URL <https://www.w3.org/WAI/WCAG21/quickref/#seizures-and-physical-reactions>.
- [32] Stephen Goldsmith. As the chorus of dumb city advocates increases, how do we define the truly smart city? *Data-Smart City Solutions*, 2021. URL <https://datasmart.hks.harvard.edu/chorus-dumb-city-advocates-increases-how-do-we-define-truly-smart-city>.

-
- [33] Iulian Sandu Popa Juan Pan, Mohammad A. Khan. Proactive vehicle re-routing strategies for congestion avoidance. Technical report, New Jersey Institute of Technology, 2012. URL <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=053e1f65d081fc16dfd29957ae58ec82b0651a08>.
- [34] Keith O'Brian. What is smart transportation? *IBM*, 2023. URL <https://www.ibm.com/blog/smart-transportation/>.
- [35] Guillaume Leduc. Road traffic data: Collection methods and applications. Technical report, Ministère de la Transition Ecologique, 2008. URL https://www.researchgate.net/publication/254424803_Road_Traffic_Data_Collection_Methods_and_Applications.